The Fourth Dimension

Group 10

Mohamed Elkhaled
Muhammad Hamdan
Jose Francisco De La Cruz
Shahneel Lalani
Nicholas Vitale
Akhil Mathew
Elam Bruce

Introduction

A calendar software is a basic necessity found in almost every kind of device. Calendars seem to be a practical software that can be easily used by many people across different platforms. Since it is already being used by different companies, there is also a large amount of information out there that can assist us on how to do certain tasks or see how other programmer's approach an issue we may face. Our design can be seen used across multiple platforms since this type of software is already seen in most devices.

Cost Estimation for Hardware and Software

Hardware:

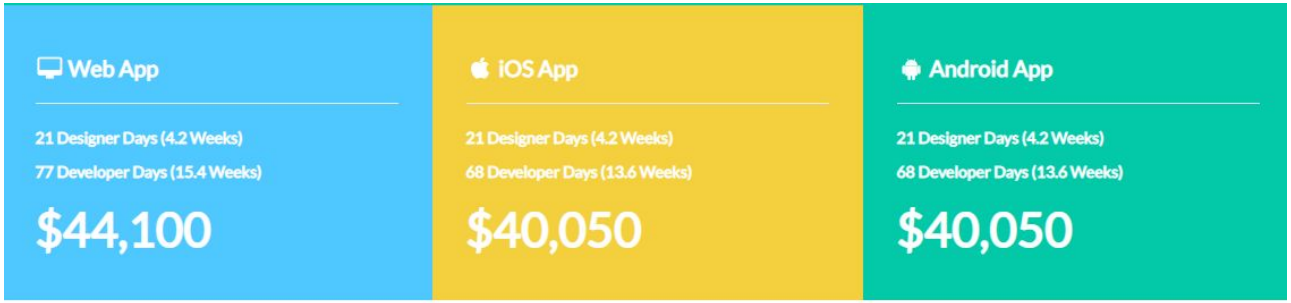
Estimated cost of hardware products including app infrastructure services (servers, hosting, domains), databases used for storage such as user data, and back-end infrastructure to allow for app scalability and integration: ~$250/month

Software

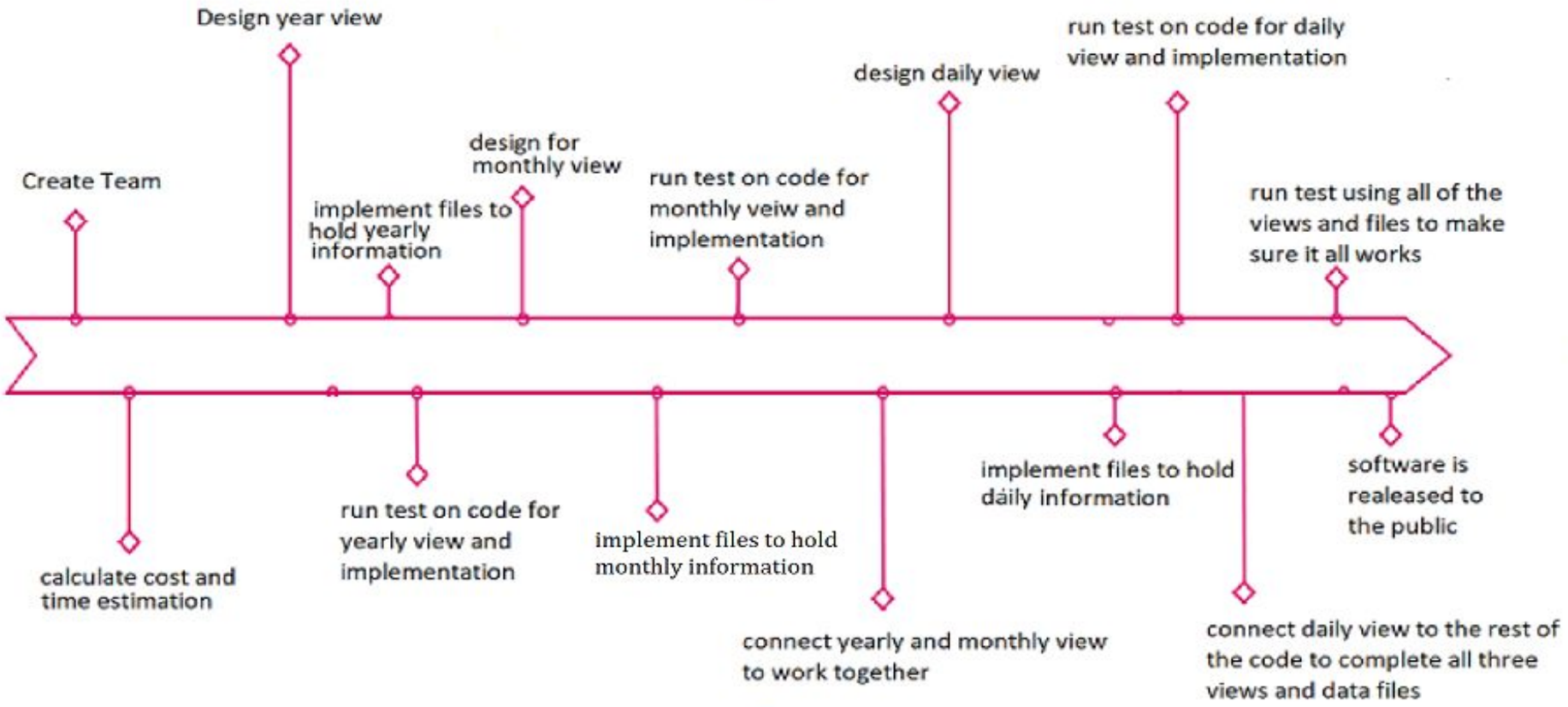
Due to the number of free IDE's available to choose from to develop our software application on and various open-source software to use, we would not incur any additional costs for software products such as fees for using licensed software.

