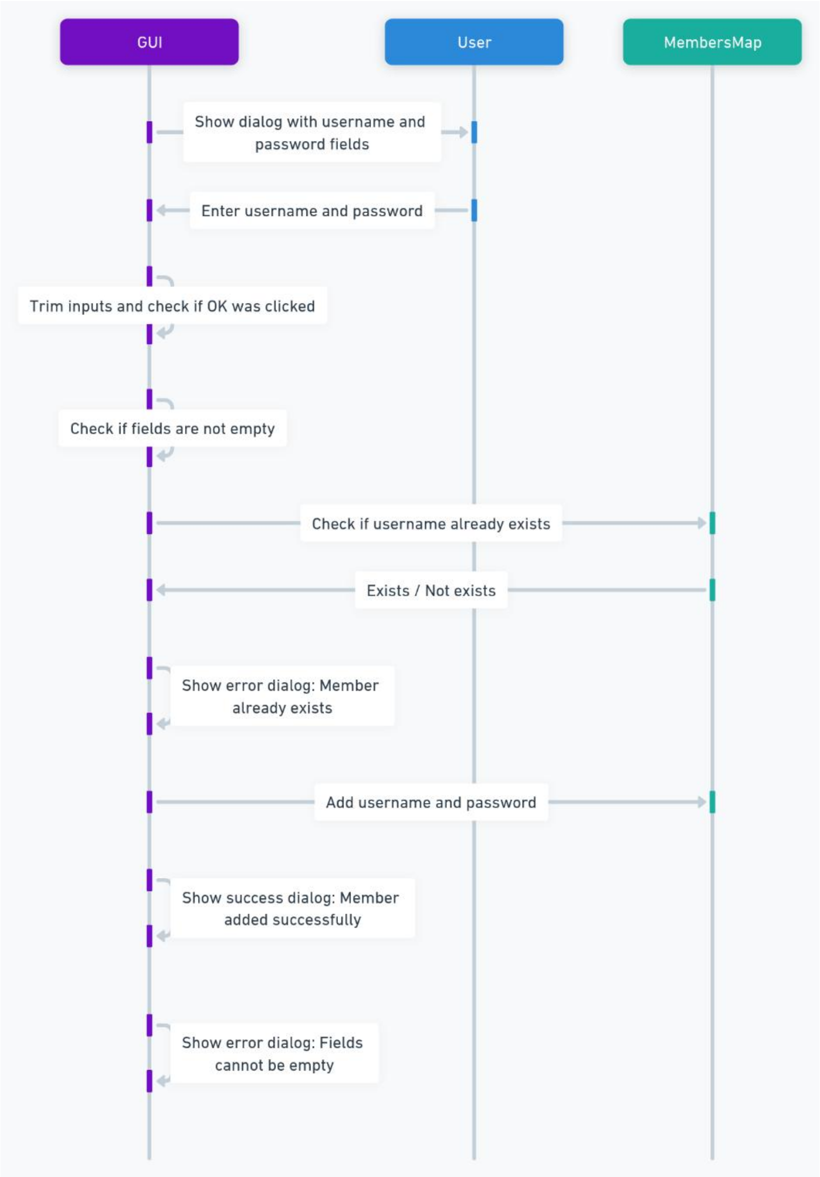
Usecase 1: Add Member

Sequence diagram

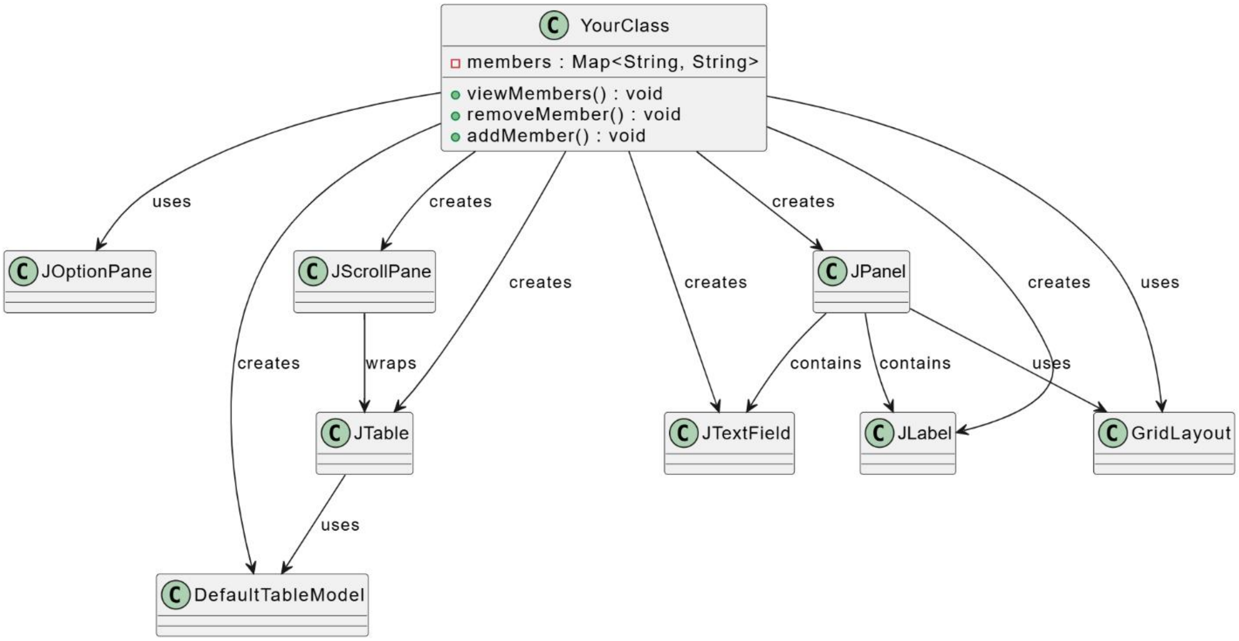


GUI
User
MembersMap
Show dialog with username and password fields
Enter username and password
Trim inputs and check if OK was clicked
Check if fields are not empty
Check if username already exists
Exists / Not exists
Show error dialog: Member already exists
Add username and password
Show success dialog: Member added successfully
Show error dialog: Fields cannot be empty

