

# Design Specification

## CRC Cards:

Class: User Login	
Responsibilities	Collaborators
<p>Manage User Profile:</p> <ul style="list-style-type: none"> <li>*Store and retrieve user Credentials (username, password).</li> <li>*Allow users to view and edit their profile information.</li> <li>*Store passwords.</li> </ul> <p>Account Management:</p> <ul style="list-style-type: none"> <li>*Create a new user account</li> <li>*Delete an existing user account</li> <li>*Validate password and password confirmation match.</li> </ul> <p>Authentication:</p> <ul style="list-style-type: none"> <li>*Create a new user account</li> <li>*Handle login errors and provide feedback.</li> </ul>	<p>Event: manages events created by the user.</p> <p>TextField and JPasswordField: for username and password input during login.</p> <p>JButton: for actions like "Login".</p> <p>JLabel: to display labels for input fields and messages to the user.</p> <p>JOptionPane: for displaying dialog messages (e.g. error message, confirmation prompts).</p>
Class: Event	
Responsibilities	Collaborators
<p>Manage Event Details:</p> <ul style="list-style-type: none"> <li>*Store event attributes: title, date, time, course, description, priority level.</li> <li>*Allow editing of event details.</li> </ul> <p>Event Status Management:</p> <ul style="list-style-type: none"> <li>*Track Event Status (pending or completed).</li> <li>*Allow marking events as complete or incomplete.</li> </ul> <p>Reminders:</p> <ul style="list-style-type: none"> <li>*Set and manage reminder settings (type, time interval).</li> </ul> <p>Conflict Detection:</p> <ul style="list-style-type: none"> <li>*Check for time conflicts with other events.</li> <li>*Alert user of potential scheduling conflicts.</li> </ul>	<p>User Login: owner of the event.</p> <p>Calendar: displays the event in various calendar views.</p> <p>Reminder: manages reminders associated with the event.</p> <p>TextField: for inputting text fields like title and course.</p> <p>TextArea: for multi-line input like the description.</p> <p>JComboBox: for selecting date, start time, end time, and priority level.</p> <p>JCheckBox: for enabling/disabling reminders.</p> <p>JButton: for actions like "Save Event", "Remove Event", "Mark as Complete".</p> <p>JLabel: for labels and displaying event information.</p>

Class: Calendar	
Responsibilities	Collaborators
<p>Display Events:            *Provide daily, weekly, and monthly views of events            *Display event titles on their scheduled dates            *Indicate completed (green) and pending (blue) events.</p> <p>Event Management:            *Add events to the calendar.            *Remove events from the calendar            *Update event displays when events are added, edited, or removed.</p> <p>Navigation:            *Navigate between different dates and views.            *Update the display based on user navigation.</p> <p>Event Interaction:            *Allow users to select events to view details.            *Support editing or deleting events</p>	<p>Event: contains the events to be displayed.</p> <p>User Login: accesses events associated with the logged-in user.</p> <p>JPanel: for organizing the calendar layout.</p> <p>JList: for displaying lists of events in calendar views.</p> <p>JButton: for navigation controls (e.g. next day, previous week).</p> <p>JLabel: for displaying the current date, week, or month.</p> <p>MouseListener: to detect clicks on events and display event details.</p>
Class: Reminder	
Responsibilities	Collaborators
<p>Manage Reminder Settings:            *Store reminder type (email, push notification).            *Store reminder time interval before the event.            *Enable or disable reminders for an event.</p> <p>Send Reminders:            *Schedule reminders based on event time and reminder settings.            *Dispatch reminders via the chosen method.</p> <p>Customization:            *Allow users to customize reminder preferences per event.            *Support different reminder intervals.</p>	<p>Event: the event associated with the reminder.</p> <p>JCheckBox: for selecting reminder types and time intervals.</p> <p>Timer (existing Java library class): for scheduling reminders.</p> <p>JOptionPane: for displaying reminder notifications.</p>

