CS 3354 Software Engineering

Final Project Deliverable 2

Group Number 4

Comet Calendar

Soundarya Baskar, Varsha George, Sarah Hasan, Allen Hernandez, Anchal Sachdev, Anusha Saha, Don Thai

Group Number: 4

Title of Project: Comet Calendar

Link to project repository: https://github.com/vxg190029/3354-CometCalendar.git

Group Members: Soundarya Baskar, Varsha George, Sarah Hasan, Anchal Sachdev, Anusha Saha, Don Thai, Allen Hernandez

Delegation of Tasks:

Soundarya Baskar

- Committed README file into the repository
- Created the functional and non-functional requirements
- Created a test plan for our software to check if the user enters a valid date for creating an event
- Committed JUnit method and test files into the repository
- Edited presentation video and uploaded to Youtube

Varsha George

- Created GitHub repository
- Added team members and TA to repository
- Created the use case diagram1
- Created Project Scheduling plan
- Created Algorithm Estimation Method
- Uploaded Final Report to Github

Sarah Hasan

- Planned/discussed with group in meetings for tasks
- Created the use case diagram2
- Conclusion

Allen Hernandez

- What you are doing / planning to do regarding the feedback provided for your project proposal
- Created sequence diagrams corresponding to the second user case diagram.
- Helped with the conclusion, objective of project, and references.

Anchal Sachdev

- Decided on the software process model
- Created Architectural Diagram (MVC)
- Worked on comparison of our work with similar designs (Apple Calendar, Notion)

Anusha Saha

- Created the class diagram
- Worked on the comparison of our work with similar designs
- Uploaded presentation slides to Github

Don Thai

- Made another commit including a pdf file named project_scope. Made the contents of the file identical to the corresponding project of the predefined topic
- Assisted in describing what we are doing / planning to do regarding the feedback provided for your project proposal.
- Made individual sequence diagrams for several use cases
- Estimated cost of software products
- Estimated cost of personnel
- Estimated cost of hardware products

Project Deliverable 1 content:

Instructor Feedback:

Good choice for a topic as it promises to facilitate students' planning efforts for their courses. It is great to see a detailed break down of the tasks fairly delegated. Good job. In the final report, please make sure to include comparison with similar applications -as there are plenty-, make sure that you differentiate your design from those, and explicitly specify how. Please share this feedback with your group members. You are good to go. Have fun with the project and hope everyone enjoys the collaboration.

List what you are doing / planning to do regarding the feedback provided for your project proposal.

- In the process of making the designs for our project, we will compare with existing designs in order to create a different, more functional design that implements a better UI with more efficient use of software processes. Moreover, we will explicitly state where they differ from the average design.

Which software process model is employed in your project and why?

- We will be using the Incremental Process Model**.** The reason we are using this is because it is in line with how the team plans on implementing the Comet Calendar. We would like to implement our individual parts of the calendar in an incremental fashion with an emphasis on communicating and planning our individual tasks.

List of software requirements including:

- **Functional requirements**
    - A user shall be able to view the monthly, weekly, daily, and overall agenda view along with the snippet for each day.
    - A user shall be able to add, edit, and delete events to the calendar.

- The system shall send an event alert at the time of the event.
- The system shall check for time conflicts when adding events.
- A user shall be able to share the event to other calendar users through the internet.
- The system shall color code holidays and weekends.
- The system shall support zooming in/out and scrolling through the calendar.

Non-functional requirements

- Product Requirements
  - Efficiency
    - The system shall be able to handle 1000 events per year without it affecting its performance.
  - Performance
    - The system shall take 1.5 seconds to load the entire calendar with events.
  - Space
    - The software should not exceed more than 1GB of storage as a whole.
  - Dependability
    - The system shall not exceed 5 seconds of downtime during the day.
  - Security
    - The system shall not share any information provided by the user, including personal information and calendar information without permission.

    - Usability
        - The system should be easy to use by everyone and should be organized in a way that user errors are minimized

- Organizational Requirements
    - Environmental
        - The system shall be able to run on operating environments of Windows, Mac OS, Android, and iOS.
    - Operational
        - The system shall be used to view the days of the year and organize events throughout the year according to the user.
    - Development
        - The software shall be created using Java programming language.

- External Requirements
    - Regulatory
        - The system shall be approved by Apple and Android and follow their privacy policies before publication.
    - Ethical
        - The system shall keep all information confidential and will only share information with permission.
    - Legislative
        - The system shall implement user privacy laws as set in The Federal Trade Commission Act (FTC)[1914].
    - Accounting
        - The system shall be created within the funds given to create the software.

- Safety/security
  - The system shall keep all user data safe and protected through various means of data protection.

