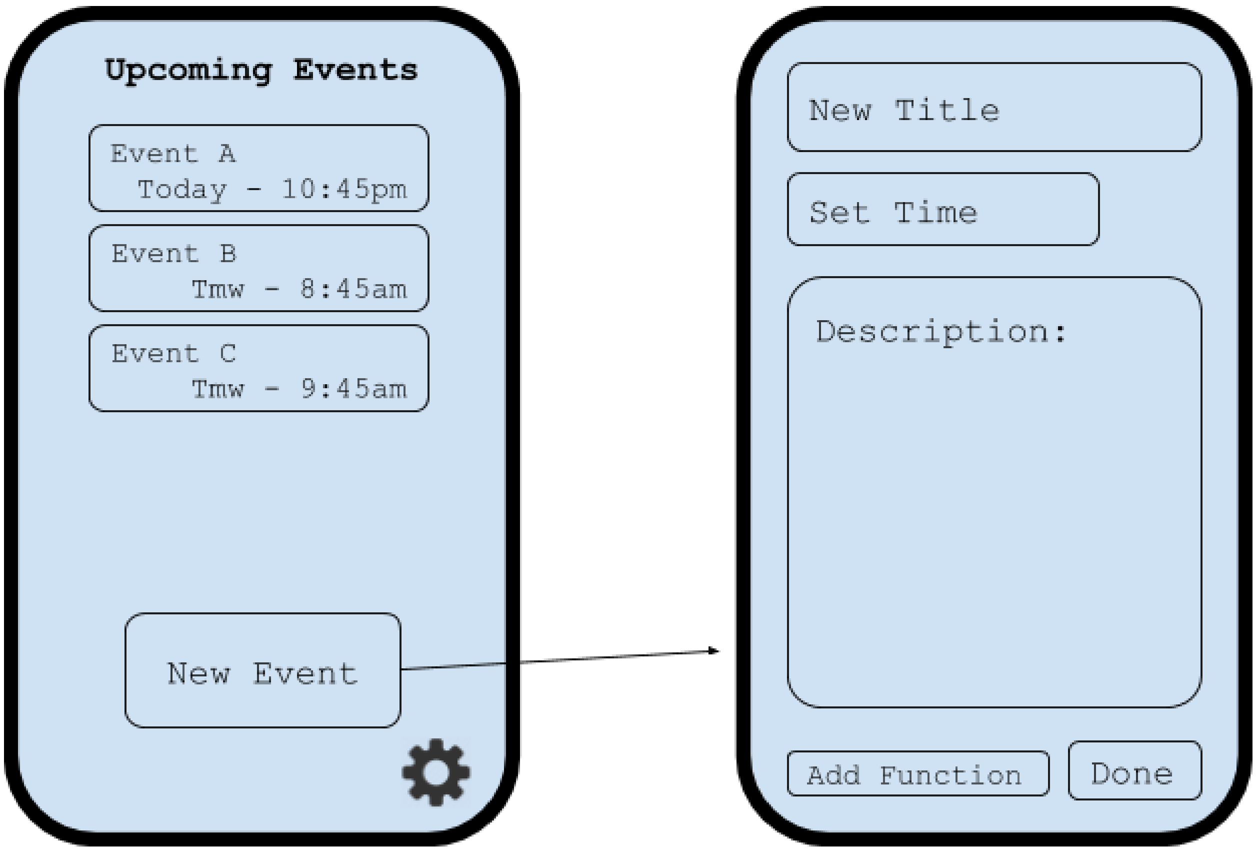
xTeam 158
Nick Jacobs, Jasmine Prianta, Yueyu Wang, Alex Grabowski

Always on Time

In the complex world that we live in, people are always busy with one thing or another, and problems can easily get forgotten in place of more pressing issues. In order to help people organize their lives better and ensure that everything that needs to get done is given time to do so, people need an organized time structure of when to do things.

