جامعة الأميرة
نورة بنت عبد الرحمن
كلية علوم الحاسب والمعلومات

Princess Nourah bint
Abdulrahman University
College of Computer and
Information Sciences

# CS 385: Software Engineering
# Term Project

Title: Task Management System (TMS)

**First Semester 2024**

| Content | Student Name | FOR INSTRUCTOR'S USE ONLY | | |
|---|---|---|---|---|
| | | Max. Mark | Mark Obtain | Notes |
| Cover and Table of Contents (Report Style) | | 0.5 | | |
| Project Description | | | | |
| Functional Requirements (user and system requirements) | | 2 | | |
| Non-Functional Requirement | | | | |
| Context Diagram with description | | 0.5 | | |
| Use-case Diagram (for the whole system) | | 1.5 | | |
| Scenario (MSS & Extension) Use Case 1: Create task | | 1 | | |
| Scenario (MSS & Extension) Use Case 2: View task | | | | |
| Scenario (MSS & Extension) Use Case 3: Update task | | | | |
| Scenario (MSS & Extension) Use Case 4: receive notifications | | | | |
| Scenario (MSS & Extension) Use Case 5: search task | | | | |
| Scenario (MSS & Extension) Use Case 6: share task | | | | |
| Scenario (MSS & Extension) Use Case 7: delete task | | | | |
| Scenario (MSS & Extension) Use Case 8: assign task | | | | |
| Scenario (MSS & Extension) Use Case 9: manage task | | | | |
| Class Diagram (for the whole system) | | 1 | | |
| Sequence Diagram Function 1: create task | | 1.5 | | |
| Sequence Diagram Function 2: delete task | | | | |
| Sequence Diagram Function 3: Assign task | | | | |
| Sequence Diagram Function 4: Manage users | | | | |
| Activity Diagram Function 1: Click or tap here to enter text. | | 0.5 | | |

| | | | | |
|---|---|---|---|---|
| **Activity Diagram**<br>**Function 2 :** Click or tap here to enter text. | | | | |
| **Activity Diagram**<br>**Function 3:** Click or tap here to enter text. | | | | |
| **Activity Diagram**<br>**Function 4 :**Click or tap here to enter text. | | | | |
| **State Diagram (If any)** | | **0.5** | | |
| **System Architecture** | | | | |
| **TOTAL SCORE** | | **9** | | |

# Contents

# 1. Chapter 1

System descriptions

Our system is designed to help small teams and organizations to manage tasks more efficiently by providing a set of features. it allows users to create tasks with detailed descriptions, set deadlines, assign responsibilities to team members, sends reminders that notify users of upcoming deadlines, display overview of all tasks and their status (completed or still needs work) and enable team members to discuss ideas and share files directly within the platform. Overall, TMS is a useful system for improving communication and productivity.

The development Process model used

For our Task Management System (TMS), the model that will be used is the Incremental Development model to ensure flexibility and responsiveness to user needs. This approach not only lets us get the product out faster but also ensures that it meets our team's requirements more effectively.

## System Requirements

***Functional Requirements***

*User Requirements*

1. The system shall provide user registration and login functionalities.
2. The system shall allow users to Create, Read, Update, and Delete tasks.
3. Each task shall include a title, description, due date, priority, and status.
4. The system shall send reminders to users for upcoming deadlines.
5. The system shall store user data and tasks in a persistent database.
6. The system shall provide a search feature to locate tasks by keywords or tags.
7. Users shall be able to assign tasks to specific team members.
8. Users shall be able to share tasks with other users via a sharing option.

*System Requirements*

1.The system shall provide user registration and login functionalities.

1.1 The system shall allow users to register with a unique username, email, and password.

1.2 The system shall validate user credentials during login.

1.3 The system shall store user registration data in a secure, encrypted format.

2.The system shall allow users to Create, Read, Update, and Delete tasks.

2.1 The system shall display a list of all tasks in a readable format.

2.2 The system shall allow users to update the details of an existing task.

2.3 The system shall allow users to delete tasks, with confirmation prompts to prevent accidental deletion.

3.Each task shall include a title, description, due date, priority, and status.

3.1 The system shall enforce mandatory fields (title, description, due date, priority, and status) when creating or updating a task.

3.2 The system shall provide default options for task priority (High, Medium, Low).

3.3 The system shall ensure that the task status can be marked as either "Pending" or "Completed."

3.4 The system shall validate the due date to ensure it is a future date.

4.The system shall send reminders to users for upcoming deadlines.

4.1 The system shall trigger reminder notifications at configurable intervals (24 hours, 7 days) before a task's due date.

4.2 The system shall send reminders via email and/or push notifications, depending on user preferences.

5.The system shall store user data and tasks in a persistent database.

5.1 The system shall use a relational database to store user profiles, task details, and metadata.

5.2 The system shall ensure that all data is stored with proper indexing for fast retrieval.

5.3 The system shall back up the database regularly to prevent data loss.

6.The system shall provide a search feature to locate tasks by keywords or tags.

6.1 The system shall allow users to search tasks using a search bar or filter.

6.2 The system shall support search queries based on task title, description, or tags.

6.3 The system shall return relevant search results sorted by due date or priority.

### 7.Users shall be able to assign tasks to specific team members.

7.1 The system shall allow users to select a team member when creating or updating a task.

7.2 The system shall notify the assigned team member via email or push notification when a task is assigned to them.

7.3 The system shall allow users to reassign tasks to a different team member if needed.

### 8.Users shall be able to share tasks with other users via a sharing option.

8.1 The system shall provide a "Share Task" button for each task, allowing users to share tasks with other registered users.

8.2 The system shall send a notification to the recipient when a task is shared with them.

8.3 The system shall allow users to set permissions for shared tasks, such as view-only or edit permissions.

***Non-Functional Requirements***

### User requirements:

1. The interface shall be intuitive and easy to use
2. User authentication and access control must function without failures
3. The system must handle errors and provide clear feedback to users.

### System requirements:

1.The interface shall be intuitive and easy to use

1.1 The system shall be designed to ensure that users can access any core feature, including task creation, task assignment, reminders, and file sharing, within a maximum of three clicks from the main dashboard.

