Final Report

SEG2105 - Android Household Chore Project

Trevor Siu (7866981)

Chieh Ko (8477221)

Yujia Hang (8433929)

Team Member Contribution:

	Deliverable 1	Deliverable 2	Deliverable 3	Deliverable 4
Trevor Siu	40%	30%(UML diagram)	50%	30%
Chieh Ko	35%	35%(Sequence diagram)	30%	40%
Yujia Hang	25%	35%(State Machine Diagram)	20%	30%

Introduction:

The application we developed for the final project is a household chore management application used to assign members in a group chores and tasks. The goal of the application was to provide a simple user interface that allowed family members to communicate and work together in a efficient and timely manner to accomplish required tasks. This is done by allowing users to create accounts and join a group of people to complete tasks and obtain rewards. Throughout this project, aside from developing the actual application, a lot of planning was involved to provide a clear concept of the application itself through deliverable submissions ranging from design specifications, use cases, UML diagrams and layout designs.

Objectives:

The objective of this project was to allow students to gain practical skills and knowledge in the development of android applications using the provided Android Studio software. Additionally, this project allowed students to implement the concepts learned in class and to practice collaborating in a group environment. Finally, the project allowed students to explore the Firebase database environment and had an optional task for students to allow users to create personal accounts on the database.

Software Requirements (Deliverable 1):

Function Requirements:

1. The system shall allow user to create tasks with description and task name.
2. The system shall be able to choose between displaying the entire group tasks or just his own tasks.
3. The system shall let the current user know the score of all user in the group.
4. The system shall allow user to sign out from the current account.
5. The system will allow all user to assign tasks to other members.
6. The system will send a message when a task is deleted.
7. The system shall send a message to the user to confirm score has be added after completion of a task.
8. The system will allow user to change their icon.
9. The system will send a confirm message before deleting a task.
10. The system will allow user to edit the detail information of a task(name,description,due date).
11. The system shall have an inventory page to show items needed for all tasks.
12. The system will user to change status of a task(Regular, Urgent).
13. The system will only allow user to assigned items that are in the inventory to a task.

14. The system shall allow user to delete task, which will remove the task from the database but no points added for completion.
15. The system shall allow users to adjust the amount of items on the shopping list.
16. The system shall require both name and amount when user is adding a new shopping list item.
17. The system shall display a task in red when it is classified as urgent.
18. The system shall be able to accept new users.
19. The system shall require email, first name, last name and password to create a new user.
20. The system shall not allow user to complete tasks for other user .

Non-Functional Requirements:

1. The system shall be developed on the Android Studio platform.
2. The system shall be connected to a real time database.
3. The android application shall not exceed over 100 megabyte and should be around the average of 15 megabyte.
4. The system shall be compatible on android version 6.0 and above.
5. The system must be implemented before Nov / 20 / 2017.

System Use Cases:

Use Case 1:

Name: Updating items on the shopping list

Actor: User

Precondition: There must be at least one items on the shopping list

Summary: When the user wish to update an item on the shopping list, he/she will go to the shopping list page and select the item. System will ask if the user need to update the amount and name. System will update the items database accordingly.

Actor Action	System Response
1.Choose the shopping list page. 3.Select the item they would like to update. 5. Change the update amount or use the autofill amount click “Update”.	2.Display shopping list page 4a. Display item update dialog 4b. Prompt actor to answer if they need to modify the amount and name. 6a.Update amount. 6b.Update item name.

	6c.Display shopping list page.

Use Case 2:

Name: Creating a new user

Actor: User

Precondition: User is in the sign in page and user not signed in

Summary: When a user choose to join a group in the system, the user must enter a invitation code and if access to the group is valid, the administrator shall be notified.

