**CS3354 Software Engineering**

**Final Project Deliverable 1**

DigiPlan: Plan Out Your Day

By:

Hamza Bhatti, Muhammad Mubeen, Prajwal Paul, Saad Bhatti, Mohammad Mubashir, Abdus Quadri, Sai Tanjavuru.

**Project Proposal**

**Title**: DigiPlan - Plan out your day.

**Members**: Hamza Bhatti, Muhammad Mubeen, Prajwal Paul, Saad Bhatti, Mohammad Mubashir, Abdus Quadri, Sai Tanjavuru.

**Task**: We will be creating the calendar software application that will allow for calendar view in weeks, months and days. The application will also allow for events to be created and plan accordingly, with no conflicts. In addition, the application will mark special days such as holidays and different categories of events in different colors.

**Reason for Choice**: We chose this task because as college students, we understand how important it is to keep a calendar/planner in order to succeed. It is common to forget assignments and deadlines as our days can get quite busy. This application allows us and others to know what items to expect each day and also set deadlines for the completion of a task or assignment. This application’s main audience is higher-level students and busy software companies.

**Instructor Feedback:** The instructor provided us feedback stating our project plan was sufficient with the requirements and instructed us to proceed. With that, we were required to do research on existing applications/software that correlate with ours. Using that information, we will be exploring ideas that differentiate our software.

“Thank you so much for the original ideas you proposed. Essentially, each team can go ahead and start working on their proposed project. In the meantime, please make sure that you include the following in your **final project deliverable 2 (final deliverable)**:

- A comprehensive research to find similar project implementations. Cite your findings properly using IEEE citation format.

- Make sure that you are adding extra feature(s) to uniquely differentiate your design from already existing similar implementations. Clearly explain what these feature(s) are.

- A comparison of your design with similar implementations in the field. This could be in any format of your choice, such as a table, paragraphs, charts, etc.”

**1.** Github repository:

These members created a Github account.

* Muhammad Mubeen - Mubeeni35
* Hamza Bhatti
* Saad
* Prajwal Paul - prajwalpaul1206
* Mohammad Mubashir
* Abdus Quadri
* Sai Tanjavuru

The repository was created by Saad.

* [sbh111](https://github.com/sbh111)/[CS3354-DigiPlan](https://github.com/sbh111/CS3354-DigiPlan)

Saad added all seven members and two TAs as contributors.

Prajwal created the first commit with the team name as the content.

Abdus made the “project\_scope” commit with the functions of the software.

Prajwal uploaded all relevant files to the team repository.

**1b. Delegation of tasks:**

Create a GitHub Repository - Saad - done

Add all members and TA as collaborators - Saad - done

Make first Commit to the repository - Prajwal - done

Make “project scope” commit to the repository - Abdus - done

Create Project Deliverable 1 google doc and share with all members -Muhammad -done