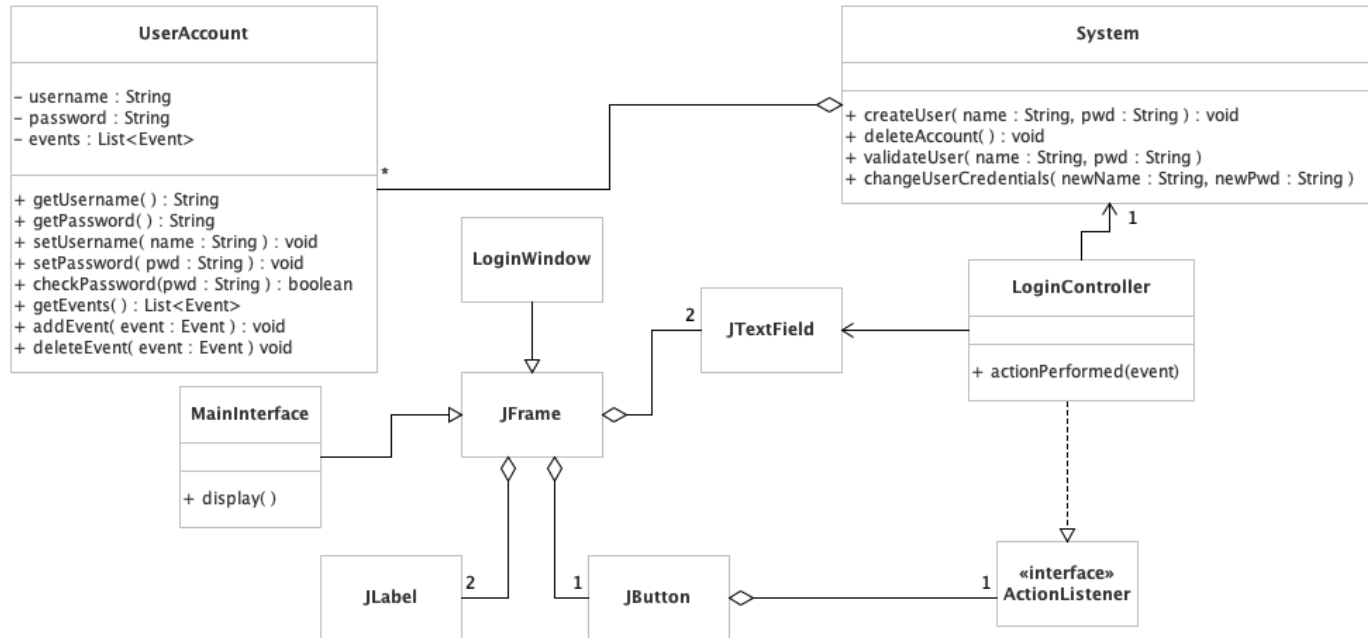
Class: Taskbar	
Responsibilities	Collaborators
Provide Navigation: *Home: Main Interface *User Profile: User settings *Add Events: Create new events *Calendar: View calendar *Route the user to the selected interface	User Login  Calendar  Event  JPanel: serves as the container for the taskbar.
Manage Button States: *Highlight the active view. *Enable or disable buttons based on context (e.g. user authentication status).	JButton: represents each navigation button (e.g. User Profile, Add Events)  ActionListener: handles events when buttons are clicked.
User Interaction Handling: *Respond to user clicks on navigation buttons. Update the UI based on navigation actions.	JLabel: enhances buttons with text labels.  JOptionPane: for displaying messages or confirmations if needed.

