Project Description: “Meet” is an application in which users upload their personal schedules and find different “meet up” times based on their availability via various features for socializing and communicating. For example, a colored indicator will represent current availability (either available or not available) and users will be able to manipulate availability spontaneously should an emergency arise or if they simply need “personal time.” Users can also indicate how they would prefer to spend certain times of the day for different activities such as “socializing”, “work”, “sleeping,” and “eating”.

Competitive Analysis: Facebook Messenger has a colored indicator that is either green when a person is active on the application, red when the person is on “do not disturb”, and void when the person simply isn’t active. “Meet” follows a similar concept, except that the indicator will be green when a user is available and yellow when the user has his or her schedule filled at the current time. Facebook Messenger further has the “Reminders” option that is useful in providing notifications and a visual for groups of people and specific friends regarding an upcoming event’s details. “Meet” eliminates the need for such an option given that everyone is constantly being updated on each other’s availability – events that are shared between users will appear on the home page. In addition, people can choose if to make certain events or activities on the schedules public or private to their friends.

My application also has a calendar feature that will display personal events. Shared events – ones that are setup intentionally between two users – can be identified as well in a similar fashion to Apple Calendar. Apple Calendar also has an ability for sharing between groups of Apple users such as family. It is separate from the Apple Reminder’s app in that each event will occur on a specific day and time. There is also an option for people sharing events to “show” changes via notifications for added, modified, or deleted events. However, members (i.e. Family) are only able to share specific times of shared events; not necessarily overall availability. The sharing option in Apple's Calendar is supported as a minor, simple feature of the app.

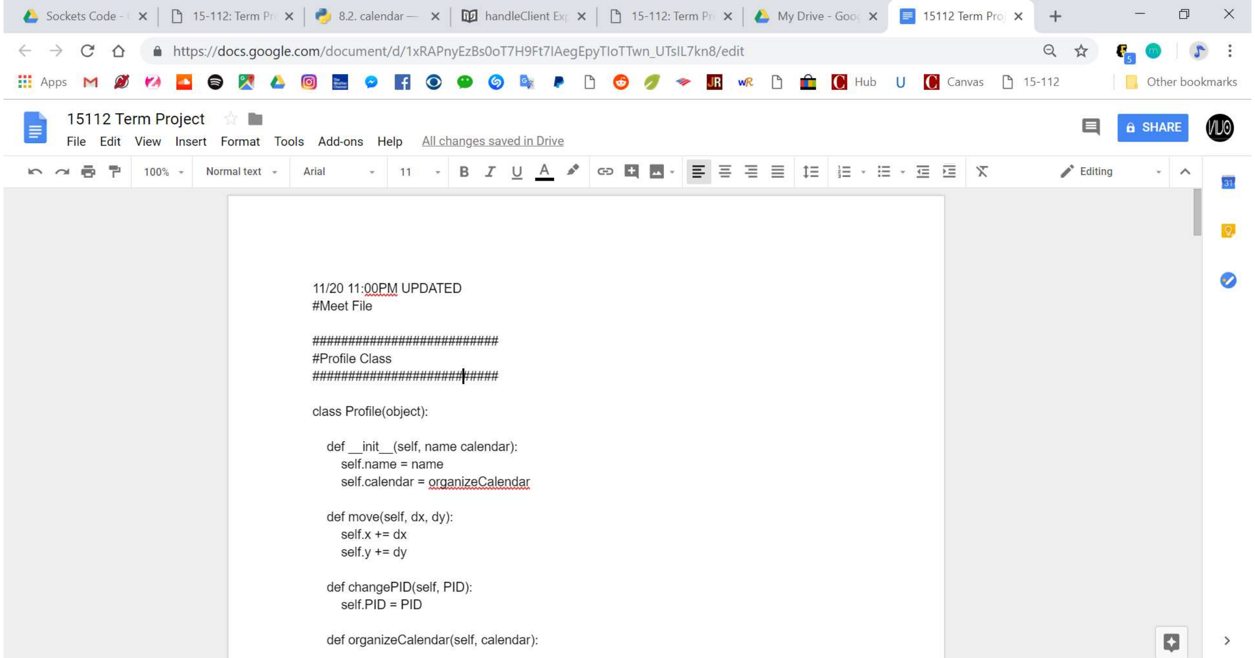
Structural Plan: Server file code will be cited from the 15112 Sockets Server Demo, created by Rohan Varma and adapted by Kyle Chin. The client file code framework will similarly be utilized; tkinter section will be modified for customization. Here, code for appearance of the user interface and navigation will be implemented. Importantly, the client file code references and begins with importing the "Meet" module before executing functions regarding tkinter. The "Meet" file will contain two primary classes. Each person logging into a server will be represented by an instance of the profile class that stores personal information such as name and an instance of calendar. The calendar class describes a person's personal schedule in which dates and times of events as well as "time usage preferred" settings are stored. These instances will need to be modified constantly for different inputs by friends and the user.

Algorithmic Plan: The application's ability to schedule and recommend times for people to meet up will be the most complex part of this project. To begin, the application must recognize the listed events on users' schedules and avoid collision between these events. The algorithm must also consider if the user has activated the "do not disturb" option or added an temporary event recently. The more users being considered for the given event, the more constraints there will be to handle naturally. Finally, the algorithm must assign priority to different times and possibly the event type (that may be marked for "sleeping" vs. "working" times) to recommend the best times for users to meet up.

Timeline Plan: By TP1, complete the profile class in the "Meet" file format a general Calendar class. In the client file, create a general homepage with an "Options" scroll-down menu and three separate modes or pages – a "calendar" showing all scheduled and shared events, a

“Meetup” page showing scheduled shared activities, and a “Today” page displayed by using tkinter. Also, basic Socket module functionality is shown with availability and Do Not Disturb icons in the top right corner. By TP2, users will be able to interact with each other in which data sent from one can be stored in another for scheduling. Users can also begin to enter their own data in and view it in some representation of a calendar on the calendar page as well as view updated and more complete visuals in the “Today” and “Friends” pages. All additional features (i.e. OpenCV) as well as a complete scheduling algorithm will be implemented into the application by TP3. In addition, all (variable) finishing touches on visuals and any animation will be finished to enhance overall appearance.

Version Control Plan: The code is backed up with Google Drive. Every hour or so of work, the code will be copy and pasted into the drive document and labeled with the date and time.



```
11/20 11:00PM UPDATED
#Meet File

##########################
#Profile Class
##########################

class Profile(object):

    def __init__(self, name calendar):
        self.name = name
        self.calendar = organizeCalendar

    def move(self, dx, dy):
        self.x += dx
        self.y += dy

    def changePID(self, PID):
        self.PID = PID

    def organizeCalendar(self, calendar):
```

Module List: Modules utilized include:

- Sockets

- OpenCV
- Python Calendar module*

TP2 Update:

Due to unforeseen issues with sockets, features such as messaging and the options tab (top left corner) will be held off. The do not disturb button is viewed as a “Y” or “N” for “Yes” and “No” regarding availability. The calendar, given trouble formatting, is currently a representation of the current month. Also, the idea of priority allocation to either being “free” time that has preferences (such as for sleep) or an event that has a “tag” representing its priority. For example, a concert is tagged as “social” and homework is tagged a “school”. The final update is that the priorities are as follows in order of descending priority; Sleep (highest priority), school, work, eat, social, and none.

TP3 Update:

Following up directly from TP2 and keeping those ideas in mind, in TP3 an animated “Added!” and “Cleared!” tab appears when events are added to user schedule. Finally, the home page will also display “*” next to all events added while the user was online – These events were recently added but are not necessarily today’s events.