Actor Action	System Response
1. User runs the application 3. User selects “New User” 5. User will fill in the form and confirm through the submission button.	2. Display sign in page. 4. Display Form page and prompt user to fill in requirements 6. Create new user, sign in user and display Home page

Use Case 3:

Name: Create Task

Actor: User

Precondition:User is signed in and is at the Task page

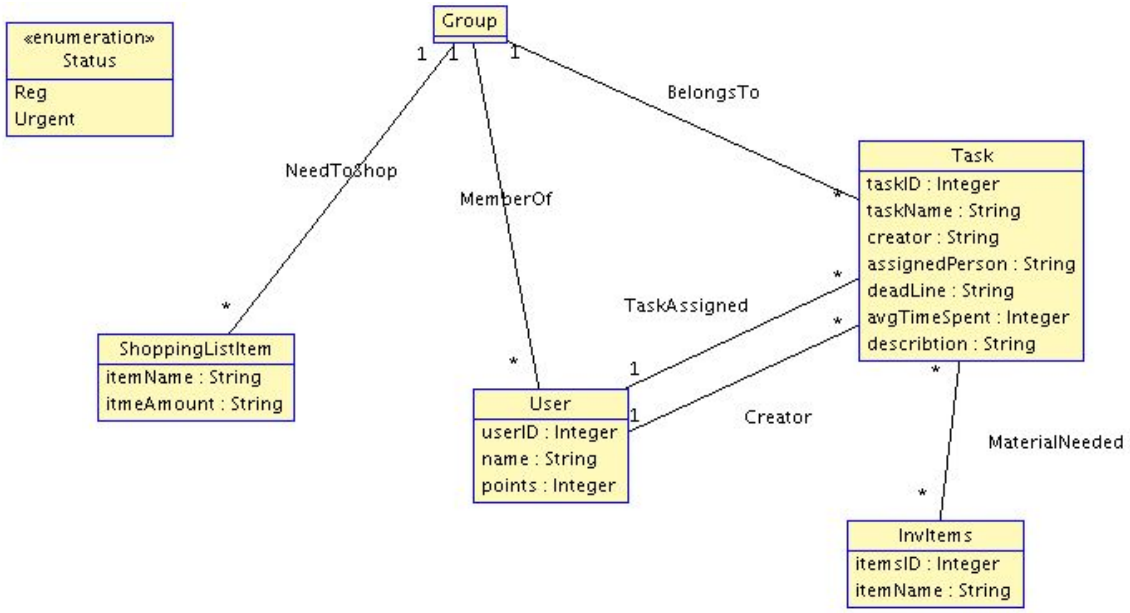
Summary: When the user wish to create a task, they will go to the create task page. User will be ask to fill in the form to create a new task. After a task is created a message will be sent to the user confirming the task has be added to the database.

Actor Action	System Response
1.User click the create task button 3.User fills in the task description and confirms through the submission button.	2a. Display the create task page 2b. Prompt the user to fill in the task description 4a. A task object is created based upon the information provided and is automatically added to task list on the database 4c. Send a toast confirming the task has been created

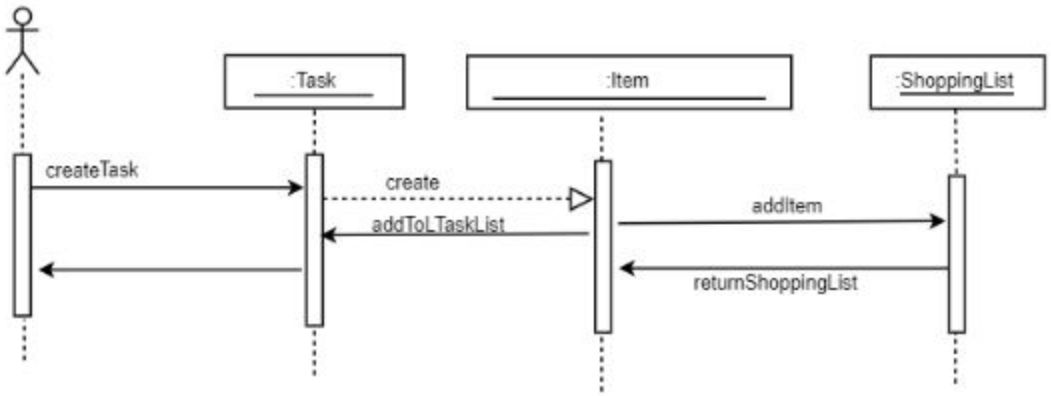
	4d. Display Home page

UML Design (Deliverable 2)

