



USAID
FROM THE AMERICAN PEOPLE

Development Experience Clearinghouse
SUBMISSION FORM

(If submitting electronically, the "comments and missing bibliographic elements" box replaces this form.)

USAID award number (contract, cooperative agreement, grant, etc.): DFD-I-01-04-00173-00	
Strategic Objective (SO) title: Promote Democratic Reform	SO number: 11
Project title: NETHAM Rule of Law Program / Justice and Enforcement	Project number:
Document title/translated title: Justice Record Automation System at the Palestinian Ministry of Justice (MOJ) Systems Analysis – Software Requirements Specification (SRS)	
Author(s): Netham Project	
Contractor or grantee name(s): DPK Consulting / a Division of ARD Inc.	
Sponsoring USAID operating unit(s): USAID West Bank / Gaza – Democracy and Governance	
Language: English	Publication date: April 2008
Abstract <i>(summary of most significant information, 250 word limit; optional):</i>	
Keywords <i>(suggested terms to describe content of document; optional):</i>	

Contact information for person submitting document:

Name: Nabil Isifan	Email: nisifan@netham.net
Telephone number: +972599266441	Today's date: 21/10/2009



West Bank and Gaza

NETHAM

Rule of Law Program

Justice and Enforcement

DFD-I-01-04-00173-00

Implemented by DPK Consulting

**Justice Record Automation System at the
Ministry of Justice (MOJ)**

Systems Analysis – Software Requirements Specification (SRS)

April 2008

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

Table of Contents

1.	Introduction.....	5
1.1	Purpose	5
1.2	Scope	5
1.3	Definitions, Acronyms and Abbreviations	5
2.	Positioning	5
2.1	Business Opportunity	5
3.	Stakeholder Descriptions	6
3.1	Stakeholder Summary.....	6
3.2	System User Summary	7
4.	Product Overview	8
4.1	Product Perspective	8
4.2	Assumptions and Dependencies	9
4.3	Needs and Features.....	10
5	Overall Description	12
5.1	Use-Case Model Survey	12
6.	Specific Requirements	16
6.1	Use Case Specifications.....	16
6.1.1	Login Use case:	17
6.1.2	Backup and Recovery Use case:	17
6.1.3	Design Outputs Use Case:.....	18
6.1.4	Feed Criminal Data Use Case:	20
6.1.5	Feed Security Info Use Case	22
6.1.6	Maintain Privileges Use Case:	22
6.1.7	Maintain Security Roles Use Case:.....	24
6.1.8	Maintain Users Use Case:	26
6.1.9	Manage Application Requests Use Case:.....	27
6.1.10	Manage Authentication Requests Use Case:.....	29
6.1.11	Manage Data Feeding Requests:	29
6.1.12	Manage Registration Requests Use Case:.....	30
6.1.13	Modify Criminal Data Use Case	31
6.1.14	Post Application Request Use Case:	33
6.1.15	Post Authentication Requests Use Case:.....	34
6.1.16	Produce Formal Documents Use Case:.....	34
6.1.17	Register Use Case:.....	35
6.1.18	Search and View Criminal Data Use Case:.....	36
6.1.19	Update Basic Data Use Case:.....	37
6.1.20	Update Detention Information Use Case:.....	38
6.1.21	View Auditing Reports Use Case:.....	38
6.2	Supplementary Requirements.....	40
	Appendix B – Use Case Diagram	43
	Appendix C – Data Flow Diagrams.....	44
	Appendix D – Conceptual ERD	48
	Appendix E – State Charts.....	49
	Appendix F – Business Model.....	50
	Appendix G- Registration Form Model.....	51

1. INTRODUCTION

This document collects, analyzes, and defines high-level needs and features of the Justice Record Automation System. It focuses on the capabilities needed by the stakeholders, and the target users, and why these needs exist. The details of how the Justice Record Automation System fulfils these needs are explained in the use-cases and supplementary specifications.

1.1 PURPOSE

The purpose of this document is to define the detailed requirements of the proposed system in terms of the needs of the end users and how the system is supposed to behave.

1.2 SCOPE

This document applies to the Justice Record Automation System, which will be analyzed and designed by Netham. This system will allow beneficiaries to request formal documents concerning the Justice Record (if any) in a robust, transparent, and time efficient manner. Other government centers will feed the system with daily updated information.

1.3 DEFINITIONS, ACRONYMS AND ABBREVIATIONS

See the Glossary. Appendix A

2. POSITIONING

2.1 BUSINESS OPPORTUNITY

This project will replace the current scattered and incomplete criminal archiving practices with a state-of-art, online, robust and secure system.

Current criminal archiving practices are time consuming, resulting outputs are sometimes misleading and can reach dead-end and also resulting unnecessary and long delays in obtaining "non-conviction clearance certificate" needed by the citizens

2.2 PROBLEM STATEMENT

The current structure and workflow for producing the formal documents related to criminal history is incomplete and requires a deep investigation and re-structuring. There is no automated system for Justice Records and our proposed future system shall start from scratch.

The problem of	The outdated, largely manual, and time consuming Justice Record practices in Palestine.
Affects	All kinds of people in the Palestinian society.
the impact of which is	<ul style="list-style-type: none"> • Current practices in producing formal documents concerned with Justice Records is very primitive and incomplete. • Internal workflows are inconsistent and no formal or trusted source of criminal information • Criminal information is scattered and sometimes not present. • The whole process of producing formal certificates is time consuming • Personal judgment is much involved • Information sources are not trusted. • Current criminal data is in different formats (soft and hard copies) • No information security in place. • Sometimes the resulting information about criminal history is contradictory.
a successful solution would be	<ul style="list-style-type: none"> • Building an archiving criminal automation system that can process inputs from various feeding centers • This system shall be able to track civilians with criminal history • System shall be, robust, secure and provides accurate information. • The system shall be centralized and provides online access to the information to produce reports and formal certificates • The system shall provide levels of authorization and applies certain polices of security • The system will serve as a core for e-government future system.

3. STAKEHOLDER DESCRIPTIONS

3.1 STAKEHOLDER SUMMARY

Name	Description	Responsibilities
Civilians	Local civilian living in Palestine	<ul style="list-style-type: none"> • Post a Request to view Justice Record • View Justice Record
MOJ - Ministry of Justice		<ul style="list-style-type: none"> • Manage registration requests • Produce formal documents • Manage administration • Manage data authorization
MOI - Ministry of Interior		<ul style="list-style-type: none"> • Feed basic civilian data • Register to view criminal info

Name	Description	Responsibilities
Public Prosecution		<ul style="list-style-type: none"> • Register to view criminal info • Feed criminal data
Conciliation Court		<ul style="list-style-type: none"> • Register to view criminal info
First Instance Court		<ul style="list-style-type: none"> • Register to view criminal info
Religious Court		<ul style="list-style-type: none"> • Register to view criminal info
Bureau of Personnel (Diwan)		<ul style="list-style-type: none"> • Register to view criminal info
Borders' Administration		<ul style="list-style-type: none"> • Register to view criminal info
Traffic and Licensing Department		<ul style="list-style-type: none"> • Register to view criminal info
Monetary Authority		<ul style="list-style-type: none"> • Register to view criminal info
Private sector and NGOs		<ul style="list-style-type: none"> • Register to view criminal info

3.2 SYSTEM USER SUMMARY

Please note that the following users' names are based on roles (hint: we can assign the same user more than role, and we can change the roles because the whole administration and security module is dynamic)

Name	Responsibility
Special Unit (for examples at MOJ):	
<ul style="list-style-type: none"> • Administrator 	<ul style="list-style-type: none"> • Manage users and security: privileges, security authorization, manage roles, create new roles, etc • Manage system start up and shutdown • Design output documents: format, content, style of output documents and reports • Backup and Recovery
<ul style="list-style-type: none"> • Data-Authorization Personnel 	<ul style="list-style-type: none"> • Manage data feeding requests: accept or reject data feedings (centralization) • Monitor all transaction (read, write (delete and update)) done on the system data.
<ul style="list-style-type: none"> • Registration-Authorization Personnel 	<ul style="list-style-type: none"> • Manage registration requests
<ul style="list-style-type: none"> • Formal-Documents Personnel 	<ul style="list-style-type: none"> • Produce formal documents (e.g. عدم المحكومية) • Produce data reports • Manage Requests
Criminal-Data Viewer	<ul style="list-style-type: none"> • Post registration requests • View Criminal information • Post Requests

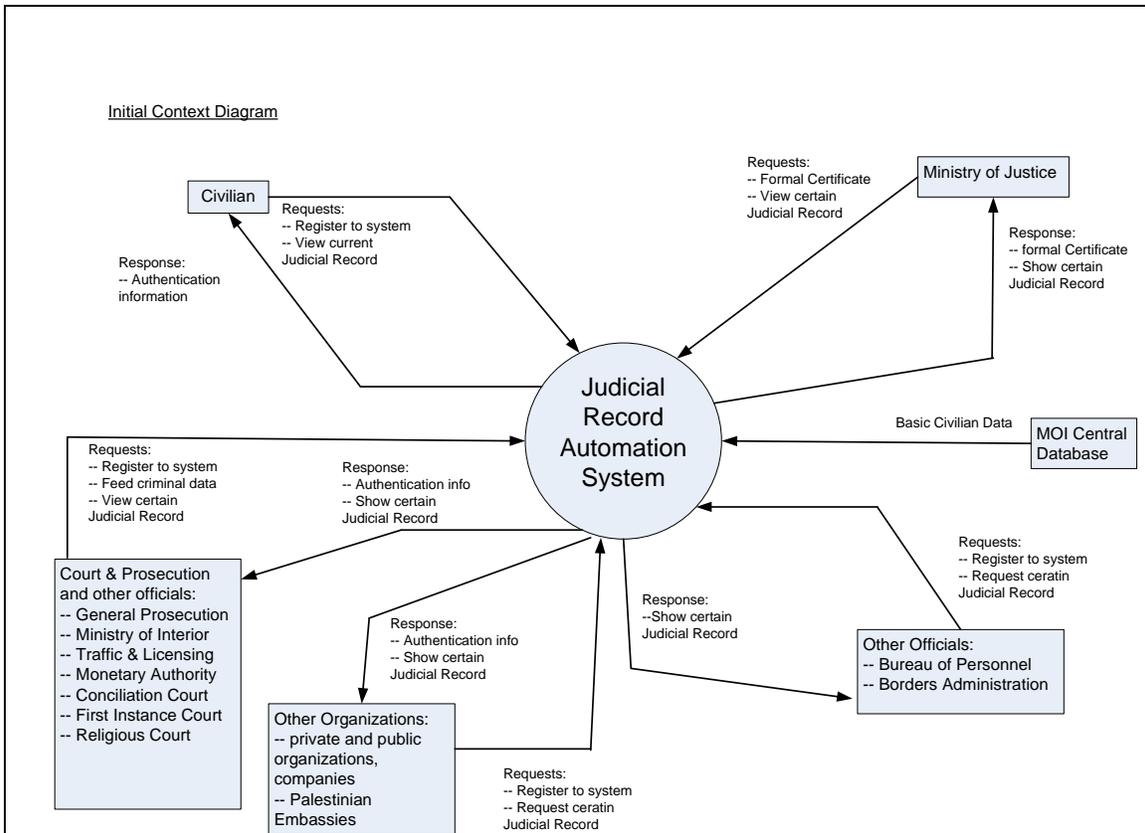
- Criminal-Data Feeder
 - Feed criminal data
 - Change criminal data
 - View Criminal information
- Security-Data Feeder
 - Feed security data
 - Change security data
- Detention Facility Officer
 - Update detention information

4. PRODUCT OVERVIEW

This section provides a high level view of the Justice Record Automation System, interfaces to the external PNA Central Database (MOI) and the system configuration.

