



Case Study : Prison Management System

SOFTWARE ENGINEERING

CECSC11

Department of Computer Engineering

Submitted By:

ASHISH KUMAR 2019UCO1518

SANDEEP JAIN 2019UCO1522

CHETAN RAGHAV 2019UCO1525

INDEX

<u>S. No.</u>	<u>Title</u>	<u>Pg. No.</u>
1	<u>Problem Statement</u>	2
2	<u>Use Case Diagram</u>	3
3	<u>Login</u>	4
4	<u>Maintain Items</u>	6
5	<u>Maintain Inventory</u>	9
6	<u>Maintain Request Queue</u>	13
7	<u>Acknowledge Resource Request</u>	17
8	<u>Place Order from Order Queue</u>	19
9	<u>Maintain Staff Records</u>	20
10	<u>View Staff Records</u>	24
11	<u>Maintain Prisoner Records</u>	25
12	<u>Maintain Prisoners' Medical Records</u>	30
13	<u>View Prisoner Record as per Authorization Level</u>	34
14	<u>Request Visit</u>	36
15	<u>Acknowledge Prison Visits</u>	38
16	<u>Maintain Visit Records</u>	39
17	<u>File Complaint</u>	43
18	<u>Update Complaint Status</u>	44
19	<u>View Audit Logs</u>	46
20	<u>View Prison Details</u>	47
21	<u>Enter Arrival/Leave Time</u>	48
22	<u>Maintain Department Records</u>	50
23	<u>ER Diagram</u>	54
24	<u>Class Diagram</u>	55
25	<u>Sequence Diagrams</u>	56

PROBLEM STATEMENT

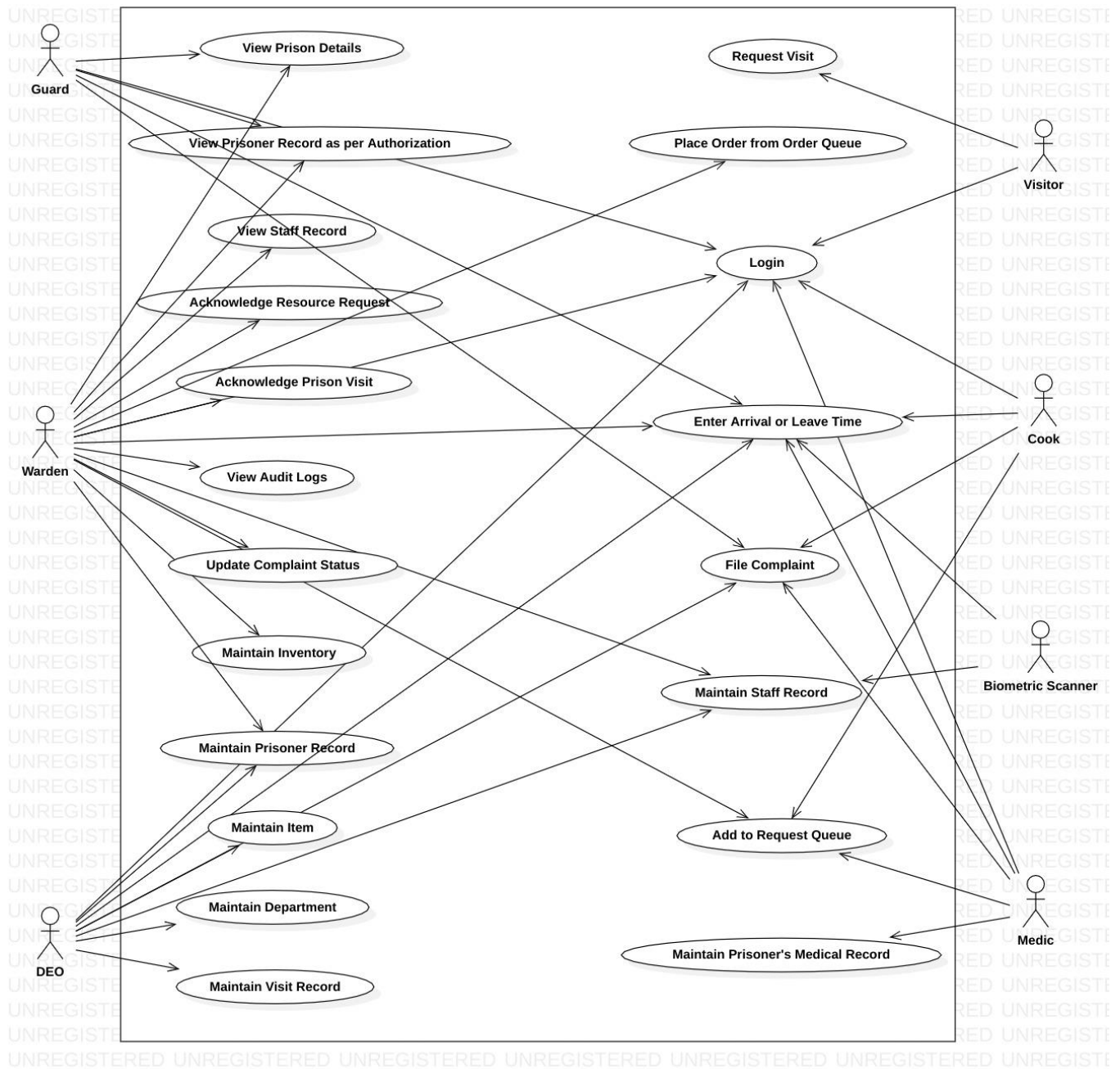
Continuous use of a file-based system in prisons has led to inconsistency, redundancy, data insecurity, lack of proper backup measures leading to untimely reports production, difficulties in managing data and underutilization of resources like time.

Such complex scenarios are often not easy to satisfy manually; thus call for computer software to handle. It would be best to not only access but also update information from anywhere on the globe, at any time.

The prison consists of warden, guards, prisoners and other staff; each staff member has a unique id associated with them.

It is required to develop a system that will manage information regarding resources (human and other assets) such as inventory count, staff records, prisoner records, audit logs, prison details, prison visits, complaints. The system should have the ability to request for resources and also order for said resources. It should be able to schedule prison visits and update audit trails by automation.

USE CASE DIAGRAM



USE CASE DESCRIPTION

1) LOGIN

a) Introduction: This use case describes how a user logs into the Prison Management System.

b) Actors:

- i) Warden
- ii) DEO
- iii) Guards
- iv) Medic
- v) Cook
- vi) Visitor

c) Pre-conditions: None

d) Post-conditions: If the use case is successful, the actor is logged into the system. If not, the system state is unchanged.

e) Basic flow: The use case starts when the actor wishes to login into the Prison Management System.

- i) The system requests that the actor enter their user-id and password.

- ii) The actor enters their user-id and password.
- iii) The system checks the actor's credentials and logs them into the system.

f) Alternate flow:

- i) **Invalid user-id:** If the actor's user-id is not present in the database, the system displays an error message. The actor can choose to either enter the user-id again or cancel the login, at which point, the use case ends.
- ii) **User-id and password don't match:** If the user-id and password don't match, the system displays an error message. The actor can choose to re-enter the password or cancel the login, at which point, the use case ends.

g) Special requirements: None

h) Use case relationships: None

2) MAINTAIN ITEM

a) Introduction: Allows the DEO to make changes in the item list. This might include things like beds, sheets, medical supplies, etc.

b) Actors:

i) DEO

c) Pre Conditions: The actor must be logged into the system.

d) Post Conditions: If the use case is successful, the item is updated.

e) Basic Flow: Starts when the actor chooses to update the item.

i) The screen displays the list of the items in the item database.

ii) The actor chooses to update/ delete from / add to item

(1) If the actor chooses to remove an item and the “Remove item” sub-flow is executed.

(2) If the actor chooses to add items to the list and the “Add item” sub-flow is executed.

(3) If the actor chooses to edit items from the list the “Update item” sub-flow is executed.

iii) Subflow

(1) Remove Item

- (a) The system displays the list of items in the database.
- (b) The actor chooses an item to remove.
- (c) The system checks if the item is available in inventory. If it is, the delete request is rejected.
- (d) The system prompts the actor to confirm the deletion.
- (e) The actor accepts the prompt.
- (f) The system removes the item from the database.

(2) Update Item

- (a) The system displays the list of items in the database.
- (b) Actor will select the item to be updated.
- (c) System will load the item record.
- (d) Actor will make changes to the record.
- (e) Actor will save the record.
- (f) The system will prompt the actor to confirm the update.
- (g) Actor will confirm the prompt.

(3) Add Item

- (a) The system creates an item template.
- (b) Actor enters:
 - (i) Item ID
 - (ii) Item name
- (c) The Actor clicks the “add” button
- (d) The system prompts the actor to confirm the insertion.
- (e) The system adds the item in the database when the actor wishes to submit it.

f) Alternate flow:

- i) **Add canceled:** If in the Add an item sub-flow, the user decides not to add the item information, the addition is canceled and the basic flow is restarted from the beginning.
- ii) **Update canceled:** If in the Update an item sub-flow, the user decides not to update the item information, the update is canceled and the basic flow is restarted from the beginning.
- iii) **Delete canceled:** If in the Delete an item sub-flow, the user decides not to delete the item information, the deletion is canceled and the basic flow is restarted from the beginning.

iv) **Delete Rejected:** If in the Delete and item sub-flow, the quantity of that item is not zero, the deletion is rejected and the basic flow is restarted from the beginning.

g) **Special requirements:** None

h) **Use case relationships:** None

3) MAINTAIN INVENTORY

a) **Introduction:** Allows the warden to make changes in the inventory. This might include things like beds, sheets, medical supplies, etc.

b) **Actors:**

i) Warden

ii) DEO

c) **Pre-conditions:** The actor must be logged on to the system.

d) **Post-conditions:** If the use case is successful, the inventory is updated.

e) Basic flow:

- i) The actor chooses to update the inventory.
- ii) The screen displays the list of the items in the inventory.
- iii) The actor chooses to update / delete from / add to inventory
 - (1) If the actor chooses to remove an item and the “Remove item” sub-flow is executed.
 - (2) If the actor chooses to add items to inventory and the “Add item” sub-flow is executed.
 - (3) If the actor chooses to edit items from the inventory the “Update item” sub-flow is executed

iv) Sub flow

(1) Remove item

- (a) The actor chooses an item to remove.
- (b) The system prompts the actor to confirm the deletion.
- (c) The actor accepts the prompt.
- (d) The system removes the item from the database.

(2) Add item

- (a) The system creates an item template.
- (b) Actor enters:
 - (i) Item ID
 - (ii) Item quantity
- (c) The Actor clicks the “add” button
- (d) The system prompts the actor to confirm the insertion.
- (e) The system adds the item in the database when the actor wishes to submit it.

