**CS 481 Final Project Mid Progress Report**

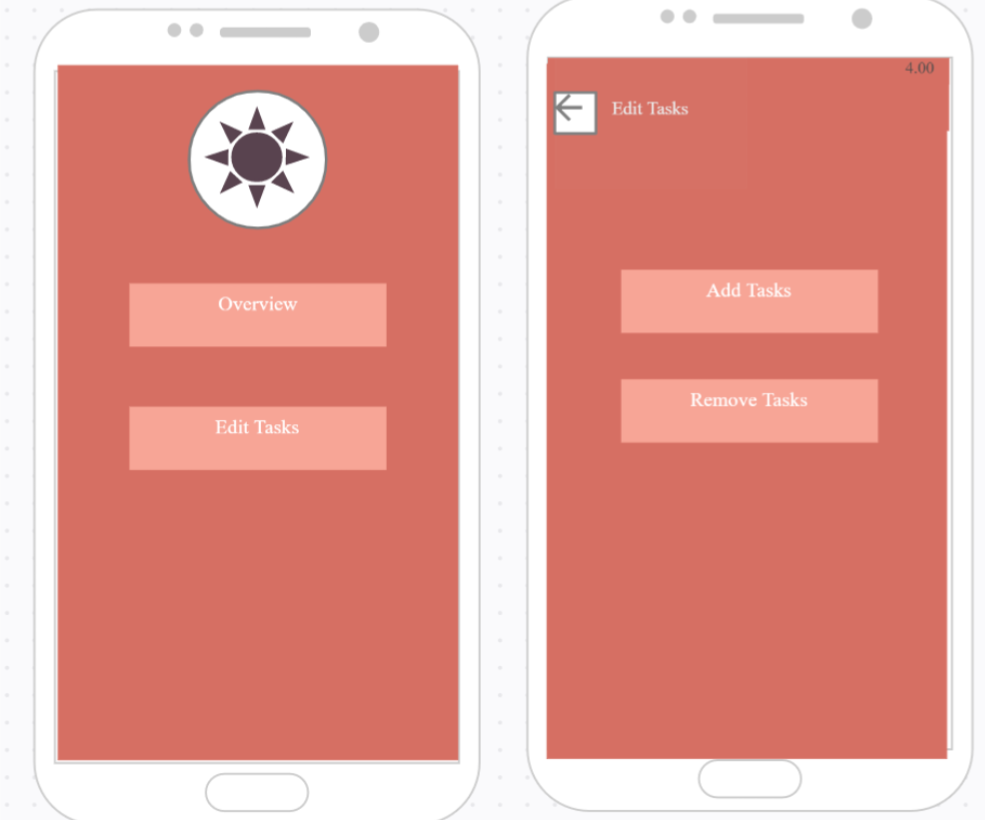
**Names: Ahmed Ali, Robert Paud, Ernest Rising, Adam Salinggih**

