CPEN 391 MODULE 2 FINAL REPORT

TEAM 13
Sirine Trigui - 46443157
Yixin Zhao - 49427140
Linda Munisi - 22535158

a. Introduction:

This project is designed to be an interaction tool for students and instructors to use on campus. Our goal is to provide instructors as well as students a handy tool for their teaching in class activities .This project then has two sides which will be described in our target market.

Below is our target market:

Our target market is intended to be generally the education industry, covering both students and instructors. For students our product provides them with a means to communicate in real time with the instructor through responding to questions posted by the instructor in real time. For the instructor, the product provides them with a means to see the students results and grade them, store them on a database for future regrade and comparison, and also the ability to store the questions to be displayed in class in the database.

b. Development Process:

At the beginning of our design, we followed the following process in development:

1. Identification of the Need / Market / Value

In the identification of market, we considered who would be the main clients for the product upon completion. We identified that the device would be of need in schools for both instructors and students. For students, it would be a good market as students would always use a device that can direct them to class as well as they can be able to use it in class to poll real time results. For instructors, the device, particularly the application would be of great use especially in the storage of the information and managing the class attendance as well as student grades in an easy and organised manner in the database

2. Specification of Requirements

The requirements of the design include creating a way to save the information for the instructor side. For this requirement, we considered two methods of storing data that

include storing it in a local database, or storing it in an online server through a realtime database. On the students side, we had to specify requirements that include a way to accommodate different types of students responses. We considered both multiple choice and text entry as requirements for students. In addition to these specific requirements, we established a common requirement for students and instructors which is a way for them to navigate different routes from one location to another.

3. Comparing Solutions and choosing the best one

Having the designs in the prior section, we made comparisons in order to make the best decision for the best choice of solution to go with our design. For the implementation on the instructor's side, we made a comparison between the local database and the real time database for storing information. Since the students will be updating their responses for responses in real time, then we decided on using a real time database to store information and hence we used the Firebase Database to store the information in the google servers.

On the students side, upon considerations that we needed the design to include different types of responses from students, hence we included two ways of entering the inputs that include multiple choices and entering of text through a text field. This way, the instructors are flexible to display questions that need short responses or multiple choices instead of being confined to one, traditional way of entering alphabets only.

4. Implementation and Design Testing

The implementation of each section was done in an agile process while testing the design in the whole process step by step. Each functionality was tested in the process to avoid accumulating bugs in the code. The testing began in the hardware side. The bluetooth being the main means of communication was tested by first testing the bluetooth dongle to see if there is connection and if commands are being sent or received. This was tested using the eclipse console by printing the commands that are responsive and those that aren't. The gps was tested by putting it in different locations and trying it in different times to see the accuracy of the time recorded by the gps.

On the software android side, testing was done through printing statements at the logcat and using println statements. For the database, manual testing was done to make sure that the data is saved at appropriate entries. The "manual" testing was done by checking the contents of the database in real time when the user (being the student or the instructor) enters the data to see if it is stored correctly. Finally all the components were tested together when the app is running to see all functionalities.

5. Validation of the Design

Upon completion, the design was validated by testing it with different individuals. The app was given to a sample of students to download and test it as if they were to use it for class. The app was also given to a sample of individuals to use as instructors to test the functionalities that were implemented.
In general, the app and the whole design proved to be useful to both students and instructors for interaction purposes.

c. Work Accomplished:

- Sign-up functionality. Our device takes the time when student synch to the Bluetooth dongle and store it in the database. Instructor can use it to check attendance as well as whether the student is late.

- Map functionality. For both student and instructor, the map view is able to show them route, distance and time from their origin and destination locations. Database functionality. Student number, sign-up time, date and responses will be stored in database. These information can be viewed only by instructor.

- Answering selected question. Student can only answer the question which is shown on the screen and enabled polling by the instructor.

- Multi-type response. Student can use the buttons to answer multiple choice questions as well as using the input box to answer the short-answer question.

- Viewing responses. Instructor can view students' sign-up time, date, and responses. All these information is sorted by their student numbers.

- Writing and uploading questions. Instructor can write their own questions and store it on database. It can be send to DE1 board to display on the touchscreen later. (should we include this?)

d. Detailed Design:

The project consists of four major parts: hardware components, software components, database and Android Application.

Hardware components includes Touchscreen, Bluetooth dongle, GPS controller, and DE1-SoC FPGA board; the software components used are NIOS II 15.0 Software Build Tools for Eclipse, and Quartus II 15.0 Version. For student, the student number, sign-up time, date, responses are stored in Firebase database, while instructor can store questions to it. The Android App has two sides, one is for students, and the other

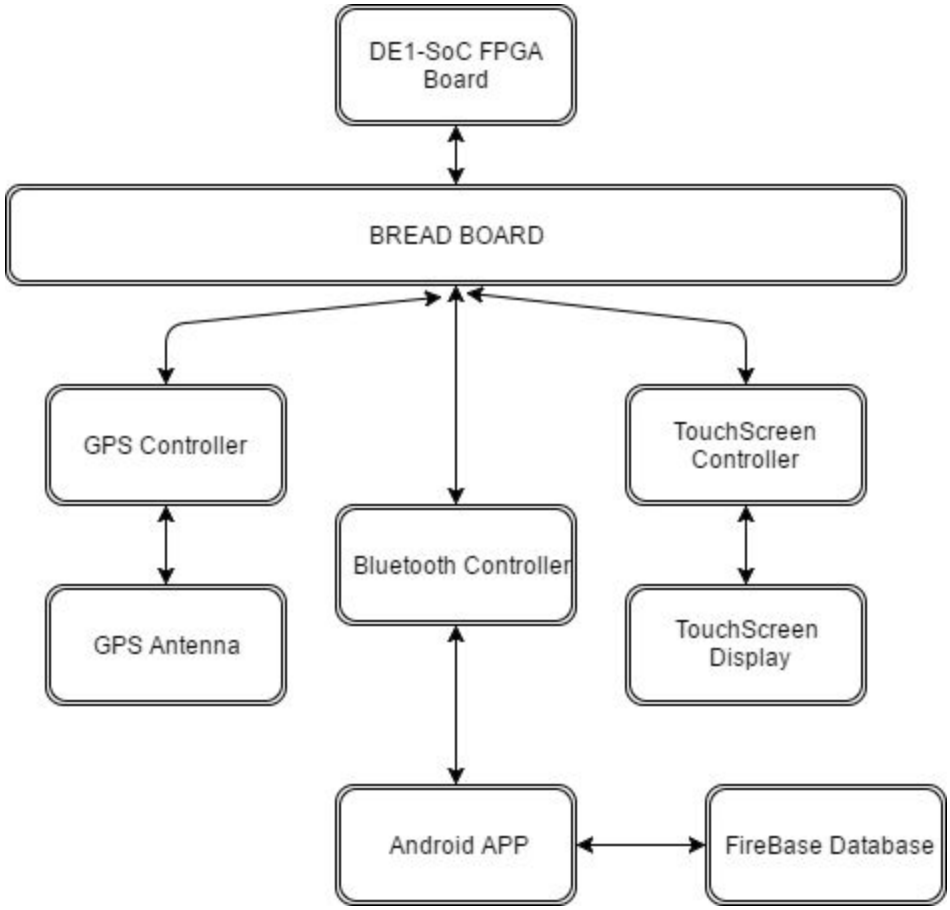
side is for instructors.

