## **Chapter Two: High-Level Design (Sequence Diagram)**

### **2.1 High-Level Sequence Diagram**

This chapter provides an in-depth exploration of how users and system components interact over time to execute critical operations such as registration, login, task creation, setting reminders, logging out, editing tasks, and managing users. Each sequence diagram captures the dynamic behavior of the system, offering a step-by-step visualization of message exchanges and decision points, which are essential for understanding the system's operational flow.

### **2.2 Sequence Diagram Components**

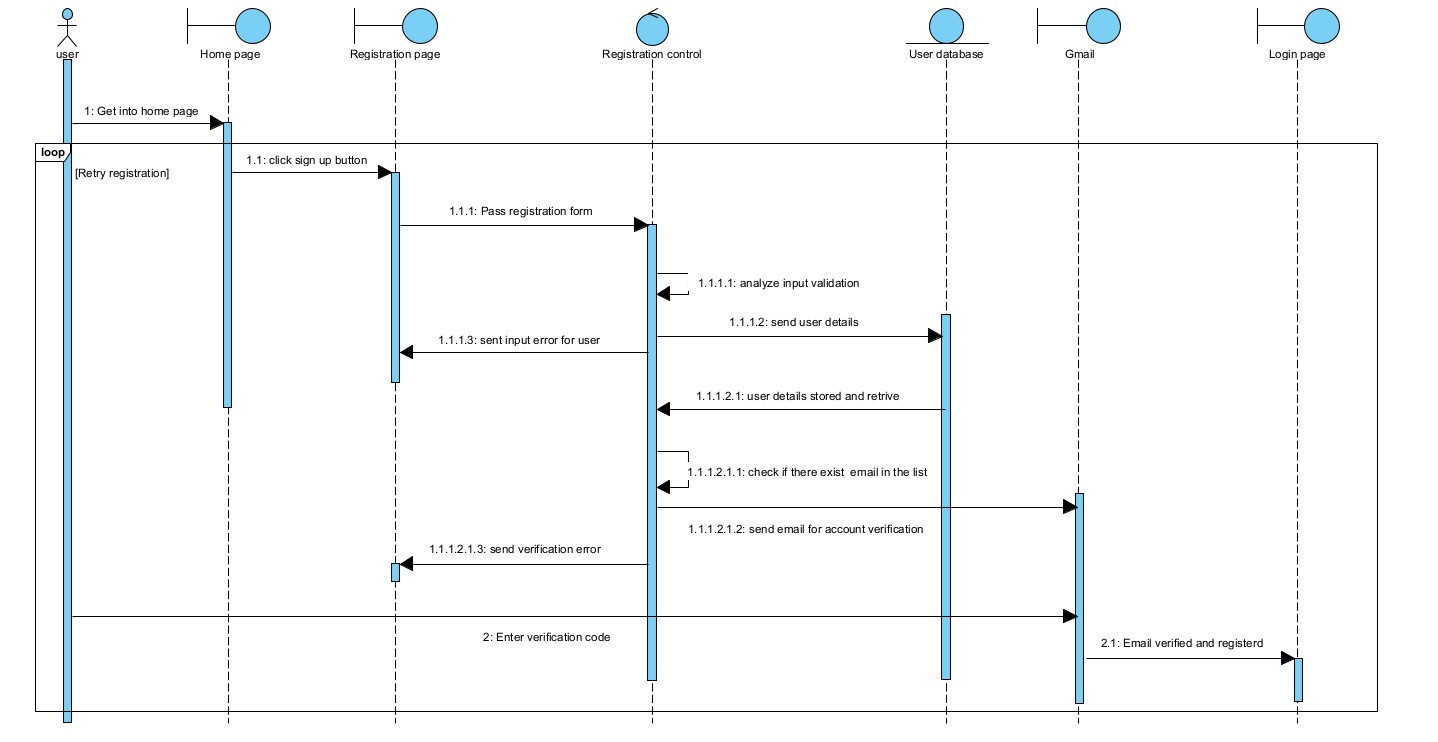
* **Actor/Object**: Represents external entities (e.g., User, Admin) and internal system components (e.g., Task Controller, User Database), each playing a specific role in the interaction.
* **Lifeline**: A vertical dashed line that indicates the existence of an object or actor throughout the scenario, showing when it is active or idle.
* **Messages**: Horizontal arrows representing the communication between components, labeled with the action or data being exchanged (e.g., "Submit login input").
* **Activation Bar**: A thin rectangle on the lifeline that signifies the duration an object is processing a message, providing insight into system responsiveness.
* **Fragments**: Special constructs like loops (e.g., [can create many times]) or alternate flows (e.g., [user confirm Logout]) that depict repetitive or conditional behaviors.