1.2 The system shall provide a clear and consistent navigation structure, utilizing menus, buttons, and links that are easily identifiable and accessible.

1. 3 The system shall include tooltips or brief descriptions for each feature to assist users in understanding the functionality without requiring extensive guidance.

1. 4 The layout shall be visually organized to prioritize frequently used features, enhancing user accessibility and reducing the time taken to perform common tasks.

2. User authentication and access control must function without failures

2.1 The system shall enforce password complexity requirements, requiring a minimum length and a combination of uppercase letters, lowercase letters, numbers, and special characters.

2.2 The system shall implement account lockout policies after a specified number of failed login attempts to prevent attacks.

3. The system must handle errors and provide clear feedback to users.

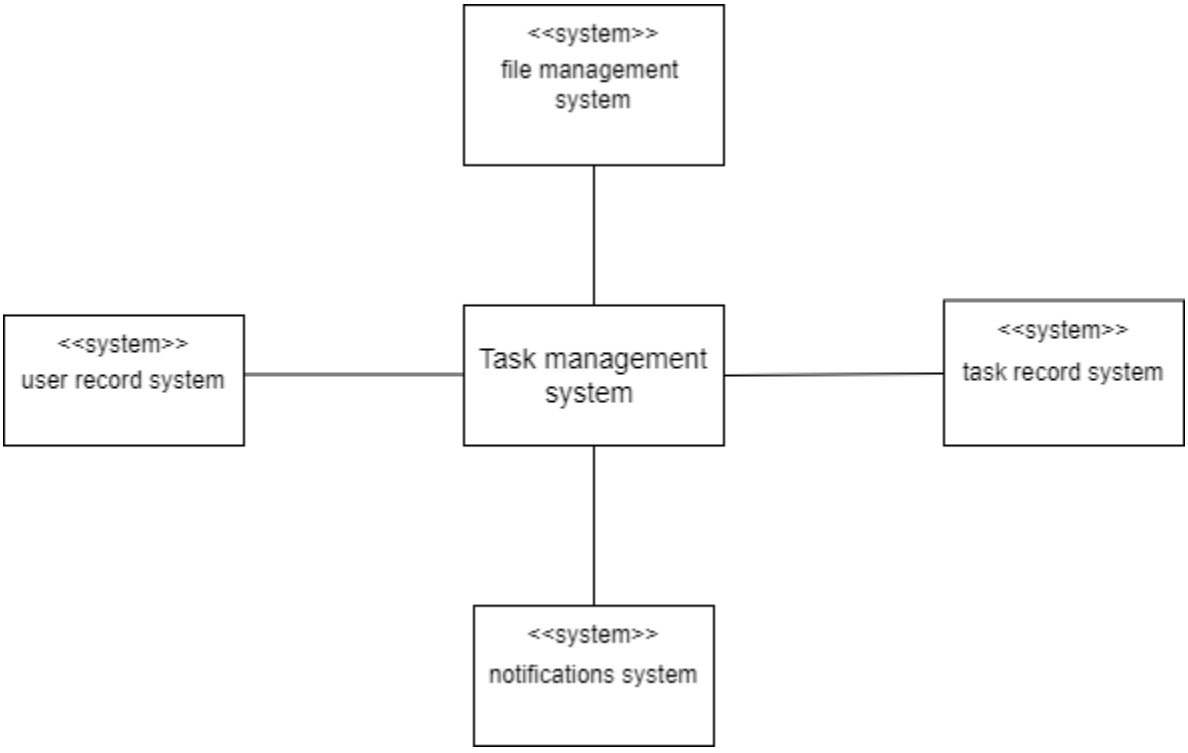
3.1 The system shall implement a comprehensive error handling mechanism to capture and log all errors that occur during user interactions and system operations.

3.2 The system shall provide clear, user-friendly error messages that explain the nature of the error and suggest corrective actions when applicable.

3.3 The system shall display error messages in a consistent format, using visual cues such as color coding to enhance visibility and understanding.

