



Schedule
Buzzers
We have you scheduled!!

Orbital - Milestone III

README File

Submitted by-
Schedule Buzzers
(Shreya Pande & Wamika Malik)

Table of Contents

Our App	3
Download our app now!	3
Motivation	3
Aim	4
User Stories	4
Scope of Project	4
Program Flow	6
Possible AddOns	7
Our Android App	7
Developer's Guide to our App (Working of our App)	7
Bugs Found While Testing	10
User Testing and Reviews	16
Future Plans for the App	16
Project Log	17

Proposed Level of Achievement: Apollo 11

Our App

Download our app now!

Our app can be downloaded on any android device using the following apk file provided in this link: <https://drive.google.com/file/d/1Suua-wjGGT2zubovyjtt8M9vdDq-Eik/view?usp=sharing>



Schedule Buzzers
We have you scheduled!!

- Key in your schedule.
- Receive reminders for your classes everyday with the list of possible buses you can board.
- Key in your assignment deadlines and be assured to submit it on time.



Motivation

When you can't remember where your class was at, especially during the first few days of the semester, having an app that reminds you when and where to go can be a huge saviour. Moreover, hunting for food

between classes can be very painful if you don't have time and don't know the right place to go to. So we could remind you of the food stalls you can visit nearby to satisfy your hunger.

Besides, we plan to let you know what are the buses that go to the destination and to make life easier, we also plan to integrate an assignment deadline reminder which ensures that you don't miss out on important stuff.

Aim

We hope to help people schedule their lives at NUS. We hope to provide an app that is capable of performing multiple functions such as providing reminders for classes, giving suggestions on the buses to board, the most important - keeping a track of all the assignment deadlines and last but not the least, satisfy your hunger craving in the midst of a busy day by providing you with a list of the restaurants in the area.

User Stories

1. As a student who has taken up too many modules, I would like to be able to key in my schedule for the semester and receive reminders for the same, so that I can avoid missing some of my classes.
2. As a student with early morning classes, I would like to be able to know about the possible direct bus routes to get to my class, so that I can avoid wasting time by taking the wrong bus or a longer route to get to class.
3. As a student with different modules, I would like to be reminded of all the assignments for the week and their due dates, so that I can keep track of pending assignments and set my priorities accordingly.
4. As a student with a tight schedule and love for food, I would like to be able to quickly find out about the food options around me, so that I can satisfy my hunger.

Scope of Project

Schedule Buzzers is an application that allows students to key in their schedules at the beginning of the semester and receive reminders of the classes that they need to attend, along with the possible busses that make a stop at the destination. The application also provides a list of all the food outlets in and around a certain area (where your class is currently at) to grab a quick snack.

Students also have the option to key in their assignments and the due dates for the same and receive reminders for that as well.

- Features **completed** by the mid of June:
 1. Database in firebase that stores all user information

2. Use react-native to create the home page and sign in/sign up page and use firebase for user authentication
 3. Create a main screen with navigation options
 4. Work on the add class screen, linking the information input by the user to the database in firebase
 5. Work on the add assignment screen, linking the information input by the user to the database in firebase
 6. Add a profile page that displays user information, possibly with a profile picture
 7. Link the account settings page to firebase to allow user to modify personal information
 8. Make a database of all NUS shuttle bus routes
 9. Create a database of all food places in NUS
 10. Create terms and condition page and a general information page
 11. Create a schedule timetable
 12. Create an assignment view page
- Features **completed** by the mid of July:
 1. Work on code to provide notifications to the user reminding the user about classes and/or assignments by adding classes and assignments as events to the device's calendar.
 2. Implement a code that shows possible busses that can be taken given the starting location and destination of the user.
 3. Implement a code that shows food places around a specific location
 4. Work on making the app more user friendly
 5. Work on features specific to Android OS.

Program Flow



The Database stores the following data:

1. Account details of each user
2. Schedule and assignment details of each user
3. List of food places at different locations in NUS
4. List of busses with routes within NUS

Possible AddOns

To further enhance the features of our app, we can implement a GPS Tracking System within the app in order to be used in the “Find Food” and “Which bus goes there?” part of our project. We also plan to include a map in the “Which bus goes there?” part of the app to make it more interactive for the user. We will use this tracker to track the current location of the user and display this location as the default location around which the user wants to find food options or buses. The user can also change this location depending on his/her needs. We also plan on allowing the user to upload an image for profile picture in webp or gif format. Currently, the app and its features are available only for android users. We plan on expanding it to iOS users as well to expand our user base.