(3) Update item

- (a) The system displays the list of items in the database.
- (b) Actor will select the item to be updated.
- (c) The system will load the item record.
- (d) The actor will make changes to the record.
- (e) The actor will save the record.
- (f) The system will prompt the actor to confirm the update.
- (g) The actor will confirm the prompt.

f) Alternate flow:

- i) **Add canceled** - If in the Add an item sub-flow, the user decides not to add the item information, the addition is cancelled and the basic flow is restarted from the beginning.
- ii) **Update canceled** - If in the Update an item sub-flow, the user decides not to update the item information, the update is cancelled and the basic flow is restarted from the beginning.
- iii) **Delete canceled** - If in the Delete an item sub-flow, the user decides not to delete the item information, the deletion is cancelled and the basic flow is restarted from the beginning.

g) **Special requirements:** None

h) **Use case relationships:** None

4) MAINTAIN REQUEST QUEUE

a) Introduction: Allows actors to maintain requests for resources/supplies. This includes adding, updating and deleting requests for medical resources.

b) Actors:

- i) Medics
- ii) Warden
- iii) Cook

c) Pre-conditions: The actor must be logged into the system.

d) Post-conditions: If the use case is successful, the resource request is added/updated/deleted from the system. Otherwise, the system state is unchanged.

e) Basic flow: Starts when the actor wishes to specify the function they would like to perform (add/update/delete).

- i) The system requests the actor to specify the function they would like to perform (add/update/delete).
- ii) One of the sub-flow will execute after getting the information.

- (1) If the actor selects “Add a request”, the “Add a request” sub-flow is executed.
- (2) If the actor selects “Update request”, the “Update request” sub-flow is executed.
- (3) If the actor selects “Delete request”, the “Delete request” sub-flow is executed.

iii) Subflow

(1) Add a request

- (a) The system generates a blank request.
- (b) The actor will fill in the equipment and quantity.
- (c) The actor will press submit.
- (d) The system will prompt the user to confirm the addition.
- (e) The actor will confirm the prompt.
- (f) The request will be added to the database by the system.

(2) Update a request

- (a) The system requires the actor to search for the particular record they want to update. Search can be done via date, ordered item.

- (b) The actor will select amongst all the appropriate search results shown by the system.
- (c) The system will open the selected record and make suitable updates and select submit.
- (d) The system will prompt the actor to confirm the addition.
- (e) The actor will confirm the prompt.
- (f) The information will be updated in the database by the system.

(3) Delete a request

- (a) The system requires the actor to search for the particular record they want to delete. Search can be done via date, ordered item.
- (b) The actor will select amongst all the appropriate search results shown by the system.
- (c) The system will open the selected record and prompt the actor to confirm the deletion.
- (d) The actor will confirm the prompt.
- (e) The system deletes the request from the database.

f) Alternate flow:

- i) Add canceled:** If in the Add a request sub-flow, the actor decides not to place a request, the addition is cancelled and the basic flow is restarted from the beginning.
- ii) Update cancelled:** If in the update a record sub-flow, the actor decides not to update the request information, the update is canceled and the basic flow is restarted from the beginning.
- iii) Delete canceled:** If in the delete a request sub-flow, the actor decides not to update the request information, the deletion is canceled and the basic flow is restarted from the beginning.

g) Special requirements: None

h) Use case relationships: None

5) ACKNOWLEDGE RESOURCE REQUEST

- a) Introduction:** Allows the actor to accept or reject resource requests.
- b) Actors:**
 - i) Warden
- c) Pre-conditions:** The actor must be logged into the system.
- d) Post-conditions:** If the use case is successful, resources may be added to the order queue and removed from the request queue, or request is rejected, else, the system state remains unchanged.
- e) Basic flow:** The actor chooses to view the request queue.
 - i) The screen displays a queue of requests.
 - ii) The actor selects a request from the queue to process.
 - iii) The actor is prompted to authorize or reject the request.
 - iv) One of the sub-flow will execute after getting the information.

- (1) If the actor chooses to authorize the request, the “Authorize” sub flow is executed.
- (2) If the actor chooses to reject the request, the “Reject” sub flow is executed.

v) Sub flow

(1) Authorize

- (a) The request is processed and the items are added to the order queue.
- (b) The request is removed from the request queue.

(2) Reject

- (a) The request is rejected and the item is not added to the order queue.

f) Alternate flow: None

g) Special requirements: None

h) Use case relationship: None

6) PLACE ORDER FROM ORDER QUEUE

- a) **Introduction:** Allows the actor to place an order of items in the order queue.
- b) **Actors:**
 - i) Warden
- c) **Pre-conditions:** The actor must be logged into the system.
- d) **Post-conditions:** If the use case is successful, items from the list are ordered, else, the system state is unchanged.
- e) **Basic flow:** The actor chooses to place order
 - i) The actor is presented with a list of items from the order queue which will be ordered.
 - ii) The actor proceeds by pressing the 'Order' button on the screen.
 - iii) The items from the order queue are ordered.
- f) **Alternate flow:**
 - i) **Order canceled:** The actor might press the 'Cancel' button, the operation is canceled, in which case, the use case ends.

g) **Special requirements:** None

h) **Use case relationship:** None

7) MAINTAIN STAFF RECORDS

a) **Introduction:** Allow actors to maintain staff details.
This includes adding, changing, and deleting staff information.

b) **Actors:**

i) DEO

ii) Warden

iii) Biometric Scanner

c) **Pre-conditions:** The actor must be logged onto the system.

d) **Post-conditions:** If the use case is successful, the staff information is added/updated/deleted from the system. Otherwise, the system state is unchanged.

e) **Basic flow:** Starts when the actor wishes to specify the function they would like to perform (add/update/delete).

- i) The system requests the actor to specify the function they would like to perform (add/update/delete).
- ii) One of the sub-flow will execute after getting the information.
 - (1) If the actor selects “Add staff record”, the “Add staff record” sub-flow is executed.
 - (2) If the actor selects “Update staff record”, the “Update staff record” sub-flow is executed.
 - (3) If the actor selects “Delete staff record”, the “Delete staff record” sub-flow is executed.

iii) Subflows

(1) Add a staff

(a) The system requires the actor to enter:

- (i) Name
- (ii) Address
- (iii) Date of Birth
- (iv) Phone number
- (v) Type of staff
- (vi) Gender
- (vii) Leaves Left
- (viii) Email

- (ix) Department ID
- (x) Fingerprint
- (b) The actor proceeds to add the information.
- (c) The system prompts the actor to confirm the actions.
- (d) The actor confirms the actions.
- (e) The system generates a unique id and displays it.
- (f) The system inserts the staff record in the database.

(2) Update staff record

- (a) The system requires the actor to enter the staff-id.
- (b) Actor enters the staff-id. The system retrieves and displays the staff information.
- (c) Actor makes the desired changes to the staff information.
- (d) The system prompts the actor to confirm the actions.
- (e) The actor accepts the prompt
- (f) After changes, the system updates the staff record with changed information.

(3) Delete staff record

- (a) The system requests the actor to specify the staff-id.
- (b) Actor enters the staff-id. The system retrieves and displays the staff information.
- (c) The system prompts the Actor to confirm the deletion.
- (d) The system deletes the staff's record.

f) Alternate flow:

- i) **Staff-Id not found:** If in the update or delete sub-flows, a staff with the specified id does not exist, the system displays an error message. The actor may enter a different id or cancel the operation. At this point, the use case ends.
- ii) **Update canceled:** If in the update staff sub-flow, the actor decides not to update the staff record, the update is canceled and the basic flow is restarted from the beginning.
- iii) **Delete canceled:** If in the delete sub-flow, the actor decides not to delete the staff record, the delete is canceled and the basic flow is restarted from the beginning.

iv) **Insertion cancelled** - If in the add sub-flow, the actor decides not to add the staff record, the insertion is canceled and the basic flow is restarted from the beginning.

g) **Special requirements:** None

h) **Use case relationships:** None

8) VIEW STAFF RECORD

a) **Introduction:** Allow the actor to view staff records.

b) **Actors:**

i) Warden

c) **Pre-conditions:** Actor must be logged on to the system

d) **Post-conditions:** If the use case is successful, the staff record is displayed on screen, else, the system state remains unchanged.

e) **Basic flow:**

i) Begins when the actor wishes to view staff record

ii) The system requires the actor to enter the staff-id and search.

iii) The system retrieves the required details of the personnel and displays the results.

f) Alternate flow:

i) **Staff-Id not found:** If the staff-id does not exist, the system displays an error message. The actor may enter a different id or cancel the operation, at which point, the use case ends.

g) Special requirements: None

h) Use case relationship: None

9) MAINTAIN PRISONER RECORDS

a) Introduction: Allow actors to maintain prisoner details. This includes adding, editing and deleting prisoner information.

b) Actors:

i) DEO

ii) Warden

c) Pre-conditions: The actor must be logged into the system.

d) Post-conditions: If the use case is successful, the prisoner information is added/updated/deleted from the system. Otherwise, the system state is unchanged.

e) Basic flow: Starts when the actor wishes to specify the function they would like to perform (add/update/delete).

i) The system requests the actor to specify the function they would like to perform (add/update/delete).

ii) One of the sub-flow will execute after getting the information.

(1) If the actor selects “Add a prisoner record”, the “Add a prisoner record” sub-flow is executed.

(2) If the actor selects “Update prisoner record”, the “Update prisoner” sub-flow is executed.

(3) If the actor selects “Delete prisoner record”, the “Delete prisoner” sub-flow is executed.

iii) **Subflows**

(1) **Add a prisoner record**

- (a) The system requires the actor to enter:
 - (i) Prisoner Id
 - (ii) Name
 - (iii) Address
 - (iv) First Contact Number
 - (v) Date of Birth
 - (vi) Cell Number
- (b) The actor proceeds to add the information.
- (c) The system prompts the actor to confirm the actions.
- (d) The actor confirms the actions.
- (e) The system generates a unique id and displays it.
- (f) The system inserts the prisoner's record in the database.

(2) Update prisoner record

- (a) The system requires the DEO to enter the prisoner-id.
- (b) The actor enters the prisoner-id. The system retrieves and displays the prisoner information.

- (c) The actor makes the desired changes to the prisoner information and proceeds to update the information.
- (d) The system prompts the actor to confirm the actions.
- (e) The actor accepts the prompt
- (f) After changes, the system updates the prisoner record with changed information.

(3) Delete prisoner record

- (a) The system requests the actor to specify the prisoner-id.
- (b) The actor enters the prisoner-id. The system retrieves and displays the prisoner information.
- (c) The system prompts the actor to confirm the deletion.
- (d) The system deletes the prisoner's record.

f) Alternate flow:

- i) **Prisoner not found:** If in the update or delete sub-flows, a prisoner with the specified id does not exist, the system displays an error message. The

actor may enter a different id or cancel the operation. At this point, the use case ends.