## UML Diagrams:

### LoginWindow(class diagram)

Use Case: Login

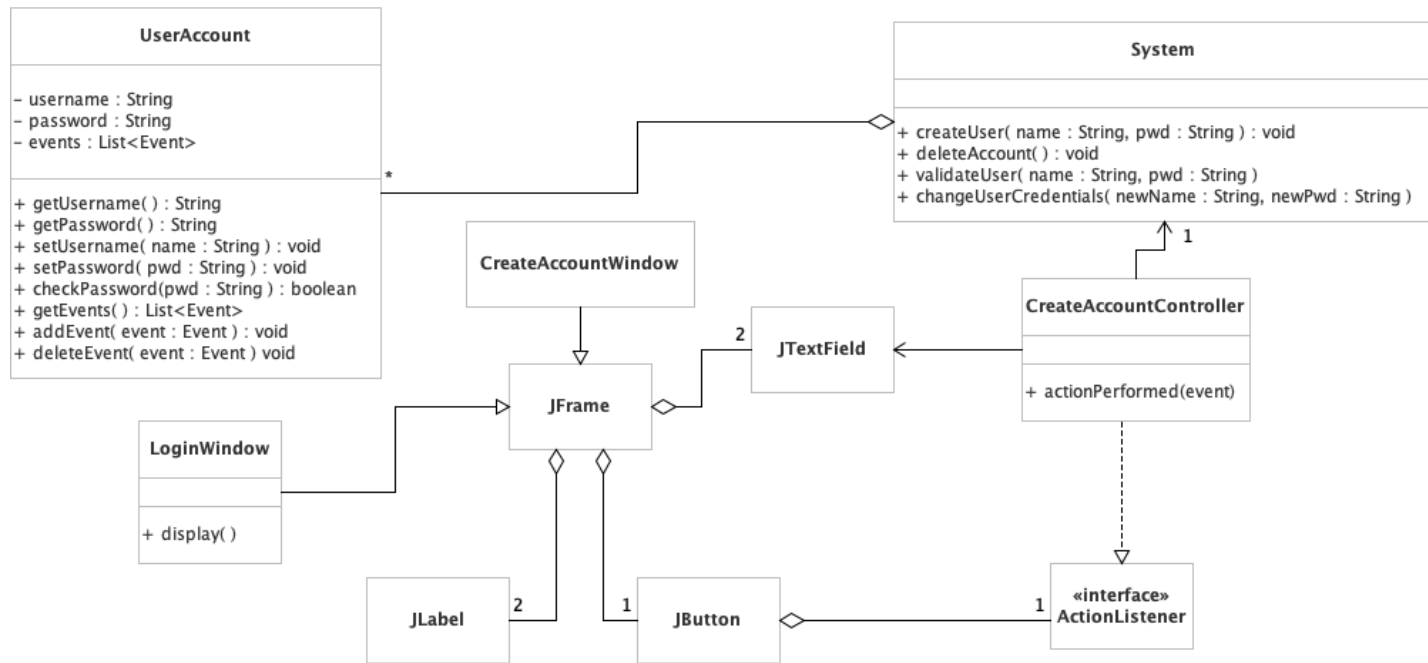
- 1) The user carries out Initiate Planner Application.
- 2) The System displays login window : username and password fields.
- 3) The user enters the username and password associated with their account.
- 4) The user clicks the "Login" button.
- 5) System validates above.
- 6) The System routes the user to the MainInterface.



## CreateAccountWindow(class diagram)

Use Case: Create Account

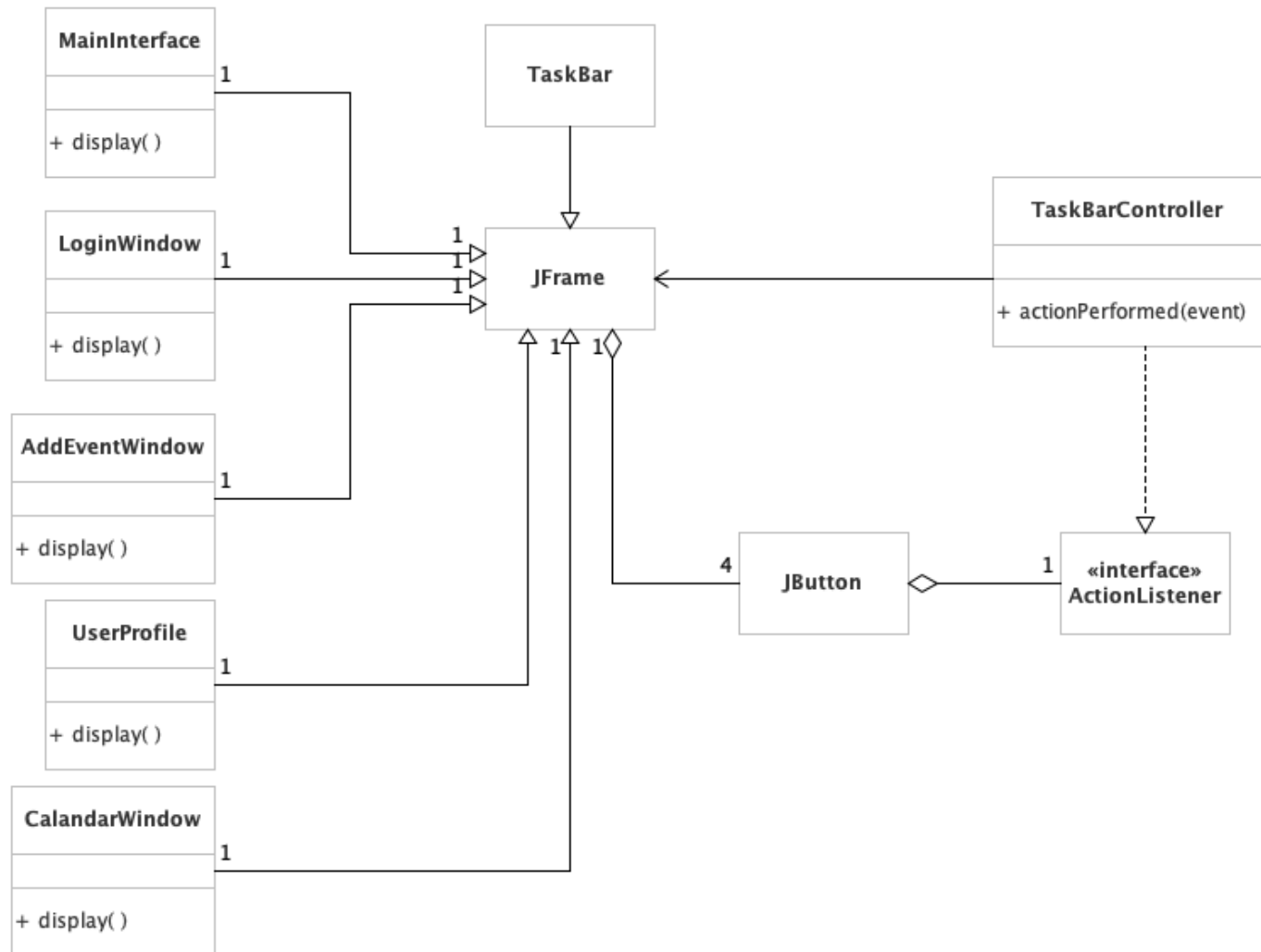
- 1) The user carries out Initiate Planner Application.
- 2) The System displays login window :  
username and password fields
- 3) The user clicks the "Create Account" button.
- 3) The user inputs the username and password associated with their new account.
- 4) The user clicks the "Create Account" button.
- 5) System creates the account.
- 6) The System routes the user to the LoginWindow.



