[TASK TRACKER]

Final Report

Information Technology Capstone Project

COMP5703

Group Members

1. Lanxin Zheng (510080761) [Group Leader]
2. Yupeng Liu (510151153)
3. Dali Yuan (500646054)
4. Xufeng Zheng (510564869)
5. Ce Gao (500324657)
6. Lingda Zeng (510316040)
7. Zhengyu Jiang (510640406)

CONTRIBUTION STATEMENT

Our group, taking project CS75-1, with group members Ce Gao, Dali Yuan, Lanxin Zheng, Lingda Zeng, Xufeng Zheng, Yupeng Liu, Zhengyu Jiang, would like to state the contributions each group member has made for this project during semester 1 2022:

- [Ce Gao]: I mainly implemented the notification page functions, including the display of the message list, the definition and storage of two message categories, the layout of message items, etc., the delete function of parent/creator permission messages, and the message background and some interface beautification of the message list.
- [Dali Yuan]: Complete the development and design of all functions related to the goal part of the application. This includes: creator&parent users' viewing, adding, modifying, deleting, and submitting goal functions(for their child users) and pages , and child users' viewing goal functions and pages. Design and implement the goal-related application data structure and database access, complete the UI design and optimization of the application's goal-related pages, and complete the user input detection of all goal-related pages. Complete individual reports and participate in the division and writing of each group report.
- [Lanxin Zheng]: All pages' page beautification in Main Interface, such as edit user information Interface, children account progress display Interface and so on. Login page beautification, registration page beautification. Implement Creator and Parent edit other family member information function and account can remain logged in state after logging out. Communicate with client about weekly meeting time, group report preparation and revision, etc.
- [Lingda Zeng]: The functions of creator and parent personal information interface, the reading and display of user basic information, and the modification of user basic information, follow the interface design of creator and parent after the completion of subsequent functions
- [Xufeng Zheng]: Mainly focus on the function implementation. Database configuration and deployment. Log in, log out and registration function implementation. Google log in implementation. QR code generation and QR code login implementation. Display users, add users, delete users, view user's information (for Creator, parent, and child) implementation. Splash screen design. Session implementation to remember user's information. Some basic design for the main page.
- [Yupeng Liu]: Mianly part of functions related to task, including viewing, adding, editing, submitting tasks for parent and creator users and viewing tasks for child users. Implemented the entrance page of setting tasks and goals part. Designed the data structure of task, including the application and the database. Polished all the pages related to the task setting and the entrance page of setting tasks and goals. Wrote some parts of project proposal and final report.
- [Zhengyu Jiang]: Implement functions related to child personal information interface, including the display of child basic information like name, current score, circular progress bar, biggest available goal and milestone status. Design the pages of showing the child's completed task and exchanged goal. Implement some

functions that a group member view the basic information of other children, including showing the last completed activity of a child, the last exchanged goal of the child and the next target goal of the child. The beautification of the user interface related to child personal information interface.

All group members agreed on the contributions listed on this statement by each group member.

Signatures:

Lanxin Zheng:

Yupeng Liu:

Dali Yuan:

Xufeng Zheng:

Ce Gao:

Lingda Zeng:

Zhengyu Jiang:

ABSTRACT

With the current popularity of intelligent machines, more and more children are getting used to the convenience brought by machines, leaving many family daily affairs or some activities to machines, and gradually losing contact and communication with their families. Order solve this problem, the project developed a family APP “Task Tracker” using Kotlin and Firebase cloud database, for this app, we mainly develop it for the mobile android mobile platform. Then, with regard to the phenomenon that children gradually lose communication with their families, the task mechanism of this family task tracking app can enable children to communicate and discuss more, thus increasing the time for parents to communicate with their children at home, during the follow-up activities and the completion of task objectives, the app has set up a corresponding task tracking mechanism, which can help parents better track their children's task completion and progress, and effectively monitor their children's activities, so that parents can better monitor their children and educate their children, at the same time, in order to stimulate children's motivation and interest, the app has also added corresponding reward and punishment mechanisms. Through this step-by-step process, children can re participate in family activities and gradually return to the family.

This project focuses on several main points: First, the children are unwilling to participate in family activities and lack interest in the boring process of family activities, resulting in a lack of communication with their families for a long time. Second, parents can effectively supervise and timely adjust their children’s activities. Then the key requirements of this part will be gradually realized in the project development

TABLE OF CONTENTS

1. INTRODUCTION

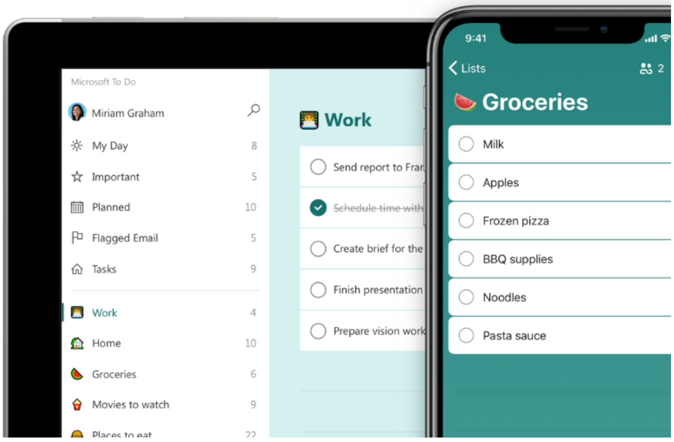
With the widespread popularity and application of mobile applications, and the increasing demand for mobile terminal products, the demand for web products is decreasing, so in this project, we mainly develop for the mobile platform, while choosing the more widely used android mobile platform. This project needs to develop the family task tracking APP, its purpose is to help children, who lack communication with their family for a long time, and due refuse to participate in family activities and are gradually become disconnected from the family return to the family. Therefore, in this APP, family members set activities and tasks that need to through communication with children, then use corresponding reward and punishment functions to let children re-participate in family communication, besides, every activity and score can be clearly displayed on the app. At the same time, parents can also track their children's activity process in real-time through the app, so that their children can get feedback on every effort, gradually stimulate their interests, cultivate their sense of responsibility and return to the family building.

2. RELATED LITERATURE

In this project, we developed a task tracker application for families to use to track children’s daily life tasks and reward them. Most of the applications about our application have been reviewed. Considering the application is a mobile application, our reviews are limited to mobile application markets including Apple AppStore and Google Play. Some applications are providing task tracking functions but none of them is designed for family users.

Two types of task tracing applications are concluded based on our reviewing. Personal use is the first type. People may use these applications to create their list of tasks and manage their daily work. The second type is the applications that can help companies and groups to assign work to employees or group members. The leadership of the companies or groups may use this type of application to evaluate the performance of each member in the groups or companies. Some detailed information about some famous applications counted in these two types will be introduced in the next section.

2.1 Microsoft To Do

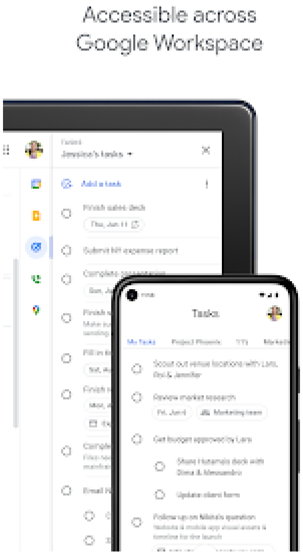


Picture 2.1 Microsoft To Do App (Microsoft, 2022)

Microsoft To Do is an application that can help users to make a priority-based tasks list to manage the tasks easily (Microsoft, 2022). It can be operated on multiple platforms including Windows, iOS, Android, MacOS and Web. As a part of Microsoft Office 365 service, all the data can be synced between these platforms via the Internet.

People can use the application to create their own tasks lists by themselves. Tasks assigning function is not the scope of the service. Thus, the application may not be suitable for family use. Parents cannot arrange tasks for their children with this application

2.2 Google Tasks

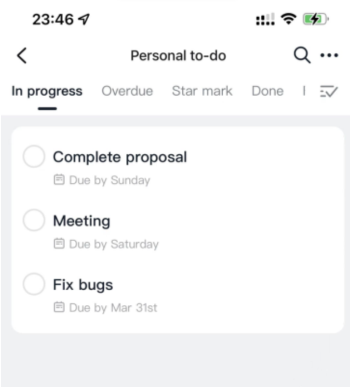


Picture 2.2 Google Tasks (Google LLC, 2022)

As a competitor with Microsoft, Google also provides their to do list application, Google Tasks. Google Tasks provides a similar service as Microsoft To Do. Because Android is developed by Google, the collaboration with other Google services such as Google Calendar and Gmail should be the advantage of the application. The Android system environment can provide users a better experience while using the services and applications.

Just like mentioned in the previous section of Microsoft To Do, Google Tasks is also designed for individual use. Family users may not use the application to track the tasks for children properly.

2.3 DingTalk



Complete proposal
Due by Sunday

Meeting
Due by Saturday

Fix bugs
Due by Mar 31st

Picture 2.3 Screenshot from DingTalk Application

DingTalk, developed by Alibaba Group, is designed for groups or companies to manage tasks and communications. With the instant messaging functions, the team members and companies' employees can communicate online. There is also a task managing functions to help the project team leaders monitor the progress of the projects.

It is a suitable application for remote working and it can increase the efficiency of remote working (DingTalk, 2021). During the Covid-19 pandemic, many schools have to teach their students remotely. Some schools chose this application to provide a remote study environment with a special edition of this application, DingTalk Lite. The special edition is made for education users, which can provide online broadcasting, assign and receive homework functions (DingTalk, n.d.).

DingTalk and DingTalk Lite are designed for corporations or work teams. Although the functions of the application can support the parents to track tasks for their children, it is too complicated to learn how to use the application, the children may not understand how the application works.

3. RESEARCH/PROJECT PROBLEMS

3.1 Research/Project Aims & Objectives

The goal of the project is to develop a family interaction app based on a scoring system, which transplants the traditional model of children doing household tasks in exchange for prizes online. It allows parents to set tasks more efficiently and children to redeem the prizes more easily. The app provides a bond of intimacy, better-linking parents and children and enhances family interaction.

In addition, as a directed cultivation tool, it can train children's handy skills in doing housework, which is conducive to cultivating children's habit of getting paid through labour, developing an independent character, and improving the child's ability to live independently.

3.2 Research/Project Questions

The task-reward family interaction APP is a market gap and has to take the risk of being the "first", and the market reaction may not be as expected.

The task-reward parent-child interactive APP transplants the traditional task-reward model from offline to online, and some functionalities may not be implemented as the complexity of the transplantation.

The initial app was written in React, and the client required a rewrite of the code as they wanted to use Kotlin.

The app used the Firebase Realtime database service as a third-party database for the login and registration functionalities, therefore provisioning the Firebase environment and dependencies would create a heavy workload.

User roles include Creator, Parent, and Child. Different user roles operate differently and have different permissions. Creator users log in using traditional, while the rest of users log in using generated QR codes. The parent has the right to create and manage tasks and goals, and submit them when the child completes the task. Child users can only see the tasks assigned by the parent, and other operations cannot be implemented. The diversification of user roles brings difficulties in functional realization and increases the risk of the project.

3.3 Research/Project Scope

3.3.1 Within the scope:

1. Write the app in kotilin language.

2. Use Firebase Realtime Database as the database for the project and develop the app.

3. Create three user roles with different permissions.

4. Function realization of family interactive app based on scoring system.

5. The user interface is as beautiful and consistent as possible.

6. App maintenance management.

3.3.2 Out of scope:

1. Android/Ios dual system porting.

2. The realization of the two-way scoring function for children and parents.

3. The realization of the theme switching function.

4. METHODOLOGIES

4.1 Methods

For method, we use the combination of Waterfall and Agile to finish our project. For our working implementation, we follow the procedures of Waterfall in sequence. In another word, we design after defining requirement, develop after designing and test after developing. The reason why we follow Waterfall is that we have already got the scope of our project and it is not likely to change in the future. Follow the steps of Waterfall in sequence will save our time of finishing our project. However, we still retain some format of Agile. For example, we will organize weekly meeting with our client and we will get feedback from him so that we can apply the feedback in the next week. We organize an internal meeting every week to discuss our individual progress, the difficulties we encounter, and the plan for the next week. The format of Agile helps us adjust our workload or distribution, learn new things effectively and finish our project in time with the most ideal functions.