Our project requires communications among different components. The Eclipse runs the code that manages interactions among Bluetooth dongle, GPS chip, Touchscreen and Android APP. The APP communicates with the Eclipse as well as database.

When student signs up (connects to the Bluetooth dongle), APP sends a message, which is received by Bluetooth dongle, asking for the time. Eclipse reads the message and get the time from GPS chip, then Bluetooth dongle sends it back to the APP. The APP stores the student number, sign-up time, date and responses to the database. The information can be viewed by the instructor. And instructor can use the APP to write question and store it on the database for later use.

For the Touchscreen, when the “Start Polling” button is pressed, Eclipse uses Bluetooth dongle to notify the APP which question button should be enabled for students to answer the shown question. All the other question buttons should be disabled.

Below is a Block Diagram of our system:



Below is a detailed design

Hardware side:

These are modules that are written in Eclipse for NIOS II and are used to control the hardware used in the design.

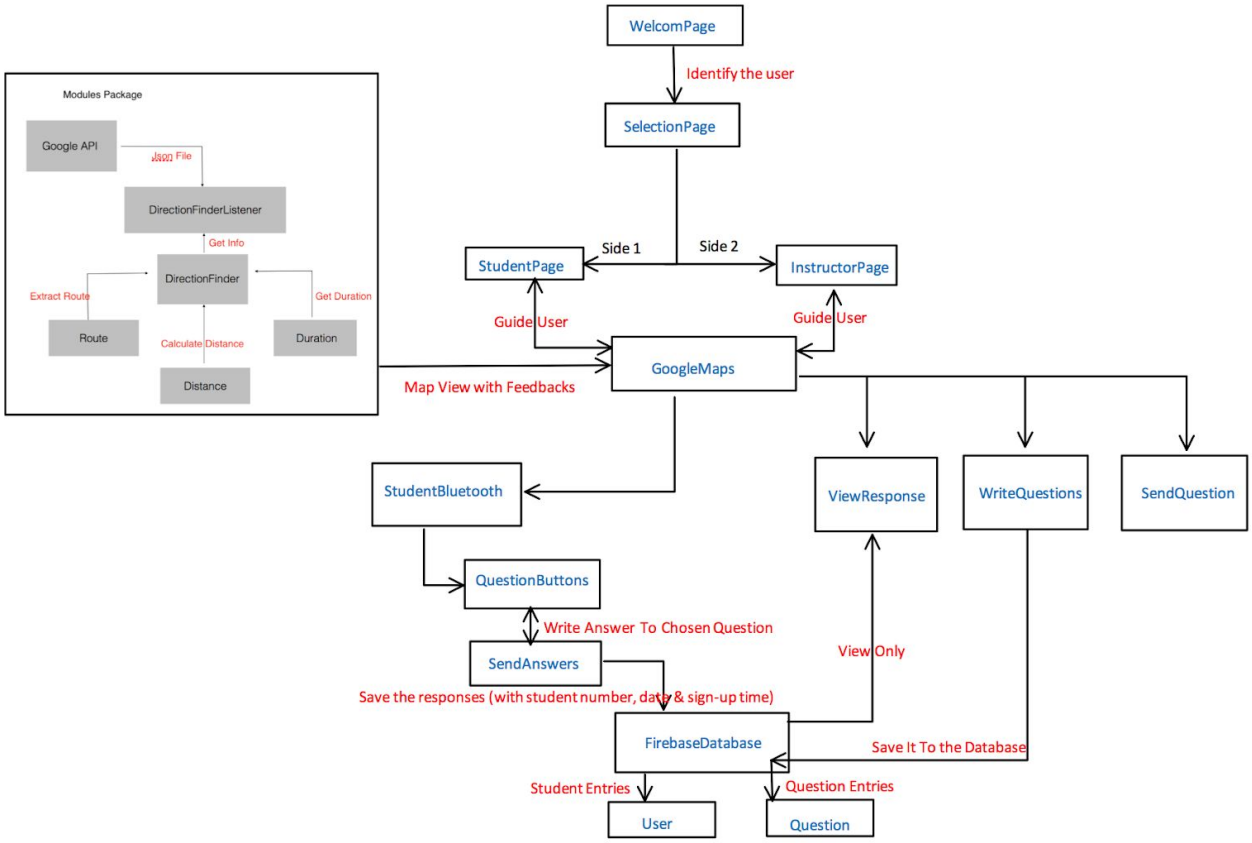
The main module which is in *hello_world.c* file and many different submodules stated below as follows:

- GPS : This module includes code that initializes the GPS, makes the settings for the GPS and extracts real time from the GPS.
- TouchScreen: This module is responsible for the touch screen logic. It includes the initialization of the touchscreen display, and the code to get the touched and released coordinates of the touchscreen.
- Bluetooth: This module is contains the code that initializes the bluetooth, enable command mode, changes the name of the bluetooth dongle, the password, and getting and sending characters.
- Graphics:
    1. Pixel: contains the logic behind reading/writing a pixel.
    2. Instantiate: contains all the definitions for the constants and parameters used in the functions.
    3. Fonts: includes the predefined arrays of fonts for different font sizes.
    4. Font_Types: includes the text fonts: capitalized and normal font.

Software side:

This is how each module is interacting with one another:

(It consists of Android side only, all the hardware part are ignored)



- main/java
  -Modules:
    1. DirectionFinder: This represents the main class for the map view.This is where all needed information is being extracted from the Json file.
    2. DirectionFinderListener: This class is responsible for generating the Json file right after providing the origin and destination location and clicking on the find path button.
    3. Distance: This class is responsible for determining the distance which will be provided to the user in kilometers or meters.
    4. Duration:This class is for providing the user by ow time time it will take them to be at their destination.
    5. Route: This class is responsible for tracing the route starting from the marker A to marker B.

  -com/example/eason/studentassistant:

    1. BluetoothAdapter: Provide a row view for the paired and the discovered bluetooths(with an icon on the left and the bluetooth Mac address).