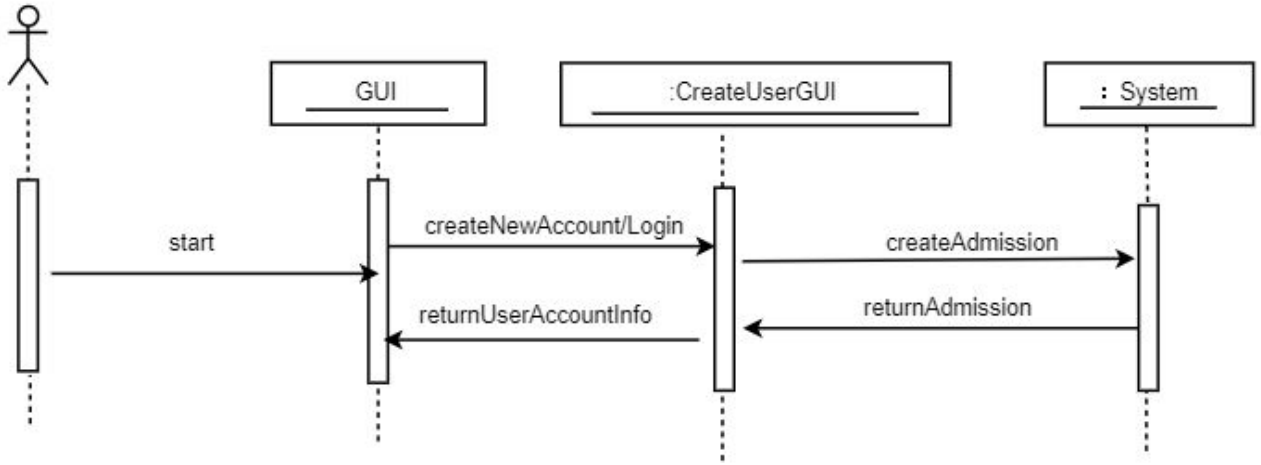
Class UML Diagram



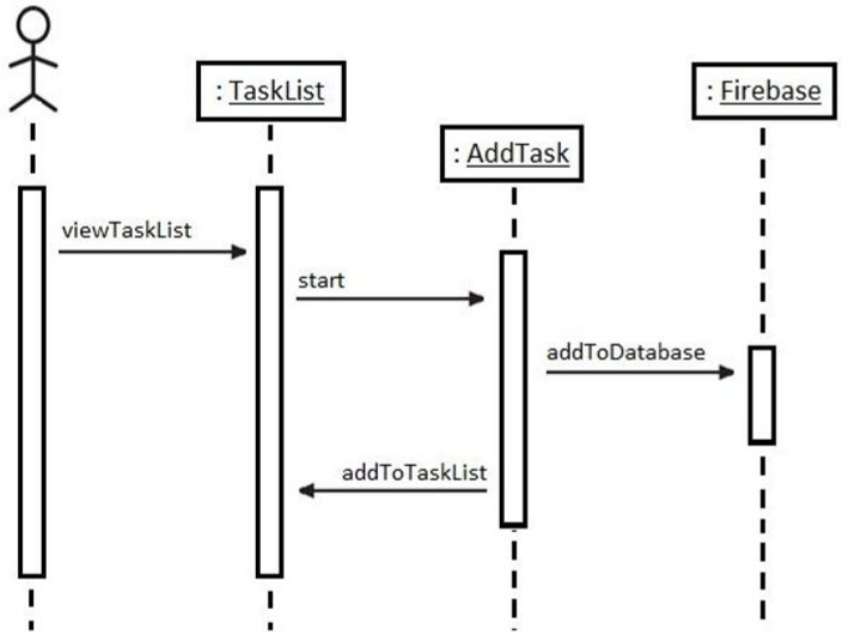
