Project Title: Task Management System (Console Application)

Project Overview: The semester-long project for this course is designed to provide you with hands-on experience in software development using C# and UML (Unified Modeling Language) concepts. You will work in teams to design, implement, and test a Task Management System as a console application. This project will require you to apply the knowledge and skills you've gained throughout the course in areas such as UML diagrams, software architecture, object-oriented programming, and console-based user interfaces.

Project Objectives:

1. Apply UML concepts to design the architecture of a Task Management System.
2. Implement the system using C# as a console application.
3. Develop a user-friendly, text-based interface for the Task Management System.
4. Implement core functionality for creating, assigning, and tracking tasks, as well as generating reports.
5. Perform integration testing and system testing to ensure the software's quality.

Use-Cases: The following detailed use-cases are part of the project, including use-cases for team members:

Use-Case 1: Sign-Up (User)

- **Primary Actor:** User
- **Description:** This use-case allows a user to sign up for the Task Management System by providing their desired username and password.
- **Basic Flow:**
    1. User selects the "Sign-Up" option.
    2. The system prompts for a username and password.
    3. User enters their chosen credentials.
    4. The system creates a new user account.

Use-Case 2: Sign-In (User)

- **Primary Actor:** User
- **Description:** This use-case allows a signed-up user to sign in to the Task Management System by providing their username and password.
- **Basic Flow:**
    1. User selects the "Sign-In" option.
    2. The system prompts for the user's credentials.

3. User enters their username and password.
4. The system validates the credentials.
5. If the credentials are correct, the user is signed in to the system.

Use-Case 3: Sign-Up (Manager)

- **Primary Actor:** Manager
- **Description:** This use-case allows a manager to sign up for the Task Management System by providing their desired username and password.
- **Basic Flow:**
    1. Manager selects the "Sign-Up" option.
    2. The system prompts for a username and password.
    3. Manager enters their chosen credentials.
    4. The system creates a new manager account.

Use-Case 4: Sign-In (Manager)

- **Primary Actor:** Manager
- **Description:** This use-case allows a signed-up manager to sign in to the Task Management System by providing their username and password.
- **Basic Flow:**
    1. Manager selects the "Sign-In" option.
    2. The system prompts for the manager's credentials.
    3. Manager enters their username and password.
    4. The system validates the credentials.
    5. If the credentials are correct, the manager is signed in to the system.

Use-Case 5: Create Task (User)

- **Primary Actor:** User
- **Description:** This use-case allows a user to create a new task by providing details such as task title, description, and due date. Note: the task has three status: it's either **to-do**, which means created but not assigned to any user, **pending**, which means task has been created and assigned to user, or **done**, which means the task has been completed by a the assigned user.
- **Basic Flow:**
    1. User selects the "Create Task" option.

2. The system prompts for task information.
3. User enters the required details.
4. The system creates a new task in the task list.

Use-Case 6: Assign Task (User)

- **Primary Actor:** User
- **Description:** This use-case enables users to assign a task to a specific team member or user.
- **Basic Flow:**
    1. User selects the "Assign Task" option.
    2. The system lists available tasks.
    3. User selects a task and specifies the assignee.
    4. The system updates the task's assignee.

Use-Case 7: Track Task Progress (User)

- **Primary Actor:** User
- **Description:** This use-case allows users to update the status and progress of a task. The user can update the progress by providing text description (dated) for each new activity they have made. So, I can user can do multiple updates (text description) per task, each update should be dated upon entry.
- **Basic Flow:**
    1. User selects the "Track Task Progress" option.
    2. The system lists tasks assigned to the user.
    3. User selects a task and updates its status and progress.
    4. The system records the changes.

Use-Case 8: Generate Task Reports (Manager)

- **Primary Actor:** Manager
- **Description:** This use-case allows managers to generate reports about task status, completion, and user performance.
- **Basic Flow:**
    1. Manager selects the "Generate Task Reports" option.
    2. The system presents options for different types of reports.
    3. Manager selects a report type and specifies criteria.

4. The system generates and displays the report.

Use-Case 9: Assign User to A team (Manager)

- **Primary Actor:** Manager
- **Description:** This use-case allows managers Assign users to his team.
- **Basic Flow:**
    1. The manager create a team and give that team a name and a special id.
    2. Manager can view all available users that are not in a team yet.
    3. Select user or more to join their team.

Use-Case 9: Team Member Task View (Team Member)

- **Primary Actor:** Team Member
- **Description:** This use-case allows team members to view tasks assigned to them, including details, due dates, and status.
- **Basic Flow:**
    1. Team Member selects the "Team Member Task View" option.
    2. The system lists tasks assigned to the team members.
    3. Team Member can view task details and their status.

Use-Case 10: Team Member Task Update (Team Member)

- **Primary Actor:** Team Member
- **Description:** This use-case allows team members to update the status and progress of tasks assigned to them.
- **Basic Flow:**
    1. Team Member selects the "Team Member Task Update" option.
    2. The system lists tasks assigned to the team members.
    3. Team Member selects a task and updates its status and progress.
    4. The system records the changes.

Deliverables:

1. UML diagrams.

2. Source code and documentation for the Task Management System as a console application.
3. Test and reports documenting the results of testing.
4. A final project report summarizing the project's design, development process, challenges, and solutions.

Grading: Your project will be evaluated based on the quality of your UML diagrams, the completeness and functionality of your Task Management System, and the documentation you provide. You will also receive feedback on your teamwork, problem-solving skills, and coding practices.

Project Timeline: This is a semester-long project, and the following timeline provides a rough breakdown of the project into milestones and deadlines:

- **Project Kick-off:** Introduction to the project and team formation.
- **UML Design:** Create and submit UML diagrams for approval.
- **Development:** Implement the system, including user interface and core functionality.
- **Testing and Debugging:** Perform testing and address any issues.
- **Documentation and Final Submission:** Submit all project documentation and the final console application.