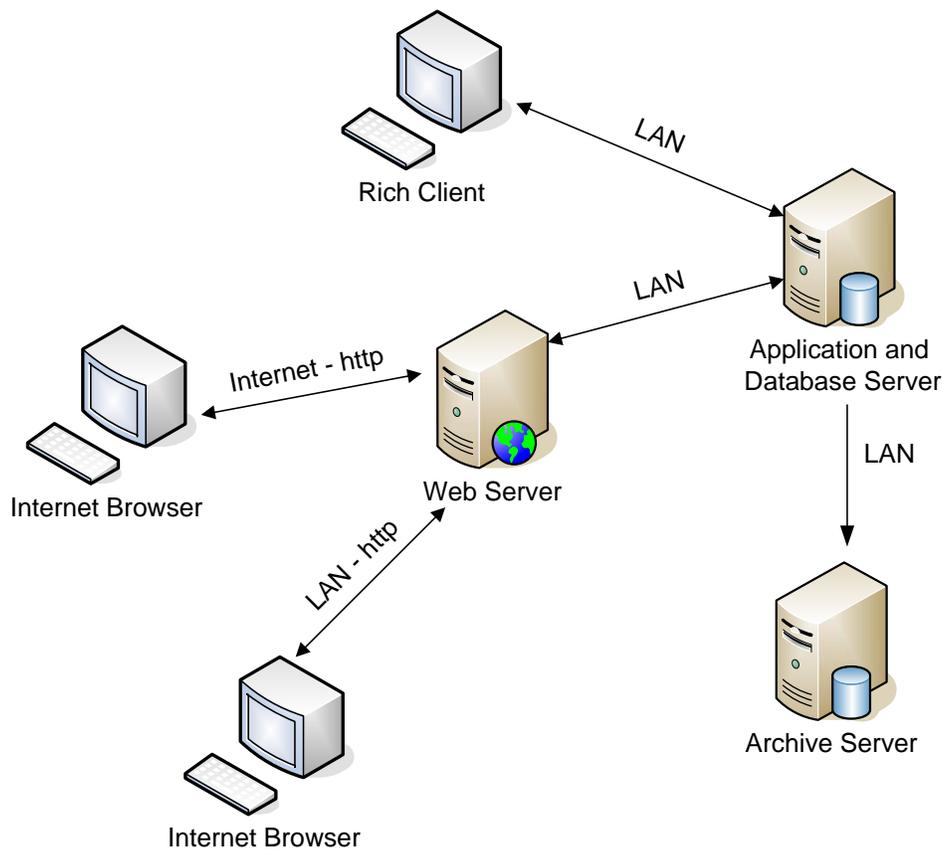
Please note that the proposed system will be implemented and delivered in releases. The first release shall integrate the Justice Records and all related system behaviour. Later releases shall integrate security data and all related system behaviour.

4.1 PRODUCT PERSPECTIVE



Context Diagram

System Overview



System Overview

4.2 ASSUMPTIONS AND DEPENDENCIES

The following is a list of main business assumptions by which if changed, will alter the Vision document.

1. The central database of MOI represents the core of our future database. The proposed system shall be able to access this database. Our system shall be able to automatically import data form MOI database very certain amount of time.
2. Based upon the first assumption, our system shall contain information about all civilians, both convicted and non-convicted
3. There should be a special unit that controls and permits the feeding of data according to certain rules (centralization). Any criminal data entered or updated must be accompanied with evidence documents. These documents will be scanned

and entered into the system. The special unit will look at these documents and then decides whether to submit the data to the system or reject it.

4. All government organizations, and NGOs and private sectors must register to the proposed system before they can post requests to view certain Justice Record, or feed our system with new information.
5. The first release of the proposed system shall permit manual data feeding of criminal data from various feed centers such as prosecution.
- 6.

4.3 NEEDS AND FEATURES

The following is a list of high level features and capabilities of the system. Please note that detailed functional requirements will be captured in the use cases.

Need	Priority
Since the first stable release will serve as the core for future e-government system; it should be designed and built using Component Based Architecture (CBA) enforcing separation of concerns. Otherwise it will be hard to maintain, extend, or understand.	High
System shall integrate basic civilian data from the database of MOI.	High
The system should be secured allowing only authenticated and authorized users to access specific functional areas.	High
System shall provide dynamic functionality in building, updating and deleting privileges, roles and user groups. Access rules should be dynamic.	High
The system shall be accessible through the Internet from PCs having any kind of internet browser. Other users such as Administrators should have special rich user interfaces (e.g. windows) to allow them to do special administrative functions (i.e backup, recovery, etc)	High
System shall allow scanned documents to be entered via scanners, and shall provide digital signature to track these scanned documents. Also, the system shall be able to output document via fax machine.	High
System shall allow data authorization administrators to accept or reject data entered to the system from other feeding centers. Any data fed to the system must be authenticated by the data administrators.	High
System shall allow certain users to register themselves in the system. The special unit personnel will decide later on whether or not to accept those users. If they were accepted, the system shall produce usernames and passwords for those users and send them via email to the users they belong to.	High
System shall allow registered users to post requests for criminal histories. These requests will be reviewed by the special unit, and they will send the replies via fax, or email later on.	High
System usability should be appropriate to normal users with little experience in computers and should be easy to use and navigate.	High
First release of the system will not include the system behavior related to security data.	High
The first release should permit manual feeding of data from several locations. Later releases shall have the ability to integrate data form other resources.	High
System shall permit certain official users to modify data they entered previously by them. But these modifications will not be accepted until the data administrators accept it.	High
System should uniquely identify civilians according to their identities, and if it is not present, then the full name plus date of birth should be used.	High
System should be bi-lingual (English and Arabic) in both the database and interfaces to users.	High

Need	Priority
System shall be able to produce formal document that states whether that civilian’s criminal history is clear or not. Other formal documents such as Non-Conviction shall only be produced if the status of that citizen is “clear”.	High
System shall provide special functionality for administrators to dynamically change the design, format and content of the output formal certificates of the system	High
Only the Ministry of Justice is authorized to produce the formal output documents.	High
System shall include online help for users. Users should not require hardcopy of manual or others.	High
System shall provide a mechanism to monitor data entry, data update and data browsing to ensure security and trace back and changes done on the system’s data.	High
System shall be available 24 hours a day, 7 days a week.	High
System shall monitor certain information after it is entered to the system, and most likely that the system will modify that data (delete or change its status). This will be cleared in the next versions and will be captured in more details in the use cases.	High
System shall provide administrative functionality to allow authentication and authorization of users and user groups, granting or provoking privileges.	High

5. OVERALL DESCRIPTION

5.1 USE-CASE MODEL SURVEY

Use case list. Please refer Appendix B - “Use case model”.

Use case name	Brief description
1) Login use case	<p>This use case allows system users to login the system.</p> <p>The actor for this use case is Administrator, Criminal Data Feeder, Security Data Feeder, Data Authorization Personnel, Detention Data Officer, Formal Documents Producer, and Registration Authorization Personnel</p>
2) Backup and Recovery Use Case	This use case allows the Administrator to manually backup and recover the system

Use case name	Brief description
3) <i>Design Outputs Use Case</i>	<p>database. The actor for this use case is the Administrator</p> <p>This use case allows the Administrator to add, delete and modify system output documents. The actor for this use case is the Administrator.</p>
4) <i>Feed Criminal Data Use Case</i>	<p>This use case allows Criminal Data Feeder to add criminal information for certain citizen. The actor of this use case is Criminal Data Feeder (CDF)</p>
5) <i>Feed Security Info Use Case</i>	<p>This use case allows Security Data Feeder to modify security information for certain citizen. The actor of this use case is Security Data Feeder (SDF)</p>
6) <i>Maintain Privileges Use Case</i>	<p>This use case can extend the use case “Maintain System Security”. This use case allows the Administrator to add, delete and modify system privileges. Please note that the “details” information is about data forms and system functions (such as design outputs, view auditing reports, etc) and their access rights viewed as a table, this can be negotiable, and the designer can suggest more insight.</p> <p>What I intend to show in the two use cases (Maintain Security Roles and Maintain Privileges) is that the security permissions are dynamic, and we can change, create, modify them as we need.</p>
7) <i>Maintain Security Roles Use Case</i>	<p>The actor for this use case is the Administrator</p> <p>This use case allows the Administrator to add, delete and modify system roles information and their privileges. The actor for this use case is the</p>

Use case name	Brief description
8) <i>Maintain Users Use Case</i>	<p>Administrator</p> <p>This use case allows the Administrator to add, delete and modify system users' information. It also allows the administrator to change the users' privileges of security. This use case can be extended by Manage Authentication Requests use case as well.</p> <p>The actor for this use case is the Administrator.</p>
9) <i>Manage Application Requests Use Case</i>	<p>This use case allows the Formal Document Producer to manage requests sent by clients to view certain Justice Record. They can view later on their Justice Record via email, fax or formal document (hard copy).</p> <p>The actor for this use case is Formal Document Producer (FDP)</p>
10) <i>Manage Authentication Requests Use Case</i>	<p>This use case allows the Administrator to manage authentication requests sent by Registration Authorization Personnel.</p> <p>The actor for this use case is Administrator</p>
11) <i>Manage Data Feeding Requests Use Case</i>	<p>This use case allows the Data Authorization Personnel to manage criminal data additions and updates.</p> <p>The actor for this use case is Data Authorization Personnel (DAP).</p>
12) <i>Manage Registration Requests Use Case</i>	<p>This use case allows the Registration Authorization Personnel to manage registration requests sent by clients to get authorized access to the system.</p> <p>The actor for this use case is Registration Authorization Personnel (RAP).</p>
13) <i>Modify Criminal Data Use Case</i>	<p>This use case allows Criminal</p>

Use case name	Brief description
14) <i>Post Application Request Use Case</i>	<p>Data Feeder to modify criminal information for certain citizen. CDF can only modify criminal data entered previously by him. The actor of this use case is Criminal Data Feeder (CDF)</p> <p>This use case allows the citizen or any organization to post request to only view certain Justice Record. They have to post a request every time they need to view certain Justice Record.</p> <p>The actor for this use case is the Criminal Data Viewer (CDV).</p>
15) <i>Post Authentication Request Use Case</i>	<p>This use case is extended by Manage Registration Requests use case. This use case allows the Registration Authorization Personnel (RAP) to post a request to the Administrator asking him to create system users with certain privileges to feed and/or modify and/or view criminal info.</p> <p>The actor for this use case is the Registration Authorization Personnel (RAP).</p>
16) <i>Produce Formal Documents Use Case</i>	<p>This use case allows the Formal Document Producer to produce formal document such as “good of conduct,” to certain citizen. The actor for this use case is Formal Document Producer (FDP).</p>
17) <i>Register Use Case</i>	<p>This use case allows Criminal Data feeder and Officials to register to our system so that they can access have regular access to the system so that they can feed and/or search and view criminal information.</p> <p>The actor for this use case is the Criminal Data feeder (CDF), Detention Facility Officer, Security Data Feeder.</p>
18) <i>Search and View Criminal Data Use</i>	<p>This use case allows the Formal</p>

Use case name	Brief description
<i>Case</i>	Document Producer and Criminal Data Feeder to search and view citizen's records and criminal information.
19) <i>Update Basic Data Use Case</i>	The actor for this use case is Formal Document Producer (FDP). And Criminal Data Feeder (CDF) This use case allows the Registration Authorization Personnel to update basic citizen data. Allowing him to add new information that was not present basic data was imported from MOI database.
20) <i>Feed Criminal Data Use Case</i>	The actor for this use case is the Registration Authorization Personnel (RAP). This use case allows the Detention Facility officer to update the detention information for the criminals that are or were held in that facility and in that facility only.
21) <i>View Auditing Reports Use Case</i>	The actor of this use case is Detention Facility Officer (DFO). This use case allows the Data Authorization Personnel to view auditing reports to monitor activities done on the system. The actor for this use case is Data Authorization Personnel (DAP).

