

To-Do List App

A basic to-do list application that allows users to create, manage, and track their tasks or to-do items. Users can add tasks, mark them as complete, set reminders and delete tasks.

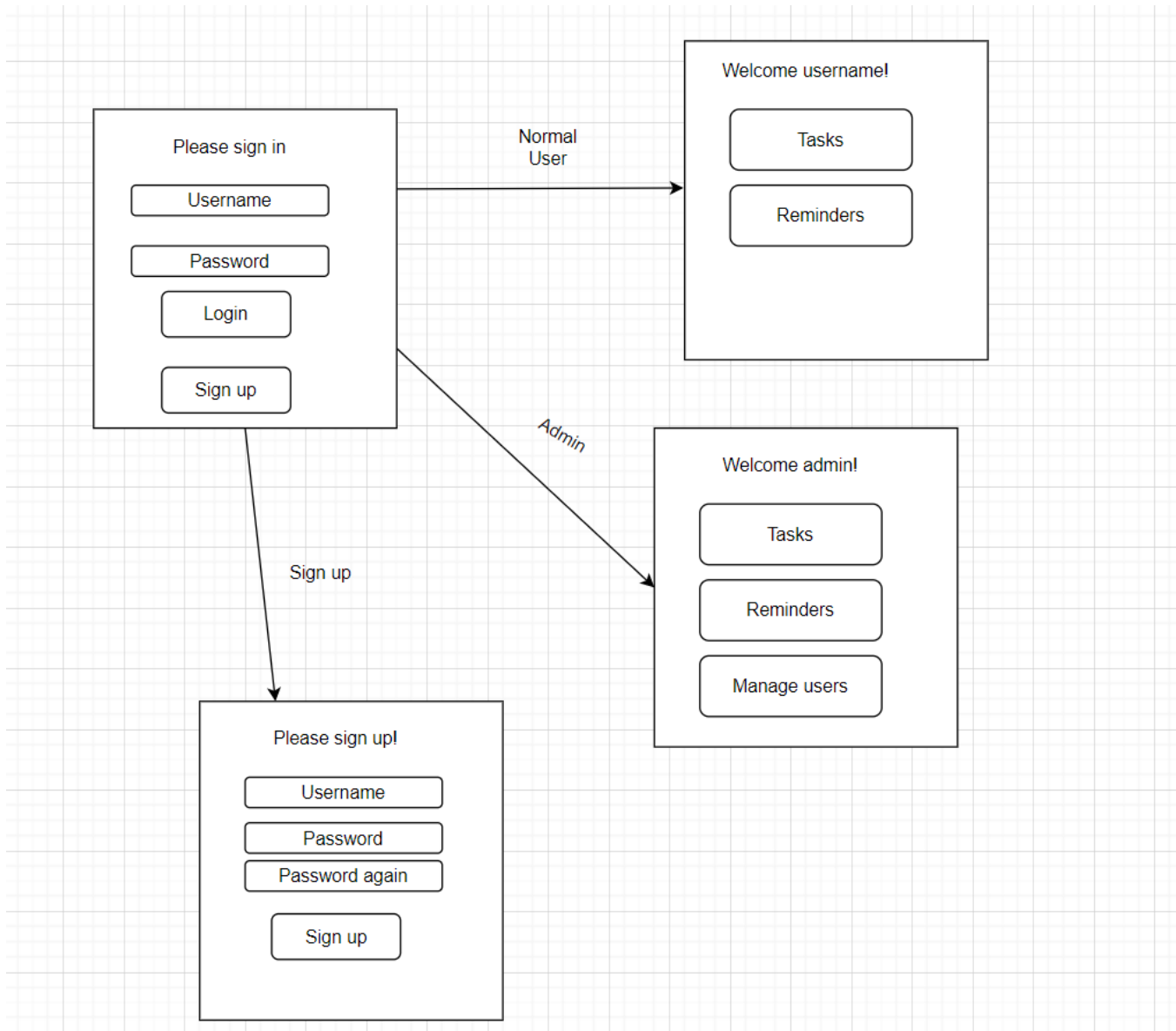
Github: https://github.com/AlexCsumb/cst338_todoApp

