West Bank and Gaza

NETHAM

Rule of Law Program

Justice and Enforcement

DFD-I-01-04-00173-00

Implemented by DPK Consulting, a division of ARD, Inc.

Information Technology
Vision Document on Mizan 2 at the Palestinian Courts

Al-Whaidi Building, 1st Floor
Ramallah
Tel: 02-2974516/7
Fax: 02-2972230

1 VISION STATEMENT 1

1.1 GOALS 1
1.2 GENERAL BENEFITS OF COURT AUTOMATION 1
1.3 BUSINESS REQUIREMENTS OF COURT AUTOMATION 2
1.3.1 Register Automation and Administrative Support 2
1.3.2 Case File Automation 2
1.3.3 Electronic communication of court data with court business partners 3
1.3.4 Internal communications: Intra-court 3
1.3.5 Internal communications: Inter-court 3
1.3.6 External communications 4
1.3.7 Improved management information: 4

2 MEZAN ..... ERROR! BOOKMARK NOT DEFINED.

2.1 MEZAN FUNCTIONALITIES 5
2.2 MEZAN ARCHITECTURE 5
2.3 CURRENT SITUATION 6
2.3.1 Case-Based Case Tracking System 8
2.3.2 Functional Evaluation of MEZAN 8
2.3.3 Technical Evaluation of MEZAN 9
2.4 SUMMARY RECOMMENDATIONS 11
2.4.1 Enhanced Function and Feature 11
2.4.2 Redevelop the current MEZAN architecture to improve maintainability; 13
2.5 FUTURE MEZAN (MEZAN V2) 15
2.5.1 MEZAN Version 2 Objectives 15
2.5.2 Mezan Version 2 Requirements 15
2.5.3 Solution Requirements 15
2.5.4 Mezan Application Architecture Considerations 19

3 ATTACHMENT 1 - MEZAN CASE TRACKING SYSTEM FUNCTIONAL STANDARDS SCORECARD 22

3.1 FUNCTIONS 23
3.1.1 Case Initiation and Indexing Function 23
3.1.2 Docketing and Related Recordkeeping Function 32
3.1.3 Scheduling Function 41
3.1.4 Document Generation and Processing Function 49
3.1.5 Calendaring Function 53
3.1.6 Hearings Function 55
3.1.7 Disposition Function 57
3.1.8 Execution Function 61
3.1.9 Case Close Function 62
3.1.10 General Accounting Function 64
3.1.11 Accounting - Front Counter and Cashiering Function 67
3.1.12 Accounting - Back Office Function 71
3.1.13 Accounting - General Ledger Function 75
3.1.14 File, Document, and Property Management Function 77
3.1.15 Security Function 82
3.1.16 Management and Statistical Reports Function 84
3.1.17 List of Code Translation Tables 90

4 ATTACHMENT 2 – TECHNOLOGY ARCHITECTURES 95

4.1 THE CLIENT/SERVER COMPUTING PARADIGM 95
4.2 ELEMENTS OF A TECHNOLOGY ARCHITECTURE 95
4.2.1 Client Computers 95
4.2.2 Server Computers 96
4.2.3 Databases 96
4.2.4 Networks 96

4.3 COMBINING ELEMENTS INTO ARCHITECTURES 97
4.3.1 Advantages of centralization 97
4.3.2 Disadvantages of Centralization 98
4.3.3 Advantages of Decentralization 98
4.3.4 Disadvantages of Decentralization 99
4.3.5 Architecture independent considerations 99
4.4 DECISION CRITERIA: QUALITY OF SERVICE 99
4.4.1 Availability 100
4.4.2 Response Time 100
4.4.3 Functional Content 100
4.4.4 Quality of Service Targets 101
4.4.5 Response time: 101
4.4.6 Functional Content: 101
4.5 TO DECENTRALIZE OR CENTRALIZE? 101
4.6 SOFTWARE ARCHITECTURE 104
4.6.1 Application Architecture 105
4.7 APPLICATION ARCHITECTURE ELEMENTS 105
4.7.1 Application Layering 105
4.7.2 User Interface Layer 107
4.7.3 Business Layer 108
4.7.4 Data Layer 108
4.8 APPLICATION ARCHITECTURE COMPONENTS 109
4.8.1 Core Framework 109
4.8.2 Data Access Application Block 109
4.8.3 Exception and Error Management 110
4.8.4 Logging and Instrumentation 110
4.8.5 Caching 111
4.8.6 State Management 112
4.8.7 Security 113

1. VISION STATEMENT

Establish comprehensive information technology portfolio and program management vision that includes planning, case management systems, infrastructure, information standards, and effective communications in order to assist the courts in business practice reengineering, development, provisioning, implementation, change management and use of technology in the courts to enhance and improve the access of justice for all in an efficient, transparent and accountable way.

1.1 Goals

- Support courts of various size, demographics and jurisdiction
- Provide systems that are adaptable given legislative and business practice change
- Increase accountability in court practice
- Increase public opinion of the Judiciary through fair and equal access for all
- Establish standard process and procedure that will ease the burden of automation cost on the judiciary
- Build automation solutions that are easily supported and maintained, reducing the cost of IT activities on the Judiciary
- Integrate with other Justice agencies to streamline and reduce inefficiencies in the entire Judicial system
- Build automation solutions that are based on industry standard, best practice and contemporary architectures, infrastructures and technology.
- reduce case backlog by streamlining business process through automation

1.2 General Benefits of Court Automation

Automation introduces new technology to support court business practices. Essentially, computers are used to replace paper registers and replace paper file folders as the principal repository of case data and documents. Paper (as a technology) has benefits and constraints; so do computers. Proper use of a technology maximizes the benefits while mitigating the constraints.

The business advantages of computers over paper record-keeping derive from three principal facts:

1. Data stored on computers may be instantaneously copied to (replicated on) other connected computers. Thus information on a computer register may be viewed and used simultaneously by an unlimited number of people, in contrast to a paper register which may be viewed by only one person at a time;
2. Data stored on computers may be instantaneously transmitted over long distances via a network. Thus data resident on a court computer may be viewed and used by geographically remote persons, for example judges at home, attorneys in their offices, police officers, corrections officers, process servers, analysts at the SJC, and the general public. In contrast, paper court documents must be physically located, copied and manually transported to remote locations, requiring hours or days and considerable staff work to accomplish.
3. Data stored on computers may be instantaneously selected, sorted, tabulated, and subject to arithmetic calculation yielding novel statistical insights into court operations for management decision-making. Such operations are practically impossible from data on

paper registers due to the overwhelming cost in staff resources and time to extract, tabulate, and analyze data.

The advantages of computers are so overwhelming at the present time that the West Bank has joined most worldwide court systems (and indeed most other business and government enterprises) in introducing this technology to support its operations. The specific applications to court business processes of the improved capabilities of automation over paper—the business requirements for court automation—are explored below.

1.3 Business Requirements of Court Automation

Business requirements for court automation are described according to the following classifications[1]:

1. Register Automation and Administrative Support
2. Case Dossier Automation
3. Electronic Communications:
    a. Intra-court
    b. Inter-court
    c. Extra-court
4. Management information:
    a. Court-specific
    b. National

1.3.1 Register Automation and Administrative Support

Automating the summary case data kept on central court registers increases the productivity of court staff performing the routine clerical and administrative tasks necessary to advance cases from filing to disposition. Tasks subject to automation include recording register data on the computer instead of on paper registers when a new case is filed, when a hearing is scheduled, when a witness is summoned, when expert testimony is ordered, when notices are prepared and sent, when judicial orders and judgments are entered, etc.

In general, computers will replace paper registers, produce notices and routine case correspondence faster and with less effort, track obligations of both parties and court, alert courts to unmet obligations, track and account for fee and fine receipts, schedule hearings and other events, and support judges to research the law and prepare good decisions and judgments more efficiently.

The current MEZAN case tracking system automates some of these processes.

1.3.2 Case File Automation

Paper court documents are presently kept in a case file folder (dossier) initiated when a case is originally filed in a court. In the future, each document may be in an electronic form, stored on the computer, and displayed on demand when and where needed. In electronic form, the case dossier may by accessed simultaneously by multiple persons: the judge, attorneys for parties, parties, court clerks charged with maintaining the dossier (adding, replacing documents, etc.) and other authorized users: appeals courts, execution department, police, prosecutor, corrections, etc.

No longer is access restricted to a single, unique folder of paper which must be carefully guarded and transported and may be used by only one person at a time. Automation of the dossier contributes to security against document loss or damage: electronic copies may be easily created and stored in multiple locations (e.g. Court and SJC) to safeguard against fire or other natural disasters, theft and other loss.

[1] For a more detailed discussion of business requirements for court automation see Functional Standards for Court Automation available on-line at http://www.ncsconline.org/D_Tech/standards/

When needed, paper copies of individual documents may be printed instantly, certified as official, and distributed to requestors. Indeed, electronic copies of an individual document, or the entire dossier, may be transmitted instantaneously virtually anywhere in the world.

The current MEZAN case management system can store electronic documents (in addition to summary case data) and effectively create an electronic case dossier. This functionality is not presently used.

1.3.3 Electronic communication of court data with court business partners

The previous example introduces a powerful new capability to the courts: instantaneous electronic transmission of court data within the court, between courts, and to external court business partners.

1.3.4 Internal communications: Intra-court

Court personnel must communicate with each other on a regular basis. The electronic case registers and the electronic case file built into the automated case management system facilitate communication of case status without physically viewing central paper registers, or physically transporting paper case dossiers or memos.

For many other communications, local e-mail can be an efficient and effective replacement for circulating memoranda and face-to-face meetings.

Case summary data (representing data on paper registers) is currently electronically communicated between court staff using MEZAN (intra-court). Missing is the ability to coordinate and transfer data between courts in the same location or across court locations.

1.3.5 Internal communications: Inter-court

Courts communicate with each other. When a case is appealed, lower court case information is transferred to Appeals courts; on remand, appeal court case information is transferred to the originating lower court. In a paper-based system, the paper dossier is transmitted and party identification, decision and judgment data is manually re-entered into the receiving court registers; this process may require several days. With electronic registers, the transfer and recording of data in the receiving court can be instantaneous, avoiding clerical work and time for data entry and ensuring accurate transcription. Similarly, the entire lower court case file record may be electronically communicated to appeals judges for inspection and analysis.

The electronic system can easily link and keep track of multiple case identifiers used in different courts (different case numbers). For perhaps the first time, the time required for the court *system* (first instance, appeals, execution) to <u>completely</u> resolve a citizen complaint may be determined and analyzed.

Court personnel must also communicate with each other on a regular basis, both within a court, between courts, and between courts and the SJC. Electronic messaging in the form of personal and group broadcast e-mail facilitates such communication and provides unique capacity to rapidly and effectively communicate court and national policy and procedural information.

Inter-court communication can provide judges access to the decisions and judgments of peer-level or appeals courts in the West Bank, improving the quality of judicial decision-making and possibly helping to reduce the frequency of appeals.

Inter-court transfer of case data is not currently supported by MEZAN, but the potential exists, subject to further development. Inter-court e-mail is not widely supported.

1.3.6 External communications

E-services for inquiry: Electronic communication opens the possibility for on-line, public and attorney access to court information via the Internet, and avoids physical trips to the courthouse to access and view information (hearing schedules, new case file documents, etc.). This capability reduces congestion in the courthouse, reduces staff time required to serve public and attorney inquiries and requests, and contributes to a more efficient organization.

E-services for filing: Electronic communication opens the possibility for electronic filing of documents (pleadings, etc.) by prosecutors and attorneys, rather than physically bringing them to the court for inclusion in the case dossier. This further reduces court visits and congestion, and reduces clerical staff time required to input case data and process and maintain case files.

E-notification: Electronic communication opens the possibility for electronic notification of parties (e.g. attorneys) by the court to attend hearings (summons), and electronic transmission of other court documents that must be served (e.g. decisions, judgments, warrants, etc.). This speeds service and can contribute to shorter case processing times.

Integrated Justice System: courts are one component of a larger criminal justice system including police, (prosecutors), and corrections (detention and release-supervision). Electronic communications between agencies may be used for arrest notification, warrant service, scheduling of defendant appearance in court hearings, scheduling testimony of police officers, transmission of official documents, and court notification of detention and sentences served. Public safety is increased by efficient and effective communication between the criminal justice system components.

Legal Research: External communications can provide judges access to international legal research materials and databases, keeping judges informed of current legal trends and providing knowledge resources for specialized legal fields (e.g. admiralty law, international trade laws and treaties, etc.).

The MEZAN project has planned a network to connect the courts and provide these inter-court and external communications capabilities however the information may not be real-time and may also not be usable in transactional systems as the data does not flow two ways..

1.3.7 Improved management information:

Electronic data stored in a database opens the possibility to generate reports of summary information on court activities that are practically impossible to obtain from paper registers. No longer must register data be collected and transcribed by hand, and file folders sampled and searched to obtain case data, then painstakingly analyzed and calculated to produce reports for court decision makers. Rather, electronic register data can be efficiently selected, combined, sorted, statistically analyzed, and distributed to decision-makers virtually instantaneously. Data from multiple court locations may be compared and combined to generate regional and national statistics. Court presidents and SJC personnel may make future decisions using electronic reports based on complete, accurate, and up-to-date data.

Electronic reports offers the Court President an opportunity to monitor and manage a much wider range of court activities including detailed analysis of pending caseloads, time in process from filing to disposition, postponement rates, judicial and administrative workload and efficiencies. Electronic reports offer the SJC much greater insight into court operations for the purpose of more rapid and appropriate resource allocation, inspection functions, and improved assessment of the impact of proposed legislation affecting court functions or operations.

MEZAN provides many types of management reports for local court use and offers the capability to develop additional reports based on data stored on the computer. However, due to unconstrained data entry, decentralized code table management and missing statistical data attributes, many key performance indicator reports are not possible in MEZAN.

1.4 MEZAN Functionalities

Mezan is a broadly functional case tracking system developed in the West Bank by DPK consulting under the USAID funded Rule of Law program in the West Bank and Gaza. The system is composed of eleven distinct software programs that provide core business function to a number of court jurisdictions and departments. The following bullet points reflect at a high level the function and feature of the program. Specifically each of the Mezan software products:

- records initial case data upon filing: this includes establishing a case identifier, recording basic classification data, and recording the names and data about the parties to case (e.g. plaintiff, defendant, attorneys);
- indexes case data and supports methods to identify and locate cases based upon partial case information (e.g. party name);
- records case-related fees paid by parties;
- records documents (pleadings, evidentiary enclosures, etc.) received by the court from parties and added to the case file folder;
- prepares documents (notices, orders, judgments, etc.) generated by the court;
- schedules hearings, maintains a hearing calendar for judges, and records hearing minutes;
- keeps track of the physical location of the case file folder and maintains a history of case file movements;
- records case closure and some post-disposition activity (e.g. appeal filed);
- generates reports useful to court management to monitor workload and court performance including basic caseload and case clearance statistics required of each court by the SJC.

The application identifies individual users (court staff) and assigns different levels of database access and update privileges based upon user classification and court. User logins are unique for each court. If a user works in more than one court they will be given a separate user-id and password combination for each court for which they work.

The system allows basic case data to be updated to reflect changes as the case moves to disposition, i.e. the addition/withdrawal of parties, changes in attorneys representing parties, etc.

1.5 MEZAN Architecture

The application programs are written in Visual Basic 6 (VB 6) and the database is hosted in a SQL-Server Database Management 2000 System (DBMS). The system is deployed as a two-tier client-server architecture with desktop PCs acting as a client connected by a local area network to database servers in each courthouse. This architecture provides a rich client side interface with full windows function. Data is moved from the client to the database server as the user navigates from field to field and form to form.

MEZAN is designed to be a “data-driven” system, i.e. much of its behavior is controlled by the code tables containing user-entered data. Some data is entered only once, (e.g. Court Name) when the system is placed in service, while other data may be entered as needed during system operation to adapt to operational change (e.g. the name of a new judge, a new case type sub-classification). This provides a great deal of flexibility to adapt MEZAN to change over time within each court where implemented.

Initially, the system (application program set/database) was implemented in one court of jurisdiction, creating the first variant of MEZAN.

Thereafter, the system was adapted to support cases filed in other legal and geographic jurisdictions via software alteration facilitating unique business processes of those jurisdictions. The adaptation was accomplished by slightly modifying the initial set of computer programs and

slightly modifying the database design to accommodate minor requirement changes needed to support the processing of the new jurisdiction. The result was a system independent of the original case tracking system, i.e. a system which uses an independent set of computer programs and an independent database. Thus, effectively, a *variant* of MEZAN was created.

As the project progressed, this process was repeated to support automated case processing in the Conciliation Court jurisdictions (Civil and Criminal), the Appeals Court jurisdictions (Civil and Criminal), the Prosecution Department (criminal investigation), the Attorney General Office, the Notary Public Office, and the Execution Department. In each case, the basic MEZAN system (application program set/database) was modified slightly to reflect unique operating requirements of each environment.

1.6 Current Situation

The result today is eleven independent *variants* of MEZAN, each slightly different, operating in different West Bank justice organizations as described in the following table.

Court	Conciliation		First Instance		First Instance Appeals		Appeals	Cassation	High Court	Notification Department	Notary Public	Civil Judgment
	Civil	Criminal	Civil	Criminal	Civil	Criminal						
Ramallah	Ci	Cr	Ci	Cr	FACi	FACr	A	C	HC	*NA*	NP	CJ
Jenin	Ci	Cr	Ci	Cr	FACi	FACr				*NA*	NP	CJ
Tulkaram	Ci	Cr	Ci	Cr	FACi	FACr				*NA*	NP	CJ
Qalqilya	Ci	Cr	Ci	Cr	FACi	FACr					NP	CJ
Jericho	Ci	Cr	Ci	Cr	FACi	FACr				*NA*	NP	CJ
Hebron	Ci	Cr	Ci	Cr	FACi	FACr				*NA*	NP	CJ
Bethlehem	Ci	Cr	Ci	Cr	FACi	FACr				*NA*	NP	CJ
Salfet	Ci	Cr									NP	CJ
Dura	Ci	Cr									NP	CJ
Tobas	Ci	Cr									NP	CJ
Halhul	Ci	Cr									NP	CJ

Table 1 – Court Location, Legal Jurisdiction and Software Variant Matrix

Software Variant	Abbr
Appeals	A
Cassation	C
Civil Judgments	CJ
Courts Civil	Ci
Courts Criminal	Cr
First Instance Appeals Civil	FACi
First Instance Appeals Criminal	FACr
High Court	HC
Not Automated	*NA*
Notary Public	NP

The table reflects that there are eleven variants, each with its own set of extended software function and feature as well as an extended data structure. Each implementation of a software variant at a particular court location is in essence its own island of information. Currently there are 110 databases implemented in the West Bank providing for no cross court function and no centralized 'coding' enforcement. Several issues exist as a result of this approach to court automation;

1. As each court is an island of information any cross court capability is difficult to achieve. Such as; case transfer from one jurisdiction to another (i.e., prosecutor filing to court), case lookup, national statistical reports.
2. Centralized 'code' enforcement is difficult to achieve as the information must be updated in multiple jurisdictions and databases throughout the West Bank
3. As each software variant is built upon it's own source code, common functions are not easily maintained or enhanced. Adding a function commonly used across multiple jurisdictions must be made in eleven physical sets of code, increasing the chance of regression error eleven times over, with each regression error having to be repaired eleven times. This issue in itself is crippling to an IT organization and it's ability to meet the demands of the business.

1.6.1 Case-Based Case Tracking System

MEZAN is categorized as a "case-based" case tracking system as no centralized management of participant information exists and little exists in the way of case management functionality. Each case created in MEZAN is a container including information specific to the parties on the case and no reuse of the person or organization information from one case to another is achieved. Searching for cases involving a particular person or organization might return many records for the same person/organization, one for each case the entity is involved and with no accurate means of knowing whether the records are indeed the for the same entity. Modern court systems typically follow a "participant-based' model where parties to cases are related through appropriate data structures to one or more cases and matters. By following a participant-based approach to the software architecture significant improvements can be seen in service of notifications as address information is accurate and up to date, cross case visibility of cases for a person/organization enhancing the sentencing component of the judicial process is possible, and inter-agency coordination of case information can be more easily achieved.

1.6.2 Functional Evaluation of MEZAN

To best determine the feature and function of MEZAN and understand any current deficiencies, an industry standard assessment scorecard has been used to evaluate the software. Attachment 1 contains the National Center for State Courts Civil Functionality scorecard which was used in this assessment. The scorecard reflects basic functions of Industry-Standard civil case management software systems and is often used by courts to evaluate vendor solutions prior to procurement. The scorecard is broken down into the following functional categories;

- Case Initiation and Indexing
- Docketing and Related Record Keeping Function

- Scheduling Function
- Document Generation and Processing Function
- Calendaring Function
- Hearing Function
- Disposition Function
- Execution Function
- Case Close Function
- Accounting Function
- File, Document and Property Management Function
- Security Function
- Management and Statistical Reporting Function

By using this scorecard we can understand the current state of the MEZAN function and feature against industry standards and best practices to help point out any category/function in need of improvement. All areas of deficiency should be assessed for applicability to the West Bank Judicial program goals and the necessary plan implemented to achieve those goals

1.6.3 Technical Evaluation of MEZAN

A series of interviews with SJC, UNDP and Netham project staff were conducted to garner a complete understanding of the technical infrastructure leveraged in MEZAN. The following section details the findings of that assessment. To fully understand the content of this section the reader should refer to Attachment 2 Technology Architectures.

User Interface and Business Logic

The Mezan 1 application is written in the Microsoft VB6 development language and architected to be a two-tier fat client software solution. By doing so the software is able to meet many demands of heavy usage, particularly as the logic and user interface of the software resides completely on the client. Please see Attachment 2 for a discussion of this architecture for pros and cons.

Data and Information Storage

The Mezan 1 application stores all data and information on Microsoft SQL Server database servers installed throughout the West Bank. The technology itself is fully functional and provides many advanced ease-of-use features for administration.

Applications

There are eleven distinct software programs and data structures being supported in the West Bank to serve the needs of the following courts and departments;

- Civil Courts
- Criminal Courts
- First Instance Civil Appeals
- First Instance Criminal Appeals
- Appeals
- Cassation

- High Court
- Notary Public
- Civil Judgment
- Prosecution

Each software program is 100% independent of the next.

1.7 Summary Recommendations

1.7.1 Enhanced Function and Feature

To achieve the level of court automation required for the SJC to effectively and efficiently deliver services to the public the following advancements in function and feature are offered. These advancements are based on industry best practice as described in Attachment 1 and expert opinion of court automation and business experts.

The Mezan 1 application should be modified to address the issues identified in Attachment 1 and at a minimum address the following high-level issues:

1. Improved data quality issues through tightened data entry controls such as; date validations, edit masks, mandatory vs optional attribute declaration.
2. Improved data quality issues through increased use of centrally managed 'code' tables. i.e., hearing postpone reasons must be entered and be controlled through SJC approved values. The following are possible candidates for centrally managed 'National' 'code/reference' data
    a. Judges
    b. Attorneys
    c. Case Classifications
    d. Registry entries
    e. Hearing Postponement Reasons
    f. Case Dispositions
    g. Participant Attributes; Eye color, Hair color, National Identifiers, etc..
    h. Case Initiation Types

3. Improved cross court functionality. By providing a robust cross-court capability the data entry burden at subsequent case initiation will be reduced as case and party details can be copied from one jurisdiction to the next. This can be achieved through case transfer functionality such as;
    a. Transfer Case Registration data on Appeal
    b. Transfer Case Registration data on Remand
    c. Transfer Case Registration data on Change of Venue or incorrect filing
    d. Transfer Criminal Case Registration data from Prosecutor/Public Attorney to Court
    e. Transfer Notices to geographically appropriate Notification Department/Court for delivery
    f. Transfer court case data to appropriate Execution Department for registration of claim filed in Execution dept. (an Execution Dept. may execute judgments from multiple courts).
4. Improved case action tracking through implementation of a centralized Register of Actions capability. Doing so will provide the ability for courts to view prior historical events and at the same time automate and provide for improved case workflow through event management (see #5 below). These items can be achieved through implementation of a Single (unified) Case Register of Actions[2] with sub-views for:
    a. Case Documents received and issued by the court including case initiation documents
    b. Requests Received (by party or all parties)
    c. Judicial Orders Issued
    d. Hearings (held and scheduled)
    e. (User case-related Comments—not legal events)
    f. (other "events")
5. Improve and control workflow of court cases through creation of an event management workflow component. This function will allow the user to schedule and track future events;

[2] chronological summary list of all completed case "events"

record event completion & spawn consequent scheduled events; generate alerts when past due. Improved in this item is also the ability to measure time between events critical to understanding and improving the efficiency of court judicial process.

6. Improvements in Recording Delivery of Notices. By doing so the SJC can reduce the cost of operating the Judiciary by reducing the number of continuances/postponements due to non-notification of parties as well as increase the publics perception of court efficiency and effectiveness.
7. Improved Case Financials providing for better control and tracking of monies due the court.
8. Improved national statistical reporting. This item must be addressed in concert with improvements to case numbering as well as the issue of controlled "coding" and data quality.
9. Improved Document management. This item will reduce the burden on the court for physical case file processing by providing an electronic image of critical case file attachments. This item also reduces the ability for corruption as the case file archival is achieved as a byproduct of this function. Judges can access case records electronically reducing the need for them to have the physical file in their presence and permit them to work while out of the office.
10. An E-filing capability should be researched and implemented if legally possible. This function will reduce the burden on the court for data entry as Attorney's can complete the necessary online forms to initiate a case at the court. A byproduct of this function is improved accuracy in case filing attributes such as type of case, filing date and time, and filer identity.
11. Improved access to court records. The judiciary should improve their ability to deliver basic case information to entities such as Attorneys and citizens. Access to information outside of the physical court facilities will reduce burden on court staff by answering basic questions such as next court date, current disposition of case and judgment summary. Additionally this item will increase citizen involvement and in turn increase the publics perception of a fair and efficient judicial system
12. Improved legal research capability through the use of published judgments and decisions in a tool that provides Judicial Authorities access to information needed to adjudicate fairly and efficiently.