4.2 Data Collection

We utilize the database in Firebase. Firebase is an application development platform which helps developers quickly develop a high-quality application. After we connect our project with Firebase, we are able to write code in our project in order to access data in Firebase database. The database in Firebase is a NoSQL. As a NoSQL database, the data is stored as key-value format. For example, in our project, there is a conception called child. As a child, he has many properties. For example, his name, the current score he has, the tasks he has completed and the rewards he has exchanged. All of these properties are stored as the keys of the child and corresponding value can be set to the keys. Specifically, by accessing the data in Firebase database, we need to utilize relevant Firebase classes and invoke their relevant methods based on our purpose.

4.3 The technique in the project

For development language, we choose Kotlin. Kotlin became the official language of android development in 2019. Before 2019, Java was the official language of android development. Compared with Java, Kotlin has many advantages. Firstly, Readability. By accomplish the same function, it is able to write less code by Kotlin than Java. Secondly, Null-safe. NullpointerException is the most annoying exception to developers. In Java, NullpointerException is a runtime exception. However, Kotlin can deal with this issue at compile time. It will greatly reduce the time of API debugging and make a project runs smoothly. Thirdly, in Java, we need to write getter and setter method for model classes. These classes are mainly intended to store data. In Kotlin, we do not need to write getter and setter method for them. In addition, compatibility. Compatibility means Kotlin has been made to be compatible with Java. It can be used with existing Java classes and won't cause any errors. The compiler will allow the code containing the Java and Kotlin classes to work flawlessly.

4.4 Deployment

We will zip our project as a zip file and deliver it to our client. By deploying our project, our client needs to download android studio in his computer. If he wants to run it on his android mobile phone, he needs to connect his computer to his mobile phone with a data cable. If he detects some bugs or some functions that are not meet demand, we will fix it on our side and give him the latest zip file after we finish.

4.5 Testing

Testing is a necessary step in software development. We use Espresso to write unit test to check whether all developed functions work as expected. For every single function, we will emphasize on considering edge cases in order to make sure the function will work as expected with any input. When all of our unit tests run successfully, our project is deliverable in principle. Unit Testing can help developers do many things. For instance, it can help developers debug functions and check whether these functions satisfy their expectations. In addition, it can help developers test the performance, as the time-consuming of implementing a function. Moreover, it can improve the stability of a project when it is refactored. Developers just need to run the unit test they write before to test whether a function is influenced by the refactor.

5. RESOURCES

5.1 Hardware & Software

Task Tracker runs on Android smartphones and the Android mobile emulations on Windows, Macs and other systems.

5.1.1 Software

When developing Task Tacker, Android Studio was used the project.

Android Studio 7.0.2 (and the more advanced version): Android studio not only provides a Kotlin programming environment but also provides a lot of software development tools that can help the app building and developer. Like,

1. Android Virtual Device Manager: It enables developers to manage AVDs running on Android virtual machines.

2. Android Emulator: Using the Android Emulator QEMU-based device virtual machine, developers can run their applications and perform debugging and testing in an Android runtime environment that is real-time.

3. Hierarchy Viewer: Assist developers with user interface design. It can update the position of interface components in real-time once developers make changes to the interface, reducing the necessity to execute an app.

4. Android Monitor: Logcat and some networking capabilities are all incorporated into Android Studio for the sake of application debugging and helping developers to analyze bugs.

5. Gradle: It is used to build projects and to help them run. And, it includes a build cache to improve the efficiency of Maven builds. Its script format is easier to understand than Ant's. Ant's script configuration format is XML, and unless the project is quite small, the XML files quickly become unmanageable.

5.1.2 Database

Firebase: It provides free access to real-time databases, error reporting in great detail, authentication, and cloud computing capabilities, all of which were essential to the development of our mobile applications.

Managing Task Tracker users in a simple and secure manner, Firebase authentication supports a variety of authentication methods, including email addresses/passwords and third-party providers which including Google.

5.1.3 Hardware

Some Android phones, like Honor, were testing QR code scanning and mobile messaging.

5.2 Materials

YouTube

We used Youtube to learn Kotlin programming and how to implement Task Tracker features such as QR Code scanning, date picker implementation, user login status maintenance, etc.

Git bash

It is used to realize multi-person collaborative programming in project.

Github

This website can keep our code source.

CSDN and StackOverflow

They are websites for developers to discuss programming issues and share programming skills and tips. They can help us to fix bug.

Developer.android.google.cn

It provides the document for developers to learn about Android Studio settings.

Figma

Product prototyping tool, which is used for the UI design of the App in the project.

5.3 Roles & Responsibilities

Xufeng Zheng

Role: Developer,Tester

Responsibility: His responsibilities are to implement the page function of the Main Interface, as well as the Main Interface function test and page test.

Specific implementation includes Login, Register, all user family members display function, individual family members information display function, add family members function, children's brothers and sisters page display function, children users click to view other brothers and sisters display their tasks to achieve the function and so on.

Lanxin Zheng

Role: Developer, Group leader

Responsibility: Her responsibilities include UI beautification of the Main Interface, implementation of some main page functions, coordination of meeting time between team and client, etc.

Specific implementation includes the beautification of the Main interface's Home Page, QR code display page , user information display page, new family members creating page, children's siblings task showing page and target progress display page, user information change page function realization and page beautification, and so on.

Zhengyu Jiang

Role:UI designer, Developer

Responsibility: His responsibilities are Person Interface---Children pages' functions implementation, UI design, and UI beautification.

Specific implementation includes the function realization of children's personal information page, UI design, and UI beautification, as well as the function realization of children's task list display and goal list display in the personal interface, UI design and UI beautification, etc. Perform functional and interface tests on the children’s interfaces.

Linda Zeng

Role: UI designer, Developer

Responsibility: His responsibilities are Person Interface---Parent and Creator pages’ functions implementation, UI design, and UI beautification.

Specific implementation includes the function realization of creator and parent's information page, UI design and UI beautification, as well as the function realization, UI design and UI beautification of creators and parents' information showing and editing pages in the person interface, etc. Perform functional and interface tests on the parents' and creators’ interfaces.

Yupeng Liu

Role: Developer, Tester

Responsibility: His responsibilities are Setting Interface---Task pages’ functions implementation, UI design, and beautification and tests.

Specific implementation includes the functions implementation, UI design, and beautification of Parents and Creator can add task interface, task display interface, modify the interface, function realization of task display interface, and the function realization of children's target progress area in the interface design of the home page of the setting page. Perform functional and interface tests on the Task interface.

Dali Yuan

Role: Developer, Tester

Responsibility: His responsibilities are Setting Interface---Goal pages’ functions implementation, UI design, and UI beautification and tests.

Specific implementation includes the function realization of Parents and Creator can add the target interface, target display interface, target modification interface, and the target display interface, and the UI design and UI beautification of them, as well as the interface design of the main page of the setting page, to achieve the display function of the target progress area of children. Perform functional and interface tests on the Goal interface.

Ce Gao

Role: Developer, Tester

Responsibility: His responsibilities are Notification Interface---Goal pages' functions implementation, UI design, and UI beautification and tests.

Specific implementation includes Notification interface function realization, UI design, and UI beautification. When the child completes the task and goal, the system can record the behavior and the notification interface can display the notification. Perform functional and interface tests on the Notification interface.

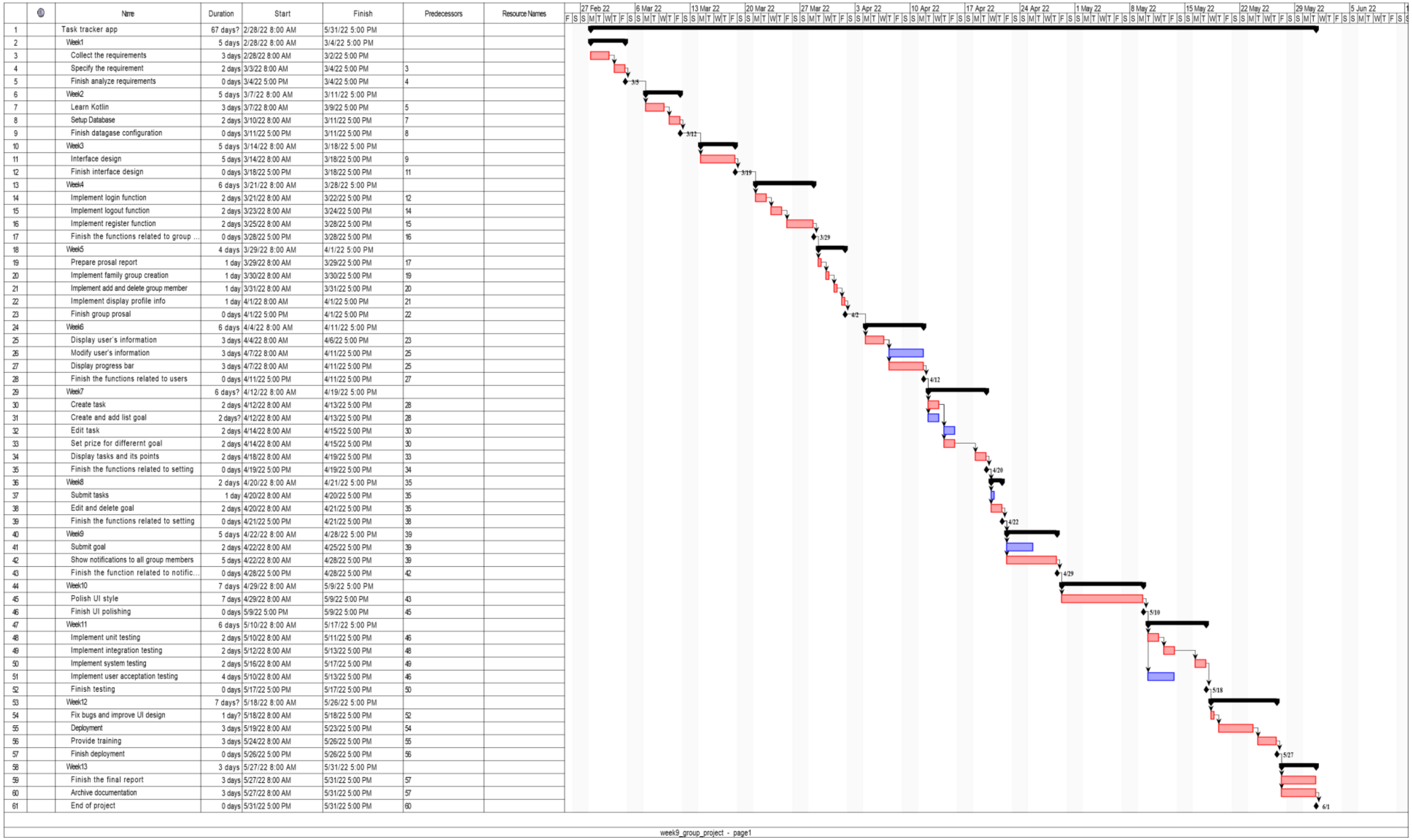
6. MILESTONES / SCHEDULE

In this project plan, we spend one week to analyse the requirements from client, that's what we did in week 1. From Week 2 to Week 10, we mainly focus on the stage of development. After that in Week 11 and Week 12, we focus on Testing and deployment. Finally in Week 13 we provide the training and documentation to client.