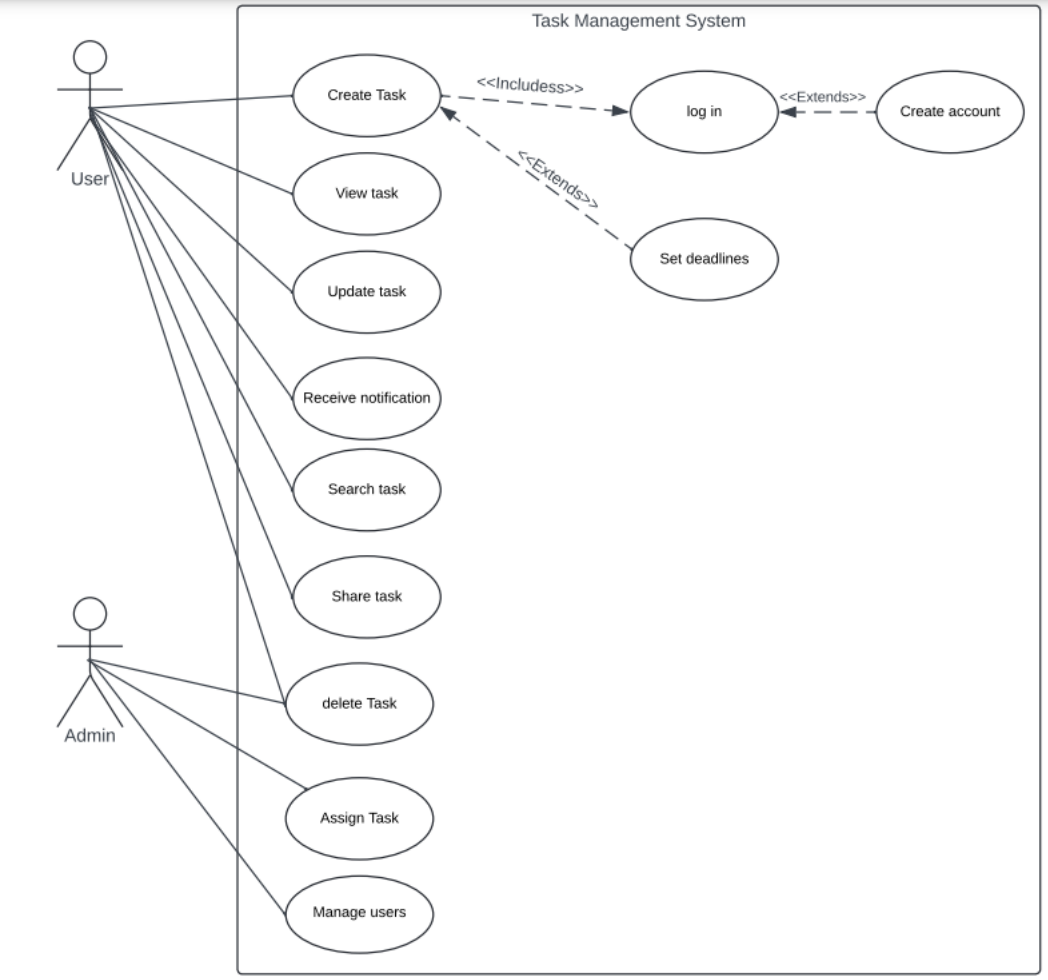
## Chapter 2

Context Diagram



## System Use-case Diagram with description

The Task Management System use-case diagram depicts interactions between User and Admin, each with distinct functions. Users can create, view, update, search, delete, and share tasks, set deadlines, and receive notifications about task updates. Admins, on the other hand, have additional privileges, such as deleting tasks, assigning tasks to users, and managing user accounts. Both users and admins must log in to access the system, with an option to create an account if they are new. This setup ensures that users can manage personal tasks efficiently while admins have control over system-wide task and user management.



Use case scenario (MSS +Extensions)

***Use case 1 Create task***

| | | |
|---|---|---|
| **Use-case Name** | Create task | |
| **Goal** | The use case allows users to create a new task | |
| **Actors** | user | |
| **Precondition** | -the user must be logged into the system.<br>-the task management system must be available and operational. | |
| **Main Success Scenario** | **1** | the user logs into the system. |
| | **2** | the user access to the "create task" interface. |
| | **3** | the user inputs task details. |
| | **4** | optionally the user can set a deadline for the task. |
| | **5** | the system confirms task creation. |
| **Postconditions** | -the new task is created and stored in the system.<br>-the new task is ready to be viewed, updated, assigned, or deleted. | |
| **Extension** | **1.a** | The user attempts to log in but enters incorrect credentials. |
| | **3.a** | The user enters task details but forgets to fill in a required field |
| | **4.a** | The user tries to set a deadline but selects a date in the past. |

***Use case 2 View task***

| | | |
|---|---|---|
| **Use-case Name** | View task | |
| **Goal** | The use case allows users to view details of a task. | |
| **Actors** | User | |
| **Precondition** | -the user must be logged into the system.<br>-the task must exist in the system and accessible to the logged-in user.<br>-the task management system must be operational. | |
| **Main Success Scenario** | **1** | the user logs into the system. |
| | **2** | the user access to the "view tasks "section. |
| | **3** | the user selects a specific task from the list. |
| | **4** | system displays the task details. |
| **Postconditions** | -the user views the details of the selected task. | |
| **Extension** | **2.a** | The section takes too long to load, causing frustration. |
| | **3.a** | The user attempts to select a task from the list, but the list is empty or does not display the expected tasks. |
| | **4.a** | If the user manages to select a task but encounters a technical issue the task details do not load. |

***Use case 3 Update task***

| | | |
|---|---|---|
| **Use-case Name** | Update task | |
| **Goal** | The use case allows users to modify the details of an existing task. | |
| **Actors** | User | |
| **Precondition** | -the user must be logged into the system.<br>-the task to be updated must exist and accessible to the logged-in user. | |
| **Main Success Scenario** | **1** | the user access to the "update task" section. |
| | **2** | the user selects the task they want to update. |
| | **3** | the user updates the task information (change title, description, status, deadline). |
| | **4** | the system saves the updated task information. |
| **Postconditions** | -the task is updated with the new information<br>-the system saves the changes. | |
| **Extension** | **2.a** | The user selects a task, but the task list is empty or does not display the expected tasks. |
| | **3.a** | If the user manages to find a task but tries to change the title to an empty string, the system does not allow it. |
| | **4.a** | The system displays an error message, and the updates are not saved. |

***Use case 4 Receive notification***

| | | |
|---|---|---|
| **Use-case Name** | Receive notification | |
| **Goal** | The use case allows users to receive notification alert related to the task. | |
| **Actors** | user | |
| **Precondition** | The use case allows users to receive notification alert related to the task. | |
| **Main Success Scenario** | **1** | when an event occurs, the system generates a notification. |
| | **2** | the system sends notification to the relevant users. |
| | **3** | the notification is delivered to the user. |
| | **4** | the user views the notification and takes necessary action. |
| **Postconditions** | -the user receives the notification about the relevant task event.<br>-the notification is displayed in the system. | |
| **Extension** | **1.a** | An event occurs, but there's a failure in the notification generation process. |
| | **2.a** | The system attempts to send notifications, but there's a problem with the user list. |
| | **3.a** | If the notification is sent, it might be delayed due to network issues or server overload. |

***Use case 5 Search task***

| | | |
|---|---|---|
| **Use-case Name** | Search task | |
| **Goal** | To enable users to efficiently search for specific tasks within the task management system based on various criteria, such as title, description, status, or deadline. | |
| **Actors** | User | |
| **Precondition** | - The user must be logged into the system.<br>- The user must have access to the "Search Task" feature.<br>- The task database must contain existing tasks. | |
| **Main Success Scenario** | **1** | User Accesses Search Task Feature. |
| | **2** | User Initiates Search. |
| | **3** | System Displays Results |
| | **4** | User Views Task Details |
| **Postconditions** | - The user receives a list of tasks that match the search criteria.<br>- The user can view details of the tasks that are displayed in the search results.<br>- Any relevant filters or search parameters used by the user are maintained for future searches. | |
| **Extension** | **1.a** | Access Denied |
| | **2.a** | Invalid Search Criteria |
| | **3.a** | The system displays an error message. |

