Nina Cheng
OPIM 244
Spring 2019
Project Planning and Requirements

I. Problem Statement

A. On campus, students are constantly overwhelmed by emails regarding student activities and events. Their inboxes are flooded by Canvas notifications, club emails, and newsletters. This results in a twofold impact. First, the issue increases the levels of student stress as a result of information overload, which does not allow students to get information that may actually help with their decision making. Second, students actually miss out on interesting and relevant events, which harms their college experience. The opportunity here is to help students stay informed about events on campus and help them stay organized in order to make a stress-free decision about what events they would like to attend when they have free time.

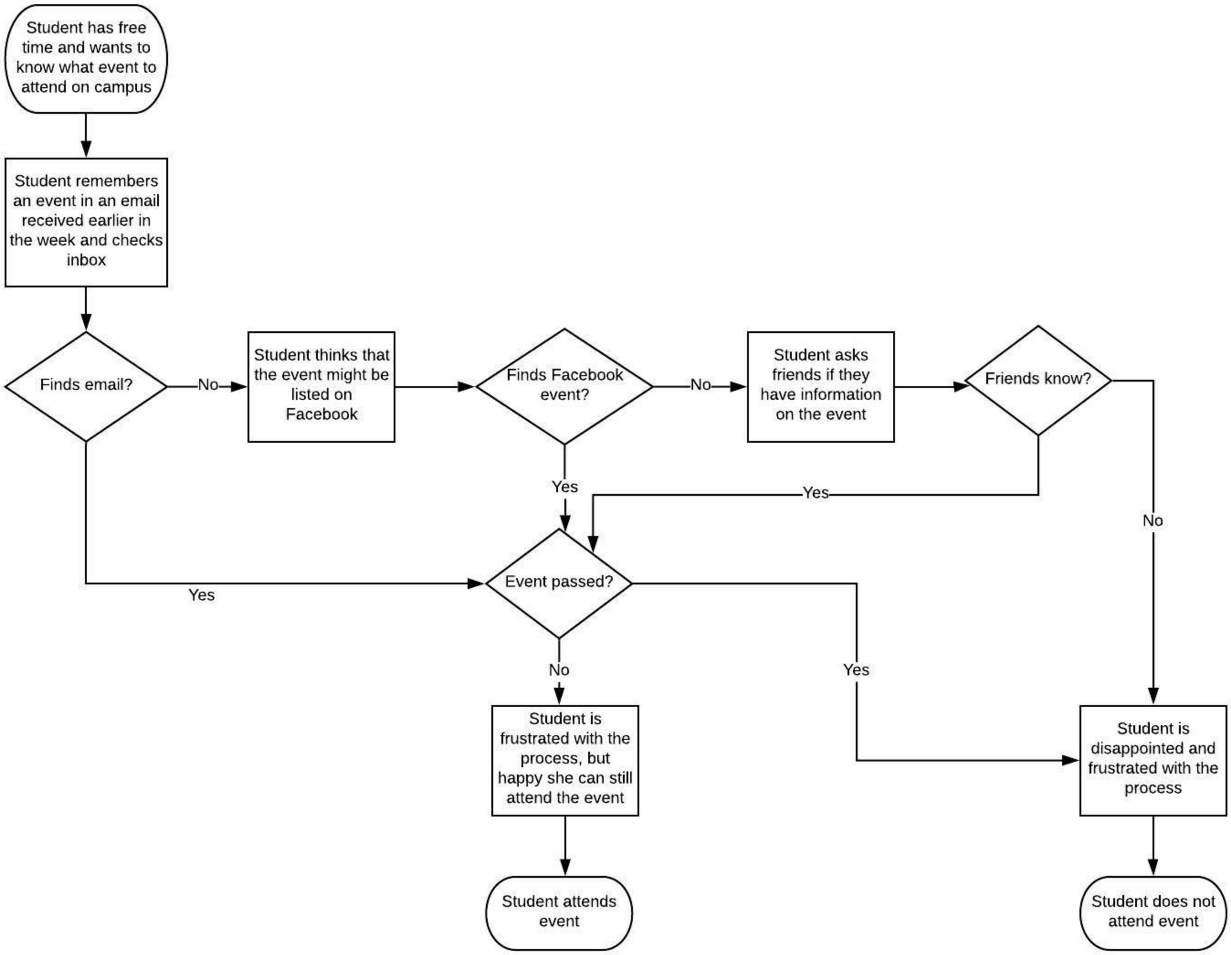
II. User Needs

A. College students are affected by this problem. Students need a better way to receive and analyze information regarding events on campus so that they can make informed decisions. Right now, they are getting too much information, which can hinder their decision making because they feel overwhelmed by the amount of information thrown at them and will also miss out on events they were unaware of. Their experience could be improved if there was an application that could list upcoming events at the university when the user has the free time to glean this information. For example, when the student realizes he or she has some free time over the next few days, he or she could prompt the application to see what events are happening over the next few days, rather than sift through countless emails.