1 Milestone	Tasks	Reporting	Date
Week-1 Analyze requirements	Collect the requirements from client's specification	Client meeting to specify the requirement	26-2-2022
Week-2 Database configuration	Learn Kotlin Setup database	Client meeting to review the requirement	5-3-2022
Week-3 Interface design	Design style and appearance for different interface	None	12-3-2022
Week-4 Finish some basic functions	Login, logout, register	None	19-3-2022
Week-5 Finish group proposal Finish the functions related to group owner	Proposal Report Due The creation of family group Add and delete group member Display profile information	Client meeting to review the functionality implementation	26-3-2022

Week-6 Finish the functions related to users (parents, child)	Display user’s information (points, next task and goal, feedback for last task etc.) User can modify their information Display progress bar for the tasks	Client meeting to review the functionality implementation	2-4-2022
Week-7 Finish the functions related to setting part (Develop this part in advance)	Parents and group owner can edit and create tasks Parent can set prize for different goal. And children can view it The page should display some common tasks and corresponding points	Client meeting to review the functionality implementation	9-4-2022
Week-8 Finish the functions related to setting part	Children can submit the tasks Children can submit the Goals (Add functions in advance.)	Client meeting to review the functionality implementation	16-4-2022
Week-9 Finish the functions related to notification part Finish progress report	Show notification if points changed Show notification to all group members if one achieves the certain goal Progress Report Due	Client meeting to review the functionality implementation.	1-5-2022
Week-10 UI polishing Change the function due to client’s requirement	UI polishing Modify the function	None	8-5-2022
Week-11 Finish testing part Application deployment	Unit testing Integration testing System testing User acceptation testing	Client meeting to deploy the system	15-5-2022

	Deployment		
Week-12 UI polishing Debug Training provided to client	Keep polishing the UI interface Find bugs and fix bugs	Client meeting to provide training Demonstrate the whole application and all the functions	22-5-2022
Week-13 Documentation	Final Report Archive all the documents	Client meeting to submit all the documents	29-5-2022



	ⓘ	Name	Duration	Start	Finish	Predecessors	Resource Names
1		Task tracker app	67 days?	2/28/22 8:00 AM	5/31/22 5:00 PM		
2		Week1	5 days	2/28/22 8:00 AM	3/4/22 5:00 PM		
3		Collect the requirements	3 days	2/28/22 8:00 AM	3/2/22 5:00 PM		
4		Specify the requirement	2 days	3/3/22 8:00 AM	3/4/22 5:00 PM	3	
5		Finish analyze requirements	0 days	3/4/22 5:00 PM	3/4/22 5:00 PM	4	
6		Week2	5 days	3/7/22 8:00 AM	3/11/22 5:00 PM		
7		Learn Kotlin	3 days	3/7/22 8:00 AM	3/9/22 5:00 PM	5	
8		Setup Database	2 days	3/10/22 8:00 AM	3/11/22 5:00 PM	7	
9		Finish datagase configuration	0 days	3/11/22 5:00 PM	3/11/22 5:00 PM	8	
10		Week3	5 days	3/14/22 8:00 AM	3/18/22 5:00 PM		
11		Interface design	5 days	3/14/22 8:00 AM	3/18/22 5:00 PM	9	
12		Finish interface design	0 days	3/18/22 5:00 PM	3/18/22 5:00 PM	11	
13		Week4	6 days	3/21/22 8:00 AM	3/28/22 5:00 PM		
14		Implement login function	2 days	3/21/22 8:00 AM	3/22/22 5:00 PM	12	
15		Implement logout function	2 days	3/23/22 8:00 AM	3/24/22 5:00 PM	14	
16		Implement register function	2 days	3/25/22 8:00 AM	3/28/22 5:00 PM	15	
17		Finish the functions related to group ...	0 days	3/28/22 5:00 PM	3/28/22 5:00 PM	16	
18		Week5	4 days	3/29/22 8:00 AM	4/1/22 5:00 PM		
19		Prepare prosal report	1 day	3/29/22 8:00 AM	3/29/22 5:00 PM	17	
20		Implement family group creation	1 day	3/30/22 8:00 AM	3/30/22 5:00 PM	19	
21		Implement add and delete group member	1 day	3/31/22 8:00 AM	3/31/22 5:00 PM	20	
22		Implement display profile info	1 day	4/1/22 8:00 AM	4/1/22 5:00 PM	21	
23		Finish group prosal	0 days	4/1/22 5:00 PM	4/1/22 5:00 PM	22	
24		Week6	6 days	4/4/22 8:00 AM	4/11/22 5:00 PM		
25		Display user's information	3 days	4/4/22 8:00 AM	4/6/22 5:00 PM	23	
26		Modify user's information	3 days	4/7/22 8:00 AM	4/11/22 5:00 PM	25	
27		Display progress bar	3 days	4/7/22 8:00 AM	4/11/22 5:00 PM	25	
28		Finish the functions related to users	0 days	4/11/22 5:00 PM	4/11/22 5:00 PM	27	
29		Week7	6 days?	4/12/22 8:00 AM	4/19/22 5:00 PM		
30		Create task	2 days	4/12/22 8:00 AM	4/13/22 5:00 PM	28	
31		Create and add list goal	2 days?	4/12/22 8:00 AM	4/13/22 5:00 PM	28	
32		Edit task	2 days	4/14/22 8:00 AM	4/15/22 5:00 PM	30	
33		Set prize for differernt goal	2 days	4/14/22 8:00 AM	4/15/22 5:00 PM	30	
34		Display tasks and its points	2 days	4/18/22 8:00 AM	4/19/22 5:00 PM	33	
35		Finish the functions related to setting	0 days	4/19/22 5:00 PM	4/19/22 5:00 PM	34	
36		Week8	2 days	4/20/22 8:00 AM	4/21/22 5:00 PM	35	
37		Submit tasks	1 day	4/20/22 8:00 AM	4/20/22 5:00 PM	35	
38		Edit and delete goal	2 days	4/20/22 8:00 AM	4/21/22 5:00 PM	35	
39		Finish the functions related to setting	0 days	4/21/22 5:00 PM	4/21/22 5:00 PM	38	
40		Week9	5 days	4/22/22 8:00 AM	4/28/22 5:00 PM	39	
41		Submit goal	2 days	4/22/22 8:00 AM	4/25/22 5:00 PM	39	
42		Show notifications to all group members	5 days	4/22/22 8:00 AM	4/28/22 5:00 PM	39	
43		Finish the function related to notific...	0 days	4/28/22 5:00 PM	4/28/22 5:00 PM	42	
44		Week10	7 days	4/29/22 8:00 AM	5/9/22 5:00 PM		
45		Polish UI style	7 days	4/29/22 8:00 AM	5/9/22 5:00 PM	43	
46		Finish UI polishing	0 days	5/9/22 5:00 PM	5/9/22 5:00 PM	45	
47		Week11	6 days	5/10/22 8:00 AM	5/17/22 5:00 PM		
48		Implement unit testing	2 days	5/10/22 8:00 AM	5/11/22 5:00 PM	46	
49		Implement integration testing	2 days	5/12/22 8:00 AM	5/13/22 5:00 PM	48	
50		Implement system testing	2 days	5/16/22 8:00 AM	5/17/22 5:00 PM	49	
51		Implement user acceptation testing	4 days	5/10/22 8:00 AM	5/13/22 5:00 PM	46	
52		Finish testing	0 days	5/17/22 5:00 PM	5/17/22 5:00 PM	50	
53		Week12	7 days?	5/18/22 8:00 AM	5/26/22 5:00 PM		
54		Fix bugs and improve UI design	1 day?	5/18/22 8:00 AM	5/18/22 5:00 PM	52	
55		Deployment	3 days	5/19/22 8:00 AM	5/23/22 5:00 PM	54	
56		Provide training	3 days	5/24/22 8:00 AM	5/26/22 5:00 PM	55	
57		Finish deployment	0 days	5/26/22 5:00 PM	5/26/22 5:00 PM	56	
58		Week13	3 days	5/27/22 8:00 AM	5/31/22 5:00 PM		
59		Finish the final report	3 days	5/27/22 8:00 AM	5/31/22 5:00 PM	57	
60		Archive documentation	3 days	5/27/22 8:00 AM	5/31/22 5:00 PM	57	
61		End of project	0 days	5/31/22 5:00 PM	5/31/22 5:00 PM	60	

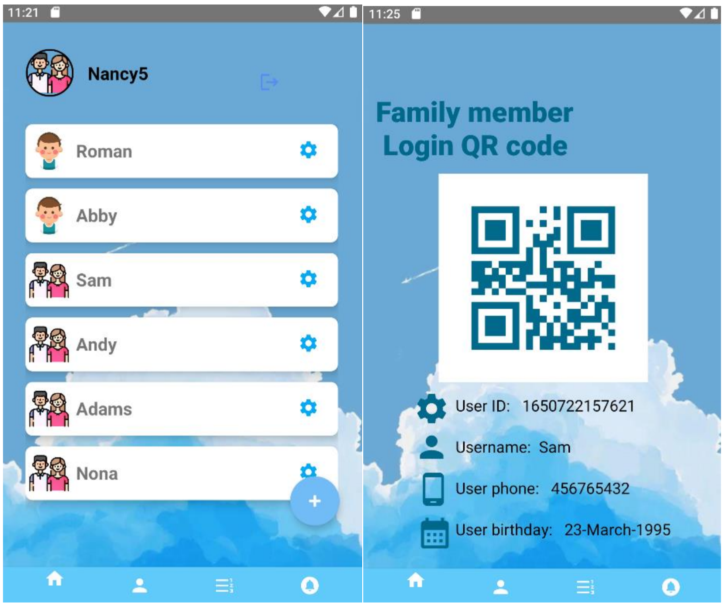
week9_group_project - page1

7. RESULTS

7.1 Main Interface(Included Login)

The Main Interface is designed to display all family members and information about family members.

For the Creator Account, Creator Account should be the top manager in a family. There is only one Creator in a family. Parent and children accounts can be created and QR codes can be shared so that parents and children can log in to the APP by scanning. As the top manager, the account owner can view and change the information of any family member (excluding his personal information) on the main page.



User ID: 1650722157621

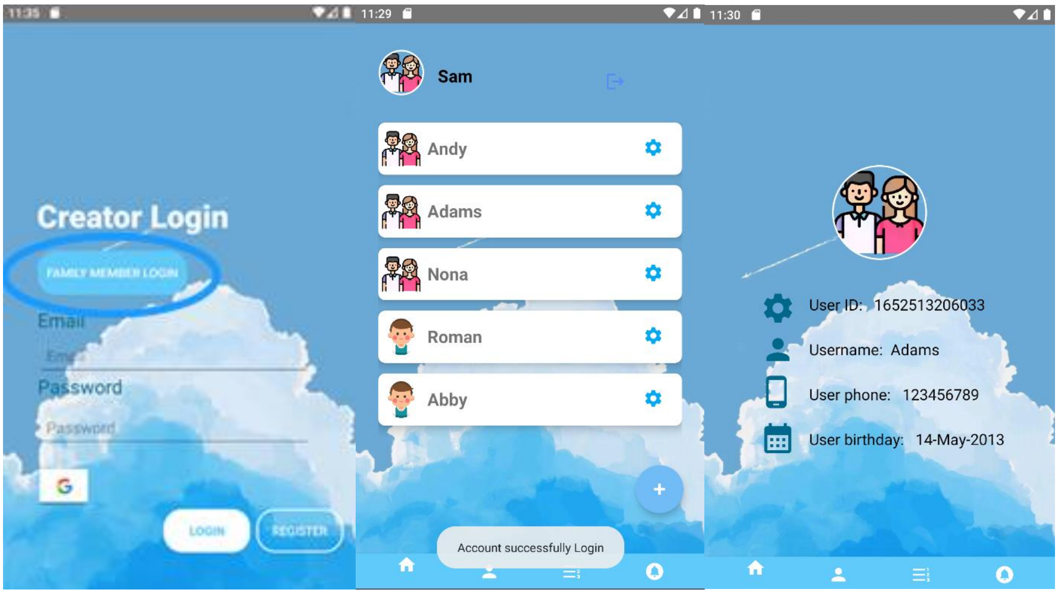
Username: Sam

User phone: 456765432

User birthday: 23-March-1995

Picture 7.1.1 Main pages in the Creator-sides

For the Parent Account, There can be multiple Parent Accounts in a family. As the second administrator, account owners can create new accounts but cannot share QR codes to invite others to join the family. This function is mainly used to add children's accounts to the Parent Account (Our Client thinks some children cannot use the app due to age restrictions, so they do not need to scan QR codes to log in). Parent Account can view and change the information of any account except Creator on the Main page.