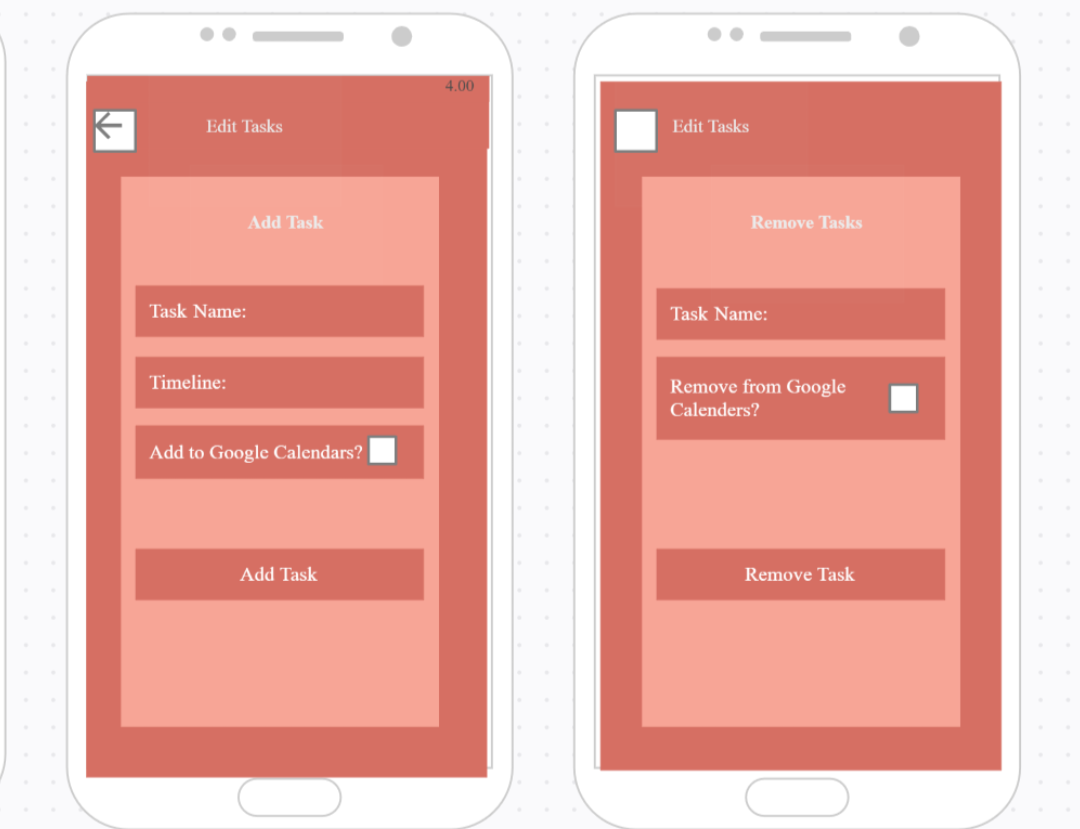
Application Summary

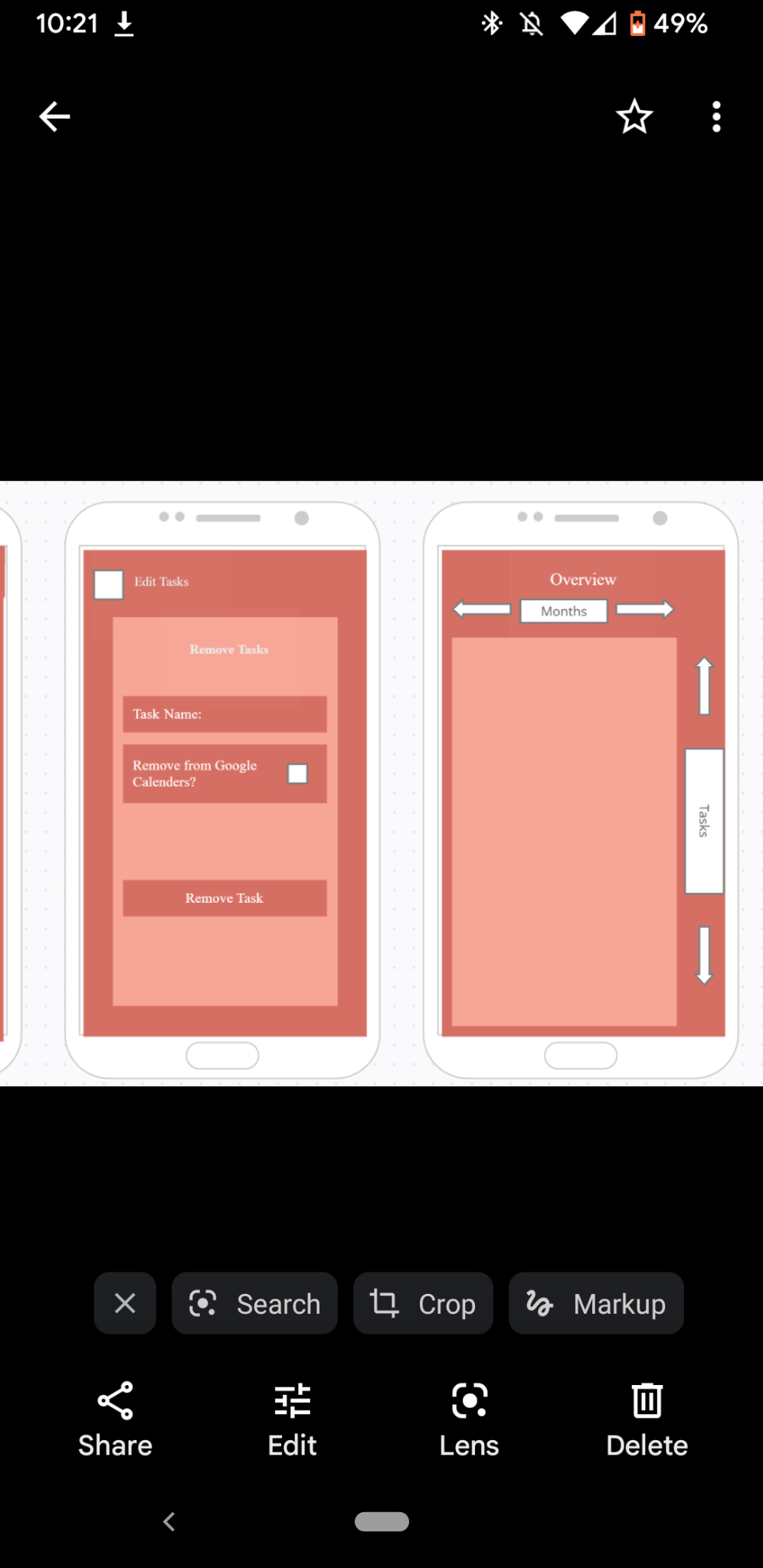
Our application will be a Task/Workflow overview planner. We will allow users to input their long-term goals/tasks into the app, and display them in a Gant chart format. It will show how long they have to complete any tasks based on their input and what other tasks are incoming as well. We plan to push notifications to the user when an upcoming task must be started, we will also incorporate Google calendar syncing via their API, and use a simple database design to store the users data.