6. SPECIFIC REQUIREMENTS

This section of the Software Requirements Specification contain all the software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements and testers to test that the system that satisfies those requirements. Since use-case modeling is used, these requirements are captured in the use case specifications.

6.1 USE CASE SPECIFICATIONS.

Please also refer to Appendix C - DFDs

6.1.1 Login Use case:

a. Brief Description

This use case allows system users to login the system.

The actor for this use case is Administrator, Criminal Data Feeder, Security Data Feeder, Data Authorization Personnel, Detention Data Officer, Formal Documents Producer, and Registration Authorization Personnel

b. Flow of Events

The use case begins when the User types his/her name and password on the login form

b.1. Basic Flow – Login to System

1. User selects the “submit” option
2. System validates the user’s username and password, and logs him. to the system.
3. System loads the appropriate main form for the user, and use case ends

b.2 Alternative Flows

b.2.1 Invalid Name / Password

If in the basic flow the system cannot find the name or the password is invalid, an error message is displayed. The actor can type in a new name or password or choose to cancel the operation, at which point the use case ends.

c. Special Requirements

There are no special requirements associated with this use case.

d. Preconditions

No preconditions for this use case

e. Post-conditions

System should log the user’s entry (date-time, username) to the system in the log files

User must logout or system will automatically logout upon exit

f. Extension Points

There are no extension points associated with this use case.

6.1.2 Backup and Recovery Use case:

a. Brief Description

This use case allows the Administrator to manually backup and recover the system database.

The actor for this use case is the Administrator.

b. Flow of Events

The use case begins when the administrator selects “Backup and Recovery” option.

b.1. Basic Flow – Generate Backup

1. System displays a list of all system backups done so far
2. Administrator selects “new backup” option
3. System displays folder dialog requiring the destination path for the resulting backup file(s). the default path should be last backup path.

4. Administrator browse for the target backup folder, then selects “ok” option
5. System generates a backup file(s) for the differential database changes and save it in the destination folder and displays a confirmation message “backup files created successfully”
6. Administrator selects “ok”
7. System refreshes the backup list adding new backup detail (date, file name, file destination path).
8. use case ends

b.2. Alternative Flows

b.2.1 Recover System Database

1. The Administrator selects “Recovery” option
2. System displays a list of all backups done so far.
3. Administrator selects one backup from the list.
4. System validates that the backup files exist
5. System imports all data form files into the database and displays a confirmation message “Recovery done successfully”.
6. Administrator selects “ok”
7. use case ends

c. Special Requirements

No special requirements

d. Preconditions

Login.

e. Post-conditions

System should log all activities done to the system in the log files.

Administrator must logout or system will automatically logout upon exit

f. Extension Points

No extension points

6.1.3 Design Outputs Use Case:

a) Brief Description

This use case allows the Administrator to add, delete and modify system output documents.

The actor for this use case is the Administrator.

b) Flow of Events

The use case begins when the administrator selects “maintain outputs” option.

b.1. Basic Flow – Add new Output Document

1. System displays a list of all outputs
2. Administrator selects “new” option.
3. System displays a list of pre-set output templates that the administrator can choose from, one of which is the blank (empty) output document in design view.

4. Administrator adds basic output document information such as name, and description then changes the document's design, format and/or content data. Then he selects "save" option
5. System displays a print preview of the modified document, and a verification message "This document will be saved, ok cancel?"
6. Administrator selects "ok"
7. System saves the modified document, and closes the design view. And refreshes the output documents list.
8. Steps 1-7 are repeated if the administrator wants to modify several output documents.

b.2. Alternative Flows

b.2.1 Modify an Output Document

1. System displays a list of all outputs
2. Administrator selects one output from the list.
3. System displays the selected output document in the design view
4. Administrator changes the document's design, format and/or content data. Then he selects "save" option
5. System displays a print preview of the modified document, and a verification message "This document will be saved, ok cancel?"
6. Administrator selects "ok"
7. System saves the modified document, and closes the design view.
8. Steps 1-7 are repeated if the administrator wants to modify several output documents.

b.2.2 Delete an Output Document

1. System displays a list of all outputs
2. Administrator selects one output from the list.
3. System displays the selected output document in the design view
4. Administrator selects the "delete" option
5. System displays a print preview of the modified document, and a verification message "This document will be deleted, ok cancel?".
6. Administrator selects "ok".
7. System deleted the selected output document, closes the design view and refreshes the output documents list.
8. Steps 1-7 are repeated if the administrator wants to delete several output documents.

b.2.3 Preview an Output Document.

At any point of the above sub-flows after selecting certain output document; the administrator can select to preview the selected output document. The system will open that document in preview mode. And the administrator can choose either to close the preview mode or print it on printer.

c) Special Requirements

No special requirements

d) Preconditions

Login.

e) Post-conditions

System should log all activities done to the system in the log files.
Administrator must logout or system will automatically logout upon exit

f) Extension Points

No extension points

6.1.4 Feed Criminal Data Use Case:

a) Brief Description

This use case allows Criminal Data Feeder to add criminal information for certain citizen.

The actor of this use case is Criminal Data Feeder (CDF)

b) Flow of Events

The use case begins when the CDF selects “add criminal information” option.

b.1 Basic Flow – Add Final Judgment

1. Use case “Search and View Criminal Data” is included here
2. CDF chooses certain citizen from search result after making sure that it is the citizen he is looking for.
3. System opens empty criminal form record for that citizen. The form has two parts: “Judgment” part and “Status” part
4. CDF starts with “Judgment” part and selects “final judgment” from judgment type, then enters the following information:
 - select “court type”
 - court name
 - select Criminal Act Type (e.g. crime, felony or fine)
 - Criminal act
 - Date of judgment
 - Brief of judgment (text)
 - Minutes of judgment (scanned)
5. Then CDF starts entering the second part (Status). He enters the following information:
 - selects “convicted with enforcement of the Decision” or “convicted with no enforcement of decision” or “not convicted” from convicted type
 - if he selects “convicted with enforcement of the Decision”, then he enters the followings:
 - Date of judgment
 - Duration of judgment
 - Expected date of release
 - Detention facility center name.
 - if he selects “convicted with no enforcement”, then he enters the followings:
 - Duration of judgment.
 - if he selects “not convicted”, he does not enter any other information.
6. CDF then selects “add” option

7. System validates data entered, if data entered is proper, system adds the new Justice Record for that citizen and displays a confirmation message “Justice Record is added, ok”. Only if the CDF selected “not convicted” the system sets the final-status of that citizen to “clear”, otherwise the final-status is set to “judged”. Please refer to appendix E – state charts
8. CDF selects “ok”.
9. Use case ends.

b.2 Alternative Flows

b.2.1 Add preliminary Judgment

1. Use case “Search and View Criminal Data” is included here
2. CDF chooses certain citizen from search result after making sure that it is the citizen he is looking for.
3. System opens empty criminal form record for that citizen. The form has two parts: “Judgment” part and “Status” part
4. CDF starts with “Judgment” part and selects “preliminary judgment” from judgment type, then enters the following information:
 - Court type
 - Court name
 - case number
 - Application number
 - Action such as: travel prohibit
 - Brief of judgment
 - minutes of judgment (scanned)
5. CDF then selects “add” option
6. System validates data entered, if data entered is proper, system adds the new Justice Record for that citizen and displays a confirmation message “Justice Record is added, ok”. System sets the final-status of that citizen to “judged”
7. use case ends

c) Special Requirements

All transactions shall be logged in the system. Actor can make undo actions for the information entered.

d) Preconditions

Login.

Citizen record has to exist in the system

e) Post-conditions

New criminal information are not committed to the system until they are managed and accepted by the Data Authorization Personnel.

System should log all activities done to the system in the log files.

CDF must logout or system will automatically logout upon exit

f) Extension Points

No extension points.

6.1.5 Feed Security Info Use Case

a) Brief Description

This use case allows Security Data Feeder to modify security information for certain citizen.

The actor of this use case is Security Data Feeder (SDF)

b) Flow of Events

The use case begins when the SDF selects “maintain security information” option.

b.1 Basic Flow – Modify Security Information

1. Use case “Search and View Criminal Data” is included here
2. SDF chooses certain citizen from search result after making sure that it is the citizen he is looking for.
3. System opens Justice Record for that citizen in a read only view, and opens security information section in read/write view.
4. SDF can uncheck previous security flags, or add new security flags. For every flag he checks, he can add extra information as text. In both cases he should provide supporting documents (scanned).
5. SDF then selects “add” option
6. System validates data entered, if data entered is proper, system modifies the security information as entered. And shows a message “security information is changed, ok”
7. SDF selects “ok”.
8. Use case ends.

c) Special Requirements

Actor can make undo actions for the information entered.

d) Preconditions

Login.

e) Post-conditions

Security changes are not committed to the system until they are managed and accepted by the Data Authorization Personnel.

System should log all activities done to the system in the log files.

SDF must logout or system will automatically logout upon exit

f) Extension Points

No extension points.

6.1.6 Maintain Privileges Use Case:

a) Brief Description

This use case can extend the use case “Maintain System Security”.

This use case allows the Administrator to add, delete and modify system privileges. Please note that the “details” information is about data forms and system functions (such as design outputs, view auditing reports, etc) and their access rights viewed as a table, this can be negotiable, and the designer can suggest more insight.

What I intend to show in the two use cases (Maintain Security Roles and Maintain Privileges) is that the security permissions are dynamic, and we can change, create, modify them as we need.