Sam

Andy

Adams

Nona

Roman

Abby

Account successfully Login

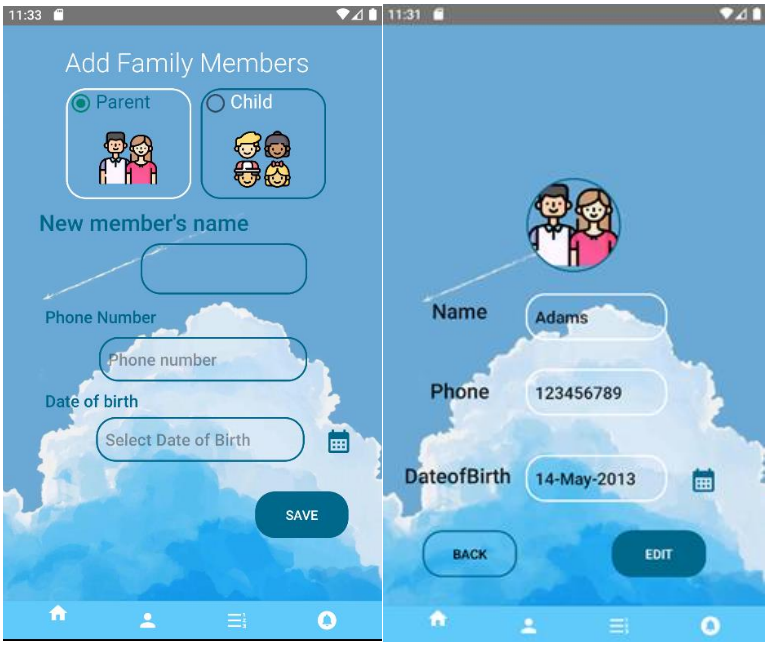
User ID: 1652513206033

Username: Adams

User phone: 123456789

User birthday: 14-May-2013

Picture 7.1.2 Main pages in the Parent-side



● Parent ○ Child

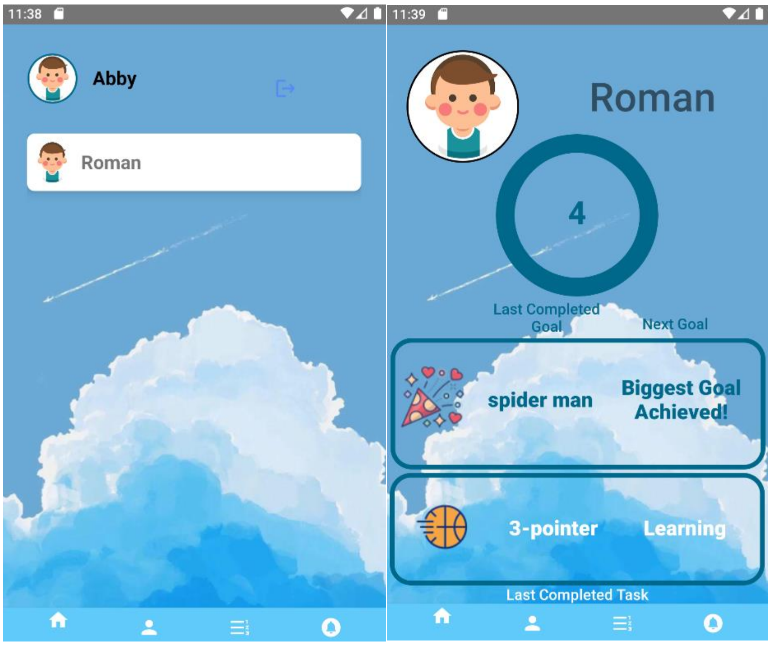
Name: Adams

Phone: 123456789

DateofBirth: 14-May-2013

Picture 7.1.3 Main pages in both Creator-side and Parent-side

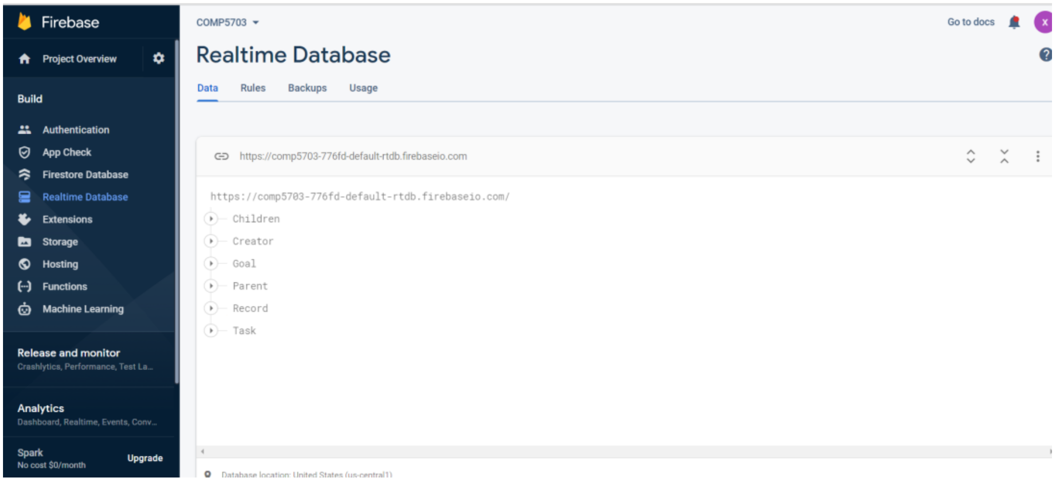
For the Children account, the Children Account is the Account with the lowest permission in the APP. The Account owner can only check the completion of tasks and goals of the Children's Account of the same type on the main page. Our client thinks that restricting children's behavior and not permitting them to modify the page will fundamentally reduce the chance of incorrect typing from them.



Picture 7.1.4 Main pages in the Children-side

About the database and test of main page

Firstly, in order to store information into firebase, we need to create a firebase base project and connect real time database with our project.



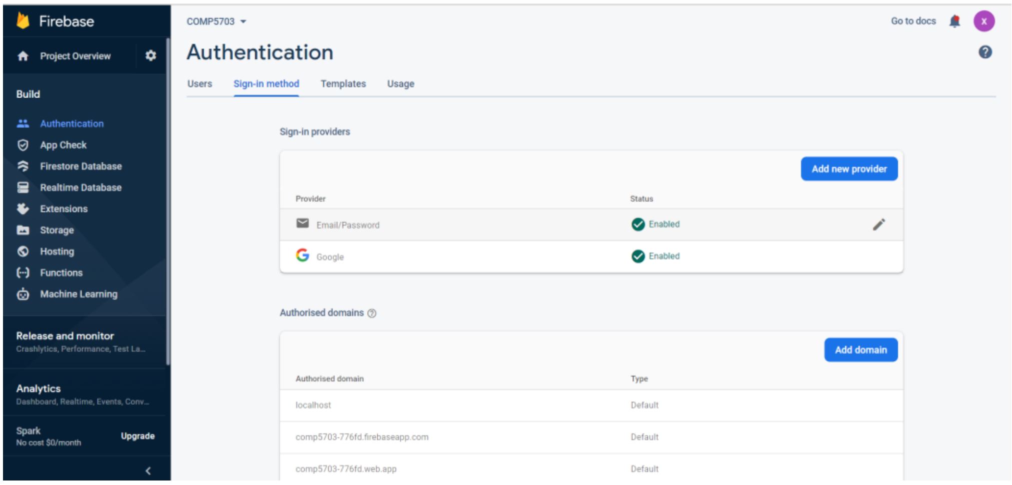
https://comp5703-776fd-default-rtdb.firebaseio.com

```
https://comp5703-776fd-default-rtdb.firebaseio.com/
  Children
  Creator
  Goal
  Parent
  Record
  Task
```

Database location: United States (us-central1)

Picture 7.1.5 real time database in firebase

After that we need to enable Authentication method (password/email and Google sign in), then we can use Firebase Authentication which is a requirement from client.



Authentication

Sign-in providers

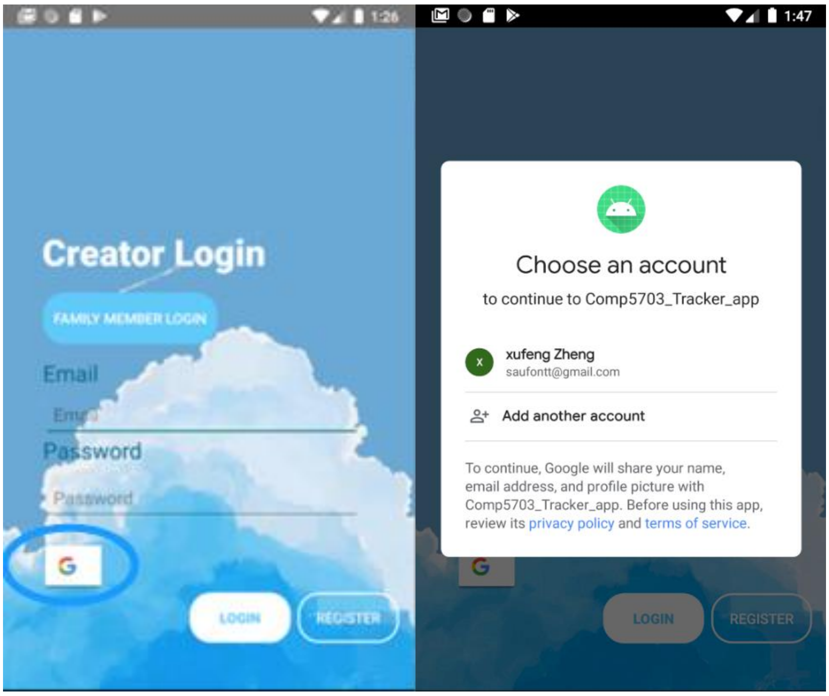
Provider	Status
Email/Password	Enabled
Google	Enabled

Authorised domains

Authorised domain	Type
localhost	Default
comp5703-776fd.firebaseapp.com	Default
comp5703-776fd.web.app	Default

Picture 7.1.6 Authentication service from Firebase

In the login page, user (Creator) can log in with their email and password, and user can also login with their Google account. Parent/Child can scan the QR code to login by clicking the family member login button.



Choose an account

to continue to Comp5703_Tracker_app

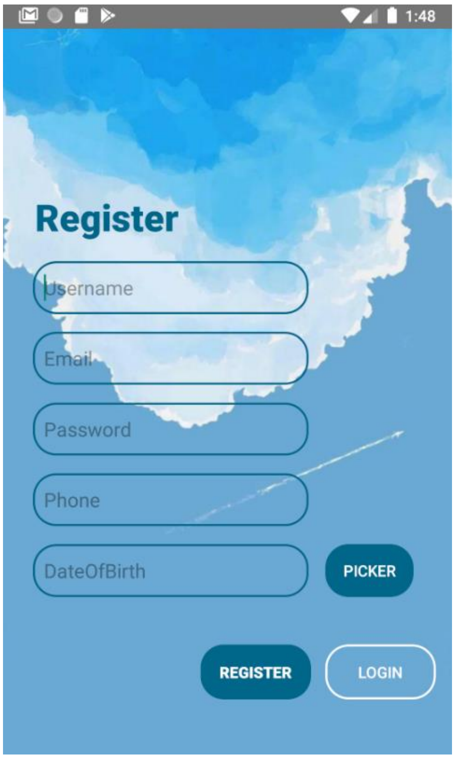
xufeng Zheng
saufontt@gmail.com

Add another account

To continue, Google will share your name, email address, and profile picture with Comp5703_Tracker_app. Before using this app, review its privacy policy and terms of service.

Picture 7.1.7 Google-Login in the app

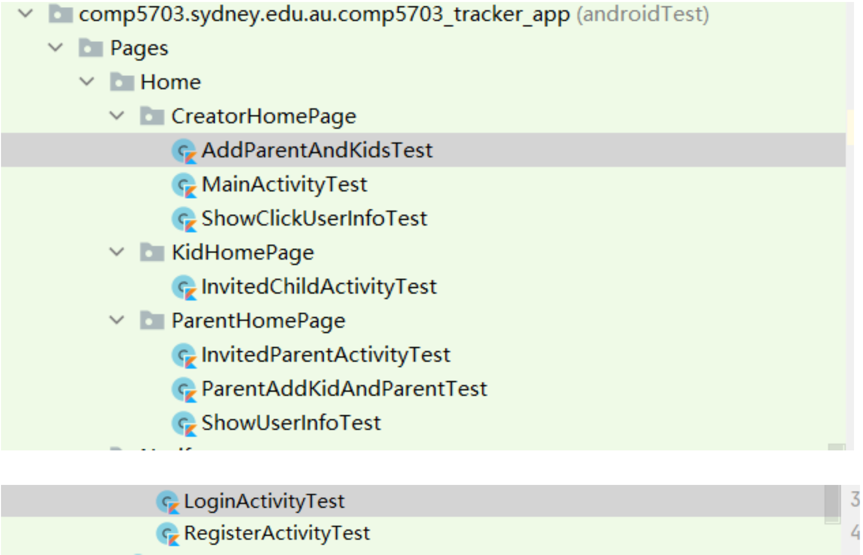
When user click register button, they can register an account by inputting their information.