Figure 1:

Figure 1 depicts a Use Case Diagram

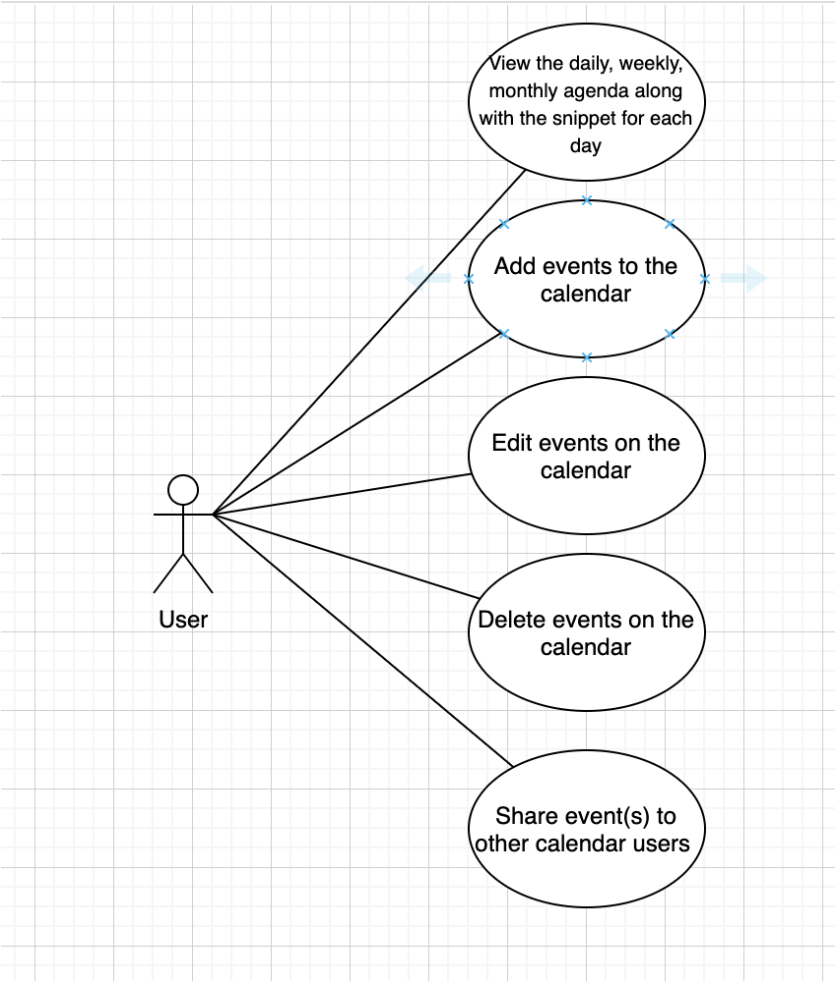


Figure 2:

Figure 2 depicts a Use Case Diagram

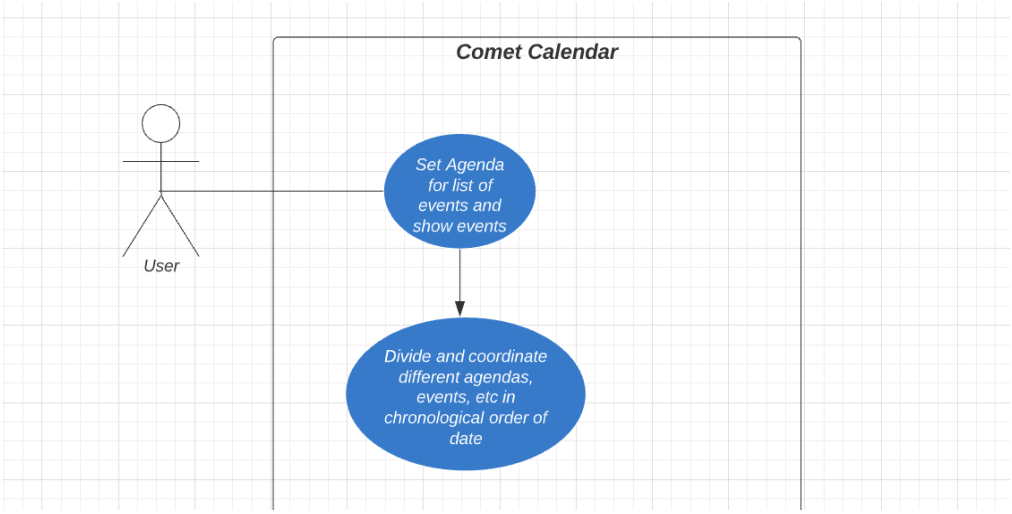


Figure 3:

Figure 3 depicts a Sequence Diagram

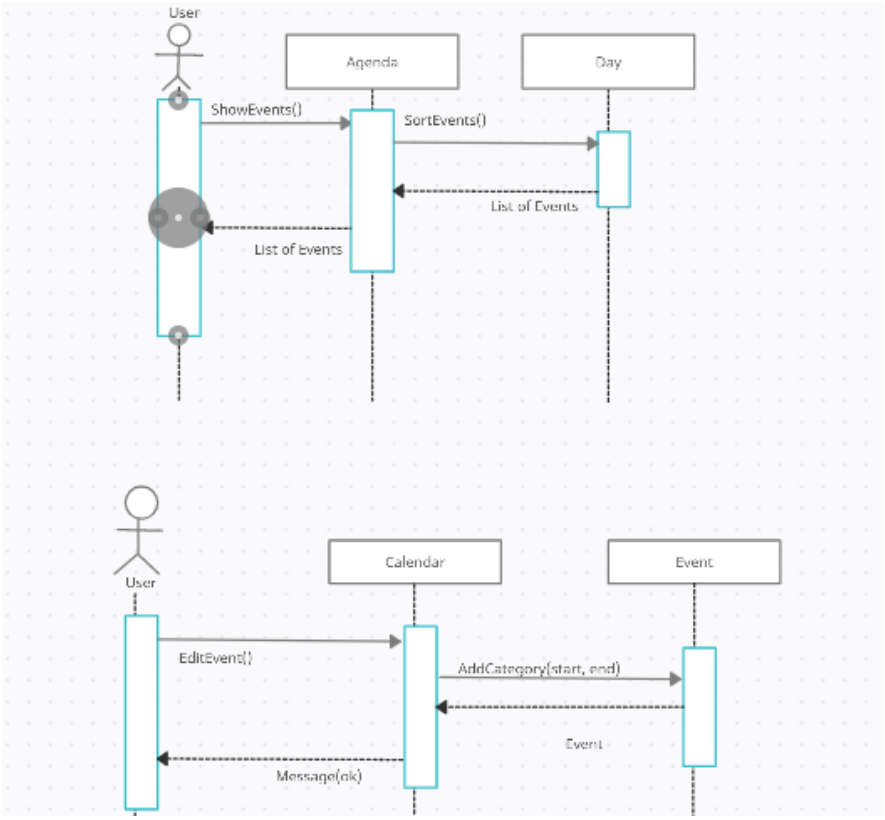


Figure 4:

Figure 4 depicts a Sequence Diagram

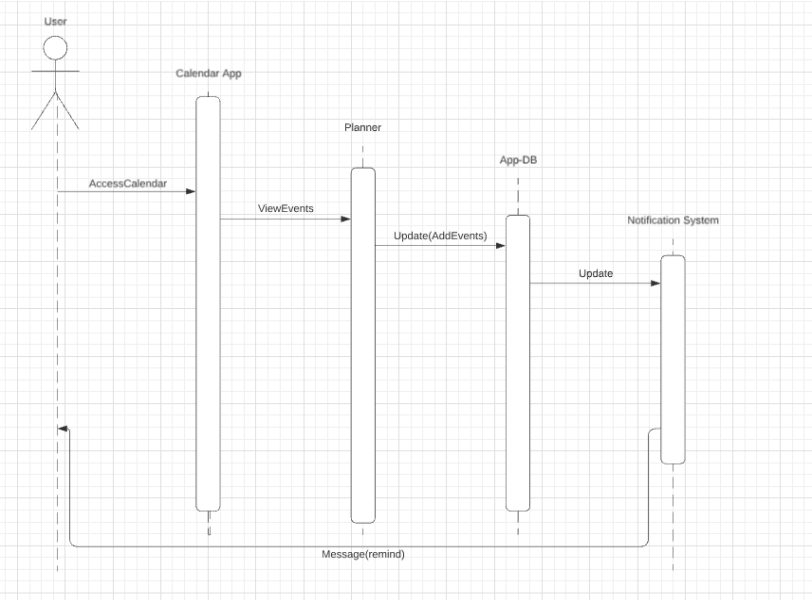


Figure 5:

Figure 5 depicts a Sequence Diagram

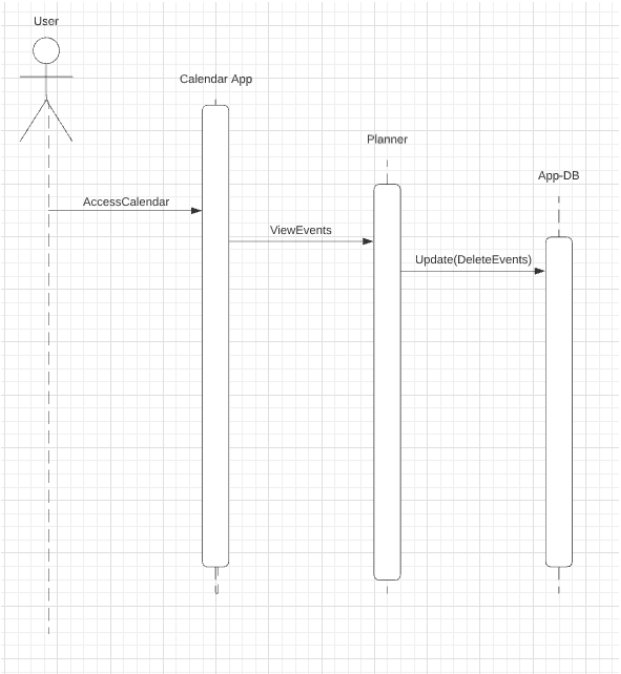


Figure 6:

Figure 6 depicts a Sequence Diagram

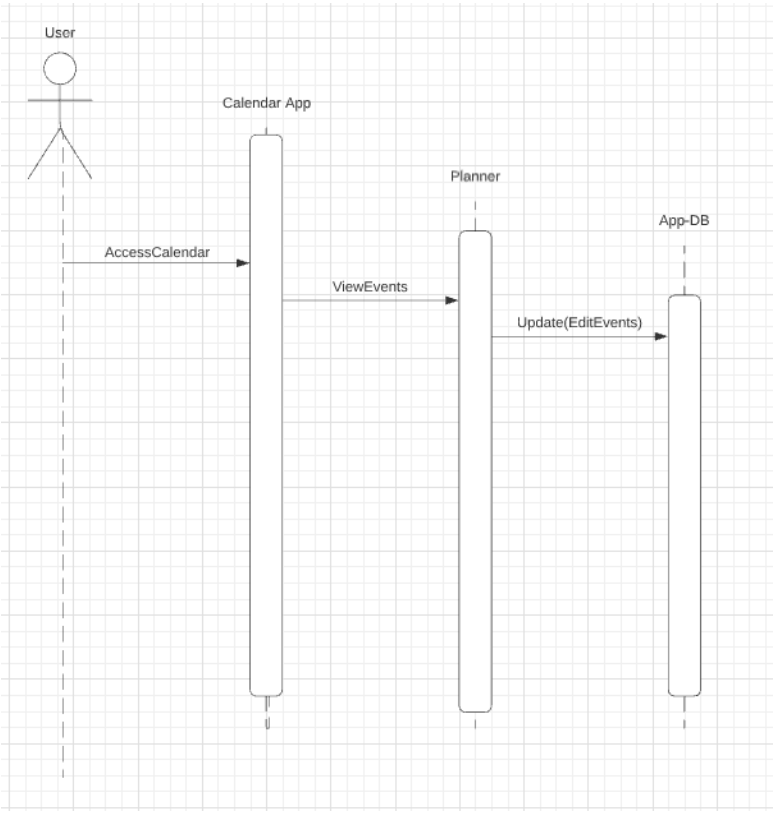


Figure 7:

Figure 7 depicts a Sequence Diagram

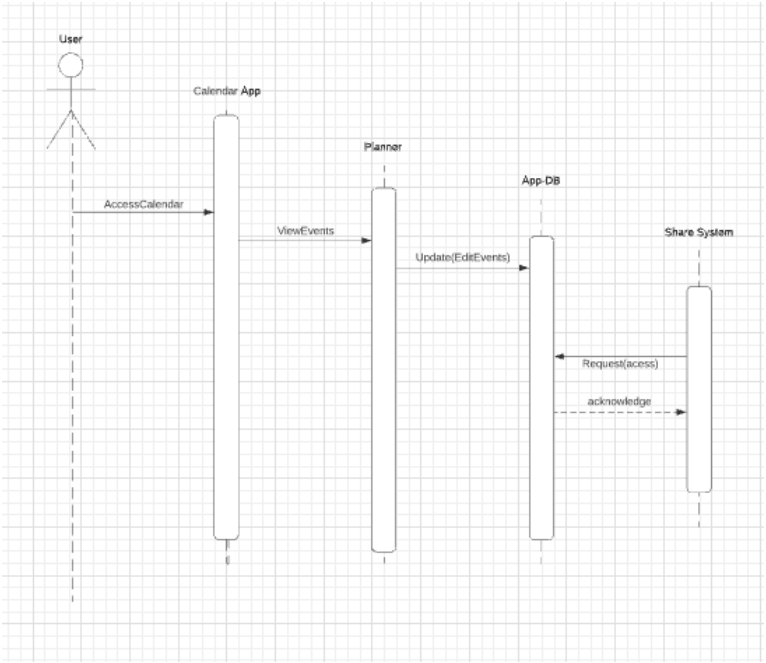


Figure 8:

Figure 8 depicts a Class Diagram

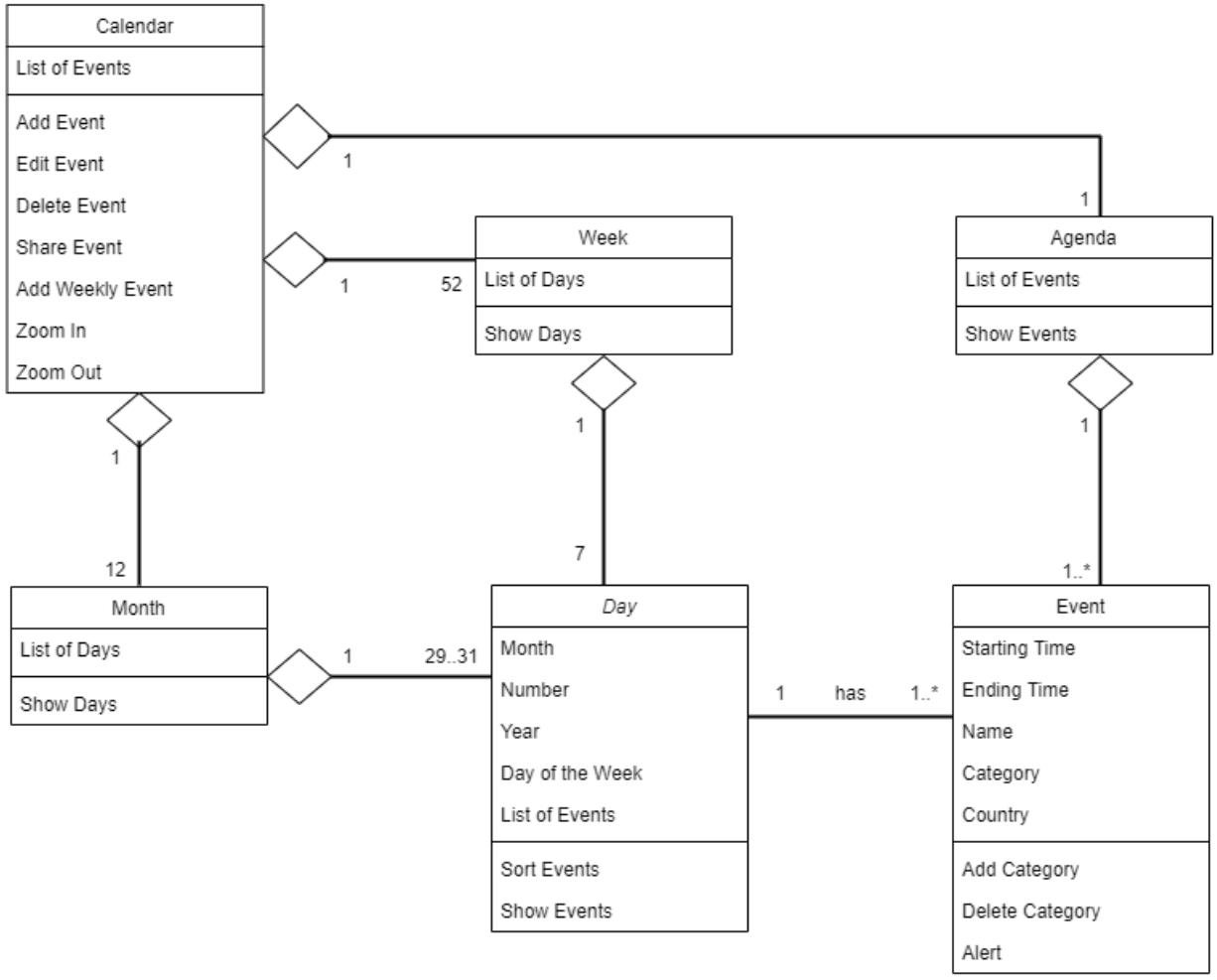
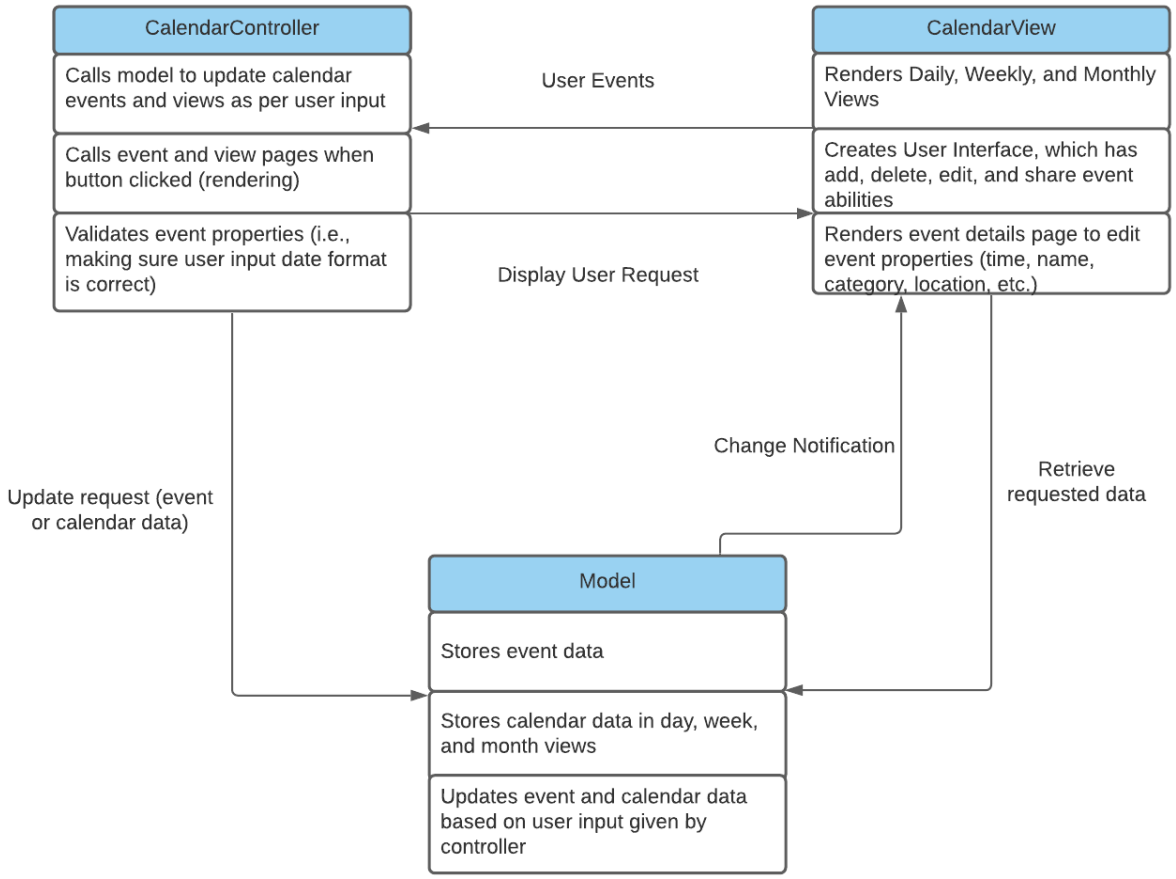