The actor for this use case is the Administrator.

b) Flow of Events

The use case begins when the administrator selects “maintain privileges” option.

b.1 Basic Flow – Add Privilege

1. System displays a list of levels of system privileges with all associated privileges at for each level
2. Administrator selects “new privilege” option
3. System displays an empty privilege form
4. Administrator types in the new privilege information: privilege name, and description. Then selects “add” option
5. System validates data entry, generates a new ID for this privilege.
6. System displays a list of all privilege’s details
7. Administrator selects one or more details to associates them with this privilege, then selects “ok”
8. System associates the new details with this privilege closes the details form, and refreshes the detail form.
9. When the administrator is done with this privilege, he selects “add” option
10. System adds the new privilege, closes the privilege form and refreshes the privileges list.
11. steps 2-5 are repeated if the administrator wants to add several privileges

b.2 Alternative Flows

b.2.1 Modify a Privilege

1. The Administrator selects "modify Privilege" option
2. The system displays a blank privilege form.
3. The Administrator types in the privilege name he wishes to modify.
4. The system retrieves the privilege information and displays it on the screen.
5. The administrator modifies one or more of the privilege information: name, description. Then selects “details” option
6. System displays a list of all details associated with this privilege
7. Administrator selects/deselect one or more details then selects “ok”
8. System associates the new details, and closes the details form, and refreshes the details list
9. When the administrator is done modifying this privilege, he selects “ok” option
10. System modifies the new privilege, closes the privilege form and refreshes the privileges list.
11. Steps 1-10 are repeated for each privilege the Administrator wants to modify. When edits are complete, the use case ends.

b.2.2 Delete a Privilege

1. The Administrator selects "delete privilege" option
2. The system displays a blank privilege form.

3. The Administrator types in the privilege name for the privilege that's being deleted.
4. The system retrieves the privilege and displays the privilege information in the form.
5. The Administrator selects "delete."
6. The system displays a delete verification dialog confirming the deletion.
7. The Administrator selects "yes."
8. The privilege is deleted from the system.
9. Steps 2-8 are repeated for each privilege deleted from the system. When the Administrator is finished deleting privileges from the system the use case ends.

b.2.3 Privilege Already Exists

If in the "Add a privilege" sub-flow the system finds an existing privilege with the same name an error message is displayed "Privilege Already Exists". The Administrator can either change the name, or cancel the operation at which point the use case ends.

b.2.4 Privilege Not Found

If in the "Modify a Privilege" or "Delete a Privilege" sub-flows the privilege name is not located, the system displays an error message, "Privilege is Not Found". The Administrator can then type in a different privilege name or cancel the operation at which point the use case ends.

c) Special Requirements

Actor can make undo actions for the information entered.

d) Preconditions

Login.

e) Post conditions

System should log all activities done to the system in the log files.
Administrator must logout or system will automatically logout upon exit

f) Extension Points

No extension points

6.1.7 Maintain Security Roles Use Case:

a) Brief Description

This use case allows the Administrator to add, delete and modify system roles information and their privileges.

The actor for this use case is the Administrator.

b) Flow of Events

The use case begins when the administrator selects "maintain security" option.

b.1 Basic Flow – Add Role

1. System displays a list of all system roles
2. Administrator selects "new role" option

3. System displays an empty role form
4. Administrator types in the new role information: role name, and description. Then selects "add" option
5. System validates data entry, generates a new ID for this role, adds the new role.
6. System displays a list of privilege levels say three, each with its associated privileges
7. Administrator selects one or more privileges to associates them with this role, then selects "ok"
8. System associates the new privileges with this role closes the privileges form, and refreshes the privileges list.
9. When the administrator is done with this role, he selects "add" option
10. System adds the new role, closes the role form and refreshes the roles list.
11. steps 2-5 are repeated if the administrator wants to add several roles

b.2 Alternative Flows

b.2.1 Modify a Role

1. The Administrator selects "modify Role" option
2. The system displays a blank role form.
3. The Administrator types in the role name he wishes to modify.
4. The system retrieves the role information and displays it on the screen.
5. The administrator modifies one or more of the role information: name, description. Then selects "privileges" option
6. System displays a list of all privileges associated with this role
7. Administrator selects/deselect one or more privileges then selects "ok"
8. System associates the new privileges, and closes the privileges form
9. When the administrator is done modifying this role, he selects "ok" option
10. System adds the new role, closes the role form and refreshes the roles list.
11. Steps 1-10 are repeated for each role the Administrator wants to modify. When edits are complete, the use case ends.

b.2.2 Delete a Role

1. The Administrator selects "delete role" option
2. The system displays a blank role form.
3. The Administrator types in the role name for the role that's being deleted.
4. The system retrieves the role and displays the role information in the form.
5. The Administrator selects "delete."
6. The system displays a delete verification dialog confirming the deletion.
7. The Administrator selects "yes."
8. The role is deleted from the system.
9. Steps 2-8 are repeated for each role deleted from the system. When the Administrator is finished deleting roles from the system the use case ends.

b.2.3 Role Already Exists

If in the "Add a role" sub-flow the system finds an existing role with the same name an error message is displayed "Role Already Exists". The Administrator can either change the name, or cancel the operation at which point the use case ends.

b.2.4 Role Not Found

If in the "Modify a Role" or "Delete a Role" sub-flows the role name is not located, the system displays an error message, "Role is Not Found". The Administrator can then type in a different role name or cancel the operation at which point the use case ends.

c) Special Requirements

Actor can make undo actions for the information entered.

d) Preconditions

Login.

e) Post-conditions

System should log all activities done to the system in the log files.
Administrator must logout or system will automatically logout upon exit

f) Extension Points

This use case can be extended at any point by the "Maintain Privileges Use Case"

6.1.8 Maintain Users Use Case:

a) Brief Description

This use case allows the Administrator to add, delete and modify system users' information. It also allows the administrator to change the users' privileges of security. This use case can be extended by Manage Authentication Requests use case as well.

The actor for this use case is the Administrator.

b) Flow of Events

The use case begins when the administrator selects "maintain users" option.

b.1 Basic Flow – Add User

1. System displays a list of all system users
2. Administrator selects "new user" option
3. System displays an empty user form
4. Administrator types in the new user information: user full name, system username, password, email, and selects user's role. Then selects "add" option
5. system validates data entry, generates a new ID for this user, adds the new user, sends an email for that user with his new account information, closes the user form and refreshes the users list,
6. System sends the new account information (username, password, etc) to the email of the new user.
7. steps 2-5 are repeated if the administrator wants to add several users

b.2 Alternative Flows

b.2.1 Modify a User

1. The Administrator selects "modify User" option
2. The system displays a blank user form.
3. The Administrator types in the user name he wishes to modify.
4. The system retrieves the user information and displays it on the screen.
5. The administrator modifies one or more of the users' information fields: name, user name, and his role.

6. When changes are complete, the Administrator selects "save."
7. The system updates the user information.
8. Steps 2-7 are repeated for each user the Administrator wants to modify. When edits are complete, the use case ends.

b.2.2 Delete a User

1. The Administrator selects "delete user."
2. The system displays a blank user form.
3. The Administrator types in the user name for the user that's being deleted.
4. The system retrieves the user and displays the user information in the form.
5. The Administrator selects "delete."
6. The system displays a delete verification dialog confirming the deletion.
7. The Administrator selects "yes."
8. The user is deleted from the system.
9. Steps 2-8 are repeated for each user deleted from the system. When the Administrator is finished deleting users from the system the use case ends.

b.2.3 User Already Exists

If in the "Add a user" sub-flow the system finds an existing user with the same name an error message is displayed "User Already Exists". The Administrator can either change the name, or cancel the operation at which point the use case ends.

b.2.4 User Not Found

If in the "Modify a User" or "Delete a User" sub-flows the user name is not located, the system displays an error message, "User Not Found". The Administrator can then type in a different id number or cancel the operation at which point the use case ends.

c) Special Requirements

User's passwords always appear in their encrypted form. The administrator can never see the real password. Actor can make undo actions for the information entered.

d) Preconditions

Login.

e) Post-conditions

System should log all activities done to the system in the log files. And the password should be saved in encrypted format.

Administrator must logout or system will automatically logout upon exit

f) Extension Points

There are no extension points associated with this use case.

6.1.9 Manage Application Requests Use Case:

a) Brief Description

This use case allows the Formal Document Producer to manage requests sent by clients to view certain Justice Record. They can view later on their Justice Record via email, fax or formal document (hard copy).

The actor for this use case is Formal Document Producer (FDP).

b) Flow of Events

The use case begins when the FDP selects “manage requests” option.

b.1 Basic Flow – Accept Request

1. System displays a list of all unread requests sorted by date (an option for sorting by other keys such as request source must also be available)
2. FDP selects one request form the list
3. System opens request information and displays it on screen
4. FDP opens supporting documents
5. If the FDP is convinced with the information in the supporting documents, he selects “accept” option. Produce Formal Document use case is included here.
6. After FDP chooses certain output document, system closes request form, changes the request status into “accepted” and refreshes the requests list.
7. Steps 2-10 are repeated to manage several requests

b.2 Alternative Flows

b.2.1 Reject Request

1. System displays a list of all “unread” requests sorted by date
2. FDP selects one request form the list
3. System opens request information and displays it on screen
4. FDP opens supporting documents
5. FDP finds that some supporting information is missing, or incomplete, he selects “reject” option
6. System opens message form
7. FDP types why he rejected the request, then selects “send”
8. System sends the message content to the client’s email
9. System closes the request form, changes the request status into “rejected” and refreshes the requests list.
10. System sends an email to the client to inform him that his/her request was rejected.

c) Special Requirements

In the basic flow after step 5, use case Produce Formal Documents is included. All transactions should be logged by the system.

d) Preconditions

Login.

e) Postconditions

System should log all activities done to the system in the log files.
FDP must logout or system will automatically logout upon exit

f) Extension Points

No extension point.

6.1.10 Manage Authentication Requests Use Case:

a) *Brief Description*

This use case allows the Administrator to manage authentication requests sent by Registration Authorization Personnel.

The actor for this use case is Administrator

b) *Flow of Events*

The use case begins when the Administrator selects “manage authentication requests” option.

b.1 Basic Flow – Accept Request

1. System displays a list of all unread authentication requests sorted by date (other sort keys must also be available including request source)
2. Administrator selects one request form the list
3. System opens request information and displays it on screen
4. Maintain Users use case can be extended here
5. When the Administrator is finished with this request (Accepted and generated user) he selects “ok”
6. System closes request form, changes its status from “unread” into “accepted” and refreshes the requests list.

b.2. Alternative Flows

No alternatives so far.

c) *Special Requirements*

No special requirements

d) *Preconditions*

Login.

e) *Postconditions*

System should log all activities done to the system in the log files.

Administrator must logout or system will automatically logout upon exit

f) *Extension Points*

Maintain Users use case can be extended at step 4 in the basic flow.

6.1.11 Manage Data Feeding Requests:

a) *Brief Description*

This use case allows the Data Authorization Personnel to manage criminal data additions and updates.

The actor for this use case is Data Authorization Personnel (DAP).

b) *Flow of Events*

The use case begins when the DAP selects “manage data feeding requests” option.

b.1 Basic Flow – Accept Addition or modification request

1. System displays a list of all unread requests sorted by date (other sort keys such as request source must be included)
2. DAP selects one request form the list
3. System opens request information and displays it on screen
4. DAP opens supporting documents
5. If DAP is convinced with supporting document, he selects “accept” option.
6. System closes request form and changes its status from “unread” into “accepted”. And refreshes the requests list. So the changes or addition made by Criminal Data Feeder is now committed to the system.
7. Steps 2-7 are repeated to manage several requests

b.2 Alternative Flows

b.2.1 Reject Request

1. System displays a list of all “unread” requests sorted by date
2. DAP selects one request form the list
3. System opens request information and displays it on screen
4. DAP opens supporting documents
5. RAP finds that some supporting information is missing, or incomplete, he selects “reject” option
6. System opens message form
7. DAP types in the reason why he rejected the request, or what is missing or wrong in the request. Then selects “ok”
8. System closes message form closes the request form, changes the request status into “rejected”, and refreshes the requests list.
9. System sends the content of the message form as an email to the client to inform him that his/her request was rejected plus the reasons.

c) Special Requirements

No special requirements

d) Preconditions

Login.

e) Post-conditions

System should log all activities done to the system in the log files.