Include title and members in doc - Abdus - done

Address feedback provided for proposal - Prajwal - done

Choose software process model and why - Prajwal - done

List software requirements - Sai - done

Create use-case diagram - Abdus - done

Create sequence diagrams for each case - Saad - done

Create class diagram - Sai, Hamza, Prajwal - done

Provide architectural design of the project - Mohammad, Hamza - done

Final Check 1 - Muhammad

Final Check 2 - Prajwal - done

Submit Project Deliverable 1 - Prajwal

Create Project Deliverable 2 google doc and share with all members - Hamza

Include title and members in doc - Mohammad

Copy and paste everything from PD1 unto doc - Abdus

Create Project Scheduling using automated tool - Sai

Cost Estimation - Saad

Effort Estimation - Abdus

Pricing Estimation - Mohammad

Hardware cost Estimation - Muhammad

Software cost Estimation - Hamza

Personnel cost Estimation - Abdus

Compile all pricing and estimations - Prajwal

Create Test Plan for Software - Saad

Compare work with similar designs - Sai

Conclusion - Prajwal

Format references (if any) properly - Muhammad

Final Check 1 - Hamza

Final Check 2 - Prajwal

Submit Project Deliverable 2 - Hamza

Create google slides and share among members - Abdus

Divide topics of presentation among members - All

Work on respective portion of the presentation and create slides - All

Personal Rehearsal with timing to keep under the time limit - All

1 Group Rehearsal (more if needed) - All Final Presentation - All

1. **Which software process model is employed in the project and why?**

After incorporating the requirements of the software and understanding the purpose of the final product, we decided to use the prototyping evolutionary model. We decided to pick this process as the model employed in the project because this model is iterative. It allows developing an increasingly more complete version of the software. In the prototyping model, the process starts with communication. We are able to get the requirements from the project scope and define overall objectives. As the prototype is created, it is placed under scrutiny and sent back to development to improve based on the needed requirements. An advantage of using this type of process is that the customers and developers play an important part in the process.

1. **Software Requirements  
   3.a.) Functional requirements**

Functional Requirements:

* Application should allow the user to view the calendar in weeks, months, and days.
* Application should allow the user to plan, schedule, delete and modify events and tasks.
* Application should alert the user if there is a conflict of one or more events at time of creation of said event.
* Application should remind the user of the event or task.
* Application should mark special days and holidays.
* Application should allow user to color code events and tasks
* Application should allow user to show all events as a list.

**3.b.) Non-functional requirements**

Non-functional Requirements:

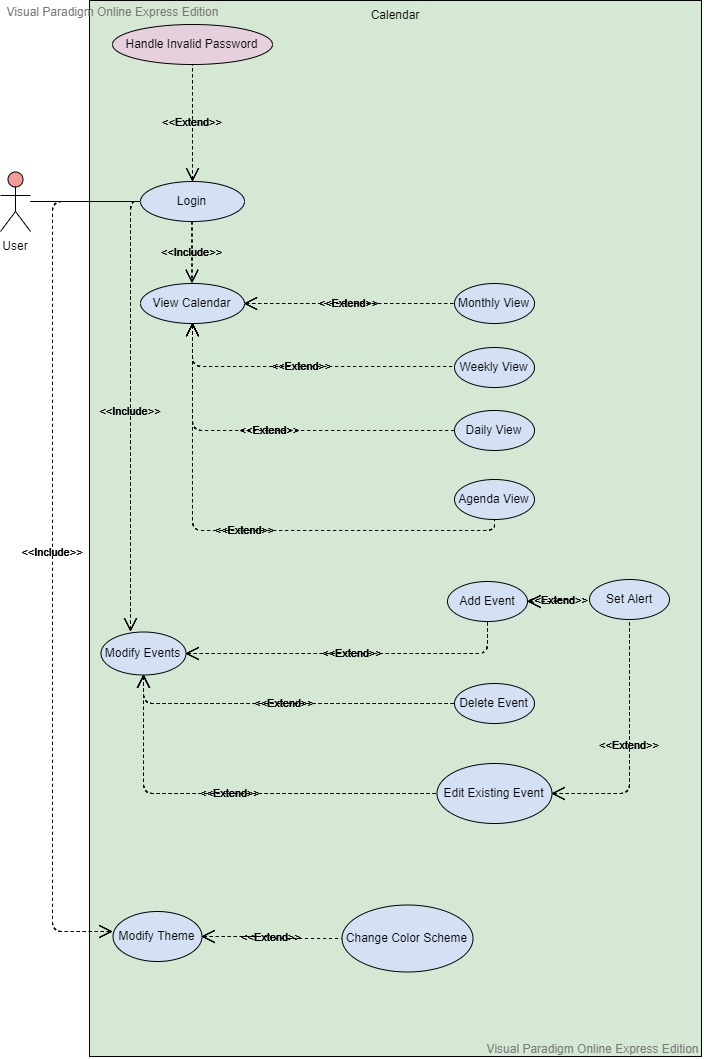
1. Needs to be space-efficient
2. Needs to be available all hours, on all platforms with no downtime
3. Needs to be responsive (fast)
4. Needs to be reliable