Our Android App

Our app aims to allow the user to enjoy basic features of various apps like keying in schedule and assignments, receiving the notifications/alerts about the same, searching for food outlets and buses from one place to another on one single platform.

Developer’s Guide to our App (Working of our App)

Our app starts off by prompting the user to sign up or sign in using google. This has been implemented using the firebase authentication service. This authenticated account is linked to the cloud firestore which stores all information pertaining to a particular user. The profile page and account details are stored and updated via the cloud firestore.

When the user uses the add class function, the details are checked to ensure there is no duplicacy of classes. If timings clash with another class, either completely or partially, the user is notified accordingly. The algorithm also checks if class timings are proper, for example if the user fills in the class start timing as 1200 hours and the end timing as 1000 hrs which is not possible, the user is notified about the same by an alert. If the user wants to remove a class from the schedule, the user is required to only enter the necessary details i.e day of the class, module name and the class type of identification in the database. The algorithm for checking is triggered when the remove button is pressed by the user. If the class exists in the database, then it is successfully removed, else, the user is prompted about the absence of any such class in the database.

The add assignment page works in a similar manner to the add class page. The user is supposed to fill in all the fields but the assignment name field and the deadline are compulsory. It is using the assignment name that the assignment is identified in the Cloud Firestore. In case the user does not fill in either or both of the compulsory fields, an alert is sent to the user to notify the user about the same. The algorithm of the add assignment page also checks for duplicates in the assignments. If the user fills in an assignment name that already exists for the same or a different module, the user is notified about the same. There is an algorithm to check for improper deadlines as well i.e deadlines where date has already passed. If the deadline is such that the date has already passed, the user is notified about it and is asked to key in a proper deadline. For removal of an assignment, the user has to just key in the assignment name. The remove button for the add assignment page works in the same way as the one in the add class page. The add assignment page offers an extra feature of update using which the user can search for an already existing assignment in the database and also change the details of the assignment in the database.

The current schedule and current assignment pages are used to display the timetables and all the assignments of the user respectively. These pages are linked to the Cloud Firestore and hence any changes in the cloud firestore by the user due to addition, removal or updates using any of the add pages are immediately reflected in the timetables and the assignments.