***Use case 6 Share task***

| | | |
|---|---|---|
| **Use-case Name** | Share task | |
| **Goal** | To enable users to share tasks with other users within the task management system, allowing for collaboration and improved task visibility. | |
| **Actors** | User | |
| **Precondition** | - The user must be logged into the system.<br>- The user must have permission to share tasks.<br>- The task must already exist in the system. | |
| **Main Success Scenario** | **1** | User Accesses Share Task Feature |
| | **2** | User Selects Share Option. |
| | **3** | User Confirms Sharing. |
| | **4** | System Processes Sharing |
| | **5** | System Confirms Sharing. |
| **Postconditions** | - The selected task is successfully shared with the specified user(s).<br>- The shared user(s) receive a notification about the shared task.<br>- The task's visibility and access permissions are updated to reflect the sharing. | |
| **Extension** | **1.a** | User Lacks Permission. |
| | **2.b** | Access Denied on Acceptance. |

***Use case 7 Delete task***

| **Use-case Name** | Delete task | |
|---|---|---|
| **Goal** | Allows admin and users to remove a task from the system | |
| **Actors** | Admin, user | |
| **Precondition** | -Admin or user must be logged in.<br>-The task to be deleted must exist | |
| **Main Success Scenario** | 1 | The admin or user selects the task they wish to delete |
| | 2 | The system displays the task details for confirmation |
| | 3 | The admin or user confirms the deletion |
| | 4 | The system removes the task from the database |
| | 5 | The system displays a conformation massage indicating successful deletion |
| **Postconditions** | -The task is removed from the system<br>-Any related notifications or dependencies are updated | |
| **Extension** | 2.a | If the admin or user does not have permission to delete the task the system displays an error message and does not proceed with the deletion |
| | **3.a** | **If the admin or user decides not to delete the task the system cancel the deletion and returns to the task list** |
| | **4.a** | **If the system encounters an error while deleting the task it displays an error message and rolls back the changes** |

***Use case 8 Assign task***

| **Use-case Name** | Assign task | |
|---|---|---|
| **Goal** | Allows the admin to assign tasks to teams within the system | |
| **Actors** | admin | |
| **Precondition** | -Admin must be logged in.<br>-The task must exist.<br>-The team to be assigned must exist and be active. | |
| **Main Success Scenario** | 1 | The admin selects a task that needs to be assigned to a team |
| | 2 | The system displays a list of available teams who can be assigned the task |
| | 3 | The admin selects the appropriate team from the list |

| | | |
|---|---|---|
| | 4 | The system assigns the task to the team selected and updates the task management system |
| | 5 | The system sends a notification to the team about the new task |
| | 6 | The team receives the notification, and the task is added to their task list |
| **Postconditions** | The task is assigned to the specified team.<br>-The assignee receives a notification | |
| **Extension** | 2.a | If there no available teams to assign the task to the system displays a massage indicating that no teams currently available |
| | 4.a | If the system encounters an error while assigning the task it displays an error message and rolls back the changes |

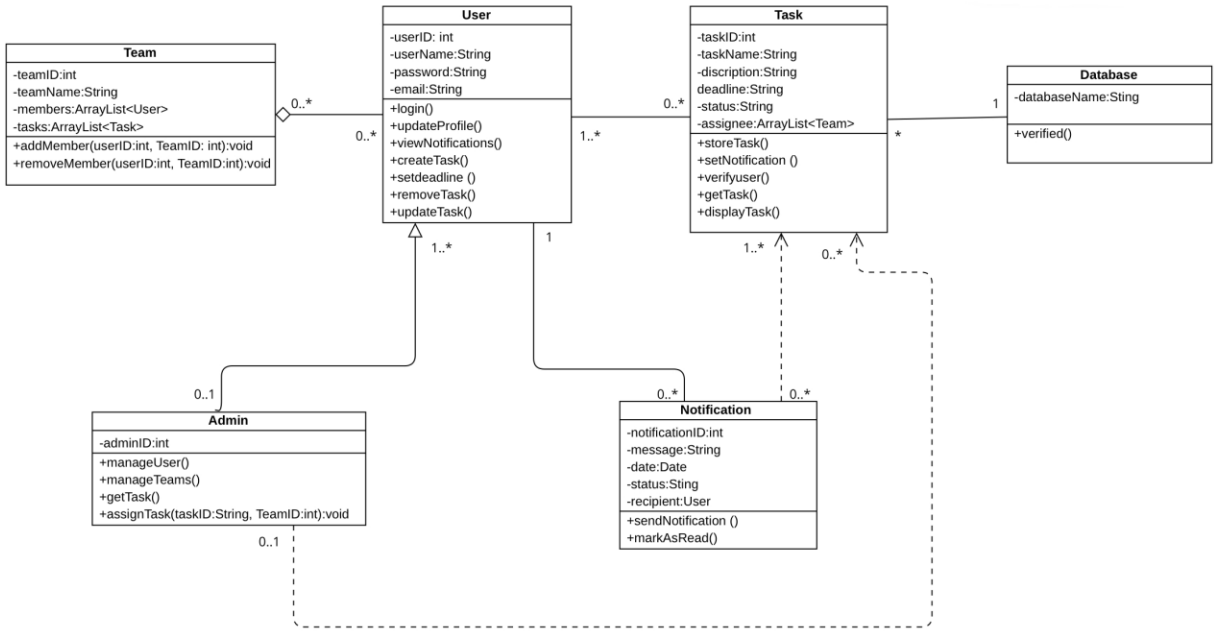
***Use case 9 Manage users***

| | | |
|---|---|---|
| **Use-case Name** | Manage users | |
| **Goal** | Allows the admin to manage users accounts | |
| **Actors** | admin | |
| **Precondition** | -Admin must be logged in | |
| **Main Success Scenario** | 1 | The user selects the manage users option |
| | 2 | The system displays a list of the users in the system |
| | 3 | The admin reviews the user details and can perform the following actions: 1. Add new user<br>1. Delete user<br>2. Edite the details of the user |
| | 4 | The admin makes the desired changes |
| | 5 | The system updates the user record |
| | 6 | The system provides a confirmation message |
| **Postconditions** | -User records are updated according to the action taken | |

| | | |
|---|---|---|
| | -The user list reflects any modifications. | |
| **Extension** | 3.a | If the admin wants to add new user the system provides a form for the necessary user details such as name, email |
| | 5.a | If the system encounters error while updating the user record it displays an error massage and rolls back the changes |