1.7.2 Redevelop the current MEZAN architecture to improve maintainability;

Several aspects of MEZAN may be changed to improve maintainability. Each should be carefully studied by qualified technical experts prior to adopting a particular path. These include:

1. **Unify the database schema**. The objective is to have a single, unified DB schema associated with one of multiple (e.g. 11) application program sets in order to serve a particular court jurisdiction or office. In any particular *variant* of MEZAN, some tables present in the schema may not be used. Each *variant* of MEZAN will have to be modified and tested to utilize the unified schema. Nevertheless, in the long run this approach will reduce the complexity of the technical environment and simplify subsequent change. Thus, this will increase maintainability.
2. **Unify the application program architecture**. It is possible to re-unify the MEZAN application code with the objective of having a single set of application programs serve all court jurisdictions or offices. This will introduce additional application complexity to present functional differences (menus presented, data accessible, etc.) associated with users in different courts or offices. An alternative objective might be to simply reduce the number of different *variants* based on the degree of similarity between existing *variants* of MEZAN and the likelihood of future functional divergence. For example, there might be one *variant* for courts, another *variant* for Prosecutors, etc. This approach may have merit if some of the existing MEZAN *variants* are nearly identical, while others have diverged more significantly from the "base" MEZAN.
3. **Leverage state of the art technologies and architectures in Mezan**. To improve the delivery of service and at the same reduce the cost of development services the Mezan application should be engineered from the ground up. Engineering a solution with reusable components and in a framework that can be extended over time is paramount to the success of the automation. In addition using more robust architectures such as .NET V2 framework the application would provide a scalable and secure solution.

1.7.3 Mezan V2 Software Design and Development Approach

Court automation projects of similar size and complexity have taken years to implement and are typically done with a large staff of skilled business analysts and software professionals. In my 14 years of court automation experience I have seen similar projects that have taken a minimum of 2 years and on average 3.5 years to design, build and deploy with adequate function and feature given the criteria and objectives included in this document.

Given the simplicity of the Mezan V1 architecture and solution and the complexities introduced in the proposed new Mezan V2 judicial enterprise architecture that serves not only one court department and jurisdiction but many departments and jurisdictions, including sophisticated security and data exchange concerns, I would recommend that the West Bank project look to accomplish the Mezan V2 task by either hiring a professional firm to design, build and implement the solution or buy an existing solution and customize it with help from experts. All existing judicial automation experts in the West Bank would continue to support the Mezan V1 solution and assist in providing the subject matter expertise and project oversight to the professional firm responsible for the Mezan V2 solution.

One option that seems promising given the amount of overlap in function and feature between the West Bank solution and the Kingdom of Jordan solution is to exchange software as in the past. In 2004 the Kingdom of Jordan was given permission by USAID to leverage the Mezan V1

application operational within the West Bank. The Mezan V1 application was brought to Jordan and with minor customizations implemented in 50 court locations. After several years of use in the Jordanian courts it became clear that the overhead in software maintenance and support, the limits in cross court functionality, lack of data validation and control, and lack of management information required an investment to move the software to the next generation. In 2007 Jordan began the process of moving the application into a new technology, and reengineering the function and feature, lending itself to international standards and best practices. The Jordanian Mizan V2 is built on a .NET architecture and replicates much of the feature and function of the original Mezan V1 solution but in a reengineered manner permitting streamlined cross court functionality, improved control of input data, and much improved management information. By the end of 2008 the application was not only developed but deployed in all 50 court locations. This aggressive development and deployment schedule could not have been accomplished had it not been for three factors;

- Mizan V1
- Knowledgeable judicial business analysts and Mizan V1 court automation experts
- A professional Information Technology partner

Each of the above factors played a significant role in designing, building and implementing the new solution in less than 24 months.

If the West Bank were to take the Mizan V2 application and leverage it as a starting point for meeting the needs of this Vision statement, I believe that at least 2 years and hundreds of thousands of dollars would be saved in design, development and implementation costs. As the Jordanian solution is built on the proposed technologies contained within this vision the time required to adapt the software to the West Bank is minimized and could be implemented within 6 – 12 months. One approach to verifying the function and feature of the Jordanian solution in the West Bank is to implement it as-is with only changes to notices and code data in a pilot court location. Once operational the system can be reviewed for important deficiencies in West Bank procedure and modified quickly prior to implementation throughout the West Bank.

1.8 Future Mezan (Mezan V2)

1.8.1 MEZAN Version 2 Objectives

1. **Improve long-term maintainability of MEZAN**
   a. Reduce the number of distinct variants to be maintained
   b. Simplify development of new functions used by multiple departments
   c. Provide for geographically-centralized database(s) and processing serving multiple

2. **Create an improved Technical/Architectural database and code platform for the development of new MEZAN functionality**
   a. Remedy weaknesses in database schema design, naming conventions
   b. Enforce business rules at the database level rather than in program code where feasible
   c. Strengthen auditing policy for changes to important data
   d. Reduce size of client to facilitate operation over low-speed wide-area network

3. **Develop certain new MEZAN functions as outlined in Attachment 1**
   a. Implementation of these functions shall be contained in MEZAN V2 releases subsequent to the initial release

1.8.2 Mezan Version 2 Requirements

1. ***Redevelop (redesign and recode) MEZAN version 1 (MEZAN V1)*** *to* produce a new application called MEZAN version 2 (MEZAN V2) according to the specifications listed under Solution Requirements (below).

   - Essentially, MEZAN V2 will integrate and collapse multiple MEZAN V1 variants into one of two unified MEZAN V2 variants
   - The initial release of MEZAN V2 (MV 2.0) must be effectively "transparent" to the existing MEZAN V1 user community providing for an effective transition from V1 to V2.

2. ***Convert existing courts using MEZAN V1 to use MEZAN V2***. This process requires conversion of existing MEZAN V1 data into a new MEZAN V2 database with little or no manual entry of data. The data conversion and implementation of MEZAN V2 must be accomplished with minimum disruption to court operations.

3. ***Design and code enhancements to MEZAN V1 functionality***

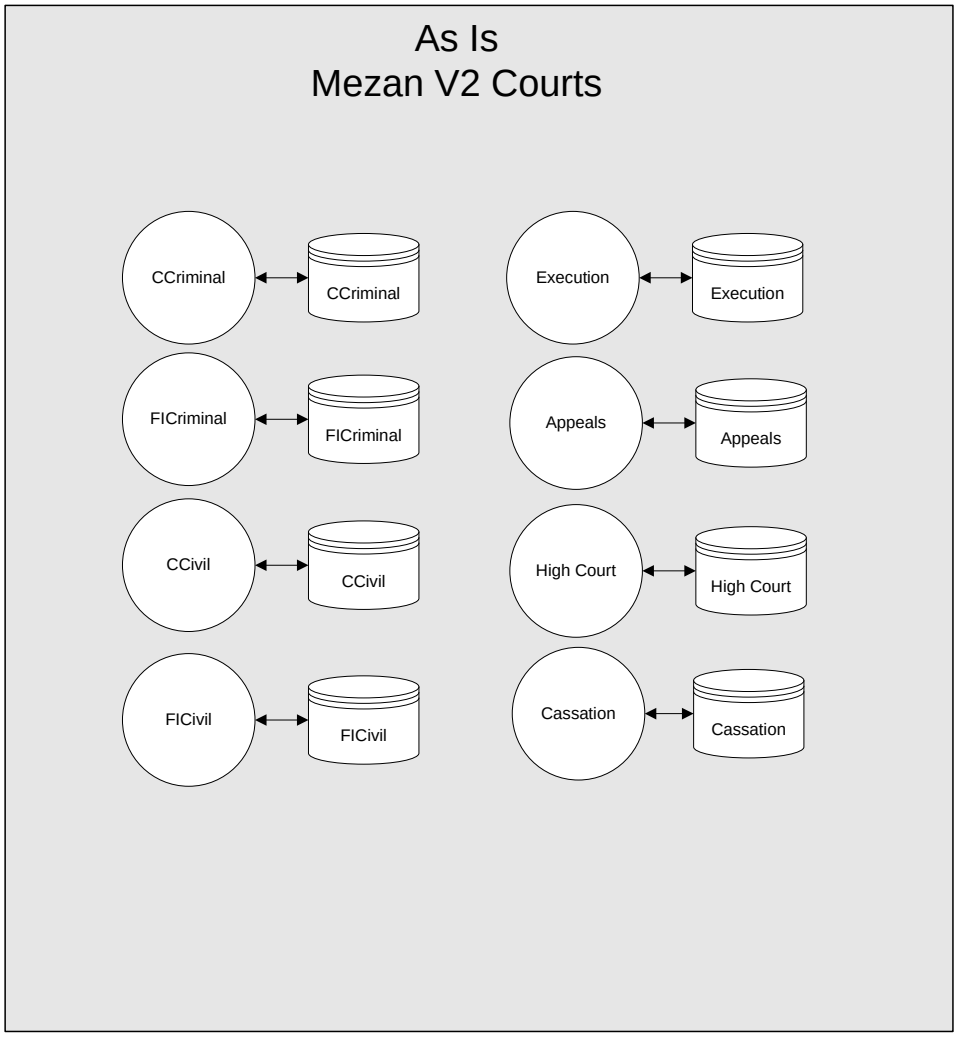
1.8.3 Solution Requirements

1. **Unify multiple MEZAN V1 variants**

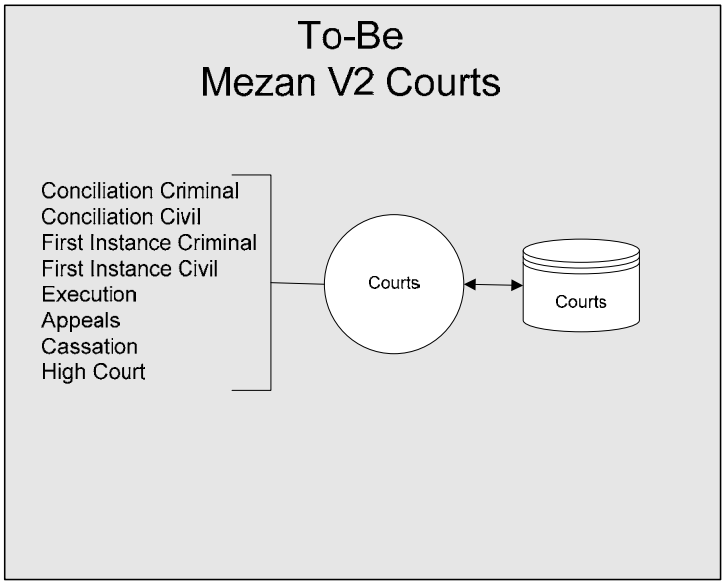
   To reduce the burden on maintenance activities for software change and provide for possible centralized sharing of data between levels of court and geographic location of court, the MEZAN V1 Variant schemas shall be unified into ***one*** or ***two*** new Mezan V2 schemas and the Variant program code shall be unified into ***two*** new MEZAN V2 source modules and executables:
   - Mezan V2 - Courts

a. ***Mezan V2 - Courts***

i. Mezan V2 - Courts will unify the following existing Mezan V1 variants:
- Conciliation/Civil **(محكمة صلح حقوق)**
- Conciliation/Criminal **(محكمة صلح جزاء)**
- First Instance/Civil **(محكمة بداية حقوق)**
- First Instance/Criminal **(محكمة بداية جزاء)**
- Appeals (**محكمة الاستئناف)**
- Cassation (**محكمة التمييز)**
- High Court of Justice (**محكمة العدل العليا)**
- Execution department (**دائرة التنفيذ)**



As-Is Mezan V1 Courts



***T*o-Be Mezan V2 - Courts**

ii. Mezan V2 - Courts must be readily extensible to permit it to be adapted to support case processing in additional West Bank court environments. In particular, the design of MEZAN V2 - Courts must anticipate the provision of Mezan V2 services to additional court jurisdictions not currently within scope of SJC Mezan v1.

iii. As a result of this unification, Mezan V2 will support the ability to store case data from multiple court legal jurisdictions (e.g. Cassation, Appeals, First Instance, etc.) in a single physical database.

iv. As a result of this unification, Mezan V2 will support the ability to store case data from multiple "legal procedure code types" (e.g. civil, criminal) in a single physical database.

v. As a result of this unification, MEZAN V2 will support the ability to store case data from multiple "appeal levels" (e.g. 1st instance—original jurisdiction, 2nd instance—appeal, 3rd instance—appeal, etc.) in a single physical database.

vi. MEZAN V2 will support the ability to provide different automated functions and procedures depending on the court legal jurisdiction, legal procedure code type, and/or appeal level.
    1. Such differences are largely characterized by operational differences which may be observed in the existing MEZAN V1 variants.
    2. Future functionality must be able to be readily conditioned on such dependencies and tailored to the specific sub-environment.

vii. MEZAN V1 is constrained to store case data from each distinct legal (and geographical) jurisdiction in a separate physical database. MEZAN V2 will preserve the option to do so in order to facilitate migration from MEZAN V1 to MEZAN V2

2. Provide for *future* geographic centralization of MEZAN services

a. As a result of this centralization, MEZAN V2 will support the ability to store case data from multiple court geographical jurisdictions (e.g. Ramallah, Jenin, Jericho, etc.) in a single physical database.

b. MEZAN V2 will support the ability to provide different automated functions and procedures depending on the court geographical jurisdiction

c. MEZAN V1 is constrained to store case data from each distinct (legal and) geographical jurisdiction in a separate physical database. MEZAN V2 will preserve the option to do this in order to facilitate migration from MEZAN V1 to MEZAN V2. Additionally the nationwide communications infrastructure is not yet available and as such the proposed solution should not assume all data services will be centralized. It is possible that the network will support the centralization of all data services at the SJC; however the proposed solution should not assume this is so. A solution that is functional in both a regional and centralized data services architecture is desired

d. MEZAN V1 is designed to operate efficiently only over a local area network (LAN); MEZAN V2 must be capable of operating efficiently over a suitable and currently-available wide-area network as well.

3. National Settings and Local Settings

Certain data parameters are to be set based upon uniform national standards and may not be modified by local MEZAN users; other data parameters may be modified locally. MEZAN V2 shall support both national and local parameter settings. National settings shall be downloadable from a central server at the SJC and stored locally for use by a local MEZAN application until changed at the national level

National settings include:

- Judge identifiers and data attributes
- Attorney identifiers and data attributes
- MEZAN User Identifiers and access control administration
- Case (cause of action) Classification parameters
- Legal jurisdiction identifiers and data attributes
- Geographic jurisdiction identifiers and data attributes
- Selected Notices, Forms and Reports

4. Access Control and User Privileges:

a. Administration of user access control shall be nationally centralized

b. User access shall be governed at the screen level and include Read, Write, and no-access options, typically users shall have write access at only a single court (Legal Jurisdiction and Geographical Jurisdiction)

c. Access control shall enable Users to be granted view access to case records in other Legal or Geographical Jurisdictions

d. All existing user roles defined in MEZAN V1 shall be preserved; additional roles may be created

e. The design should ensure access security at the Database-level, supplemented by Application-level access controls as necessary. The desire of this requirement is to prevent access of data services by means other than the front-end application or other approved applications

f. Use a password management policy with the following features:

- Account locking - when a particular user exceeds a designated number of failed login attempts, the server automatically locks that user account
- Password Aging and Expiration - when the specified amount of time passes and the password expires, the user or DBA must change the password
- Password history - control user ability to reuse an old password
- Password Complexity Verification – for example insure the password meets the following requirements:
    1. Is at least four characters long
    2. Differs from the user name
    3. Has at least one alpha, one numeric and one punctuation mark character
    4. Is not simple or obvious, such as welcome, account, database, or user
    5. Differs from the previous password by at least 3 characters

5. **Development DBMS: Microsoft SQL Server shall be the DBMS used for MEZAN V2**

6. **Database features should be used to insure integrity of data. The following are given as examples of such features;**
    - Use of fine-grained access control (FGAC);
    - Use of database integrity constraints:
        - c - check
        - p - primary key
        - u - unique
        - r - references (foreign key)
        - v - view with check option
        - o - read only on a view
    - All tables should include the following minimum auditing columns:
        - created_by –user who created the DB record
        - creation_date –date of creation
        - last_updated_by –user who lastly update the DB record
        - last_update_date –date of update
    - DB tables with high importance must be audited to keep track of all DML operations (insert, update, delete);
    - Enforce application business rules on DB level

7. **Development Language and Architecture**
   VB.NET will be used to develop a new Mezan V2 application. Features of the .Net V2 framework should be used to insure scalability, maintainability, security and extensibility should be used. A solid architecture engineered to maximize the efficiency of software development resources should be developed prior to all application development commences.

1.8.4 Mezan Application Architecture Considerations

The Mezan solution should take into consideration high-level concepts as mentioned below. These concepts should be addressed in both the applications and technology threads of the project and during the requirements analysis and logical and physical design phases of the project. To that end, the following characteristics need to be addressed and reviewed during the initial system design process:

- Portability
- Extensibility

- Dependability
- Flexibility
- Interoperability
- Scalability
- Usability
- Manageability

Portability

Due to the wide variety of technical architectures currently employed in throughout the West Bank Justice agencies, the new Mezan application components must maintain a high degree of portability in order to take into account the existence of the following:

- Diverse hardware platforms
- Multiple operating systems
- Different data sets
- Specific programming languages involved

The solution should be independent of proprietary components that are unable to satisfy the full-scale operational requirements of the SJC and constrain further development and enhancement. Although complete independence is difficult to attain, intelligent business decisions early in the design and development process can enable the adoption of “industry-standard” technologies that facilitate system portability as SJC needs and requirements continue to grow in the future.

Extensibility

The Mezan Court Case Management System must be developed using a flexible and modular approach so that enhanced functionality can be added without modifying the underlying architecture of the entire application.

This can be achieved through the utilization of a standard framework that allows new modules to be plugged into the application. By pursuing an “interchangeable parts” approach, the extensibility of the application is preserved by allowing subsequent modifications and enhancements to take place without significantly impacting existing components. Identifying key components which will allow the system to be extended should be done during the logical and physical design phases of the project. Having experience with state of the art architecture and frameworks will help with initial planning activities and are critical to the successful establishment of a framework that can be utilized as a guide for the creation of all modules.

Dependability

A reliable system is one that maximizes availability and minimizes the effect on end users. Finding problems early in the system development process significantly reduces the likelihood of encountering issues in the final stages as the “go-live” date approaches. In order to identify these problems, the SJC should develop a thorough systems development and unit testing procedure designed to identify issues with software coding, operating system compatibility, and hardware functionality. Up-front testing approaches help to ensure that the system being built is dependable.

Flexibility

As with the development of any large-scale system, the demands and expectations of end users will evolve throughout the development, implementation, and maintenance phases of the project. Therefore, it will be paramount to help ensure that the system design is adaptable to these ever-changing needs and requirements.

With that in mind, the SJC should promote the development of an "open" system with a technical architecture capable of interfacing with a wide range of external systems and varying system properties and attributes. In addition, the system should support object-oriented (OO) development, which allows for the use, re-use, and modification of system objects to meet specific needs and requirements quickly and efficiently.

As new and more capable technologies continue to evolve, this system should be designed to take advantage of industry-standard technology enablers that are proven in the marketplace and recognize for product maturity and dependability.

Interoperability

In order for the Mezan Court Case Management System to be effective, courts and justice entities throughout the West Bank will need to exchange and share data in the presence of multiple hardware and software platforms. The SJC should consider a design that considers the interchange of information between disparate platforms and frameworks.

Scalability

As the load on the Mezan Court Case Management System increases over time, either by adding users or by users moving through the application faster and longer, the application should be scalable to handle the increased workload.

Scaling can occur one of two ways: (1) you can either scale up by adding processors, replacing the processors with faster ones, or adding memory; or (2) scale out by adding more nodes (additional servers). Typically, database servers are scaled up and application servers are scaled out. The Mezan Court Case Management System should be designed and constructed in anticipation of transaction volumes and data growth.

The system design should accommodate increasing system functionality and usability by allowing for system expansion and additional end-users without compromising the flexibility and scalability of the underlying system.

Usability

Usability is a very important aspect of the design of the system's Graphical User Interface (GUI) and as such should be considered from the beginning of the project. The design should consider ease of use issues such as single assignment and mass assignment of judges to cases. The interface should provide simple navigation, through use of tools, shortcuts and reusable components that reduce the learning period for the new system.

Manageability

For the Mezan Court Case Management System to effectively meet the needs and requirements of SJC, the system must be administered and maintained in a way that requires minimal effort by Information Technology experts. Several approaches exist to minimize the released efforts such as;

- A table-driven approach to development which will greatly help minimize "hard-coding," a practice whereby the values are typed in the source code rather than being retrieved from a database. As values change, simple updates to the database can be made and the distribution of software is kept to a minimum.

- The development methodology should provide for development of an operational architecture at the same time the execution and development architectures are created. The operational architecture defines the requirements, sets the standards, and provides strategic direction for the technologies used to manage the day-to-day operation of the computing environment.

ATTACHMENT 1 - MEZAN CASE TRACKING SYSTEM FUNCTIONAL STANDARDS SCORECARD

Introduction

The National Center for State Courts working with private enterprise and judiciaries throughout the United States developed a set of standard functions and features for Court Case Management Systems. These standards have been used by various courts in the selection and evaluation of case management systems and as such the document is often represented as the NSCS Case Management System scorecard in procurement vehicles employed by governments.

1.9 Functions

1.9.1 Case Initiation and Indexing Function

Description. The activities that initiate a case and maintain its index including acceptance and processing of the initial filing, associated recordkeeping and reporting, and creation and maintenance of an index for the case.

Data Types Used. The data types required by the function include the following (see Definition of Data Types for basic contents of each data type):

- Case,
- Filings,
- Party,
- Participant,
- Attorney,
- Judge,
- Scheduled events, and
- Other events and entries.
-

Sub-functions. Within the Case Initiation and Indexing Function, the sub-functions are grouped into case initiation and indexing.

MEZAN is relatively rich in Case Initiation and Indexing functions. Multiple litigants and attorneys are captured in a flexible manner. Mezan would be considered a "case-based" Case Management System as opposed to a "Participant-based" Case Management System. Participants are not centralized and little is required to enter a participant on a case. Specifically a name is all that is required to initiate a case.

MEZAN maintains court-standard Attorney and Judge tables available for assignment to case and litigant, respectively, avoiding data entry.

Weaknesses

Searching for cases in Mezan by name, ID number, Date of Birth, etc. is possible although not as effective as systems where participant information is thoroughly required. In addition Case-based systems don't provide an easy means of coordinating person information across cases. Litigants are not assigned a separate identifier, they are not centralized and related to one or more cases and only appear in the system by name. Participant-based Case Management Systems permit the selection of an already existing system and association of that person to one or more cases.

Another weakness in the filing of a case is the case classification system that weakly enforces standards. While there is a standard classification table for types of cases (GENERAL_CLAIM_TYPE), users may easily create arbitrary sub-classification codes and duplication seems common. Moreover, case classification combines several distinct categories in one classification table: legal cause-of-action types and sub-types, urgency, as well as ill-defined administrative classifications such as "retrial". The proliferation of case classification codes compromises the value of management information based on such classifications.

Apparently, the law on Civil Procedure does not require a standard form of complaint submitted by attorneys. As a result, some data which is supported by MEZAN is unavailable to the court on case initiation, e.g. ID numbers of litigants. Attorneys might also supply case classification (e.g. legal cause-

of-action, urgency, etc.) codes, as well. Attorneys must, however, identify the legal residence location of the defendant (respondent) in order to determine appropriate geographical jurisdiction.

Mezan does not capture a 'lead' attorney and as such, correspondence such as a notice of hearing, may be delivered to attorneys not currently representing the party. i.e., A case that has more than one attorney may be associated with attorneys no longer representing a party on the case. In addition there is no ability to indicate that an attorney is inactive and no longer certified to represent litigants.

Judges are centralized within each court location and can be assigned to a case however no assignment history exists. Additionally the auto assignment feature is limited in function as it does not permit case assignment to a pool of judges based on case type, complexity or current workload/schedule.

Cases filing fees (and other moneys) are not directly receipted by the court. Instead, the register clerk estimates fee(s) due by manual calculation, provides the estimate to the filing party who then remits the amount due to a bank (a branch bank is located in the courthouse), obtains a payment receipt, and returns the receipt to the register clerk who records the receipt on MEZAN.

In addition to filing cases, MEZAN records the filing of "Requests" (motions, petitions, etc.). A request may or may not be associated with an existing case. If a request is associated with a case, the case number is entered at the time the request data is entered—linking a request to a case. Requests are classified by a separate, user-defined classification scheme (table) which mimics and extends the GENERAL_CLAIM_TYPE table values; no standards are imposed.

1.1 Case Initiation

New cases are entered into the court computer system so that information and filings (e.g., complaints, petitions) regarding the case can be recorded, retained, retrieved, used to generate forms and other documents, and combined with information from other cases to develop reports on court activity. These entries conform to locally used conventions (e.g., in case numbers, case style or title, local jurisdiction identifiers, basic case information). Other than indexing, the most basic case initiation activities are to give the case an identifier, a description, and a case file.