### 

### **2.3 Sequence Descriptions by Use Case**

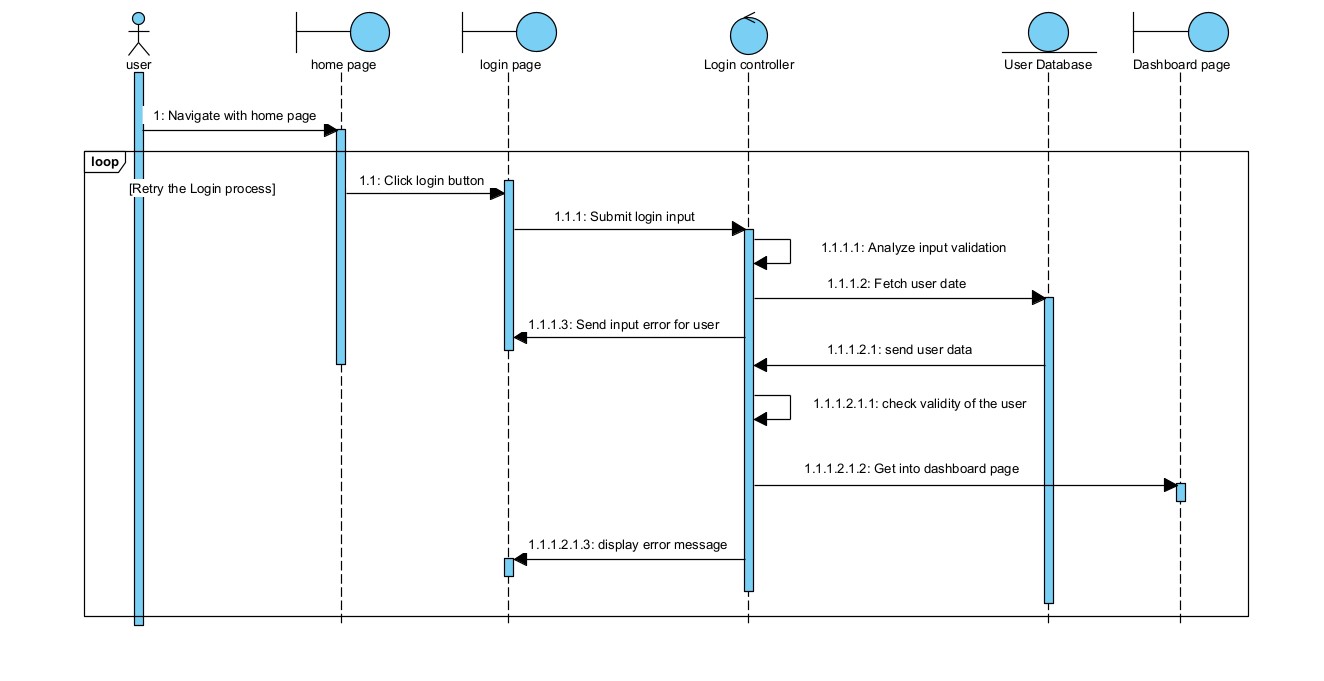
**Registration**

1. The user navigates to the home page, the entry point of the application.
2. The user clicks the "Sign Up" button, triggering the display of a detailed registration form requiring name, email, and password.
3. The user fills in the form and submits the data, which is then sent to the system for processing.
4. The system validates the input (e.g., checks for duplicate emails), stores the user data in the database, and initiates an email verification process. If validation fails, an error message is returned.  
   *Sequence Diagram*: This diagram illustrates the User interacting with the Home Page, Registration Page, Registration Controller, User Database, Gmail, and Login Page. It includes a loop for retrying registration, detailed validation steps, and the email verification workflow.