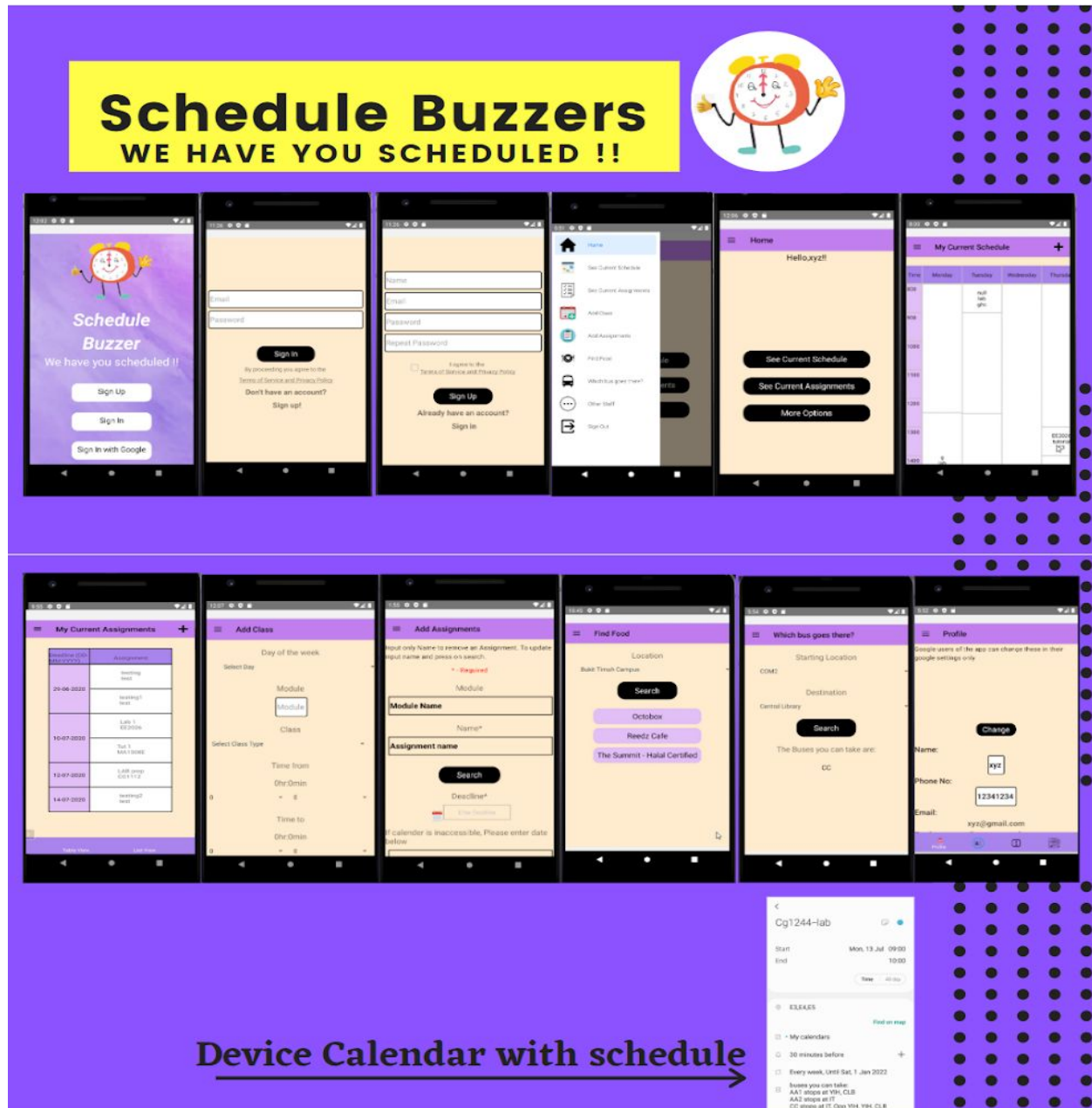
The find food page in our application is directly linked to a database that stores information about the food outlets (in the form of documents in Cloud Firestore) in NUS segregated on the basis of the location. When the user chooses a location on the app using a picker, the food outlets in and around that location are displayed. Clicking on each food outlet displays certain information corresponding to that place. This is because when the user clicks on the food outlet, the document in the database is accessed by the app and the details stored in it are displayed.

The “Which bus goes there?” page in our application is linked to a database that stores the information of all the bus routes in NUS. The data in Cloud firestore is stored in the form of stops. All the stops are collections in the database all of which stores buses going through each stop in the form of a document. Each bus document shows the stops that the bus visits after the main stop(collection: stop) in the form of collections. When the user chooses the start location and the destination, then the collection corresponding to the start location is accessed and then all the documents of the bus are accessed to check if the collection corresponding to the destination exists. For those buses that the collection corresponding to the destination exists, the bus is displayed. If user keys in start and end location such that no direct bus is available, then the user is notified of the same.

The other features that our application offers is the profile, accounts, about and terms and condition. The profile and accounts pages allow the non-Google users to update their personal details on the app. The changes that the user makes on these pages are also immediately reflected in the database. For google users, they are advised to make account changes through google itself.

Our application is also linked to the built in device calendar. So whenever the user adds, removes or updates classes or assignments, the real time changes of the same are also displayed on the device

calendar. The device calendar is also used to send out alerts/notifications to the user 30 minutes before each class and 2 days before the deadline of the assignment. The alert/notification sent for the class displays the upcoming class and the location along with the buses and the stop to alight at to reach the class. In case of assignments, the alert/notification displayed shows the assignment name, its deadline and the additional notes that the user may have keyed in.



Bugs Found While Testing

- 1) *The Add Class Page took in timings that were not possible(i.e., the class ends at a time before it starts) - **completely fixed***

A condition was added to ensure that the current timings are entered. This led to another bug.