Sequence Diagram



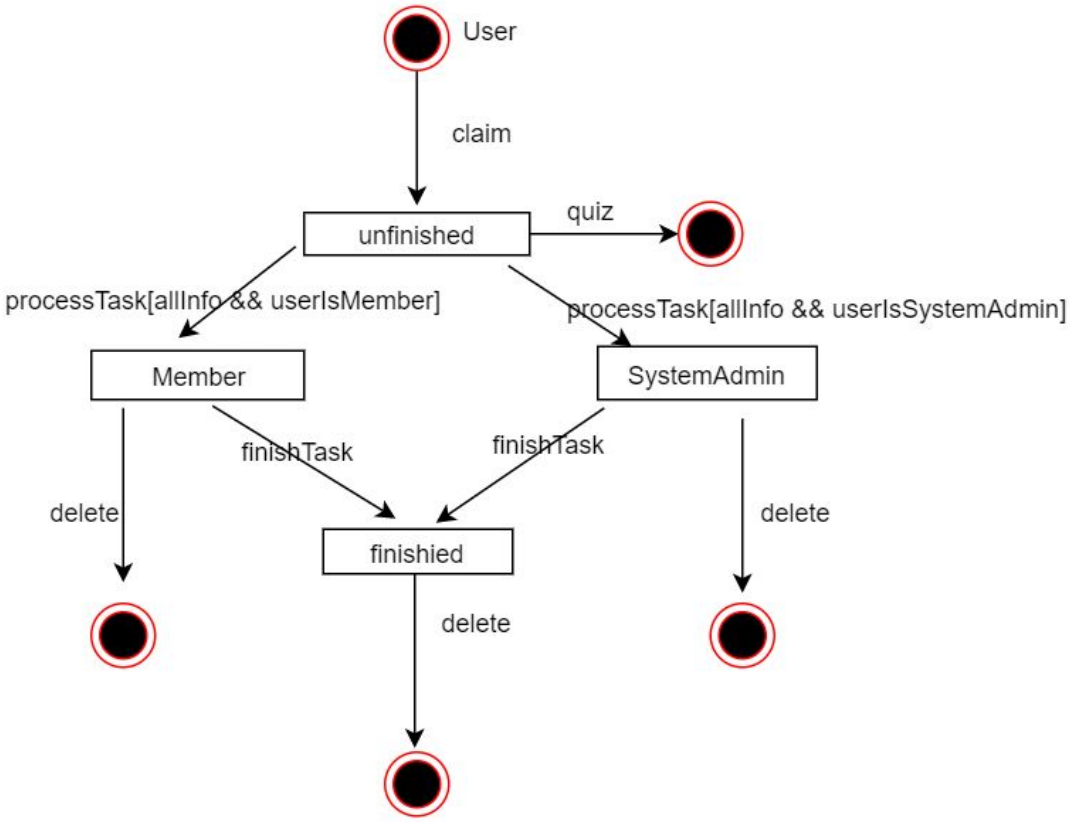
Sequence Diagram #2 - Creating a New User