- ii) **Update canceled:** If in the update a prisoner sub-flow, the actor decides not to update the prisoner record, the update is canceled and the basic flow is restarted from the beginning.
- iii) **Delete canceled:** If in the delete sub-flow, the actor decides not to delete the prisoner record, the delete is canceled and the basic flow is restarted from the beginning.
- iv) **Insertion cancelled** - If in the add sub-flow, the actor decides not to add the prisoner record, the insertion is canceled and the basic flow is restarted from the beginning.

g) Special requirements: None

h) Use case relationships: None

10) MAINTAIN PRISONER'S MEDICAL RECORDS

- a) **Introduction:** Allows medical personnel to maintain prisoner details. This includes viewing, adding, changing and deleting prisoner's information.
- b) **Actors:**
 - i) Medical staff
- c) **Pre-conditions:** The actor must be logged into the system.
- d) **Post-conditions:** If the use case is successful, the prisoner's medical information is added/updated/deleted from the system. Otherwise, the system state is unchanged.
- e) **Basic flow:** Starts when the actor wishes to specify the function they would like to perform (add/update/delete).
 - i) The system requests the actor to specify the function they would like to perform (add/update/delete).
 - ii) One of the sub-flow will execute after getting the information.

- (1) If the actor selects “Add record”, the “Add record” sub-flow is executed.
- (2) If the actor selects “Update record”, the “Update record” sub-flow is executed.
- (3) If the actor selects “Delete record”, the “Delete record” sub-flow is executed.
- (4) If the actor selects “view record”, the “view record” sub-flow is executed.

iii) Subflow

(1) Add record

- (a) The system requires the actor to take a biometric scan of the prisoner, the system will search for the prisoner in the database and will create a blank record.
- (b) The actor will fill in the appropriate details such as the cause of illness, medicines, etc.
- (c) The user will press submit.
- (d) The system will prompt the user to confirm the insertion.
- (e) The user will confirm the prompt.

(2) Update record

- (a) The system requires the actor to search for the particular record they want to update. Search can be done via date, prisoner id.
- (b) The actor will select amongst all the appropriate search result shown by the system
- (c) The system will open the selected record and make suitable updates and select submit.
- (d) The system will prompt the actor to confirm the addition.
- (e) The actor will confirm the prompt.

(3) Delete a record

- (a) The system requires the actor to search for the particular record they want to update. Search can be done via date, prisoner id.
- (b) The actor will select amongst all the appropriate search results shown by the system.
- (c) The system will open the selected record and prompt the user to confirm the deletion.
- (d) The actor will confirm the prompt.

(4) View a record

- (a) The system requires the actor to search for the particular record they want to view. Search can be done via date, prisoner id.
- (b) The actor will select amongst all the appropriate search results shown by the system.
- (c) The system will open the selected record.

f) Alternate flow:

- i) No prisoner record found:** If in the update/delete/add record flow, the system fails to locate any appropriate prisoner record or prisoner's medical record. The actor can re-submit the input parameters or cancel the operation, at which point the use case ends.
- ii) Add canceled:** If in the Add record sub-flow, the actor decides not to add the prisoner information, the addition is cancelled and the basic flow is started from the beginning.
- iii) Update canceled:** If in the update a record sub-flow, the actor decides not to update the prisoner information, the update is canceled and the basic flow is started from the beginning.

iv) **Delete canceled:** If in the delete a record sub-flow, the actor decides not to update the prisoner information, the deletion is canceled and the basic flow is started from the beginning.

g) **Special requirements:** None

h) **Use case relationships:** None

11) VIEW PRISONER RECORD AS PER AUTHORIZATION LEVEL

a) **Introduction:** Allows the actor to view prisoner information as per their authorization level.

b) **Actors:**

i) Guard

ii) Warden

c) **Pre-conditions:** The actor must be logged onto the system.

- d) Post-conditions:** If the use case is successful, the prisoner information is displayed by the system. Otherwise, the system state is unchanged.
- e) Basic flow:** Begins when the actor wishes to view a prisoner's record
- i) The system requires the actor to enter the prisoner-id and the details they want to view.
 - ii) The system retrieves the required details of the prisoner.
 - iii) The details are displayed.
- f) Alternate flow:**
- i) **Actor not authorized:** If the security level of the information requested is more than the authorization level of the requesting actor, the request is canceled by the system and message displayed. At which point, the use case ends.
 - ii) **Prisoner not found:** If while requesting the information of a prisoner, the prisoner-id does not exist, the system displays an error message. The actor may enter a different id or cancel the operation. At which point, the use case ends.
- g) Special requirements:** None

h) Use case relationships: None

12) REQUEST VISIT

a) **Introduction:** Allow actors to request visits.

b) **Actors:**

i) Visitor

c) **Pre-conditions:** The actor must be logged onto the system.

d) **Post-conditions:** If the use case is successful, the visit request is queued in the system. Otherwise, the system state is unchanged.

e) **Basic flow:** Starts when a wishes to request a visit.

i) The system requires the visitor to enter:

- Prisoner ID
- Visitor ID
- Date-Time of Visit
- Status
- Relationship with prisoner
- Purpose for visit

- ii) The system displays a list of available slots for the given prisoner.
- iii) The actor chooses from among the list of slots.
- iv) The system prompts the actor to confirm the actions
- v) The actor confirms the action.
- vi) The system generates a unique visit-id and adds the visit record to the queue to be acknowledged by the warden.

f) Alternate flows

- i) **Prisoner not found:** If while requesting a visit with a prisoner, the prisoner-id does not exist, the system displays an error message. The visitor may enter a different id or cancel the operation. At this point, the use case ends.
- ii) **No available time slot:** If there are no available time slots, the system displays the message, and the use case ends.
- iii) **Request canceled:** If the visitor decides not to request the visit, the request is canceled and the basic flow is restarted from the beginning.

g) Special requirements: None

h) Use case relationships: None

13) ACKNOWLEDGE PRISON VISITS

a) **Introduction:** Allow actor to authorize or reject prison visit requests

b) **Actors:**

i) Warden

c) **Pre-conditions:** The actor must be logged into the system

d) **Post-conditions:** If the use case is successful, the visit is either authorized or rejected, else, the system remains unchanged.

e) **Basic flow:** The actor chooses to view the request queue.

i) The system displays the queue of requests.

ii) The actor selects the request to process.

iii) The actor will authorize or reject the selected request.

iv) The concerned parties are notified by their contact information.

f) Alternate flow: None

g) Special requirements: None

h) Use case relationship: None

14) MAINTAIN VISIT RECORDS

a) **Introduction:** Allow actors to maintain visit details. This includes changing visits.

b) **Actors:**

i) DEO

c) **Pre-conditions:** The actor must be logged onto the system.

d) **Post-conditions:** If the use case is successful, the visit information is updated from the system. Otherwise, the system state is unchanged.

e) Basic flow: Starts when the actor wishes to specify the function they would like to perform (add/update/delete).

i) The system requests the actor to specify the function they would like to perform (add/update/delete).

ii) One of the sub-flow will execute after getting the information.

(1) If the actor selects “Add record”, the “Add record” sub-flow is executed.

(2) If the actor selects “Update record”, the “Update record” sub-flow is executed.

(3) If the actor selects “Delete record”, the “Delete record” sub-flow is executed.

(4) If the actor selects “view record”, the “view record” sub-flow is executed.

iii) **Subflows**

(1) **Add a record**

(a) The system requires the actor to enter the visit ID, the system will search for the visit in the database and will create a blank record.

(b) The actor will log in the visit time.

- (c) The user will press submit.
- (d) The system prompts the actor to confirm the actions.
- (e) The actor confirms the actions and the system adds the record.

(2) Update a record

- (a) The system requires the actor to search for the particular record they want to update. Search is done via visit ID.
- (b) The actor will select amongst all the appropriate search result shown by the system
- (c) The system will open the selected record and make suitable updates and select submit.
- (d) The system will prompt the actor to confirm the addition.
- (e) The actor will confirm the prompt.

(3) Delete a record

- (a) The system requires the actor to search for the particular record they want to delete. Search is done via visit ID.
- (b) The actor will select amongst all the appropriate search results shown by the system.

- (c) The system will open the record and prompt the user to confirm the deletion.
- (d) The actor will confirm the prompt and the system will delete the record.

f) Alternate flow:

- i) **Add canceled:** If in the add a record sub-flow, the actor decides not to add the visit record, the update is canceled and the basic flow is restarted from the beginning.
- ii) **Update canceled:** If in the update a record sub-flow, the actor decides not to update the visit record, the update is canceled and the basic flow is restarted from the beginning.
- iii) **Delete canceled:** If in the delete a record sub-flow, the actor decides not to delete the visit record, the update is canceled and the basic flow is restarted from the beginning.
- iv) **Visit ID not found:** If the visit-id does not exist, the system displays an error message. The actor may enter a different id or cancel the operation. At which point, the use case ends.

g) Special requirements: None

h) Use case relationships: None

15) FILE COMPLAINT

a) **Introduction:** Allows the actors to file a complaint.

b) **Actors:**

- i) Medical staff
- ii) Cook
- iii) Prisoner
- iv) DEO

c) **Pre-conditions:** The actor must be logged into the system.

d) **Post-conditions:** If the use case is successful, the system logs in the complaint. Otherwise, the system state is unchanged.

e) **Basic flow:** Starts when the actor wishes to file a complaint.

- i) The system will generate a blank complaint template.

- ii) The actor will fill in the appropriate details and click on submit
- iii) The system will prompt the actor to confirm their actions.
- iv) The actor will confirm the prompt.

f) Alternate flow:

- i) **Complaint submission cancelled:** If the actor decides not to submit the complaint, the submission process is cancelled and the basic flow is restarted from the beginning.

g) Special requirements: None

h) Use case relationships: None

16) UPDATE COMPLAINT STATUS

- a) **Introduction:** Allows the actors to update complaint status.
- b) **Actors:**
 - i) Warden
- c) **Pre-conditions:** The actor must be logged into the system.

d) Post-conditions: If the use case is successful, the actor is able to update the status of the complaint.

Otherwise, the system state is unchanged.

e) Basic flow: Starts when the actor wishes to update the status of the complaint.

i) The system will display a list of complaints filed.

ii) The actor will select the appropriate complaint

iii) The system will display the complaint

iv) The actor will review the complaint and change the status to “reviewed”.

v) The system will prompt the actor to confirm the changes

vi) The actor will confirm the prompt.

f) Alternate flow:

i) **Complaint status update cancelled:** If the actor decides not to update the complaint status, the submission process is cancelled and the basic flow is restarted from the beginning.

g) Special requirements: None

h) Use case relationships: File complaint

17) VIEW AUDIT LOGS

a) Introduction: Allow warden to view the audit logs.