Table 1.1 – Case Initiation Sub-functions

[redacted]

1.1.1	generate and assign case number using locally defined format *Case number is generated and may be over-ridden by user. Case number format is YYYY/NNNNN where YYYY is year and NNNNN is a sequence number, beginning at 00001, within year. Case numbers are unique only within a database; each "court" is assigned a (separate) database where "court" is a combination of geographical location (e.g. Ramallah, Jenin,, etc.), legal jurisdiction (e.g. Conciliation, First Instance, Appeal, etc.) and department (e.g. criminal, civil departments within Conciliation and First Instance jurisdiction). As the number does not contain court location and jurisdiction identifiers statistics at the national level are difficult.*	*Yes, partially*	all	
1.1.2	generate locally defined case title or style (i.e., short phrase that identifies case and includes plaintiff and defendant names) from party names and other information.	*no, (not used)*	all	
1.1.3	generate and assign separate party identifier (e.g., party number) for each plaintiff and defendant *Parties are uniquely identified within the database, but explicit party identifiers do not appear (are not disclosed) to the system user. Parties appear as a {person name, role} where "role" is among {plaintiff, defendant, attorney, judge, witness, expert}.*	*no*	all	
1.1.4	conduct locally used review processes to ensure case should be accepted by court and display results (e.g., attorney not suspended for failure to pay) *The system does not enforce acceptance rules (e.g. by not permitting the case to be filed); acceptance rules are manually enforced by court personnel (registrar, judge). Attorneys no longer authorized to represent litigants can be added to a case and permitted to file.*	*no*	all	
1.1.5	enter reason for initiation (e.g., new filing, transferred from another jurisdiction, reopened or remanded case, counter or cross claims, de novo appeal according to local procedures) *Mezan supports new cases filed, old cases reopened after a period of suspension, cases remanded from appeal court, and cases that are counter-claims, but the "reason for initiation" is not coded and captured. A new case number is generated for such cases and (except for counterclaims) no reference is made to the related case number.*	*no*	all	

1.1.6	enter locally used court identifiers (e.g., district court) and court location identifiers (e.g., county number) *Court attributes (such as court name, address, chief judge name, etc.) are captured and primarily used as title information for printed court notices. However, there is no uniform court identification method used nor are court identification attributes instantiated in the database (see 1.1.1).*	*no*	all	
1.1.7	enter other identifiers (e.g., parcels in real property rights cases) and establish relationships with parties *Attorneys are associated with litigants in a many-to-many relationship. Real estate parcels subject to encumbrance by judgment (or lien) are recorded in the Notary Public database.*	*no*	all	
1.1.8	enter in docket or register of actions case initiation information including information on initial filing noted above and basic case information (e.g., case type, case category, case status, case title or style, parties, attorneys, and docket-related events) (see also Docketing and Related Recordkeeping Function) *This information is recorded as "transactions" and appears in the case transaction register which has the effect of an "audit trail" of court-employee actions. The transaction register is not routinely used as a public "register of action", however and as such can not be used easily to track time between events for any given case.*	*yes, partial*	all	
1.1.9	enter in docket or register of actions information for parties and participants as individuals (e.g., Ann Smith) or organizations (e.g., Acme Asbestos Company) with primary contact person if organization (see also Docketing and Related Recordkeeping Function) *(see 1.1.8)*	*yes, partial*	all	
1.1.10	support electronic filing (e.g., directly from attorneys' offices) and move designated data (e.g., tagged basic case information) from electronic document to civil case processing system (see also Multifunction Capabilities and Integration and Security Function regarding verification of electronically entered data)	*no*	large	sm all
1.1.11	generate receipt for or notify appropriate parties that case filing received and accepted, and give them assigned case number (notice, including electronic acknowledgment, would apply primarily when case transferred from another jurisdiction or filed electronically) (see also Document Generation and Processing Function) *No printed receipt is generated. The case number assigned may be orally communicated to a filing party.*	no	all	

1.1.12	record if time-sensitive filing that requires rapid action (e.g., restraining order, stay request, ex-parte filing, bankruptcy filings) (see also Scheduling Function) *The "case classification" contains "urgent" table values that may serve to identify urgent cases. At present time, urgency is informational only; MEZAN does not alert court to urgent matters. Case classification is multi-dimensional (records attributes of multiple case characteristics, e.g. cause of action, urgency, etc.); currently, no case-processing decisions are made based on case classification. For Requests, urgency may be inferred from the "request type"; such designation is informational only. Additionally no case event tracks are present within Mezan to aid in the movement of a case along a particular timeline. (see 1.1.13).*	*yes, partial*		All
1.1.13	support differential case management (i.e., different categories of cases are processed differently such as in time-sensitive filings, cases processed under different rules or time standards, specific judicial assignment for specific types of cases) and other case management methods (users enter local differential case management parameters and time standards into code translation tables; see List of Code Translation Tables). PLEASE NOTE: differential case management may entail highly complex computer programming because it may permit the user to define complete case processing profiles (e.g., containing processing rules and schedules for each event) for each case type and case category) *no Event Track functionality exists within Mezan.*	no, partial	large	Small
1.1.14	create groups of related cases (e.g., several tort cases filed against same defendant by different plaintiffs, multiple-plaintiff asbestos cases, other class action cases) from single or multiple filings such that initial and subsequent entries can be applied to each case in group (see also Docketing and Related Recordkeeping Function) *Cases can be associated between jurisdiction but not within jurisdiction and as such cases are not treated as a group*	no	large	Small

1.1.15 establish relationships between cases and case categories and court types (e.g., small claims), locations, and departments (e.g., for large courts with multiple locations) (see also List of Code Translation Tables) *Cases are related to a "court type" {location, legal jurisdiction, and department} only by their presence in the physical database for that court (see 1.1.1).* *Cases are classified according to the GENERAL_CLAIM_TYPE table which represents primary case classification categories. This table may only be updated by a MEZAN administrator. In addition, users may add user-defined sub classifications as necessary.* *Multiple "categories" may be assigned to a case; categories include cause-of-action classification (and may include other, unrelated, classification parameters).*	yes, partial		All
1.1.16 prompt user when parties already exist that relate to new case, followed by user-initiated search for duplicate parties and attorneys that user can transfer into current case if appropriate to avoid unnecessary data entry (e.g., using party names, addresses, and other identifiers noted above) *Litigants, witness and expert witness data is re-entered anew for each case in which they participate and not coordinated between court within and location and certainly not for courts outside the current location.*	No		all

1.2 Indexing

The index is created at case initiation and maintained throughout the life of a case. The index allows users to make rapid inquiries of the database by searching selected items (or key fields). A user can make increasingly specific inquiries of the database based on the information provided during a preliminary index search.

The overall purpose of an index is to allow users to look up cases or parties and view index information such as each party's name, role in the case, and whether the party has an attorney; case type; case number; date filed; and a cross reference to other parties in the case (e.g., the parties named in the case title or style). Users who know some specific piece of information about a case—but do not know the case number—may access the index to look up the case number or whether the court database contains information on a specific case or party. If the system returns multiple matches, the index helps users find the specific case or party they are seeking and then retrieves basic information from the index on that case or party.

The index should allow users easy interfaces with (1) other parts of the system such as docketing, scheduling, calendaring, and accounting for potentially all information (including financial information) on that case and related cases and (2) the inquiry and report generation capabilities (see Appendix A) for more varied displays and reports.

System designers must decide how the index will exist within their computer system. For instance, the index can be a "physical" entity in which all of the index information resides in a single place in the database, or it can be a "logical" entity that gathers the index information from several places. Regardless of whether the index is a physical or logical entity, the indexing system must make information easily accessible (i.e., in a manner that requires no additional user actions to correlate and manipulate index data from several places) for a specified case or party.

Table 1.2 – Indexing Sub-functions

1.2.1	create and maintain locally defined index that (1) contains basic index information (e.g., each party name, role in case, and whether party has an attorney; case type; case number; date filed; and cross reference to other parties in case [e.g., other party named in case title or style]), (2) permits database look-up by a choice of key fields (e.g., party name, party role, case filed date range) and, if record found, (3) permits retrieval and display of index information, and (4) permits easy interfaces with other parts of civil case processing system as noted below *Extensive "search" functionality permits identification of a case (or cases) which match user-entered search criteria including the criteria suggested above. Summary case data is displayed for case(s) found by search, permitting more accurate identification of the specific case sought. For each case found by search, the Case Inquiry display may be readily accessed for detailed case information. A concern exists here as relates to the identification of cases by person name or other criteria. As no strict control is placed on the end user when entering person attributes and because no central participant structure exists it is difficult to find all cases related to a person or to ensure that the information on file is correct for each record of the participant.*	*yes*	all	

1.2.2 handle look-up and retrieval sub-functions by identifying a specific party name, party role, case filed date range—if necessary, after eliminating other cases or parties that satisfy original look-up—and then obtaining index information by selecting from list of matching cases or parties or by using key fields noted above (e.g., user requests list of parties named Smith, system returns list of Smiths, user selects desired Smith from list by clicking on proper line or entering proper key fields, system returns index information on cases involving that Smith) *A concern exists here as relates to the identification of cases by person name or other criteria. As no strict control is placed on the end user when entering person attributes and because no central participant structure exists it is difficult to find all cases related to a person or to ensure that the information on file is correct for each record of the participant*	*Yes, partial*	all	
1.2.3 allow users easy interface with other parts of the system such as docketing, scheduling, calendaring, and accounting for potentially all related case and financial information (i.e., on specific case, its parties, its participants, its attorneys and on cases related to specific case and to its parties, participants, and attorneys) and with the inquiry and report generation capabilities for more varied displays and reports (see also Inquiry and Report Generation sections [Appendix A]) *Menu navigation is generally efficient for the purpose of locating and modifying case information once a case number is identified and entered. The result of an index search, however, yields a list of case numbers which match the search. To see further details of the case, a case number from the list must be manually re-entered on the update/modify screen. Also, as no central register of actions exists for docketing purposes it is difficult to find all docketing related actions on the case. You must visit several screens to visualize the activities that have occurred on the case.*	*Yes, partial*	all	

1.2.4	permit name search on various combinations of a person's or party's name (e.g., full name, last name only, part of first or last name, other options noted in Inquiry Section [Appendix A]) *A concern exists here as relates to the identification of cases by person name or other criteria. As no strict control is placed on the end user when entering person attributes and because no central participant structure exists it is difficult to find all cases related to a person or to ensure that the information on file is correct for each record of the participant*	*Yes, partial*	all	
1.2.5	if attorneys included in index, allow multiple names and bar identifiers	*Yes*		all
1.2.6	include index information in index record as noted above or make this information easily accessible (e.g., in a manner that requires no additional user actions)	*Yes*	all	
1.2.7	permit updating of index based on occurrence of specific case events (e.g., motions filed, dispositions decided)	*Yes*	all	
1.2.8	extract, print, or otherwise produce (e.g., microfiche) with appropriate security restrictions index information arranged according to various components of index (e.g., party, case number, case status) (see also Security Function) *A case index is printed at the end of each month. This may be too infrequent to serve as an effective backup method for business-continuance in the event the automated system is "down" for an extended period of time.*	*Yes*	all	
1.2.9	retrieve basic index information on all cases associated with specific participant *Mezan permits retrieval of all cases matching a supplied party name. However, since party data is entered anew for each case, mis-spellings or other variations in participant name may cause a failure to retrieve some cases associated with that participant. The system would need to move from a case-based case tracking system to a participant–based case management system to fulfill the requirements of this bullet..*	*no*	large	small
1.2.10	accommodate aliases in conjunction with indexing and processing of party names as appropriate	*no*		all

1.9.2 Docketing and Related Recordkeeping Function

Description. The activities associated with entering in the docket (or register of actions in some jurisdictions) including: (1) that a document (e.g., complaint, answer or response, cross complaint) has been filed; (2) that, in some instances, a filed document (e.g., certificate of readiness, demurrer, motion to strike) is the basis for placing a case on the court's calendar for a hearing or other review; and (3) what occurred at the hearing or other review.

This volume adheres to the following three basic characteristics of docketing:

- The docket is a record of concluded actions, known in some jurisdictions as a register of actions. This volume does not use the terms "docket" and "docketing" in any of the other connotations used in some courts, such as a term to represent the court calendar for a given day.

- As a record of concluded actions, the docket is never anticipatory. The content of the docket entry of a completed event, however, may be anticipatory (e.g., docket entry that scheduling of a hearing has been completed, while the content of the entry says the hearing will occur in the future).

- The docket's entries show the existence of a document that is part of the official court record. Some courts include other completed actions in the docket (e.g., completed unofficial administrative scheduling action), but these standards limit actions recorded in a docket to those intended for the official court record.

Docketing activities include the following functions:

(1) record in a docket or register of actions the results of events (e.g., dates, parties and other participants, and other information on initial filings, pleadings, calendared matters, and dispositions; dates and other issuance information on notices, summons, civil warrants, and other documents generated by the system; dates and outcomes of hearings; and post-disposition activities) based on the documents filed and financial transactions during the life of a case;
(2) maintain the docket or register of actions;
(3) maintain records used in the docketing function; and
(4) produce related outputs.

The docket or register of actions, which is arranged by filing date, is the primary chronological record of documents that have been filed and court orders or judgments that arise from calendared matters during the life of a case.

Users enter information in the docket or register of actions as court events are completed. The docketing function differs from the scheduling and calendaring functions (covered later in this volume) in that scheduled events and calendared matters are to be acted on in the future. For example, the clerk would enter a scheduled event on a future calendar but not in a docket or register of actions. If the clerk places a matter on a judge's calendar as a result of the activities associated with the scheduled event, the clerk dockets the fact that a hearing or other review has been calendared. (Recall from the data type definitions that the term "judge" includes judges and other judicial officers such as ADR providers [e.g., mediators, arbitrators].)

From a computer system perspective, the docket is a logical entity and not a physical repository of information as in manual case processing. Recordkeeping related to the docket refers to the computer's ability to access, correlate, and manipulate records (e.g., code translation tables, case records, party records) in a manner that produces the required information on a given case and on cases that have a

particular relationship to the given case. The computer produces this information as if it were in a physical docket book. The Information Relationships section addresses this situation and gives examples.

Additionally, when the system inputs or outputs docket information, it assists the user by providing prompts, selected printouts or displays of docket contents, an audit trail of who updated the docket, and other utility services. The Input/Output Management section addresses this capability.

Data Types Used. The data types required by the Docketing and Related Recordkeeping Function include the following (see Definition of Data Types section for basic contents of each data type):

- Case,
- Filings,
- Party,
- Participant,
- Attorney,
- Judge,
- Financial,
- Exhibits,
- Summons and other served processes,
- Forms and other documents issued by court,
- Hearing, and
- Disposition.

Sub-functions. Within the Docketing and Related Recordkeeping Function, the sub-functions are grouped into case information, event information, information relationships, and input/output management.

MEZAN does not maintain an explicit Register of Actions (ROA) as contemplated by the foregoing functional group, i.e. a generally-viewable chronological summary list of all completed case "events".

MEZAN maintains five separate sub-register(s) of certain event types as part of the case record. In particular, there is a "register" of Fee Payments recorded, a "register" of Documents filed, and a "register" of Notices prepared, a "register" of Hearings held, and a "register" of Requests received. Such registers are viewable in various case inquiry screens. The registers are maintained (added to) as incoming case documents or requests are received and added to the case file folder, as outgoing notices are prepared using MEZAN document preparation functions, as fee-payment receipts are presented by a party, and as hearing minutes are added. Historical entries of these past events are permitted to be modified by the user and/or deleted by a user supervisor (the Quality Control Unit) with no record retained of the previous data—thus case history events may be lost.

MEZAN also maintains a chronological list of "transactions", actions performed by court personnel in the course of maintaining case information on MEZAN. The transaction register is intended to serve as an audit trail of database updates and not as a register of 'official' docket entries. This is similar to a Register of Actions (i.e. it is chronological; it reflects changes to the database which reflect the changing case record over time; etc.). However, while the transaction log may record the fact that the database has been changed, it does not identify the specific data added, changed, or deleted and therefore has limited utility as a repository of case-historical data.

Each register entry within a group (and between groups) is independent and represents a completed action. No register entries predict or schedule future actions for court follow-up. For example, receipt of a pleading does not predict notification of defendant, nor a defendant "answer" due within 15 days of successful notification (see Scheduling functional group). As such this is one item that should be remedied before MEZAN can be labeled a Case Management System.

While MEZAN has the capability to record the service of notices, this function is presently not utilized in the court due, evidently, to an awkward interface for posting such service.

The transaction log is not routinely viewed or referenced by court personnel seeking to review the event history of the case. Rather, various screens in MEZAN record event history embedded in the overall case record, and users refer to appropriate MEZAN screens to view that history (e.g. the "sub-registers" identified above: notices sent, documents filed, fees paid, etc.). Thus, MEZAN tends to reflect the current state of the case file, and the historical record is "inferred" from the current content of the case record rather than residing in a single location—a Register of Actions.

Since a Register of Actions does not formally exist as such, the following sub-functions are not strictly supported by MEZAN, and "no" is indicated regarding support of the sub-function. However, where similar system capabilities exist, they are noted below.

2.1 Case Header

When the system creates the docket using entries made during case initiation and supplemented by subsequent user entries, the docket receives information on the initial filing and basic case information such as case type, case category, case status, case title or style, parties, attorneys, and docket-related events. As the case progresses, this information is maintained and additional information is recorded—primarily on events in the flow of the case as described in the Event Information below.

Table 2.1 – Case Header Sub-functions

2.1.1	maintain case information originally entered during case initiation in docket or register of actions including information on initial filing and basic case information (see also Case Initiation and Indexing Function) *No ROA is maintained. Case data originally recorded during case initiation persists as long as it is not changed or deleted. If it is subsequently changed, the fact that it was changed is recorded, but the original case data is lost.* *Information about data changes is recorded as "transactions" and appears in the case transaction register which has the effect of an "audit trail" of court-employee actions.* *Transactions recorded are defined by Transaction Table entries. However, the transaction register does not contain the data entered and/or changed or deleted, and is not routinely used as a public "register of action". (see 1.1.8)*	*No*	all	

2.1.2 maintain information originally entered during case initiation for parties and participants as individuals (e.g., Ann Smith) or organizations (e.g., Acme Asbestos Company) with primary contact person if organization (see also Case Initiation and Indexing Function) *See 2.1.2. Party additions are recorded in the transaction register as they are posted to the database by a "save" operation. Historical data pertaining to changed or deleted parties is lost (although a transaction record may reflect the fact that a change or deletion occurred).*	*No*	all	

2.2 Event Information

As the case progresses and events are completed, summary information about each event (e.g., filings, hearing results, requests for execution, dispositions) is entered into the docket. While some events may trigger an update to the case information in the docket (e.g., party name change, attorney change), event entries generally are not updated unless they have been entered incorrectly; subsequent events are entered separately.

Table 2.2 – Event Information Sub functions

[redacted]

2.2.1	enter and maintain information (e.g., document title, filing party, fees received, real property parcels) and dates on filings and other completed events not previously in system (e.g., party added or deleted, motion filed, answer or response filed, or hearing date set). *Six "sub-registers" of specific action types are maintained as part of the MEZAN case record:* *1. a "register" of Fee Payments recorded,* *2. a "register" of Documents filed,* *3. a "register" of Notices prepared,* *4. a "register" of Hearings held,* *5. a "register" of Requests received* *6. a "register" of Transactions performed on the case record.* *These registers are not integrated into a single "register of actions", and historical data changed or deleted is not maintained but is lost.*	*No*	all	
2.2.2	create docket entry and update case information based on occurrence of specific events that can be completely or partially transferred from another function such as hearing scheduled (e.g., motion granted or denied; see also Calendaring Function), hearing results (e.g., summary judgment; see also Hearings Function), dispositions (e.g., disposition date, type of disposition, information on judgment; see also Disposition Function and Accounting - Back Office Function), and requests for enforcement of judgment (see also Execution Function) *No docket entries are created in MEZAN. To see the current state of information one must visit each of the sub-functions of MEZAN; Notices, Case Details, Hearings, etc. The transaction log keeps track of all updates made to the case file however the information is not organized in such a way as to be used as a register of actions, rather to be used as an audit trail.*	*No, partial*	all	
2.2.3	create docket entry based on electronic documents distributed by other functions (e.g., notices, warrants, orders) (see also Document Generation and Processing Function, Hearings Function, and Disposition Function) *See 2.2.2. Additionally no action occurring in one function triggers an action in another function.*	*No, partial*	large	small

2.2.4	permit user to identify and retrieve electronic documents by identifying them on each detailed list of docket events (e.g., with icon adjacent to event such as motion for dismissal filed indicating that motion filed electronically) and easy display or printout of electronic document (e.g., motion that was filed) *Copies of Notices generated by the court using MEZAN are retained in the system and may be reproduced easily. They do not appear in a "register of actions", however, but are integrated into the MEZAN case record as a sub-register. MEZAN does not accept incoming non-court-generated electronic documents (although it has the capability (not presently used) to capture electronic documents and associate them with a case). Information generated on a notice is point in time, thus reprinting a notice may produce a different document than the original if indeed the information was changed prior to a reprint.*	*No, partial*	large	small
2.2.5	allow single events to create multiple docket entries (e.g., event is hearing; docket entries are attorney withdrawal and hearing results) *Use of MEZAN case update screens often update multiple data items; these automatically generates multiple transactions on the transaction log however for purposes of case management and docketing the transaction log is not sufficient.*	*no, partial*	all	
2.2.6	enter, maintain, and display or print information on special case processing requirements or orders (e.g., sealed case or document) (see also Case Initiation and Indexing Function and Security Function)	*no*	large	small
2.2.7	maintain case information as official court record in accordance with state and local statutes or rules *A Register of Action is not required as part of the "official court record" in Palestine and did not exist as such in the legacy paper-register system which MEZAN is designed to replace.*	*no*		all

2.3 Information Relationships

For single and multiple cases and persons, the system should maintain relationships between different kinds of information contained in the docket and inform users of cases, activities, parties, and other entities that would affect or be affected by the information at hand. The capability to establish and apply such relationships greatly assists users in entering and synchronizing data throughout the system.

Table 2.3 – Information Relationships Sub-functions

2.3.1	maintain information on multiple parties, participants, and attorneys in a case such as personal information, status including dismissals, current addresses, address histories, voice and facsimile telephone numbers, e-mail addresses (see also List of Code Translation Tables) *multiple parties are supported, and current personal information is recorded and available on the case record; historical information (e.g. address histories) is not retained upon update. Currently the name is the only mandatory field and as such little additional information is recorded in the system.*	*yes, partial*	all	
2.3.2	maintain multiple current and historical addresses, with beginning and ending dates, for each party, participant, and attorney	*no*	all	
2.3.3	enter, change, or withdraw attorneys for specific cases (or groups of cases) or parties (or groups of parties) with dates when active and inactive	*no*	all	
2.3.4	maintain information on law firms and associate attorneys and firms (e.g., to permit mail to be sent to each attorney in a firm, to list all cases being handled by a specific firm or attorney)	*no*	all	
2.3.5	maintain, or be able to construct in a manner that requires minimal user action, information and relationships on multiple cases, judges, attorneys, and parties (e.g., designate lead attorney, transfer group of cases or parties from one judge or hearing date to another in single transaction) (see also Case Initiation and Indexing Function)	*no*	all	
2.3.6	permit, with proper authorization (e.g., supervisor approval), deletion of specific docket entries and all related data (e.g., deletion of pleading and fee information causes related docket and accounting information to be deleted)	*no*	all	

2.3.7	apply a specific change to multiple dockets, parts of dockets, or groups of cases as if they were a single docket or case (e.g., correction of fee entry causes fee distribution amounts to be modified, change of Judge Smith's courtroom causes all records containing old courtroom number to be changed to new courtroom number, transfer group of cases to new judge when former judge retires or conflict arises, transfer group of cases to another division)	*no*	all	
2.3.8	link and display information on docket entries for events related to current docket entry (e.g., when defendant files motion that opposes previously filed motion of plaintiff, defendant's motion would be linked to original plaintiff's motion filed, and new motion filed would be linked to all pending motions in case with information displayed on who filed motions, factors involved, and pending decisions)	*no*		all

2.4 Input/Output Management

A group of utility-type sub-functions support input to and output from docketing and other functions. These sub-functions support code translation tables, user prompts, workstation usage records, docket displays, and input templates of standard court documents.

Table 2.4 – Input/Output Management Sub-functions

2.4.1	maintain and properly use code translation tables defined by user (see also List of Code Translation Tables) *some code translation tables may code multiple characteristics that should be divided into separate tables, e.g. case "category". Code tables are entered into each MEZAN database. As 110 databases currently exist the coordination of centrally defined and administered code tables is very difficult. Additionally users can alter code tables used in case management and reporting processes which make statistical reporting and management difficult.*	*Yes, partial*	all	
2.4.2	provide prompts to help users (e.g., list of codes and translations that apply to data entry situation that currently confronts user, updates required in cases related to case being updated) *Entry screens distinguish required data from optional data. Pull-down menus are available for code entry.*	*yes, partial*	all	

2.4.3	create, maintain, and produce (according to user-specified criteria such as selected workstation[s] or selected case[s]) audit trail identifying persons who made docket and other entries and when they made entries (highlighting when filing occurred if filing and entry dates are different) (see also Security Function and Accounting - Back Office Function) *Since there is no docket (Register of Actions) there is no such function. However, the transaction log serves as an audit trail.*	*no, partial*	all	
2.4.4	print or display all, part, or summaries of docket(s) (e.g., events in register of actions, all parties, summaries of judgment information, case age) for specific case or group of cases and for life of case or specific date range in chronological or reverse chronological order (see also Management and Statistical Reports Function)	*no*	all	
2.4.5	support electronic filing (e.g., directly from attorneys' offices) of pleadings and other documents (see also Multifunction Capabilities and Integration and Case Initiation and Indexing Function)	*no*	large	Small
2.4.6	maintain file of input templates available to users to create input documents and, as necessary, associated cover sheets (for use when pleadings are filed electronically) and relate each template to court event(s) (e.g., to correlate templates with events and to allow details of specific electronically filed complaint to be inserted into "boilerplate" text of complaint form for hard copy printout) (see also Multifunction Capabilities and Integration and Document Generation and Processing Function) *No electronic filing templates are available.*	*no*	large	Small
2.4.7	maintain and print or display history of changes in judge assignment including those by challenges (e.g., preemptory challenge) and showing present and former judges and reasons for change	*no*	large	small
2.4.8	maintain and print or display history of attorney changes for specific case or party	*no*	all	

1.9.3 Scheduling Function

Description. The activities associated with scheduling upcoming events, maintaining and displaying information on scheduled events, and monitoring adherence to schedules. Scheduling contrasts with docketing in that scheduling addresses events that are not calendared (i.e., placed on a calendar for judicial hearing or other review) and have not yet happened, while docketing addresses completed activities.

Courts schedule the following two basic types of events:

- In many civil courts, deadlines are set for specific events (e.g., answer or response) when a case is filed and assigned a case number (signifying that the court has accepted the case). Other deadlines are established for submission of documents (e.g., affidavits) and completion of other actions (e.g., submission of exhibits) as the case progresses. These deadlines often conform to time intervals based on the case's differential case management category, case type, or case category (see also List of Code Translation Tables). Deadlines define the schedule within which the case moves to disposition, which may be by trial or before the trial, for example, by default, dismissal, withdrawal, or conference.

- Courts also schedule trials and other judicial proceedings (e.g., motion hearings, conferences aimed at pre-trial settlement) and quasi-judicial events (i.e., ADR such as mediation, arbitration).

While most courts regard scheduled events as administrative activities and not part of the official court record, these events may initiate an action that is part of the official court record. For example, an event that violates time standards because it does not occur by its scheduled date may initiate a hearing to determine why the case is out of compliance; the hearing would be scheduled, placed on a court calendar, and become part of the official court record.

The Scheduling Function includes the scheduling of judicial and ADR events; the Calendaring Function covers the calendaring of matters placed on a judge's calendar for hearing or other review. The distinction between scheduled and calendared events takes on greater significance as access to court records—particularly electronic access—increases. While courts permit access to official court records such as calendars and hearing results, internal work such as schedules should have more protection. Access to an amalgamation of schedules and calendars, moreover, could confuse outside persons unfamiliar with court procedures and terminology. For example, a tickler reminding a clerk to pull a file and determine whether a hearing can be scheduled may cause the outside person to believe the hearing actually has been calendared. Finally, from a technical perspective, there is an intrinsic difference between internal, administrative items such as schedules and calendars, hearing results, and other items in official court records—access to schedules, when granted, is a "pull" operation, and access to calendars is a "push" operation.