Fully dressed usecase

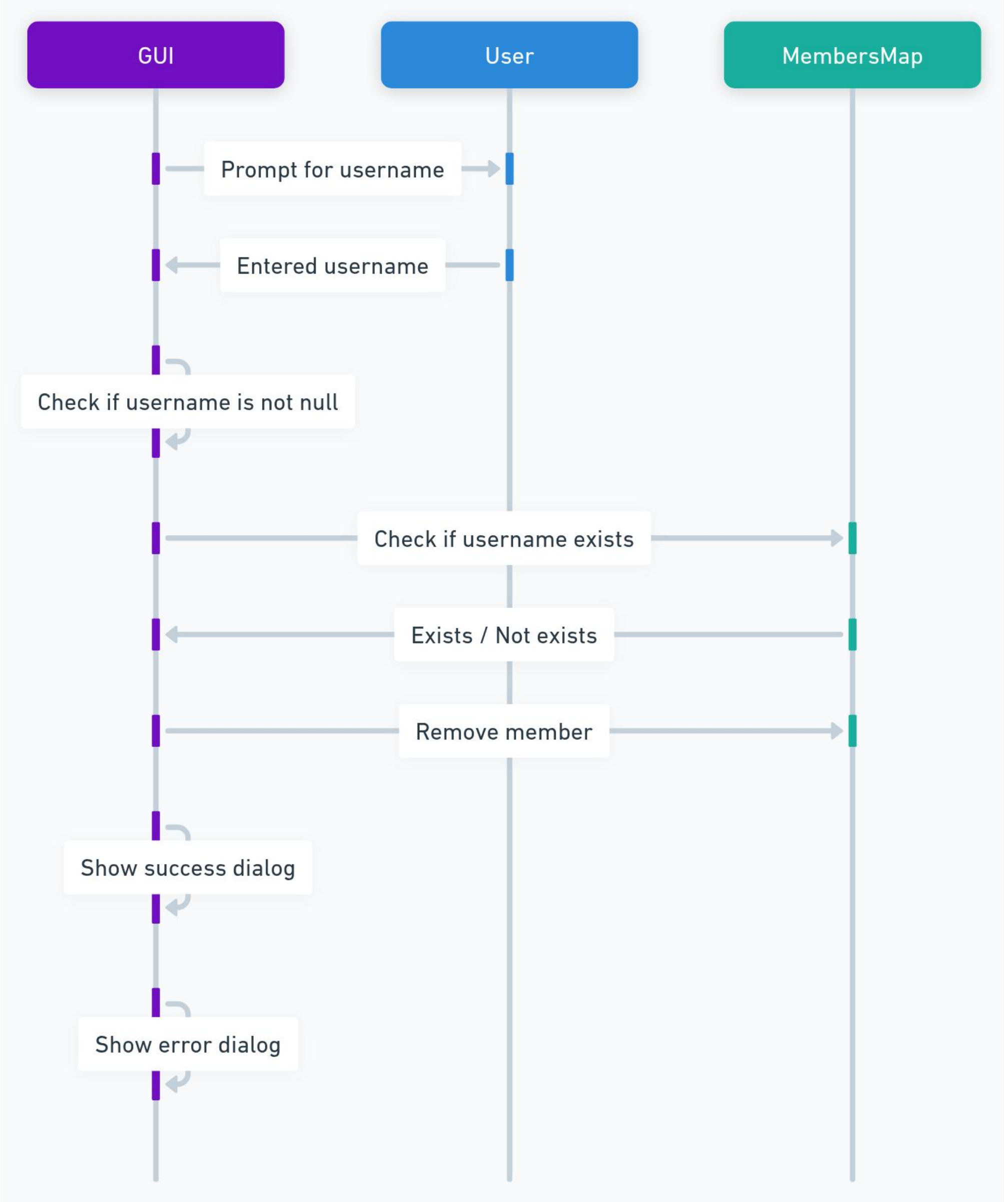
Use Case Element	Details
Use Case ID	UC-MM-01
Use Case Name	Add Member
Primary Actor	Admin
Secondary Actors	None
Brief Description	This use case allows an admin to add a new member by entering a username and password.
Preconditions	Admin is logged in successfully.- “Manage Member” interface is displayed.
Postconditions	A new member is added to the system (in-memory HashMap).
Main Success Scenario	1. Admin clicks "Add Member". 2. System prompts for username and password. 3. Admin enters valid input. 4. System checks for duplicate username. 5. If unique, system adds member. 1. 6. System shows success message.
Extensions (Alternate Flows)	**3a. Empty Fields:** 3a1. System displays "Fields cannot be empty".- **4a. Duplicate Username:** 4a1. System displays "Member already exists".
Special Requirements	Member credentials are stored temporarily in-memory. - Password is visible as plain text.
Frequency of Use	Frequent during member registration phase.
Open Issues	No encryption for password. - No data persistence after app closes.