Audit logs are records of all events and changes. For example, change in state of inventory or order queue.

b) Actors:

i) Warden

c) Pre-conditions: Actor must be logged into the system

d) Post-conditions: If the use case is successful, audit logs are displayed on the screen, else, the system state remains unchanged.

e) Basic flow: The actor chooses to view the audit logs

i) The system will list down the different types of audit logs. (Staff audit, order queue audit, biometric audit etc.)

ii) The actor will select the apt category

iii) The system will display the logs corresponding to the selected category.

f) Alternate flow: None

g) Special requirements: None

h) Use case relationship: None

18) VIEW PRISON DETAILS

a) **Introduction:** Allow actors view prison details such as number of inmates, number of staff, prison id, prison address.

b) **Actors:**

i) Warden

ii) Guards

c) **Pre-conditions:** The actor must be logged into the system

d) **Post-conditions:** If the use case is successful, prison details are displayed on the screen, else, the system state remains unchanged.

e) **Basic flow:** The use case begins when the actor chooses to view the prison details.

i) The system retrieves the prison details.

ii) The system displays the result.

f) **Alternate flow:** None

g) **Special requirements:** None

h) Use case relationship: None

19) ENTER ARRIVAL/LEAVE TIME

a) **Introduction:** Allows an actor to log their arrival/leave time.

b) **Actors:**

- i) Medical staff
- ii) Biometric Scanner
- iii) Warden
- iv) Cook
- v) Guard
- vi) DEO

c) **Pre-conditions:** None

d) **Post-conditions:** If the use case is successful, the system logs in the entry containing the arrival/leave time and the id of the staff, along with whether it's for arrival or leave. Otherwise, the system state is unchanged.

e) **Basic flow:** Starts when the actor wishes to enter their arrival/leave time

- i) The system requests the actor to scan his registered finger.
- ii) The system will locate the employee in the database and make an entry.

f) Alternate flow:

- i) **Employee record not found:** If the employee record is not found after scanning, the system will display an error message, the actor can then perform a rescan or cancel the request, at which point the use case ends.

g) Special requirements: None

h) Use case relationships: None

20) MAINTAIN DEPARTMENT RECORDS

- a) **Introduction:** Allow actors to maintain department details. This includes adding, editing and deleting department information.
- b) **Actors:**
 - i) DEO
- c) **Pre-conditions:** The actor must be logged into the system.
- d) **Post-conditions:** If the use case is successful, the department information is added/updated/deleted from the system. Otherwise, the system state is unchanged.
- e) **Basic flow:** Starts when the actor wishes to specify the function they would like to perform (add/update/delete).
 - i) The system requests the actor to specify the function they would like to perform (add/update/delete).
 - ii) One of the sub-flow will execute after getting the information.

- (1) If the actor selects “Add a department”, the “Add a department” sub-flow is executed.
- (2) If the actor selects “Update department record”, the “Update department” sub-flow is executed.
- (3) If the actor selects “Delete department record”, the “Delete department” sub-flow is executed.

iii) Subflows

(1) Add a department

- (a) The system requires the actor to enter:
 - (i) Department Id
 - (ii) Name
 - (iii) Budget
- (b) The actor proceeds to add the information.
- (c) The system prompts the actor to confirm the actions.
- (d) The actor confirms the actions.
- (e) The system generates a unique id and displays it.
- (f) The system inserts the department’s record in the database.

(2) Update department

- (a) The system requires the DEO to enter the department-id.
- (b) The actor enters the department-id. The system retrieves and displays the department information.
- (c) The actor makes the desired changes to the department information and proceeds to update the information.
- (d) The system prompts the actor to confirm the actions.
- (e) The actor accepts the prompt
- (f) After changes, the system updates the department record with changed information.

(3) Delete department

- (a) The system requests the actor to specify the department-id.
- (b) The actor enters the department-id. The system retrieves and displays the department information.
- (c) The system prompts the actor to confirm the deletion.
- (d) The system deletes the department's record.

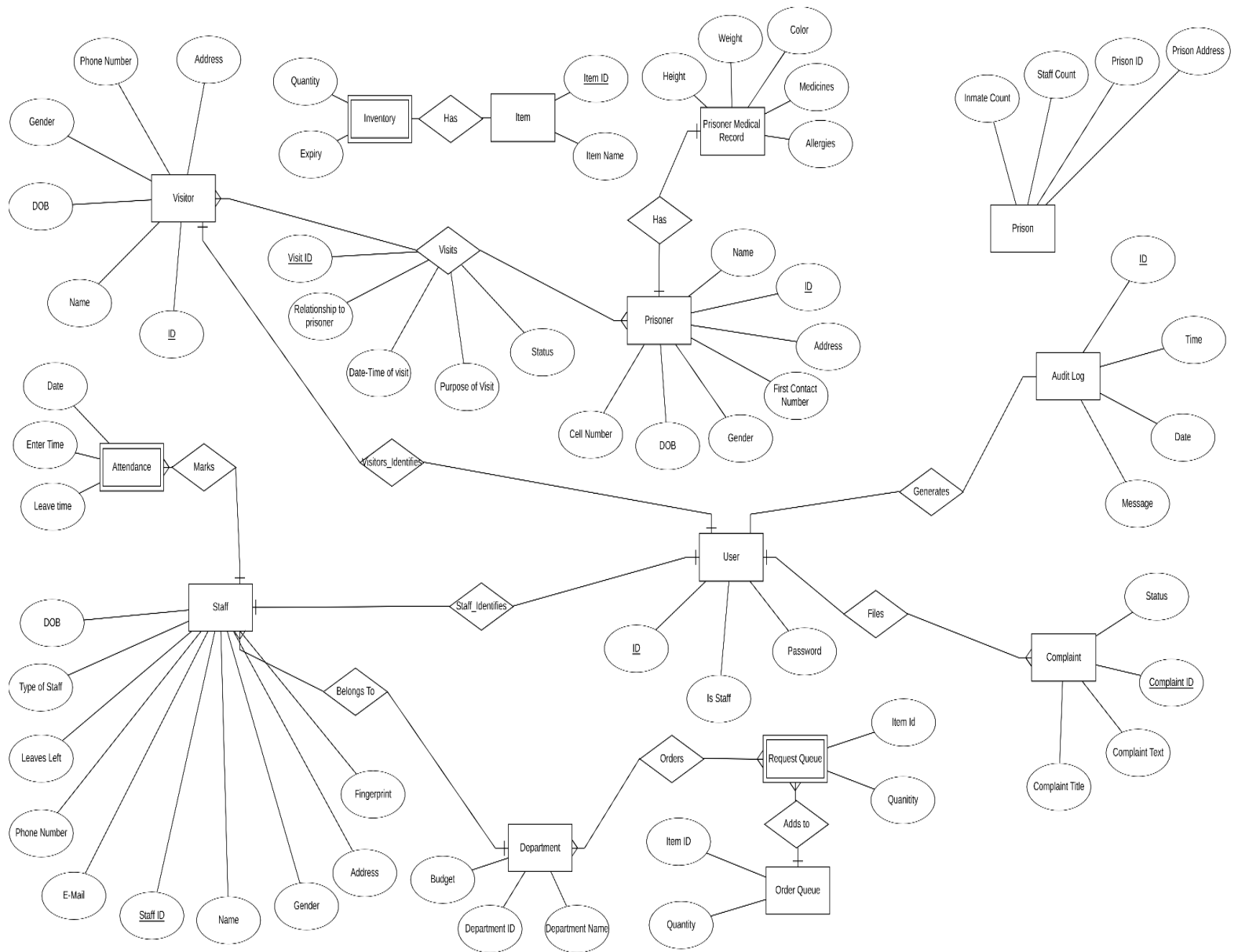
f) Alternate flow:

- i) Department not found:** If in the update or delete sub-flows, a department with the specified id does not exist, the system displays an error message. The actor may enter a different id or cancel the operation. At this point, the use case ends.
- ii) Update canceled:** If in the update a department sub-flow, the actor decides not to update the department record, the update is canceled and the basic flow is restarted from the beginning.
- iii) Delete canceled:** If in the delete sub-flow, the actor decides not to delete the department record, the delete is canceled and the basic flow is restarted from the beginning.
- iv) Insertion cancelled** - If in the add sub-flow, the actor decides not to add the department, the insertion is canceled and the basic flow is restarted from the beginning.