Figure 9:

Figure 9 depicts the Architectural design: Model-View-Controller (MVC) pattern



Project Deliverable 2 Content:

Option 2: Pre Recorded presentation

URL: https://www.youtube.com/watch?v=Ja330PWQYsE

Project Scheduling, Cost, Effort, and Pricing Estimation, Project Duration and Staffing:

- Project Scheduling:
    - The start date of this calendar app will be Jan 10th, as the group will present the final project presentations on December 7th. After the presentation, there is winter break, so the start date will only be in January. Creating this app will take around 370 hours, as there are many components and features that have to be taken care of. Considering the group will work 20 hours/week on the app, the app will take around 17 weeks to build; 17 is the exact number of weeks it will take to build the app, but given some off days and holidays, the group will finish the app in 18 weeks. This project schedule will be from January 10, 2022 to May 13, 2022, excluding weekends.
- Cost, Effort and Pricing Estimation
    - Function Point (FP)
        - The algorithmic estimation method that will be used in this project is the Function Point method. This method will be very beneficial for this project for a few reasons. The FP method is used to measure the project development along with maintenance throughout the entire project, which is very much needed for a project as complex as this one. This is the perfect method to display the amount of time the project will take and

how much each person in the group will have to work on the project.

Figure 10:

Figure 10 depicts a Function Point Table

Function Category	Count	Simple	Average	Complex	Count x Complexity
number of user entries	10	3	4	6	40
number of user outputs	12	4	5	7	48
number of alerts	3	3	4	6	9
number of data files	18	7	10	15	126
number of views	4	5	7	10	28
				GFP	251

Algorithmic Estimation Technique: Function Point (FP) Method

1. Determine function category count
    - number of user entries = 10
    - number of user outputs = 12
    - number of alerts = 3
    - number of data files = 18

- number of views = 4

2. Determine complexity
   - The complexity for the number of user entries and views are average. The complexity for the number of user outputs, alerts, and data files are simple.

3. Compute gross function point (GFP)
   - GFP = 10 x 4 + 12 x 4 + 3 x 3 + 18 x 7 + 4 x 7 = **251 FP**

4. Determine processing complexity (PC)
   - The processing complexity values for the 14 questions are all average, except the performance and user-friendliness are essential.

5. Compute processing complexity adjustment (PCA)
   - PCA = .65 + 0.01 x (12 x 3 + 2 x 5) = **1.11**

6. Compute function point (FP) using the formula: FP = GFP × PC
   - FP = GFP × PCA = 251 x 1.11 = **278.61 FP**

The estimated effort is:

E = FP/productivity = 278.61/50 = 5.57 ≈ 6 person-weeks

Team size = 7, project duration is: D = E / team size = 6/7 ≈ 1 week

- Estimated cost of hardware products (such as servers, etc.)
  - For a project on this level, a business server is required, and most business servers will generally need $1000 to $2500 per server for enterprise-grade hardware. We also need to consider buying a server

rather than renting one, which will need more hardware and we will need to consider costs outside of just the server, such as server hardware. This includes, but not limited to the following: CPU, hard drives, RAM, chassis, motherboard, processor power supply. There's also other costs to consider, such as server maintenance and replacement, electricity and cooling costs, as well as hardware support. With all of these factors considered, the total cost per server could be closer to the $2000 territory.

- Estimated cost of software products (such as licensed software, etc.)
    - The cost to build a calendar app on average costs about $27,500 but can go even higher, as high as $30,000 as the price fluctuates depending on how many features it has. There is a method that can potentially cut the cost of developing this app; there is a high chance that someone has created a version of a calendar application and is selling the source code for that app for $50-$150 per license , meaning that we can publish it to the app store once, but we will still have to adjust the app template as well as the UI to fit the needs of the Comet Calendar, but we are developing the software ourselves. Also to make it available on the app store such as the Apple app store, the Apple Developer license is required which costs $99 per year and must be renewed annually in order for the app to remain in the store. For Android, the price is $25, but this is a once off payment and does not need to be renewed. This calendar app has potential risks as it has software handling user sensitive information, such as user locations, user appointment details, and other private information the app would need software to protect this information like firewalls or software to prevent malicious attacks which can cost $100 – $500 per month for a small-sized network to $500 – $2,000 per month for a medium-sized network.