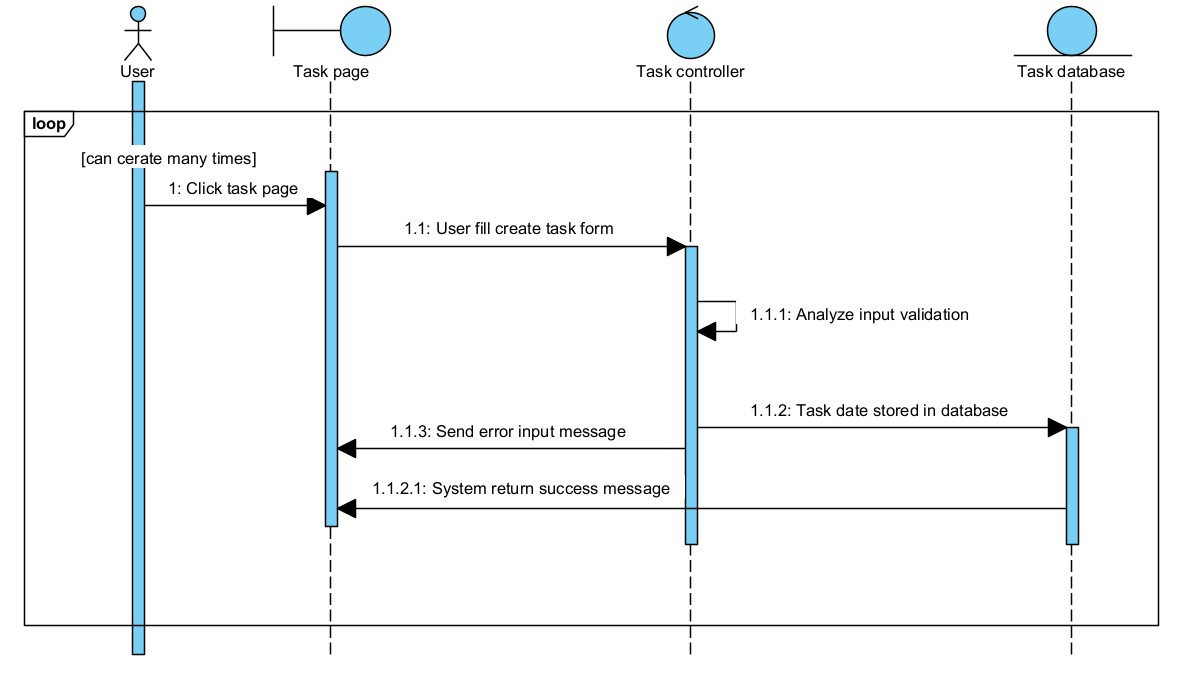


**Login**

1. The user enters their email and password into the login form, initiating the authentication process.
2. The system forwards this input to the login controller, which validates the credentials against the database.
3. Upon successful validation, the user is redirected to their personalized dashboard; otherwise, an error alert is triggered to prompt re-entry.  
   *Sequence Diagram*: This diagram depicts the User interacting with the Home Page, Login Page, Login Controller, User Database, and Dashboard Page. It includes validation steps, error handling with specific messages, and the transition to the dashboard.