- a) *The “wrong timing” alert would be given even when the class end time had not been entered(since it was 0 by default). - **completely fixed***

The conditions were added when storing the information in the database. Thus, the information is stored only if the timings are valid, else error alert is given.

- 2) *The calendar in the Add Assignment page is inaccessible on web - **completely fixed***

An instruction has been added to manually input the date in case the calendar is inaccessible.

- 3) *If a user views the assignment/schedule timetable after making updates, the updates are not shown - **completely fixed***

Real-time updates have been enabled by adding a function to do so.

- 4) *The timetable would not display correct slots for classes in case the classes started/ended some minutes past the hour - **completely fixed***

This was an error with the logic used. Changes were made to the algorithm to allow variable timings to be entered and displayed correctly in the timetable. This led to another bug.

- a) *If the user updates the schedule by adding a class from 8 am to 9 am, the class is not displayed - **completely fixed***

This bug was recently found and it was realised that the timings are stored as strings instead of numbers, this resulted in firebase orderBy() giving the wrong order and thus data was not properly displayed. This bug has been rectified by changing the code to store timings as numbers rather than strings.

- 5) *If a time slot is already filled in the schedule or an assignment has already been entered by the user, no warning messages are provided if the user tries to re-enter the information. This information causes the previous information entered to be bugged as well - **completely fixed***

This bug will be fixed by adding a condition before updating the database. An algorithm is to be written which checks for overlapping timings before adding the class to the schedule.

- 6) *After a certain number of reads, the Firebase Quota exceeds and user is unable to access his/her data - **not yet fixed***

To fix this bug, the firebase plan has to be updated to a paid plan. This will be done when the app is ready to be available to users. Another alternative is to shift the data to another database which can be freely accessed.

- 7) *The profile page initially only displayed the user email correctly on the page and did not display rest of the user detail - **completely fixed***

This bug was because the user email and password and the rest of the details were stored in different databases. This was resolved by bringing all the details and the user email with password together in one database.

- 8) *The published app crashes when the Add/Update button on Add Assignment page is triggered - **completely fixed***

This bug is due to the deadline being input as a date in firebase. Timestamp/new Date() function doesn't give a format compatible with all devices. Therefore, it caused the app to crash. This bug was fixed by inputting and evaluating the dates as strings.

- 9) *The app would crash if the add button in Add Class page or the search button in Add Assignment page was pressed without entering the required fields. - **completely fixed***

This bug was fixed by adding appropriate conditions to display error messages when the buttons are pressed with empty required fields.

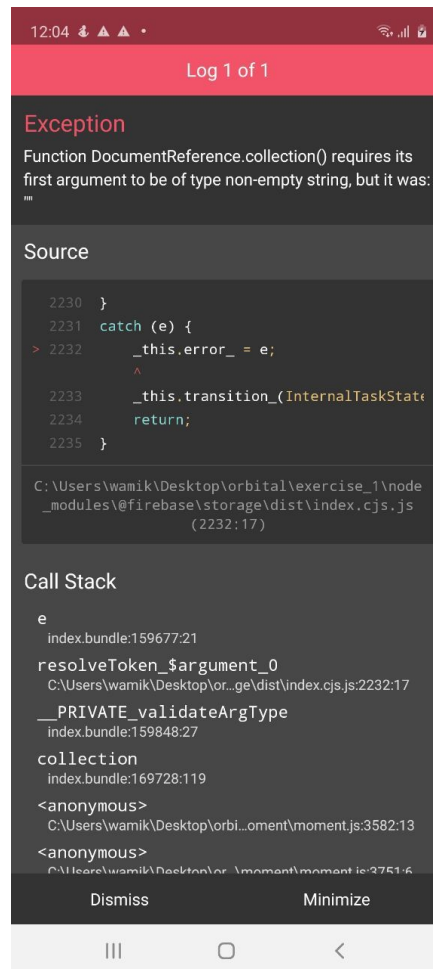
- 10) *When adding events to the calendar, device considers input time as GMT - **completely fixed***

First, the device timezone is determined and the appropriate time is subtracted from the input time before sending it to the device's calendar.