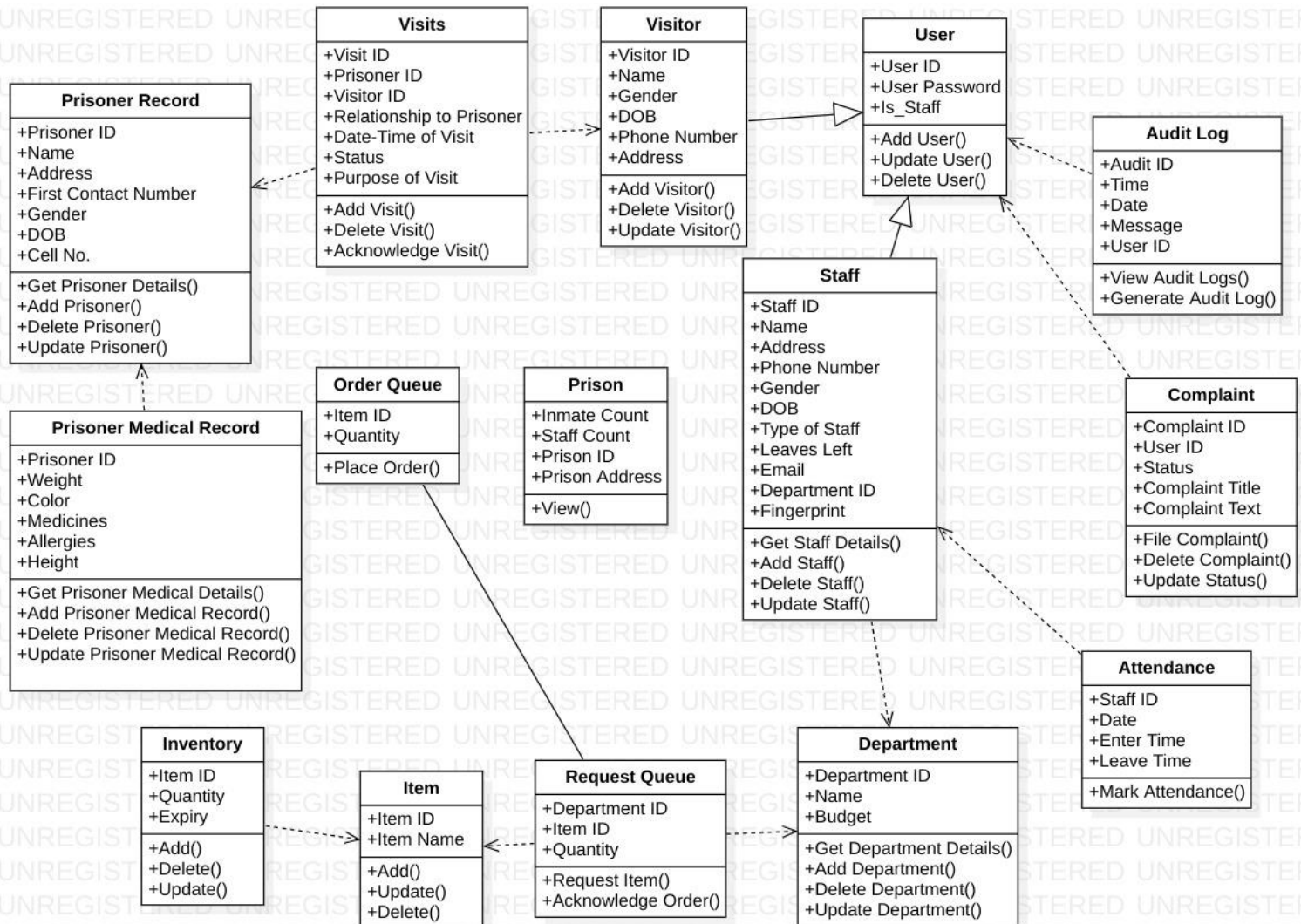
g) Special requirements: None

h) Use case relationships: None

ER DIAGRAM

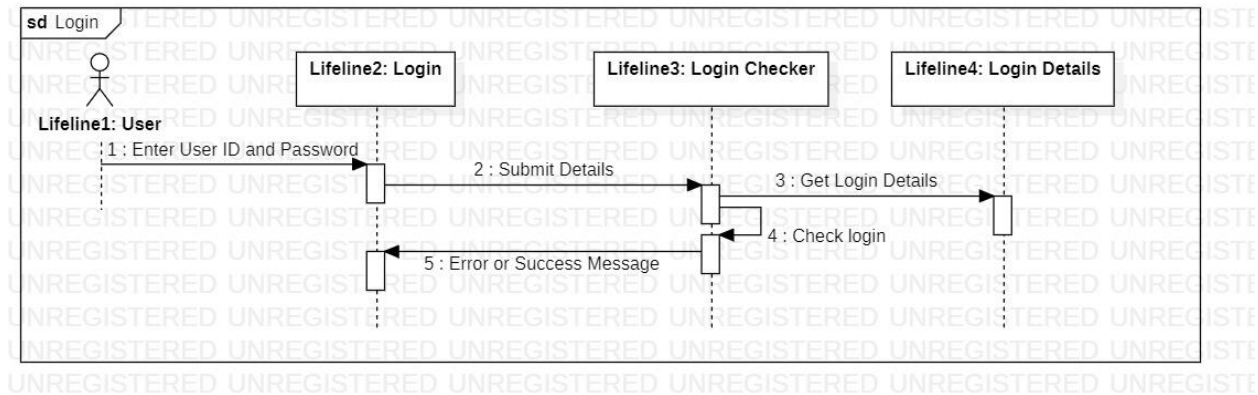


CLASS DIAGRAM

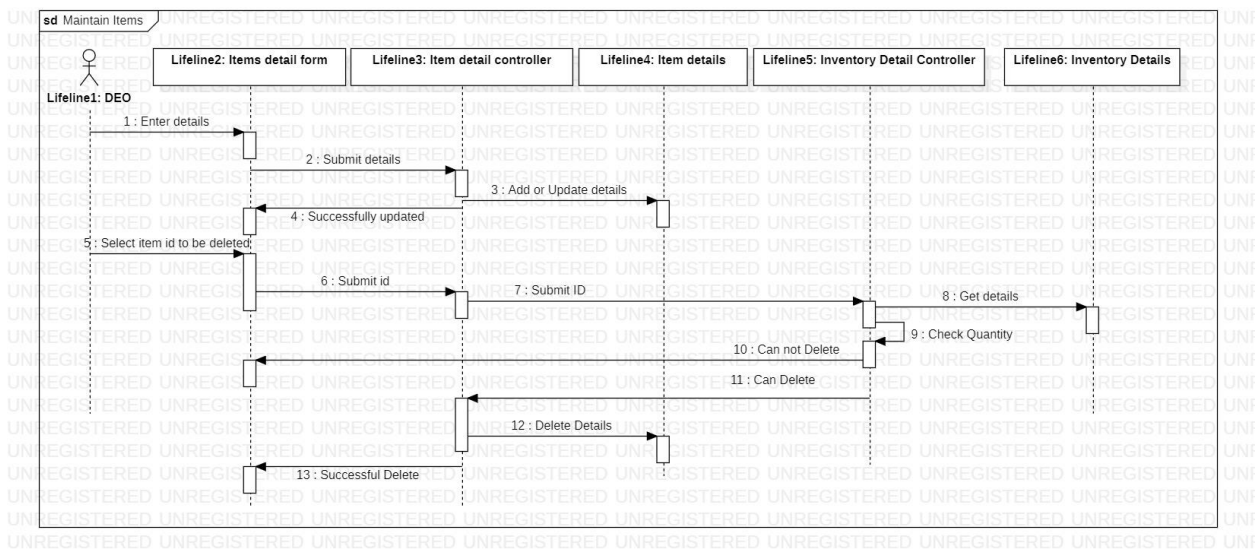


SEQUENCE DIAGRAMS

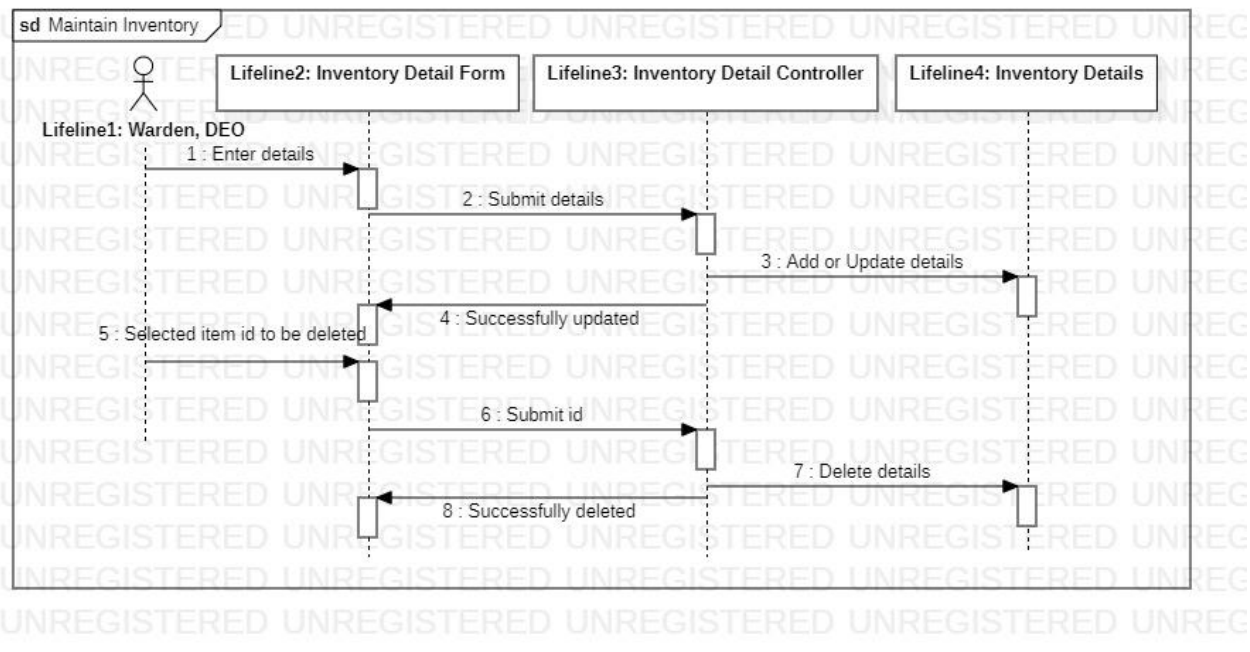
1. Login



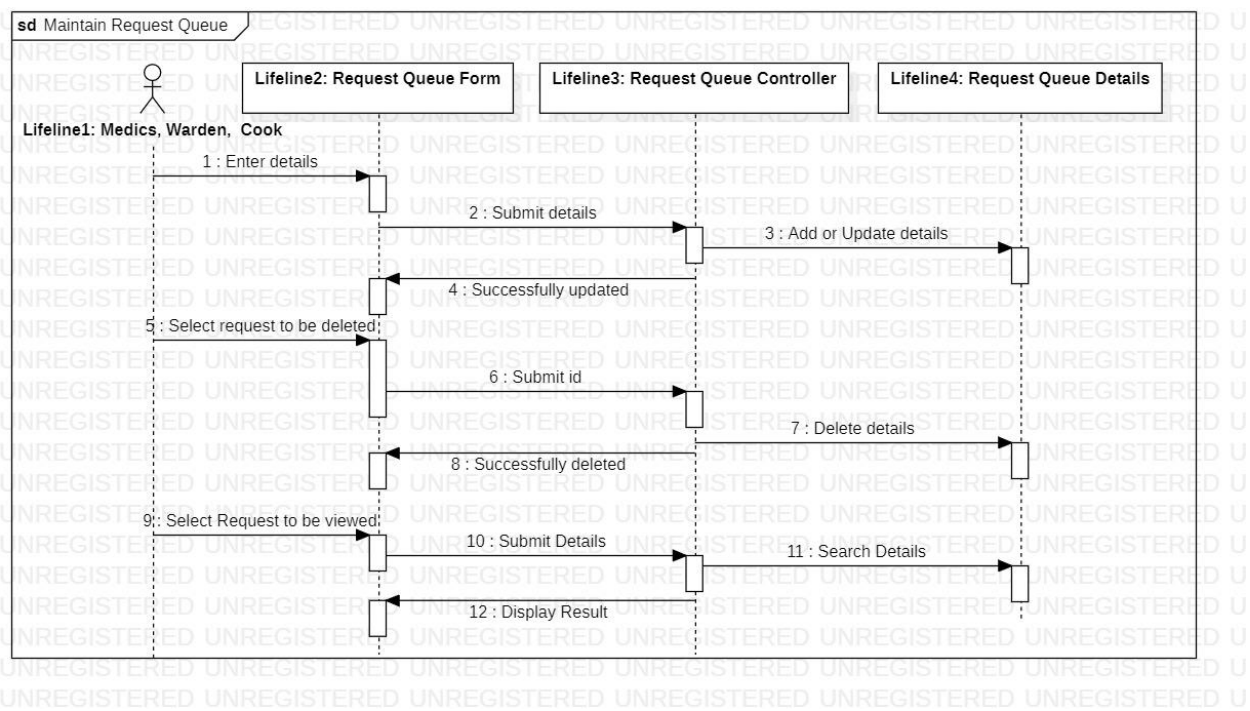
2. Maintain Items