Analogous to scheduling and calendaring, docketing relates closely to scheduling. Whereas docketing, as described in the Docketing and Related Recordkeeping Function, records completed actions, scheduling is anticipatory because it always establishes a future target date that may be rules-based or algorithmically determined for specific case types or categories (e.g., disposition time standards for small claims, unlawful detainer, or general civil complaints for damages).

Data Types Used. The data types required by the Scheduling Function include the following (see Definition of Data Types for basic contents of each data type):

- Scheduled events,
- Case,
- Party,

- Participant,
- Attorney,
- Judge, and
- Noncourt agencies.

Sub-functions. Within the Scheduling Function, the sub-functions are grouped into schedule creation, person and resource assignment, ticklers and other user alerts and prompts, and schedule and case management.

MEZAN is weak in supporting scheduling functions.

MEZAN supports two scheduled events: the next hearing date for either (a) a case or (b) a request. A "next hearing date" is first created when the case or request is filed. Thereafter, when hearing minutes are entered, another "next hearing date" must be supplied unless the case or request is either placed "on-hold" (a Yes/No status flag) or is "disposed" by final judgment (case) or decision (request).

However, no alerts are generated when the date for a scheduled event has passed and no follow-up minute entry has been made. Thus, some cases may fall into a "limbo" where they will remain unless and until some external event triggers review and update of the case status.

Requests (motions, petitions, etc.) may be either related or unrelated to a Case. If related, a list of requests received is available as a sub-register from Case Inquiry. When a request related to a case is scheduled for hearing, however, MEZAN does not check to see if the case itself is scheduled for a future hearing. As a result, multiple hearings for a case may be scheduled on different days.

Document (Notice) delivery is not scheduled; while the event "notice delivered" can be posted in MEZAN, the capability is not presently used, i.e. notice delivery is not presently recorded.

3.1 Schedule Creation

Before considering the people and other resources that will serve as the foundation for schedules, guidelines must be established for determining what to schedule, what conditions trigger scheduling, and how to schedule multiple entities (e.g., events, parties, cases) that relate to each other.

Table 3.1 – Schedule Creation Sub-functions

3.1.1	schedule events and groups of events (e.g., after case filed, set deadlines for service of summons, return of service, filing of answer or response) *MEZAN supports scheduling of hearing dates (events) for cases and requests. It also suggests an initial hearing date upon case filing. The date is calculated based upon a user-established court attribute (table entry) which specifies the maximum number of days between case filing and initial hearing (e.g. 15 days). The suggested date may be overridden by the system user.* *Entry of hearing minutes permits access to the judge's calendar for purpose of manually reviewing caseload in order to schedule the next*	*no, partial*	all	

	hearing. The system does not, however, automatically calculate or suggest a date. *No other events are scheduled by MEZAN..*			
3.1.2	initiate schedule of future events based on user input or occurrence of prior events (e.g., after small claims case filed, schedule hearing and deadline for sending notices to parties and participants) *With the exception of the scheduled initial hearing dates, no recorded events generate a schedule for future events.*	*no*	all	
3.1.3	allow multiple cases and events to have same scheduled date and time (e.g., multiple complaints regarding same problem to be heard together) *Multiple cases are routinely scheduled for hearing at the same time as is common in the nature of Civil Law hearing systems (each case is afforded a relatively short time before the bench).*	*yes*	all	
3.1.4	schedule maximum number of cases for specific time interval by event (e.g., hearing) type	*no*	large	small
3.1.5	schedule group of related cases as if group was a single case	*no*	all	
3.1.6	provide manual override to automatic scheduling to allow user to substitute deadlines for specific situations *Applies to date suggested for initial hearing.*	*yes*	all	
3.1.7	apply specific change (e.g., reschedule all cases to be heard by judge who is unavailable due to illness) to multiple schedules for group of cases as if group was a single case	*no*	all	
3.1.8	identify and display scheduling conflicts as noted in next group of sub-functions (see also List of Code Translation Tables)	*no*	all	
3.1.9	when multiple schedules change, modify records of all related parties, participants, calendars, docket entries, and other data and functions including displaying scheduling conflicts, suggesting resolutions, allowing user overrides, and rescheduling only with user approval (see also List of Code Translation Tables)	*no*	large	small

3.2 Person and Resource Assignment

This section covers standards for assignment of the proper people (e.g., judges, attorneys, parties, participants) and resources (e.g., court or meeting room) to create reliable schedules.

Most of these standards specify fully automated functions, particularly in large courts with many people and resources to schedule. Short of full automation, the computer could assist the user in manual assignment by displaying the requisite information—a process that may be appropriate for small courts.

Table 3.2 – Person and Resource Assignment Sub-functions

3.2.1	maintain waiting list of cases to be scheduled for specific date, date range, judge, courtroom, and other entities	*no*	large	small
3.2.2	when creating schedules, consider (1) availability of judges, attorneys, parties, participants, and court facilities; (2) weekends, holidays, and other days generally unavailable for court activities (e.g., training, retreats, judicial conferences) and days when specific individuals are unavailable; (3) scheduling conflicts to extent information is in system (e.g., all law officer and witness schedules will not be in system), but allow manual scheduling at user discretion in spite of conflicts (e.g., conflicts due to judicial absences, attorney vacations, law officer schedules) (see also List of Code Translation Tables) *MEZAN only schedules hearings; only judge (availability) resources are considered in scheduling hearings.*	*yes, partial*	all	
3.2.3	maintain availability information on judges, attorneys, parties, participants (e.g., interpreters, out-of-state witnesses), court facilities, and other scheduling factors noted in this section *Judge availability information is maintained. Information on court work days and holidays is maintained.*	*yes, partial*	all	

3.2.4	relate individual judges and groups of judges to courtrooms, locations, departments, department staff (e.g., reporter, bailiff; judge also may be considered staff), and case management tracks over permissible assignment time frames (e.g., in court with rotating judge assignments, a specific judge hears small claims cases in a particular courtroom during a given month) (see also List of Code Translation Tables) *Courtroom resources are not considered in scheduling (hearings) and no court room field exists in the application.. Hearings are typically held in judicial chambers, and each judge typically has his/her own chamber. Judge data is not explicitly associated with a particular court in MEZAN except by presence of judge data in the physical database for that court.* *Judges are not associated with departments, case tracks, case categories, etc..*	*no*	large	small
3.2.5	assign specific case categories (e.g., tort, contracts, real property rights, small claims within civil case type) to specific departments according to user-defined case/department relationships (e.g., business cases assigned to business courts)	*no*	large	small
3.2.6	assign and reassign individual and groups of judges using one or more of the following methods: randomly, according to predefined rules (e.g., by case category, by case status, by hearing type, by judge rotation policies, by judge caseload balancing policies), according to existence of specific conditions (e.g., conflict of interest), according to dates and times specific judges available to hear specific matters (e.g., motions on Wednesday afternoon) *Judges can be assigned on a sequential round robin fashion but not based on case type, court or workload.*	*no*	all	
3.2.7	assign related cases, as designated by user, to same judge and group together on schedule (e.g., multiple complaints regarding same problem or person) (see also Case Initiation and Indexing Function and Docketing and Related Recordkeeping Function)	*no*	large	small
3.2.8	reassign individual or group of cases from one judge or calendar to another as if group was a single case (e.g., judge retires or moves to appellate court) *Judges can not be mass re-assigned from one judge to another in MEZAN.*	*no*	all	

3.3 Ticklers and Other User Alerts and Prompts

The computer should generate ticklers, other alerts, and prompts to inform users (including individual users and workgroups) of impending or expired schedule deadlines, of completed schedule events, and of required scheduling actions that relate to the current activity.

Table 3.3 – Ticklers and Other User Alerts and Prompts Sub-functions

3.3.1	provide tickler capability: identify events coming due or overdue, periods about to expire or expired (e.g., answer or response due), events of which user should be aware based on locally defined needs (e.g., approaching maximum number of continuances); prompt or notify users; and initiate proper functions (e.g., generate notice regarding potential default) (see also Accounting - Back Office Function)	*no*	all	
3.3.2	provide user-activated or -deactivated visual reinforcement (e.g., flashing text, colors on screen, or computer icon) to ensure user sees tickler message	*no*		all
3.3.3	identify completed events and prompt users (e.g., summons served, awaiting answer or response)	*no*	all	
3.3.4	generate report or display that lists all events due on specific date or date range sorted by date, event, or other criteria	*no*	all	
3.3.5	prompt user to schedule predefined related cases (e.g., other complaints regarding same problem) (see also Case Initiation and Indexing Function and Docketing and Related Recordkeeping Function where relationships are defined to system—some automatically and some manually)	*no*	large	small
3.3.6	generate alert when approaching maximum number of events normally permitted on schedule (e.g., based on differential case management category, case type, case category) (see also List of Code Translation Tables)	*no*	large	small
3.3.7	generate alert when displaying cases that are not public record (e.g., confidential cases) (see also Security Function)	*no*	all	
3.3.8	generate alert when judges, attorneys, parties, participants, court facilities, and other scheduling factors unavailable	*no*	all	

3.4 Schedule and Case Management

The civil case processing system must provide highly flexible, user-defined printouts and displays of scheduling information by various groups (e.g., by day, judge, or courtroom). The system also must accommodate various case management (e.g., fast track for time-sensitive filings, specific judicial assignment for specific types of cases, use of mediators and arbitrators) methods and provide other support functions.

MEZAN displays and prints calendars of scheduled hearings by individual judge and by the court overall.

Table 3.4 – Schedule and Case Management Sub-functions

3.4.1	maintain and display information on scheduled events (e.g., next scheduled event, all scheduled events, interface with docket to view past events) *Scheduled hearings are supported.*	*yes, partial*	all	
3.4.2	print each schedule upon user request (e.g., judge's calendar by day)	*Yes*	all	
3.4.3	create, maintain, and display or print administrative or clerk's calendar that shows all cases with action pending within specific date range (e.g., show upcoming events to help clerk with intra-office work prioritization and management), and update calendar when pending actions completed *Scheduled hearings are supported. Clerks may use judge calendar (or court calendar) for hearing preparation (e.g. for selecting case files to be used in court). There is no separate "clerk's calendar".*	*yes, partial*	all	
3.4.4	enter completed events noted on administrative or clerk's calendar into docket as noted in Docketing and Related Recordkeeping Function *No clerk's calendar and no Docketing function..*	*no*	all	
3.4.5	print or display attorneys who have cases with future court dates sorted by various criteria (e.g., law firm, attorney)	*no*	all	
3.4.6	print or display schedules for various persons (e.g., judges, attorneys); and facilities (e.g., courtrooms) within specific period *only judges.*	*yes, partial*	all	
3.4.7	generate docket entry based on scheduled and completed events (see also Docketing and Related Recordkeeping Function) *no case register of actions*	*no*	all	
3.4.8	track conformance to time standards (e.g., answer or response due 30 days after service to defendant) including modifications (e.g., move from one case management track to another), overrides (e.g., override requirement that answer or response due in 30 days and manually enter 60 days), and suspension (e.g., suspend mental health classification) of time counting under certain	*no*	large	small

	conditions (see also List of Code Translation Tables)			
3.4.9	support differential case management, ADR (e.g., mediation, arbitration), and other case management methods (e.g., schedule events within various sets of differential case management rules, schedule ADR events) (see also List of Code Translation Tables).	*No*	large	small
3.4.10	include case age with any display of case status or adherence to schedules (e.g., tracking conformance to time standards) *Case age is calculated for management report(s) displaying aggregate average age of cases. However, individual case age is not calculated and displayed in results grids along with the case number or in data entry or retrieval screens.*	*No*		all

1.9.4 Document Generation and Processing Function

Description. The activities associated with generating, distributing, and tracking documents that notify persons of past and upcoming events and other court actions. The categories of documents in this section are (1) those that typically require service by a law enforcement officer or other authorized process server with a return of service such as civil warrants, summons, and complaints; (2) those that are given or sent by mail to litigants with a proof or certificate of service such as notices and letters; and (3) those that are sent with no proof of service or used internally such as forms, letters, and brief reports (as opposed to more lengthy and complex documents described in the Management and Statistical Reports Function or produced by word processing).

Many of these documents contain court seals and standard text into which the text and data that pertain to a specific case are inserted and signatures affixed. To help produce these frequently used documents, the civil case processing system allows users to create, store, and maintain forms—or output templates—that contain "boilerplate" text and may be imaged to permit court seals and signatures. When users need to complete one of these forms, the system accesses the appropriate output template and the user inserts the text and data for a given case. The text and data may be newly entered or received from sources such as electronic filing, the Internet, local or remote scanners or facsimile machines, and case processing and word processing systems (see document management coverage in Multifunction Capabilities and Integration and File, Document, and Property Management Function; see also Appendix B for a discussion of electronic filing).

The documents may be generated automatically following a specific event (e.g., notices to specific parties and participants when hearings are scheduled) or result from a user entry (e.g., civil bench warrants), and they may be either printed and distributed manually or distributed electronically (see Multifunction Capabilities and Integration). Users must track served documents from the time they are sent out until the person who has been served appears at the prescribed time and place.

This section excludes documents that record hearing results such as court orders and minutes, which are covered later in the Hearings Function; materials used in file tracking (e.g., case file labels, exhibit and property destruction notices), which are covered later in the File, Document, and Property Management Function; and financial documents
(e.g., judgment forms), which are covered later in the Disposition Function and the accounting functions.

Data Types Used. The data types required by the Document Generation and Processing Function include the following (see Definition of Data Types section for basic contents of each data type):

- Summons and other served processes,
- Forms and other documents issued by court,
- Scheduled events,
- Hearing,
- Financial,
- Case,
- Party,
- Participant,
- Attorney, and
- Non-court agencies.

Sub-functions. Within the Document Generation and Processing Function, the sub-functions are grouped into document generation and document utilities.

MEZAN supports a relatively rich document generation functions. Many document templates are stored and pre-filled with standard data upon retrieval (case number, court and judge name, party name(s),

standard text, etc.). The resulting partially-completed document may be edited and customized by the user as required using MS Word. An index of documents prepared is maintained in the case record (effecting one portion of a "register of action"), and an electronic copy of the document is retained in the database for future retrieval on demand.

No scheduling is performed regarding service and delivery of the notice. No scheduling is performed on obligations which may be imposed on the recipient upon receipt of the notice (e.g. delivery of a defendant "answer" to a plaintiff pleading, delivery of requested expert testimony, delivery of an objection or challenge, etc.).

4.1 Document Generation

This category consists of all documents generated by the system including those that typically are served by a process server, such as a law enforcement officer, and those that are simply mailed or given to a party, attorney, or participant.

Table 4.1 – Document Generation Sub-functions

4.1.1	generate notices or electronic acknowledgments and notify appropriate parties that filings, pleadings, and other documents received and accepted, particularly when a document is filed electronically (see also Multifunction Capabilities and Integration and Case Initiation and Indexing Function)	*no*	large	small
4.1.2	generate documents (e.g., summons, civil warrants, notices) triggered by a specific event (e.g., hearing scheduled) *documents are not triggered by specific "events" recorded in system, but must be initiated manually as the system user recognizes the need.*	*no*	All	
4.1.3	generate miscellaneous documents (e.g., for re-scheduled and canceled events, plaintiff claims forms, other types of forms) *a broad variety of printed documents ("notifications") may be generated from templates stored in the system. Notification forms are associated with either "cases" and "requests" (i.e. petitions).*	*yes*	All	
4.1.4	generate special notices (e.g., judge assignment, courtroom change, attorney change, schedule change, other courtesy notices) when requested	*yes*	All	
4.1.5	in cases with multiple active parties, generate single notice for attorney who represents multiple parties *Manually controlled by clerk which generates the appropriate number of Notice copies for distribution.*	*yes*	All	

4.1.6	in cases with multiple active parties, show names and primary (e.g., as designated by party or attorney) addresses of all other active parties and attorneys on notice to specific active party and show names and primary addresses of all active parties on file copy of notice	*yes*	All	
4.1.7	print documents individually or in batches in local courts or central location	*yes*	All	
4.1.8	distribute documents electronically (e.g., documents to be served to process server; notices and other documents to litigants and attorneys; notices, warrants, and other documents to be entered in docket) (see also Multifunction Capabilities and Integration and Docketing and Related Recordkeeping Function)	*No*	large	small
4.1.9	track document service, return of service, proof or certificate of service, re-service if necessary, and any other events *MEZAN has capability to record document delivery by service agent. The capability is apparently awkward to use and is not presently employed. Service of documents generated is not scheduled or tracked.*	*No*	all	
4.1.10	perform document generation, printout, and distribution sub-functions for group of cases as if group was a single case	*No*	all	

4.2 Document Utilities

This category includes various utility functions that support document generation such as output templates (i.e., forms that may be imaged to permit court seals and signatures into which text can be inserted), standard text (e.g., “boilerplate” text used in many documents), and recipients for specific documents.

Table 4.2 – Document Utilities Sub-functions

[redacted]

4.2.1 in conjunction with Docketing and Related Recordkeeping Function, allow users to create and maintain files of output templates and standard text, including entire paragraphs, and use files to (1) create official court documents by inserting text into templates (e.g., civil warrants with text and images of court seals and signatures) and (2) create other documents consisting of only text (e.g., some types of notices) (see also External Interfaces [Appendix A]) *Routine notifications are generated from templates, with user-determined text added as needed. MEZAN supports using MS Word for preparation of word-processing documents (e.g. notices, judgments) permitting free-form text creation and formatting. Word documents prepared in this way are linked to a specific case and may be retrieved and displayed. MEZAN does not support an integral "text library" or "phrase bank" which may be used to "cut and paste" blocks of standard text into documents.*	*Yes, partial*		all
4.2.2 relate each output template and text noted above to document(s) and court event(s) for which they are used	*No*		all
4.2.3 maintain only files of standard text and use to create entire documents or to insert text into "boilerplate" court forms; relate each group of text to document(s) and court event(s) for which they are used *While MEZAN permits generation of documents from templates, there is no "text library" to obtain sentences, phrases, paragraphs, etc. in order to construct such documents.*	*No*	all	
4.2.4 provide capability to retrieve addresses of attorneys, parties, and participants who should receive specific documents from various locations in system and database (e.g., attorney, party, participant records or tables) (see also List of Code Translation Tables) *Attorneys only*	*Yes, Partial*	all	
4.2.5 produce electronic forms and other documents noted above; distribute documents and receive responses (e.g., return of service) electronically (see also Multifunction Capabilities and Integration)	*No*	large	small

1.9.5 Calendaring Function

Description. The activities associated with the creation of calendared matters including the generation, maintenance, and, in some instances (e.g., electronic), distribution of court calendars for each type of hearing (e.g., jury trial, nonjury trial, motion hearing) or settlement conference. While not customarily part of calendaring, within this volume calendaring includes ADR events, such as mediation and arbitration.

Calendaring encompasses all proceedings at which arguments, witnesses, or evidence are considered by a judge or administrative body in court events such as trials and motion hearings; conferences aimed at pre-trial settlement; and ADR events such as mediation and arbitration. (Recall from the data type definitions that the term "judge" includes judges and other judicial officers such as ADR providers [e.g., mediators, arbitrators].)

Calendaring is the deliberate act of placing a matter on a judge's calendar for a hearing, trial, or ADR event on a particular date. The calendared activity, which may be immediate or at a future date, refers to court business conducted by a judge, usually with counsel and litigants present and resulting in a decision by the judge. The action, ruling, order, or judgment from the event causes production of a document that, with the calendar itself, is part of the official court record. The clerk dockets the result through an entry reflecting the action taken (e.g., a minute order or other document issued by the court); these activities are described in Docketing and Related Recordkeeping Function and Hearings Function.

Calendaring has two connotations. First, it refers to the placement of a matter within a single case—or a group of cases treated as a single case—on a judge's calendar. Such a calendar may contain only one matter but often includes several matters (e.g., law and motion, settlement conference). Second, the calendar refers to a complete list of what will be heard or considered by the court because either the court or counsel placed the matter on the calendar for hearing or review.

From another perspective, the calendar can be characterized by a particular case type or category (e.g., law, motion), all matters set for a particular courtroom on a given day or range of days, or all matters set for all judges of a trial court on a given day or range of days. In the given courtroom, the judges may function individually or as members of teams or panels.

Data Types Used. The data types required by the Calendaring Function include the following (see Definition of Data Types section for basic contents of each data type):

- Case,
- Scheduled events,
- Party,
- Participant,
- Attorney,
- Judge, and
- Non-court agencies.

Sub-functions. Within the Calendaring Function, the sub-functions are grouped into calendar creation and calendar management.

5.1 Calendar Creation

Hearing schedules (see Scheduling Function) provide the source information for court calendars. The Calendaring Function creates calendars by accepting schedule information, combining it with information from other functions (e.g., basic case information from the Docketing and Related Recordkeeping

Function, judges' notes), and arranging the information into the calendar format. As the hearing date approaches, users maintain calendars by entering changes (e.g., add witnesses, change attorneys, return to scheduling because case continued) and generate calendars (usually by printout) for distribution.

Table 5.1 – Calendar Creation Sub-functions

5.1.1	create, generate, and maintain calendars based on scheduling information (see also Scheduling Function) for each type of hearing (e.g., jury trial, non-jury trial, motion, conference, dismissal) or mixed hearings (e.g., motions and settlements) for specific periods (e.g., daily, weekly, monthly) and according to various criteria (e.g., judge, date, time, case type, case category, other elements of calendar profiles) *MEZAN provides printed calendars by date for judge or court overall. On-line calendars permit users to click on dates and easily view hearings scheduled. Calendars are not generated by case classification (category) or type of hearing.*	*yes*	all	
5.1.2	transfer easily and quickly between scheduling, calendaring, and other parts of the system when creating calendars (e.g., to view information on other cases, parties, participants)	*yes*	all	
5.1.3	create and maintain judges' notes (i.e., judges' notes and comments for use with calendar) for judges' viewing only in accordance with local rules and statutes (see also Security Function)	*no*		all

5.2 Calendar Management

Between the time the calendar is created and the hearing date, users perform various calendaring functions such as finalizing the calendar at a prescribed cutoff point, printing the calendar, distributing it to judges and strategic courthouse locations for posting, and producing summary reports.

Table 5.2 – Calendar Management Sub-functions

5.2.1	create and print calendars individually (e.g., for a judge or courtroom) or batch (e.g., for posting throughout courthouse) according to various criteria including date, judge, or courtroom. *Calendars are available by Judge and date. Courtrooms are not supported.*	*Yes, partial*	all	
5.2.2	distribute calendars electronically (e.g., jury manager, court reporters, sheriff) (see also Multifunction Capabilities and Integration) *Calendars are only available for view by system users; at present, system users include only*	*no*	large	small

	internal court personnel. If broader access to the court database were permitted by appropriate security modifications, calendar information could be distributed more broadly			
5.2.3	generate and display or print summary of upcoming hearings for a judge or in a courtroom over a specific period (e.g., a week)	*yes*	all	
5.2.4	display or print summary calendar information (e.g., for use in courtroom and could contain case number, hearing type, case title or style, hearing date and time, and other essential information from calendar) and provide interface to other parts of system to access other types of information (e.g., on parties) (see also Management and Statistical Reports Function)	*yes*	all	

1.9.6 Hearings Function

Description. The activities associated with recording the results of calendared events and notifying parties of court decisions. In the context of this volume, calendared events include all proceedings in which arguments, witnesses, or evidence is heard by a judge or administrative body. This encompasses court events such as trials and motion hearings; conferences aimed at pre-trial settlement; and ADR events such as mediation and arbitration. Even though most cases reach an important intermediate milestone (e.g., in a motion hearing) or culminate when they are adjudicated in a trial or some type of ADR event, the Hearings Function imposes only the two functions noted above—recording results and notifying parties—on civil case processing systems.

Minute entries (normally annotated on the calendar or on separate forms) and court orders record hearing results and document for the parties the findings resulting from judicial or quasi-judicial events. In performing these tasks, the Hearings Function relates closely to the Calendaring Function, Disposition Function, and Case Close Function.

As the hearing progresses, the judge may request a warrant, some type of form, or some other document which would be generated and printed as described in the Document Generation and Processing Function.

The Hearings Function uses the term “judgment” in two contexts—first, as the general term for any disposition that results from a court decision; second, to connote the information contained in a judgment such as the judgment amount, debtor information and amount, creditor information and amount, and payment provisions. This function relates closely to the Disposition Function, which discusses judgments in these contexts and covers judgment forms that document the terms of the judgment.

Data Types Used. The data types required by the Hearings Function include the following (see Definition of Data Types section for basic contents of each data type):

- Hearing,
- Scheduled events,
- Case,
- Party,
- Participant,
- Attorney,
- Judge,

- Non-court agencies, and
- Exhibits.

Sub-functions. The hearings sub-functions, which should accommodate various types of hearings and ADR events (e.g., jury trial, nonjury trial, motion hearing, mediation, arbitration), are given in the following table:

MEZAN supports hearings by facilitating creation of hearing minutes. Minutes entered result in a MS Word document which is generally unstructured and at user (court clerk) discretion. In particular, failure-to-appear may be recorded as a text entry, but such data is unstructured and therefore unavailable for analysis. As there is no effective event scheduling function in MEZAN, scheduling and recording future events (e.g. compliance with orders issued at a hearing) is not supported.