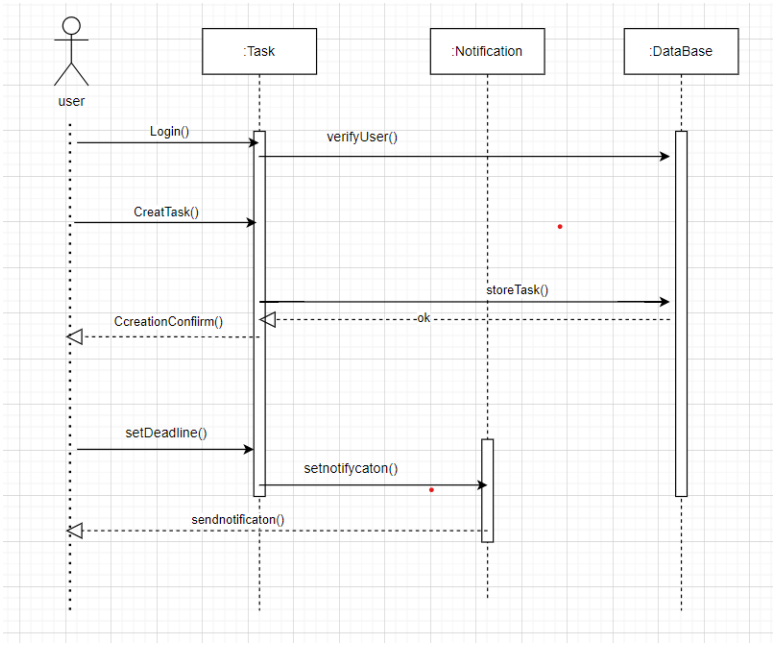
# 3. Chapter 3

System Class Diagram

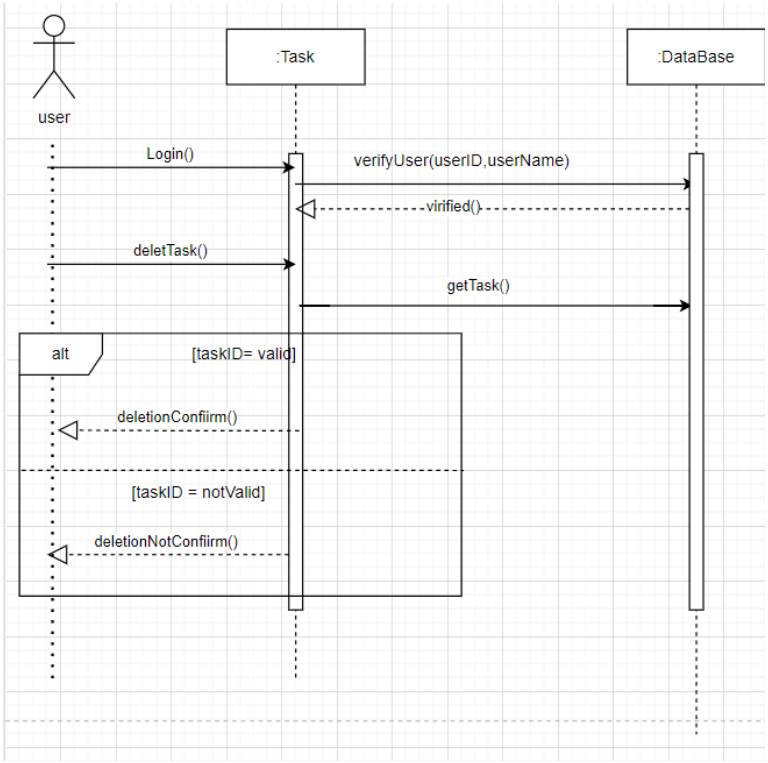


# Sequence Diagrams

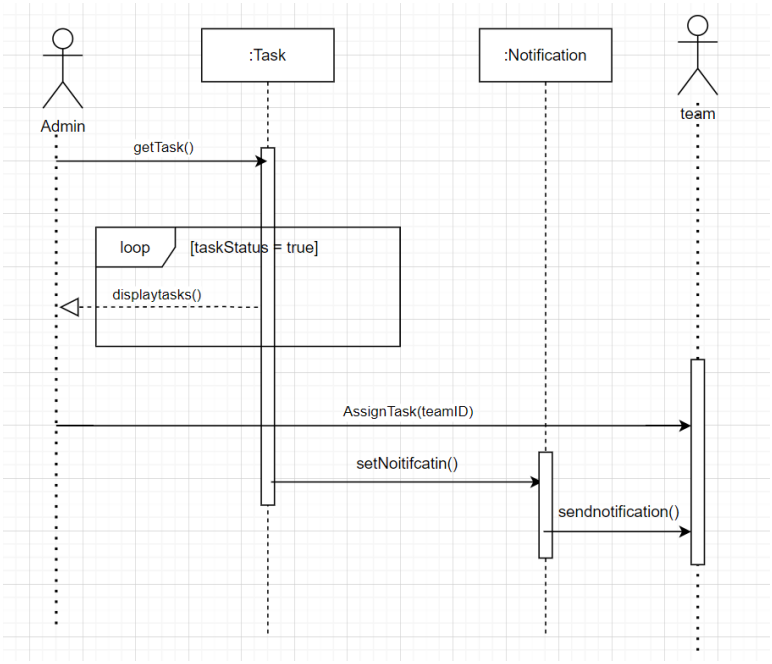