Cost Estimation of Personnel

Based on details/features for a medium sized app with medium complexity, a polished UI level, email/password sign-up and Google sign-up, cloud syncing capabilities, calendaring, display of maps data/geolocation, ability for bookings within the app, social media sharing, push notifications, multilingual support, ability to connect to one or more third party services, performance monitoring, API for others to integrate with our app, two-factor authentication.



Project Timeline



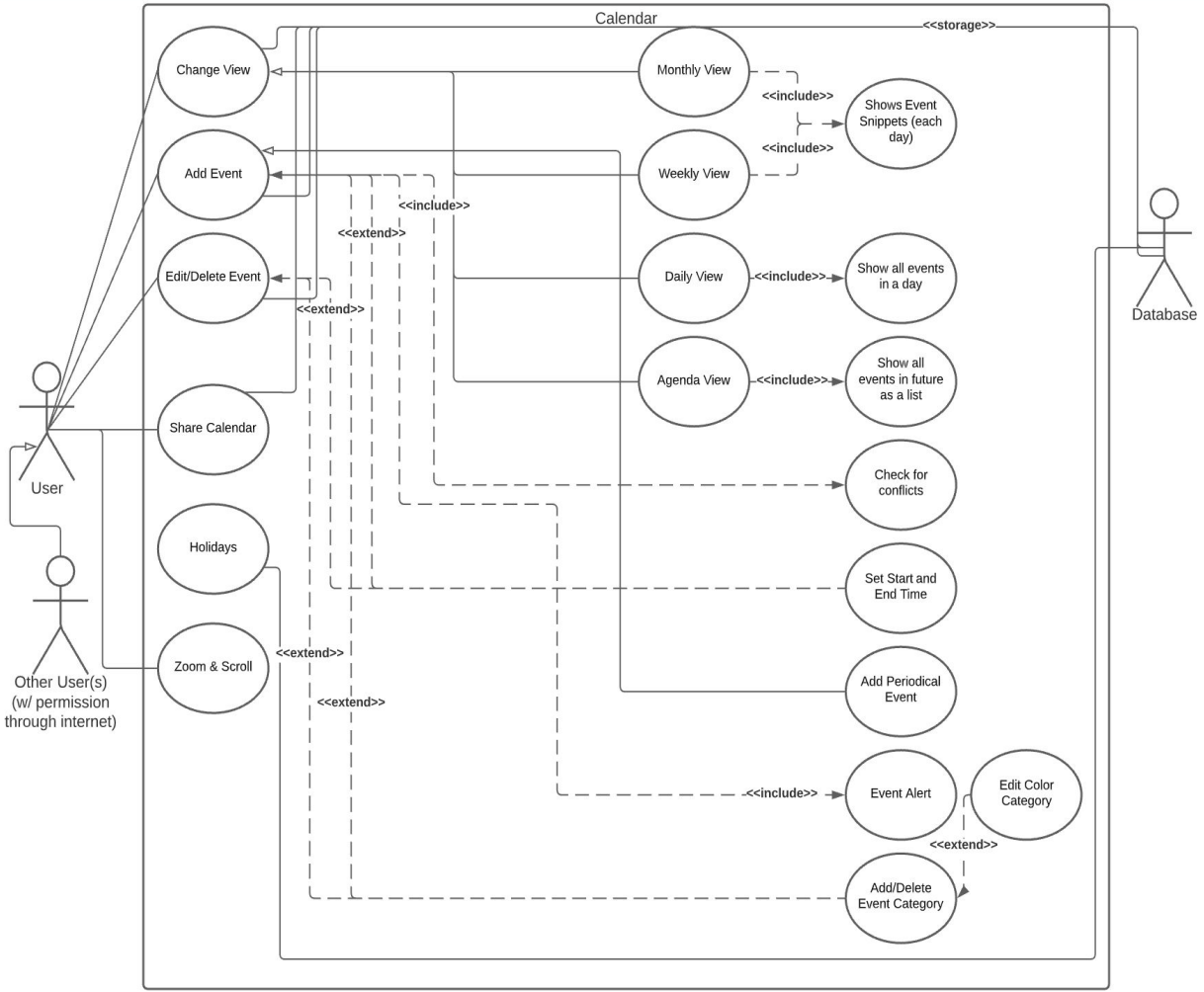
Functional Requirements

- A User will be able to add events on a certain date *
- A User will be able to check general holidays or events on any given date*
- The Calendar will save any and all events that the user adds
- The user will be able to plan his/her day hour by hour for each date*
- Users will be able to share their day plan through text or email
- This software will be available for download on any smart phone or PC