DAP must logout or system will automatically logout upon exit

f) Extension Points

Post Authentication Request use case can be extended at step 5 in the basic flow.

6.1.12 Manage Registration Requests Use Case:

a) Brief Description

This use case allows the Registration Authorization Personnel to manage registration requests sent by clients to get authorized access to the system.

The actor for this use case is Registration Authorization Personnel (RAP).

b) Flow of Events

The use case begins when the RAP selects “manage registrations” option.

b.1 Basic Flow – Accept Request

1. System displays a list of all unread registration requests sorted by date (other sort keys such as request source must be included)
2. RAP selects one request form the list
3. System opens request information and displays it on screen
4. RAP opens supporting documents
5. If RAP is convinced with supporting document, the RAP selects “accept” option. Post Authentication Request use case can be extended here.
6. After RAP is finished with this request, he/she selects “close” option
7. System closes registration request form and changes its status from “unread” into “accepted”. And refreshes the requests list.
8. Steps 2-7 are repeated to manage several requests

b.2 Alternative Flows

b.2.1 Reject Registration Request

1. System displays a list of all “unread” requests sorted by date
2. RAP selects one request form the list
3. System opens request information and displays it on screen
4. RAP opens supporting documents
5. RAP finds that some supporting information is missing, or incomplete, he selects “reject” option
6. System opens message form
7. RAP types in the reason why he rejected the request, or what is missing or wrong in the request. Then selects “ok”
8. System closes message form closes the request form, changes the request status into “rejected”, and refreshes the requests list.
9. System sends the content of the message form as an email to the client to inform him that his/her request was rejected plus the reasons.

c) Special Requirements

No special requirements

d) Preconditions

Login.

e) Post-conditions

System should log all activities done to the system in the log files.
RAP must logout or system will automatically logout upon exit

f) Extension Points

Post Authentication Request use case can be extended at step 5 in the basic flow.

6.1.13 Modify Criminal Data Use Case

a) Brief Description

This use case allows Criminal Data Feeder to modify criminal information for certain citizen. CDF can only modify criminal data entered previously by him.

The actor of this use case is Criminal Data Feeder (CDF)

b) Flow of Events

The use case begins when the CDF selects “modify criminal information” option.

b.1 Basic Flow – Add Final Judgment

1. Use case “Search and View Criminal Data” is included here
2. CDF chooses certain citizen from search result after making sure that it is the citizen he is looking for.
3. System displays a list of Justice Records entered by current actor for that citizen
4. CDF chooses one Justice Record.
5. System opens selected Justice Record in edit view
6. CDF can update the data. (for the data structure, please refer to use case Feed Criminal Data). He also adds supporting documents (scanned)
7. CDF then selects “save” option
8. System validates data entered, if data entered is proper, system updates the new Justice Record for that citizen and displays a confirmation message “Justice Record is updated, ok”. System updates final-status for that citizen according to the same rules as in use case “Feed Criminal Data”
9. CDF selects “ok”.
10. Use case ends.

b.2 Alternative Flows

b.2.1 No Justice Records

If in the basic flow at step 3, if there is no Justice Records for that citizen, or current system user did not previously enter Justice Records for that citizen; then the system will show a message “no Justice Records found, ok”. The user can then refine his/her search.

c) Special Requirements

All transactions shall be logged in the system. Actor can make undo actions for the information entered.

d) Preconditions

Login.

e) Post-conditions

Modifications to the criminal data are not committed to the system until they are managed and accepted by the Data Authorization Personnel.

System should log all activities done to the system in the log files.

User must logout or system will automatically logout upon exit

f) Extension Points

No extension points.

6.1.14 Post Application Request Use Case:

a) Brief Description

This use case allows the citizen or any organization to post request to only view certain Justice Record. They have to post a request every time they need to view certain Justice Record.

The actor for this use case is the Criminal Data Viewer (CDV).

b) Flow of Events

The use case begins when the CDV selects “Post Application Request” option.

b.1 Basic Flow – Post Application Request

1. System displays an empty application form
2. CDV types in some of the citizen information: name according to ID, ID number, name according to passport, passport number, sex, city, marital status, mother name, place of birth, Date of Birth, Religion, former personal name, email, fax. If the citizen is posting a request for another citizen, he/she should scan and attach power of attorney to the request.
3. CDV selects the format of the requested information (email, fax, or hard copy). If he chooses “hard copy”, he should also state from where he will collect it.
4. Then selects “post” option
5. System validates data entry, if data entered is ok, system displays a confirmation message “your request will be followed”
6. Use case ends

b.2 Alternative Flows

b.2.1 Data not valid, or missing

If in the basic flow the CDV did not provide all data, or some of the data is missing in the application, system displays a message showing which data is missing or wrong. CDV re-enters the missing or wrong data again.

c) Special Requirements

Actor can make undo actions for the information entered.

d) Preconditions

Login.

e) Post-conditions

CDV must logout or system will automatically logout upon exit

f) Extension Points

There are no extension points associated with this use case.

6.1.15 Post Authentication Requests Use Case:

a) Brief Description

This use case is extended by Manage Registration Requests use case. This use case allows the Registration Authorization Personnel (RAP) to post a request to the Administrator asking him to create system users with certain privileges to feed and/or modify and/or view criminal info.

The actor for this use case is the Registration Authorization Personnel (RAP).

b) Flow of Events

The use case begins when the RAP selects “Post Authentication Request” option.

b.1 Basic Flow – Post Application Request

1. System displays a request form automatically containing the email of the original client of the request
2. RAP selects one or more of the needed functionality: feed data, modify data, search and view data. Then selects “ok”
3. System displays a confirmation message “your request will be followed”
4. Use case ends

b.2 Alternative Flows

No alternatives so far

c) Special Requirements

Actor can make undo actions for the information entered.

d) Preconditions

Login.

This use case is extended by Manage Registration Requests use case

RAP must logout or system will automatically logout upon exit

e) Post-conditions

RAP must logout or system will automatically logout upon exit

f) Extension Points

There are no extension points associated with this use case.

6.1.16 Produce Formal Documents Use Case:

a) Brief Description

This use case allows the Formal Document Producer to produce formal document such as “good of conduct, ” to certain citizen.

The actor for this use case is Formal Document Producer (FDP).

b) Flow of Events

The use case begins in the basic flow of use case Manage Requests after step 5..

b.1. Basic Flow – Produce Formal Document

1. System displays a list of all formal documents
2. FDP chooses one or more document form the list

3. System displays a list of sending choices (e.g. email, fax, printer)
4. FDP chooses one or more of sending choices
5. System closes the lists, and sends the documents as requested (by email, fax or printed on printer).

c) Special Requirements

No special requirements

d) Preconditions

Login.

e) Post-conditions

System should log all activities done to the system in the log files.
FDP must logout or system will automatically logout upon exit

f) Extension Points

No extension points for this use case

6.1.17 Register Use Case:

a) Brief Description

This use case allows Criminal Data feeder and Officials to register to our system so that they can access have regular access to the system so that they can feed and/or search and view criminal information.

The actor for this use case is the Criminal Data feeder (CDF), Detention Facility Officer, Security Data Feeder.

b) Flow of Events

The use case begins when the CDF selects “Registration” option.

b.1 Basic Flow – Post Registration Request

1. System displays an empty registration form
2. CDF enters the following information: *please refer to attached document named: Registration form*
3. CDF can also attach documents. Then selects “ok” option
4. System validates data entered, and if it is ok, system displays confirmation message “your registration request will be followed”
5. CDF selects “ok”.
6. use case ends.

b.2 Alternative Flows

b.2.1 Incorrect data input.

If in the above basic flow the CDF entered invalid data, the system shows an error message “incorrect data entered” and shows which fields are wrong. The user can change these data and re submits the form.

c) Special Requirements

Actor can make undo actions for the information entered.

d) Preconditions

Login.

e) Post-conditions

System should log all activities done to the system in the log files.
CDF must logout or system will automatically logout upon exit

f) Extension Points

There are no extension points associated with this use case.

6.1.18 Search and View Criminal Data Use Case:

a) Brief Description

This use case allows the Formal Document Producer and Criminal Data Feeder to search and view citizen's records and criminal information.

The actor for this use case is Formal Document Producer (FDP). And Criminal Data Feeder (CDF)

b) Flow of Events

This use case begins when the FDP or CDF chooses "Search Citizens" options. Only the FDP can extend the use case "Produce Formal Documents".

b.1 Basic Flow – Search and Find Citizen

1. System displays an empty civilian form
2. FDP types in some of the citizen information: name according to ID, ID number, name according to passport, passport number, sex, city, marital status, mother name, place of birth, Date of Birth, Religion, former personal name, . Then selects "search" option
3. System retrieves citizen information and displays it on screen along with all of his/her criminal data in a Read Only view. Extension point: Produce Formal Documents use case

b.2. Alternative Flows

b.2.1 Citizen Not Found

If in the basic flow the citizen is not located, the system displays an error message, "Citizen Not Found". The FDP can then type in a different id number or cancel the operation at which point the use case ends

c) Special Requirements

No special requirements

d) Preconditions

Login.

e) Post-conditions

No post conditions..
FDP or CDF must logout or system will automatically logout upon exit

f) Extension Points

In the basic flow, after step 3, use case Produce Formal Documents can be extended only if actor was FDP.

6.1.19 Update Basic Data Use Case:

a) Brief Description

This use case allows the Registration Authorization Personnel to update basic citizen data. Allowing him to add new information that was not present basic data was imported from MOI database.

The actor for this use case is the Registration Authorization Personnel (RAP).

b) Flow of Events

The use case begins when the RAP selects “update citizen information” option.

b.1 Basic Flow – Update citizen record

1. RAP selects “update civilian information” option
2. System displays an empty civilian form
3. RAP types in some of the citizen information: name according to ID, ID number, name according to passport, passport number, gender, city, marital status, mother name, place of birth, Date of Birth, Religion, former personal name, . Then selects "search" option.
4. System retrieves citizen information and displays it on screen.
5. RAP modifies current data, or adds new information such as photo, finger print, etc. then selects “save” option.
6. System updates citizen record.
7. Use case ends

b.2 Alternative Flows

2.2.1 Citizen Not Found

If in the basic flow the citizen is not located, the system displays an error message, "Citizen Not Found". The RAP can then type in a different id number or cancel the operation at which point the use case ends.

c) Special Requirements

Actor can make undo actions for the information entered.

d) Preconditions

Login.

e) Post-conditions

System should log all activities done to the system in the log files.
RAP must logout or system will automatically logout upon exit

f) Extension Points

There are no extension points associated with this use case.

6.1.20 Update Detention Information Use Case:

a) Brief Description

This use case allows the Detention Facility officer to update the detention information for the criminals that are or were held in that facility and in that facility only.