## *Sequence diagram for Create task*



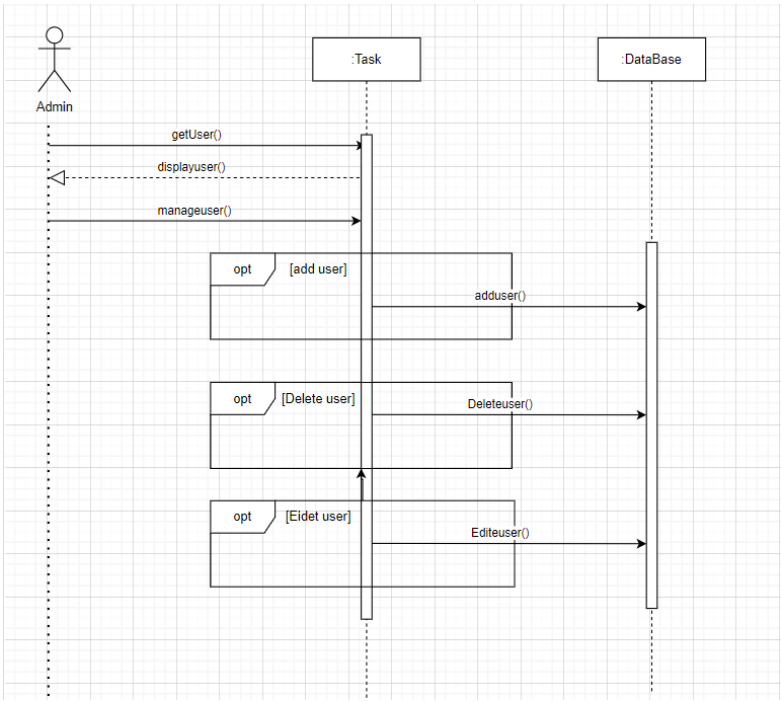

## *Sequence diagram for Delete Task*



## *Sequence diagram for Assign Task*



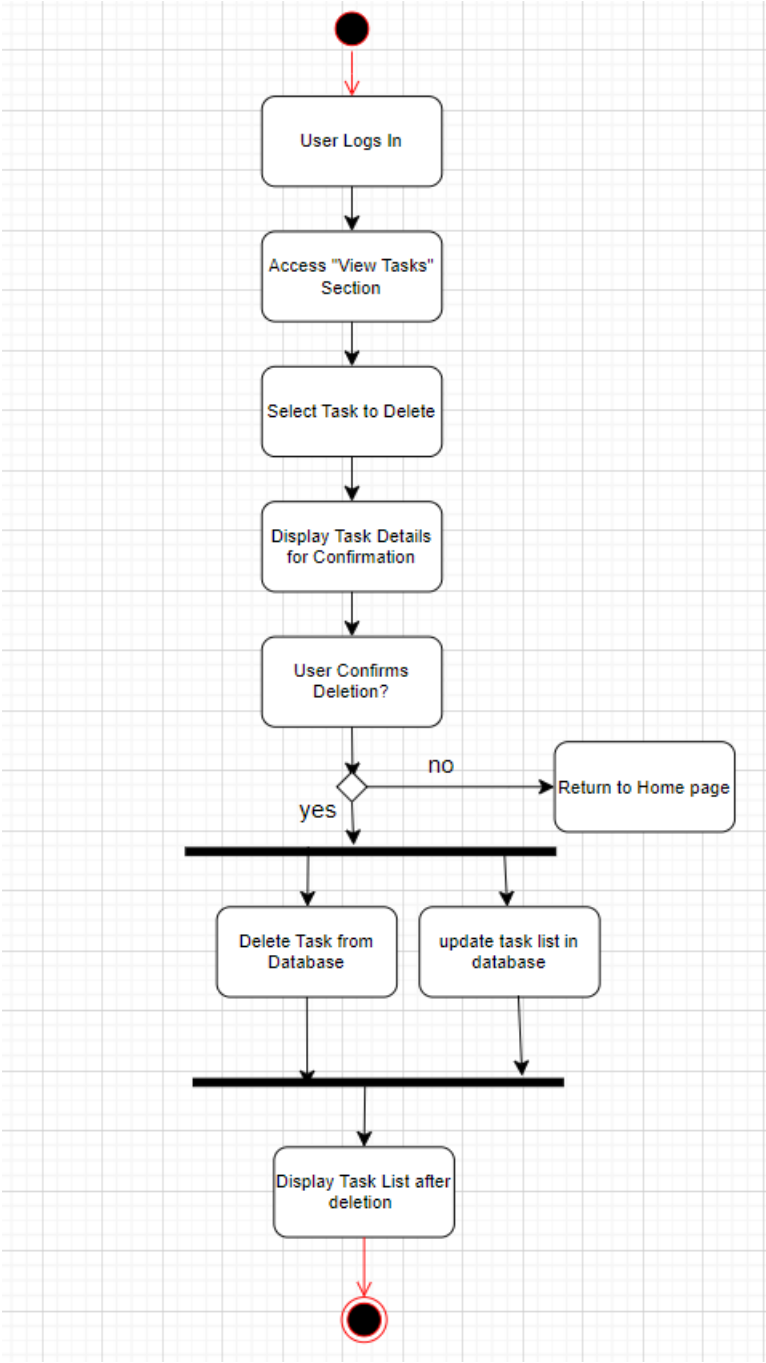