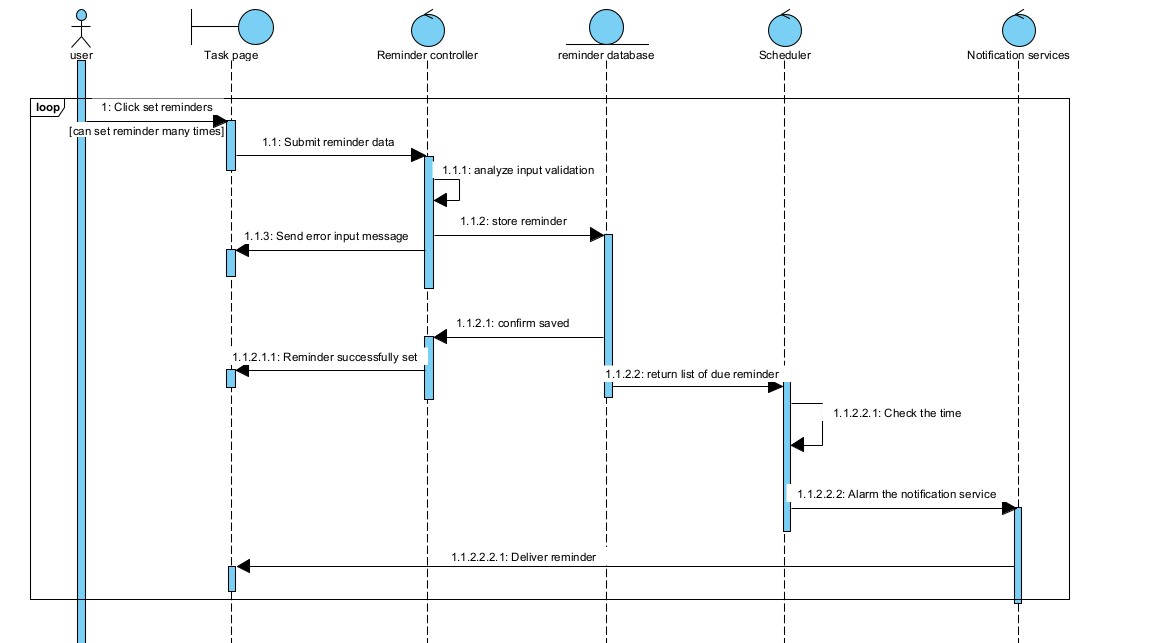


**Create Task**

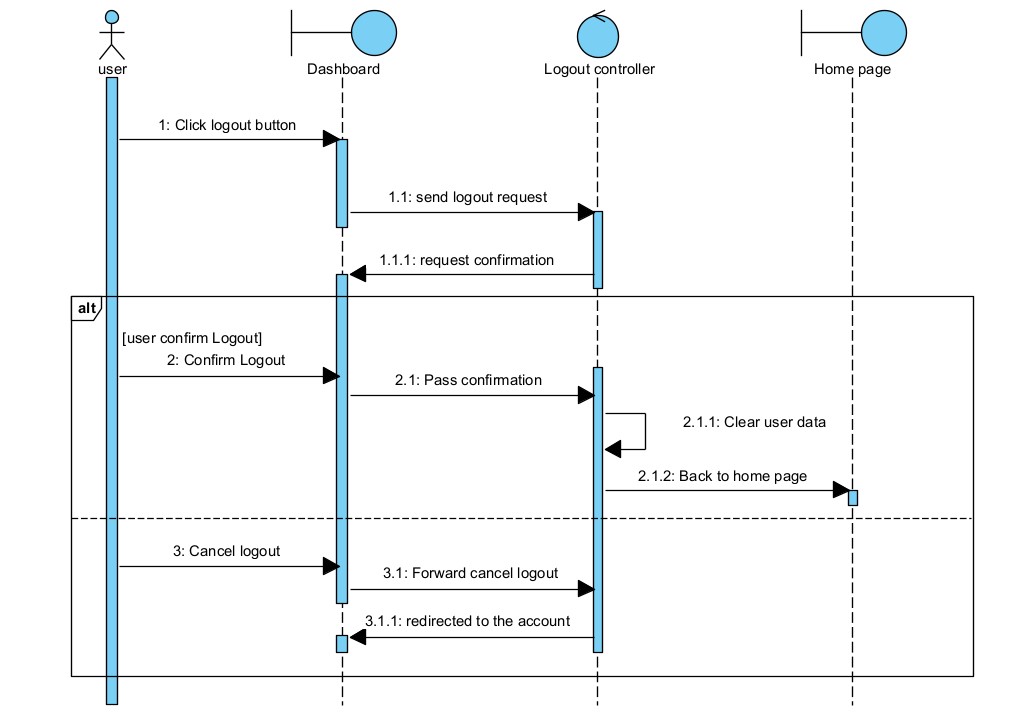
1. The user clicks the "Create Task" button on the task page, signaling their intent to add a new task.
2. The system responds by displaying a form where the user can input details such as title, description, due date, and priority.
3. The user submits the completed form, and the system performs input validation before saving the task to the database.
4. A success message confirms the task has been added to the user's list, or an error message is shown if validation fails.  
   *Sequence Diagram*: This diagram illustrates the User interacting with the Task Page, Task Controller, and Task Database. It includes input validation, error handling, and a confirmation step with a loop for multiple task creations.  
   