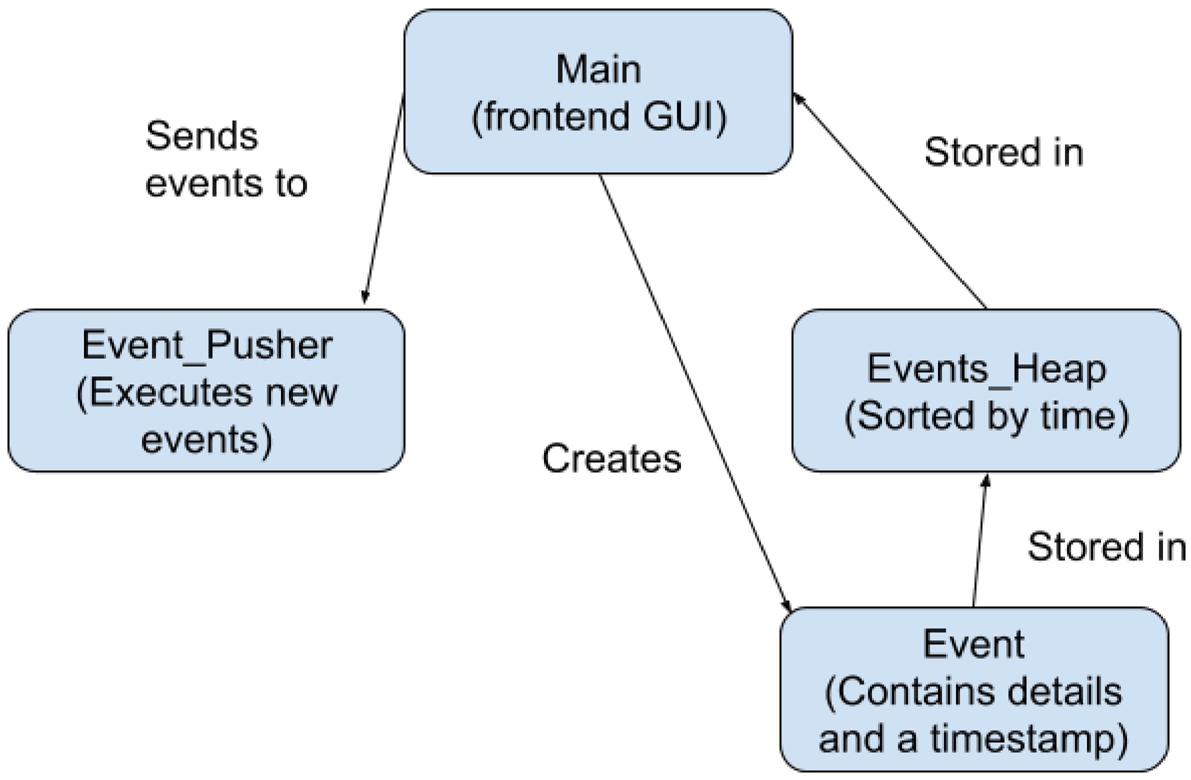
This program would allow the user to set various tasks to automatically happen at a time that they select. Whether it be a reminder to do laundry when they get home or a script set to go off at a certain time that cleans junk from their computer, the user would be able to create an organized time scheme of when to get things done. This would be used by anybody, but in particular people with a lot of different tasks to do in their lives, such as students.

The frontend GUI of the program would primarily consist of the main screen, from which the user may select actions such as scheduling a new event, changing in-app settings, as well as a visual diagram showing a shortlist of upcoming events. From the main menu, the user would be able to enter a secondary menu, where they could create a new event. In this menu, there would be options for setting the time and date of the event, a title for the event, as well as space, to add anything else needed to complete the event (descriptive text, images, code to run, etc).



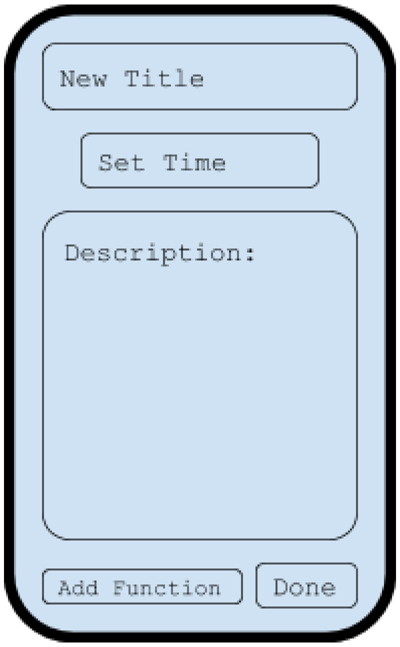
On the backend, there would be one main class that runs the main screen of the program, inside of which there would be an instance of a heap containing the events, sorted by the time until they execute. The main program would constantly check only the event on top, removing it and sending a message to the user. The heap would always handle keeping events in order.