Picture 7.1.8 Register page

For the testing part, we test all the pages under main page by using Espresso testing to ensure that the UI displays the correct widgets. Meanwhile the widgets must display the correct data storing in firebase. So, we can make sure that there are no bugs in our application when we handover the project to client.

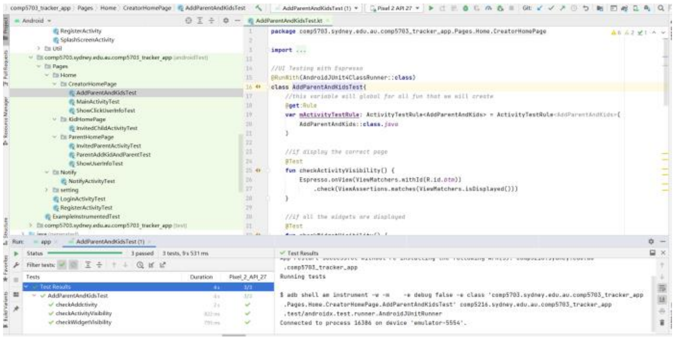
(All the UI testing under main page/home page and log in page)



```
comp5703.sydney.edu.au.comp5703_tracker_app (androidTest)
  Pages
    Home
      CreatorHomePage
        AddParentAndKidsTest
        MainActivityTest
        ShowClickUserInfoTest
      KidHomePage
        InvitedChildActivityTest
      ParentHomePage
        InvitedParentActivityTest
        ParentAddKidAndParentTest
        ShowUserInfoTest

        LoginActivityTest
        RegisterActivityTest
```

Picture 7.1.9 Test-list in Main pages

Some testing result under main page/login page.

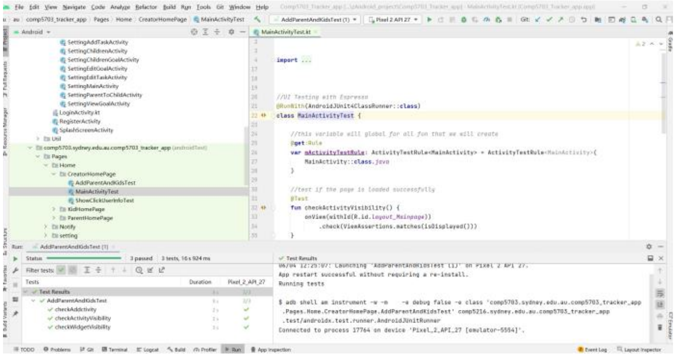


```
package comp5703.sydney.edu.au.comp5703_tracker_app.Pages.Home.CreatorHomePage

import ...

//UI Testing with Espresso
@RunWith(AndroidJUnit4ClassRunner::class)
class AddParentAndKidsTest{
    //this variable will global for all fun that we will create
    @get:Rule
    var mActivityTestRule: ActivityTestRule<AddParentAndKids> = ActivityTestRule<AddParentAndKids>(
        AddParentAndKids::class.java
    )

    //if display the correct page
    @Test
    fun checkActivityVisibility() {
        Espresso.onView(ViewMatchers.withId(R.id.btn))
            .check(ViewAssertions.matches(ViewMatchers.isDisplayed()))
    }

    //if all the widgets are displayed
    @Test
```

```
Status  3 passed  3 tests, 9 s 531 ms
Test Results               4 s     3/3
  AddParentAndKidsTest     4 s     3/3
    checkAddctivity        2 s     ✓
    checkActivityVisibility 832 ms ✓
    checkWidgetVisibility  735 ms  ✓
```

```
 .comp5703_tracker_app
Running tests

$ adb shell am instrument -w -m    -e debug false -e class 'comp5703.sydney.edu.au.comp5703_tracker_app
 .Pages.Home.CreatorHomePage.AddParentAndKidsTest' comp5216.sydney.edu.au.comp5703_tracker_app
 .test/androidx.test.runner.AndroidJUnitRunner
Connected to process 16386 on device 'emulator-5554'.
```

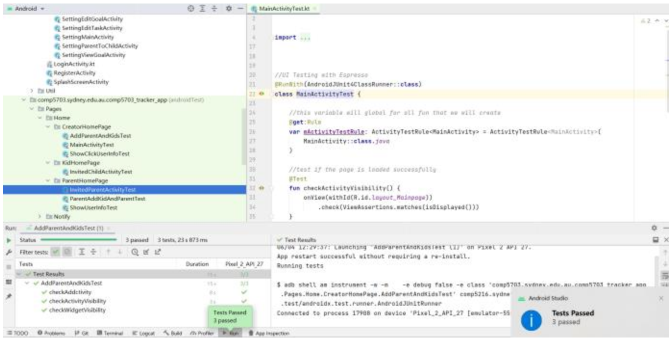
Picture 7.1.10 Test Result of Main pages (1)



```
import ...



//UI Testing with Espresso
@RunWith(AndroidJUnit4ClassRunner::class)
class MainActivityTest {

    //this variable will global for all fun that we will create
    @get:Rule
    var mActivityTestRule: ActivityTestRule<MainActivity> = ActivityTestRule<MainActivity>(
        MainActivity::class.java
    )

    //test if the page is loaded successfully
    @Test
    fun checkActivityVisibility() {
        onView(withId(R.id.layout_Mainpage))
            .check(ViewAssertions.matches(isDisplayed()))
    }
```

```
06/04 12:25:07: Launching 'AddParentAndKidsTest (1)' on Pixel 2 API 27.
App restart successful without requiring a re-install.
Running tests

$ adb shell am instrument -w -m    -e debug false -e class 'comp5703.sydney.edu.au.comp5703_tracker_app
 .Pages.Home.CreatorHomePage.AddParentAndKidsTest' comp5216.sydney.edu.au.comp5703_tracker_app
 .test/androidx.test.runner.AndroidJUnitRunner
Connected to process 17764 on device 'Pixel_2_API_27 [emulator-5554]'.
```

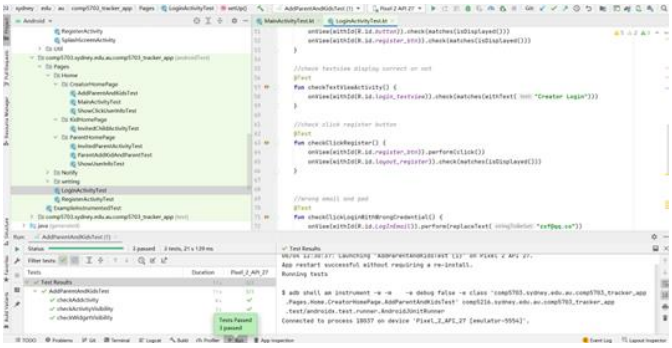
Picture 7.1.11 Test Result of Main pages (2)



```
import ...



//UI Testing with Espresso
@RunWith(AndroidJUnit4ClassRunner::class)
class MainActivityTest {

    //this variable will global for all fun that we will create
    @get:Rule
    var mActivityTestRule: ActivityTestRule<MainActivity> = ActivityTestRule<MainActivity>(
        MainActivity::class.java
    )

    //test if the page is loaded successfully
    @Test
    fun checkActivityVisibility() {
        onView(withId(R.id.layout_Mainpage))
            .check(ViewAssertions.matches(isDisplayed()))
    }
```

```
06/04 12:29:37: Launching 'AddParentAndKidsTest (1)' on Pixel 2 API 27.
App restart successful without requiring a re-install.
Running tests

$ adb shell am instrument -w -m    -e debug false -e class 'comp5703.sydney.edu.au.comp5703_tracker_app
 .Pages.Home.CreatorHomePage.AddParentAndKidsTest' comp5216.sydne
 .test/androidx.test.runner.AndroidJUnitRunner
Connected to process 17988 on device 'Pixel_2_API_27 [emulator-55
```

Picture 7.1.12 Test Result of Main pages (3)



```
        onView(withId(R.id.button)).check(matches(isDisplayed()))
        onView(withId(R.id.register_btn)).check(matches(isDisplayed()))
    }

    //check textview display correct or not
    @Test
    fun checkTextViewActivity() {
        onView(withId(R.id.login_textview)).check(matches(withText( text: "Creator Login")))
    }

    //check click register button
    @Test
    fun checkClickRegister() {
        onView(withId(R.id.register_btn)).perform(click())
        onView(withId(R.id.layout_register)).check(matches(isDisplayed()))
    }


    //wrong email and pwd
    @Test
    fun checkClickLoginWithWrongCredential() {
        onView(withId(R.id.LogInEmail)).perform(replaceText( stringToBeSet: "csf@qq.co"))
```

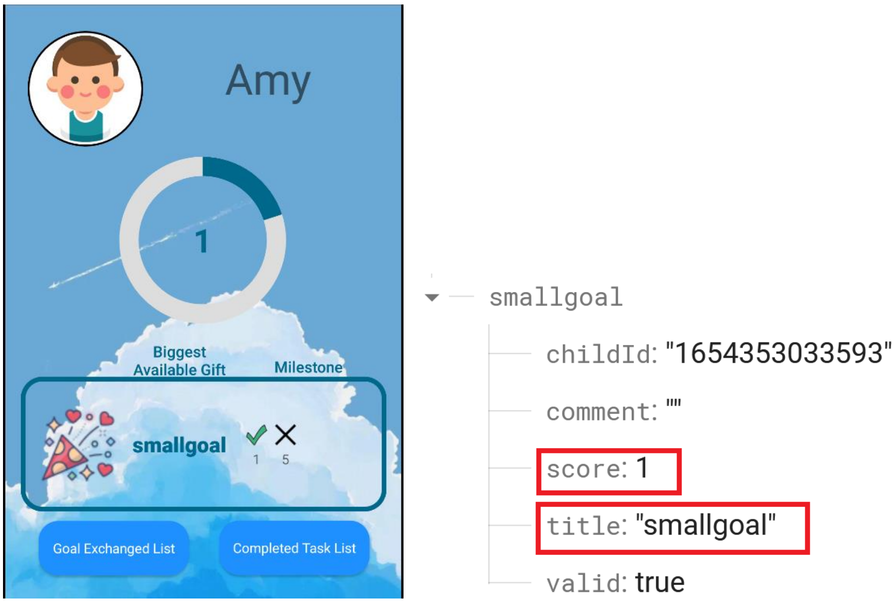
```
06/04 12:30:37: Launching 'AddParentAndKidsTest (1)' on Pixel 2 API 27.
App restart successful without requiring a re-install.
Running tests

$ adb shell am instrument -w -m    -e debug false -e class 'comp5703.sydney.edu.au.comp5703_tracker_app
 .Pages.Home.CreatorHomePage.AddParentAndKidsTest' comp5216.sydney.edu.au.comp5703_tracker_app
 .test/androidx.test.runner.AndroidJUnitRunner
Connected to process 18037 on device 'Pixel_2_API_27 [emulator-5554]'.
```

Picture 7.1.13 Test Result of Main pages (4)

7.2 Personal Page

Child:

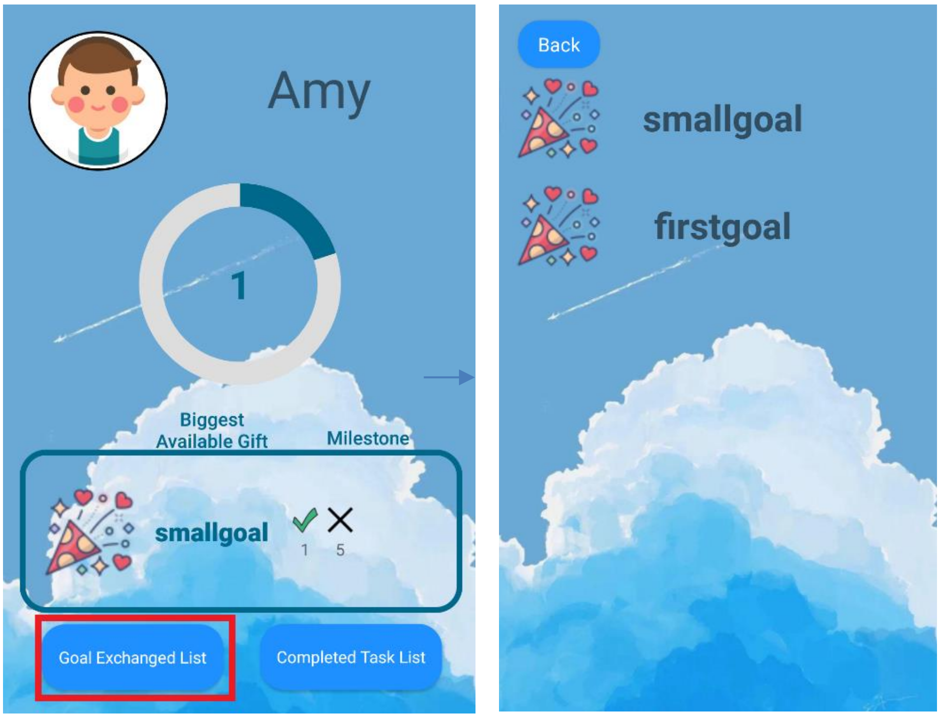
A child is able to see his information in his personal page. For example, his name, his score, his circular progress bar (his current score / his biggest target milstone), his biggest available gift and his milestone status. Specifically, as shown in the picture, the name of the child is Amy. His current score is 1 point. His biggest target milestone is 5 points (as indicated in milestone part). His circular progress bar shows 1/5 progress. His biggest available gift is called “smallgoal” since he has completed the milestone of 1 point. The title “smallgoal” and the score of 1 point are stored at the same level of a node in the database. In milestone status part, apparently, he has completed the milestone of 1 point and has not completed the milestone of 5 points.



```
smallgoal
    childId: "1654353033593"
    comment: ""
    score: 1
    title: "smallgoal"
    valid: true
```

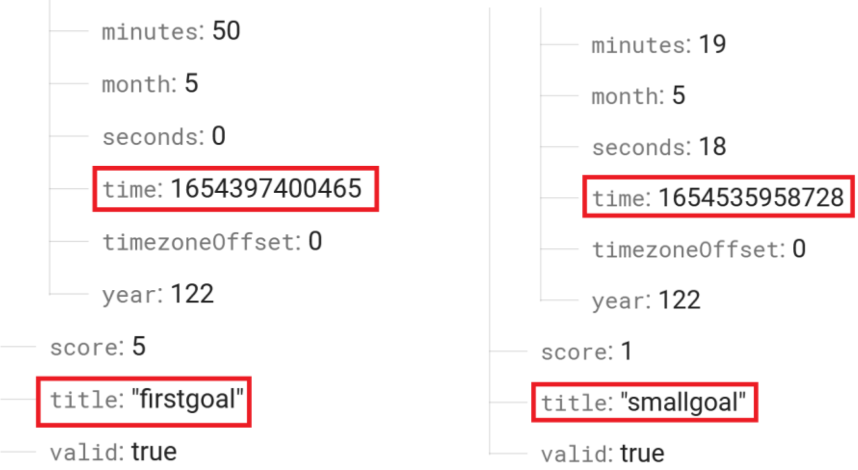
Picture 7.2.1 Children’s Task and goal data from database

A child is able to navigate to his goal exchanged list from his personal page.



Picture 7.2.2 Children’s goal-lists showing

As shown above, the child has already exchanged two goals. The goals are descendingly ordered by the date that his parent submitted for him, from top to bottom.

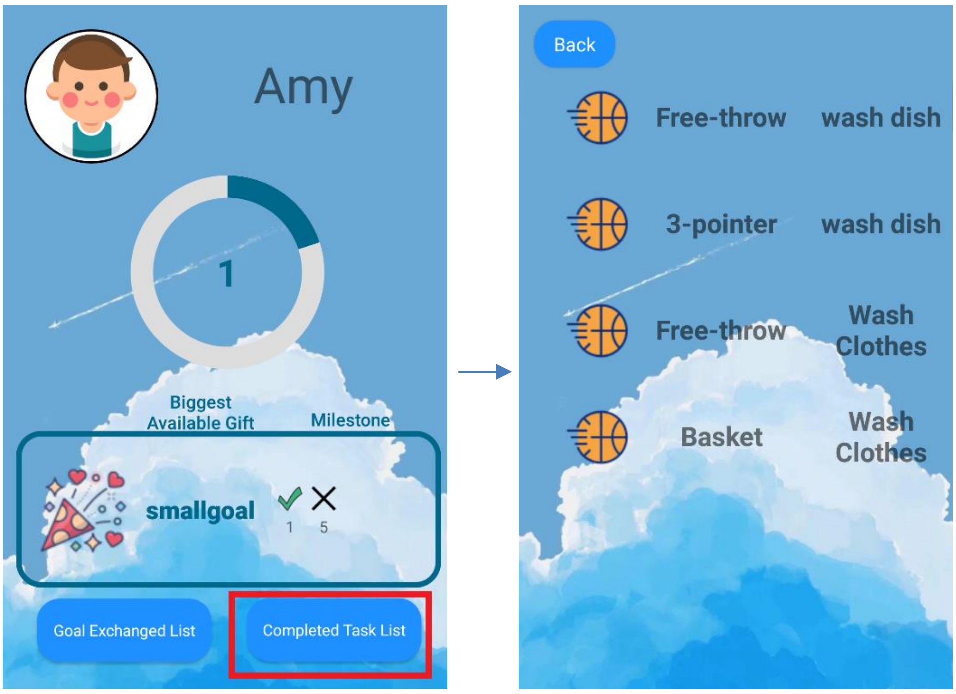


```
    minutes: 50
    month: 5
    seconds: 0
    time: 1654397400465
    timezoneOffset: 0
    year: 122
score: 5
title: "firstgoal"
valid: true
```

```
    minutes: 19
    month: 5
    seconds: 18
    time: 1654535958728
    timezoneOffset: 0
    year: 122
score: 1
title: "smallgoal"
valid: true
```

Picture 7.2.3 Children’s goal-lists’ related data

A child is able to navigate to his completed task list from his personal page.



Picture 7.2.4 Children’s recent task-lists showing

As shown above, the child has already completed 4 tasks. The tasks are also descendingly ordered by the date that his parent submitted for him, from top to bottom. Free-throw means the child gets 1 point for the task. Basket means the child gets 2 points for the task. 3-pointer means the child gets 3 points for the task.

For the milestone display part, if the space can not contain all milestones directly, the child is able to scroll the space horizontally to see all the milestone status. For goal exchanged list and completed task list, if the space can not contain all information directly, the child is able to scroll the space vertically to see all the information.

Parent and Creator:

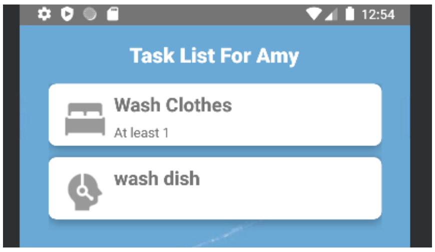
When a Parent user logs in to the personal information interface, the information page will search and read the current user information from the current firebase database. Since the ordinary parent account is created using the only specified QR code, the status bar of the mailbox will be fixed during the information display process, and the basic confidence display interface will display the name, mobile phone number, and birth date of the parent user, Then users can modify it through the edit function

As for the Creator account, the account is the creator of the entire family user. It has a unique designated mailbox and can be used to log in during the login process. As the mailbox information is a unique account ID, it cannot be modified in the information editing part. Like the Parent account, the name, email, mobile phone number and birth information will be displayed in the personal confidence display interface. If the user chooses not to fill in, The system will give the user a default basic information, at the same time, the creator user can also modify the basic information. At the same time, the user's password can also be modified using mailbox authentication according to the database management mechanism of the firebase.

7.3 Task and Goal Setting Page

Task & goal setting part is responsible for displaying, managing and submitting/exchanging tasks and goals for each child in the family. For parent and creator users, all the functions in this part are accessible. For children users, only the tasks and goals viewing function is given.

In order to access task part, creator/parent users need to click the third button in the bottom navigation bar. For child users, after the correspond button in the navigation bar has been clicked, they can enter the task page. All the assigned tasks will be displayed on the screen, which can be seen from Picture 7.3.1. For creator and parent users, the system will show a children list to let users choose the target child. After clicking the CardView of the child, the task list will be shown on the screen, which can be seen from Picture 7.3.2.

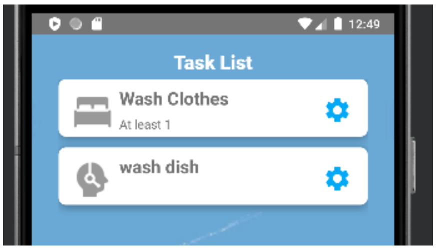


Task List For Amy

Wash Clothes
At least 1

wash dish

Picture 7.3.1 Children's View of Task List



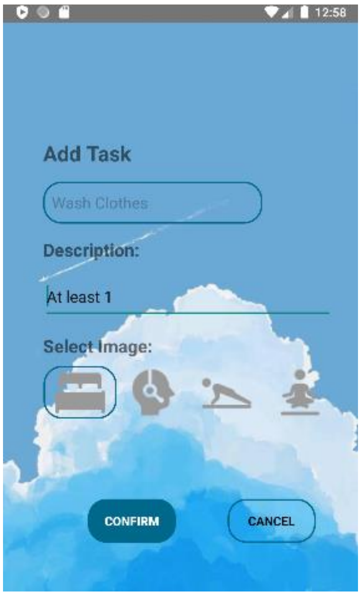
Task List

Wash Clothes
At least 1

wash dish

Picture7.3.2 Parents' View of Task List

As can be seen from Picture 7.3.2, each task has a setting icon. If the user clicks it, a page for editing tasks will be shown as Picture 7.3.3 shows. The title of the task cannot be edited, users may change the description and task icon part. If the user would like to remove a task, they need to long click the CardView of the task and the application will ask the user to confirm their behavior as Picture 7.3.4 shows. The button of "New Task" will lead users to the creating task page as Picture 7.3.5 shows, which allows users to assign a new task for children. For the submitting task part, once the CardView of each task is short clicked, a window will indicate users to select the score of the task, which can be seen from Picture 7.3.6.



Add Task

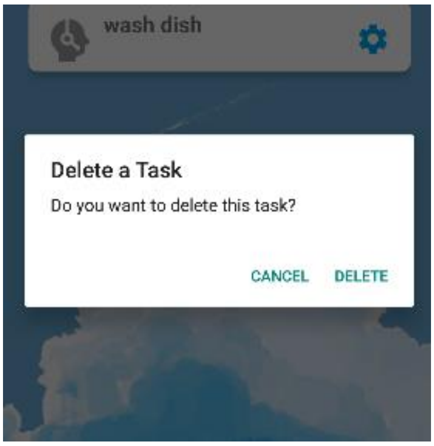
Wash Clothes

Description:

At least 1

Select Image:

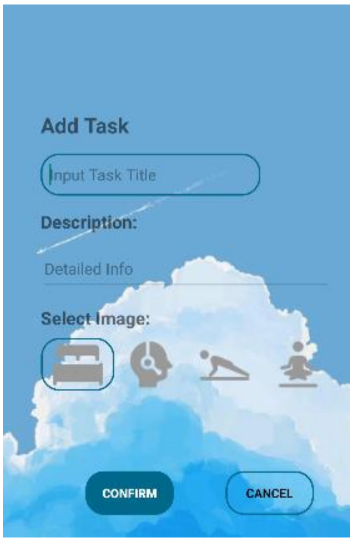
Picture 7.3.3 Editing Task Page



Delete a Task

Do you want to delete this task?