Table 6 – Hearings Sub-functions

6.1	provide for minute entry using one of the methods noted below	*yes*	all	
6.2	generate worksheet, calendar, or some other document suitable for on-line, rapid, in-court minute entry (see also Calendaring Function) *MEZAN permits direct electronic entry of hearing minutes using computers in courtrooms.*	*yes*	large	Small
6.3	generate and display or print worksheet, calendar, or some other document suitable for manually recording minutes (see also Document Generation and Processing Function and Calendaring Function) *Printed judge calendar may be use however what is intended by this question is the printing of a structured minute entry document..*	*Yes, partial*		All
6.4	enter, store, and display or print minutes recorded on calendar or worksheet	*yes*	all	
6.5	provide edits and prompts with on-line minute entry capability (see also Security Function) *Content of minute entry is an un- structured MS Word document template with parties listed which may be checked to reflect attendance. Because this data is embedded in a Word document, however, data attendance (or failure to attend) is not available for automated analysis. Most minute entries are created as free-form text and therefore not edited by the system to assist in ensuring a complete hearing record. There are no prompts to record certain facts/events based on, say, case classification. Follow-up is at user discretion.*	*no*	large	Small
6.6	enter, store, and document minute orders, including informal minute orders when there is no corresponding calendared event (e.g., ex parte matters), according to local court rules (see also List of Code Translation Tables)	*yes*	all	
6.7	use events captured in minutes to update records throughout system (e.g., information on judgments working with Disposition Function and accounting	*no*	all	

	functions; attorney withdrawals working with Docketing and Related Recordkeeping Function; adjournments, continuances, and cancellations working with Docketing and Related Recordkeeping Function (for docket updates), Scheduling Function (for rescheduling of hearing), Document Generation and Processing Function (for notice generation), Calendaring Function (to place on future calendar when scheduled), and other functions; and rulings taken under advisement on submitted matters) (see also Integration Between Functions) *Minute entries are not structured (see 6.5) to enforce follow-up data entry for specific hearing-related events. However, MEZAN supports easy navigation to set a date for a future hearing, record reason for hearing continuance or "postponement", and enter judgment data.*			
6.8	create and print court orders resulting from hearings and other judicial and ADR events *MEZAN supports notice generation capability which may produce such documents during/after court hearings. Not linked to minute entry.*	*yes, partial*	all	
6.9	distribute court orders resulting from hearings and other judicial and ADR events electronically to outside parties and internally to be entered in docket (see also Multifunction Capabilities and Integration, Docketing and Related Recordkeeping Function, and Disposition Function) *orders are not distributed electronically.*	*no*	large	small
6.10	enter information in court orders and judgments resulting from hearings and other judicial and ADR events as events in docket (if not entered automatically through previous sub-function) (see also Docketing and Related Recordkeeping Function and Disposition Function) *notices generated using notification function are recorded in notice history section of the case record. There is no Register of (all) actions, as such.*	*no*	all	
6.11	distribute court orders resulting from hearings and other judicial and ADR events based upon party's preference (e.g., mail, facsimile, e-mail) if multiple distribution methods are available	*no*		all

1.9.7 Disposition Function

Description. The activities associated with disposing all or part of a case or individual parties in a case due to a judgment, which is any type of disposition resulting from a court decision after a trial; ADR (e.g., mediation, arbitration), default, dismissal, withdrawal, settlement, transfer out to another jurisdiction, or consolidation. This function supports the user in accomplishing the actions called for in court orders.

The Disposition Function relates closely to the Hearings Function, in which judgments are determined, but not documented. The Disposition Function receives information from the Hearings Function on cases disposed by trial, ADR, and any other types of disposed cases. It also receives information on disposed cases from other functions, primarily the Docketing and Related Recordkeeping Function. It interacts with the Execution Function in processing judgments and often functions contiguously with the Case Close Function in disposing and closing cases.

The term "judgment" refers to the general term for any disposition that results from a court decision as noted above, as well as information contained in a judgment such as the judgment amount, debtor information and amount, creditor information and amount, and payment provisions.

The Disposition Function also encompasses the creation of a judgment form at case disposition to document the judgment information. During the post-judgment period, unless follow-up action is required (e.g., request for execution as described in the Execution Function), courts normally track judgment payments reactively—not proactively in an explicit effort to track satisfactions of judgment—as information becomes available to them (e.g., in memoranda of credit or garnishment of return). With this proviso, exchange of judgment information occurs within court system functions such as the Hearings Function, which supplies information from subsequent hearings that relate to the judgment; the General Accounting Function and Accounting - Back Office Function, which provide information on amounts paid, due, overdue, and disbursed; and the Execution Function, which provides information on any judgment executions that may be needed. Exchange of judgment information may occur with
(1) other governmental units at the federal, state, and local levels (e.g., sheriff for garnishments and court orders);
(2) private organizations
(e.g., credit reporting companies and collection agencies); and
(3) other users (e.g., attorneys, litigants, researchers).

Data Types Used. The data types required by the Disposition Function include the following (see Definition of Data Types for basic contents of each data type):

- Disposition,
- Case,
- Party,
- Participant,
- Scheduled events,
- Financial, and
- Other events and entities (i.e., parcels).

Sub-functions. The disposition sub-functions could apply to entire cases, individual parties (e.g., if some, but not all, parties in multiple-party case settle), individual parcels (i.e., in real property rights cases), or individual causes of action (e.g., when each claim in a multiple-claim promissory note constitutes a separate cause of action, information usually should be recorded on the disposition of each cause and of the entire case). These sub-functions are given in the following table:

Disposition of a case in Palestine is achieved upon the rendering of a verdict during a hearing in the presence of the defendant or the defendant's representative followed by the filing of a written judgment. Alternatively, a case may be dismissed after a set period of inactivity (during which the case is "on hold" or suspended—effectively for lack of prosecution) or transferred to another jurisdiction. Through code tables, MEZAN supports a classification of how the verdict was communicated to the defendant and a classification of the type of disposition (as above).

Table 7 – Disposition Sub-functions

[redacted]

7.1	record disposition type (i.e., type of judgment) including those involving entire cases, individual parties, parcels in real property rights cases, and cross complaints *MEZAN supports disposition type codes applied to individual "defendants" upon a judgment which disposes the overall case. Partial judgments are not supported. Disposition of cross-complaints requires separate judgments—for disposition, cross-complaints are effectively independent cases.*	*yes*	all	
7.2	identify inactive cases and groups of cases (e.g., no activity for 6 months) and prompt user regarding appropriate action (e.g., schedule hearing, prepare notice of motion to dismiss, extend dates) *MEZAN does not identify inactive cases or active (i.e. undisposed) cases for which no hearing has been scheduled.*	*no*	all	
7.3	process information (e.g., update docket and other records, if not updated automatically as noted below, through Docketing and Related Recordkeeping Function) and produce documents (e.g., judgment form; see also Hearings Function) for dispositions (i.e., judgments) by trial, ADR such as mediation or arbitration, default, dismissal, withdrawal, settlement, transfer out to another jurisdiction, or consolidation *Judgment template is available for use by judge to prepare formal decision document; the judgment document is linked to the case record. Summary judgment text (abstract of judgment) may be entered as text into MEZAN case record for easy review. Method of disposition is recorded as disposition code.*	*yes, partial*	all	
7.4	process information and produce documents (e.g., writ of execution, abstract of judgment) on post-judgment activities (e.g., in response to requests for execution with information on monetary and non-monetary judgments including parties, damages, non-monetary awards, pertinent dates, assignees, payments, and credits, enter and update records when judgments vacated or amended) (see also Execution Function and Accounting - Back Office Function) *Execution of judgments is completely separate from the court case process which arrives at a judgment.*	*no*	all	
7.5	distribute disposition and post-judgment documents noted above electronically external to court and internally to be entered in docket (see also Multifunction Capabilities and Integration and Docketing and Related Recordkeeping Function)	*no*	large	small

7.6	create, print, and maintain separate judgment indices (i.e., judgment book) that show original and subsequent judgments (e.g., containing dates, amounts, modifications, satisfactions, judge) by case and party (see also Case Initiation and Indexing Function and Execution Function) *MEZAN does not support multiple judgments on a single case. Cases may be "suspended", i.e. placed in suspense status which may result in dismissal after a specified period of time has elapsed with no further activity. MEZAN does not track these cases, however, or alert the court that such cases are candidates for dismissal. If a suspended case is re-opened (re-activated, renewed), it is assigned a new case number which is heard 'de novo' and not linked to the prior case number.*	*no*	all	
7.7	create, display, and maintain separate disposition and judgment screens that show original and subsequent judgments (e.g., containing amounts, modifications, and satisfactions) for each case and party (see also Case Initiation and Indexing Function and Execution Function) *Original judgment data is maintained by case and by party within case.*	*No*	all	
7.8	allow for multiple judgments in cases involving multiple parties *Judgment data is maintained by case and by party within the hearing minutes. No separate function exists for each party judgment..*	*No*	all	
7.9	update each case in group of disposed (e.g., dismissed) cases as if a group were a single case (see also Docketing and Related Recordkeeping Function)	*no*	all	

1.9.8 Execution Function

Description. The activities associated with execution of a judgment. These situations normally arise when the court is requested to assist with collection of the monetary judgment specified in a court order by obtaining information on the status of judgment payments and balance due by issuing documents such as memoranda of credit and garnishments of return. The Execution Function interacts with the Hearings Function and Disposition Function in these tasks.

Data Types Used. The data types required by the Execution Function include the following (see Definition of Data Types for basic contents of each data type):

- Disposition,
- Case,
- Party,
- Participant,
- Financial, and
- Noncourt agencies.

Sub-functions. The following table gives the execution sub-functions.

The Execution Department in Palestinian courts supports the execution of judgments from multiple courts and court levels. MEZAN supports tracking execution cases which are treated entirely independently from the original court case which resulted in a court judgment (award). Many, if not most, court functions applicable to adjudication of the original complaint pertain to execution activities. As a result, MEZAN's support of execution cases parallels that of Civil cases: case filing, hearing scheduling for case and requests, fee processing, etc. The MEZAN programs and database that supports the Execution department are derived from the basic MEZAN programs and database that supports court case processing; most variation appears in the code tables used. No electronic communication between databases is supported, e.g. case data (litigants, etc.) must be re-entered by the Execution department staff once initiated there by a filing.

Table 8 – Execution Sub-functions

8.1	process requests for execution of judgments and establish cross references for each execution sub-function given below to judgment index and judgment screen (see also Disposition Function)	*no*	all	
8.2	process objections to execution	*yes*	all	
8.3	record fully, partially, and unsatisfied executions (e.g., all obligations satisfied; see also Case Close Function) *Records fully-satisfied executions only. Other execution files are considered "pending".*	*yes, partial*	all	
8.4	update each case in group of cases for which execution requested as if group was a single case (e.g., same judgment terms and execution requirements for each case in group)	*no*	all	

1.9.9 Case Close Function

Description. The activities associated with final closure of a case (i.e., case status becomes "closed"). These activities normally are part of case disposition, but this volume addresses the Case Close Function separately from the Disposition Function to accommodate the rare instances when the two functions are separate (e.g., due to court policy or because cases may be considered disposed upon receipt of judgment forms prepared by attorneys, but not officially closed until final orders are received).

Case closure normally occurs when the case is disposed, which usually means the court has issued a final order disposing all parties and all issues and has statistically closed the case.

Case closure, however, seldom causes a case to be removed from the civil case processing system and placed in an archive file. Cases are archived according to state and local records management policies, and at this point the case becomes operationally closed.

From the perspective of a civil case processing system, the Case Close Function and sub-functions address statistical closure (i.e., the closure that relates to disposition), and the File, Document, and Property Management Function addresses operational closure (i.e., the closure that relates to archiving).

Data Types Used. The data types required by the Case Close Function include the following (see Definition of Data Types section for basic contents of each data type):

- Disposition,
- Case,
- Party,
- Participant,
- Financial,
- Other events—transfer/consolidate out,
- File management, and
- Non-court agencies.

There is no distinction in MEZAN between case closure and disposition.

Sub-functions. As noted above, the case close sub-functions would either be performed separately in the Case Close Function or in a continuum consisting of the Disposition Function and the Case Close Function. These sub-functions are:

Table 9 – Case Close Sub-functions

9.1	receive information from Disposition Function and record reason for closure (e.g., case disposed after trial, ADR such as mediation or arbitration, default, dismissal, withdrawal, settlement, transfer out to another jurisdiction, or consolidation) (see also Multifunction Capabilities and Integration)	*yes*	all	

9.2	establish cross references between consolidated cases for docketing, scheduling, notice generation, and other functions	*no*	large	small
9.3	close case (e.g., update docket; generate required forms, notices, reports for that case) (see also Docketing and Related Recordkeeping Function, Document Generation and Processing Function, Management and Statistical Reports Function)	*no*	large	small
9.4	generate overall case closure reports (e.g., cases closed over specific period with reason closed and other information such as uncollectable obligation balance; see also Management and Statistical Reports Function)	*no*	all	

1.9.10 General Accounting Function

Description. The activities associated with satisfying the court's fiduciary responsibilities including receipt of funds, posting case-related funds to a case fee record, posting noncase-related funds to other types of records, maintaining account records, disbursing funds, generating checks, billing, producing payment agreements, producing notices required for collection activities, reconciling bank accounts, and producing documents required to satisfy county, state, and federal auditing agencies.

In this volume accounting activities differ from civil case processing system functions covered previously because many accounting functions are performed by different personnel and may be supported by a different computer system. The accounting functions usually are divided between the clerk's case processing staff and a finance unit that may be in the clerk's office, an executive branch unit (e.g., county finance), or a court administrative office. The civil case processing system typically performs the functions described earlier in this volume, and it may support some or all of the accounting functions. Accounting support, however, could be provided by a financial system that performs functions such as budgeting, payroll, accounts payable, accounts receivable, fixed assets, journaling and general ledger, and trust fund management.

Because of the ambiguities in the division of functions between case processing and accounting, the financial system functions are divided into three groups:

- "case processing" functions that apply directly to civil case processing systems. These functions receive fees and other payments; generate receipts; maintain a limited number of bank accounts to hold received funds until they are sent to the proper person, agency, or account; and prepare reports on these activities.

- "financial functions" that support case processing. Functions in this group handle a wide range of interest-bearing and noninterest-bearing accounts, process accounts receivable, distribute funds, adjust fund balances, maintain journals and general ledgers, and produce end-of-period reconciliations and other summaries and reports.

- other functions—such as budgeting, payroll, and fixed assets—that relate only tangentially to case processing.

The following accounting sections cover functions in the first and second groups—case processing and financial—because that is where they normally reside in systems and organizations. In these accounting sections, each group of sub-functions are categorized according to whether they typically are case processing or financial. Functions that can occur either in case processing, financial, or both are categorized as "case processing or financial."

The case processing sub-functions are mandatory or optional for civil case processing systems as noted for each sub-function. The financial sub-functions designated as mandatory should be present in some system(s)—either a civil case processing system, a financial system, or an integrated system—but not necessarily in the categories shown below. This also applies to the financial sub-functions designated as optional.

The case processing and financial functions relate closely to each other, to other case processing and financial functions, and to accounting equipment. For example, many accounting functions cause a docket entry; judgment processing involves the Docketing, Hearings, Disposition, and Execution functions as well as accounting functions; many accounting reports relate to the other management and statistical reports; and the system may be required to interface with court cash register systems for funds collection and receipting. Because of these and many other interfaces, if the civil case processing and financial systems are separate, the interface between them must be such that they operate as if they were a

single system from the users' perspective. The accounting sections in this volume note only the most significant interfaces between the case processing and financial functions.

The Accounting - Front Counter and Cashiering Functions and Accounting - Back Office Function sections below cover the case processing and financial functions relative to each of the two office locations. The final accounting section covers general ledger functions. This section addresses common general accounting functions.

Data Types Used. The data types required by the General Accounting Function include the following (see Definition of Data Types for basic contents of each data type):

- Case,
- Party,
- Participant,
- Attorney,
- Disposition, and
- Financial.

Sub-functions. The general accounting sub-functions—all of which may be either case processing or financial—that are either common to one or more of the subsequent accounting sections or cannot be categorized into one of those sections are:

Palestinian courts are not responsible for receipting or accounting for financial obligations. Based upon court estimate of fees, litigants remit directly to the state treasury and provide a receipt to court staff as evidence that fees have been paid. MEZAN accepts data entry into the case record of fee amount, date paid, and treasury receipt number. The receipt may, or may not, record the case number in a "notes" field associated with an amount paid depending on the entry of the accounting clerk. A single receipt may reflect multiple payments associated with a case, but should not reflect payments made for multiple cases.

As a result, MEZAN performs no accounting functions. MEZAN may generate management reports from data collected on case receipts.

Table 10 – General Accounting Sub-functions

Either case processing, financial, or both				
10.1	comply with generally accepted accounting principles (GAAPs) for governmental entities (which implies courts or state must define applicable GAAPs)	*no*	All	
10.2	provide appropriate security and authorization for all accounting functions (see also Security	*no*	All	

	Function)			
10.3	allow user to override any data supplied automatically by system (e.g., funds distributed according to predetermined formula)	*no*	All	
10.4	generate accounting notices (e.g., for payment) at front counter or in back office (see also Document Generation and Processing Function)	*no*	all	
10.5	transfer funds from one case to another case or between accounts in a given case (e.g., to rectify error if jury fees posted in court reporter fund) (see also Accounting - Back Office Function)	*no*	all	
10.6	support trust fund (i.e., moneys held in trust that may be disbursed upon court order or for services rendered) accounting (e.g., post trust funds transactions to case; track receipts, disbursements, account status; credit interest; process refunds and forfeitures) (see also Accounting - Front Counter and Cashiering Function and Accounting - Back Office Function)	*no*	all	

1.9.11 Accounting - Front Counter and Cashiering Function

Description. The activities associated with the cashiering station of the front counter in the clerk's office where litigants and their representatives submit payments required by the court.

Data Types Used. See data types in General Accounting Function section.

Sub-functions. Within the Accounting – Front Counter and Cashiering Function, the sub-functions are grouped into funds collection, receipt generation, and bookkeeping.

11.1 Funds Collection

This group of sub-functions applies to all case processing and addresses the activities associated with calculating the amounts due and accepting payments from litigants and their representatives.

Table 11.1 – Funds Collections Sub-functions

	All case processing			
11.1.1	permit payment to be accepted for cases filed but not docketed completely (e.g., all data, such as party names, not entered into system) and recorded by entering minimal amount of data (e. g., case number, case type, case category, case style or title, name of party submitting payment, date of payment, nature of payment) as precursor to full docket entry	*no*	all	
11.1.2	accept payments by various methods (e.g., cash, check, credit card, fee waiver)	*no*	all	
11.1.3	accept payments by electronic funds transfer (see also Multifunction Capabilities and Integration)	*no*	large	small
11.1.4	accept payments from attorneys by electronic funds transfer from attorney bank accounts, debiting accounts established by attorneys to cover court expenses, debiting attorney credit card accounts, and on-line check writing (see also Multifunction Capabilities and Integration)	*no*	large	small
11.1.5	compute fees based on occurrence of specific event (e.g., initial filing, motion filing)	*no*	all	
11.1.6	identify existence of fee waivers or deferrals, display message (e.g., indigent, governmental waiver), process appropriately (e.g., case filed but waiver deferred pending judicial review)	*no*	all	
11.1.7	allocate fees associated with nonparties (e.g., from couriers, media) that may or may not be case related (e.g., for forms, document copies, certified copies) and process appropriately (e.g., not docketed if not related to specific case)	*no*	all	
11.1.8	record fees, other moneys collected, and related information (case related and noncase related)	*yes*	all	
11.1.9	accept multiple types of payments in single	*yes*	all	

	transaction (e.g., cash, check) *Multiple payments related to a single case may appear on a single receipt. Alternatively, multiple receipts may be generated. Payments related to different cases should appear on separate receipts with distinct receipt numbers. Since this is a Treasury activity, there is no way to guarantee this restriction. The case number is a "note" field and may, or may not, appear on a receipt.*			
11.1.10	accept multiple payments for single case with capability to process as either single payment or separate payments (e.g., voiding, receipting) *See 11.1.9 above*.	*yes*	all	
11.1.11	accept single payment for multiple cases with capability to process separately for each case (e.g., voiding, receipting) *See 11.1.9 above*.	*no*	all	
11.1.12	permit payments to be voided and re-entered before daily balancing with proper security provisions (see also Security Function)	*no*	all	

11.2 Receipt Generation

This group of sub-functions applies to all case processing and addresses the activities associated with generating and printing receipts for payments from litigants and their representatives.

Table 11.2 – Receipt Generation Sub-functions

	All case processing			
11.2.1	generate and print receipts with proper identifiers (e.g., fee code, court location and address) based on collections with user option to receive single or multiple copies	*no*	all	
11.2.2	generate and distribute electronic receipts for electronic payments (see also Multifunction Capabilities and Integration)	*no*	large	small
11.2.3	generate and print receipts with unique, locally defined, sequential receipt numbers	*no*	all	
11.2.4	generate and print multiple receipts from one financial transaction covering multiple payments for multiple cases or purposes (e.g., attorney files and pays fees for several cases in one trip to courthouse)	*no*	all	
11.2.5	generate and print either a single receipt or multiple receipts from one financial transaction covering multiple payments for single case (e.g., attorney files and pays fees for pleading, forms, and copies for given case in one trip to courthouse)	*no*	all	
11.2.6	permit receipts to be reprinted (e.g., if printer malfunctions during printout) with same receipt numbers	*no*	all	

11.3 Bookkeeping

This group of sub-functions applies to all case processing and addresses the activities associated with front counter recordkeeping, primarily involving payments from litigants and their representatives and receipts generated in return for these payments.

Table 11.3 – Bookkeeping Sub-functions

All case processing			
11.3.1 establish individual (e.g., for case or party) and combined (e.g., funds held short term by clerk) bank accounts when initial fees received for new case for subsequent use in back office (see also Accounting - Back Office Function)	*no*	all	
11.3.2 record and maintain front-counter bookkeeping information on receipts and disbursements (e.g., payer, payee, receipt number, case number, purpose of payment or disbursement)	*no*	all	
11.3.3 provide secure passwords for each cashier (see also Security Function)	*no*	all	
11.3.4 identify cashier with all transactions (e.g., receipts, reports)	*no*	all	
11.3.5 compute totals, list transactions, and balance for each cash drawer, register, cashier, and fee type	*no*	all	
11.3.6 list contents of each drawer (e.g., cash, checks, credit card receipts, fee waivers, money orders)	*no*	all	
11.3.7 print summary for each cashier including totals for each type of payment (e.g., cash, checks, credit card receipts, fee waivers, money orders) (see also Accounting - Back Office Function)	*no*	all	
11.3.8 list any discrepancies among payments, receipts, and cases over specific periods for each cashier for whom above summary shows imbalance for any type of payment (see also Accounting - Back Office Function)	*no*	all	
11.3.9 permit individual cashiers to open and close at least daily (e.g., when several cashiers work different shifts at same register during same day)	*no*	all	
11.3.10 allow supervisor to correct payment type (e.g., cash, checks, credit card receipts, fee waivers, money orders) with proper security provisions (see also Security Function)	*no*	all	
11.3.11 suspend cashier operations multiple times during day (e.g., close without balancing to permit lunch and other breaks)	*no*	all	
11.3.12 permit transactions that arrive after cashier closeout to be entered as transaction for next day	*no*	all	

11.3.13 print system wide daily cash receipts journal (see also Accounting - Back Office Function)	*no*	all	

1.9.12 Accounting - Back Office Function

Description. The activities associated with back office financial recordkeeping and related functions such as maintaining account records; conducting funds transfer and other financial transactions; and producing reconciliations, statements, reports, and other documents.

Data Types Used. See data types in General Accounting Function section.

Sub-functions. Within the Accounting - Back Office Function, the sub-functions are grouped into account management; funds transfer, distribution, and disbursement; updates to accounts and other records; and summaries.

12.1 Account Management

This group of sub-functions addresses the activities associated with maintaining accounts, identifying and alerting users to abnormal conditions and producing supporting documentation, maintaining cross references to records external to the system, and maintaining code translation tables that pertain to accounting. As shown below, these sub-functions apply to case processing, financial, or both.

Table 12.1 – Account Management Sub-functions

Case processing			
12.1.1 maintain financial parts of case files and docket (e.g., payments received, liabilities with linkage to accounts receivable in finance) (see also Docketing and Related Recordkeeping Function)	*no*	all	
12.1.2 debit accounts established by attorneys to cover court expenses, and credit attorney accounts based on electronic funds transfers from attorney bank accounts, debiting attorney credit card accounts, and writing on-line checks (see also Multifunction Capabilities and Integration)	*no*	large	small
12.1.3 maintain standard tables for court costs and fees (see List of Code Translation Tables)	*no*	all	
Case processing or financial			
12.1.4 maintain and track various types of individual (e.g., case or party) and combined (e.g., funds held short term by clerk) bank accounts (e.g., interest bearing, noninterest bearing, installment, pay-through) and balances by case, due date, and party (a few accounts, such as attorney accounts and funds held short term by clerk, are case processing; most accounts, such as trusts and most escrow accounts, are financial)	*no*	all	
12.1.5 identify and record arrearages, generate alerts when scheduled payments not made (e.g., for fee waivers or deferrals now due), and take or prompt user to take appropriate action (see also Scheduling Function)	*no*	all	
Financial			
12.1.6 track status of accounts referred to other agencies	*no*	all	

or organizations (e.g., state tax intercept to recover previously waived fees) for collection			
12.1.7 produce correspondence such as payment notices and dunning letters (see also Scheduling Function and Document Generation and Processing Function)	*no*	all	

12.2 Funds Transfer, Distribution, and Disbursement

This group of sub-functions addresses the activities associated with distributing fees to other governmental units (e.g., law enforcement, state and local treasurers), sharing financial information with other governmental and private entities (e.g., banks, collection agencies), and processing disbursements. As shown below, these sub-functions apply to case processing, financial, or both.

Table 12.2 – Funds Transfer, Distribution, and Disbursement Sub-functions

Case processing			
12.2.1 record funds received from other local, state, and private units (e.g., court-ordered payments such as state tax intercepts to recover previously waived fees)	*no*	large	small
Case processing or financial			
12.2.2 share information with state agencies to coordinate collection of court-ordered payments (e.g., to recover previously waived fees)	*no*	large	small
12.2.3 place hold on disbursements	*no*	all	
12.2.4 provide information for disbursement of undistributed or unclaimed moneys (e.g., jury fees posted for settled cases, unreturned checks for moneys paid by court), update ledgers, and produce reports	*no*	all	
Financial			
12.2.5 electronically authorize and transfer collected fees to other units (e.g., appellate court for appealed cases) (see also Multifunction Capabilities and Integration)	*no*	large	small
12.2.6 compute parts of fees to be distributed to other local and state units according to predefined formula (e.g., portion of fees for county parks, county library, other purposes) and permit distribution formula override by appropriate authority	*no*	all	
12.2.7 compute parts of fees to be distributed to other local and state units according to predefined formula and distribute these moneys electronically (e.g., portion of fees for county parks, county library, other purposes) (see also Multifunction Capabilities and Integration)	*no*	large	small
12.2.8 produce report showing distribution formula, moneys distributed to other local and state units over specific period, and how formula was used to compute distributions (see also Management and	*no*	all	

Statistical Reports Function)			
12.2.9 initiate, print, and disburse sequentially numbered checks, stop issuance on checks, void checks, identify and process outstanding checks, report on checks that have cleared, and record checks on check register	*no*	all	
12.2.10 initiate, print, and disburse refund checks individually or cumulatively over specific periods (e.g., for filing fees collected in error); record checks on check register	*no*	all	

12.3 Updates to Accounts and Other Records

This group of sub-functions addresses the activities associated with processing financial transactions, calculating charges and producing bills for amounts owed the court, and processing bank deposits. As shown below, these sub-functions apply to case processing, financial, or both.