* Organizational Requirements:

1. UI/UX shouldn’t be cluttered
2. Should be straightforward to use
3. Users should be able to authenticate themselves using their login

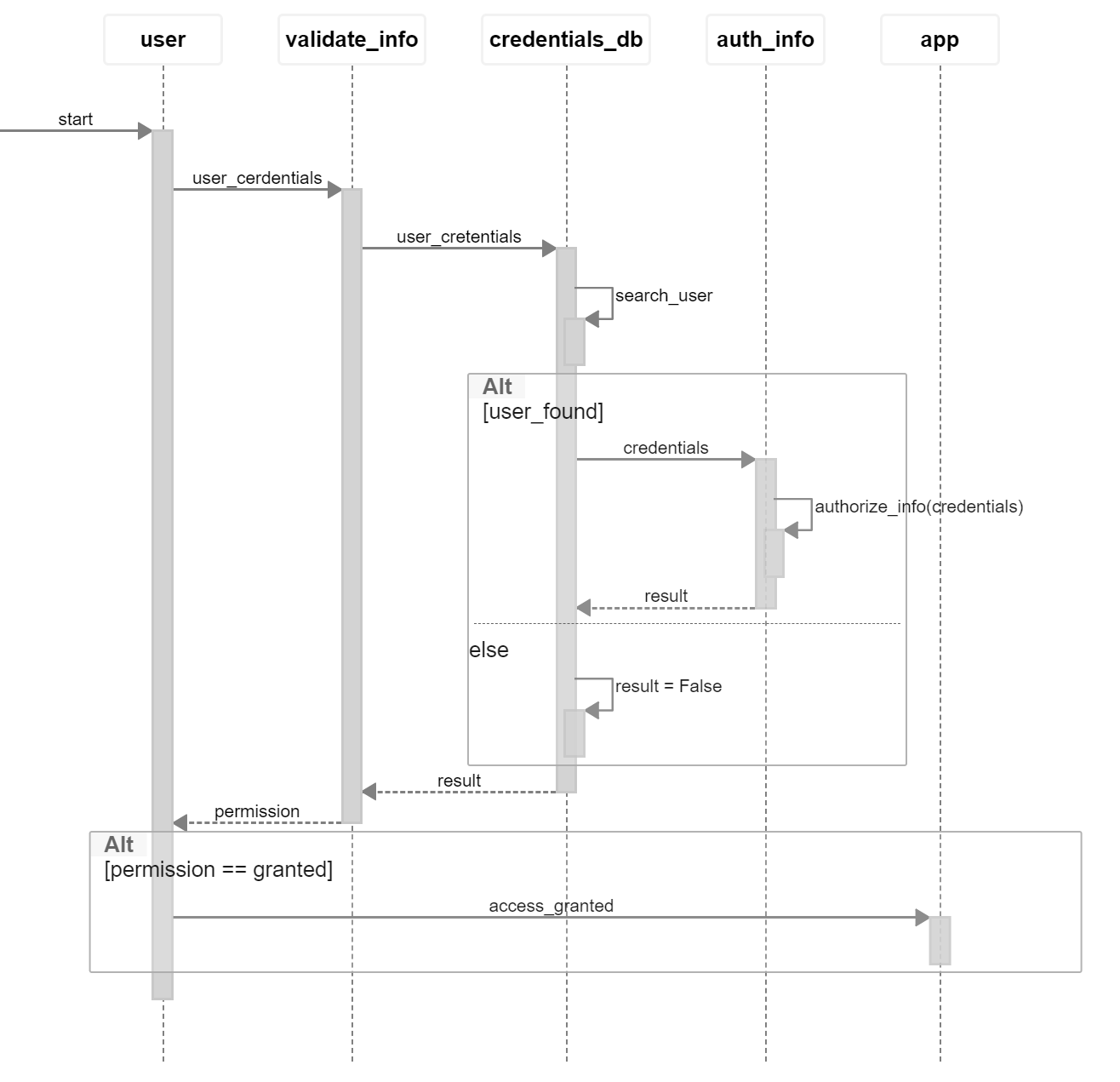
* External Requirements:

1. User data should be protected (encrypted)

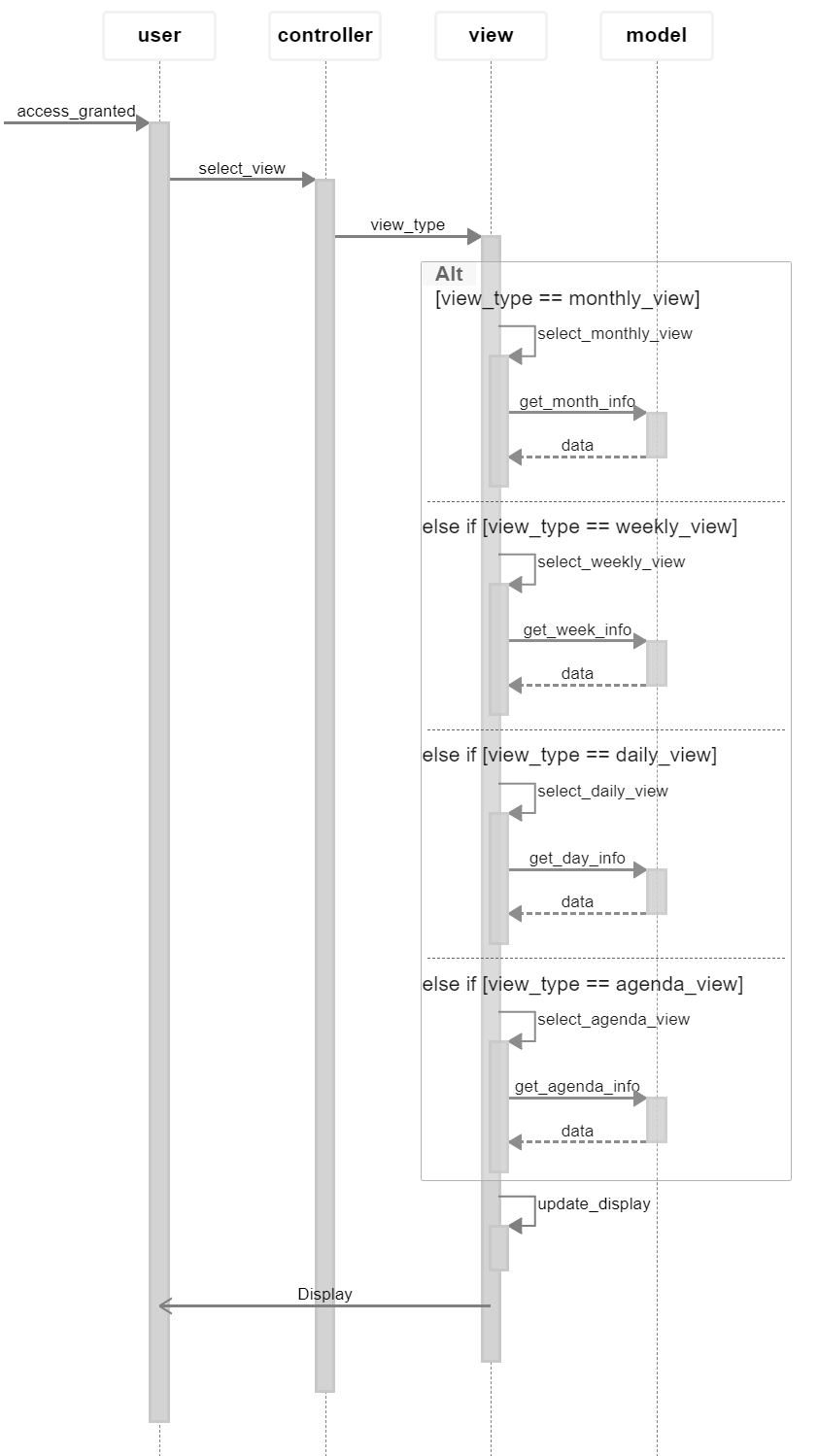