Class diagram

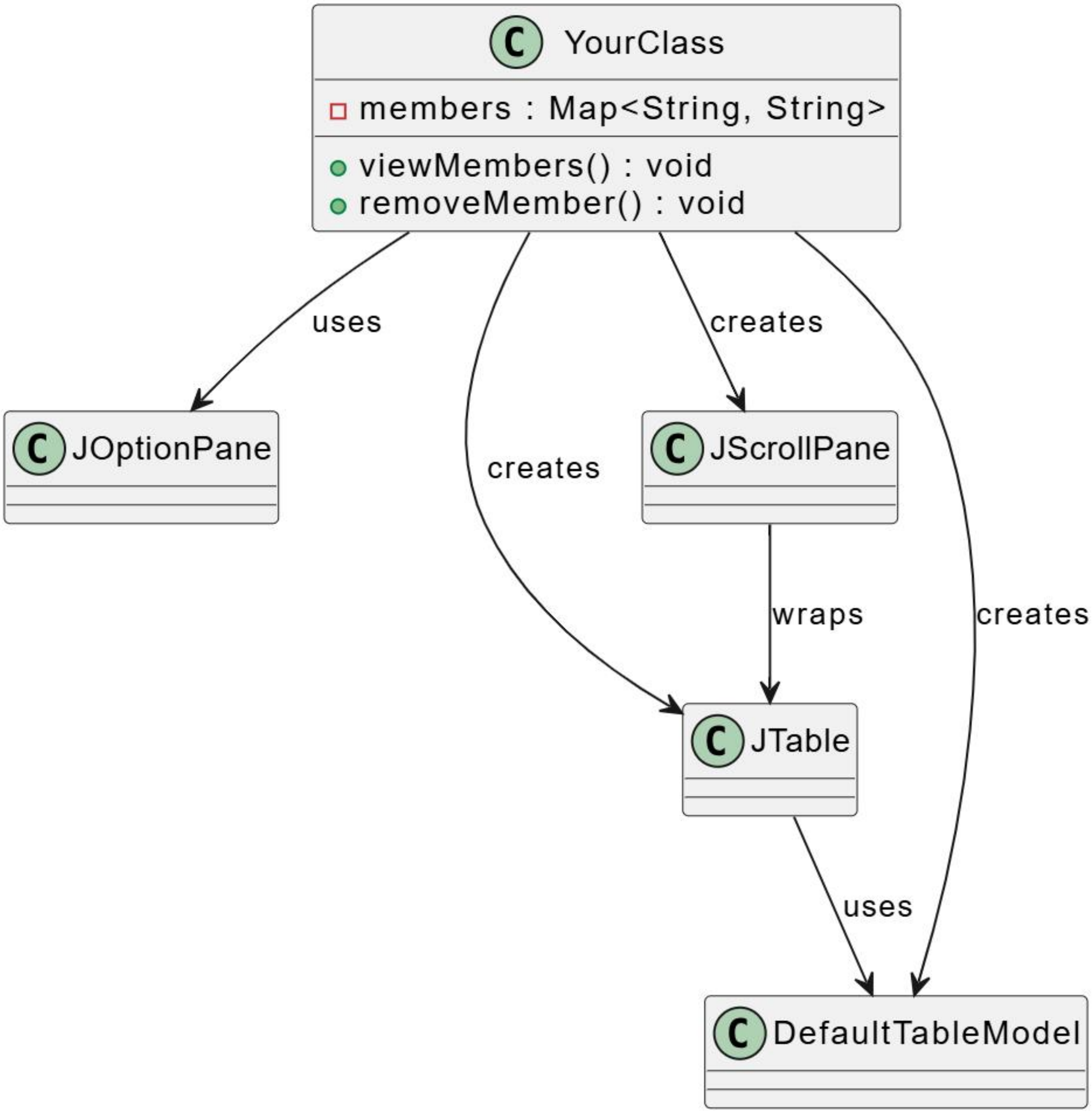


Usecase 2: Remove Member