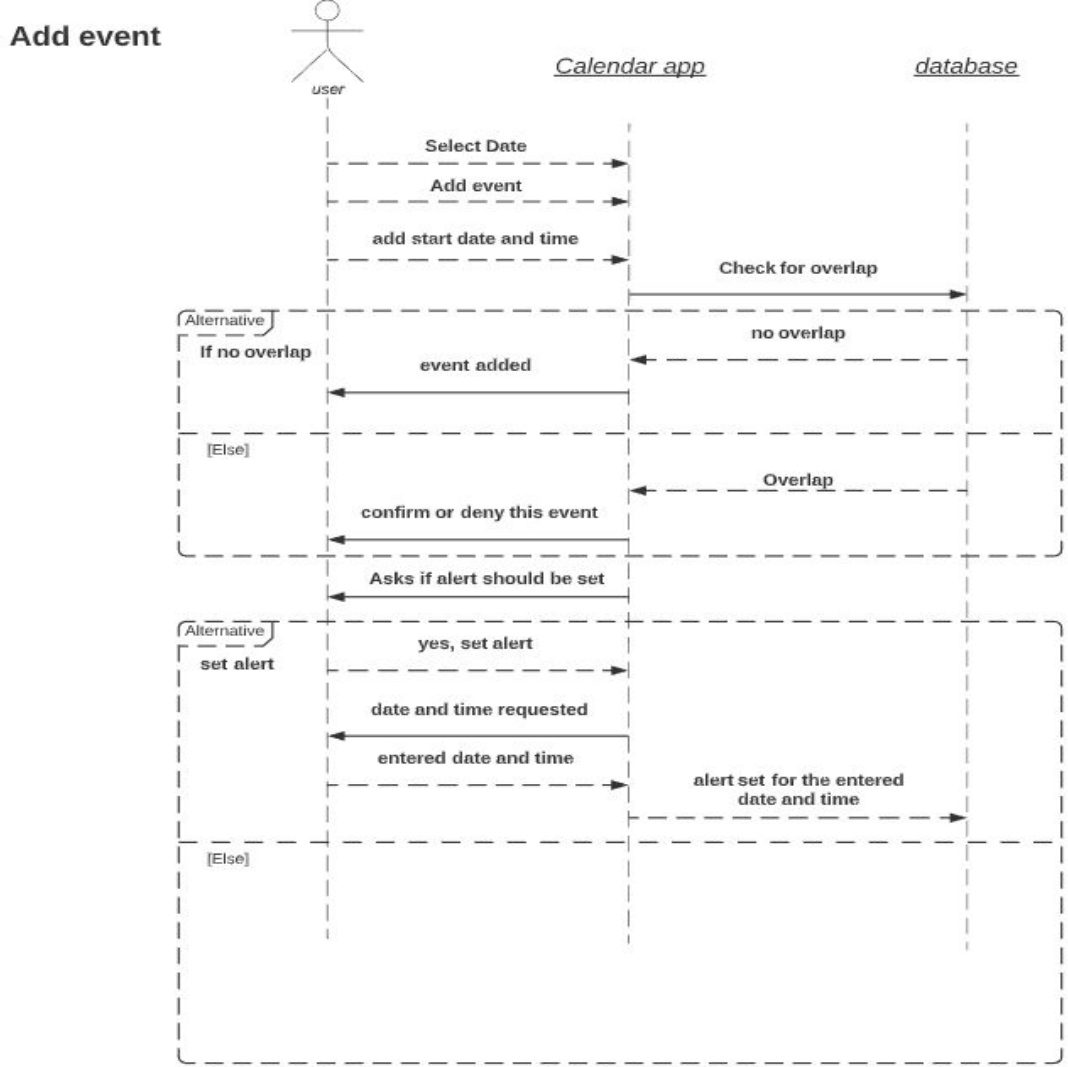
Non-Functional Requirements

- Usability Req: software will be able to move from PC to phone with no problem*
- Dependability Req: Software will auto update when one is available
- Security Req: software will require a password before opening
- Regulatory Req: ASSUMING IT'S FINE BY STATE LAW, The software can save data users put in their daily use for law enforcement benefits only *
- Ethical Req: software will not share any user data for a profit
- Environmental Req: Must have a smart device or PC to download
- Operational Req: when adding must be connected to wifi
- Development Req: when developing must have access to a MacBook for app store use
- Performance Req: Software will load up completely in 3 seconds*
- Space Req: Software will be relatively small to be able to be downloaded on any smart device*
- Accounting Req: ASSUMING NO LAWS STOP USERS: there will be no restrictions on who can download the software
- Safety Req: Per customer Safety the data collected will not be shared with anyone for any reason unless for legal issues