Table of contents

Initial Layout	2
Use Case Model	3
Entity Relationship Diagram (ERD)	4
Use Case 01: Predefined Users	5
Use Case 02: Persistence	6
Use Case 03: Add a user	7
Use Case 04: Delete a user	8
Use Case 05: Something really cool	9
Use Case 06:	10
Use Case 07:	11
Use Case 08:	12
Use Case 09	13
Use Case 10	14
Use Case 11	15
Use Case 12	16

Initial Layout

Include a layout similar to the one shown below. This can be created using screenshots from Android Studio, using [Draw.io](https://draw.io), or even sketched out on paper (or a tablet if you are fortunate enough to have one).

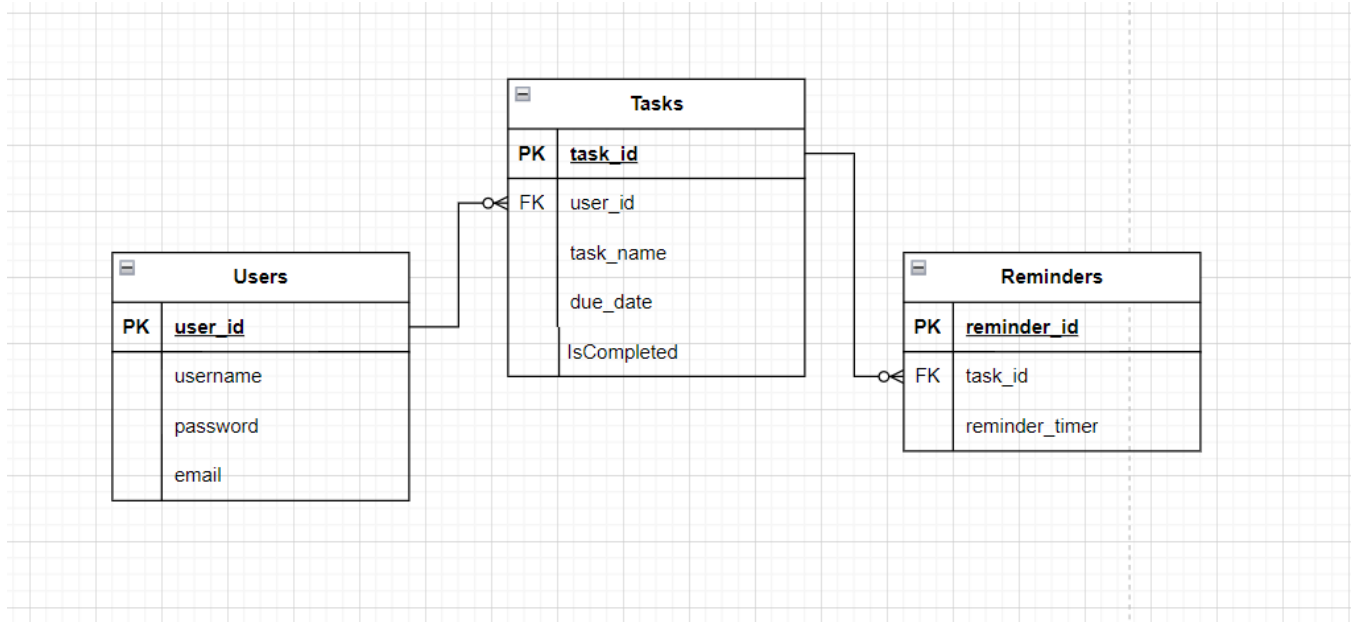


Use Case Model

Insert your actor diagrams here. It should show the users and their roles and the use cases they complete. This does not need to be 100% comprehensive but I'd like to see at least 2 actors with three actions each.