Table 12.3 – Updates to Accounts and Other Records Sub-functions

Case processing			
12.3.1 post case-related receipts to accounting records and docket or register of actions; associate receipts with proper case, account, or case activity (see also Docketing and Related Recordkeeping Function)	*yes*	all	
12.3.2 post case-related disbursements to accounting records and docket or register of actions; associate disbursements with proper case, account, or case activity (see also Docketing and Related Recordkeeping Function)	*no*	all	
12.3.3 display or print lists of transactions (e.g., receipts, disbursements, interest accruals listed by fee type or chronologically) for specific cases and accounts over specific periods (e.g., monthly for life of case) (see also General Accounting Function and Management and Statistical Reports Function)	*no*	all	
12.3.4 record changes to accounting records that result from court orders (e.g., order for refund of jury fees) and modify appropriate records	*no*	all	
12.3.5 post (as noted above), process (i.e., tasks noted throughout these accounting sections), and track (e.g., principal, interest, costs, attorney fees) garnishments and partial payments (e.g., through memorandum of credit) from litigants subsequent to judgments (see also General Accounting Function, Disposition Function, and Execution Function)	*no*	all	
Case processing or financial			
12.3.6 post interest accruals to accounting records (e.g., interest accrued daily to overall account, such as for all trust accounts, and post to individual trust accounts at end of month); associate accruals with proper account	*no*	all	

12.3.7 generate and print invoices for and document collection of all moneys (e.g., fees for reservice of process)	*no*		all
12.3.8 apply corrections without changing or deleting transactions, record and store adjusted financial entries (e.g., bank adjustments for errors or bad checks), and modify amounts due with proper authorization	*no*	all	
1.9.12.1 Financial			
12.3.9 post noncase-related receipts to accounting records and associate receipts with proper account	*no*	all	
12.3.10 post noncase-related disbursements to accounting records and associate disbursements with proper account	*no*	all	
12.3.11 accrue charges to case based on occurrence of specific events (e.g., motion filed), periodically apply debits and costs to accounts (e.g., attorney and media accounts), and produce account statements	*no*		all
12.3.12 create payment schedule, apply payments received to scheduled amount due, and produce reports on overdue amounts (e.g., for previously waived fees)	*no*	all	
12.3.13 calculate and record bank deposits	*no*	all	

12.4 Summaries

This group of sub-functions addresses the activities associated with generating the various listings and reports that document financial activities (e.g., transactions, reconciliations, audit trails) over specific periods (e.g., daily, weekly, monthly, quarterly, annually). As shown below, these sub-functions apply to case processing or financial.

Table 12.4 – Summaries Sub-functions

Case processing or financial			
12.4.1 for specific periods produce separate reports showing (1) cases for which fees received, no fees received, fees waived, no fees due; (2) all adjustments to accounts; (3) accounts receivable or payable for each case	*no*	all	
Financial			
12.4.2 list bank deposits in various groupings (e.g., totals for cash, check, credit card) showing account in which funds to be deposited	*no*	all	
12.4.3 print bank deposit slips for specific banks and periods	*no*	all	
12.4.4 for specific periods, compare court record of checks with bank record of checks; produce list of discrepancies, outstanding checks, and current court and bank balances; reconcile bank accounts; produce report giving discrepancies for all reconciliations	*no*	all	

12.4.5 produce list of items that remain open for accounts that carry balance forward from one period to next period	*no*	all	
12.4.6 produce trial balance (e.g., at end of month before posting to general ledger) and balance reports for each account over specific period	*no*	all	
12.4.7 produce precheck register (e.g., to view checks prior to printing register) and check register over specific period	*no*	all	
12.4.8 total and reconcile receipts over specific period for multiple cashiers to calculate bank deposits (see also Accounting - Front Counter and Cashiering Function)	*no*	all	
12.4.9 produce summary reports for each cash drawer, cash register, and cashier (see also Accounting - Front Counter and Cashiering Function)	*no*	all	
12.4.10 produce report containing information on fees waived and associated payments	*no*	all	
12.4.11 produce report showing financial status and history (e.g., information on transactions, account balances, discrepancies) for each account (see also Management and Statistical Reports Function)	*no*	all	
12.4.12 generate other periodic financial reports based on various criteria including at least account aging, audit trail, and journal reports (see also General Accounting Function and Management and Statistical Reports Function)	*no*	all	
12.4.13 produce lists arranged according to user-selected criteria for any type of financial transaction (e.g., fees received by date, fee type, or party; receipts by reason for payment or by party) (see also Accounting - Front Counter and Cashiering Function)	*no*	all	

1.9.13 Accounting - General Ledger Function

Description. The activities associated with general and subsidiary ledger functions.

Data Types Used. See data types in General Accounting Function section.

Sub-functions. The general ledger sub-functions are all financial.

Table 13 – Accounting – General Ledger Sub-functions

All financial			
13.1 create and maintain system-defined and user-customized chart of accounts	*no*	all	
13.2 maintain journal and, if appropriate, subsidiary ledger for each account by posting debits, credits, and adjusting entries	*no*	all	

13.3	populate subsidiary ledger automatically using data from other parts of system (e.g., Disposition Function, Execution Function, other accounting functions)	*no*	large	small
13.4	reconcile and balance all accounts	*no*	all	
13.5	create general ledger by posting journal entries, subsidiary ledger totals, and other information to each account in chart of accounts	*no*	all	

1.9.14 File, Document, and Property Management Function

Description. The activities associated with (1) creating, storing, managing, tracking, archiving, and disposing of manual, electronic, and imaged case files; (2) managing electronic and imaged documents; and (3) receiving, tracking, and returning or destroying exhibits and other property gathered by the court relative to its cases (but not fixed assets and similar property of the court).

Within the context of this volume, file management refers to case files stored either manually or on a computer medium (e.g., magnetic or optical disk). Case files must be tracked from the time the case is initiated until the files are destroyed. For manual files, this means tracking their physical location during their entire life cycle as active, inactive, archived, and destroyed files. Since multiple users can access electronic files concurrently with no movement of physical files, tracking the physical location of electronic files is relevant only when their storage medium (e.g., magnetic or optical disk) has been moved to an off-line facility (e.g., separate storage location for disks containing archived records).

These standards generally apply to imaged files without delving into the specifics of an imaging operation (e.g., scanning, retrieval, storage), but they do not assume an imaging capability because that is related technology and not a case processing function (see External Interfaces in Related Technical Considerations [Appendix A]).

Document management embraces the input and output, indexing, storage, search and retrieval, manipulation, maintenance, protection, and purging of electronic and imaged documents. Some document management systems may provide advanced capabilities in the above functions, as well as additional features such as document version control and workflow for document routing to specific workstations. At least rudimentary document management capabilities must exist either in the civil case processing system or in a separate document management system that can interface with the civil case processing system. In addition to this section, the Document Generation and Processing Function and Security Function describe these rudimentary document management standards. The System Capabilities section part of Related Technical Considerations (Appendix A) discusses advanced capabilities. Exhibits and other property consist of items submitted to substantiate a litigant's case or to provide needed information to the court.

Data Types Used. The data types required by the File, Document, and Property Management Function include the following (see Definition of Data Types for basic contents of each data type):

- File management,
- Case,
- Party, and
- Exhibits.

Sub-functions. Within the File, Document, and Property Management Function, the sub-functions are grouped into file tracking, file archival and destruction, reporting and utility, document management, and exhibit management.

14.1 File Tracking

In accordance with local and state rules governing record retention, case records must be identified when they are created at case initiation; stored as active, inactive, and archived files as they progress through their life cycle; and tracked until they are destroyed. Therefore, the record custodian must know the location of case files at all times.

The file tracking sub-functions differ depending on whether the files are manual or electronic. As noted above, the physical location of manual files must be tracked during their entire life cycle. Conversely, as long as electronic files reside on the system's primary storage medium (presumably on-line storage), their location need not be tracked. Usually this situation prevails when the files are active and sometimes when they are inactive (e.g., depending on the reason they are inactive). Archived electronic files usually are moved to off-line storage.

The sub-functions given below cover file tracking through the life cycle of case files—when they are active, inactive, archived, and destroyed—to the extent local and state rules allow for these life-cycle stages.

MEZAN tracks file folder transfers into and out of the Registrar office and maintains a history of file folder transfers. When used properly, MEZAN therefore records the current location of the case file folder.

Typically, cases pending a hearing are retained in the Judge's chamber until hearing minutes and other paperwork is complete; cases are then returned to the Registrar office until the next scheduled hearing.

Table 14.1 – File Tracking Sub-functions

14.1.1 generate labels for manual case files (see also Case Initiation and Indexing Function)	*yes*	all	
14.1.2 generate indicators (e.g., color coded labels) with information on checked-out manual files to replace those files in cabinet	*no*		all
14.1.3 track manual case files from time checked out of clerk's office through each borrower until returned to clerk's office relative to location, borrower, date removed, reason file needed, date returned or transferred, and other data	*yes*	all	
14.1.4 maintain location (e.g. storage facility, location in facility, reel number, and location on reel) for manual and electronic archived files	*no*	all	
14.1.5 maintain last location of manual and electronic destroyed files	*no*	all	
14.1.6 maintain audit trail of each case file location with information similar to that noted above for file tracking (see also Docketing and Related Recordkeeping Function)	*yes*	all	

14.2 File Archival and Destruction

In accordance with local and state rules for record retention as noted above, both manual and electronic case files pass from active to inactive status, and eventually they are archived and ultimately destroyed (or totally purged if an electronic file). At some point in its life cycle, the file is moved from on-line storage to off-line storage and eventually sent to an off-site storage facility. While the file resides in off-line computer storage, many courts retain summary information on the case in active storage to help access the archived file.

MEZAN does not support file archiving of closed case files. Electronic case records are retained in the database indefinitely.

Table 14.2 – File Archival and Destruction Sub-functions

14.2.1 identify cases to be archived and later destroyed (see also Case Close Function)	*no*	all	
14.2.2 identify cases to be retained permanently	*no*	all	
14.2.3 process files according to local and state rules for becoming archived, destroyed, or transferred to storage facility (see also List of Code Translation Tables)	*no*	all	
14.2.4 identify summary information (e.g., indices) to be retained in active or semiactive files	*no*	all	
14.2.5 generate and print reports showing archived and destroyed or transferred cases	*no*	all	

14.3 Reporting and Utility

Case processing systems often perform various reporting and utility functions as part of file management.

Table 14.3 – Reporting and Utility Sub-functions

14.3.1 generate reports on file management activities (e.g., inactive files and purged reports)	*no*	all	
14.3.2 perform utility functions (e.g., copy information such as docket entries and parties) from one case to another	*no*	all	

14.4 Document Management

Document management addresses the rudimentary document management capabilities for electronic and imaged documents (with the proviso that these standards do not assume an imaging capability) received from sources such as electronic filing, the Internet, local or remote scanners or facsimile machines, and case processing and word processing systems (see Appendix B for a discussion of electronic filing). The documents include the internally generated forms, letters, and brief reports described in the Document Generation and Processing Function. Document management capabilities must exist either in the civil case processing system or in a separate document management system that can interface with the civil case processing system. The capabilities shown in the table below are in addition to those noted in the File Tracking and the File Archival and Destruction section of this function and in Document Generation and Processing Function and Security Function.

Documents (e.g. attorney-generated, court-oriented documents such as pleadings), evidentiary documents submitted to the court (e.g. contracts, deeds, etc.), and other physical evidence submitted to the court is described and entered into the system as part of the "case file". Information may be retrieved by examining the "case documents" listed on MEZAN. This facility effectively provides an on-line "table of contents" listing paper documents in the case file folder.

Little data is recorded about the document besides the date received: The party submitting the document is not identified. MEZAN supports a general capability to link electronic documents (e.g. text documents,

scanned documents, etc.) to the case record if a file identifier is provided, but this facility is not currently used except for court-generated Notices (see Document Generation functions).

Table 14.4 – Document Management Sub-functions

14.4.1 support input, output, storage (including indexing or an equivalent capability), and search and retrieval of electronic and imaged documents *documents linked to case and stored with case summary data in database; no general indexing.*	*no, partial*	large	small
14.4.2 provide capability to toggle between views of several different documents	*no*	all	
14.4.3 provide capability to interface with document management system that is separate from case processing if civil case processing system excludes document management capabilities	*no*	all	
14.4.4 provide capability to use same document management system for imaging if imaging is included in overall case processing	*yes*	all	
14.4.5 support manipulation and maintenance of electronic or imaged documents (e.g., to produce documents that include parts of several electronic or imaged documents; see Document Generation and Processing Function) *Documents (notices) generated by Court are stored as revisable-form text (MS Word documents).*	*yes, partial*		all

14.5 Exhibit Management

Exhibits and other property must be identified when received and tracked in an analogous manner to files.

Exhibits (including official documents such as contracts, deeds, etc.) are tracked as if they were documents filed in ("enclosure to") the case file folder. No identification of the filing party is entered, however, to facilitate return of property. The relationship of an exhibit to a litigant is controlled by the document "type" attribute.

Table 14.5 – Exhibit Management Sub-functions

14.5.1 record receipt of exhibits and other property (including party submitting, exhibit or property description, exhibit or property status such as submitted into evidence), generate tag for exhibits and other property, relate to specific case, generate receipts *Exhibits treated as document received for inclusion in associated case file; no attributes regarding submitting party, status, or tagging.*	*No*	all	
14.5.2 generate exhibit and property numbers or other identifiers	*No*	all	
14.5.3 track location and status of exhibits and other property	*No*	all	

14.5.4 record return or destruction of exhibits and other property	*no*	all	
14.5.5 generate notices (1) to reclaim exhibit or property when court's usage completed and (2) to inform owner that exhibit or property destroyed (see also Document Generation and Processing Function)	*no*	all	
14.5.6 print or display lists of exhibits and other property according to case, party, and other parameters	*no*	all	

1.9.15 Security Function

Description. The activities associated with ensuring the integrity of the civil case processing system, its data, and its documents during normal operations and after a system failure or outage. This is accomplished through a combination of features in the case processing application software, the normal computer hardware and system software, and special-purpose hardware and software.

Depending on the type of user, the system and its data and documents must be protected at three basic levels:

- Level 1 - For court users (e.g., clerk's office staff)—who individually have different privileges on the system but collectively can enter data and documents, access most data and documents, and change some data and documents—the system, data, and documents must be protected from unauthorized access and erroneous entry.

- Level 2 - For official users outside the court who frequently submit filings and need information from the system (e.g., attorneys of record), there must be protection from access to unauthorized parts of the system, from submission of erroneous data and documents, and from direct entry of data and documents (i.e., only Level 1 users would be permitted to enter data and documents directly into the system).

- Level 3 - For unofficial users (e.g., the public), there must be protection from any access that goes beyond viewing limited parts of the system's data and documents.

The security standards are incremental in the sense that those applicable to Level 1 also apply to Levels 2 and 3, and those that apply to Levels 1 and 2 also apply to Level 3. Unless otherwise indicated, standards covered in this description apply to all three levels.

The application software should contain carefully designed input edits to improve data quality and integrity by checking data entered into the system.

Normal features provided by computer and software vendors protect the system and database from unauthorized access. Local and remote log-on and password protection restricts access to the civil case processing system, and database security at the file and record levels prevents all but selected groups of users from viewing specific files, modifying specific files, or deleting specific files. (As used in this section, "files" mean all types of files including those used to store data, documents, and programs.)

The increased security risk of Level 2 users over Level 1 users arises during electronic data exchange—particularly electronic filing. While the risk of direct data or document entry is minimal, the possibility exists that the data and documents originally sent differ from those ultimately received because, for example, they became corrupted during transmission. The court should devise a method to ensure the integrity of these data and documents—normally through civil case processing system edits or, more reliably, through special-purpose security hardware or software with features such as user authentication (verify who sent data), data integrity (verify same data sent and received), and non-repudiation (sender cannot later deny sending information).

Access to the system and database by the public and other outside unofficial persons (i.e., Level 3 users) lead to additional security requirements. For example, as noted in the External Interfaces section of Related Technical Considerations (Appendix A), the public could be given access over the Internet or allowed to access the system directly from specified locations (e.g., kiosks). Either of these alternatives presents potential problems because unknown users who do not have individually assigned passwords and other identifiers would have access. While minimal security for these users

would be the restriction that they view but not modify or delete data and documents, more restrictive measures probably would be needed as noted below in the sub-function table.

Data Types Used. The Security Function requires the use of potentially all data types.

Data access and update privileges are controlled by MEZAN program code rather than Operating System, Database Management System, or other external software. Access permissions are granted at the screen level to various classes of users. Each user is assigned a user class. Some actions (such as data deletion) may require "supervisor" permission. Some actions (such as changes to standard code tables) may require "supervisor" permission.

In practice, passwords are often shared among several staff members (e.g. between a judge and a judicial assistant). Policies, such as changing passwords on a regular basis or use o, industry standard "strong" password requirements, are not enforced.

Sub-functions. Security sub-functions are:

Table 15 – Security Sub-functions

15.1	perform user-defined edit and data validation checks such as content of each individual data field (e.g., proper format for a date) and relationship of data field to other data (e.g., date of answer or response after date filed)	*yes, partial*	all	
15.2	ensure each document and its contents sent by user (e.g., attorney) matches with that same document and its contents received by court for electronically filed cases and other information received electronically to ensure that court is referencing and retrieving correct information	*no*	large	small
15.3	ensure electronic records cannot be modified without supervisor notification	*Yes*	large	small
15.4	allow access and similar privileges based on authorizations defined, maintained, and controlled by users (e.g., access authorization tables; see also List of Code Translation Tables)	*yes*	all	
15.5	restrict local and remote access and permissible operations (i.e., view; add; change; delete; combinations of view, add, change, delete; and output) on case types, case categories, files, parts of files, and system functions from other system functions, device (e.g., terminals, personal computers [PCs]) locations, users, and groups of users *Access permissions based on screen.*	*Yes, partial*	all	
15.6	restrict local and remote access to certain cases and classifications of cases (e.g., sealed cases, mental health cases) from specific system functions, device (e.g., terminals, PCs) locations, users, and groups of users in accordance with rules, statutes, or court orders	*No*	all	

15.7 provide adequate security if public access allowed (e.g., view but not modify or delete data and documents, separate subset of database established and maintained specifically for public access [which raises issue of how often to refresh or update subset]; security at lower levels than file or record level such as at field level; "firewalls" that restrict access to only some of system and database and secure other parts)	*No*	all	
15.8 provide audit trails that show which users and workstation locations logged on to system during specified period *Transaction log serves as audit trail for database changes. Merely being logged on to view records is not recorded.*	*yes, partial*	all	
15.9 provide secure passwords for user *No password policy is enforced; passwords are routinely compromised or shared.*	*Yes, partial*	all	
15.10 allow authorized user correction of individual or groups of cases when data entry error occurs (e.g., renumber group of cases if error occurs when entering group of new cases numbered sequentially and error in first case entered causes numbers of subsequently entered cases to be changed)	*No*	all	
15.11 maintain and display audit trail of file additions, modifications, and deletions (e.g., filings entered into docket) including who made entry, when entry made, whether date entered and date filed differ (see also Docketing and Related Recordkeeping Function) *The fact that a file modification was made is recorded, but the original data modified is not recorded.*	*yes, partial*	all	
15.12 provide for disaster recovery (e.g., reconstruct status of system and its case processing and financial functions and data such as permitting access authorization tables and cash register totals to be reconstructed) *MEZAN database journaling function if/when implemented may be used to recover database in the event of system failure. Backups are not routinely executed in Palestine as each remote court contains it's own technology. IT support and maintenance is difficult. Continuity of court operation is thus at great risk.*	*Yes, partial*	all	

1.9.16 Management and Statistical Reports Function

Description. The activities associated with reporting caseload, caseflow, and workload statistics and other court financial, operations, and staff management information. While the standard method of presenting this information would be printed reports, at least summaries of the information should be

available through other types of presentations
(e.g., graphs, charts) when requested by the user.

Data Types Used. The Management and Statistical Reports Function requires the use of all data types.

Sub-functions. Within the Management and Statistical Reports Function, the sub-functions are grouped into statistics and management information.

16.1 Statistics

As a by product of day-to-day case processing, the system produces statistics for local use that satisfy the reporting requirements of the judicial branch and state agencies. These ***statistics*** appear in reports that are either produced locally by the civil case processing system or at the state level by a system located there after being sent from the local courts. The statistical reports generally fall into three categories: caseload, caseflow, and workload.

Caseload reports present statistics for each case type and, in many instances, case category (e.g., tort, contracts, real property rights, small claims within the civil case type) for a specific time period on the number of cases pending at the beginning of the period, the number of cases filed or reopened during the period, the number of cases disposed or stayed
(i.e., delayed or otherwise removed from the court's control) during the period, and the number of cases pending at the end of the period. The reports also may provide details on these basic pending, filed, and disposed statistics (e.g., percent of total caseload filed, disposed cases as percent of filings, manner of disposition).

Caseflow reports present statistics for each case type and, in many instances, case category for specific time intervals based on the age of pending cases (e.g., how many have been pending for 30, 60, or 90 days), case age at disposition (e.g., disposed within 60, 120, or 180 days), number of pending cases at each proceeding stage (e.g., number of pending awaiting answer or response, awaiting mediation, awaiting trial), and average time intervals between proceeding stages (e.g., between initial filing and answer or response).

Workload analysis presents statistics for each case type and, in many instances, case category based on trends (e.g., changes in numbers and percentages of filings to dispositions, percentage changes in filings in successive reporting periods and successive years, percentage changes in manner of disposition).

To produce statistics beyond the local civil case processing system, statistical reporting must occur from the local system to the local, state, and possibly national levels. To satisfy this requirement, electronic interfaces should exist between local systems and systems of at least the local and state court administrators. Also, there must be a means of consolidating data from local systems to produce uniform state-level statistics (such as could be accomplished through data warehousing).

Table 16.1 – Statistics Sub-functions

16.1.1 satisfy reporting requirements of judicial branch and state agencies as noted in remainder of this table	*No*	all	
16.1.2 verify data sent to judicial branch and state agencies using techniques such as aggregate totals	*No*	all	
16.1.3 transfer statistical and case data to judicial branch and state agencies electronically (see also Multifunction Capabilities and Integration)	*No*	all	

16.1.4 produce caseload, caseflow, and workload reports, either by overall count or by list of cases (e.g., pending cases arranged according to various criteria such as by case type, case category, nature of action, event status, or judge; active cases not scheduled for hearing arranged according to various criteria such as by case type, or reason not scheduled; disposed cases arranged according to various criteria such as by case type, case category, disposition type, nature of action, or judge; cases pending specific action such as pending annual review or recommendation for transfer; cases with specific status such as by reason adjourned; judicial workloads; and weighted caseload summaries)	*Yes, partial*	all	
16.1.5 produce statistical reports associated with financial activities (see also accounting functions)	*No*	all	
16.1.6 incorporate data from diverse courts throughout state (e.g., large and small courts) into uniform statewide statistics (examples of situations that must be reconciled in statewide statistics—some events may occur in all courts statewide but have subevents that occur only in large courts; statistics in large and small courts may be recorded based on different case management methods)	*No*	all	

16.2 Management Information

While management reporting is a mandatory capability for every civil case processing system, the specific management reports needed by a given court depend on highly personalized management styles. There are, however, some reports that any court needs, and these reports are designated as mandatory in the sub-function table below. The reports designated as optional are only a few examples of the many reports that civil case processing systems could produce.

Some management reports, presumably the mandatory reports and selected other reports (e.g., the optional reports listed below), are preprogrammed into the civil case processing system, and some are generated on an ad hoc basis (see also Inquiry and Report Generation sections of Related Technical Considerations [Appendix A]). The judges and other managers in each court must decide which reports they need on a continuing basis, and these reports would be preprogrammed. Invariably a court will need additional reports as conditions, personnel, and preferences change, and those additional reports can either be programmed or created on an ad hoc basis and saved.

As used in this section, the term “reports” refers to outputs to display devices and to file extractions for transfer to other systems, Internet posting, or the standard printed output. The detailed content and format of these outputs, even though preprogrammed (e.g., by a software vendor), would be determined by the local court users.

Table 16.2 – Management Information Sub-functions

16.2.1 produce reports listed below as printed reports, displays, or extracted files suitable for transfer to other systems or Internet posting	*yes*	all	

16.2.2 produce report that permits monitoring conformance with time and other performance standards relative to various guidelines (e.g., ABA Time to Disposition Standards) and tracking criteria (e.g., case age, case status, judge, exceptional cases such as complex litigation)	*Yes, partial*	all	
16.2.3 track ADR provider (e.g., arbitrator, mediator) assignments, decisions, and performance criteria (e.g., settlement rates)	*No*	all	
16.2.4 produce various detail and summary reports giving docket contents for specific cases and groups of cases by case and party (e.g., chronological list of all or some events such as filings, summaries of related cases for specific party, case summary sheets; see also Docketing and Related Recordkeeping Function)	*No*	all	
16.2.5 produce various detail and summary reports giving docket contents for specific persons (i.e., party, participant, attorney) and groups of persons by case and party (e.g., person who is both plaintiff's attorney and defendant in malpractice suit; see also Docketing and Related Recordkeeping Function)	*No*	all	

Table 16.2 – Management Information Sub-functions (continued)

16.2.6 produce report that summarizes calendars sorted according to various criteria (e.g., by case type, case category, judge or other judicial officer, attorney, defendant, date) (see also Calendaring Function)	*no*	all	
16.2.7 produce report similar to calendar summary described above that shows whether specific cases have been disposed with cross references to calendars in which they were disposed (see also Calendaring Function)	*no*	all	
16.2.8 produce report identifying amounts owed and waived for each person or organization (e.g., fee waivers for specific parties, balance due on attorney accounts, amounts due as result of attorney sanctions)	*no*	all	
16.2.9 list cases (all, active, inactive) for specific attorney and provide related information (e.g., case status)	*no*	all	
16.2.10 provide audit trail reports that show (1) which users and workstation locations logged onto system during specified period and (2) file additions, modifications, and deletions (e.g., filings entered into docket) including who made entry, when entry made, whether date entered and date filed differ (see also Docketing and Related Recordkeeping Function and Security Function)	*yes, partial*	all	
16.2.11 list and give supporting information (e.g., party such as debtor or creditor, date of judgment, amount of judgment) on all cases with open judgments	*no*	all	
16.2.12 list and give supporting information (e.g., case number, party name, dates warrant issued and served) on all cases with open warrants	*no*	all	
16.2.13 list all cases that have been continued over specific period according to various criteria (e.g., judge, party) and give supporting overall information (e.g., number per case, per judge, per attorney, per requester, and where granted)	*no*	all	
16.2.14 capture and track duration of trials by user-specified criteria such as courtroom, judge or other judicial officer, whether jury or non-jury, and how estimated duration of trial compares with actual duration	*no*	all	
16.2.15 produce report showing status of motions and related petitions and requests including motions waiting for hearing or under advisement	*no*	all	

Table 16.2 – Management Information Sub-functions (continued)

16.2.16 capture and track locally defined milestone events (e.g., initial filing, answer or response, settlement conference) for specific cases or groups of cases (e.g., case classification such as medical malpractice, judge, court division), giving more flexible caseflow information (e.g., elapsed time between user-specified events) than is available in standard statistical reports described in previous section	*no*	large	small
16.2.17 maintain and report on current and past judge assignment (including specific cases, case types, case categories), recusal, challenges, hearing results, reassignment, disqualification with reasons where appropriate (see also Scheduling Function)	*no*		all
16.2.18 produce index of executions and garnishments sorted according to various criteria (e.g., by execution number, requester name, date issued, date returned)	*no*		all

1.9.17 List of Code Translation Tables

Most modern systems save storage space and expedite data entry by using various types of codes instead of their corresponding—and generally more lengthy translations
(e.g., county code instead of county name). Such systems must have a method of associating each code with its corresponding translation.