III. Current State Processes

A. Students currently receive information about events on campus through a variety of media. First, students primarily receive emails from student clubs, Georgetown organizations, and Canvas notifications. This floods their inboxes, overwhelming students and causing them to miss important events they otherwise would have attended if they had known about them. Moreover, many of these students receive unwanted emails from these organizations. Furthermore, all these clubs and organizations have Facebook pages which create public events and invite students to them. While this is a great reminder for students, these event notifications are also mixed in with other Facebook functions, including getting tagged in photos. Students need one application to streamline all these notifications and make their lives easier.

B. Current State Process Diagram:


Student has free time and wants to know what event to attend on campus
Student remembers an event in an email received earlier in the week and checks inbox
Finds email?
No
Student thinks that the event might be listed on Facebook
Finds Facebook event?
No
Student asks friends if they have information on the event
Friends know?
Yes
Yes
No
Yes
Event passed?
No
Yes
Student is frustrated with the process, but happy she can still attend the event
Student is disappointed and frustrated with the process
Student attends event
Student does not attend event

IV. Potential Solutions

A. Set up an email notification system where students can subscribe to specific types of events happening at Georgetown University. For example, the Cawley Career Center allows students to subscribe to email newsletters involving specific industries. This solution filters the information that students receive in the first place, making it more likely that they are receiving information that is relevant for them and filtering the information so that it is easier to find in the future. However, it exacerbates the information overload issue, as students are already receiving an excessive quantity of emails.

B. Create an interface in which a student will input activities they are interested in attending. Essentially, the student notes the time and date of the event. Many Georgetown students use Google Calendar to maintain their schedules, but they don’t want to overwhelm their calendars with activity ideas. With this app, students can organize the activities listed in their emails, and when they are free, they can ask the app to tell them what activities are available that day. The benefit of this solution is that all activities in the system are already pre-approved by the student who put in the data in the first place. However, this creates extra work for the student to update and maintain the system.

C. The student can download a Google Calendar version of Georgetown University's events, allowing the program to contain a directory of all official events on campus. Write a Python app that lists upcoming events from this calendar and allows the student to add any interesting event to his or her own personal Google Calendar. This is an easier process for students to track and organize events they are actually interested in.

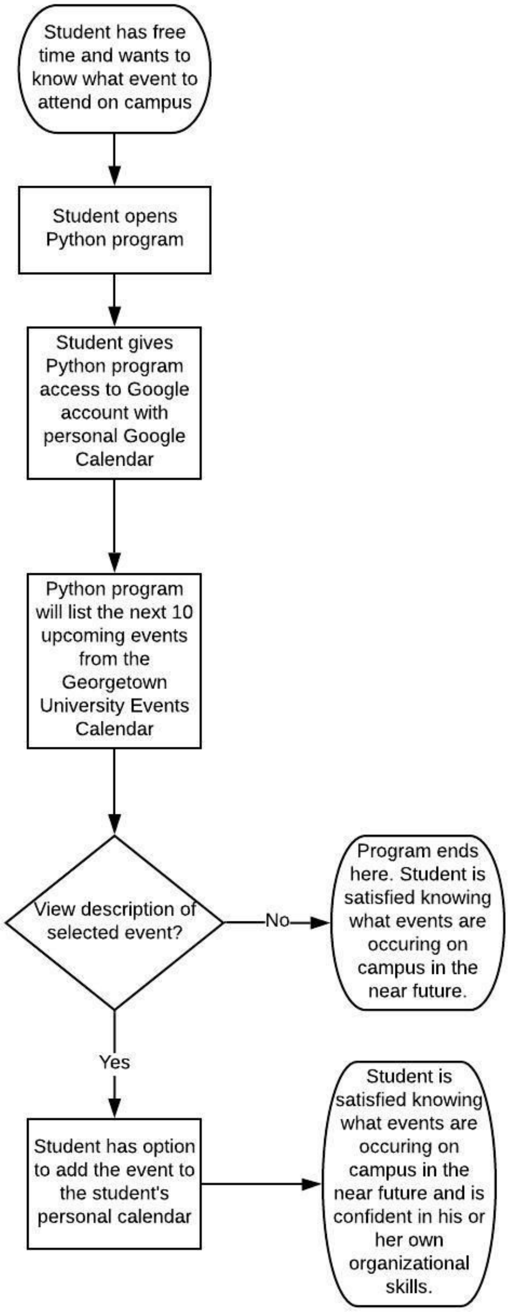
V. Proposed Solution

A. My proposed solution will be similar to solution C listed in the section above. This is the most straightforward method. Many students at Georgetown University already use Google Calendar, and a student interested in using this Python application would already be organized and proactive about attending events. Therefore, this solution best suits the target user's needs in a simple manner. I will write a Python program that pulls information from the Georgetown University Events Google Calendar for Georgetown University via Google Calendar. Overall, this solution simplifies the process in a number of ways. First, the student is confident that the system will not miss any official Georgetown events since the calendar will have all the events preloaded. This program also saves students time, as they no longer have to jot down notes about events they are interested in attending, overpopulate their existing calendars, or waste time searching their notifications to find event details.