2. ChoiceAdapter: Provide a row view of the given choices “A, B, C, D, E” which is used on the SendAnswer activity.
3. FirebaseDataBase: Manages the initialization of the database, and creation of the database reference and corresponding objects
4. GoogleMaps: Provides the map view and some other feedback as distance and duration (being extracted from a Json file)
5. InstructorBluetooth: Displaying paired and discovered bluetooths for the instructor upon trying to connect.
6. InstructorPage: Provide several choices for the instructor: whether to upload questions , to view responses or to send the question to be dislayed on the touch screen.
7. ListViewAdapter: This represents the row of the FiredataBase list(providing time of arrival and student answers)
8. MyCustomArrayAdapter: Provide a row view for the paired and the discovered bluetooths(with an icon on the left and the bluetooth Mac address).
9. Question: Manages the question entry in the database. Each question entry consists of 5 questions and instructor can enter five questions to display on the touchscreen
10. QuestionButtons: This page is for displaying several question button through which the user (student) will be able to select their choice.
11. SelectionPage: This page has the code that displays the user categories, whether student or instructor for the user to choose.
12. SendAnswer: Answer to be submitted and recorded on the database to be graded later.
13. StudentBluetooth: Page consists of initialization and display of paired bluetooth devices and discovered bluetooth devices on the students’ side of the app.
14. StudentPage: This page asks students for their student number and store them using Shared Preference. Clicking on “Connect” button will direct students to the StudentBluetooth activity.
15. User: Manages the user entry in the database. Each user entry consists of a student number, date, time they synced with bluetooth and corresponding response for each question.
16. WelcomePage: The page that manages the splash-screen with the app name and app logo.
17. WriteQuestion: Page that has an edit text for instructor to enter the questions to be stored in the database

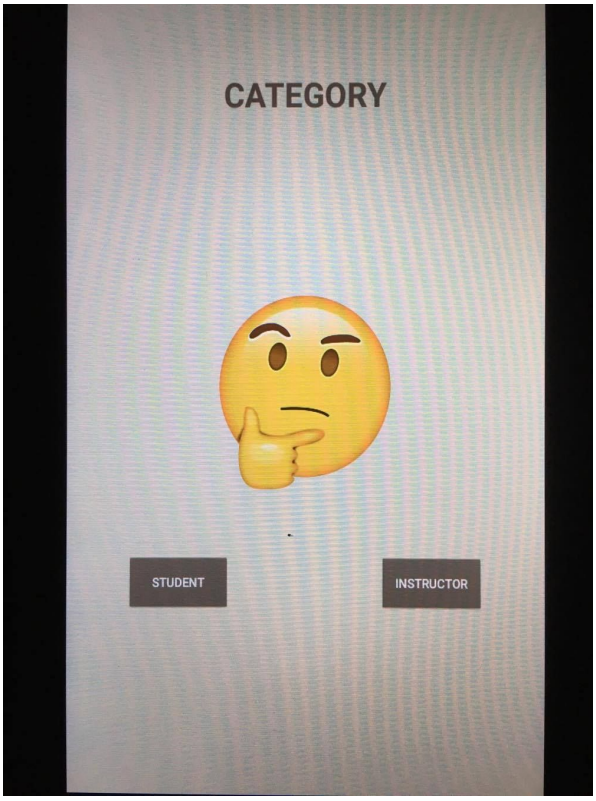
List of functionality and features (from the user's perspective)

- Bluetooth connectivity: This functionality was completed, and the communication via bluetooth was one to one for demonstration purposes. If the product was to be developed further, the bluetooth can be setup to act as a server so that it can support multiple users at the same time.
- On touch screen display and control: This function was implemented completely enabling the instructor to display questions on the screen and control the display of the answers
- Map view: This function was completely implemented enabling the users to get directions from one location to another.
- On database storage: Function was successfully completed enabling the storage of students information in the database and the instructor's questions in real time.
- Automatic grading: This functionality was not extensively implemented, the extent it was implemented is the instructor entering manually the grades.
- Unique access through providing student s ID number for their login in the app
- Detection of arrival time of students(included for presence and participation grades)
- Uploading questions prior to class: Store information on database then are being downloaded to be displayed on the touch screen for students to answer.

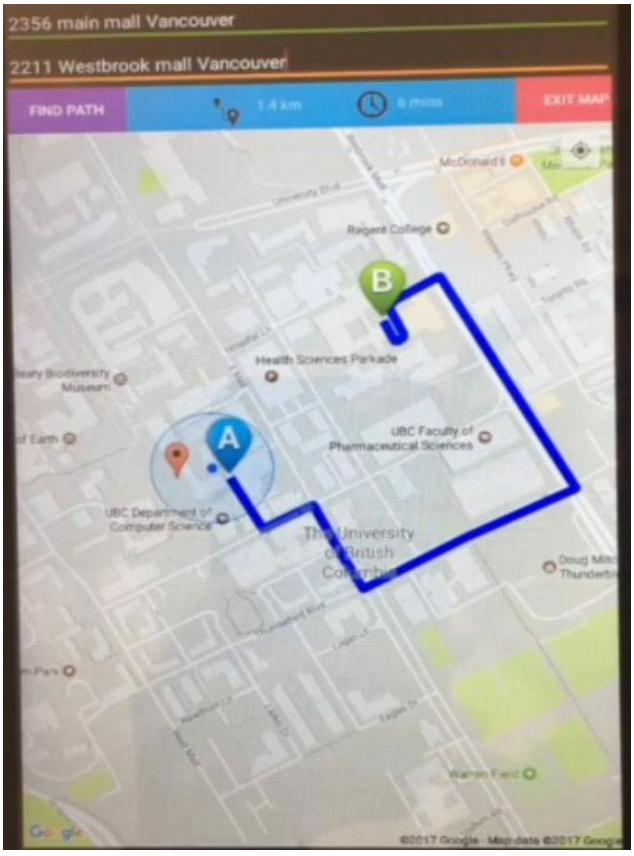
e. User Interface and description:

This is the first window that will show up the moment scholar plus is being launched.

This window represents which category the use belong to. If the user is a student, the student button should be clicked which leads to other student features. In the other hand, if the user is an instructor, then the instructor button should be selected so that the user can benefit from other subsequent features.