One method of accomplishing this is for the system to maintain tables that match each group of codes with their translations (e.g., county code with the appropriate county name). When the system is implemented and subsequently when changes arise, users define the code translation tables and supply them with codes, translations, and other information that may be contained in each table (e.g., attorney addresses in attorney code translation table). Properly defined and maintained code translation tables are an efficient method of entering and storing data.

As noted earlier, these standards identify what functions civil case processing systems are supposed to perform and not how they are to perform those functions. Notwithstanding this fact, the widespread use of code translation tables suggests that these tables be used to illustrate the standard of associating codes and translations—or more basically of achieving the efficiency of code usage. The remainder of the section lists some typical tables and, for each table, gives examples of the categories of data for which codes and corresponding translations would be supplied.

Code translation tables relate closely to the data types (e.g., files in the database) covered earlier in this volume in that the tables provide the interface between the translations, which are meaningful to users, and the codes, which are stored in the database and used internally within the system. Even though, for clarity in this volume, the contents of the data types section and this section may be redundant in places, the tables and data files would complement each other with minimal redundancies in an actual system.

MEZAN is a data table driven system which provides the flexibility to apply its functional capabilities to a range of courts and court-related departments (prosecutor, execution, etc.) by simply varying the code values used in various tables. In fact, understanding of MEZAN functional capabilities is dependent upon a good understanding of the code values employed, and the relationship of various code tables to the "entity" which the code describes.

The code tables used by MEZAN are not defined with the notion of cross court data being housed together in one data source. Therefore central usage of values in the tables is not possible. If MEZAN is to provide cross-court function, nationalized statistical reports, and user friendly drop down lists based on court of jurisdiction and user permissions the 'code' tables will need to be redefined and the software modified significantly.

1.9.17.1.1 Data Type	1.9.17.1.1.1 **Examples**
account type	such as interest bearing, noninterest bearing, installment, pay-through
alternate dispute resolution (ADR) providers (e.g., mediators, arbitrators)	such as names, identifiers, addresses, case types and case categories they can handle, availability (e.g., only on Tuesdays and Thursdays), and other information on persons (e.g., attorneys) appointed by the court to impartially settle civil cases
attorneys	such as names, identifiers, firm, status (e.g., attorney sanction such as disbarment, suspension, reprimand), and other information on attorneys licensed to practice in the

	state (e.g., using the state attorney registration list) or local jurisdiction (see also Definition of Data Types for additional attorney information)
bank/company identifier	such as names, identifiers, and other information for organizations that supply services to the court (e.g., banking, payment collection, bonding) or are litigants (e.g., plaintiff, defendant, third party) (see also Definition of Data Types for additional party and participant information)
calendared event type	such as motion hearing, trial, conference with maximum number of events that can be scheduled in a given situation (e.g., combination of judge, case type, case category, courtroom, time period) (see also Definition of Data Types for additional information on hearings and other calendared events)
case close type	such as following trial, ADR (e.g., mediation, arbitration), default, dismissal, withdrawal, settlement, transfer to another jurisdiction (usually same as disposition type)
case status	such as awaiting filing of answer or response, awaiting completion of discovery, awaiting motion hearing, awaiting trial
case category	such as auto tort, medical malpractice, product liability, contracts, real property rights (see also Definition of Data Types for additional case information)
case type	such as probate, family, criminal, traffic
city/county	such as each county, city, town, and other municipality

1.9.17.1.2 Data Type	1.9.17.1.2.1 **Examples**
court identifiers	such as general jurisdiction court, limited jurisdiction court, small claims court
courtroom identifier	such as Courtroom 5 in a particular city or county as identified in the city/county table
courtroom staff	such as judge, court clerk, reporter, bailiff for a specific courtroom
courtroom type	such as courtroom, hearing room, conference room for each courtroom in the courtroom identifier table
department identifier	such as the court department that handles general jurisdiction civil cases in a particular city or county as identified in the city/county table
differential case management	such as detailed case processing rules, parameters, and schedules for each event in each case type and case

	category in courts where case types and categories are processed differently (e.g., as in time-sensitive filings) (see also event-driven systems covered in Related Technical Considerations [Appendix A])
disbursement type	such as disbursements from accounts (e.g., for fee distribution according to state, county, city formula; for undistributed or unclaimed funds)
disposition type	such as by trial, by ADR (e.g., mediation, arbitration), by transfer out to another jurisdiction, or by some other process (e.g., default, dismissal, withdrawal, settlement, consolidation) (see also Definition of Data Types for additional disposition information)
document template type	such as each type of blank document into which users enter information including input documents (e.g., complaint forms) used primarily in electronic filing and output documents (e.g., notices) that are printed and sent to or distributed electronically to litigants
document type	such as civil warrants, summons, notices, and other documents produced by court
event type	such as complaint filed, answer or response filed, motion hearing scheduled, trial scheduled, trial held, case disposed (see also Definition of Data Types for additional event information)
exhibit	such as type, status, location, test results (see also Definition of Data Types for additional exhibit information)

1.9.17.1.3 Data Type	1.9.17.1.3.1 **Examples**
exhibit retention	such as elapsed times for each type of exhibit to be retained after last activity on case before being returned to owner or destroyed
facility	such as type (e.g., off-site records storage, mental health facility), identifier
fee and service type	such as to file complaint or pleading, for services (e.g., photocopying)
fee type amount	such as preset fee amount associated with each type of document filed or issued with effective date of fee type
file access authorization	such as relationships between specific internal and external users (they may have different log-on procedures), system functions (e.g., normal case processing functions such as docketing or calendaring cannot change or delete access authorizations), and device (e.g., terminals, PCs) locations and their authorizations to view, add, change, or delete files and

	file contents
file retention	such as elapsed times for files to remain active after last activity on case, to remain inactive without further activity on case, to remain archived before destruction
filing/pleading type	such as initial complaint, answer or response, petition for dismissal (see also Definition of Data Types for additional filing information)
finding type	such as any type of judgment resulting from a court decision pursuant to trial, ADR (e.g., mediation, arbitration), default, dismissal, withdrawal, settlement, transfer out to another jurisdiction, or consolidation
hearing status	such as vacated, held, continued
holidays	such as weekends and the other locally observed holidays
judge	such as names, identifiers, availability (e.g., reviews cases in chambers each Wednesday afternoon), and other information on each judge (see Definition of Data Types for additional judge information)
judgment type	such as a court decision pursuant to a trial, ADR (e.g., mediation, arbitration), default, dismissal, withdrawal, settlement, transfer out to another jurisdiction, or consolidation
1.9.17.1.4 Data Type	1.9.17.1.4.1 **Examples**
ledger type	such as general, subsidiary
minute codes	such as events captured in minutes (e.g., information on judgments; attorney withdrawals; adjournments, continuances, and cancellations; rulings taken under advisement on submitted matters)
minute orders	such as minute order types and formats
motion type	such as demurrer, dismissal
participant type	such as witness (see also Definition of Data Types for additional participant information)
party status	such as active, dismissed, bankruptcy
party type	such as plaintiff, defendant (see also Definition of Data Types for additional party information)
payment plan type	such as installment
payment type	such as principal, interest, arrearage, garnishment
schedule conflicts	such as judge, attorneys, witnesses, courtroom

scheduled event types	such as deadlines for submission of documents (e.g., answers or responses, affidavits) (see also Definition of Data Types for additional scheduled event information)
special status	such as sealed cases, mental health cases
time sensitive events	such as events that relate to restraining order, stay request, ex-parte filings, and bankruptcy filings
time standards	such as maximum time periods between events for each event, case type, and case category to which time standards apply (e.g., answer or response due 30 days after service to defendant for regular civil cases) (see also event-driven systems covered in Related Technical Considerations [Appendix A]);
transaction type	such as financial transactions (e.g., receipts, disbursements), case processing transactions (e.g., judge or attorney change for individual or groups of cases, new complaint filing)
zip codes	such as intrastate ZIP codes and related locations

ATTACHMENT 2 – TECHNOLOGY ARCHITECTURES

Improvements in architecture do not come without a cost, and they do not all arrive at the same time. They must be carefully planned and developed based on the capabilities and economics of computer technologies. Technological constraints dictate the architecture and design of the systems used to achieve them. Implementation requires staff training and support to use the new capabilities effectively.

Before exploring the options for a technical architecture, some definition and background is required.

1.10 The Client/Server Computing Paradigm

Since the introduction of the microcomputer some 30 years ago, modern computing systems have been built upon a "client/server" computing paradigm. In this model, one computer—called the "client"—sends requests to, and receives services from, another computer—called the "server". Clearly, the client and the server computers must be electronically connected; these connections are called the "network".

In practical use, the client computer is a personal computer located on the desktop of court workers, while the server computer is located in a centralized room within the court. The client computer can run stand-alone computer programs such as MS Word, Excel, etc., and store the resulting documents and spreadsheets locally. However, it can also request and receive data from the server computer and send new data to be stored on the server.

In MEZAN, the West Bank court case management system, case register data is stored on the server and shared with all of the client computers over the court network. While some client computers may be adding new case data to the server, other client computers may be simultaneously accessing existing case records for the purpose of scheduling hearings, modifying information on litigants, or preparing notices, judicial orders, or judgments.

1.11 Elements of a Technology Architecture

1.11.1 Client Computers

Client computers (desktop computers) may be categorized according to two extremes: "thin" clients or "thick" clients.

"Thin" clients are computers with the minimum capability required to act as a client. Essentially, they are a keyboard, a display screen, and a small computer capable of little more than running an Internet browser program such as MS Internet Explorer (or alternatives such as Firefox, Opera, Netscape, etc.). Typically, thin clients have little or no capability to run complex programs such as word processing, or to store data locally (e.g. Word documents). In practice, thin clients behave much as the "dumb" terminals connected to historical mainframe computers.

Advantages of a thin client include lower purchase cost (since some components, such as disk drives and desktop software, are omitted), and more centralized control and security (since the only programs that may be run are those authorized by the central control to be on the server). Thin clients also lower costs of support staff—since the capabilities are limited and the equipment is less complex, users are less likely to get into trouble and require technical support.

The disadvantage is reduced flexibility since all desktop capability is controlled by the server. Some widely-used programs were not designed to be administered and controlled from a server and therefore may be unavailable for use. For example, MEZAN is not currently designed to be

run from a thin client; to do so would require significant re-design and re-programming of MEZAN.

"Thick" clients are computers with full functionality—they may be operated as stand-alone computers providing services such as word processing, spreadsheets, and local databases as well as acting as a client in a client/server environment. They are capable of storing significant amounts of data locally (e.g. word documents, small databases, etc); local data storage can complement server data storage in some applications, reducing the amount of data necessary to be transmitted over the network and reducing response time in some applications. Because they are full functioned, thick clients are highly customizable to the work needs of specific users—for example, a statistician may download significant quantities of data to the desktop for the purpose of performing rapid statistical analyses.

The advantage is flexibility. The disadvantages include increased purchase cost for components and software, and increased support cost to deal with a more complex and customized desktop environment. There are also increased security risks since viruses may be introduced to the system and data may be copied and taken away. These risks must be managed by appropriate security and control policies and procedures which contribute to support costs.

1.11.2 Server Computers

Servers are central departmental computers used to store large quantities of enterprise data (such as court registers, case file documents, etc.) By consolidating such data on a server, the data may be shared with multiple clients on request and may be periodically copied and stored remotely, guarding against loss or damage to critical enterprise resources.

1.11.3 Databases

Servers typically contain databases: data organized and stored in such a way as to quickly support the typical methods used to add and retrieve data as business processes demand it. Databases can contain structured data—numbers or character text adhering to standard formats (e.g. court case number, fee amount, etc.), and unstructured data—extended text fields (e.g. party name, hearing minutes, etc.), scanned documents, photographs, fingerprints, etc. Computer programs on the server—database management systems (e.g. Oracle)—respond to computer programs on the client (e.g. MEZAN) that request data (e.g. a particular case record) and display it to the user.

1.11.4 Networks

Client computers must be connected to Server computers by electrical connections collectively called the "network". Network technologies vary, dictated principally by the geographical distance between the client and the server. Generally, these technologies may be divided into "local area networks" or "wide area networks".

A local area network (LAN) is typically established within a single building or small campus of buildings where electrical wire (and/or fiber-optic cable) may be physically installed to connect clients to servers. Direct electrical or optical cable connections permit very high data transmission rates, enabling relatively large data files (e.g. a photograph or scanned document) to be routinely transmitted between clients and servers. Alternative LAN technologies include radio-frequency connections such as wi-fi which operate over relatively short distances (e.g. 100 meters) at lower data transmission rates.

A wide-area network (WAN) is established to connect clients with servers located many kilometers or even hundreds or thousands of miles away from each other. Generally, the technology to accomplish this utilizes public utilities such as the existing regional and worldwide telephone networks. These connections only permit relatively low data transmission rates which effectively preclude their use for large data files to be routinely exchanged between clients and

servers. Over time, as public utilities build up their capacity for data communications, the available data transmission rates will increase, and costs will decrease.

The Internet, and World Wide Web, operates over this WAN infrastructure; anyone who has routinely downloaded photographs or other large data files over the internet has certainly experienced the latency (time delay) caused by the relatively low data transmission rates imposed by this technology at the present time.

1.12 Combining elements into Architectures

The elements above may be combined according to various architectural models. Two extremes are considered: a "centralized" model and a "decentralized" model—each model is characterized by different location of the principal database servers. Practically, to accomplish the business requirements of automation, neither extreme is usually employed. Rather, elements of both are combined in a "hybrid" model.

In a **centralized architecture**, all servers (and the associated databases) are collected in one central location (e.g. Judicial datacenter). Court clients, located kilometers to hundreds of kilometers away, are connected as necessary using WAN (and, secondarily, LAN) network technology. Centralized architectures behave much like legacy mainframe-computer systems. This approach provides the greatest functional capability through shared information and minimizes support and maintenance costs and activities as complex and costly IT resources are centralized.

In a **decentralized architecture**, servers are geographically distributed to be closer to the location of the clients (e.g. in each courthouse). Court clients, located meters or tens of meters away, are connected using LAN network technology. This model is necessary when network infrastructure is not adequate for carrying mission critical information. However it reduces the function of automation when information sharing across physical location is necessary as in the case of inter-court case transfers or information sharing. It also significantly increases the cost and risk exposure for Information Technology provisioning, maintenance and support.

In a **hybrid architecture**, computing resources are deployed to achieve the most cost effective solution as well as provide the greatest function and feature. A hybrid solution might save all information centrally while images are stored in file server at the local site. This architecture is dyannmic but especially difficult to design, implement, manage and is limited to technology constraints. i.e., bi-directional replication.

1.12.1 Advantages of centralization

Advantages of centralized architecture include increased consolidation and control over the enterprise data resources. Fewer servers are required to support the collective court database resulting in hardware and software cost savings; although fewer in number, centralized servers must be larger, and are therefore individually more costly. Most of the highly-technical support specialists, such as server and database administrators, are centrally located; their numbers may be reduced compared to a decentralized architecture resulting in reduced staff costs. Court databases are copied and "backed-up" centrally which, if performed properly, decrease risk of data loss. Security access to server resources can be significantly increased via central administration and configuration of security controls. Increased cross court functionality is achieved in a centralized solution as the information can be stored in one data structure, in essence consolidating the replication of code tables, increasing accuracy through centralized administration. Improved enterprise performance measures are a result of more accurate information. Software and firmware upgrades can be easily implemented in one location.

1.12.2 Disadvantages of Centralization

A centralized architecture effectively "puts all the eggs in one basket" and offers the advantages and suffers the disadvantages of that approach.

If the central server fails, all courts in the nation cannot access case records and are effectively "out of business" for the duration of the outage. This becomes very costly, very quickly as court business becomes backlogged: new cases cannot be entered, hearings may be cancelled, and business partners (attorneys, litigants) cannot be served. The requirements for these services do not disappear, they are simply postponed. It may take days for courts to recover from an outage that lasts but a few hours. Such outages are likely to incur staff overtime costs to eliminate the backlog. To decrease any single point of failure risks at the central site should be architected with a high degree of redundancy in all components; data, communication/network, security, and application service components. This can be achieved simply with technologies of today.

Similarly, the failure of one or more WAN network connections may put one or more courts "out of business" for the duration of the outage. While central server availability is largely within the control of the enterprise (which is entirely responsible for it), the WAN depends upon public utilities such as the local telephone company. Telephone circuits are subject to disruption caused by natural disasters (flood, fire, earthquake, etc.) as well as human errors (such as construction activities, automobile accidents, etc.) which may temporarily destroy portions of the telephone network required to support the WAN. Such outages may last hours or days before the telephone utility repairs the damage and restores service; and such repair activity is outside of the enterprise or court control.

Some disadvantages of a centralized architecture must be mitigated by appropriate service redundancies.

Servers at the enterprise data center may be replicated locally to guard against equipment failure; "fail-over" redundancy. Indeed, the entire centralized data center may be duplicated elsewhere in a remote "disaster recovery site" designed to assume operational control and support in the event of a disaster at the primary location. Some redundancy is necessary to provide adequate level of service to remote courts; redundancy adds cost to the centralized approach but reduces the overall risk of an outage.

WAN circuits to each court must be carefully replicated, as well, using alternate data communication paths so that disruption to one path is less likely to affect the redundant path. Such redundancy requires robust telephone utility services which may not be possible for all courts.

Disadvantages of a centralized architecture include reduced capability to support some important business requirements. Because clients are geographically distributed over hundreds of kilometers, all data exchanged between clients and servers must traverse the WAN. Well designed software architectures are required to minimize the amount of data moving from the client to the servers and back. State of the art architectures as those deployed in Internet based applications are common and are now developed world wide. Any application developed today should be deployable on such enterprise architectures with little change.

1.12.3 Advantages of Decentralization

Decentralization effectively distributes the "eggs" from one "basket" to individual courts. While a server failure may disrupt operations at one court, it does not affect all remaining courts.

Costs of a WAN to connect the courts together and to the SJC are reduced (or eliminated in extreme decentralization). LANs are much more robust than WANs; LAN network outages are much less frequent and, when they occur, do not require the intervention of a third party—the

public telephone company—to repair and restore service; the network is under control of the court/SJC.

Decentralization also facilitates the gradual automation of courts, as funds and technical staff resources become available. It is unnecessary to commit to the development and operation of a sophisticated national data center, fully staffed with technical experts, in advance of the introduction of automation in each court.

1.12.4 Disadvantages of Decentralization

Decentralization requires more server equipment and software to be purchased and maintained, and this equipment is geographically scattered introducing additional personnel and/or travel costs to support. Servers in each court introduce the need for one or more technical specialists in each court to administer and maintain this relatively complex equipment. When looking at all aspects of the enterprise automation program, the costs, i.e., equipment, maintenance, support and risk mitigation, of decentralized systems typically far exceed the cost of centrally implemented, maintained and supported systems.

Backup of the data is typically performed locally by copying the data to magnetic media (e.g. tape) and storing the tapes away from the court. Failure to properly back-up data may put one court at risk, but not the all courts in the nation. (Data backup and remote storage is, of course, required in the centralized model as well, and is even more critical to perform properly if all data is centralized.) It is clear that backups of data in the West Bank project are not always consistent or complete. This must be remedied to insure that court data is not lost in the event of a technology failure.

Distribution of firmware and software updates is costly as technologists must travel to each site to upgrade the system.

Support of the system is costly as any outage requires deployment of an IT professional to the location if none exists. Desired service levels are often put at risk in a decentralized model as an outage may take hours instead of minutes to repair.

Functionality for inter-court and inter-agency communication is not easily implemented. Not impossible but the degree of sophistication built into the function is severely degraded and the sophistication to move the data from one court to another is greatly increased.

1.12.5 Architecture independent considerations

If a “thin client” architecture is implemented, technical staff costs to support desktop equipment in each court is reduced. However, this decision is independent of the decision to centralize or decentralize database servers.

Currently, a decision to implement “thin client” or centralized architecture would require a re-write of the MEZAN application software. The sophistication of skills required to architect, design, build and deploy centralized solutions is significantly higher than traditional software development projects.

1.13 Decision Criteria: Quality of Service

Decisions regarding technology architecture for court automation will affect the cost structure. An architectural choice may reduce costs in one area only to increase them in another.

Costs are incurred to provide a service—the benefits to court operations (described above) that are achieved through automation. Therefore, costs must be related to an independent variable: *Quality of Service (QOS)*. Cost structures of alternative architectures must be calculated (estimated) to identify those costs necessary to achieve an equivalent quality of service regardless of the architecture chosen.

For automated systems, the essential service variables are:

1. Functional content must support the needs of the business. ***This variable is most important.***
2. Supportability. The system should be easily supported from an IT perspective.
3. Maintenance. The system should be maintainable
4. Business Continuity. The risk of information loss is great in automated systems and thus the solution must mitigate all risk in the event of system failure or disaster such as fire, earthquake or flood.
5. Security. The system should provide adequate security as to not permit unauthorized alteration of information.

Achieving targets for these service variables are the key performance indicators (KPI) for any IT department serving a larger business enterprise.

1.13.1 Availability

Availability ("up-time") is the most fundamental and important IT performance indicator. If a system is not available, other KPI (response time and functional content) are meaningless. For mission-critical applications, where the ability to conduct court business is severely compromised if the system is "down", measured availability of better than 99.9% during court business hours in a reasonable target.

High availability is primarily attributable to good infrastructure design and support. However, Applications and Client Support Services have roles, as well. Application design or programming errors ("bugs") may cause portions of the application to be effectively unavailable to end-users while the errors are allowed to persist. And inadequate training or problem-resolution support services make the system effectively unavailable to end-users while the user's ignorance or incapacity is allowed to persist.

1.13.2 Response Time

Response time (the time required for the computer system to complete a user-initiated "transaction") is the second most fundamental IT performance indicator. Slow response time increases the time required for IT users to complete essential court business processes. In addition to increased stress and staff frustration, poor response time may cause adverse economic impact due to paid overtime or additional data-entry staff positions.

Response time should be regularly measured. In general, a measure should demonstrate sub-second response time for the vast majority (99+%) of routine data entry transactions.

Good response time is attributable to adequate infrastructure design and support including network and server capacity. It also requires effective database administration (physical database design, creation of appropriate table indexes, periodic database reorganization, etc.), and ultimately relies on sound application architecture design and programming methods.

1.13.3 Functional Content

Functional content of an organization's computer systems is the third key IT performance indicator. Functional content is proportional to the number of business processes which are assisted by IT. Rich functional content is realized through effective analysis of an organization's business processes, competent application design and programming performance, effective training and support services, good project management to coordinate the foregoing activities, and, ultimately, committed executive support to contribute the necessary resources and authority for application development and organizational change.

1.13.4 Quality of Service Targets

Failure to meet targeted QOS will affect overall organizational costs. For instance, if server or network failures occur, one or many courts may be out of service, disrupting ongoing court services to the public and developing backlogs of work to be performed, the removal of which may incur direct overtime expenses. If response time is slow, the filing of new cases will take longer as will the preparation of notices, the entry of hearing minutes, and the preparation of judgments. Again, backlogs of work may develop and incur additional staff costs to remove. Finally, automated functionality is the entire purpose of the exercise: operations supported by automation are intended to reduce staff costs in the long run and/or to increase services to the public by making employees more productive. Failure to develop or implement automated functions postpones the increased productivity of employees or delays provision of improved services to the public. These delays have costs, too.

General management and the IT department must jointly determine the target quality of service levels to be achieved by the organization. Thereafter, achievement should be regularly measured (e.g. monthly, annually) so that corrective action may be taken to remedy any deficiencies and help avoid organizational costs incurred.

Representative QOS targets:

Availability:

99.9% up-time during court business hours for each court.

Assuming court business hours comprise 10 hours/day, 5 days/week, 52 weeks a year, this level of service would permit courts to be "out of service" due to automated system failures a maximum of 2.6 hours per year.

1.13.5 Response time:

99% of data entry transactions accomplished in 1 second or less; no data entry transaction requiring more than 5 seconds to complete.

99% of document-generation functions completed in 5 seconds or less; no document-generation transaction requiring more than 20 seconds.

99% of statistical reports completed in 10 seconds or less; no statistical report completed in more than 60 seconds.

1.13.6 Functional Content:

Functionality costs are more difficult to measure. While availability and response time are quantitative measures and may be monitored and reported by suitable procedures and software, delay in developing functional enhancements incurs "lost opportunity" costs which are difficult to quantify accurately. What is the lost opportunity cost to the court system if implementation of electronic case files is delayed 5 years? 20 years? The best answer to such questions results from a thorough cost/benefit analysis. In practice, executive decisions to develop and implement new functionality are often based on incomplete information and imprecise quantification of the cost of continuing "business-as-usual". How does one calculate the cost of declining public confidence in the justice system if citizens see modern information technology employed everywhere—in banks, retail stores, airlines, utilities—but does not see it employed in the courts?