Picture 7.3.4 Confirm Deleting Window



Add Task

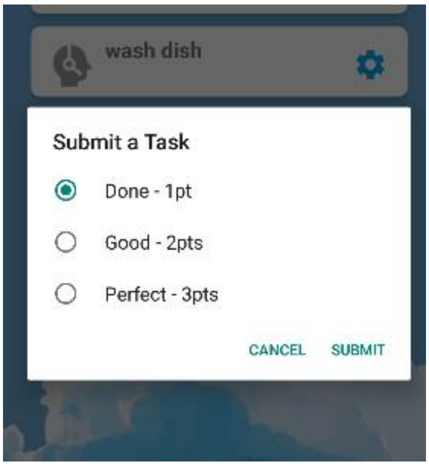
Input Task Title

Description:

Detailed Info

Select Image:

Picture 7.3.5 Add New Task Page



Submit a Task

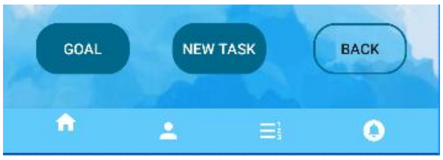
● Done - 1pt

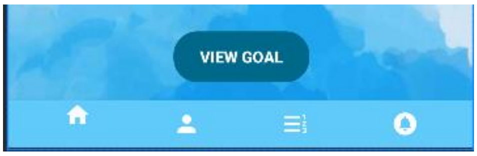
○ Good - 2pts

○ Perfect - 3pts

Picture 7.3.6 Assign Score while Submitting Task

For accessing goal part, creator&parent users need to click the ‘goal’ button(shown on Picture 7.3.7) in the task page to enter the goal page. And child users can continue click the ‘view goal’ button in their child task page to go to their goal page(shown on Picture 7.3.8).





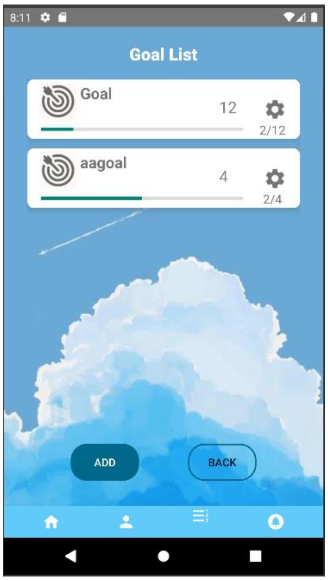
Picture 7.3.7 Goal page entry of creator&parent Picture7.3.8 Goal page entry of child

For creator&parent users, After entering the goal list page, they can see all the goals and its completing progress of the selected child user(shown on Picture 7.3.9). If they want to add a new goal for child, they can click the ‘add’ button on the list page and go to the add goal page(Picture 7.3.10). They can complete the adding of a new goal by clicking the 'confirm' button after entering the information meets the requirement (with input detection set up).

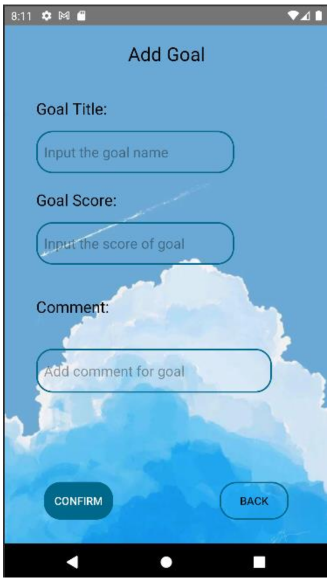
And if they want to change some details of an existing goal, they can click the ‘gear’ icon on the goal which they selected to go to the edit goal page(Picture 7.3.11). In the edit goal page, they also need to change the information and click the ‘confirm’ button with meeting the input requirement(checking).

They can also select and short-click a goal in the list to submit/exchange it with deducting required score of child by clicking the ‘submit’ button(Picture 7.3.12). And by long-clicking the goal, they can decide to delete the goal by clicking ‘delete’ button or not(Picture 7.3.13).

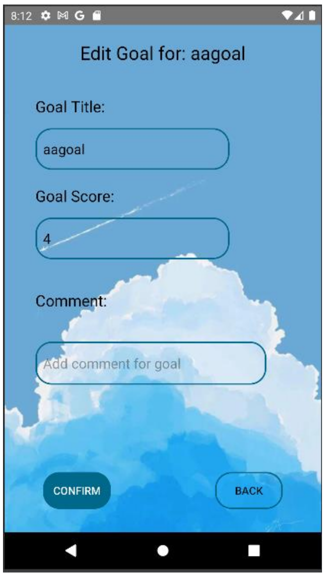
The Picture 7.3.14 shows the goal list page of child users. They can view all the information of their goals and decide what goal they want to exchange after obtaining the required scores.

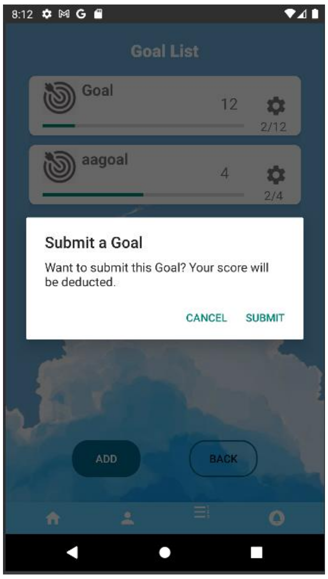


Picture 7.3.9 Goal list page



Picture 7.3.10 Add goal page



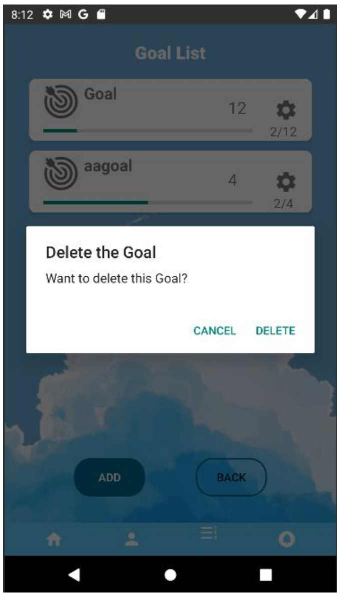


Submit a Goal

Want to submit this Goal? Your score will be deducted.

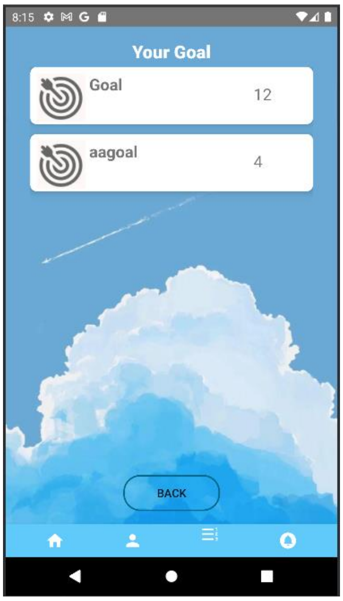
Picture 7.3.11 Edit goal page

Picture 7.3.12 Submitting goal



Delete the Goal

Want to delete this Goal?

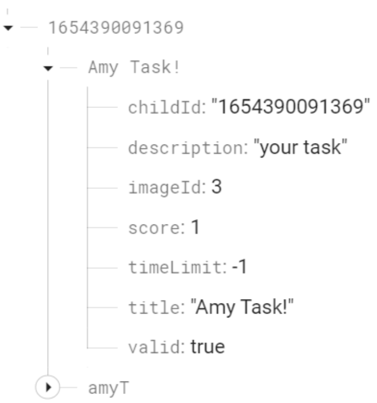


Picture 7.3.13 Deleting goal

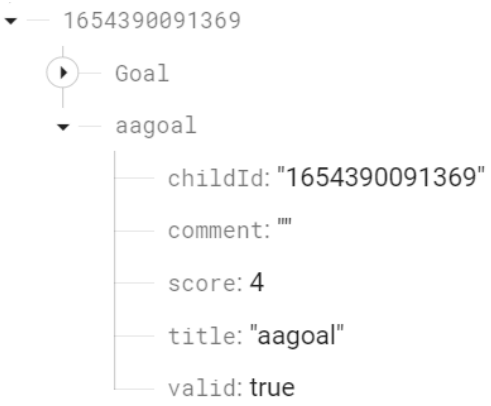
Picture 7.3.14 Goal list page of child users

All the data of tasks and goals will be stored into the database(shown in the Picture 7.3.15& Picture7.3.16). For each child, the tasks for them will be saved under "/Task/<uid>", the goals for them will be saved under "/Goal/<uid>". For each task, the data will be stored under "/Task/<uid>/<Task Title>", including the attributes of uid, task title, description, and icon id. For each goal, the data will be stored under "/Goal/<uid>/<Goal Title>", including the attributes of uid, goal title, goal score, completed time, and comment. Due to the task&goal title is a part of path in the database, the title cannot involve some special characters such as '[', ']', '#', '.', '$', and '/' because these characters will be refused by the Google Firebase Realtime

Database or damaged the structure of the database. An input check will be produced while the user try to add&edit a new task or goal.



```
1654390091369
  Amy Task!
    childId: "1654390091369"
    description: "your task"
    imageId: 3
    score: 1
    timeLimit: -1
    title: "Amy Task!"
    valid: true
  amyT
```



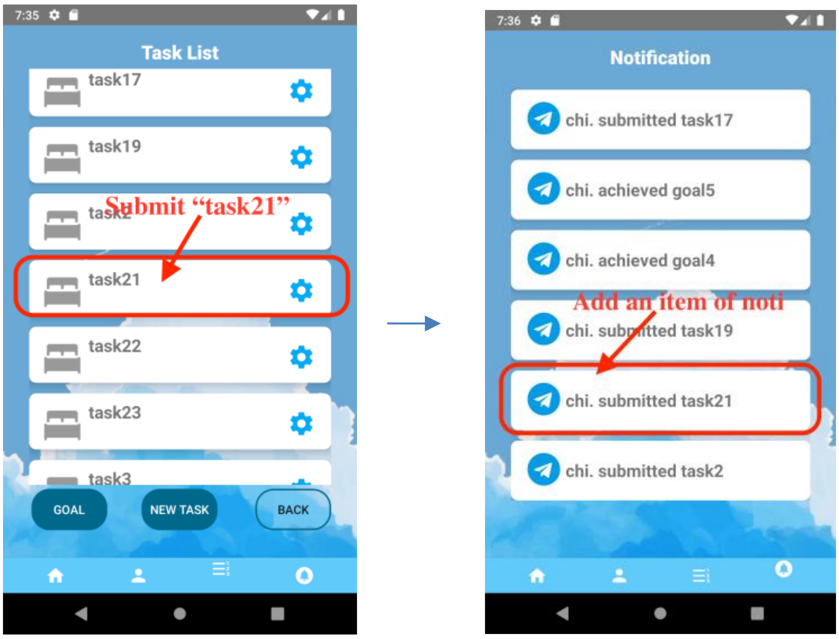
```
1654390091369
  Goal
  aagoal
    childId: "1654390091369"
    comment: ""
    score: 4
    title: "aagoal"
    valid: true
```

Picture 7.3.15 Example of Task data storing Picture7.3.16 Example of Goal data storing

7.4 Notification page

When a parent user submits a task/goal, a notification record is added to the database Record table, and the notification field includes record_from, record_name, and record_type. record_from represents a record belonging to a child, record_name is the name of the record, usually the task /goal name, and record_type is the type of the record, divided into task /goal.

After the parent user submits a task to a child on the Setting interface, a record is added to the list on the notification page.