## *Sequence diagram for Manage Users*



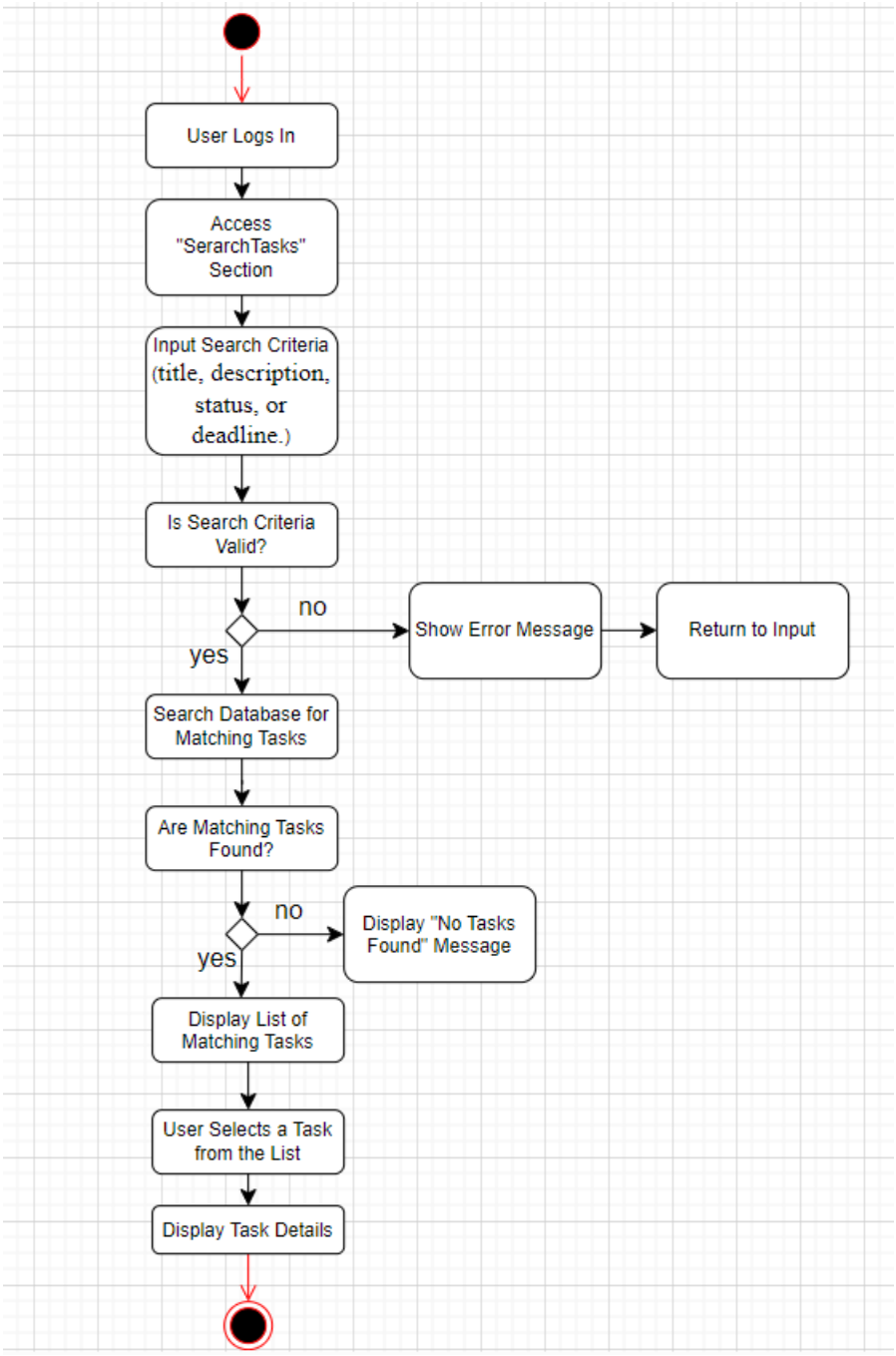
# 4. Chapter 4

## Activity Diagram

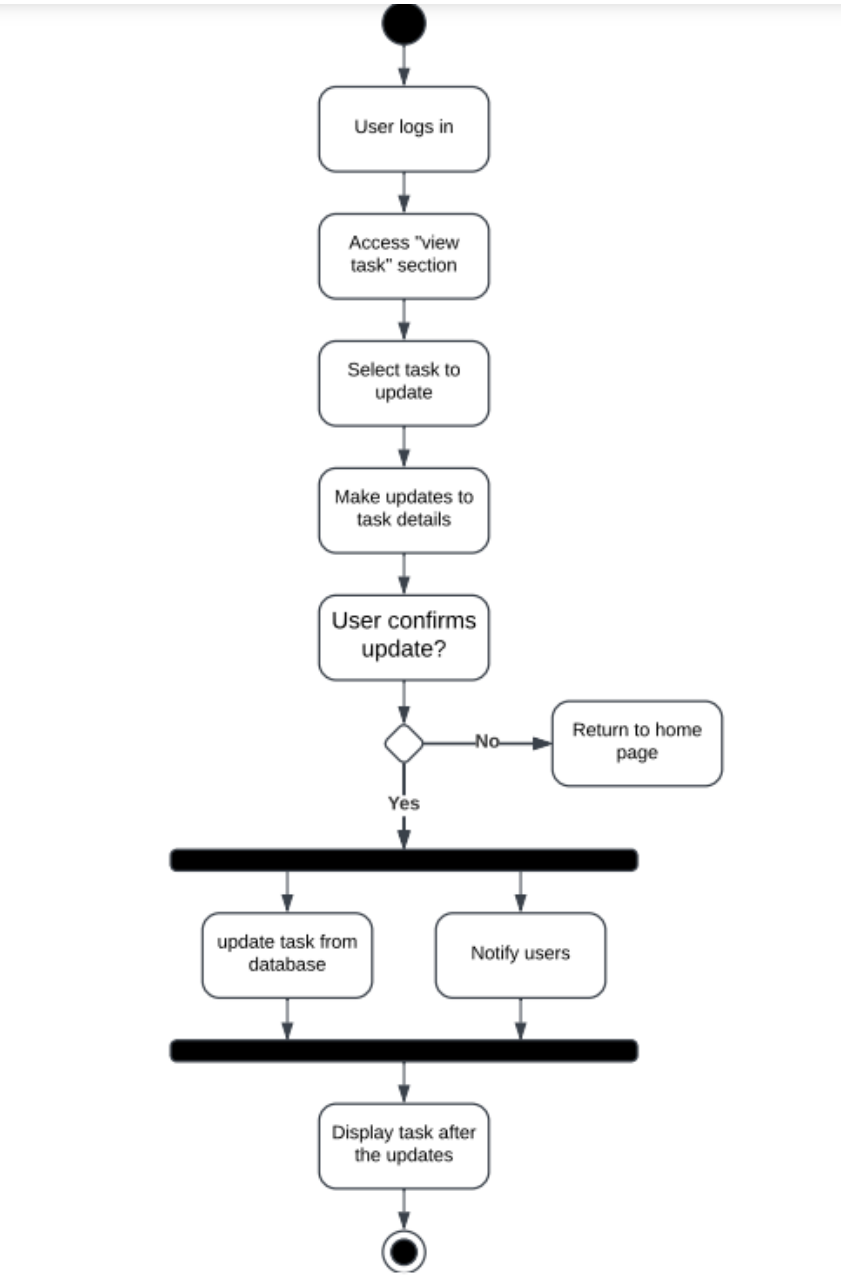
***Activity Diagram for Delete a task***