**4. Use case diagram**

**5. Sequence diagrams**

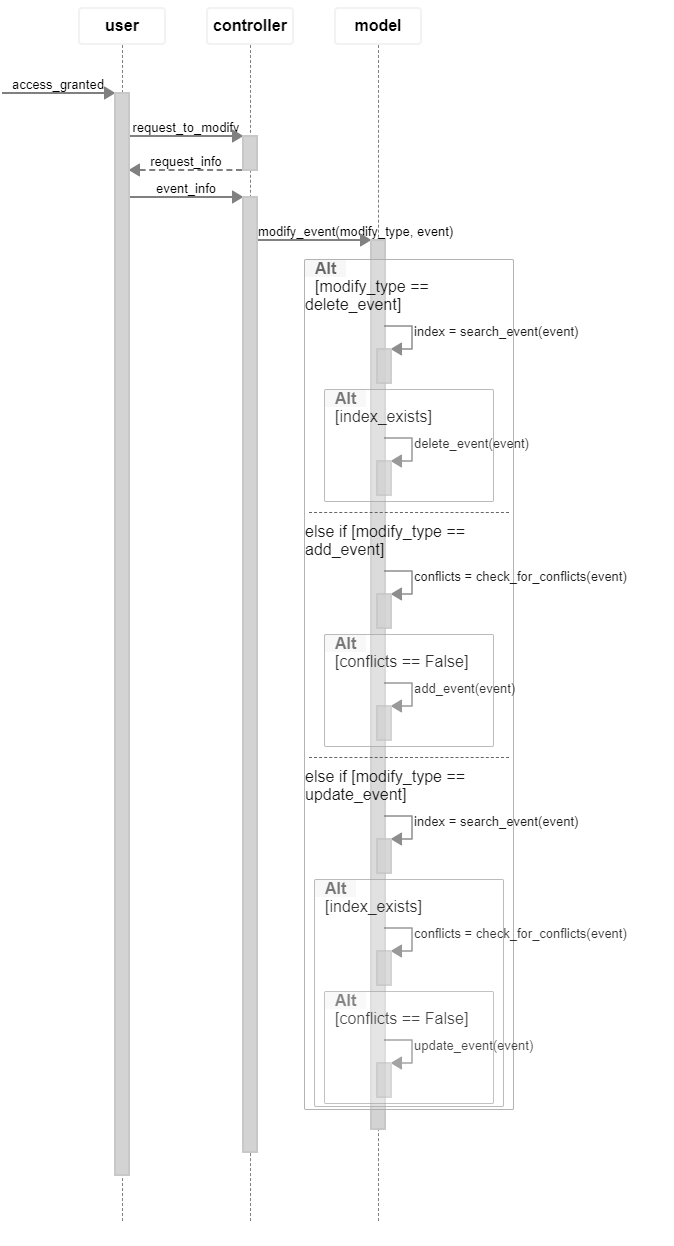
Login Sequence:



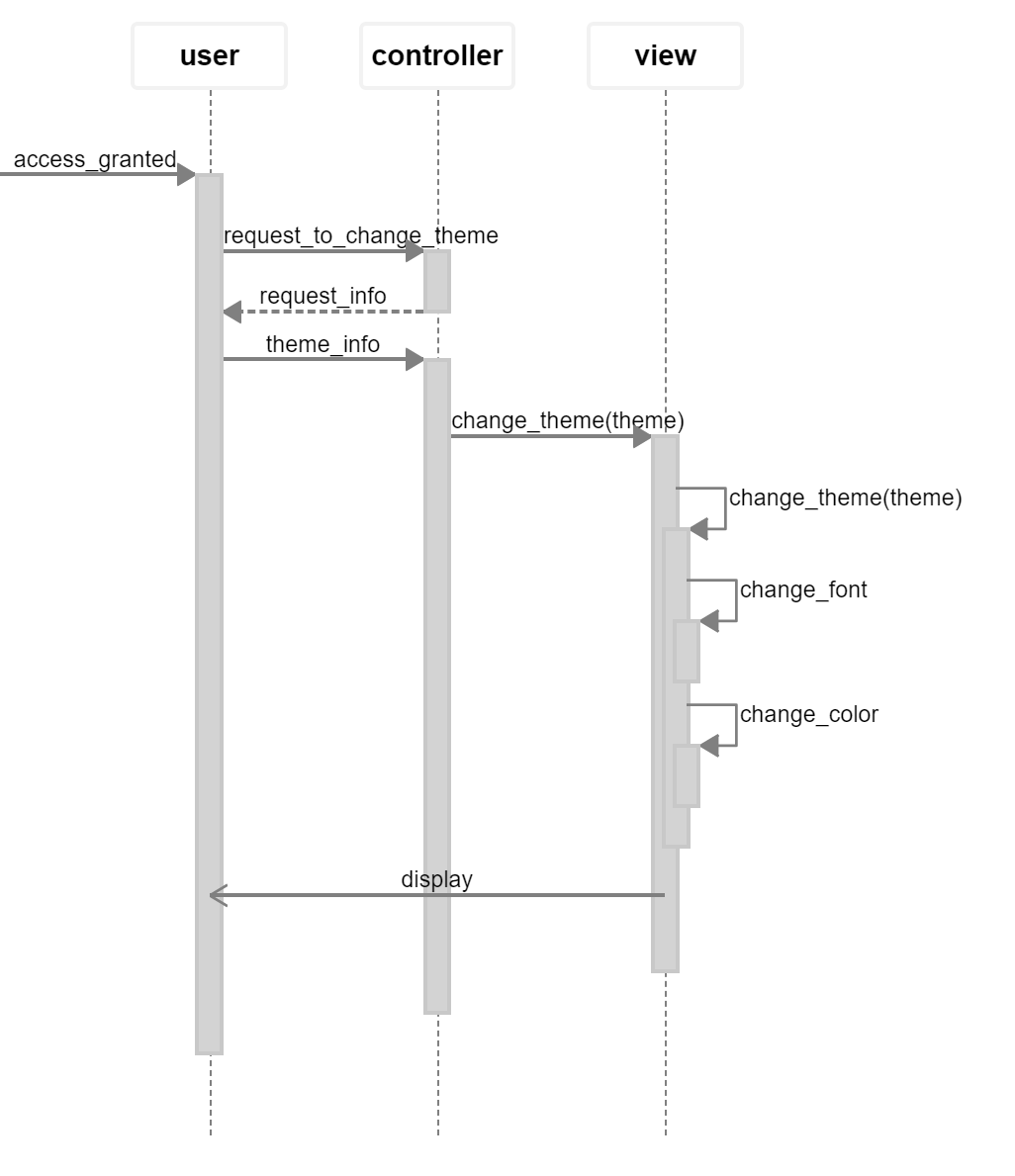
Update View Sequence:

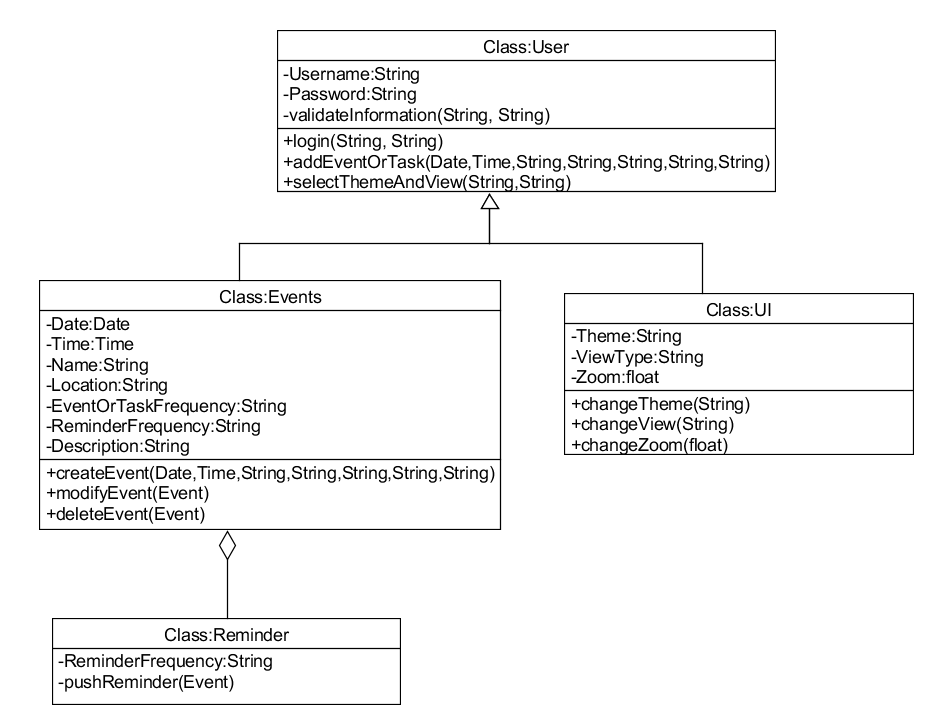


Modify Event Sequence:



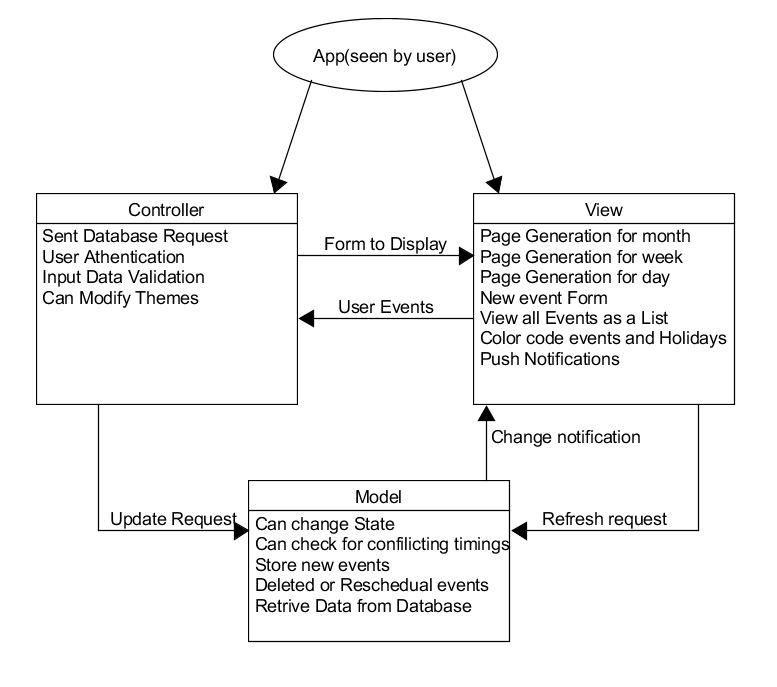
Update Theme Sequence:



**6. Class diagram** 

**7. Architectural design Model-View-Controller (MVC) pattern (similar to Figure 6.6)**

We chose the MVC architectural design for this project since we are making a calendar, which requires us to display the same information(Events, Reminders) in different forms(Days, Weeks, Months). because of the system’s ability to for data to change independently without affecting the representation and vice versa and the presentation of the data being supported in different ways with a change made in one representation shown in all of them. The digital calendar is perfect for pattern are the data will be displayed in month, day and year and data can be changed independently such as the addition of events and tasks.

****