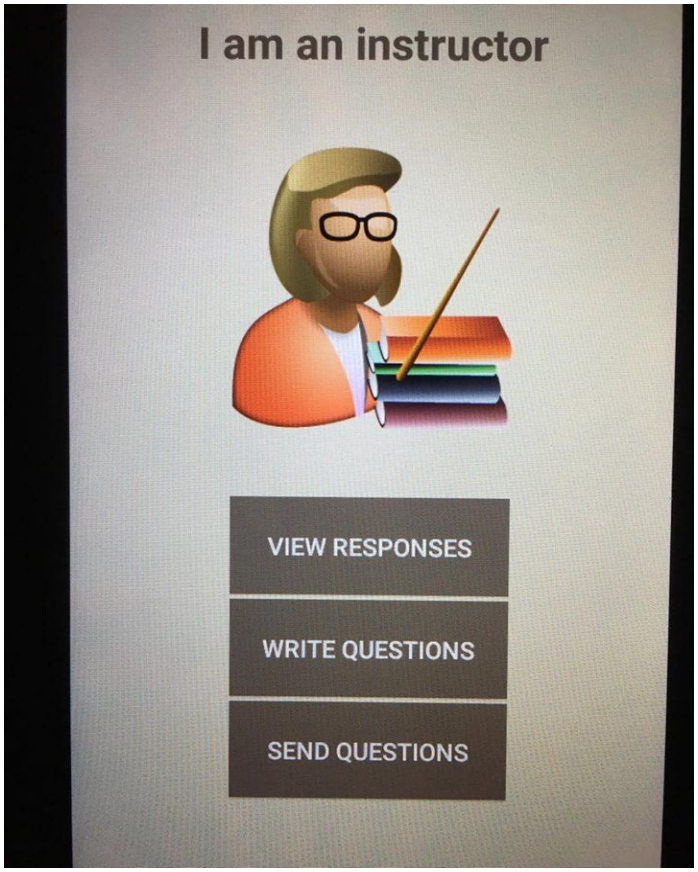


This is a common feature between the two categories. The map view can in some sort provide some feedback such as distance and duration to reach the destination assuming that the user has already provided their origin and destination locations.
Right after having entered such piece of information, the user can get their path by clicking on the find path button.
Marker A in the following screenshot represents the starting point whereas Marker B represents the end point.
The Red Marker is the default one representing the user's current location.

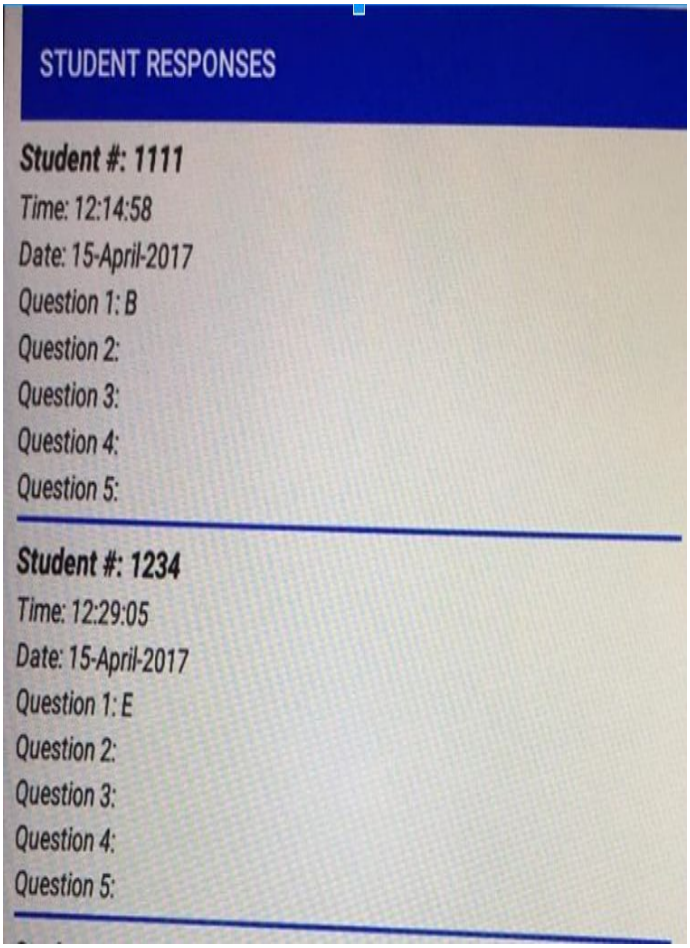


Assuming that the user in this case is an instructor, having selected the exit map button found on the map view, the user has three options:

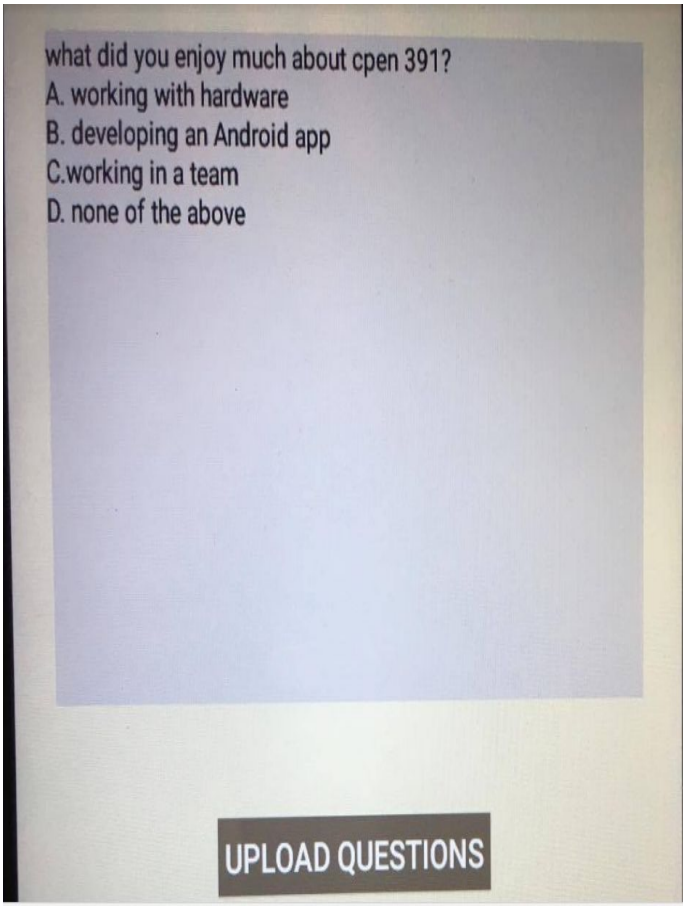
- View responses selected: Please refer to screenshot A
- Write questions selected: Please refer to screenshot B
- Send questions selected: Please refer to screenshot C