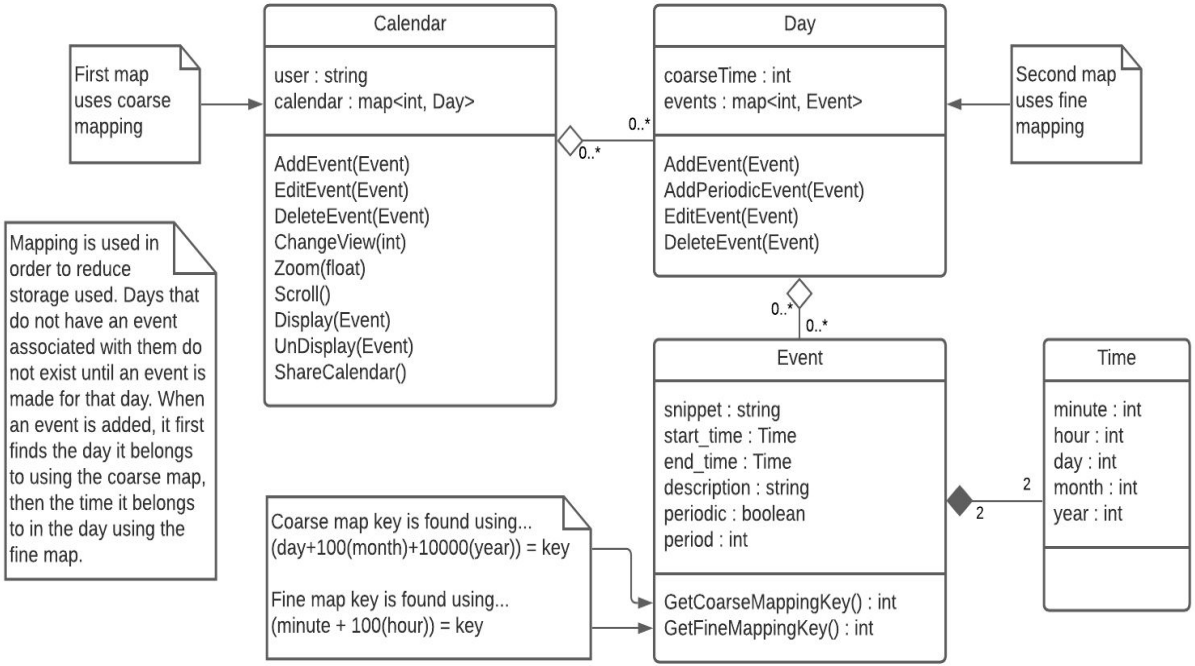
Use Case Diagram



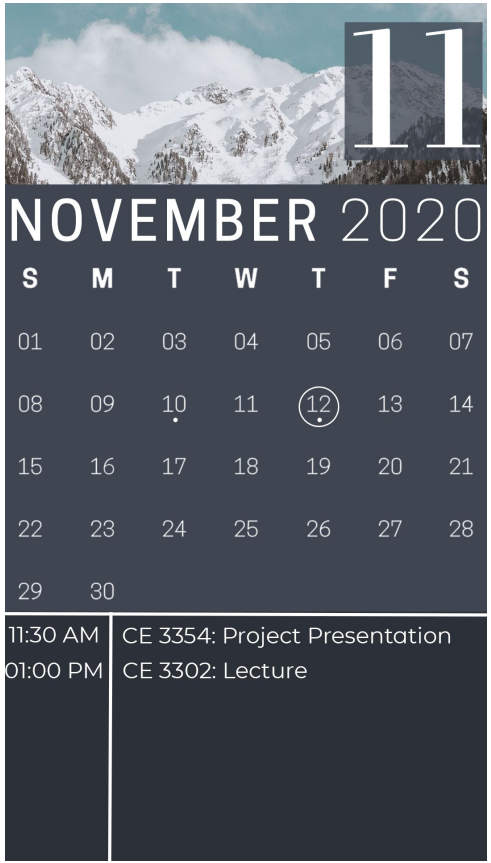
Sequence Diagram

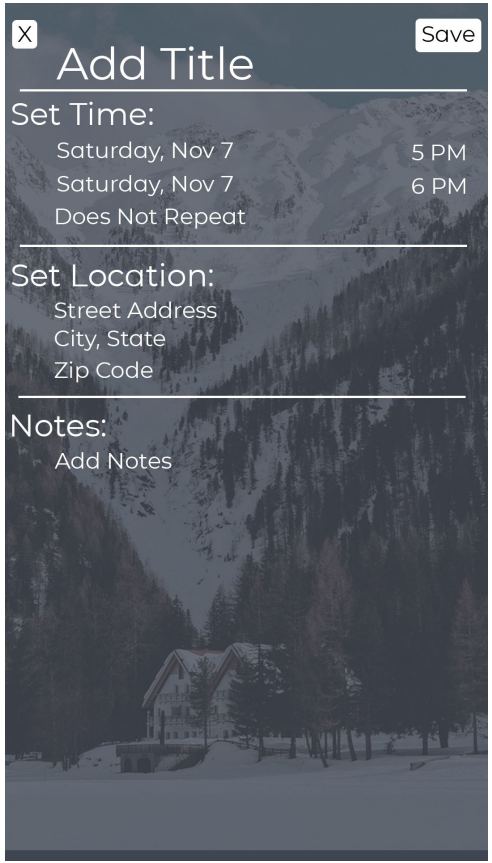


Class Diagram



User Interface Design





Architectural Design

Model-View-Controller (MVC) pattern

As we know, the Model-View-Controller (MVC) architectural pattern separates interaction and presentation with the system data. The 3 components that make up the MVC is the Model component which manages and associate's operations to a system data, View component which defines and manages how data will be presented to the user while updating its presentation if needed, and Controller component which manages user interaction. For example, Key presses, mouse clicks etc. Moreover, it passes these interactions to the view and the model. An example of the MVC pattern can be shown through a java application by implementing the model, view, and controller into separated classes which can be used to show a graphical web-based system. The strength of this pattern is that it allows data to change independently from its representation and supports the demonstration of the same data in different ways while making changes to one representation shown in all. The consensus that our group chose to do is the MVC architecture pattern because it is superior to other models for smaller, more focused applications. More so, we agreed that the view component of this architecture is very important for monthly, weakly, and daily views. Since the code we are not going to be implementing the code, the disadvantage of this pattern of being too complex will not be a concern. Another reason is because it has high cohesion and low coupling and allows use to work on code simultaneously yet separately.

Comparison To Other Work

- The 4th Dimension (4D) is an intuitive calendar application that allows users to experience the same task management skills as the world's best chief administrative officers.
- 4D divides the user's schedule into three main categories: daily, weekly, and monthly view, including an optional view called agenda.
- For example, if we use a Mac calendar we can have multiple accounts on one calendar and can manage different events in the calendar even if they are in different accounts like iCloud or Google.
- When it comes to User Interface or UI Design, apple uses color code to differentiate the work, family, or personal events.
- What makes our software different is users have the option to add driving routes to inform them of the best time to leave for their event.

Conclusion and Future Work

Overall, our project fundamentally remains the same as our original design. However, for our product to stand out from existing technologies, our team included driving routes when planning events. Although other calendars may provide warnings when events overlap, our product will ensure driving routes do not interfere with other events.

After careful consideration of our project's design, estimated cost and development time, our team concluded that the project was feasible and potentially a future endeavor we may pursue.