The actor of this use case is Detention Facility Officer (DFO).

b) Flow of Events

The use case begins when the CDF selects “update detention information” option.

b.1 Basic Flow – Update Detention for certain criminal

1. System displays a list of all criminals currently held in that facility (the detention facility name is entered previously when entering “Final Judgment” section in use case “Feed Criminal Data”).
2. DFO selects “update” option
3. system displays empty search form
4. DFO enters some or all of following: name, ID number, date of birth, place of birth.
5. System retrieves that criminal information and displays only the detention part on screen.
6. DFO enter or modifies the followings:
 - Date of judgment efficiency
 - Date of actual release.

b.2 Alternative Flows

2.2.1 Criminal not found

If in the basic flow at step 5, if no criminal was found, system will display a message “criminal not found, ok”. DFO can then refine his search criteria.

c) Special Requirements

All transactions shall be logged in the system.

Actor can make undo actions for the information entered.

d) Preconditions

Login.

e) Post-conditions

System should log all activities done to the system in the log files.

DFO must logout or system will automatically logout upon exit

f) Extension Points

No extension points.

6.1.21 View Auditing Reports Use Case:

a) Brief Description

This use case allows the Data Authorization Personnel to view auditing reports to monitor activities done on the system.

The actor for this use case is Data Authorization Personnel (DAP).

b) Flow of Events

The use case begins when the DAP selects “System Auditing” option.

b.1 Basic Flow – View Auditing Report on Specific citizen record

1. System displays a list of reports options (e.g. all transactions between dates, transactions on specific citizen, transactions done by specific user, transactions done by specific role.)
2. DAP selects “transactions done on specific citizen”
3. System displays empty citizen form
4. DAP types some of the citizen basic information, then selects “ok”
5. System displays a report of all transactions done on this citizen record
6. Use case ends

b.2 Alternative Flows

b.2.1 View Auditing Report on Specific System User

1. System displays a list of reports options (e.g. all transactions between dates, transactions on specific citizen, transactions done by specific user, transactions done by specific role.)
2. DAP chooses “transactions done by specific user”
3. System opens an empty system-user form
4. DAP types some of the system user information such as user name, role
5. System retrieves system-user
6. DAP selects “ok”
7. System displays a report of all transactions done by this system user
8. use case ends

b.2.2 Record not found

If in the basic flow or alternative flow the citizen or system-user are not located, the system displays an error message, "Citizen/System User Not Found". The DAP can then type in a different search criteria's.

c) Special Requirements

No special requirements

d) Preconditions

Login.

e) Post-conditions

DAP must logout or system will automatically logout upon exit

f) Extension Points

There are no extension points associated with this use case.

6.2 SUPPLEMENTARY REQUIREMENTS

This section defines the quality ranges for performance, robustness, fault tolerance, usability, and similar characteristics for the Justice Record Automation System

6.2.1. System Availability

The System shall be available 24 hours a day, 7 days a week.

6.2.2. System Usability

- The System shall be easy-to-use and shall be appropriate for the target market of computer-literate and normal users.
- The System shall include online help for the user. Users should not require the use of a hardcopy Manual to use the System.
- Internet Browser Compliance: System user interface through Internet shall be compliant with MS IE 6.0, 7.0 and Mozilla Firefox.

6.2.3. System Maintainability

The System shall be designed for ease of maintenance:

- Tracking: All system errors shall be logged. Fatal system errors shall result in an orderly shutdown of the system.
- Tracking: The system error messages shall include a text description of the error, the operating system error code (if applicable); the module detecting the error condition, a data stamp, and a time stamp. All system errors shall be retained in the Error Log Database.
- Modular Design: Modular design means maintenance and tracking at module level. Each module's inputs and outputs must be pre-defined in the design phase.

6.2.4. Performance

- Simultaneous Users: The system shall support up to 1000 simultaneous users against the central database at any given time and up to 400 simultaneous users against the local servers at any one time.
- Transaction Response Time: The system must be able to complete 80% of all transactions within 2 minutes

6.2.5. Expandability

- The system must be expandable and flexible in both design and implementation. The system must allow dynamic form creation in run mode and must be able to accept additional processes drill-down when needed.

6.2.6. System Scope (*horizontal integration*)

- The scope of the system will initially cover entry and authentication requirements of the Public Prosecution (PP) and Ministry of Interior (MOI). The system must be built with an e-Government application scope in-mind meaning that when integrated (if possible) in the future with the MOI citizen database and when needs of other government institution arise (in the future) to access and feed information of the system, it will allow for it.

6.2.7. System Scalability (*vertical integration*)

- The system must be able to scale up to higher levels of user size and information flow and larger database to cover the needs of the citizens for at least up to ten years from the date of its inception.

Appendix A – Glossary

Justice Record Automation System Glossary

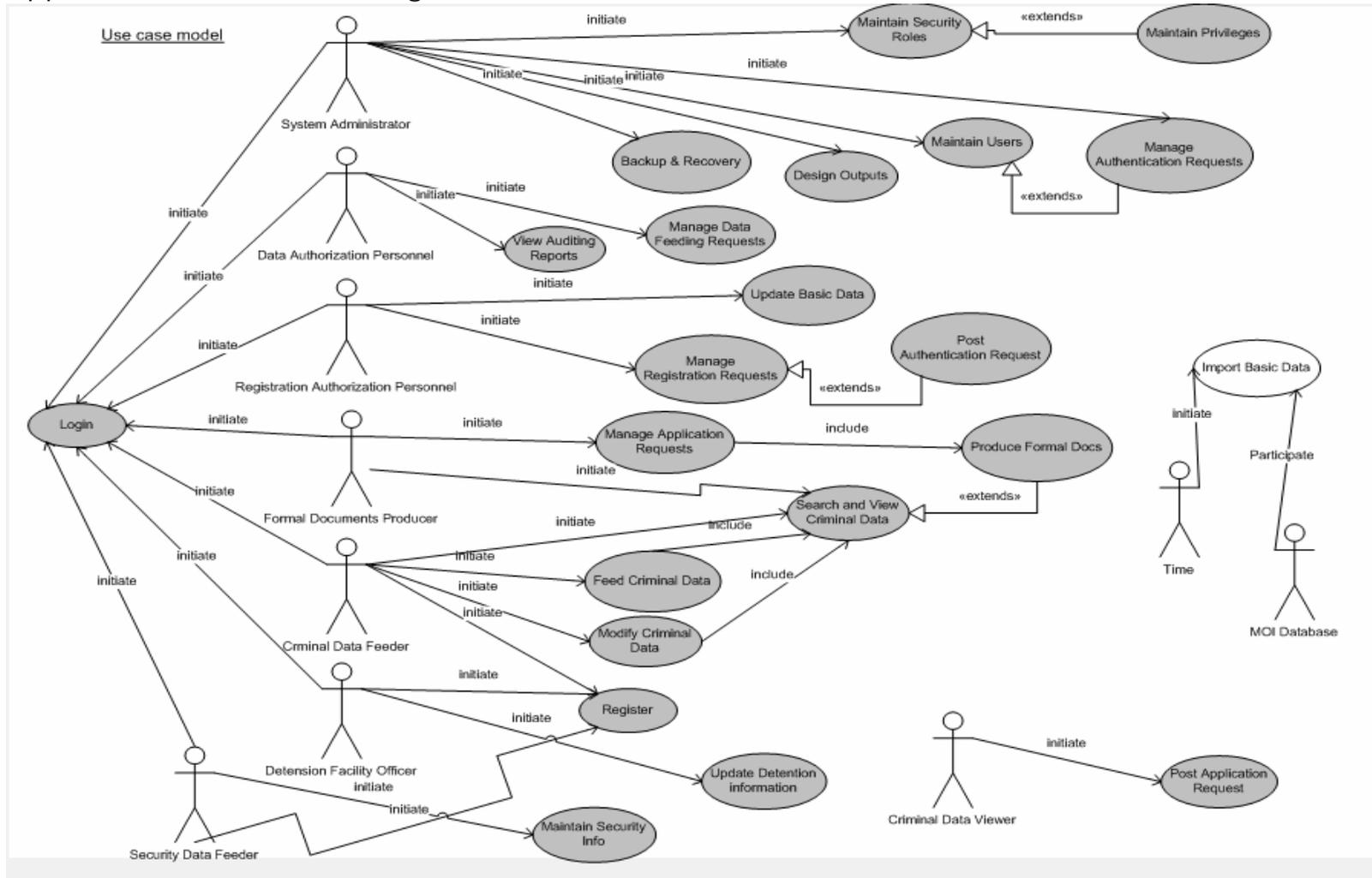
1. Introduction

The glossary contains the working definitions for all classes in the Justice Record Automation System. This glossary will be expanded throughout the life of the project.

2. Definitions

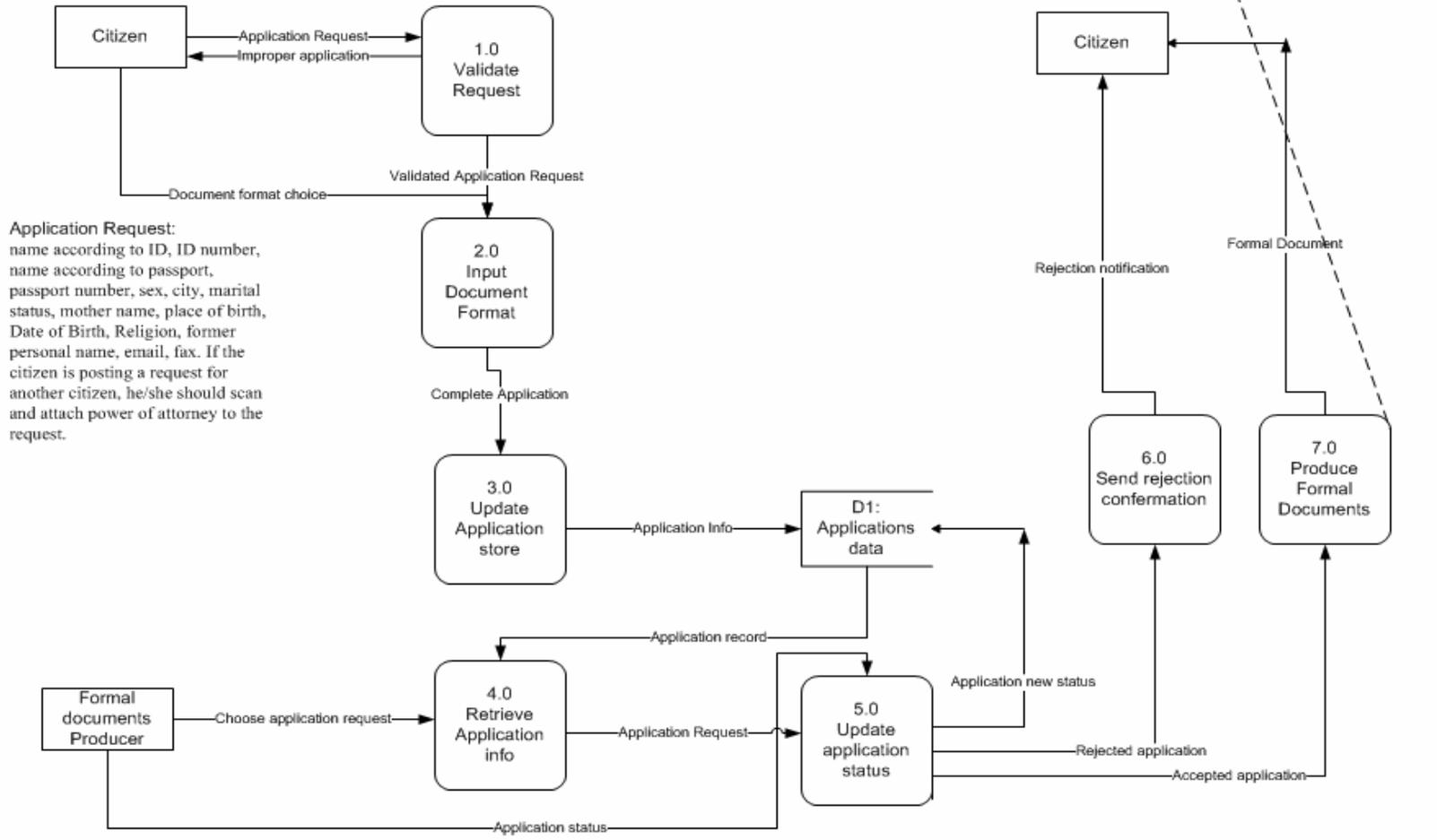
Class	Arabic definition	Description
MOJ	وزارة العدل	Ministry of Justice
MOI	وزارة الداخلية	Ministry of the Interior
PNA	السلطة الوطنية الفلسطينية	Palestinian Authority
Public Prosecution	النيابة العامة	
Conciliation Court	محكمة الصلح	
First Instance Court	محكمة البداية	
Religious Court	المحاكم الدينية	
Bureau of Personnel	ديوان الموظفين	
Borders Administration	إدارة المعابر و الحدود	
Traffic and Licensing Department	دوائر السير	
Monetary Authority	سلطة النقد	
Convicted with enforcement of the decision	محكوم مع حبس فعلي	
Clear	خالى من اية احكام جرمية	
Non-conviction	عدم محكومية	
Final judgment	حكم نهائي، فاصل	
Permeably Judgment	حكم ابتدائي	
Convicted with no enforcement of the decision	محكوم مع موقف النفاذ	