1.14 To Decentralize or Centralize?

In comparison to Decentralization, the Centralized model:

- reduces the cost of support and maintenance of distributed server equipment and software;

- increases cost of central servers and the institutional redundancy necessary to achieve QOS targets;
- reduces the number of certain skilled technical support staff employed (server and database administrators);
- increases the cost and complexity of a suitable WAN;
- increases costs for redundant WAN connections necessary to achieve QOS targets; (if the technical capacity does not yet exist in the West Bank's utility infrastructure, the QOS targets may be impossible to achieve);
- increases functionality for management reporting related to consolidated/centralized coding structures.
- Increases functionality for inter-agency communication
- Decreases risk of data loss in the event of an unplanned outage where data restoration is required as backups are centrally managed.

The Decentralized model reverses each of the above metrics.

The Hybrid model adopts some characteristics of each of the extremes in order to provide necessary functionality and good quality of service at a reasonable cost

Table 1 summarizes court business functions supported by the various architectures.

Table 2 summarizes relative costs of the various architectures.

Table 1: Functional Support Offered by Various Architectures:

	Centralized	**Decentralized**	**Hybrid**
Register Automation	Yes	Yes*	Yes*
Document imaging	**Partial**	Yes	Yes
E-communication	Yes	**No**	Yes
Intra-court	Yes	Yes	Yes
Inter-court	Yes	**No**	Yes
Extra-court	Yes	**No**	Yes
Local Court Reports	Yes	Yes	Yes
National Reports	Yes	**No**	Yes
Central Coding Structures	Yes	**No**	Yes

* Cross court transfer functionality requires registry entries be written to both source and target cases. If in a decentralized model this is not possible while in a Hybrid model it would be very difficult to implement as remote databases are not accessible in real-time without a solid wide-area network infrastructure.

Table 2: Relative Costs by Architecture: (assumes equivalent quality of service)

	Centralized	**Decentralized**	**Hybrid**
Networks-WAN/LAN	High	Low	Mid*
Court Technical Support Staff	Low	High	High
Court Customer Support Staff	Mid	Mid	Mid
Court Servers	Low	High	High
Central Servers	High	Low	Mid
Reliability*	High	Mid	Mid
Supportability	High	Low	Mid
Maintainability	High	Low	Mid
Disaster Recovery Risks	Low	High	Mid
Security Risk	Low	High	High

- the more dependent the Judiciary becomes on inter-court transactions the more costly this item becomes, virtually reaching the same cost metric as a centralized solution.

1.15 Software Architecture

This Software Architecture discussion provides a high-level overview of an application architecture and best practices approach to building the Mezan V2 Case Management Application. It outlines the architecturally significant system requirements, philosophical and technical approaches, solutions to common and specific design problems and building blocks to design and build the application.

The strategy and approach outlined in this section should be considered a guide to the Mezan V2 system design and development processes. It is important to note that this is a visionary document and is subject to change as change in the following occur; functional scope, technical requirements, future business and technical requirements.

The primary objective of this section is to provide a high-level architecture for the Mezan V2 application team as they initiate the design and development process. In addition, this section will enable the application team to validate the design approach and make necessary changes to the approach based on the findings and decision at current stage of the project life cycle.

This section will enable the various stakeholders to get a “feel” for the envisioned Mezan V2 application. In addition, this section enables the Project Management, Technical Team, and mainly the Application Team to:

- Gain a better understanding of the integrated Mezan V2 system as a whole
- Validate the technical approach
- Identify and review any ambiguity in the approach
- Define the key software building block components that make up the Mezan V2 application
- Define various Technical Environments and their components

It is important to note that this section of the Vision statement may be iteratively updated as required and when additional information is gathered, application and technical ambiguities are clarified and application and technical decisions are made that impact the high-level approach defined in this deliverable.

Also note that the concepts included herein are advanced and should be addressed by Application Architecture professionals. Qualified persons should have several years of experience designing modern enterprise architectures.

It is highly recommended that the Mezan project staff not intend to design and build architectures such as follows, but rather understand the key concepts presented, and employ an organization that can professionally construct such a solution. Another and preferred approach is to employ an organization that can bring an existing application architecture to the table. This concept of existing architectures is the fastest and most reliable way of realizing an extendable, maintainable and high performing architecture.

1.15.1 Application Architecture

The application architecture section presents a high level coarse breakdown of the system into its most fundamental parts. It should be noted that, at this point in the Mezan V2 Vision, the following view does not represent a static end state of the architectural vision. Rather, this view is dynamic in nature and likely to change as an understanding of what things are architecturally significant evolves over the system's lifetime. The application architecture envisioned should be flexible enough to permit deployment in either a centralized, distributed or hybrid environment. This will permit the Information Technology team to evolve the implementation over time as conditions in the West Bank technology infrastructure improve.

In the review of architectural elements and decisions, one might notice the omission of perhaps the most critical architectural influencer – performance. Indeed many architectural decisions are ultimately about performance in one way or another. At this stage of the architectural evolution, this omission is intentional. Wherever practical, Mezan V2 should adhere to common, industry standard, performance guidelines such as caching user interface elements, minimizing remote calls, and using stored procedures for database access. Aside from these best practices, performance optimization is a topic that should be discussed later in the architectural lifecycle, once core system elements are up and running, baseline measurements can be made, and a disciplined approach can be taken to measure before and after performance implications on a configured environment. Following international best practices for application architecture design will provide the core architecture components necessary for the Information Technology team to assess, measure and tune any issues with performance.

1.16 Application Architecture Elements

These architectural elements represent the core building blocks of the envisioned Mezan V2 application, as it will exist. To facilitate multiple deployment approaches the industry best practice is an architecture layered to permit necessary security, performance and implementation controls. The following layers that will be addressed in this section include the following;

- Application Layering
- User Interface
- Business Services
- Data Access and Service Agent Logic

1.16.1 Application Layering

Before diving into the details of the envisioned Mezan V2 flexible layering strategy, it is first important to clarify the confusion often associated with the use of the terms *layer* and *tier*. Application architects often see the word tier as implying a physical separation. A layer (sometimes referred to as a logical tier), on the other hand, implies separation into discrete concerns (e.g. the domain and data layers). Using these definitions, it is quite easy to imagine an application that has multiple layers (again, a domain and data layer) but only a single tier. Such is the case when the entire application is deployed on a single laptop.

Using the above definitions, application layering becomes a critical element of the application architecture. As such, it will be discussed in this section. An application's tiers, on the other hand, are a product of the physical deployment strategy and will be discussed in the technical architecture/deployment discussion initiated as a result of this vision statement.

Breaking down an application into layers is a fairly well understood process with well-documented benefits. These benefits include:

- The ability to minimize dependencies between layers
- The option to substitute different implementations for a particular layer (e.g. a Web-based presentation layer)
- The power to separate concerns, allowing User Interface specialists to handle the presentation layer while business analysts handle the domain logic. Separating concerns has a very positive impact on the maintainability of the application.

There are also several downsides to application layering. Foremost amongst these is that the application incurs a small performance hit.

As applications have evolved from mainframe to client/server to Web-based applications, the layering strategies have evolved along with them. For modern applications based on the Microsoft .NET Framework, applications should follow Microsoft's recommendation to use three primary layers:

Presentation Layer – Handles the provision of services and the display of information. Typically in .NET, the presentation layer is implemented using Windows Forms or Web Forms. This interpretation of this layer can be extended however to include .NET Web Services presented to external systems or a Batch API that handles the coordination of batch services.

Domain or Business Layer – Handles the application's domain logic. This involves performing calculations based on inputs and stored data, validation of any data that comes in from the presentation layer, and figuring out exactly what data logic to invoke, depending on the commands received from the presentation layer.

Data Layer – Responsible for communication with various data sources. These sources may include relational and non-relational databases, messaging systems, transaction managers, and XML-based Web services.

In addition to the three primary layers, Mezan V2 should also have a separate facility for handling actions that are not anchored in any particular layer. These actions are commonly referred to as cross-cutting concerns and cover areas such as error-handling, security, logging, and transaction management.

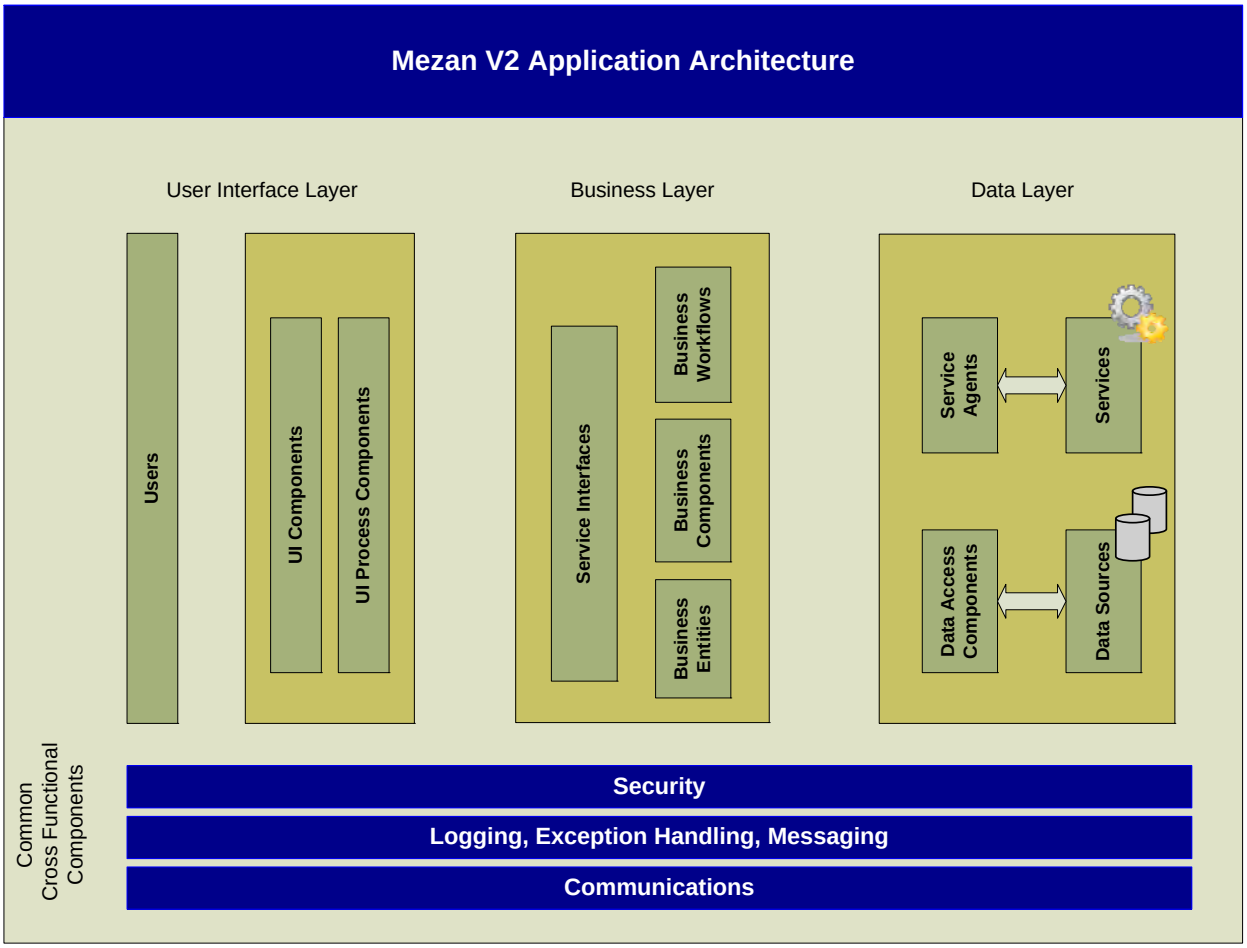


Figure x Mezan V2 Layered Architecture

1.16.2 User Interface Layer

The User Interface layer, as it is commonly implemented in modern enterprise applications, covers both the users' direct interaction with the application and some mechanism to govern the various interaction states available through the layer. In the case of the Mezan V2 application, the User Interface layer might use either one or a combination of ASP.NET Web Forms and/or VB.Net Forms for user interaction and some type of *model-view-controller (MVC)* pattern or web services pattern to govern interaction states for the user interface. As Case Management Applications in general are typically very user interface intensive the technique used most often is that of a windows client interface or in the case of Microsoft, VB.NET. Some functions of the overall Judicial process might lend themselves to a web interface and thus should be considered in the overall architecture.

User Interface – The user interface governs the synchronization and orchestration of user interactions. By providing this level of abstraction, process flow and state management logic is not hard-coded to the various presentation elements themselves. In addition, the same basic user interaction "engine" that governs predictable patterns of interaction can be reused across multiple presentation sessions. In the Mezan V2 application, the user interface could be modeled along the lines of *model-view-controller* pattern. This pattern defines a series of interactions between components with segregated responsibilities whereby a controller handles the user's request, gets the model to perform the domain logic, and then invokes the

view to create a response based on the model. Another approach and one similar to that of the Jordanian solution is to leverage a VB.NET web services model.

1.16.3 Business Layer

The Mezan V2 Business Layer forms the core of the functionality provided by the application. In Microsoft .NET applications, this business layer is usually encapsulated in methods of .NET components, which may be invoked synchronously or asynchronously. Since the business layer is responsible for such extensive application functionality, it is often difficult to discern where this layer's logic ends and other logic (user interface or data, for example) begins. For example, a sure sign that user interface logic had seeped into the business layer would be if business logic needed to be duplicated to enable this switch in the implementation of a particular Mezan V2 layer. To fully realize the benefits of the envisioned Mezan V2 layering strategy, it is best to avoid any such commingling of logic between the layers.

1.16.4 Data Layer

The architectural options available for the data layer are often the subject of some of the most heated architectural disputes. Most of this debate stems from the fact that certain groups hold particular data transport logic implementations to be universal truths that are always the best option irrespective of the business domain, application complexity and developer skill set. There really are no universal truths in the data transport logic realm; instead, there are implementations that are best suited to particular applications. Here, there is a need to identify the data transport logic implementation best suited to the needs of the Mezan V2 application, its business layer complexity, and environmental factors.

Microsoft outlines three primary implementation options for data transport:

- Use an XML string or an XML Document Object Model (DOM) object to represent business entity data

- Use a DataSet (an in-memory cache of tables) or a DataReader to represent business entity data retrieved from the database.

- Use a custom class to represent each type of business entity. This custom class may or may not implement Create-Read-Update-Delete (CRUD) behaviors within the class.

The use of the XML option should be restricted to prototype development or very small scale development, leaving the use of DataSets/DataReaders and custom classes for data transport. This does not, however, preclude the use of custom classes; DataSet, DataReader or custom classes will be used as appropriate to pass data between the various Mezan V2 layers.

1.17 Application Architecture Components

This section identifies specific application architecture components that should be used to design and build the Mezan V2 application. As each application architecture component is discussed, this section describes the options available and the possible Mezan V2 usage approach for implementation. The key driving factors for the application architecture comes from the architectural goals that include performance, scalability and maintainability. In addition, a unified approach to the architecture leads to a modular and object oriented application design.

This section is organized as follows:

- Core Framework for the Mezan V2 application describes the basic plumbing necessary to enable the application, this basic plumbing describes how database layer objects will be accessed from the application layer, how information will be cached in memory to boost performance, session information handling techniques

- Security Architecture describes how the Mezan V2 application should provide authentication and authorization services using the framework's security application block.

1.17.1 Core Framework

The goal of a Mezan V2 framework is to incrementally collect and abstract best practices for designing and building efficient distributed solutions on a consistent code base using the Microsoft .NET platform.

This application framework is a set of base services and infrastructure code that allow the application developer to focus more on actual application functionality rather than "plumbing" code. The framework is flexible and extensible so it can be modified to fit particular project needs.

1.17.2 Data Access Application Block

The data access application block provides access to the most frequently used features of ADO.NET with applied best practices. The Mezan V2 application architecture data access layer should include:

- Transparency and Consistency in the application code to interact with the Mezan V2 database

- Mechanism to place an indirection between a logical database instance and a physical database instance

- Straight forward and easy way to adjust and validate the database configuration settings

- Securely store connection strings that includes user Id and password information

The Data access block helps to meet these needs by providing a simple way for developers to use ADO.NET quickly and correctly. This block uses the cryptography application block and configuration system to securely store connection strings. The Mezan V2 strategy to implement the data access block would follow the guidelines given in .NET Data Access Architecture Guide and Microsoft Enterprise Library documentation for Data access block.

1.17.3 Exception and Error Management

The Mezan V2 core framework should implement an Exception Handling Application Block to create a consistent strategy for processing exceptions that occur throughout the architectural layers of the application. The Mezan V2 architecture should follow the guidelines detailed by the Microsoft Exception Management Guide and the Enterprise Library Exception Handling Application Block Documentation.

The exception handler would be capable of the following:

- Detecting exceptions
- Logging and reporting information
- Generating events that can be monitored externally to assist system operation

This exception block should encapsulate and abstract the details of logging and reporting from the application's business logic. This can quickly and accurately notify operators of any problems the application is experiencing, and can provide valuable information to assist developers and support services with problem resolution.

The design of the this handler provides a consistent means of detecting exceptions, logging and reporting information, and generating events that can be monitored externally to assist system operation. The Exception Handling Application Block encapsulates the logic used to perform the most common exception handling tasks into minimal application code. It relieves developers of the requirement to write duplicate code and custom code for common exception handling tasks. Furthermore, the Exception Handling Application Block allows exception handling policies to be changed after they have been deployed and to ensure that changes happen simultaneously and consistently.

Proposed Mezan V2 Exception Handling Approach

The Exception Handling Application Block should be designed to support the typical code contained in *Catch* statements in application components. It should provide commonly used exception-handling functions, such as the ability to log exception information, the ability to hide sensitive information by replacing the original exception with another exception, and the ability to add contextual information to an exception by wrapping the original exception inside another exception. Instead of repeating this code throughout identical catch blocks in an application component, the application block would allow developers to encapsulate this logic as reusable exception handlers. Exception handlers are .NET classes that encapsulate exception handling logic and implement the Exception Handling Application Block interface.

1.17.4 Logging and Instrumentation

Logging and Instrumentation provides the needed infrastructure in the Mezan V2 application to handle anticipated and unanticipated events. The examples include operating system exceptions and specific application errors. The ability to record and retrieve diagnostic information is an important characteristic of robust software. Diagnostic information may include the amount of time needed to execute a critical method, the number of transactions committed per second, or the number of users currently with active sessions.

Logging and instrumentation can be built into a .NET application using a centralized Logging Application Block.

Logging Application Block:

The Logging Application Block enables us to incorporate the standard logging functionality in a Microsoft.Net application.

Possible Mezan V2 Approach for Logging and Instrumentation

Mezan V2 should implement Logging and Instrumentation using a Logging Application Block. This Application Block will provide logging and instrumentation support in number of ways:

- Help maintain consistent logging and instrumentation practices
- Provides the application the option of logging the events to a variety of locations.
- Provides implementations that you can use to solve common application logging and instrumentation problems.
- Logging and instrumentation application blocks should be extensible, supporting custom implementations of formatters and event sinks.

1.17.5 Caching

Caching is storing data in memory for quick and rapid access. Typically, information that is costly to obtain (in terms of performance) is stored in the cache. For enterprise-scale distributed applications, Caching can help overcome following challenges:

- **Performance:** Caching improves application performance by storing relevant data as close as possible to the data consumer, rather than relying on repeated database calls. This avoids repetitive data creation, processing, and transportation.
- **Scalability**: Storing information in a cache helps save resources and increases scalability as the demands on the application increase.
- **Availability**: By storing data in a local cache, the application may be able to survive system failures such as network latency, Web service problems, and hardware failures.

Effective caching techniques will decrease the demand on the Web and database server's system resources and increases the overall application scalability.

Proposed Mezan V2 Approach for Caching

Mezan V2 should use a core frameweork Caching Application Block to provide a configurable caching solution. The caching architecture should follow the guidelines detailed by Caching Architecture Guide for .NET Framework Applications.

The Caching Application Block lets developers incorporate a local cache in their applications. It supports both an in-memory cache and, optionally, a backing store that can either be the Enterprise Library Data Access Application Block or isolated storage. The application block can be used without modification and provides all the functionality needed

to retrieve, add, and remove cached data. Configurable expiration and scavenging policies are also part of the application block's functionality.

The overall architecture of the caching block is shown below:

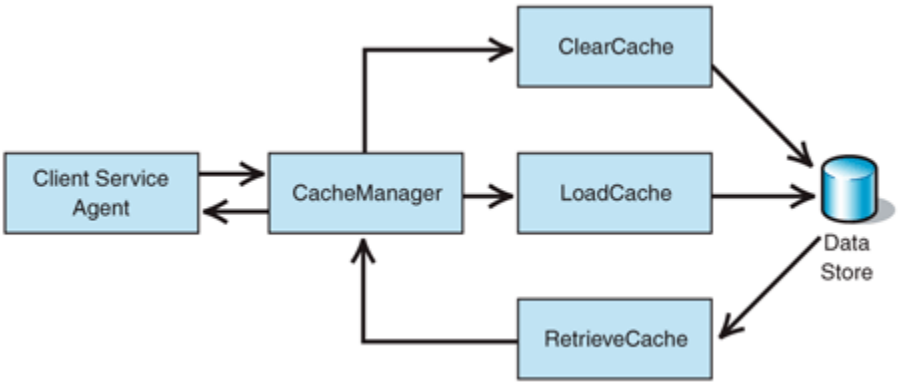


Figure 2.1.4- 1 Overall architecture of Caching Application Block

Design of the Caching Application Block

The Caching Application Block should be designed to achieve the following goals:

- To provide a set of APIs that is manageable in size
- To allow developers to incorporate the standard caching operations into their applications without having to learn the internal workings of the application block
- To be easily configurable, using the Enterprise Library Configuration Console
- To perform efficiently
- To be thread safe
- To ensure that the backing store remains intact if an exception occurs while it is being accessed
- To make sure that the states of the in-memory cache and the backing store remain synchronized

1.17.6 State Management

As the design of the Mezan V2 application should consider function over a web based interface, the application should be designed in a stateless architecture. Each request for a form is treated as a new request, and information from one request is not available by default to the next request. To help overcome this inherent limitation of modern applications, .NET includes a number of features for managing state—that is, for storing information between requests.

State information used for the following purposes, including but not limited to:

- Maintain key user identification information across transactions/ processes that span multiple screens/systems
- Associate individual information with a specific user session, or screen a user is accessing

- Manage user security access
- Maintain the status of a single transaction/ process that spans multiple screens/ systems
- Enforce timeout policies after a period of inactivity

Maintaining user session contexts and states in an efficient manner is an integral component of application design from both a usability and system performance perspective. Determining the most appropriate way to maintain user session contexts and states is dependant on the size of the information contained in the session or session context, the duration and complexity of the session, and the configuration of the server and client machines.

State Management Options in Microsoft.Net

Microsoft.Net technology presents three modes that help maintain Session State in an application. These are based on the capabilities provided by the Microsoft .NET framework:

- **In-process Mode**: In this mode session state is managed in process and is maintained on the Web server. Even though this mode has performance advantages over other modes, if the process is re-cycled, state is lost. Also, this state does not scale out seamlessly in a Web farm setting
- **Out-of-process Mode**: This mechanism involves storing state on another process running as the memory of a service. This Session state service or process can be configured to be shared across processors in one server - "Web Garden" mode or across multiple servers in a "Web Farm" mode. This provides the right balance between performance and reliability. This scales out seamlessly in a Web farm setting
- **SQL Server Mode:** This mechanism involves storing state and persisting it on the SQL server database. Even though this has very high reliability, it affects the overall performance of the system

Proposed State Management Approach in Mezan V2

No particular approach is recommended in this vision document and should be studied thoroughly when conducting the detailed design of the Mezan V2 architecture. This concept should be thoroughly addressed and included in the core framework.

1.17.7 Security

The following discussion of Security is for discussion purposes only. No specific security approach is recommended here as it should be part of a thorough detailed architecture specification. It is presented here to illustrate one such core framework security approach and advantages therein.

Security Application Block (SAB) Design

The MS Enterprise Library Security Application Block should be used to implement authentication, authorization and other security-related functionality in the Mezan V2 application.

The Security Application Block has the following features:

- It reduces the requirement to write boilerplate code to perform standard tasks.

- It helps maintain consistent security practices, both within an application and across the enterprise.
- It eases the learning curve for developers by using a consistent architectural model across the various areas of functionality provided.
- It provides implementations that can be used to solve common application security problems.
- It is extensible; it supports custom implementations of security providers.

Following diagram shows the interrelationships between the key classes in the Security Application Block.

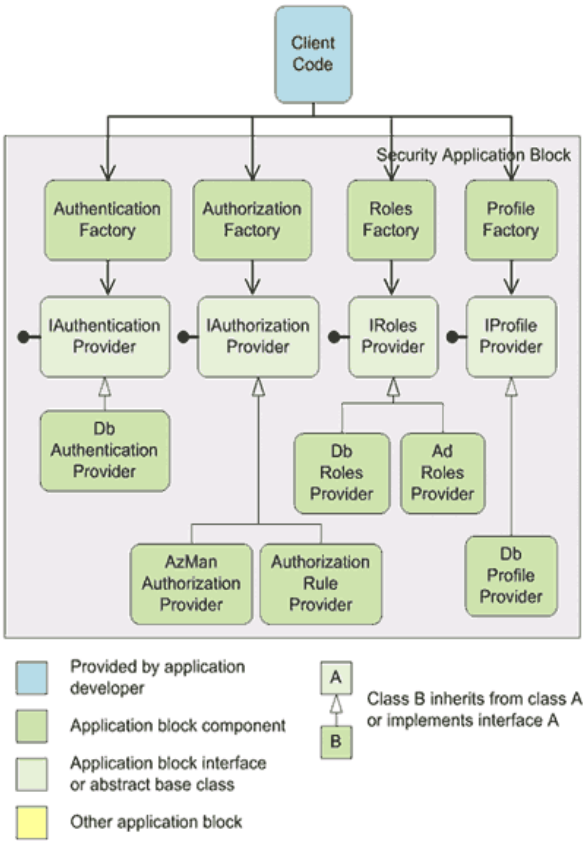


Figure 2.1.6- 1 Key Components of Security Application Block

The Security Application Block will be used to implement the fine-grained authorization for the Mezan V2 application. It will provide all the plumbing required for security like authorization, authentication, creating new users, assigning roles and rules and other security chores. Custom extensions should be built with Caching, Configuration and Data Access blocks to support the various security requirements.