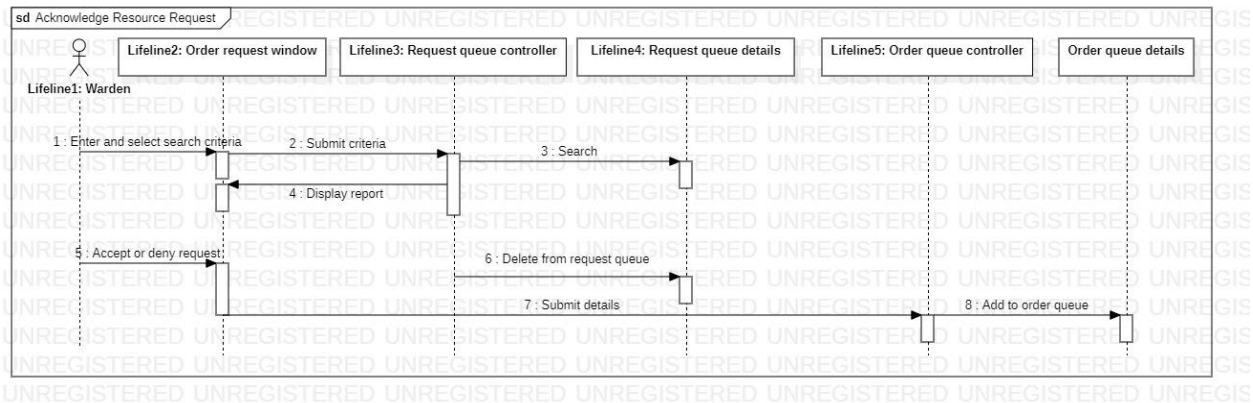
3. Maintain Inventory



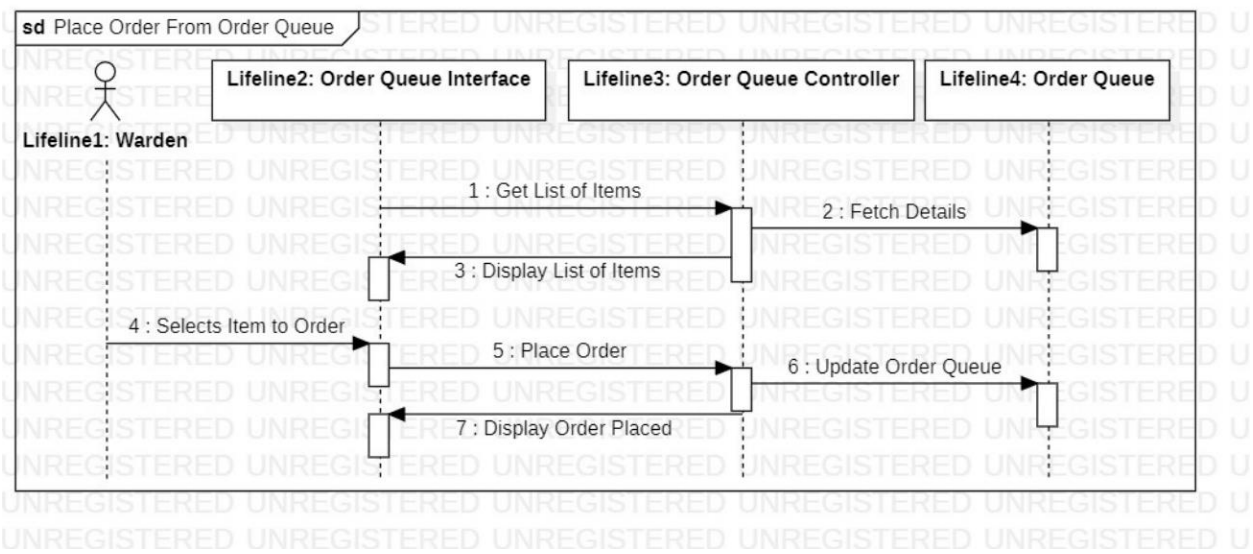
4. Maintain Request Queue



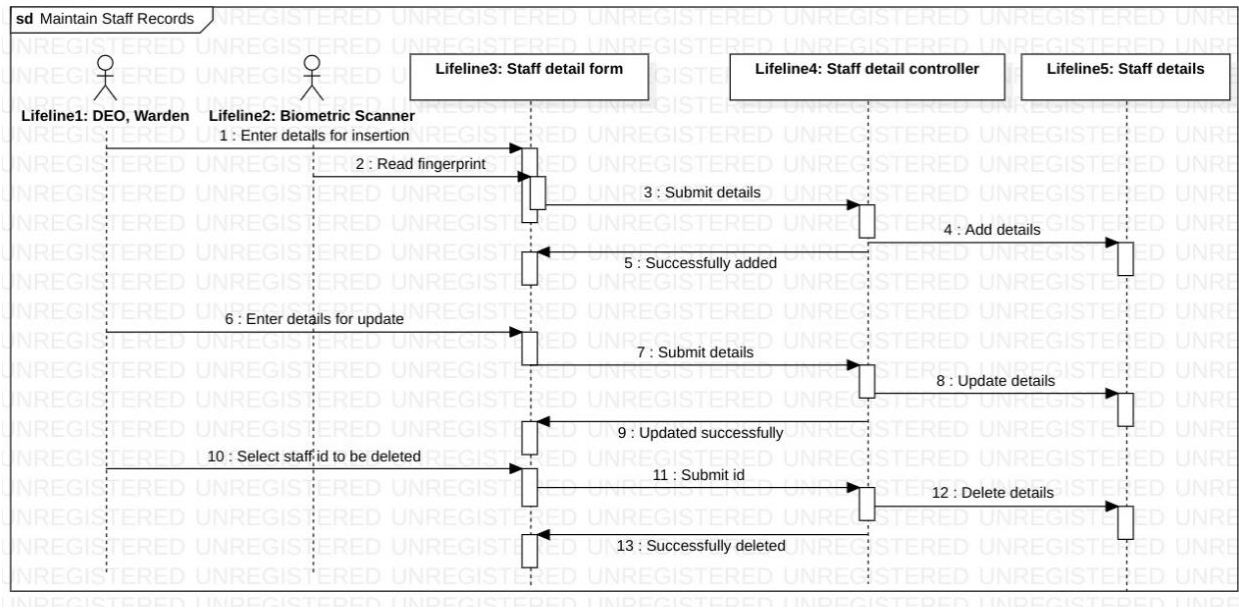
5. Acknowledge Resource Request



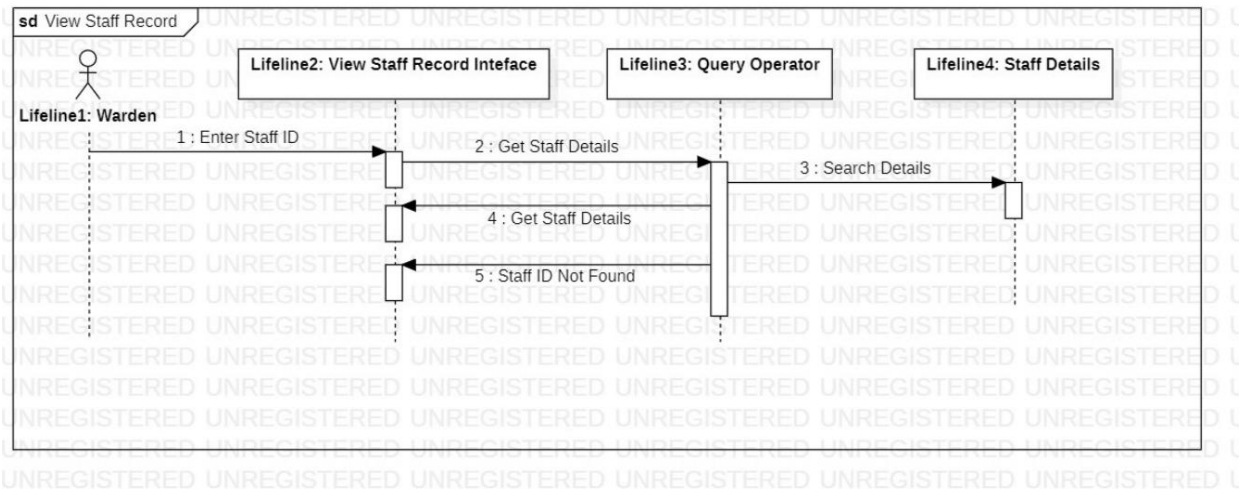
6. Place Order from Order Queue



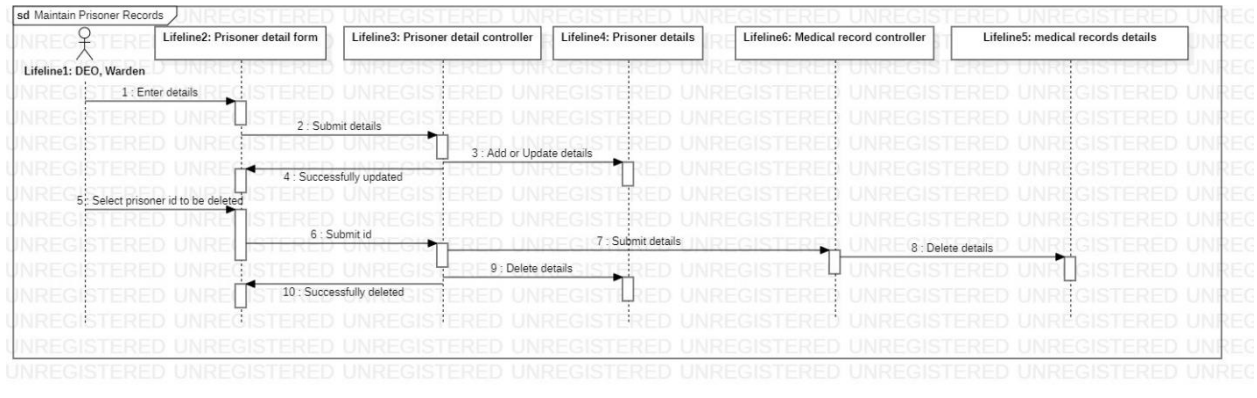