## TaskBar(class diagram)

Task Bar Navigation:

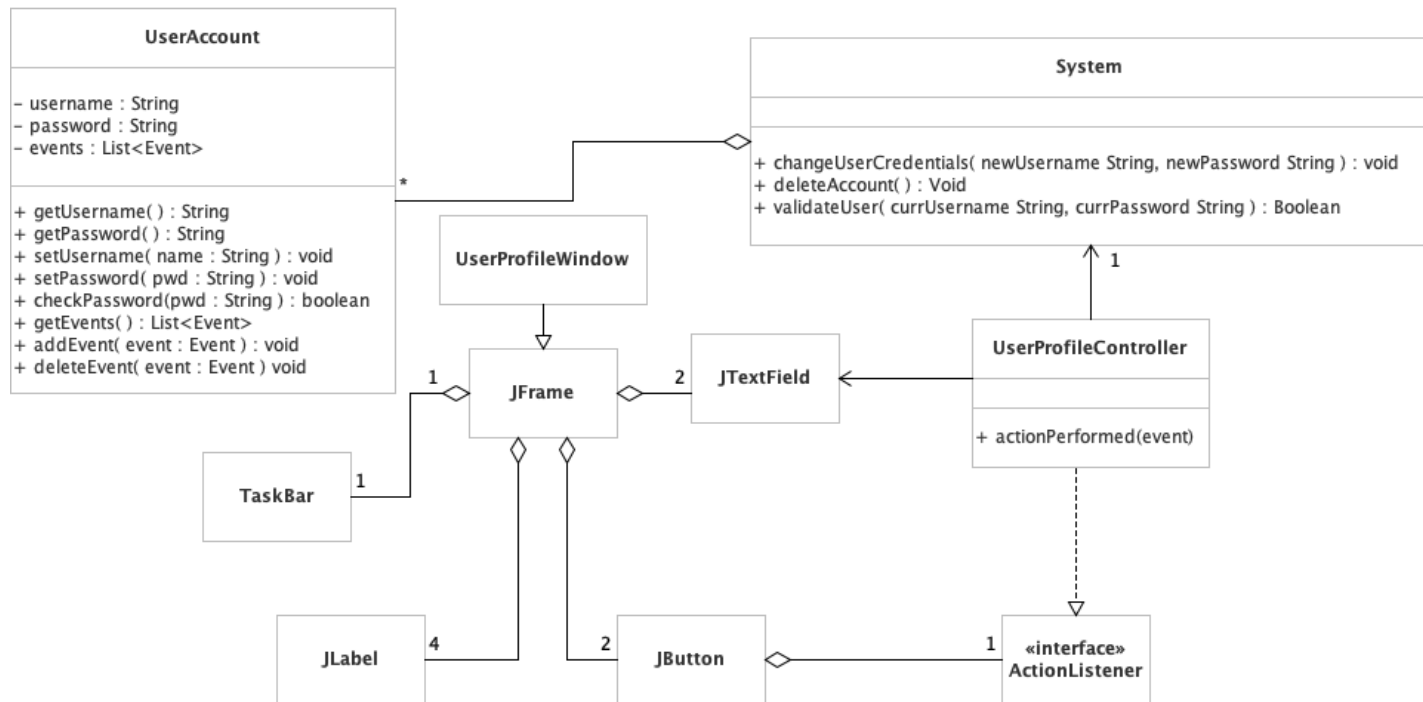
- 1) The user carries out Initiate Planner Application.
- 2) The user carries out Login.
- 3) The user clicks the button associated to the desired page.
- 4) The visibility of the current screen is set to false.
- 5) The visibility of new screen is set to true.