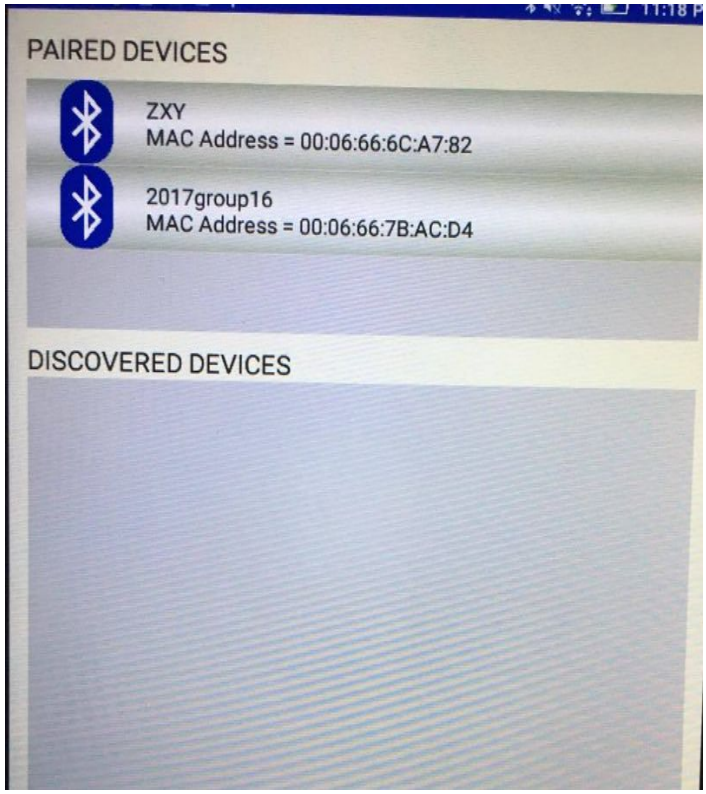
A. Having selected view responses button, student details will be displayed on this window being stored on the embedded Firebase database.
   Each row represent some specific information related to a specific student.
   Students are identified by their student numbers.
   Time shown below is the time reflecting when the student connected to the bluetooth class, to sort of help the instructor determine the student's arrival time.
   In this trial, the instructor prepared five questions to be answered in class. When the screenshot was taken, only the first two questions have been enabled.



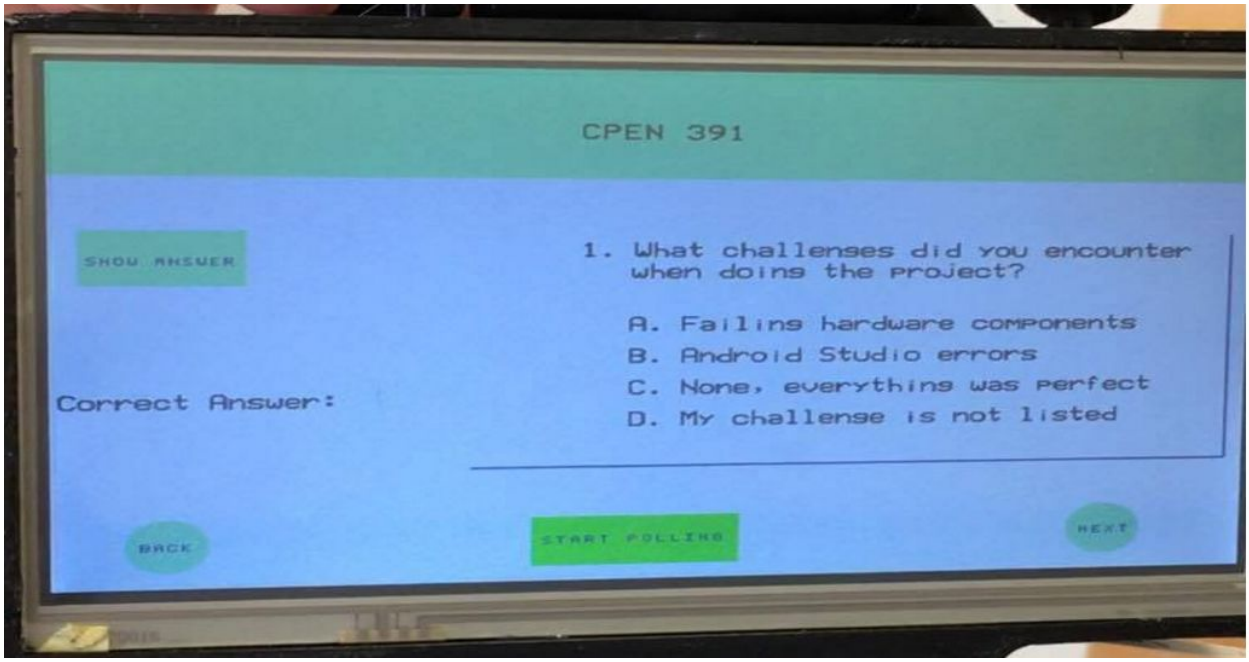
B. Having clicked on the write questions button, the instructor has the ability to write their questions prior to class and upload them to be stored on the Firedatabase by clicking on the upload questions button.



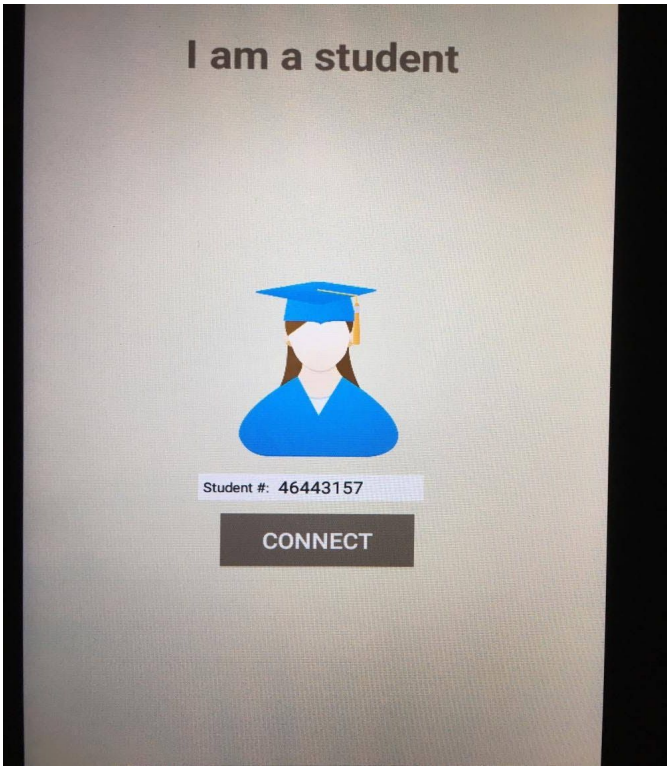
C. Having selected the last button on the instructor page, the user needs to connect through bluetooth by selecting the appropriate one to be able to send questions to be displayed on the touch screen.