7. Maintain Staff Records



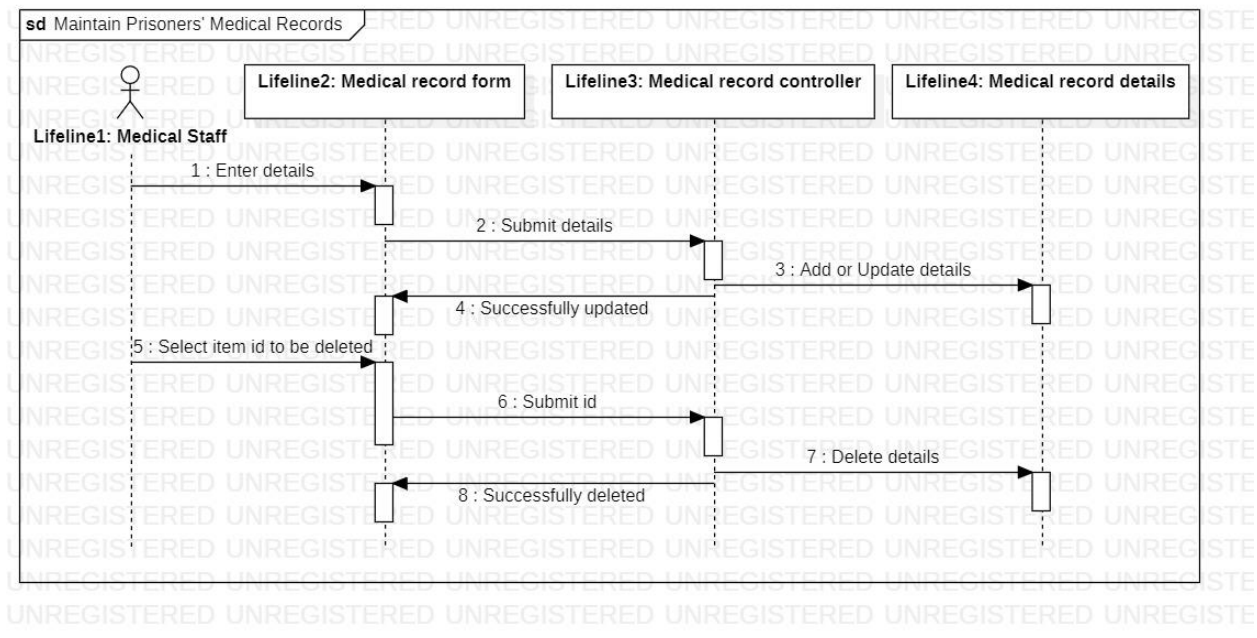
8. View Staff Records



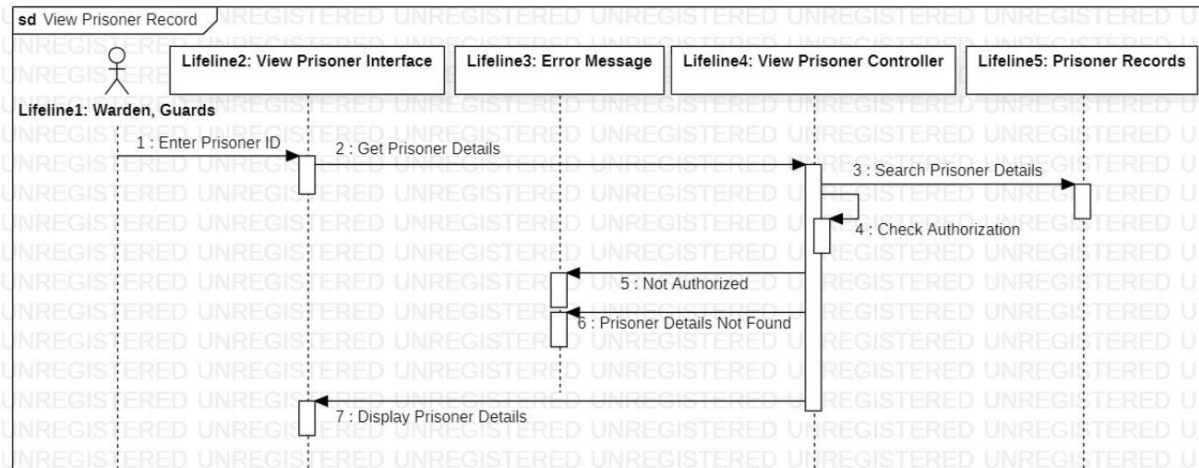
9. Maintain Prisoner Records



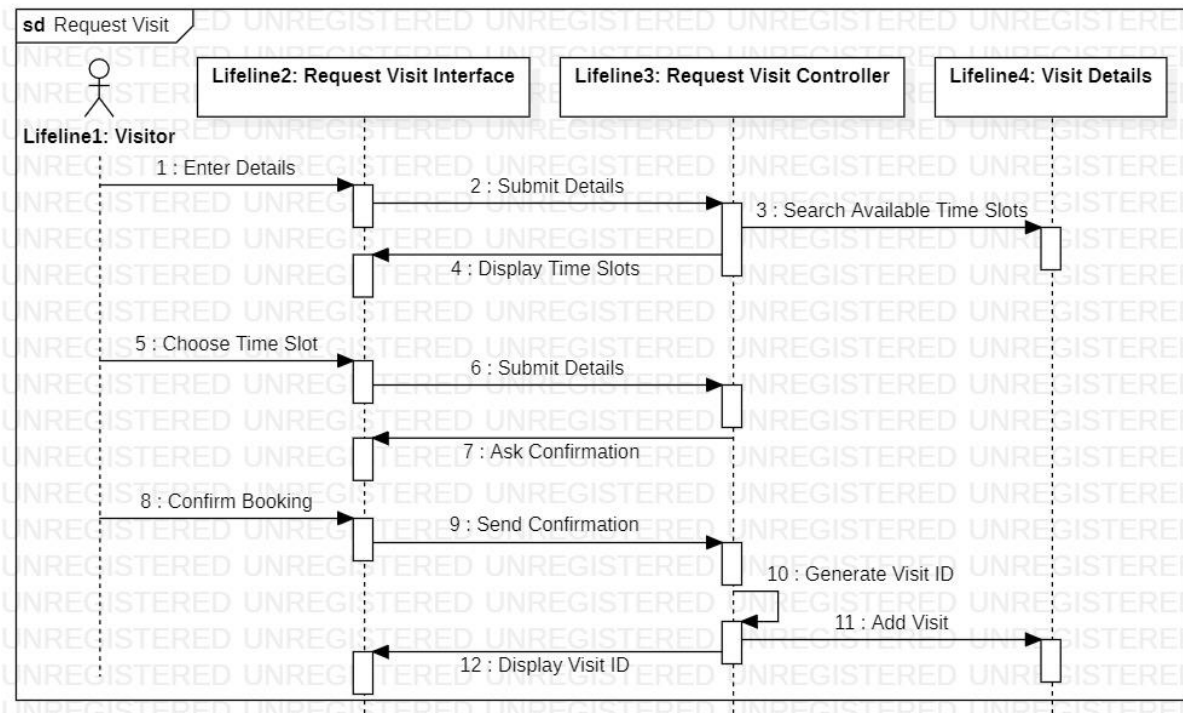
10. Maintain Prisoner's Medical Records



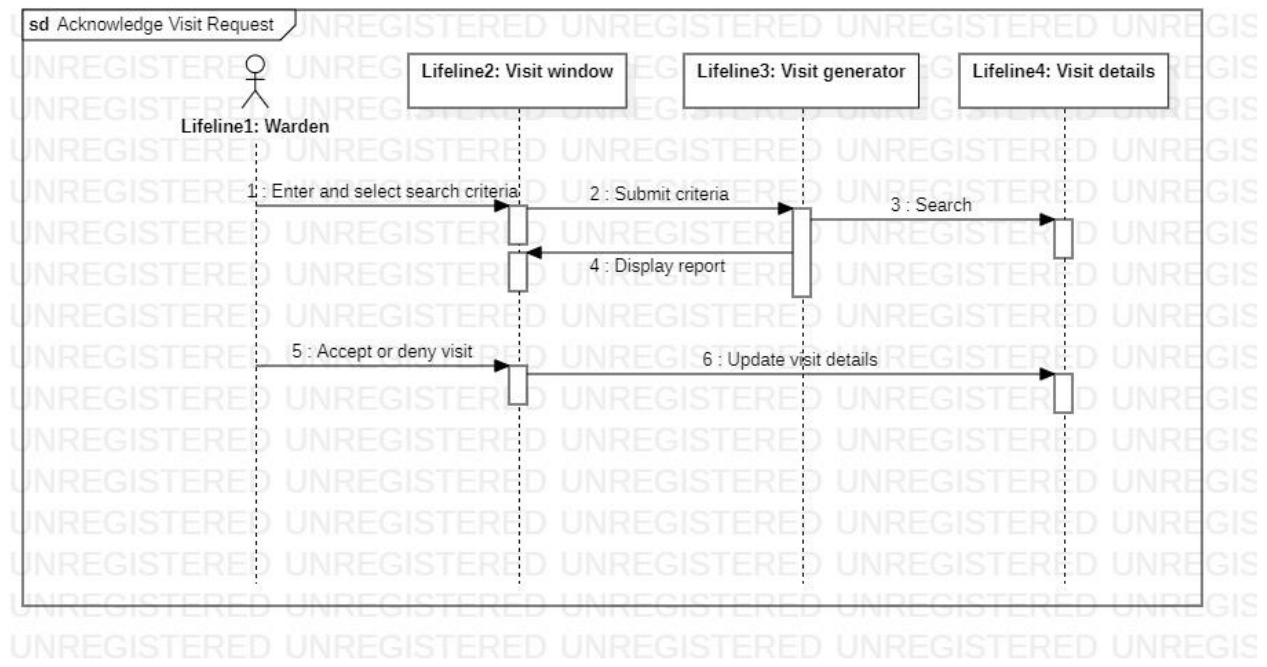
11. View Prisoner Record as per Authorization Level