## UserProfile(class diagram)

Use Case: Change Credentials / Delete Account

- 1) The user carries out Initiate Planner Application.
- 2) The user utilizes the TaskBar to navigate to the UserProfileWindow
- 3) The System gets Username and Password of the Account.
- 4) The username and password is displayed in the UserProfileWindow.
- 5.a) The user enters a new Username/Password into the TextFields and click the "Change Credentials" Button
- 5.b) The user enters their current Username/Password into the TextFields and click the "Delete Account" Button
- 6.a) The System changes the account credentials.
- 6.b) The System validates the user password and username
- 7.b) The Deletes the user's account

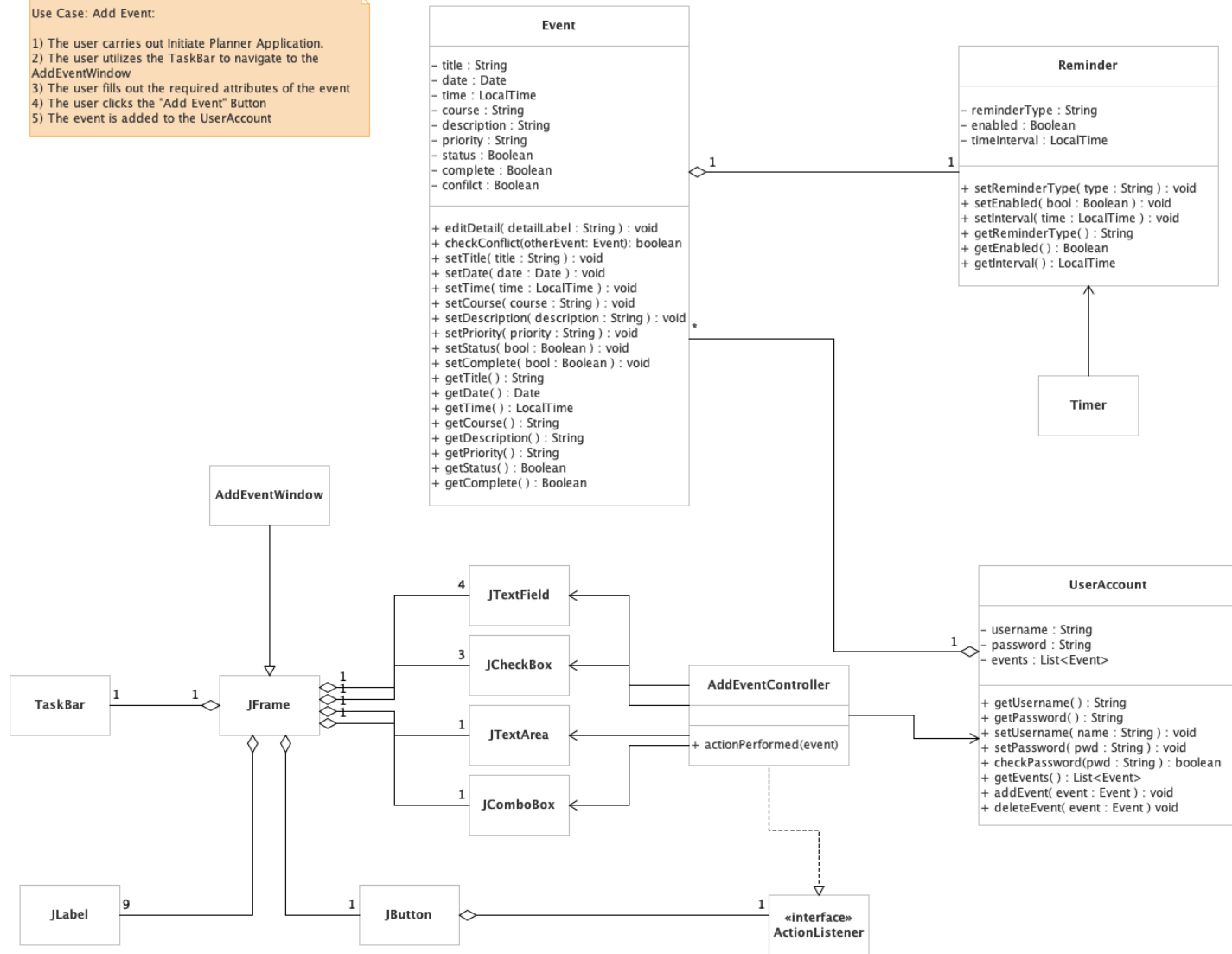




## AddEventWindow(class diagram)

Use Case: Add Event:

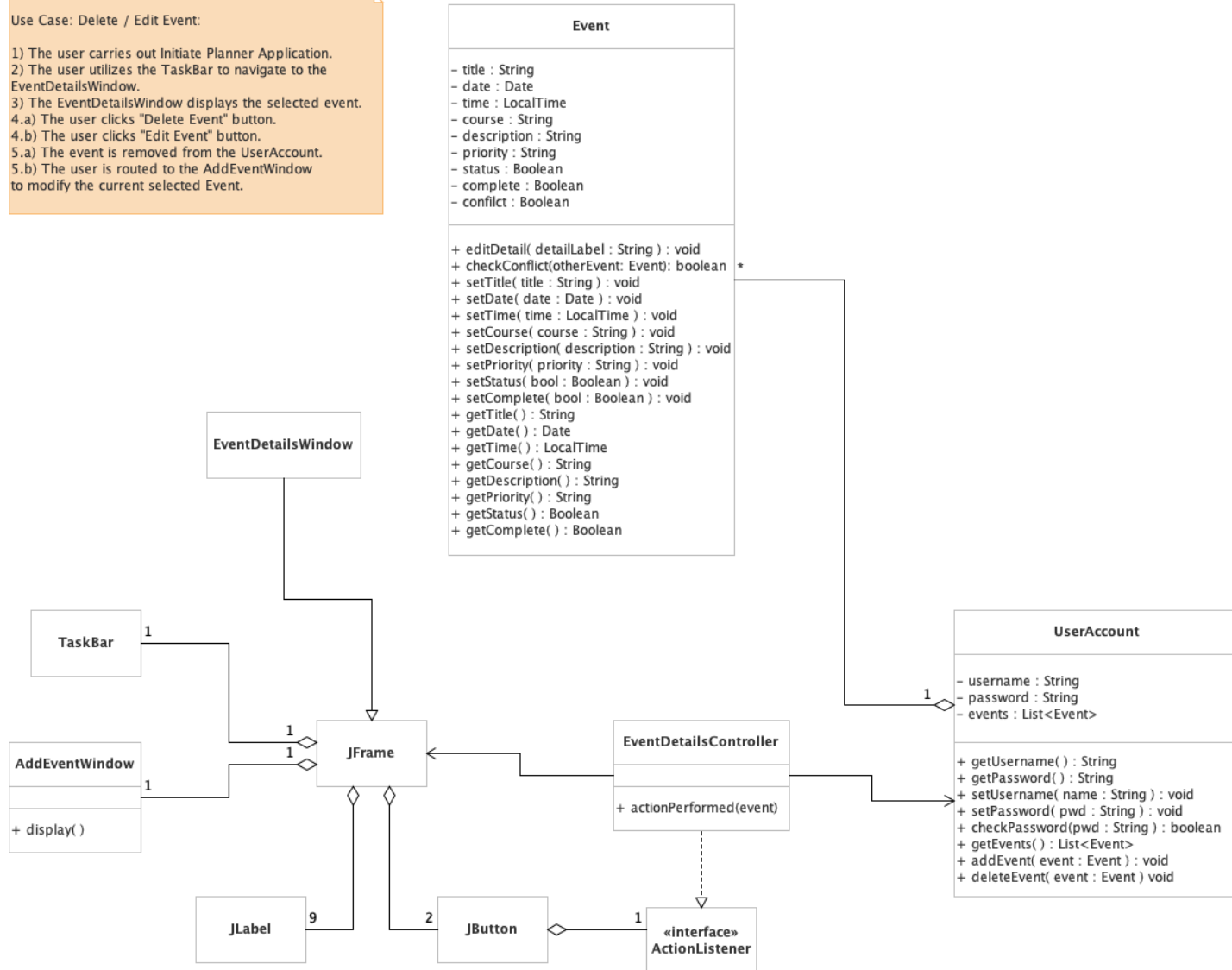
- 1) The user carries out Initiate Planner Application.
- 2) The user utilizes the TaskBar to navigate to the AddEventWindow
- 3) The user fills out the required attributes of the event
- 4) The user clicks the "Add Event" Button
- 5) The event is added to the UserAccount