**Set Reminder**

1. The user selects a task and chooses to set a reminder, specifying a time and notification method (e.g., email or popup).
2. The system saves the reminder details to the database and confirms the action to the user.
3. At the scheduled time, the scheduler checks for due reminders and triggers the notification process.
4. The notification service delivers the alert to the user, ensuring they are reminded of the task deadline.  
   *Sequence Diagram*: This diagram shows the User interacting with the Task Page, Reminder Controller, Reminder Database, Scheduler, and Notification Service. It includes validation, storage, scheduling, and delivery steps.

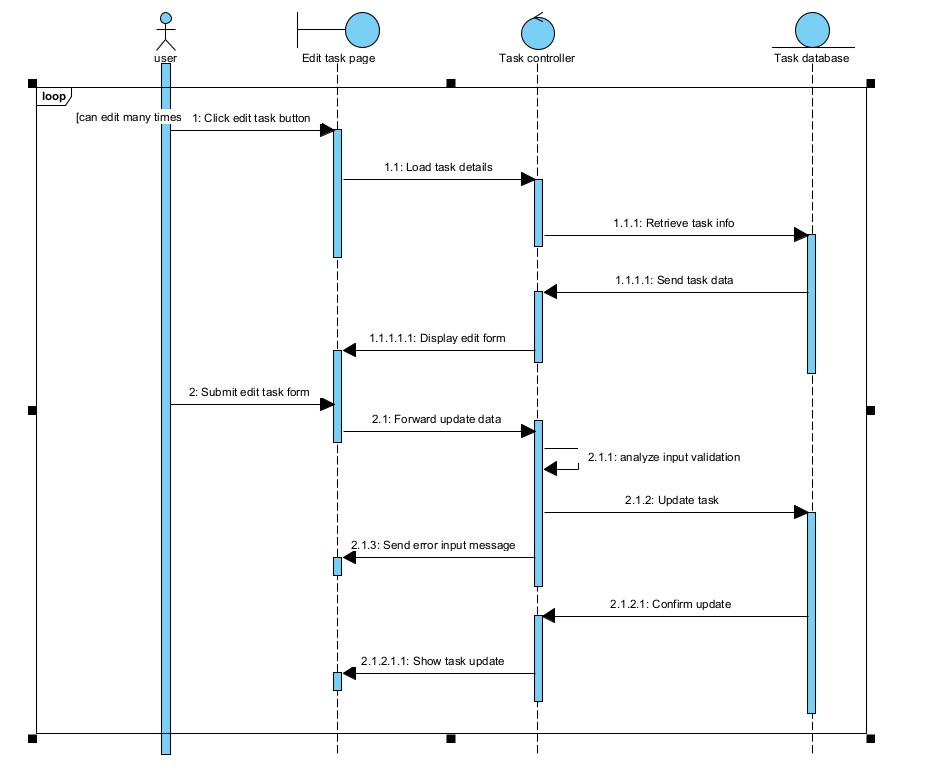


**Logout**

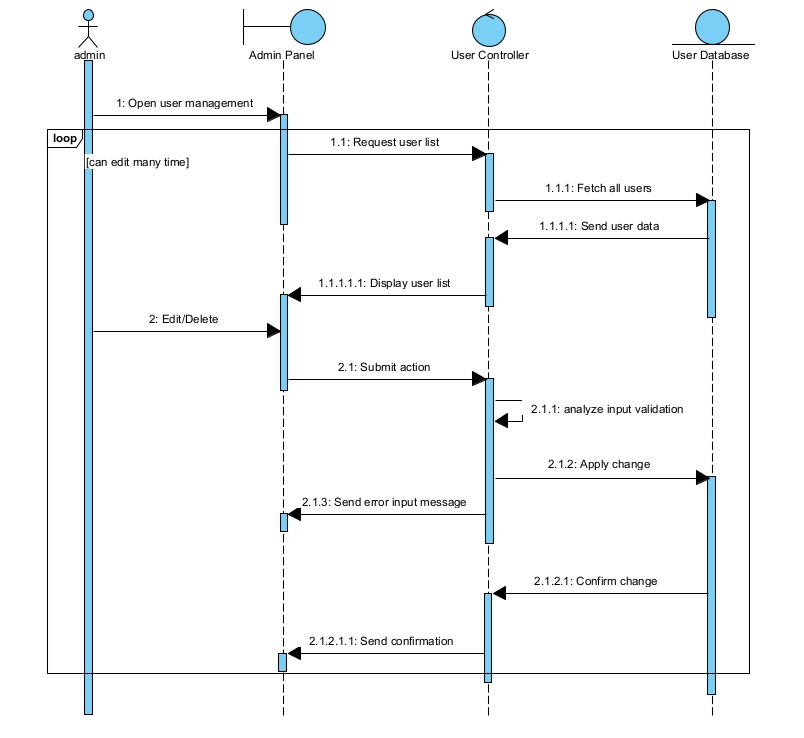
1. The user clicks the "Logout" button on the dashboard, initiating the session termination process.
2. The system requests confirmation from the user to prevent accidental logouts.
3. The user confirms the action, and the system clears session data and redirects to the home page; if canceled, the user remains on the dashboard.  
   *Sequence Diagram*: This diagram depicts the User interacting with the Dashboard, Logout Controller, and Home Page. It includes an alternate flow for confirmation or cancellation with clear redirection logic.  
   

**Edit Task**

1. The user clicks the "Edit Task" button, prompting the system to load the task's current details.
2. The system displays an edit form pre-filled with the task's data, allowing modifications.
3. The user submits the updated details, which the system validates before updating the database.
4. A confirmation message is shown to the user, or an error is displayed if the update fails.  