Sequence diagram



Class diagram

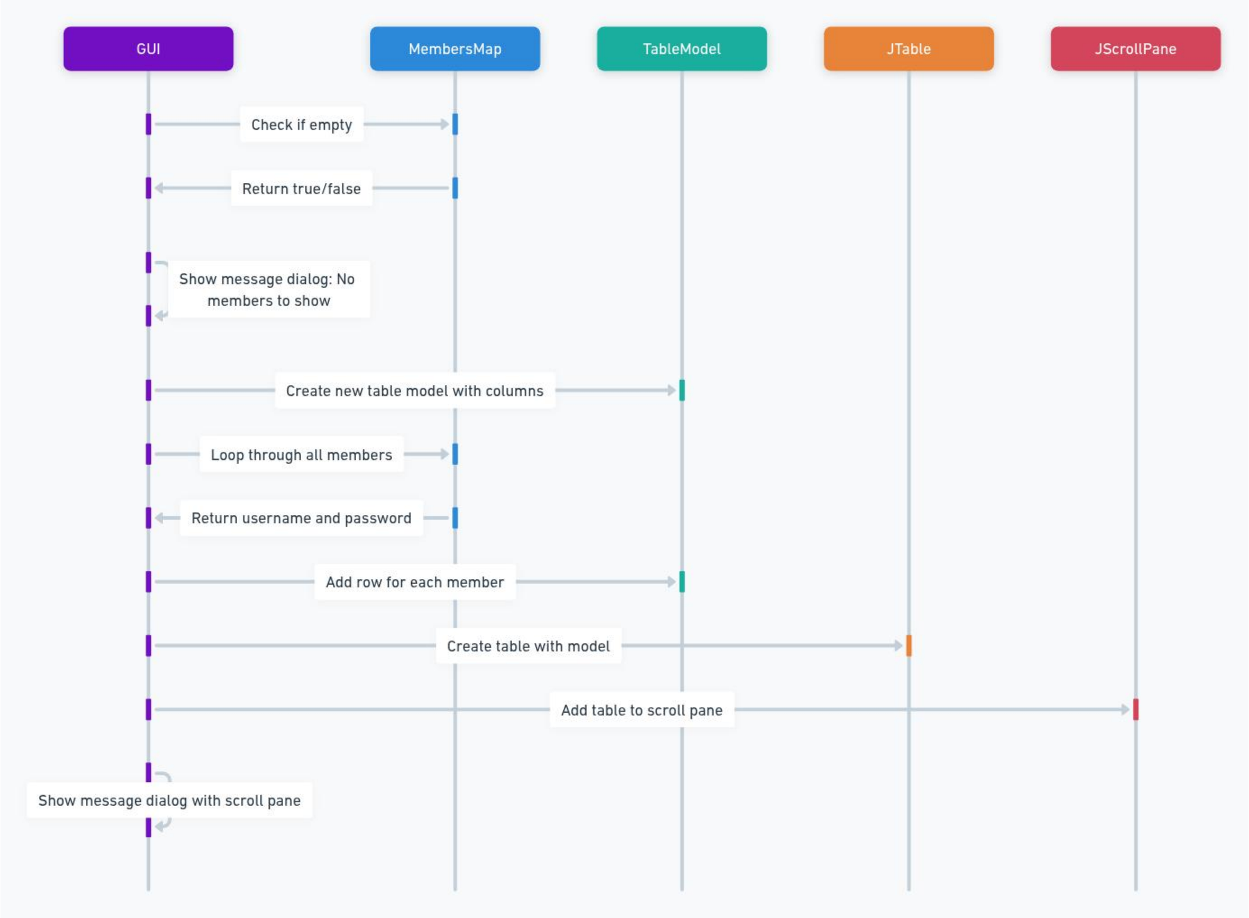


Fully dressed usecase:

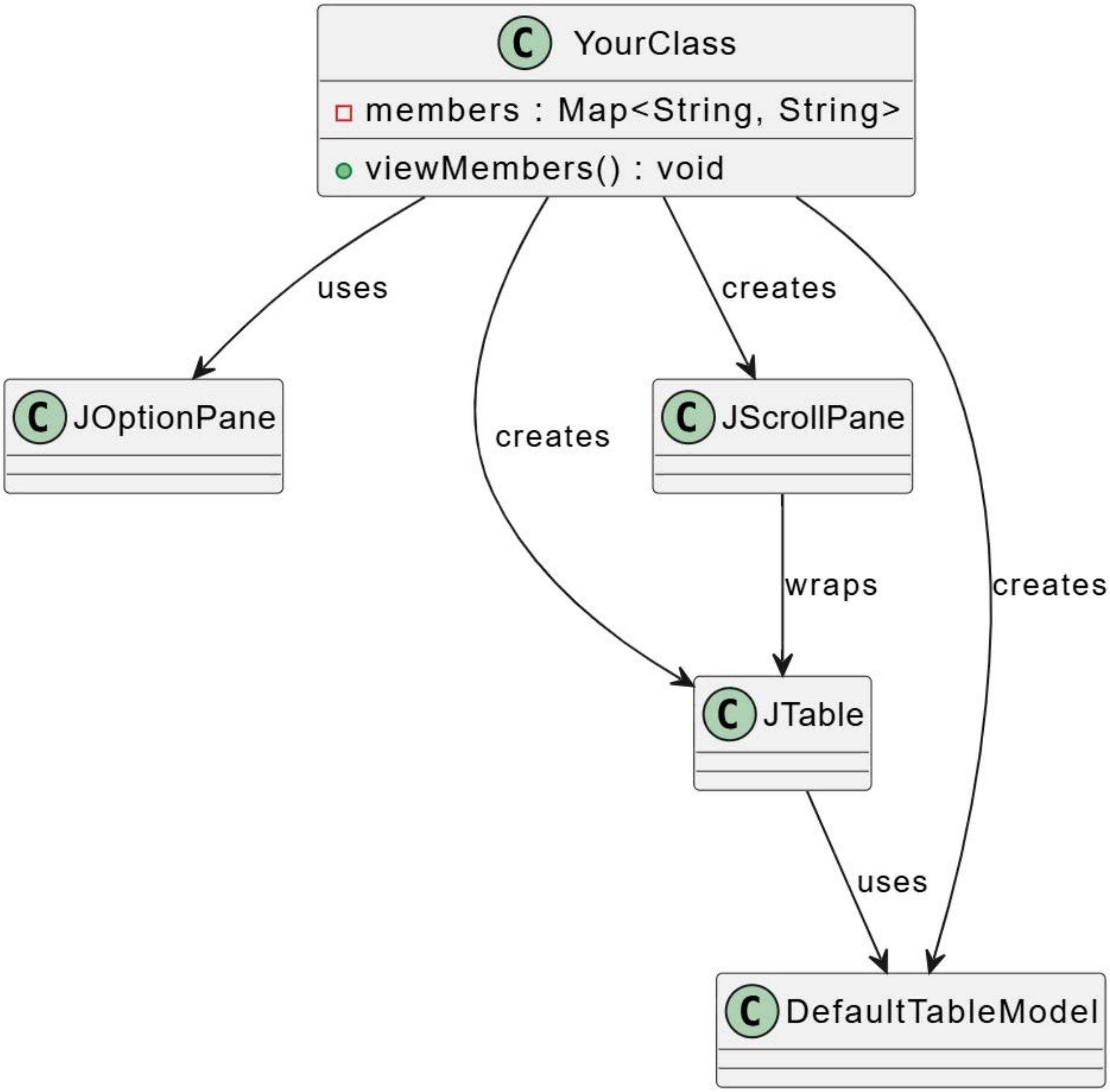
Use Case Element	**Details**
Use Case ID	UC-MM-02
Use Case Name	Remove Member
Primary Actor	Admin
Secondary Actors	None
Brief Description	This use case allows the admin to remove an existing member using their username.
Preconditions	Admin is logged in.- Members exist in the system.
Postconditions	Member is removed from the system if found.
Main Success Scenario	1. Admin clicks "Remove Member". 2. System prompts for username. 3. Admin enters a valid existing username. 4. System checks for existence. 5. System removes the member. 2. 6. Success message is shown.
Extensions (Alternate Flows)	**3a. Username not found:** 3a1. System displays "Member not found". **2a. Cancel Operation:** 2a1. Admin cancels dialog. Nothing happens.
Special Requirements	Prompt must allow manual entry. No confirmation prompt before delete.
Frequency of Use	Occasional, mostly during cleanup or admin operations.
Open Issues	No undo functionality. No confirmation step for delete action.

Usecase 3: View Members

Sequence diagram



Class diagram



Fully dressed usecase:

Use Case Element	Details
Use Case ID	UC-MM-03
Use Case Name	View Members
Primary Actor	Admin
Secondary Actors	None
Brief Description	This use case allows the admin to view all registered members in a table format.
Preconditions	Admin is logged in. Members may or may not exist.
Postconditions	Table of members is shown, or message that no members are available.
Main Success Scenario	1. Admin clicks "View Members". 2. System checks if any members exist. 3. System creates a table model. 4. All member usernames and passwords are loaded. 3. 5. Table is displayed to the admin.
Extensions (Alternate Flows)	**2a. No members exist:** 2a1. System displays message "No members to show".
Special Requirements	Member list is displayed in a scrollable JTable.
Frequency of Use	Frequent, especially to confirm member registration.
Open Issues	Passwords are shown in plain text. - No sorting or filtering functionality.