Sequence Diagram #3 - Adding a New Task

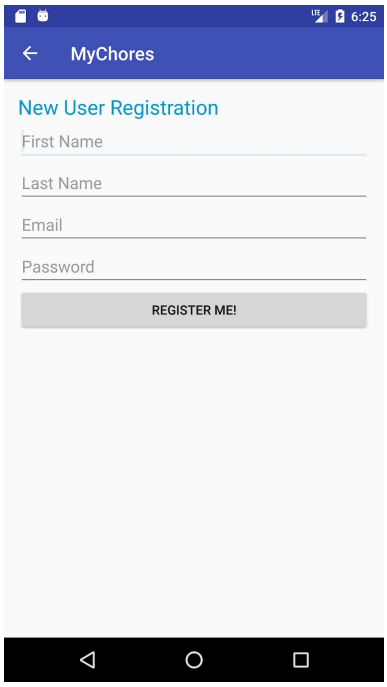


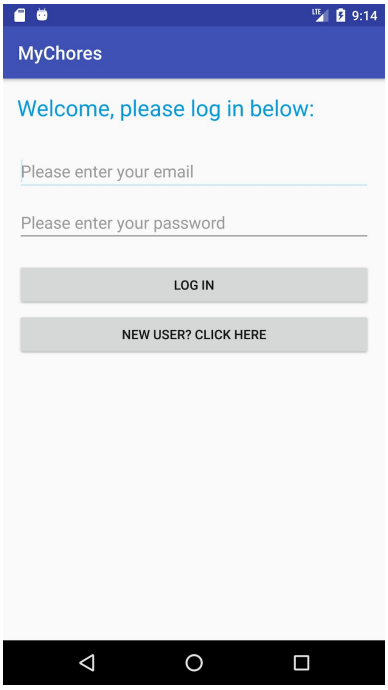
State Machine Diagram



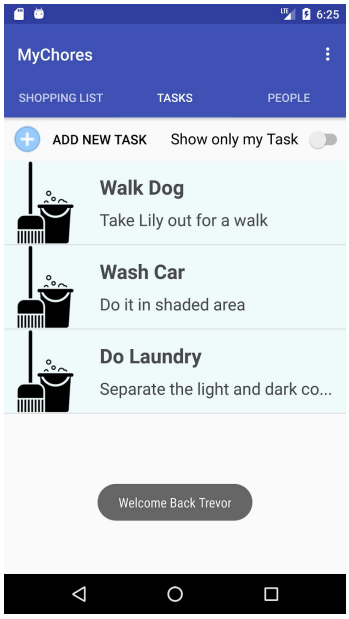
User Interface Screenshots (Deliverable 3):

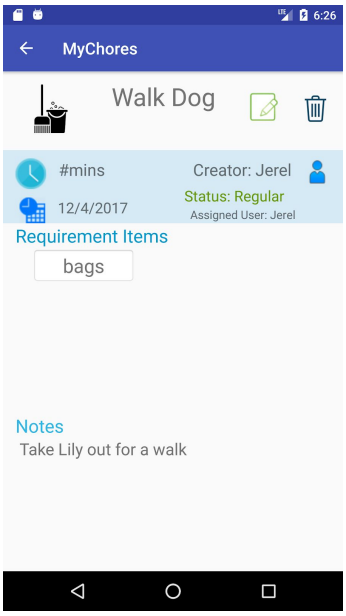
The overall design of our user interface will be simple and user friendly. Opening the application and logging in as a user will allow one to have customized information based upon him/herself. The home page of the application will essentially consist of the three main tabs perceived as the shopping list, tasks and people. Navigating between the pages are as simple as tapping the tabs or swiping across the screen between the pages while performing actions are as easy as scrolling and clicking on buttons that will guide the user to the next activity. The following screenshots shall depict the functions and concepts of our application.