Entity Relationship Diagram (ERD)

The assignment MUST use a database and must have at least three tables



Use Case 01: Predefined Users

1. User force quits the application.
2. User opens the application.
3. User selects the option to login.
4. User enters the username "testuser1" and logs in.
5. System verifies the credentials and displays the username "testuser1" in the user interface.
6. User selects the option to logout.
7. System logs out the user "testuser1" and clears the session.
8. User selects the option to login again.
9. User enters the username "admin2" and logs in.
10. System verifies the credentials and displays the username "admin2" in the user interface.
11. System displays a 'manage users' button, indicating that the user "admin2" has admin privileges.
12. Use case ends.

Alternate Flow:

At any point in the main flow, if the user enters an incorrect username or password, the system displays an error message and the use case fails.

At step 6, if the system fails to log out the user "testuser1" or clear the session, the use case fails.

At step 10, if the system fails to verify the credentials or display the username "admin2" in the user interface, the use case fails.

At step 11, if the system fails to display the specific feature or UI element indicating admin privileges for the user "admin2", the use case fails.

This use case passes if all of these conditions are met. It fails otherwise.

Use Case 02: Persistence

1. User adds an item to the database, such as a task or a reminder.
2. User force quits the application.
3. User reopens the application.
4. System retrieves the item added in step 1 from the database and displays it in the user interface.
5. User changes an item in the database, such as updating the task status or modifying a reminder.
6. User force quits the application.
7. User reopens the application.
8. System retrieves the item modifications from step 5 from the database and displays them in the user interface.
9. Use case ends.

Alternate Flow:

At any point in the main flow, if the database connection fails or encounters an error, the system displays an error message and the use case fails.

At steps 4 and 8, if the system fails to retrieve the items from the database or display them in the user interface, the use case fails.

Use Case 03: Add a user

1. Admin selects the option to add a new user.
2. System prompts the admin to enter the details of the new user, such as username, password, email, and other required information.
3. Admin enters the required information for the new user.
4. Admin submits the information to the system.
5. System validates the entered information, such as checking for username availability, password strength, and email format.
6. If the validation fails, the system displays an error message and prompts the admin to correct the entered information.
7. If the validation passes, the system creates a new user account with the entered information and stores it in the user database.
8. System displays a success message indicating that the new user has been added.
9. Use case ends.

At any point in the main flow, if the admin cancels the process, the use case ends without adding a new user.

At step 5, if the entered information fails validation, the system displays an error message and prompts the admin to correct the entered information. The use case continues from step 4.

At step 7, if the system fails to create a new user account or store it in the database due to an error, the system displays an error message and the use case fails.

Use Case 04: Delete a user

1. Admin selects the option to delete a user.
2. System displays a list of existing users for the admin to choose from.
3. Admin selects the user account to be deleted.
4. System prompts the admin to confirm the deletion.
5. Admin confirms the deletion.
6. System removes the selected user account from the user database.
7. System displays a success message indicating that the user account has been deleted.
8. Use case ends.

At any point in the main flow, if the admin cancels the deletion process, the use case ends without deleting the user account.

At step 4, if the admin cancels the deletion confirmation, the use case returns to step 3, allowing the admin to choose a different user account to delete.

At step 6, if the system fails to remove the selected user account from the database due to an error, the system displays an error message and the use case fails.

Use Case 05: Something really cool

1. User selects the "Quote of the Day" option from the application menu.
2. System fetches a random quote from a predefined list of quotes or from an external API.
3. System displays the quote prominently on the application screen, along with the author and a visually appealing background.
4. User reads and reflects on the quote, finding inspiration or motivation.
5. User has the option to share the quote on social media, save it to favorites, or dismiss it.
6. Use case ends.

Alternate Flow:

If the external API used to fetch quotes is unavailable, the system can display a default quote or show an error message to the user.

Depending on the requirements, the system could allow the user to customize the type of quotes (e.g., motivational, inspirational, funny) or set a specific time to receive the quote daily as a notification.

Use Case 06:

Use Case description

Use Case 07:

[Use Case description]

Use Case 08:

Use Case description

Use Case 09

Use Case description

Use Case 10

Use Case description

Use Case 11

Use Case description

Use Case 12

Use Case description