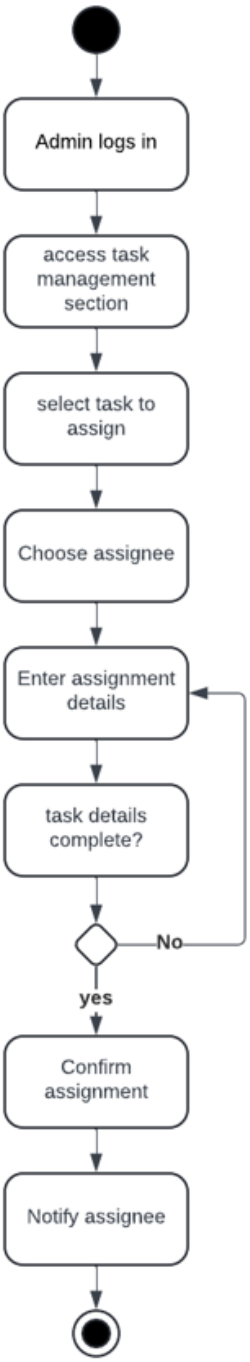
***Activity Diagram for Search a task***



***Activity Diagram for Update a task***

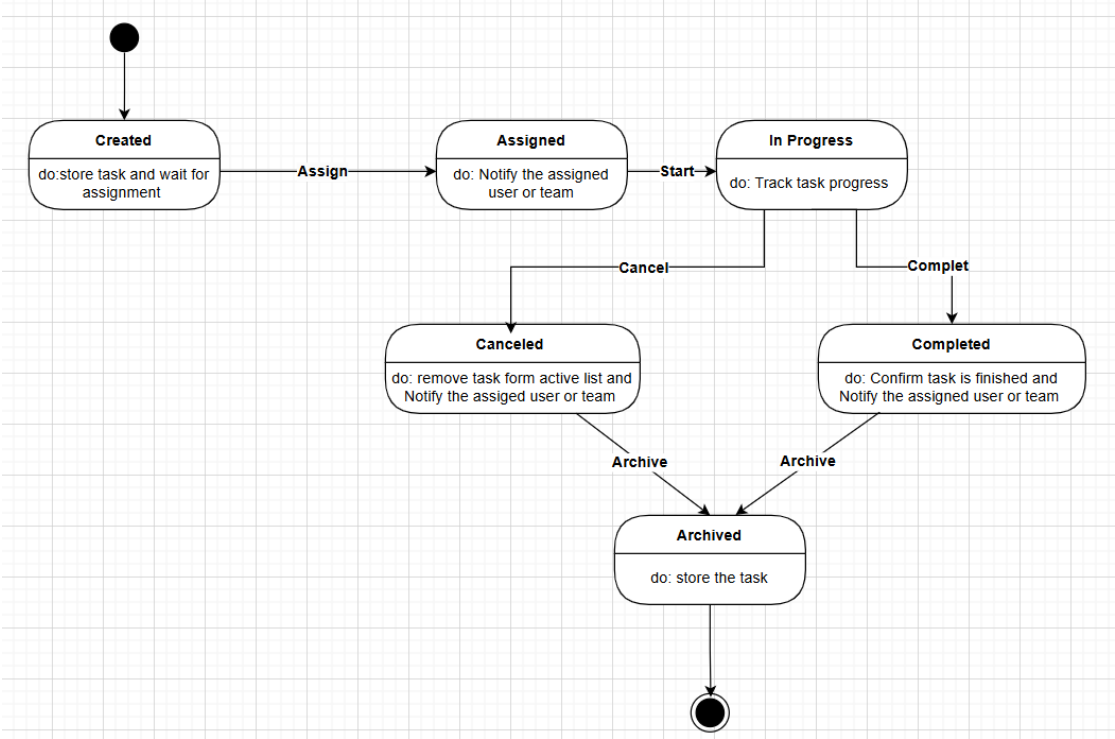


***Activity Diagram for Assign a task***



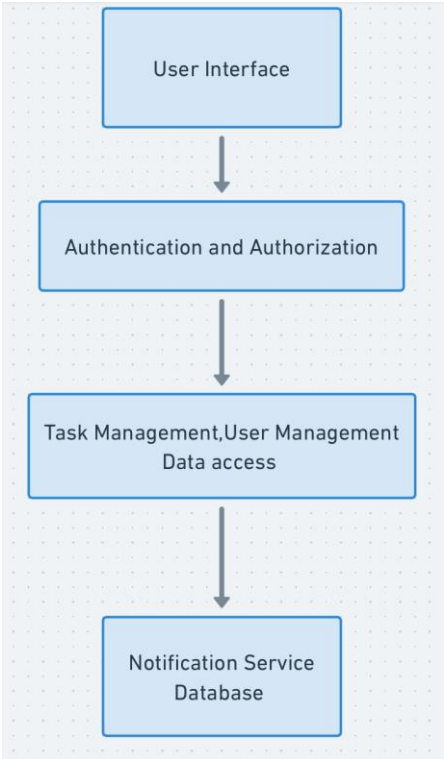
State Diagram

***State Diagram for Task***




# 5. Chapter 5

Architecture Diagram




We chose the layer architecture model since it clearly separates components from each other, and hence is more maintainable or easier to update. Given this structure, such critical job processes as authentication need to be safely handled before core functionality is accessed for user data protection. This model also supports high traffic and allows independent scaling of parts of the system. Finally, the modular architecture allows users to access the system at any time without affecting other layers or causing downtime.