Ahmed’s Part: UI, Transitions/Activities: Chart, Task Display, Overview

The initial UI of the application when a user launches the app will be a home menu screen. There will be an overview button to see their tasks laid out in a Gant chart style, it will be sorted by most recent user input, and shown horizontally alongside a month bar. On the home menu will also be a button that will allow the user to input any number of desired tasks into the application. We will ask for the task name, and the timeline they wish for it to be completed in. There will be a total of two activities, the home menu and the chart display/overview. This will be accomplished by the use of both fragments, for the task overviews, as well activities for transitioning between pages of our app. Also, through the use of floating buttons and normal buttons with conjunction to text boxes, this will allow user input.

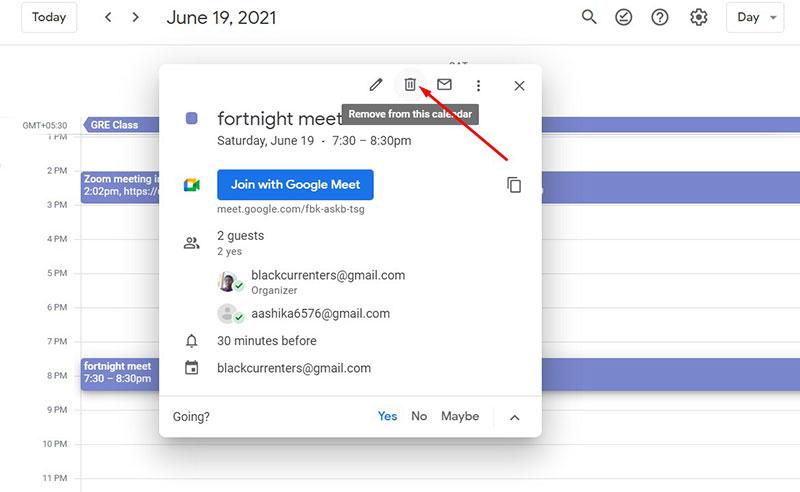
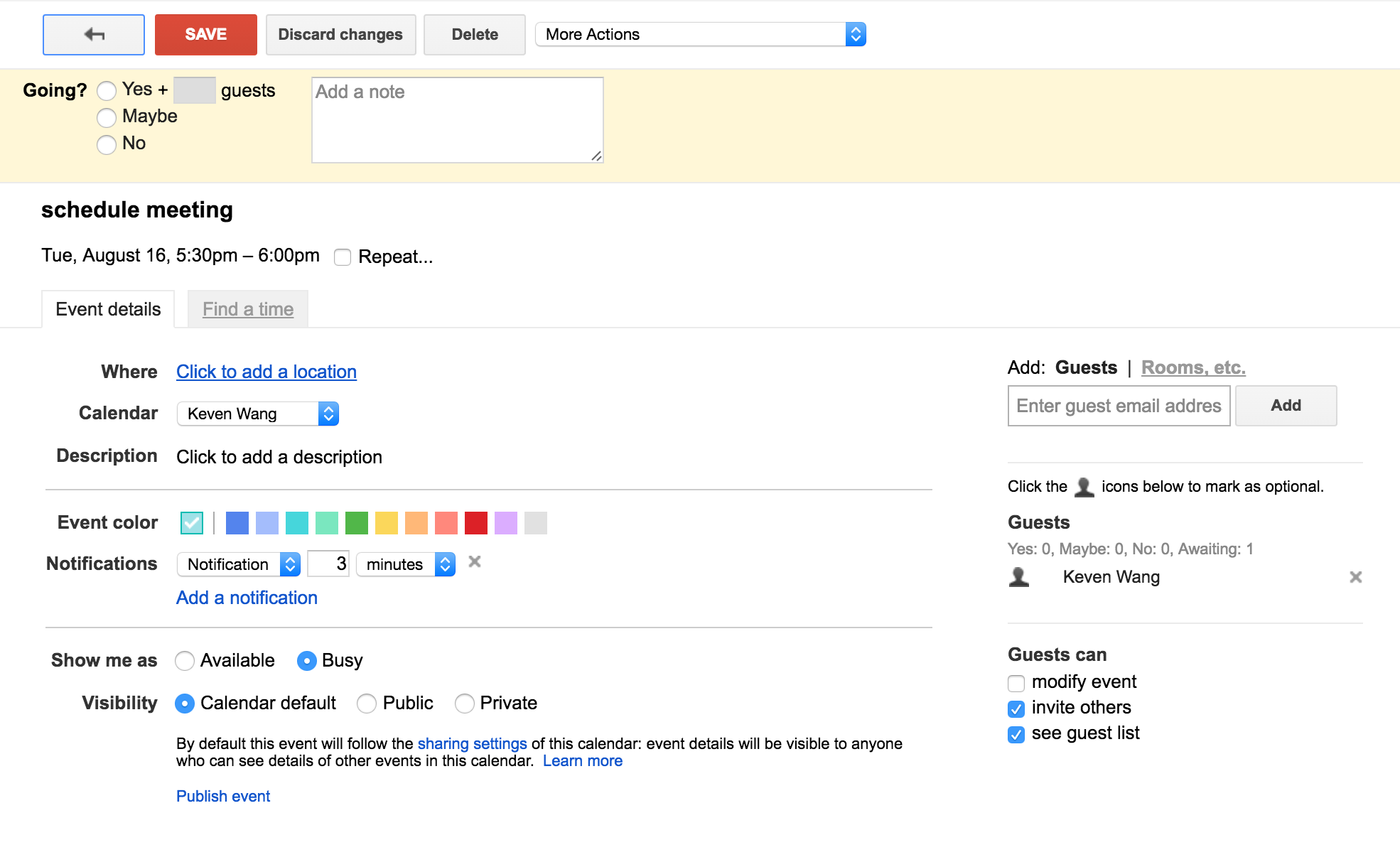
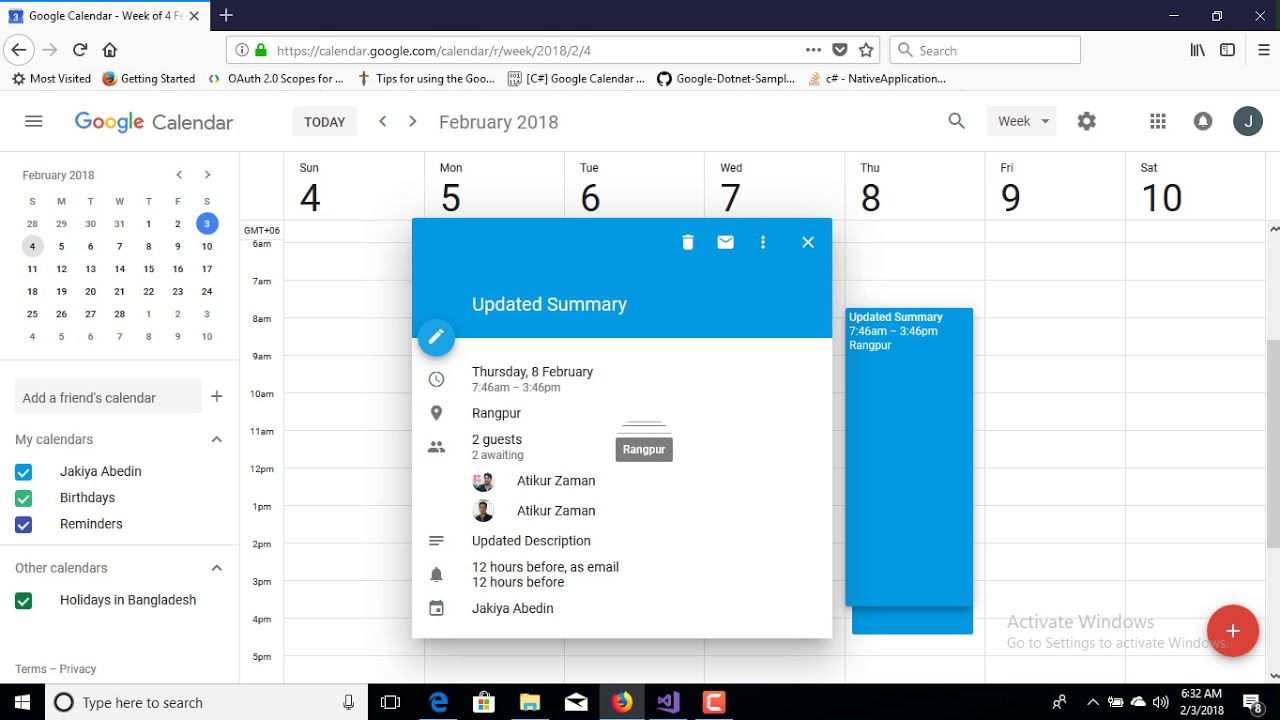