The classes would consist of the main class that runs the app, creating new events, storing the Event_Heap, and pushing events to the Event_Pusher when their time arrives. Event_Heap stores the events of type Event, and Event_Pusher activates a function described in an input Event object



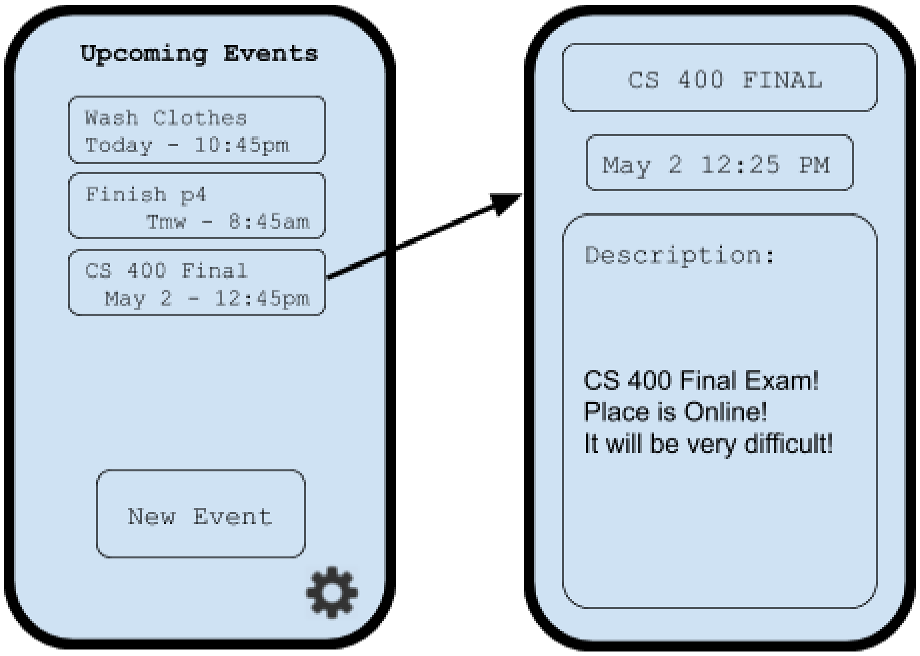
Input and Output of our Program:

The input diagram is as follows:



According to the program, you need to provide a title, time and description of your activity to create an event. The data type of the Title and Description should be String, for Time, it will provide a choice plate for you to choose the Date and Time, After choose, the time for the event should like: “Sep 30 3:00 PM”, Then, we can use Heap to sort the time to find the event’s correct place.

The output diagram is shown below:



Click each event's frame, it will jump to another diagram to show the detail of the event.

Also, when the time arrives, the Events detail diagram will automatically appear to notify you that you need to do this event.

Milestones:

1. According to our proposal, the Event class can be the most independent one, which can be finished first and tested in advance.
2. Then, the heap class which stores each event can be created and tested if it actually sorts these different events by time.
3. After that, the Event_pusher class will be created and be tested whether it shows the correct planned event at the correct time.
4. Finally, the main Class (frontend GUI) needed to be created to link other classes and build diagrams for input and output. Then test if the main class can access each function and execute each function correctly.

Assign tasks:

Alex: Will create the Event class, add each necessary data and test the class.

Yueyu: Will create the Event_Heap class and store each event in the heap, then test if it correctly stored and correctly sorted by time.

Nicholas: Will create the Event_pusher class to execute each event at the correct time, test it.

Jasmine: Will create the main class and the diagrams for input and output, link every other class to the main class, ensure the main class can be run correctly.

Together: Check the correctness of the program and find other unknown bugs, fix them