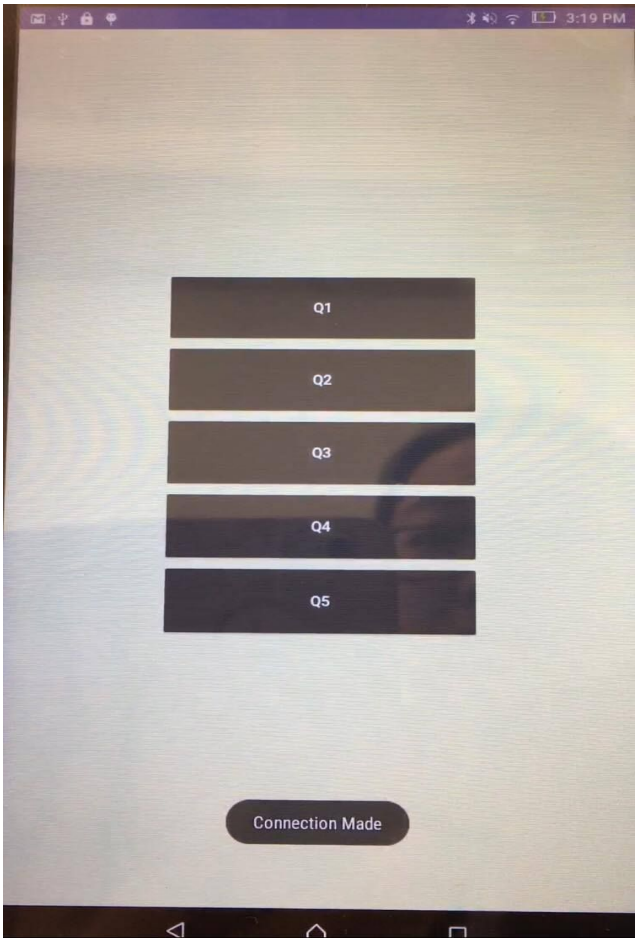
After downloading the question that has been store on the Fire database prior to class, the instructor has the ability to control the touch screen. Only when the instructor clicks on the start polling button, student then will be able to select their choice to answer the question.Clicking on the back and next buttons, previous or next questions will be displayed on the screen.
After disabling the poll, the correct answer can be shown by clicking on the show answer button.



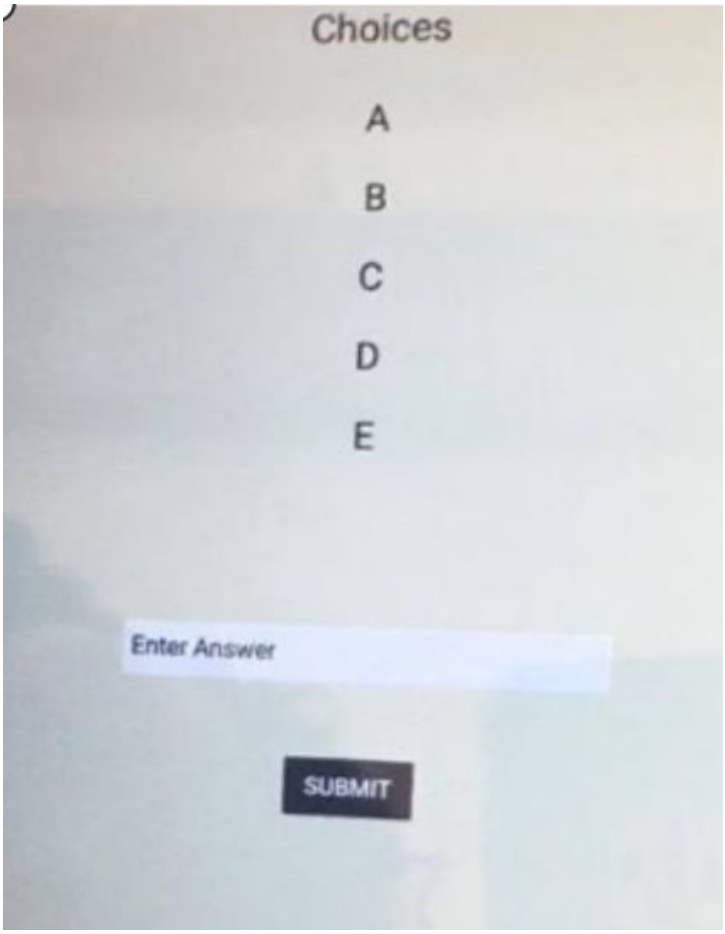
Please note that we switched to the student side: the second category.
Having selected the student button from the category page, and after locating the class (if the user is not familiar with it), the student is required to enter their student number to be able to login so that they will have access to the in class questions/activities.
Clicking on the connect button, the user will be directed to the paired and discovered bluetooths page to select which they want to connect to.



After connecting the the class bluetooth, the student will have access to answer the displayed question. Only one question is enable at a time, meaning that if the student arrived late and question one has been answered, the student will not be able to get marks for it as they did not provide their choice at the required time.



Having selected the enabled question, the student will have to make a choice and select a letter and they need to submit it to be stored on the database to be graded later.



f. Results (Solution Assessment):

The original project was designed to support two different sides:

- Instructor side:
    - Search route. The map view can be used to search route.
    - View results. All the students' responses, sign-up time, date which are categorized by their student numbers can be viewed by the instructor from the APP;
    - Write questions. Instructor is able to write question and store it on the database from the APP.
    - Upload questions. Instructor can upload questions, which are stored in database, from the APP through Bluetooth dongle to the DE1 device in order to display it on the Touchscreen.

- Student side:
    - Search route. The map view can be used to search route.

- Enter student number. Student number will be used to identify the entry of sign-up time and responses in the database.
- Connect to Bluetooth dongle. The APP displays all the paired Bluetooth devices as well as discover the unpaired ones. It stops the discover process if the specific device, which is the Bluetooth dongle of our system used in the classroom is on the paired list or discovered.
- Question availability. Students are only able to answer the question being displayed on the touchscreen and the instructor has clicked on the start polling button
- Send Answer. Students can choose different choices from provided list “ABCDE” or enter their own answer. After click “submit”, the response will be stored on the database. Students can update their answer when the question is still “available” to be answered.

g. Conclusions and suggested future work

If this project was to be developed further, there would be some features that

- More animated map view: using the GPS signal on Android device to implement real-time guidance as the one in Google Map.
- Automatic grading system: calculate student’s total grade and store it in the database.
- Multiple users: supporting more students to be connected to the bluetooth at the same time
- Customized question numbers: the system works with as many question as given by the instructor, which means updating the amount of question buttons and the entry of responses in database.