Robert’s Part: Google Calendar Integration

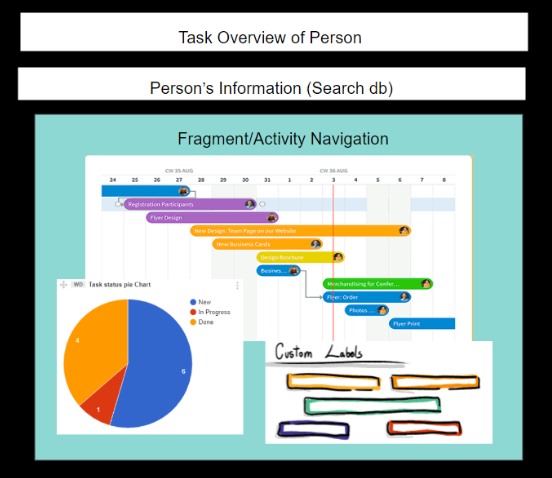
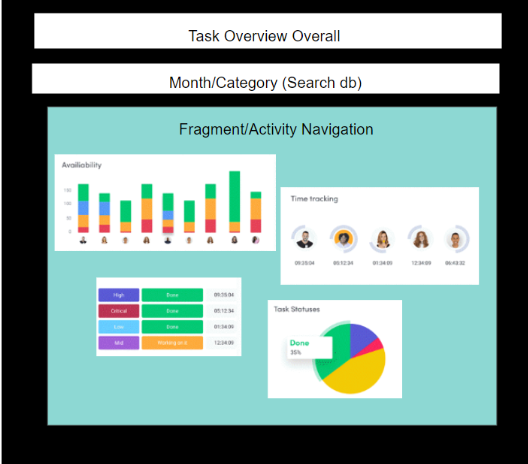
The idea behind integrating the Google Calendar API will be to simply organize, by date, the list of current tasks that the user would like to input. Implementing the Google Calendar API allows the user to have an easier experience when working with it as Google has put a lot of effort into making this API simply understandable. Each calendar created will have an associated metadata such as a calendar description or default calendar time zone. The calendar will also allow the user to create a single event or multiple recurring events that will contain information about the event(s) including title, start and end times, and attendees. The user will also have access to push notifications that can notify them of the event as it gets closer.



Ernest’s Part: Set goals with bars other stuff and linking tasks

A major facet of our application is the ability to have a visual representation of how the task/project that the user wants to be done, would be accomplished, over a given time. This idea will be based on a commonly used concept for project management called a gantt chart. The gantt chart is a set of horizontal bars that reflect the amount of time the user wants to dedicate to a certain project/task, which assists in both visualizing the workload, as all the bars are listed vertically next to the associated task, and provides a unique functionality in the user being able to scroll through the list of tasks. Also, if a task acts as a prerequisite for another one, the app will link, from one bar to another, both task bars to represent that. If time allows, working to add more ways of visualizing the data would be the next step after implementing the gantt chart. The chart data will be grabbed from the firebase database as a way to import the data. The way of implementing this is still being designed but the idea is to manipulate the line graphs that are used in the android plot library to allow straight horizontal lines to represent the different tasks. Another possible option is to use bar graphs instead, which would be more visually appealing to the eye.





Adam’s Part: Database system integration

Integration of non-relational database such as Firebase to store data created by the application in the cloud. Saving and retrieving this database will be crucial in displaying data if it’s required by user interface. We will upload various task allocation times and the category within that task into the firebase to allow the data manager/visualizer to be able to read in and display the information. Also, with using firebase, this will allow either one user or multiple users to view tasks from multiple devices.