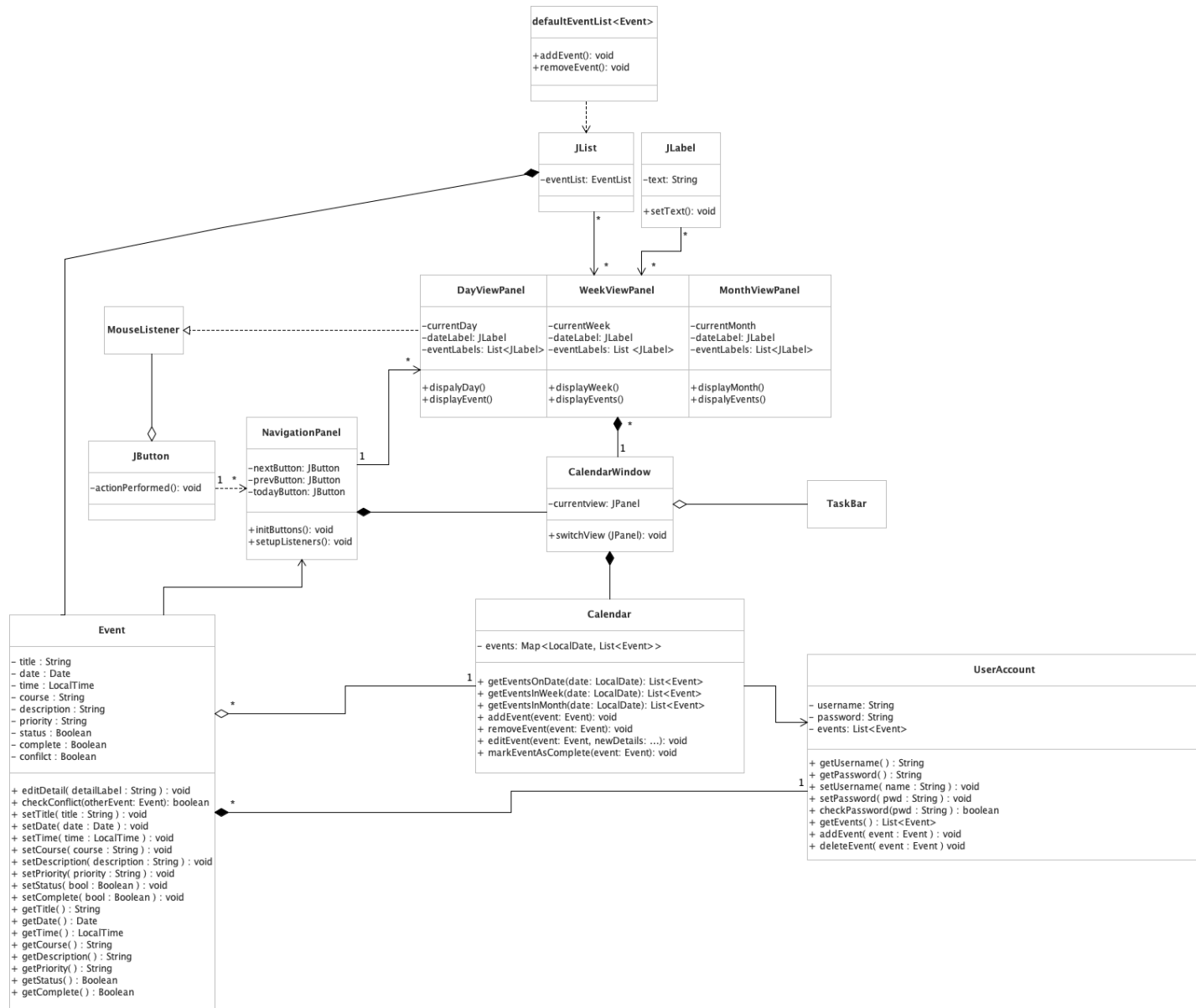
## EventDetailsWindow(class diagram)