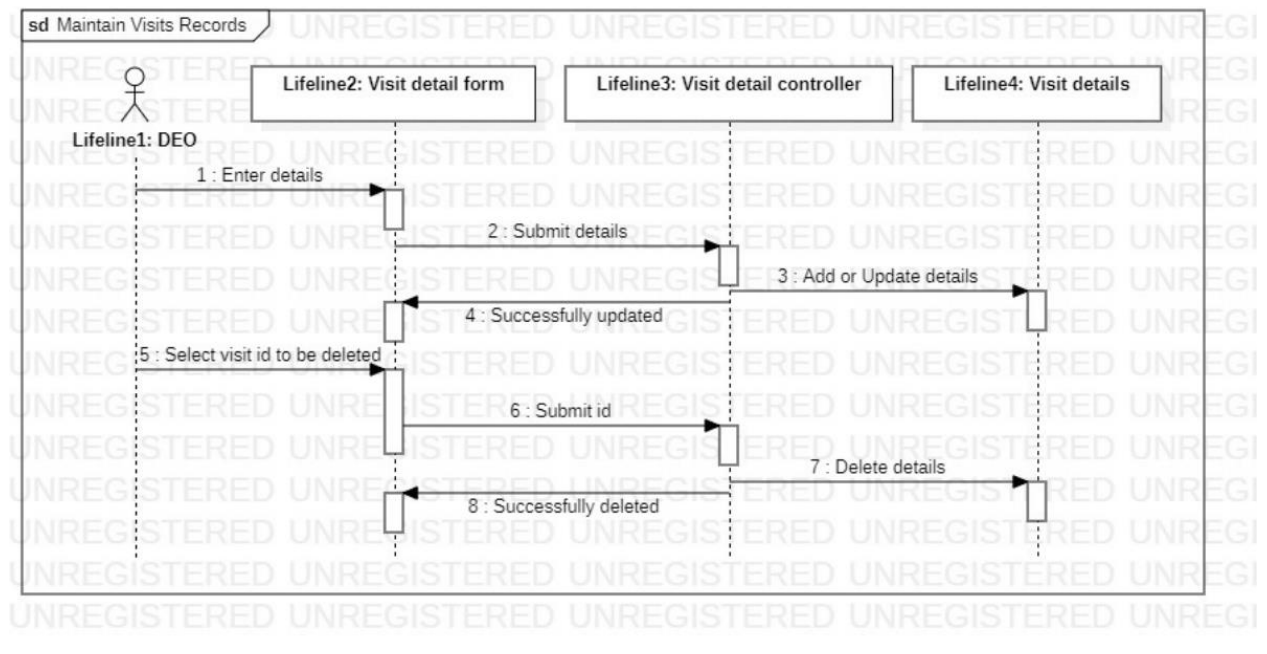
12. Request Visit



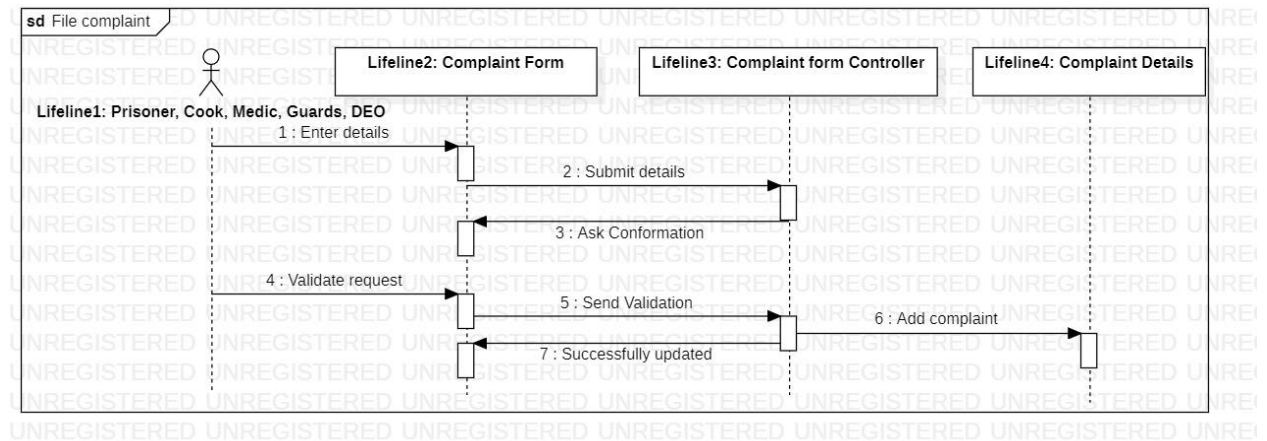
13. Acknowledge Prison Visits



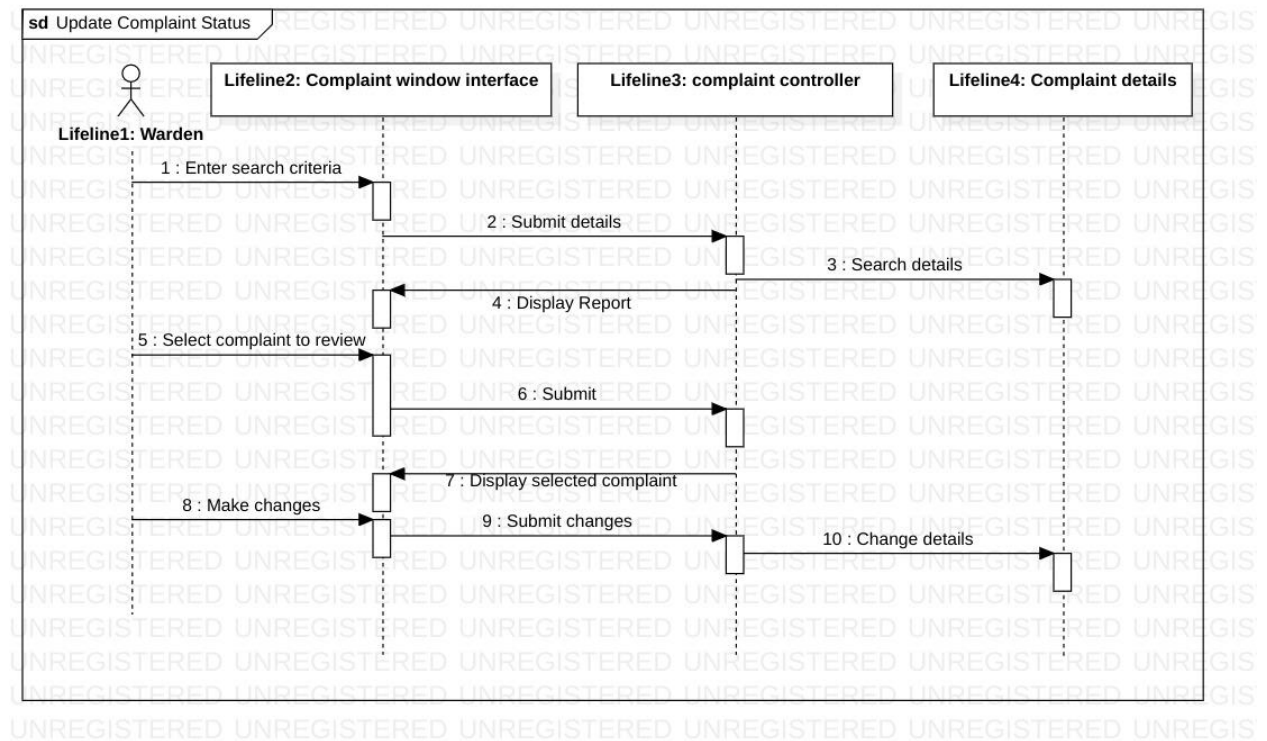
14. Maintain Visit Records



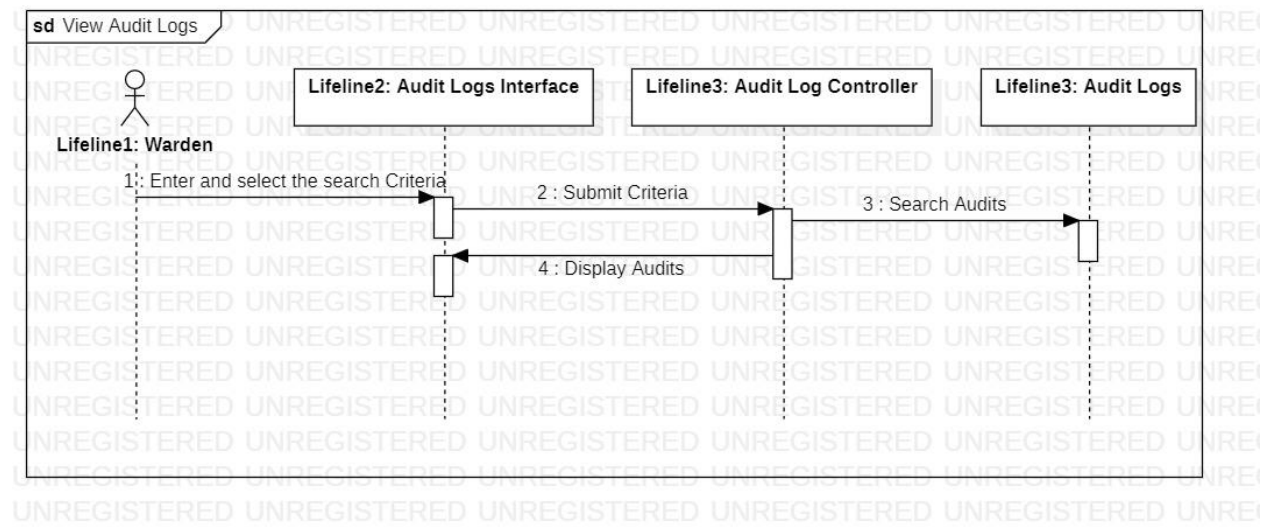
15. File Complaint



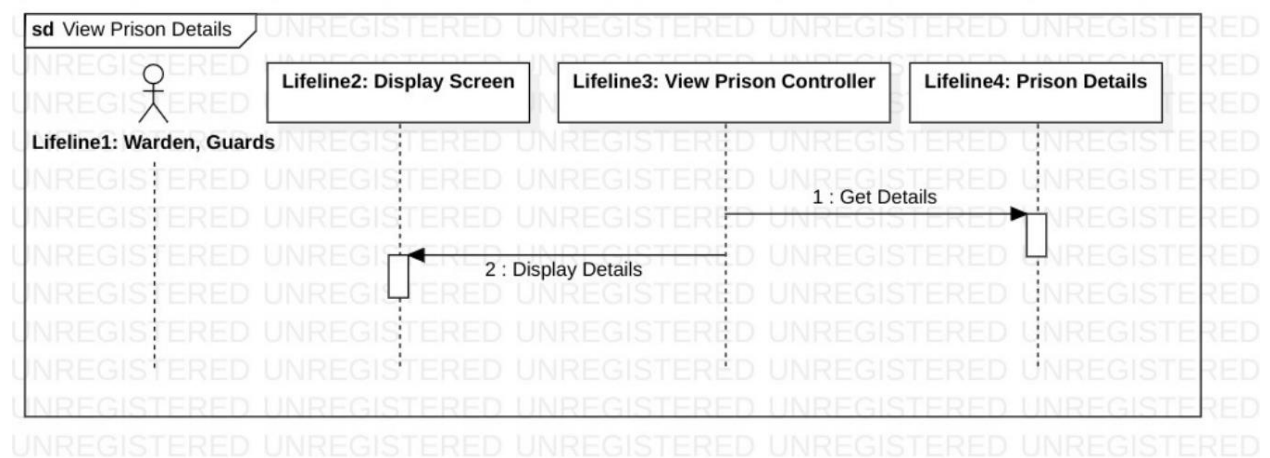
16. Update Complaint Status



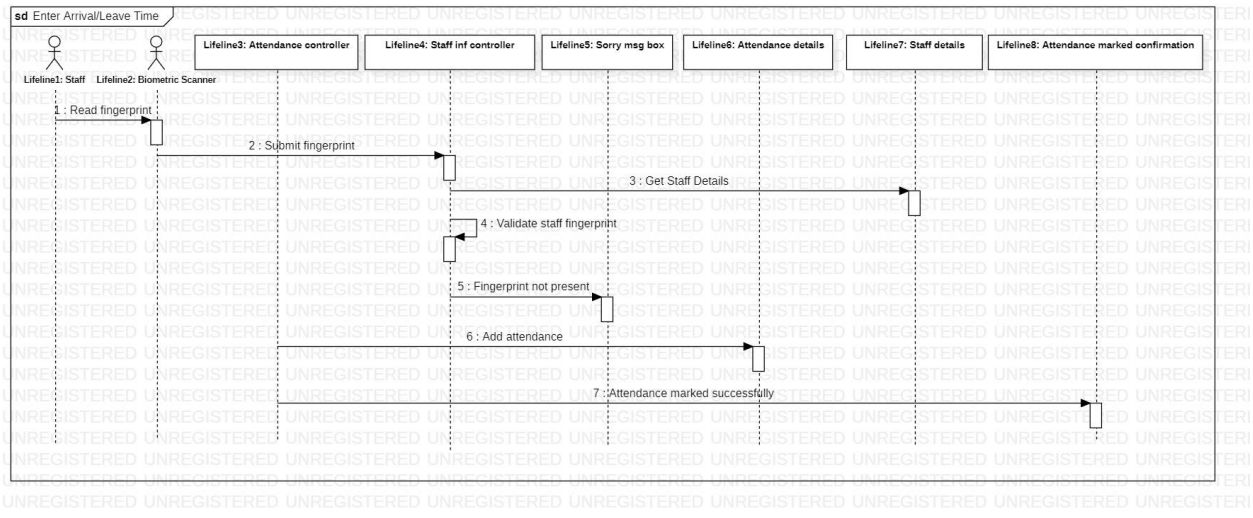
17. View Audit Logs



18. View Prison Details



19. Enter Arrival/Leave Time



20. Maintain Department Records