Appendix B – Use Case Diagram

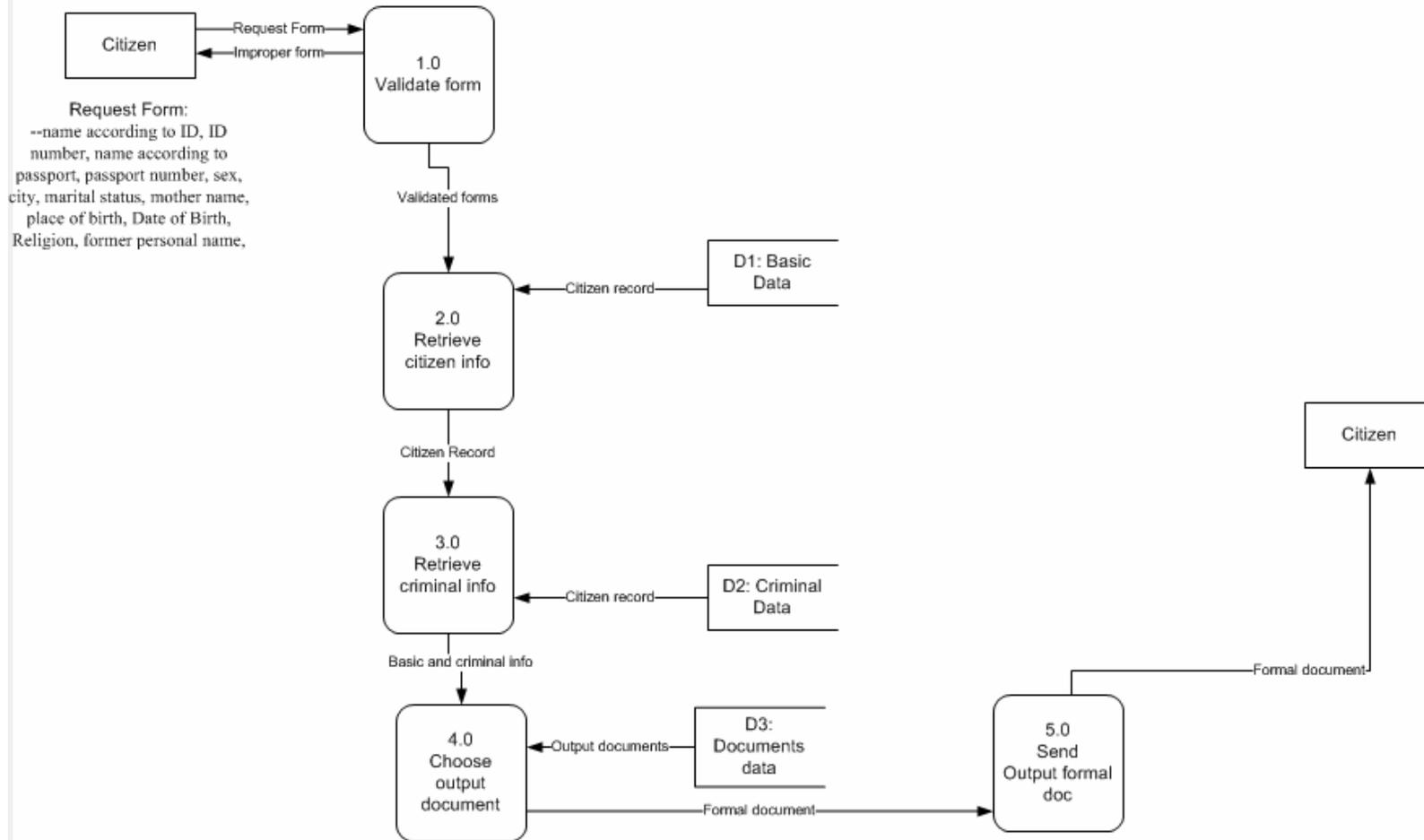


Appendix C – Data Flow Diagrams

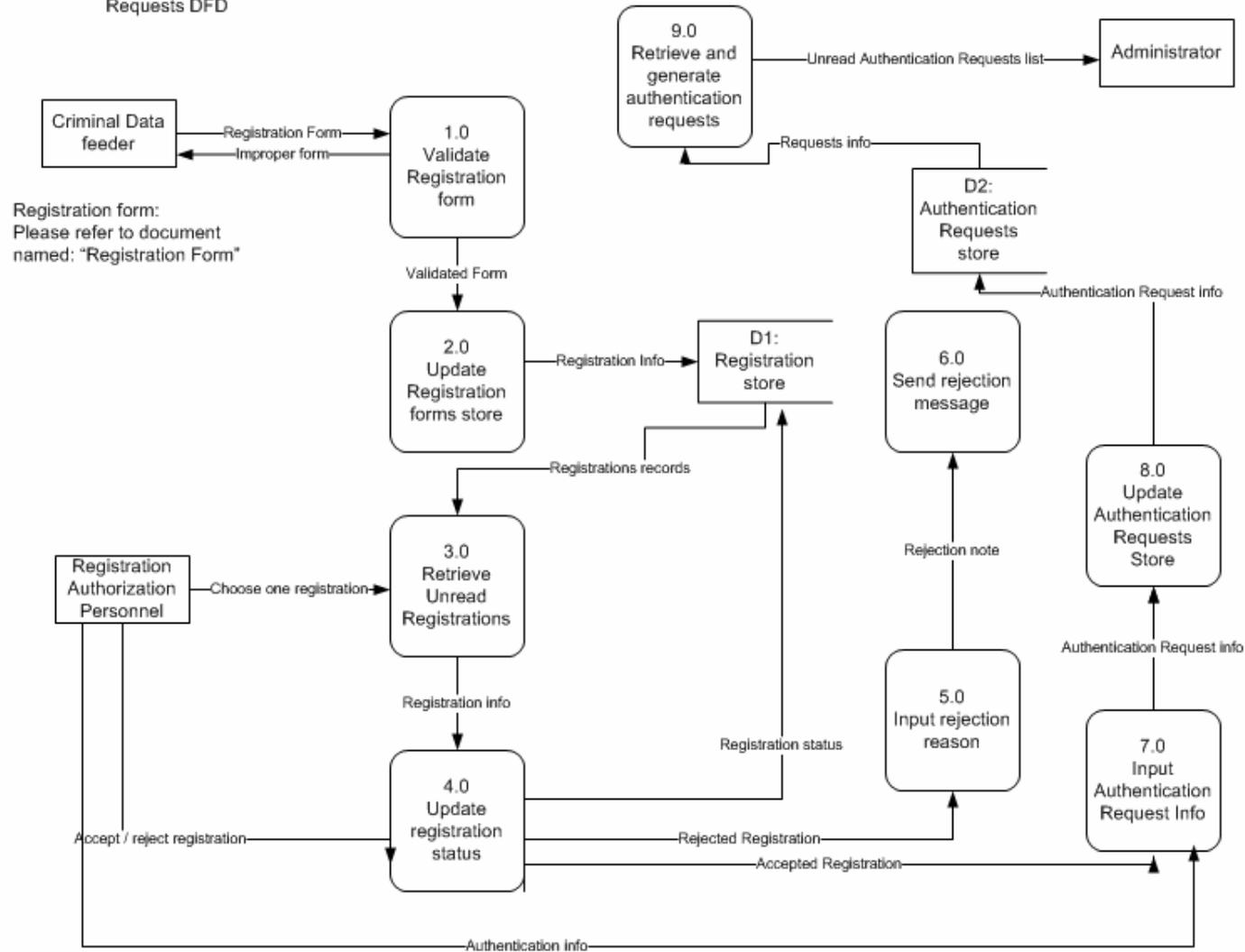
Post and Manage Application Requests-DFD