INDIVIDUAL APPENDICES:

Name: Linda Munisi # 22535158

In the whole duration of the project, most tasks in our group were worked on collaboratively as a team. For some of the big tasks, we divided them among ourselves in order to manage time and also be able to have someone responsible for each section.

One of the tasks I performed personally was the graphics in software and through hardware acceleration. Before completion of Sprint 1, I was responsible for creating graphics and displaying text that is being received from the bluetooth to be seen on the screen.

After Sprint 1, I was responsible for creating the database to store the data for the students responses and the instructor questions. In the beginning, i created a local database in android studio to store all the data that is entered by the student. The database that I created was local and upon investigating it looked like the information is stored in a file system and the developer has no easy access to the data that is stored in case they need to cross check the contents of the database.

Due to some challenges in the database systems, I decided to change the database and implement a real time database that the developer can see the contents. For this, I used a firebase database, which is a database system provided by google services. I was able to see the data that is being stored in the database and the different entries that it is stored into.

Another task that I implemented is combining of the database code to the whole app to enable the storage and the updating of the information on the app. I also created a way in which the instructor can access the data from the database and view them, and also send the questions to the touchscreen to display it for the students to view and respond to the questions.

Lastly, I was also responsible in helping to debug the integration of the bluetooth and the gps and combining all components together to make them in sync and to receive the data at the appropriate time. This part of the project was one of the difficult parts to pull through, we lost time that we could have used implementing extra functions in the project, due to the amount of time that we spent trying to make it work.

In general, or team was effective in our performance. We held scrum meetings in the beginning of each meeting in the lab, discussed what we accomplished since our last meeting, evaluated where we stuck and then discuss to apply more effort on a component that was likely to put us behind, while also setting goals and deadlines for the following meeting.

Moving ahead, I highly suggest that we spend time integrating everything as we go ahead. This is because we found out that some parts of the project would work well

individually but as soon as they were incorporated, then the whole project would break due to an unknown error. Sometimes, this was caused because of one part of the project getting stuck in a while loop and not being able to reach the other function calls, hence only one section of code works. For future developments we plan on integrating most of the android studio and hardware as soon as possible to see how the timing works and if the components work well together.

Because I was implementing the graphics and database portion of the project, to ensure that my tasks would integrate with those of the entire project, I communicated with my team to discuss how they preferred the layout and what would work best for their components. Also, I made the functions descriptive enough so that if they were to be called in a certain part of another task, then the integration wouldn't be too messy. I also communicated with my team to find out what information they wanted to be stored in the database and in what format they wanted it

Some hurdles that I had to overcome include complications in testing the graphics due to the touch screen not working and hence difficult to test the transition from one menu page to another. Another hurdle was on the application side when the app would crash with no sign of what was causing an error.

In the span of developing the project, I learnt how to time components for faster execution if needed, how to use hardware acceleration to accelerate the implementation of graphics in hardware and software using the NIOS II system, and using android studio to implement designs. Additionally, I also learnt how the configuration of putting and getting characters, the functions used to configure communication between the touch screen controller and the software work.

We did not have any team dynamic issues. Throughout the whole of module one all team members were actively participating, performing all tasks not only efficiently but also in a timely manner, and we also greatly supported each other's tasks as a team.

Some extra time was spent on learning how to use some android studio functions, and understanding what
components to use to get more optimal and better designs without the app crashing.

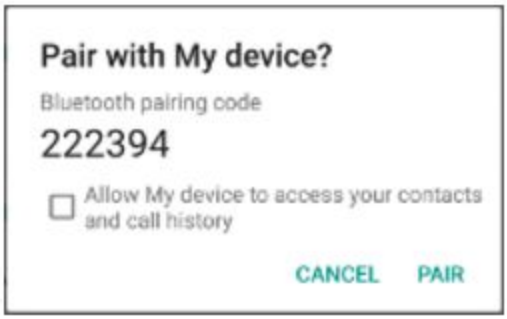
Name: Sirine Trigui #46443157

During module 2 period, tasks were divided among the team members depending on everyone's experience from last project,i.e module 1 . In case of getting stuck with an issue for several days, such was being solved collaboratively.

The tasks that I was responsible for in module 2 were:

- Setting up and establish a basic connection between the android side and the provided bluetooth dongle.
- Adding a map view to first create the outside class feature and second to improve our project's GUI.
-Integrating the GPS to extract student s arrival time which will be stored in the database to determine whether the student was late or not.

The team's goal for this module's sprint 1 was to successfully make the communication between the android App and the hardware components through the bluetooth dongle. For that, I started by going through the posted tutorials, some specific lectures and following some youtube videos to get such to work. Following the tutorial steps, first i started by adding bluetooth permissions(Bluetooth_admin & bluetooth) in the manifest.xml file.Moving on, to set up the bluetooth,I had to create a bluetoothAdapter class which returns a bluetoothAdapter to be used as our android device bluetooth. To find devices(paired and ones to be discovered), I started by gathering the paired ones if any and save then in a listView in which a bluetooth icon was put on the right and the bluetooth mac address is right next to it. To scan the bluetooths around, I created another listview in which such are being saved and can be paired if the user wants to with asking for permission to be paired first.



Once a paired one is being selected to connect to our bluetooth dongle(hardware component),discovery is being disabled as it slows down the process plus no need to continue discovering.After creating a socket through which the user is able to connect to bluetooth device, I added some functions that manipulate the read and write functionalities.

By sprint 1 deadline, we were able to send and receive data through bluetooth,hence establishing the communication between DE1 and the Android App.

After Sprint 1, I was responsible for integrating the GPS to the project to extract the time from.The data being extracted is to determined the students' arrival time to help the instructor to assign attendance marks. Such was a bit challenging as we had set it up with the whole logic and the hardware was not cooperating much.

To add more features and make our project to be used outside of the class,I added the map view functionality. After creating an API key and adding the appropriate permissions, I was working on extracting some information from the Json file that was generated as the user entered their origin and destination locations. Getting the origin and destination, we provided the distance in meters or kilometers and time that will take the user to reach their destination in minutes. Clicking on the ind Path button, such will be displayed and two markers A and B are being placed on the map directing the user.

Team members were supportive during the debugging process especially for establishing the bluetooth communication.

After each scrum meeting at the beginning of each lab, tasks accomplished were being placed under the done list where as tasks in progress were being worked on. During the meeting, each member described her roadblock if there was any and some suggestions have been given. Setting goals for next scrum and reminding ourselves with the upcoming deadlines were as well discussed.