- Estimated cost of personnel (number of people to code the end product, training cost after installation)
    - Developing an app specifically a calendar app is a process that is best when there are between 5-7 developers working on it to ensure a user-friendly application. The necessary members of this team for a successful application are a back-end developer and a front-end developer, maybe more than one of each as they are essential for the app's main operational capabilities, while a UI designer and QA specialist are needed to provide a well-functioning product and ensure user satisfaction, lastly a project manager is needed to tie everything together. The hourly rates of the development team and the design team ranges from $150 to $250 per hour per specialist, and seeing as how we have 7 specialists and a 370 hour project, each specialist would cost about $55,000-$95,000. A DevOps is also needed in order to manage the system during business hours after the app is up and functional, which considering if they are paid at a rate of $50/hour that'll be $6,000/week.

Test plan for the Software:

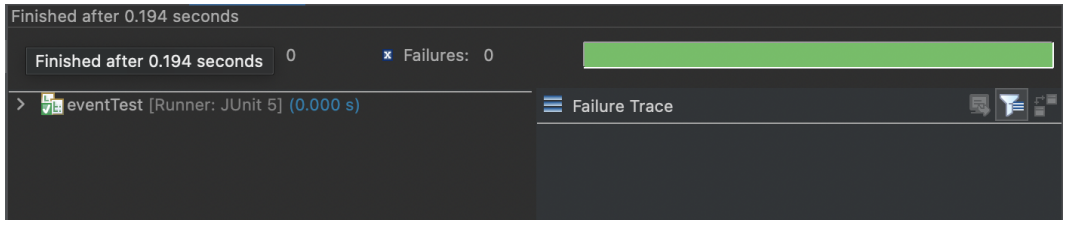
- The test plan for our software is to check if the user enters a valid date to create an event. This method validates if the month and the date are present on the calendar, making sure the month and date match up. If the month and/or date are invalid, such as not present or the user enters a negative or decimal number, then it will also return false. If it is valid, it will return true. If the method returns false, it means the event was not created. If the method returns true, it means the event was created.

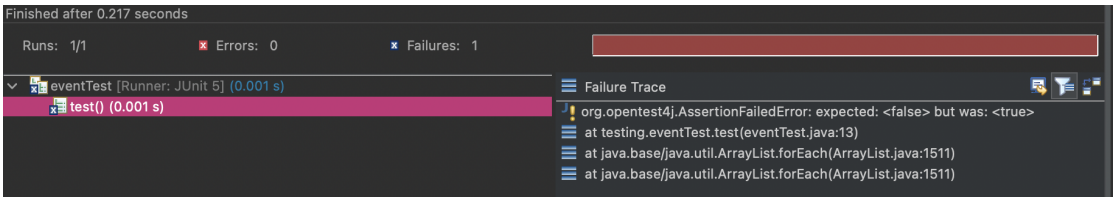
- This method checks if the month is valid and if the date provided is within the specific month. It differentiates between months that have 30, 31, and 28 days. All other inputs, including decimal, negative, and invalid numbers will return false.

```
1  package testing;
2
3  public class JUnitTesting {
4
5      public boolean event(int month, int day) {
6
7          // odd months have 31 days
8          // even months have 30 days, except 2nd month with 28 days
9          // checks if month and date are valid
10         if( (month == 1 || month == 3 ||
11            month == 5 || month == 7 ||
12            month == 8 || month == 10 ||
13            month == 12 && day <= 31) ||
14
15            (month == 4 || month == 6 ||
16             month == 9 || month == 11 && day <= 30) ||
17
18             (month == 2 && day <= 28) )
19             return true;
20         else
21             return false;
22     }
23 }
```

- Valid Case:
  - This test case checks if the input 1,31 is valid. 1,31 stands for January 31st. This test case returned true because it is a valid date present in the year.

```
1  package testing;
2
3  import static org.junit.jupiter.api.Assertions.*;
6
7  class eventTest {
8
9      @Test
10     public void test() {
11         JUnitTesting test = new JUnitTesting();
12         boolean output = test.event(1,31);
13         assertEquals(true, output);
14     }
15
16 }
```



- Invalid Case:
  - This test case checks if the input 2,28 is valid. 2,28 stands for February 28th. These values are expected to return true, but the test case is expecting false to check if the method works. This test case failed because the test case expected false, and it returned true. The date 2,28 is true. So the method works perfectly.
  -
```
1  package testing;
2
3  import static org.junit.jupiter.api.Assertions.*;
6
7  class eventTest {
8
9      @Test
10     public void test() {
11         JUnitTesting test = new JUnitTesting();
12         boolean output = test.event(2,28);
13         assertEquals(false, output);
14     }
15
16 }
```
  - 

Comparison of your work with similar designs:

- Google Calendar is another popular calendar application with many similarities to Comet Calendar. Like Comet Calendar, Google Calendar provides agenda (called schedule), daily, weekly, and monthly views. However, Comet Calendar doesn't provide yearly or "4 days" views like Google Calendar does. Both Comet Calendar and Google Calendar allow users to add an event with starting and ending time and add weekly periodical events. But Google Calendar also allows events to repeat daily, monthly, yearly, and on a custom basis. Comet Calendar only allows events

to repeat weekly. Both Google Calendar and Comet Calendar allow users to edit and delete events. Both apps send event alerts beforehand. But Google Calendar allows users to choose how much in advance the alert is sent. Both apps also allow users to share events with other users. Both apps mark holidays in special colors. Only Comet Calendar allows users to add or delete event categories and marks the weekend in special colors. Comet Calendar also allows zoom in and out functionality, unlike Google Calendar. Google Calendar is more flexible than Comet Calendar as it provides more options for views and adding events. Comet Calendar is more accessible as it allows inclusive features such as zoom in and out.

- Many OSs provide a calendar app built-in to their systems. Apple Calendar is one of the most widely used examples of an in-built calendar app. Apple Calendar can be found on iPhones, Macs, iPads, and Watches. It also allows iCloud syncing, which allows you to sync up your calendars across all of the devices you own using your Apple ID. This is similar to the model we want to set up for Comet Calendar, where we can sync a student's calendar across all of the platforms the app is available on. However, the main difference lies in the fact that Comet Calendar requires an internet connection as it is an external app rather than a built-in app. Apple Calendar also has Daily, Weekly, and Monthly views which is something that Comet Calendar implements as well. Apple Calendar, however, shows a yearly view as well, which the Comet Calendar doesn't. Both Comet Calendar and Apple Calendar also allow for the addition of events that you can color-code, share invitations with other users on their respective platforms, set a location and time for, and add notes to. Unlike Comet Calendar, Apple Calendar does not have zoom-in/zoom-out abilities, but rather selectors for the daily, weekly, monthly, and yearly views.
- Notion is a popular tool that works as an interface to conglomerate whatever the user wants to put in one place. It is commonly used by college students

to creatively organize academia and information related to classes. Notion allows you to fully create your calendar, schedule, etc. However, it is difficult for those students who would simply like to populate their calendars with events. The “Calendar View” in Notion only allows for a Monthly view, as opposed to Comet Calendar, which allows for a daily, weekly, and monthly view. In order to create a Weekly View, you have to make a “Board View”, which can be updated every week, as if it were an agenda, but again, all tags and event information has to be manually added by the user. In order to create a list of events, you can add them into a “Table View” with the addition of a dates column, but you have to fully create events on your own, as opposed to Comet Calendar, which automatically adds events to your Daily, Weekly, and Monthly Views. Both Comet Calendar and Notion have filtering abilities, where you can filter by the class you are looking for. However, it is, again, strenuous to create tags and maintain them in Notion, as it is fully user-reliant. Comet Calendar already adds those features for you. Both apps also have color-coding abilities. The main advantage of Notion over Comet Calendar is that it has the ability to be a more visually pleasing interface as it is not premade and the look of the interface is based on the user.

Conclusion:

- Purpose
    - Our purpose is to create a simple yet high-functioning calendar for attending students at UTD to help prioritize and maximize complex schedules, TO-DO lists, and basic agendas throughout the week/month.
- Project flow
    - Baseline for our project is dependent on the non-functional/functional

requirements

- Changes made throughout the project
    - The number of entries for input/output users; our results categorizes our product as an average complexity.
- Evaluation
    - J-unit test was made and provided testing for important aspects of our calendar, which results were successful. As far as calculations go, we chose the Function Point Algorithm for estimated costs of hardware and software products, as well as personnel. The total Gross Function Point is 251 FP.

References

[1.] *Javatpoint.com*. [Online]. Available: https://www.javatpoint.com/software-engineering-functional-point-fp-analysis. [Accessed: 09-Nov-2021].

[2.] A. Horton, "How to build a server with costs considered," *Knowledge Base ServerMania*, 05-Jan-2021. [Online]. Available: https://www.servermania.com/kb/articles/how-much-does-it-cost-to-build-a-server/. [Accessed: 11-Nov-2021].

[3.] "How much does cyber security cost? common cyber security expenses & fees," *Proven Data - Trusted Solutions In Recovery, Cyber, and Forensics*, 11-Mar-2021. [Online]. Available: https://www.provendatarecovery.com/blog/cyber-security-cost-expenses-fees/. [Accessed: 11-Nov-2021].

[4.] "How much does it cost to build a calendar app?," *2021 Cost to Build a Calendar App*. [Online]. Available: https://www.crowdbotics.com/cost-to-build-app-type/calendar-app. [Accessed: 11-Nov-2021].

[5.] "How many people does it take to develop a mobile app?," *Appzoro Technologies Inc Atlanta*, 11-Sep-1970. [Online]. Available: https://appzoro.com/how-many-people-does-it-take-to-develop-a-mobile-app/. [Accessed: 11-Nov-2021].