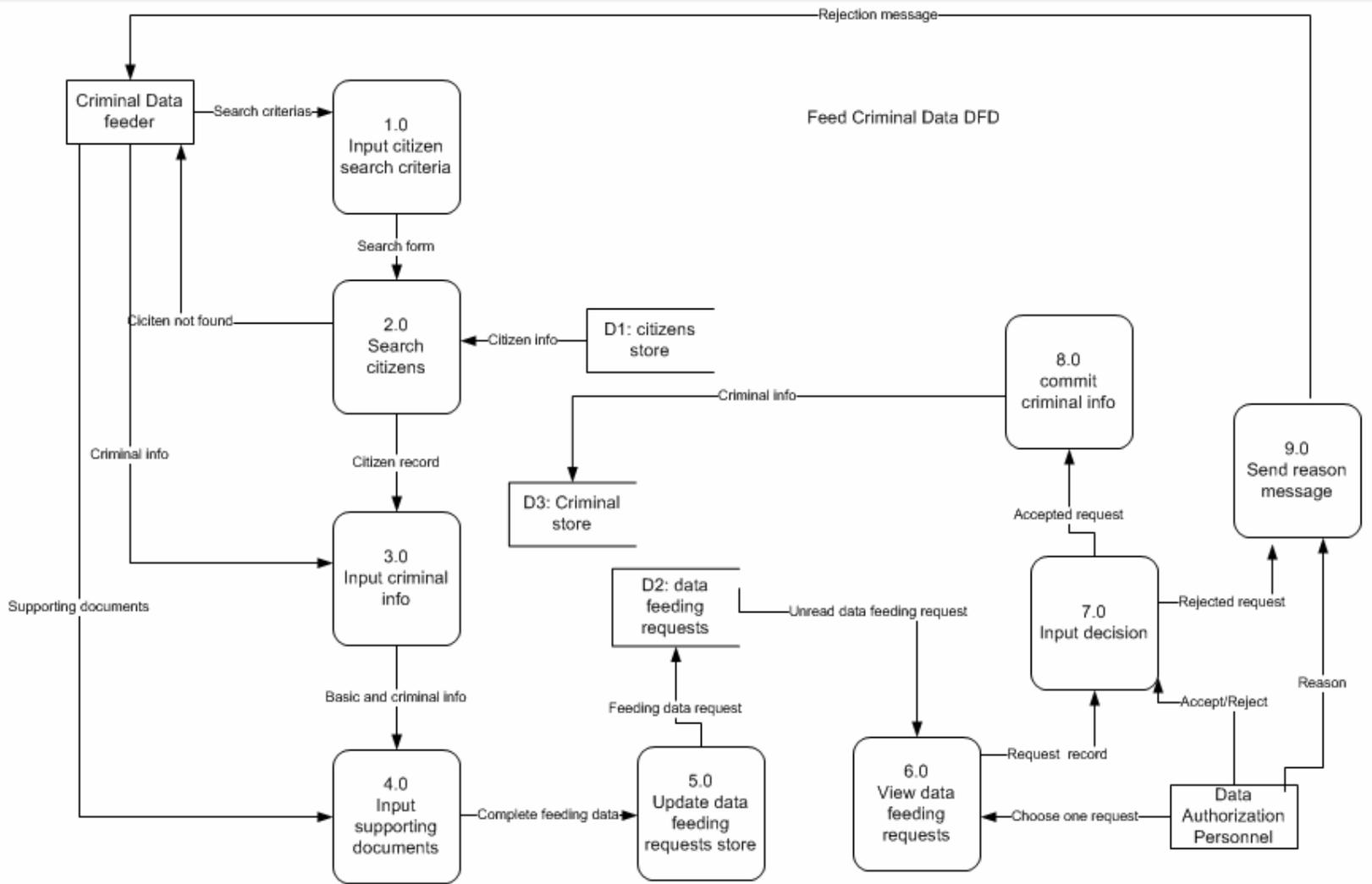


Produce Formal Documents DFD



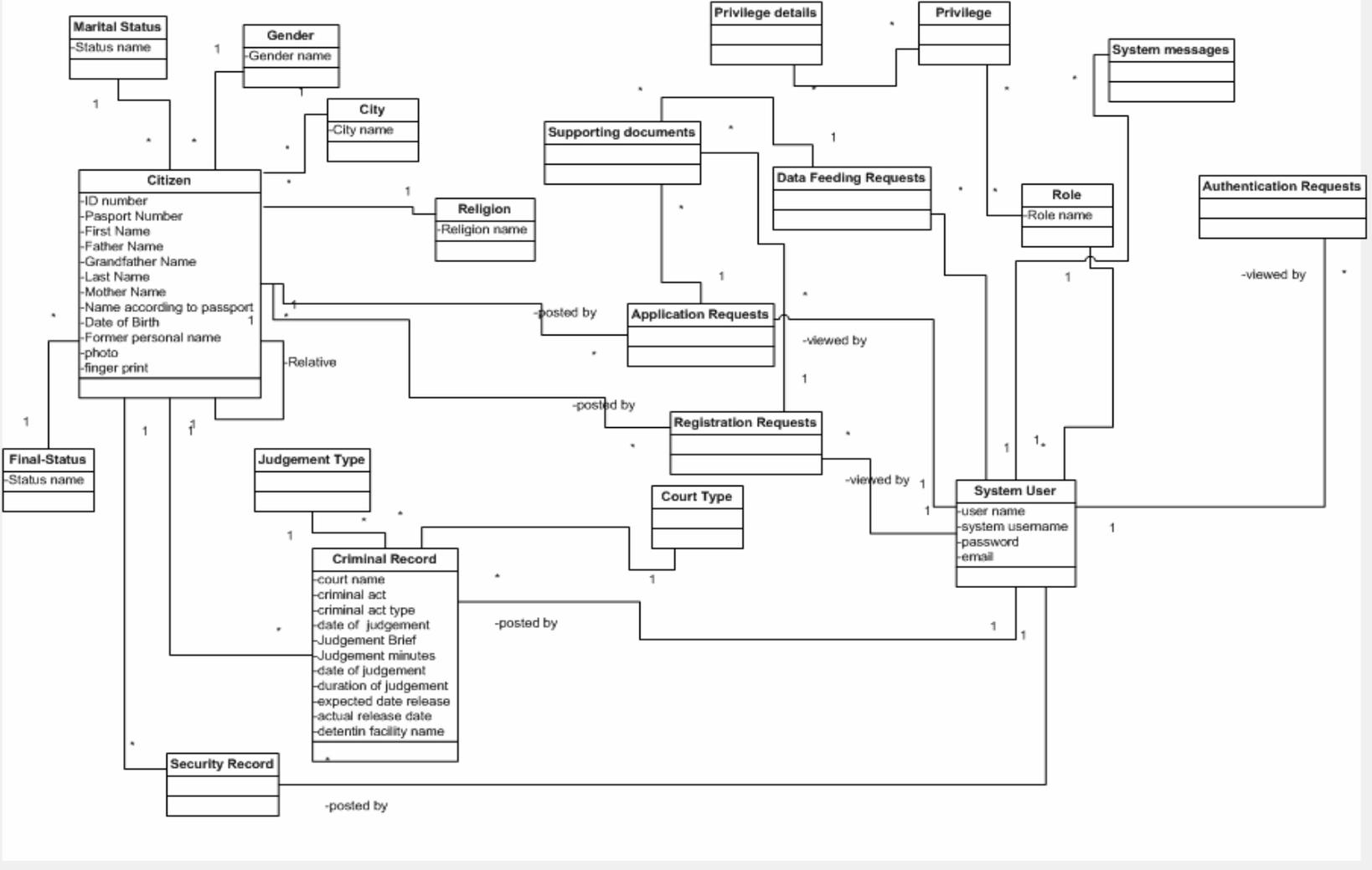
Post and Manage Registration Requests DFD



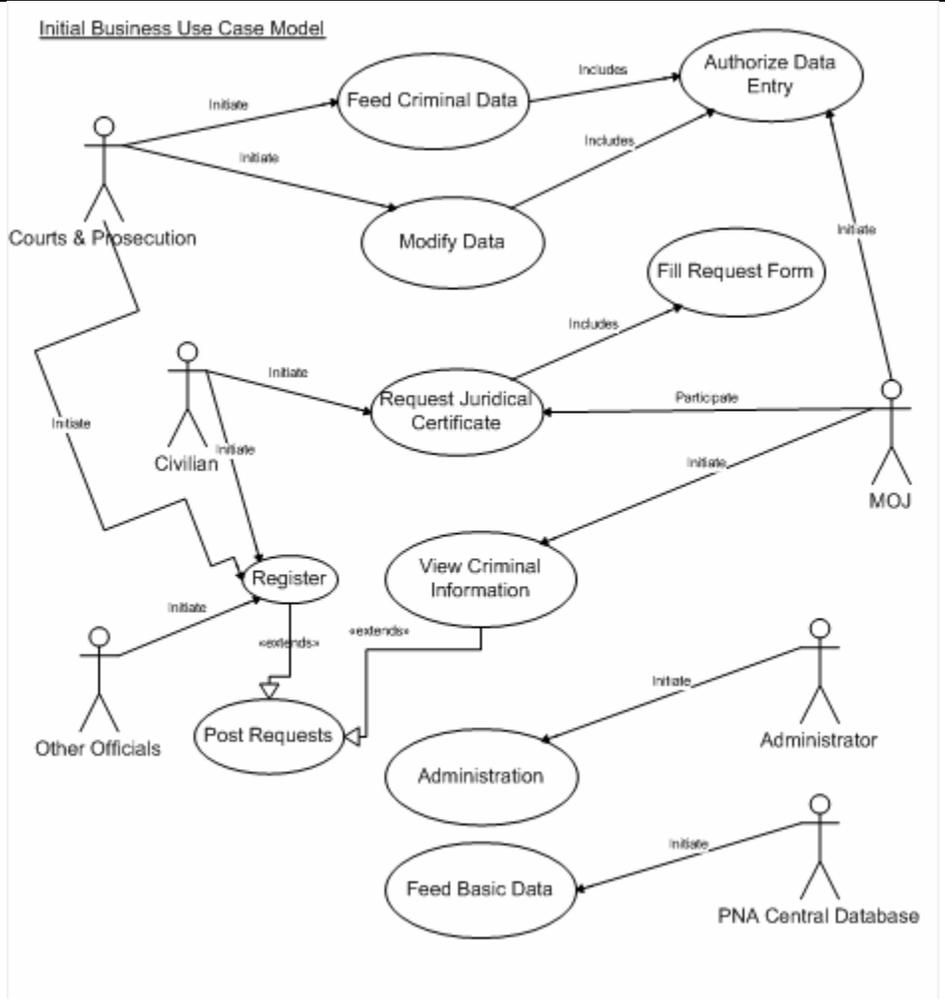


Appendix D – Conceptual ERD

Conceptual ERD



Appendix F – Business Model



Appendix G- Registration Form Model

نموذج طلب تسجيل على نظام السجل العدلي

- اسم الجهة طالبة التسجيل :
- طبيعة الإستخدام :- استخدام رسمي (مراقبة معابر وحدود، التدقيق الأمني، تغذية بيانات قضائية، تعديل بيانات قضائية، تعديل بيانات أمنية، تعديل بيانات الحجز)
-أخرى (لتوظيف في القطاع الخاص، حصول على بيانات مدفوعة الرسوم، شخصي)
(فصل)_____
- الغاية من التسجيل: الإطلاع ، تغذية معلومات
- البيانات عن الجهة طالبة التسجيل:
- بيانات شخصية : رقم الهوية، الإسم وفقاً للهوية، الإسم وفقاً لجواز السفر، تاريخ الميلاد، مكان الإقامة، البريد الإلكتروني، فاكس
- طبيعة المؤسسة (قطاع عام ، قطاع خاص، منظمة غير حكومية، هيئة دبلوماسية
في البلاد أو في الخارج، نيابة، أجهزة أمنية، شرطة)
- الموقع الجغرافي للمركز والفروع
- عدد المستخدمين المتوقعين وبياناتهم الشخصية ومراكزهم وصلاحياتهم والأقسام التابعين لها وفي أي موقع.
- نسخة عن الإذن الرسمي (قرار بالسماح بالتسجيل والإستخدام ويشمل ذلك الشركات الخاصة)
- نسخة عن قرار الصلاحيات الممنوحة للمستخدمين المتوقعين لدى الجهة طالبة التسجيل
- طبيعة المخرجات المطلوبة: (وثائق رسمية ورقية، ووثائق رسمية إلكترونية، بيانات إلكترونية، صفحة إدخال بيانات، الخ،----)
- الموقع المرجو استلام المخرجات فيه (قائمة حدد -----)