   *Sequence Diagram*: This diagram illustrates the User interacting with the Edit Task Page, Task Controller, and Task Database. It includes loading, validation, updating, and confirmation steps with a loop for multiple edits.



**Manage Users (Admin)**

1. The admin opens the user management panel from the dashboard to oversee user activities.
2. The admin requests a list of users, which the system fetches from the database and displays.
3. The admin selects an action (e.g., edit or delete), submits it, and the system applies the change after validation.
4. A confirmation is sent to the admin, or an error message is displayed if the action fails.  
   *Sequence Diagram*: This diagram shows the Admin interacting with the Admin Panel, User Controller, and User Database. It includes fetching, validation, applying changes, and confirmation steps with a loop for multiple edits.  
   

### 

### 

### 

### 

### 

### 

### 

### **2.4 Tools and Steps Used**

**Tool Used**: Visual Paradigm, a powerful UML modeling tool that supports the creation of detailed sequence diagrams.  
**Steps**:

1. Open a new sequence diagram within Visual Paradigm to begin the design process.
2. Add actors (e.g., User, Admin) and system objects (e.g., Task Controller, Database) to represent all entities involved.
3. Define lifelines for each object to show their active periods throughout the scenario.
4. Draw messages as arrows between lifelines, labeling them with specific actions or data (e.g., "Submit reminder data").
5. Incorporate fragments such as loops or alternate flows to handle repetitive or conditional interactions.
6. Review the diagram for accuracy, adjust layouts for clarity, and export it as an image file for inclusion in the documentation.