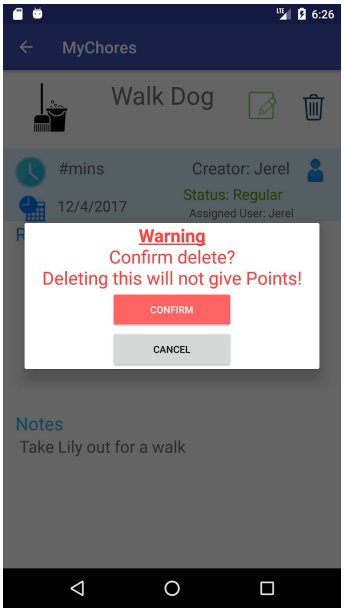




The above figures are the user sign in page. This page is first displayed when the application is executed and will prevent users from proceeding if the required information is not entered. Additionally, it will verify if the account is unique and will provide appropriate user feedback through toast messages.

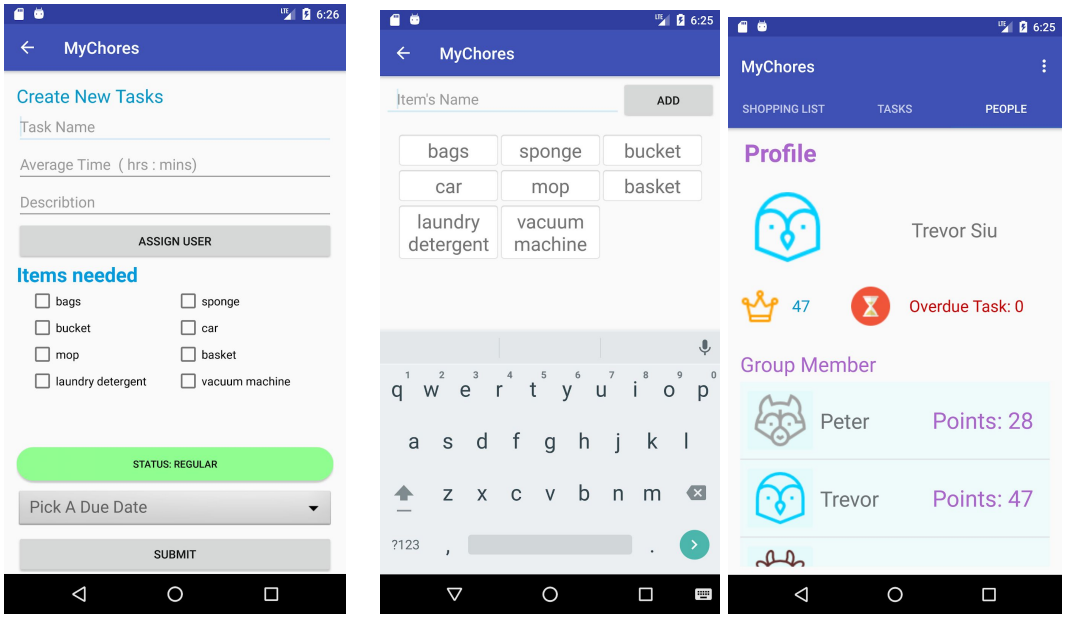




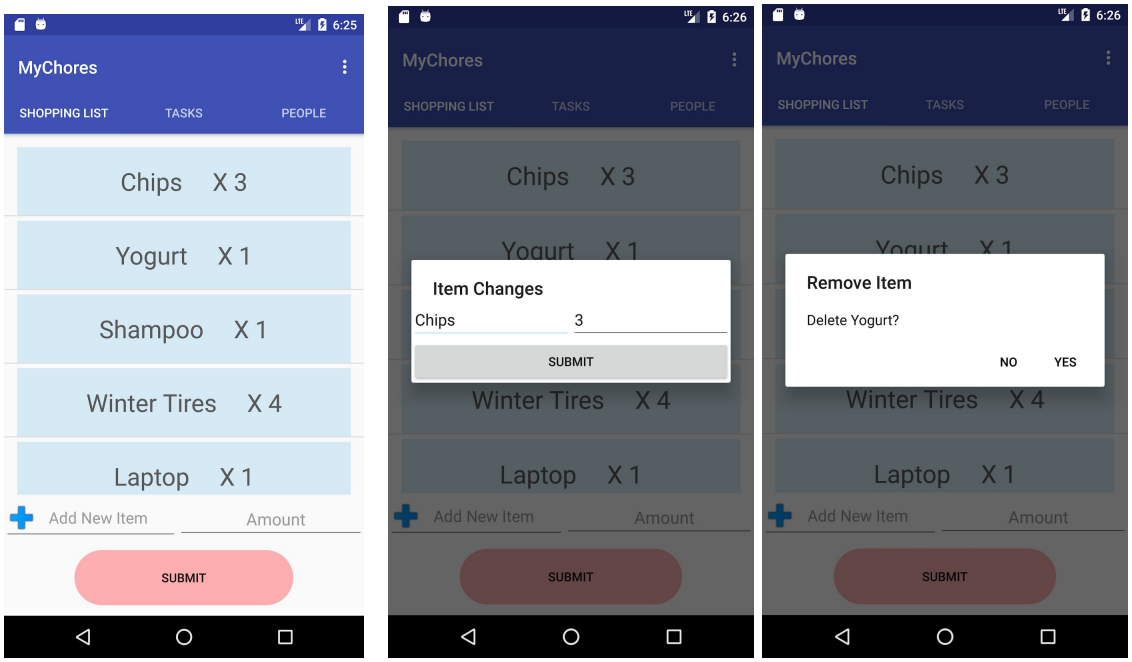


The above figures are screenshots for the task pages of our application. When a user short press on a item in the list of tasks, the application will allow the user to edit task information. A

long press on the task would allow the user to confirm that they have completed the task. However, this long press only works when the switch is switched to “show only my task” to prevent the user from accidentally confirming a task is finished when it is not. When creating a task, in the case that a user did not enter a required information, the application will not allow the user to proceed. If the did not enter something that is crucial, the app will proceed and automatically fill in the empty boxes with the appropriate message. FInally before deleting a task, a warning is displayed to double check the action the user wishes to perform.



In the above figure, the first image is part of creating the task, second image is the inventory which is the items needed list. Finally the last image is the profile tab of our application that shows the current user and other users in the same group. Notice users have reward points for completing tasks and an overdue feature. Users may change their profile picture by clicking on their profile and selecting from images that are preloaded to the application.



The above figures is for the shopping list tab. This allows the user to create a shopping list and edit it at any time by long pressing the item on the list. A short press will prompt the user if they would like to delete the item from the list. If no information is entered, the application will prevent the user from submitting the empty item.

Lessons Learned:

After programming the android project, we learned a lot about Android Studio and working with the development environment. When building an application, we learned that the application can be divided into smaller parts by having different classes for more generalized functions. Additionally, Android Studio allowed us to simplify the task of designing the user interface since it is possible to simply drag buttons, switches, textboxes, etc from a list and the code will automatically be generated. One major skill that we improved on was debugging the android application using logcat and breakpoints.

Aside from the android application itself, we learned a lot about the planning process of developing an application and the research needed prior to start developing the app. The importance of class diagrams, functional/non-functional requirements, sequence diagrams and other deliverable related tasks following up the development of the application allowed all members to understand more clearly of the general idea.

Finally a large section of our application depended on firebase and it's features. We were able to experiment and learn how to organize our application information into a database and have

it accessible on multiple devices. We were also able to secure the information through a user email login which is a really neat feature as firebase prevented many hassles for developers in such a way that the code for login and database did not have to be written from scratch.

Conclusion:

This project overall was by far one of the coolest projects we've worked on so far in our years of study. It allowed us to gain a lot of hands on experience in things that were covered throughout the course and the labs really prepared us for the final project. Even though a lot of time was spent on debugging and thinking of how to implement functions for the android application, a lot of experience was gained and it was definitely very rewarding to see the final product that was produced. One suggestion for improving the course term project next year may be to cover breakpoints and how to debug with android studio as we found debugging was a big part of the project and was not covered as much in the labs.