- 11) *Event's end cannot be added to the calendar, so it defaults the end to 1 hour after the start. - **not yet fixed***

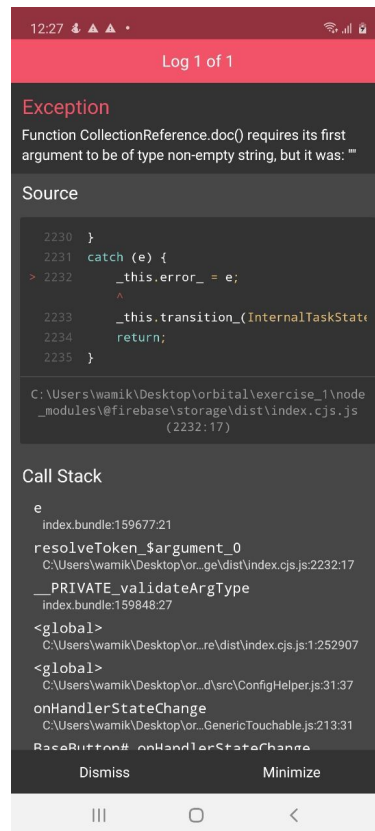
The app crashes when an enddate is added to be saved to the calendar. A work around has yet to be figured out.

- 12) *If the user enters a class name and type that is same as another class on the same day, he/she will get the following error: - **completely fixed***



A condition will be added to check if the class name and type exists already for the same day.

13) On the add assignment page, if only module name is entered and the search button is pressed:



*This error occurs - **completely fixed***

Another part was added to the existing condition to ensure that searching proceeds only if the assignment name field is not empty.

14) (improving the user interface) Assignment is added as long as assignment name and deadline exists, other fields are not necessary but it is hard for the user to detect this unless he/she tried it out - **completely fixed**

An indication can be added for necessary fields.

15) The account details page would crash if the user presses the update buttons without entering their current password. - **completely fixed**

This has been fixed by adding a condition that alerts the user to enter the current password before proceeding.

16) The account details page would update the email and password even if the new email/new password fields are empty - **completely fixed**

A condition has been added to ensure that these fields are filled before the update occurs.

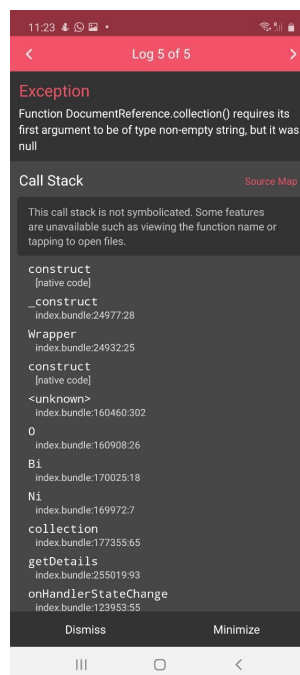
17) *The profile page doesn't update details if the photo field is null - **completely fixed***

Added a condition while updating details in the database to ensure that the photourl is an empty string if no photo is added.

18) *The class or assignment when removed is only removed from the database - not from the devices calendar - **completely fixed***

This can be fixed in a way similar to the update assignment code, wherein the event's id is used to remove the event.

19) *If the search button on find food page is pressed without choosing a location, the following error is seen: - **completely fixed***



This bug is fixed by adding a condition to alert the user if the search button is pressed without entering a location.

20) *Remove button bugs: would give an error if pressed without entering the required fields - **completely fixed***

Conditions were added to prevent this from happening. Instead, a user is given an alert to fill in the required fields.

- 21) *When viewing assignments, incorrect date would be shown if there is only one date- **completely fixed***

This bug was due to the sorting algorithm used, which did not have a condition for only one date. This condition was added to remove the bug.

- 22) *When many classes exist for a day, the clashing timings code is executed even before the existing timings array is filled - **completely fixed***

This bug was fixed by using the .then() function to ensure that chunks of code are executed sequentially.

- 23) *The Google Sign in feature worked well in debug mode but when the app was released, google sign in would cause the app to crash - **completely fixed***

This was because the debug SHA1 fingerprint had been entered in the firebase authentication settings but the same had not been done for release. Doing so corrected this bug.

- 24) *For Google Sign in navigates the user to a let's go page. This button if pressed after cancelling the google sign in would crash the app - **completely fixed***

This bug occurred because the let's go button would try to get user details without checking if the user exists. A condition was added to redirect the user to the sign in page and alert them to sign in.