For our future, I suggest that we start integrating working individual features as soon as they are done. Through module one process, we experienced that some parts of the project would work well individually but as soon as they were incorporated, then the whole project would break due to an unknown error. Such error could be due to being stuck in a while loop so that the following feature is not being reached. Even though we have been refining our logic, some details could not be seen until the actual integration. That is going to be our plan that we will follow for module 2.

Communication with other team members saves a lot of issues and helps the integration of the individual features. We agree at first about what type of values each method should return, so that it can be easily integrated. Adding comments and following the standard coding style in this case made our lives a lot easier as we had a lot of code.

Hurdles that I had to overcome include complications in getting the GPS to provide us with an appropriate data as sometimes it skips a digit and in integrating features and getting them to work all together with the logic that our team agreed on.

Through the development process, even though I dealt with Android Studio before, the experience was a lot different on this project since I worked mostly as a back end member whereas in my previous experience I worked more on the GUI. For example, I did not know how to get a map view on an App and that in fact, it requires an API key

which needs to be restricted.I also learned that bugs can be generated from both software and especially hardware. I improved my testing skills through performing regression tests by applying one change at a time and the record both what the bug was and its fix in case the same bug is being reproduced. Also, I got a taste of how both software and hardware can communicate through bluetooth and how everything can be controlled through such communication.

Team dynamic did not encounter any issues. All of us were responsible, offering help in case of need. Not likely as we were at a certain time a little bit behind on Module 1, We were on time for this module as we agreed to hardly respect the deadlines and started integrating as early as we could.

Name: Yixin Zhao #49427140

In Module 2, our team worked closer and was more efficient than Module 1. We helped each other in the debugging process and discussed a lot about our project when creating the APP.

During Module 2, I worked on exercise 2.2 which is drawing graphics. It was interesting and not too difficult, since there were many documents to look at online.

Before Sprint 1, I was working on the Android APP to create activities and modify GUI. The design of our project was developed with my group members while creating the Android APP. After the activities were done, Linda and I worked together on the communication between the Bluetooth dongle and the Android APP. We modified the send message function so that it does not read forever, and wrote the code on Eclipse to read from the Bluetooth dongle. We used some special character to determine the beginning of a message as well as the end. We also modified the Bluetooth code to stop discover process if the wanted Bluetooth dongle is paired already or discovered. This saves some time for the users as they don't have to wait till the discover to be finished to move to the next page. We decided to get the project done with only one student connects to the dongle and then we will try to upgrade the project if we have time.

After Sprint 1, I created more activities as we discussed. I added the instructor side of the APP, the send answer page and question button page along with needed functionalities. As we all agreed, it would be fair that students who are late to class can not answer the questions they missed. Therefore, I put the question buttons to control which question is available to student through the interaction with DE1 board. This page also helped us to determine which question is the student response to when we update the answer in database. The time when student sign-up is from the GPS chip, which is sent through Bluetooth dongle. So the APP sends a message to Bluetooth dongle asking for the time when connects to one, then the Eclipse reads it from the GPS and sends it through the dongle. The APP should wait for the time then store it in the shared preference. As the database will be updated when students answer question, the student number and time stored in the shared preference will be used to create a new entry in the database when student first hit "submit" along with updating the first response answered.

I also worked with Linda on integrating the database. We modified the APP in order to provide necessary information for database. For instance, we changed the student page to take a student number. This information is needed for later to store student's responses and sign-up time in database. We stored it using shared preference. The date is from the function SimpleDateFormat(). The list of user from database was accessed when we update it. If the list is empty then we create the entry

immediately, otherwise loop through the list looking for the same student number. If the entry exists, we use the message send from previous question page through function putExtra() to determine which question should we update, otherwise create a new entry and update it.

Then I worked on the integration and the communication between Eclipse and the Android App. It was a bit tricky to get the Bluetooth reading function as well as the Touchscreen detecting function working together. Both of them require constantly checking status of a register and the code was running in a while loop. I figured it out by putting the larger while loop condition as checking Bluetooth incoming register. If there is any data received, then read the Bluetooth until the end of the message, otherwise keep checking the Touchscreen. I used a special character as the signal for asking time. Therefore, when Bluetooth dongle get this signal, Eclipse reads the time from GPS for only once, then sends it to the APP. GPS reading time function is put in a function so that we don't have to worry about its while loop problem. The Android side waits for the time message after connecting to a Bluetooth dongle.

For the question buttons, I deactivated all of them from the beginning. In Eclipse, the code keeps track of which question is shown now. We tried to use an integer to track it but the Touchscreen was too sensitive, so pressing it may add or decrease more than "1" to the integer and ended up not giving us the correct question number. Therefore we switched to directly assigning values in the "case" condition. For the interaction, when the instructor click on "Start Polling", Eclipse sends question number as a String through Bluetooth dongle and we activate the question button accordingly. Because every time the APP directs students back to the question button page once hits "Submit" button, all the button are deactivated as the intent is newly created. It prevents the student from answering missing questions.

Our team was being effective through Module 2. We had scrum meeting in every lab session. First of all, checking everyone's progress; then, we would adjust our plan based the progress, and assign tasks accordingly. We also met during weekends to work on the project or debug with each other. In general, most of the work was done in paired coding and we collaborated a lot during this Module.

This time, our project idea was not well-developed from the beginning and has been changed a lot during the whole process. Therefore, it was hard to divide the tasks very clearly or making schedules for individuals. The project went smoothly though since we were only adding on to the original idea. I would suggest to avoid this situation next time because it could waste our time to work on tasks that later on become unuseful. We learnt our lesson from Module 1 so the integration was not messy. The communications between several components were completed at the end yet everything else was combined during the process.

The project seemed to be confusing at the beginning but as soon as we got into it, it became more and more clear. However, the Bluetooth connection was very hard to debug. The code was all in a big chunk so it was hard to read or debug. And when the project crushed because of the connection of Bluetooth, we had no idea how to fix it. It was mostly because of the timing when we tested the instruction side along with the student side. This was very frustrating and time-consuming. Also, the interaction between the APP and Eclipse as well as the communication among each hardware components were quite hard. The logic was a bit complex and I got lost very easily. But it turned out the logic and timing were all that mattered. As soon as I figured those out and started to follow the correct path, all that require was several attemps to find the efficient way to do it. This time I learned a lot on the Android side about the storing and communication. I also reviewed what I learned from Module 1. The hardware was all done from previous Module so the integration of Eclipse code was a refresh of memory. There was also a bit of database that I learned through working with Linda.

Our team did not have any dynamic issue. We were being supportive to each other during the whole time. Problem was being discussed among everyone. We provided ideas and tried possible solutions together when a member met a challenge. And we communicated effectively and very often so that every member knew what the others were working on, what to expect next and when to expect the task to be done.