Use Case: Delete / Edit Event:

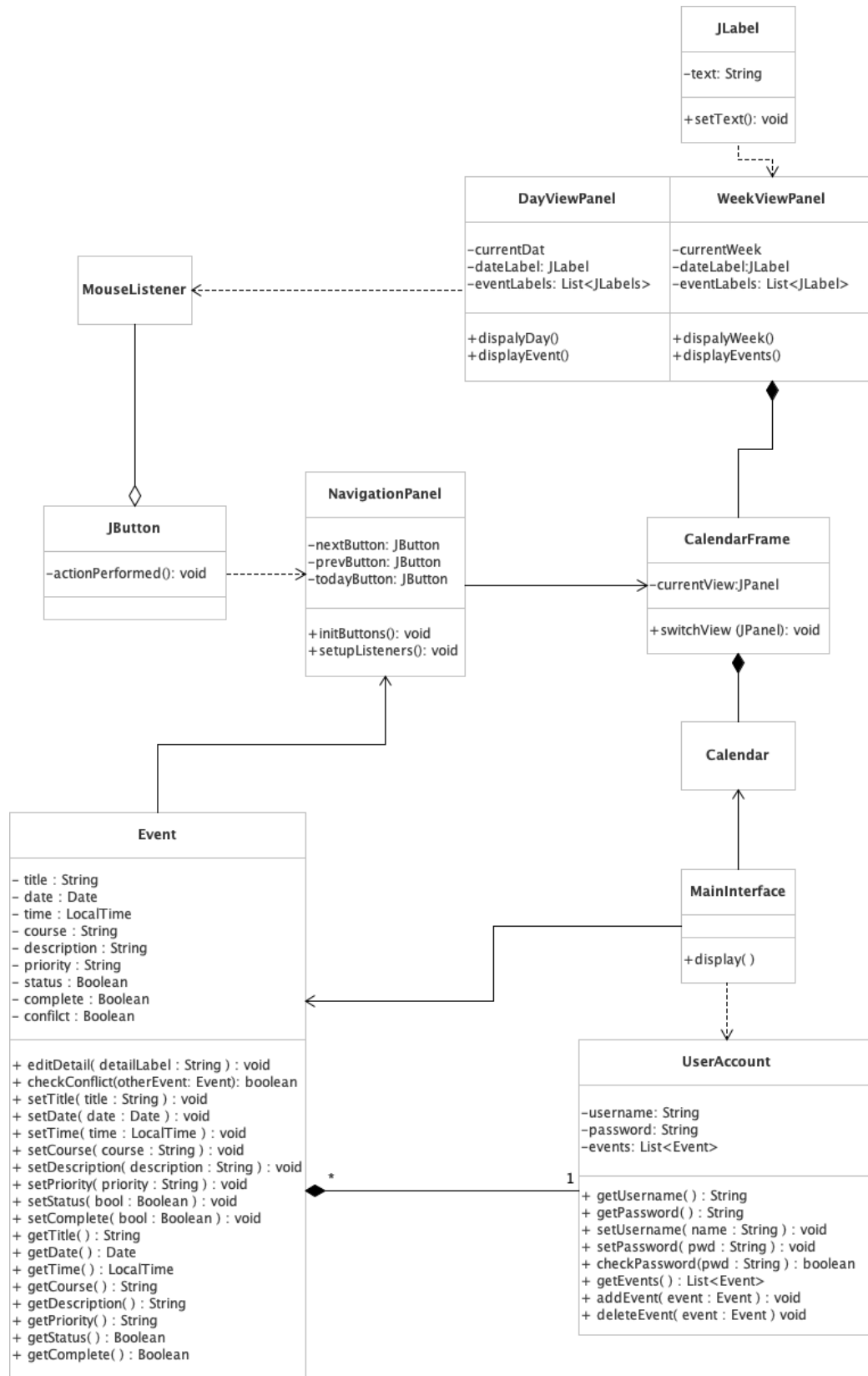
- 1) The user carries out Initiate Planner Application.
- 2) The user utilizes the TaskBar to navigate to the EventDetailsWindow.
- 3) The EventDetailsWindow displays the selected event.
- 4.a) The user clicks "Delete Event" button.
- 4.b) The user clicks "Edit Event" button.
- 5.a) The event is removed from the UserAccount.
- 5.b) The user is routed to the AddEventWindow to modify the current selected Event.



# CalendarWindow(class diagram)



## MainInterface(class diagram)



## UserLogin(sequence diagram)