VI. Future State Processes

A. When the user runs the relevant Python script, the program will first ensure that the student has given the program access to the student's personal Gmail account that contains the student's Google Calendar. Next, the program will list the next ten upcoming events from the Georgetown University Events calendar. Then, the user has the option to look at the descriptions of the event(s) that the user is interested in. Finally, the user can input the information to add to the user's personal calendar.

B. Future State Process Diagram:


Student has free time and wants to know what event to attend on campus
Student opens Python program
Student gives Python program access to Google account with personal Google Calendar
Python program will list the next 10 upcoming events from the Georgetown University Events Calendar
View description of selected event?
No
Program ends here. Student is satisfied knowing what events are occuring on campus in the near future.
Yes
Student has option to add the event to the student's personal calendar
Student is satisfied knowing what events are occuring on campus in the near future and is confident in his or her own organizational skills.

VII. System Objectives

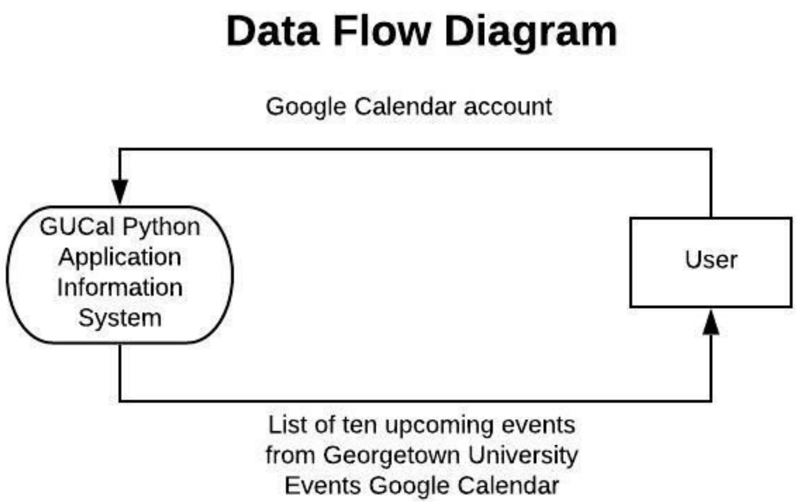
A. The system should allow the user to view the next ten upcoming events on the Georgetown University Events Google Calendar. Unlike the Georgetown Calendar interface, which can get messy when there are multiple events happening at the same time, this app will show the events in an easy-to-read list format with the date, time, and title of the event. The system will then ask the user if the user would like to see the descriptions of any of the next ten events. The system will also allow the user to input this information and create an event in the user's personal calendar based on this information. This will streamline the decision making process and prevent the user from being overwhelmed by emails, allowing the user to investigate the specific event he or she would like to attend when he or she has free time.

VIII. Information Requirements

A. In terms of inputs, the system requires the user to give the Python program access to the user's Google account in order to access the user's Google Calendar.

B. In terms of outputs, the system will print the next ten upcoming events from the Georgetown University Events Calendar for the user within the command line interface.

C. Data Flow Diagram:


Data Flow Diagram
Google Calendar account
GUCal Python Application Information System
User
List of ten upcoming events from Georgetown University Events Google Calendar

IX. Functionality Requirements

A. Desired user experience

1. The user will need to download the repository containing the necessary Python script and folders (https://github.com/nc585/google-calendar)

2. The user will open Anaconda command line prompt and navigate to the appropriate directory containing the project
3. The user will run the program which will output the next ten upcoming events from the Georgetown University Events calendar, including the date, time, and name of the event
4. The user can optionally choose to view any of the event descriptions
5. The user can choose to add any of the events to his or her own Google Calendar

B. Underlying system responsibilities

1. The user must run the program from the Anaconda command line interface
2. The program will ensure that the user has given the program access to the user's Google account
3. The program will present the next ten upcoming events from that calendar
4. The program will give the user the option to view any descriptions of the events and add any of the events to their personal Google calendar

X. Interface Requirements

A. The user will interact with the program via an Anaconda command-line application

XI. Technology Requirements

A. APIs

1. Google Calendar API will allow the developer to make a Python command-line application that makes requests to Google Calendar. This API is helpful because it will pull the data from a source that already has all the relevant information. Google has provided some information regarding this here: https://developers.google.com/calendar/quickstart/python
    - a) You will need to enable Google Calendar API on your personal Google calendar account
    - b) You will need to create a new environment and use pip to install the Google Client Library by typing the following in the command-line:
        - (1) pip install --upgrade google-api-python-client google-auth-httplib2 google-auth-oauthlib

B. Third-party Python packages

1. You will not need any third-party Python packages as the system can function with the Google Calendar API.

C. Integration

1. The software will not be deployed to a remote server and will not be integrated with hardware.
2. This program will run on your local computer.