- 25) *The UI of add class page was a bit confusing for some users - **partially fixed***

The headings were made bold to differentiate them from the details to be input. This will be further enhanced by adding borders around the input fields and making the design a bit more linear.

- 26) *Many times, when input given by the user is erroneous on the add class page, the details input are reset and the user needs to enter them again. - **not yet fixed***

To enhance user experience, we plan on retaining the information input by the user, so that the user can avoid typing details repeatedly.

- 27) *The assignment added in the device calendar spans 2 days with no difference between the last date and the day before the deadline. This can be a bit misleading for users - **not yet fixed***

The calendar event's name will be changed to reflect "1 day left" and "deadline today" for every assignment.

Points to be noted:

- 1) For removing an assignment, only the assignment name field is considered. Even if other field details are entered and are different from the actual details, they are ignored and the assignment is still removed.
- 2) If the user searches for an assignment name but while updating changes the name, this gets added as a new assignment and no changes are made to the old one. The user needs to remove it himself.
- 3) Assignments are added to the device calendar over a span of 2 days before the deadline.
- 4) The user is required to enter the timing of the classes in 24 hours format. If the user enters a class with the timings between 2100hrs and 0800 hrs, they are not displayed in the schedule. We did this mechanism taking into consideration the fact that students have classes beginning the earliest from 0800 hrs and ending the latest at 2100hrs.
- 5) The application does not allow the user to have multiple accounts in the app from the same device. This mechanism was put up considering the fact that users will always have only one account from 1 device.
- 6) The app currently does not allow the user to upload images in the format of gif and webp. We plan on including it as well in future.
- 7) The profile page details for google users can be changed but it doesn't reflect in their google account. It is local to the app.

User Testing and Reviews

In order to test out our app, we performed testing of our app with the potential users i.e students of NUS.

Reviews that we received from them are as follows:

- 1) Upon asking them about their preference between the google calendar and device calendar, one user preferred Google Calendar over the device calendar. The reason for the same being that the user can then access all her timetable on various devices using Google Calendars. Another user preferred the device calendar, since it is an app for the user on the go and hence having it on the device calendar is preferred.
- 2) The users found the app handy and useful with all the features that it provided. They like the simplicity of the UI.
- 3) The users found the UI of the app interactive and user-friendly.
- 4) Users found the inclusion of buses that they can take to each class in the alert useful.
- 5) One of the users also suggested that we can add a map or picture of the locations in the bus page of the app. The user suggested this as it would make the app even more interactive.
- 6) One of the users suggested a further improvement for the UI of the add class page. The user suggested that we make the headings bold or make a line to differentiate between the headings and the text inputs and pickers making it easier for the user.

Further Enhancements made to the App

We would like to include more features in the app for our users to have an enriching experience with it. These features are also to be included based on the user reviews that we have and the ones we get for our app in the future. Some features that we can plan on implementing are :

- 1) Giving the users an option to choose between device calendar and google calendar to store the classes and the assignments. The users will also be given an option to opt out of storing information in the calendar and getting alerts.
- 2) Include a remove all option for schedule and assignments in the schedule view and assignment view pages. This will be combined with a feature that allows users to create different timetables. For example, users can input a schedule for semester 1 classes and have a separate schedule for semester 2 classes. This would allow them to delete the timetables as and when required.
- 3) Allow the users to have a “keep me signed in option” to avoid signing in everytime.
- 4) When an alert about a class or assignment being saved is given, it would allow the user to either continue adding or view current timetables. This would ease navigation.
- 5) Allow the user to remove any schedule/assignment from the view schedule or view assignment page. This will be implemented by giving the user an option when the particular class is pressed. We will also implement a remove module feature that will remove all instances of the given module from the schedule.
- 6) Including a GPS tracker to find the current location of the user and use that to find the food outlets around the user and the buses at that location.
- 7) Having a map of the buses to give the user an idea about the time for the bus to reach the desired location.
- 8) Remove assignments automatically once the deadline has passed.

Project Log

Here's the link to our project log:

<https://docs.google.com/document/d/1IWuRswg6PzApIMm0KrvXLzTwnNsgpp-AiQctKyDTLY/edit?usp=sharing>