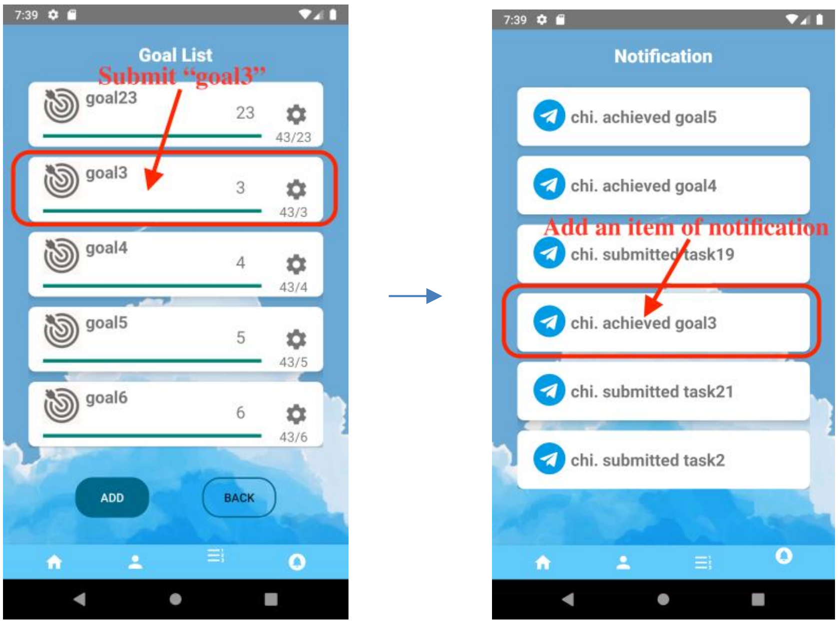
Task List

task17
task19
task2
task21
task22
task23
task3

Notification

chi. submitted task17
chi. achieved goal5
chi. achieved goal4
chi. submitted task19
chi. submitted task21
chi. submitted task2

Picture 7.4.1 Notification recording the submitting task behaviour



7:39

Goal List

Submit “goal3”

goal23 23 43/23

goal3 3 43/3

goal4 4 43/4

goal5 5 43/5

goal6 6 43/6

ADD BACK

7:39

Notification

chi. achieved goal5

chi. achieved goal4

Add an item of notification

chi. submitted task19

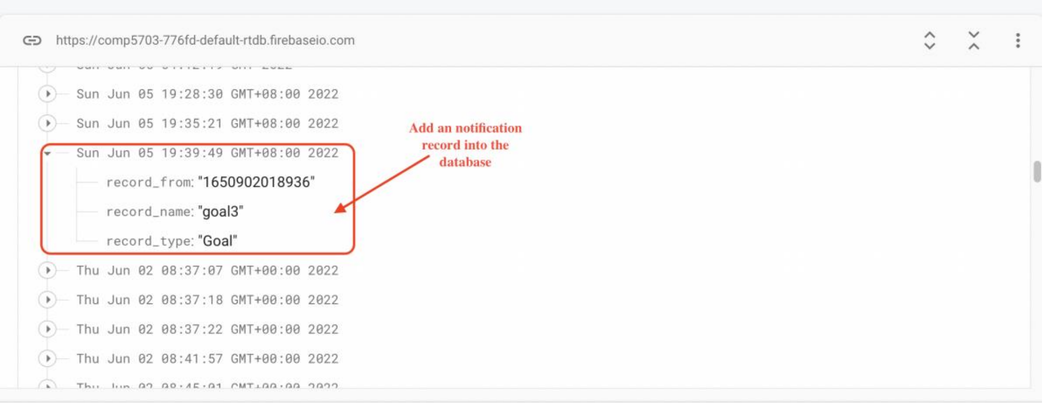
chi. achieved goal3

chi. submitted task21

chi. submitted task2

Picture 7.4.2 Notification recording the submitting goal behaviour

Meanwhile, a record is added to the “Record” table of the database.



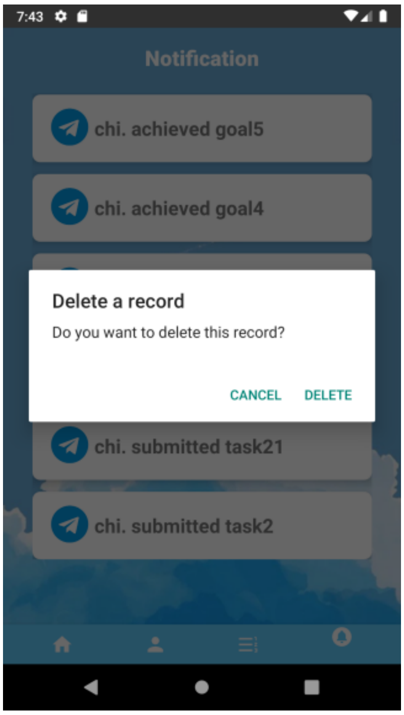
https://comp5703-776fd-default-rtdb.firebaseio.com

```
Sun Jun 05 19:28:30 GMT+08:00 2022
Sun Jun 05 19:35:21 GMT+08:00 2022
Sun Jun 05 19:39:49 GMT+08:00 2022
    record_from: "1650902018936"
    record_name: "goal3"
    record_type: "Goal"
Thu Jun 02 08:37:07 GMT+00:00 2022
Thu Jun 02 08:37:18 GMT+00:00 2022
Thu Jun 02 08:37:22 GMT+00:00 2022
Thu Jun 02 08:41:57 GMT+00:00 2022
```

Add an notification record into the database

Picture 7.4.3 Notification data recording in the database

Parent users have the right to delete a notification record. When the user long-presses a list item, a delete prompt box will pop up, and click OK to delete the record record in the database. At the same time, the item is also deleted from the list in the notification interface. Child users can only view the notification list and cannot delete list items.



chi. achieved goal5

chi. achieved goal4

Delete a record

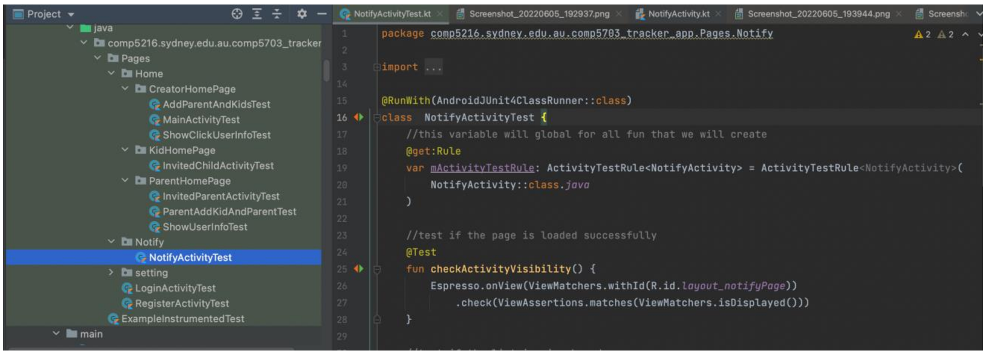
Do you want to delete this record?

chi. submitted task21

chi. submitted task2

Picture 7.4.4 Notification deleting

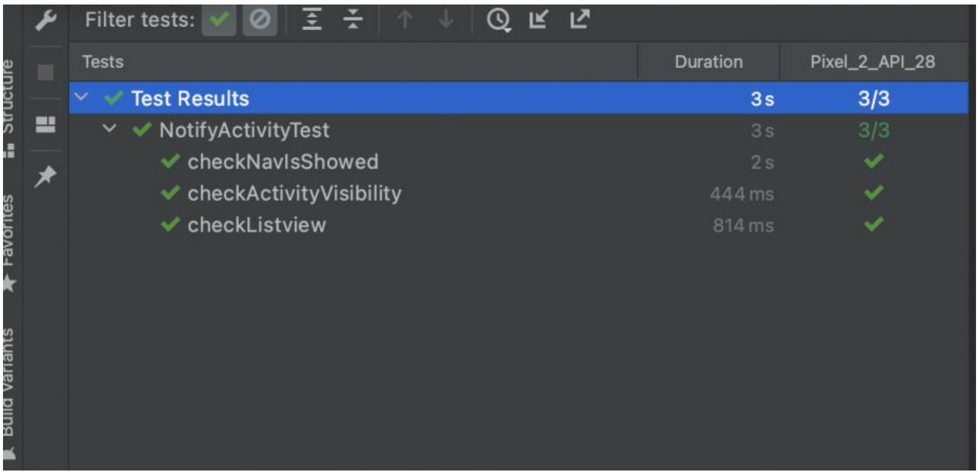
For the testing part, Espresso tests were used to test the notification page to make sure the UI displayed the correct widgets. At the same time, the widget must display the correct data stored in firebase.



```
package comp5216.sydney.edu.au.comp5703_tracker_app.Pages.Notify

import ...

@RunWith(AndroidJUnit4ClassRunner::class)
class  NotifyActivityTest {
    //this variable will global for all fun that we will create
    @get:Rule
    var mActivityTestRule: ActivityTestRule<NotifyActivity> = ActivityTestRule<NotifyActivity>(
        NotifyActivity::class.java
    )

    //test if the page is loaded successfully
    @Test
    fun checkActivityVisibility() {
        Espresso.onView(ViewMatchers.withId(R.id.layout_notifyPage))
            .check(ViewAssertions.matches(ViewMatchers.isDisplayed()))
    }
```

Picture 7.4.5 Notification page test

All online tests pass correctly:



Tests	Duration	Pixel_2_API_28
✓ Test Results	3 s	3/3
✓ NotifyActivityTest	3 s	3/3
✓ checkNavIsShowed	2 s	✓
✓ checkActivityVisibility	444 ms	✓
✓ checkListview	814 ms	✓

Picture 7.4.6 Notification page test result

8. DISCUSSION

8.1 For the results:

All in all, our project complete all the required functions according to the needs of the client:

In the login&register page of the project, we provide creator users with several different ways to register and log in to the account, which can well meet the usage habits of different users. After completing the creation of parent/child users in the main page, we also provide the parent and child users with an entry to scan the code to log in on the login page. This can minimize the using threshold for some younger users, they do not need to complete the more complicated registration and login operations by themselves, and also improve the usability of our application.

In the main page of the project, we clearly show each member of the family group. Creator&parent users can add or delete the account of the corresponding family member here, or click to view any family member, and get the login QR code of the corresponding account, which also avoids multiple repeated registrations that may occur among members of the same family. Child users can click here to view the situation of other children in the same family, which can help them understand the situation of other children and encourage them to encourage each other.

In the personal page of the project, all users can see their personal information. We gave the child user more information to display, including his score and some goal progress displays, to help him better understand his status. For creator&parent users who use different login methods, they can also improve or modify their personal information here. For the modification of the password, we also adopted the method of sending an email to confirm to strengthen the security.

In the setting page of the project, creator&parent users can easily select any child user to view or modify their task&goal. We also implemented user input detection in both parts to ensure the normal operation of the program, and each goal has a corresponding progress bar, which can help users better understand the completion progress of each child user. In addition, we do not need child users to complete task submission or goal exchange, but they can still see each of their tasks and goals on this interface, helping them communicate more with their parents and focus on completing what they want. on the target.

In the notification page of the project, child users can see the completion of each task or goal and receive reminders, while creator&parent users can see the completion of all child users in the family. This can help children better understand their accomplishments and gain a sense of accomplishment, and it is also convenient for parents to call on family members to discuss and celebrate some of the major goals their children have accomplished.

In order to verify the normal operation of the above page functions, we also wrote tests in different ways for different pages.

8.2 Implication and significance:

By using our project applications, it will have different impacts on our target customers and the existing application market.

For the customers:

For parents: Parents can keep abreast of their children's task progress through the notifications sent by our application and their children's information interface, celebrate the achievement of goals with their children, and increase their usual attention to their children. At the same time, parents and children set goals together, which also strengthens communication with children, and is conducive to improving parent-child relationship and creating a good family atmosphere. Parents can also supervise each other with their children to help them break some bad habits, which is also a kind of exercise for themselves.

For children: It can help children develop good habits, promote good communication between children and parents, and gradually cultivate children's labor awareness. At the same time, it also cultivates children's self-discipline and planning ability, allows children to actively participate in beneficial activities such as housework and sports, enhances children's physique, and cultivates children's interests in an all-round way.

For related application markets:

We found that there is no similar application occupying a monopoly position in each application market, so this project is also a supplement to the corresponding software market. And our application is designed to allow all family members to participate, and can also provide interactivity and related functions that other applications do not have, so our application also has a certain originality (there is no market for the same well-known app for the user base). Finally, the project also has the ability to further improve the function and expand the audience on the basis of the original user group.

9. Limitations and Future Works

Although our project has completed all the current needs of the client well, there is still room for improvement. Below is a list of our ideas for improvement and future solutions to some of the current limitations of the project:

1. The interface content is too monotonous, so more functions can be added on the Main page in the future. For example, the Main page of children users can display today's tasks. Parent can display the most recent child task completion.

2. Combine the task addition with the main page, because the main function of the app is for parents to assign tasks to their children, so the home page should have the main function.

3. Main page can realize the user switch function in the future, such as re - scan code login to switch users and so on.

4. If there are too many data item in firebase, then it will be very slow to fetch data from firebase. Maybe we could do some research on how to create an index in firebase to accelerate the data fetching in the future.

5. Sometimes if the laptop is not good enough, then the main thread will be overloaded due to the large tasks (fetching data from database, complex calculation, etc.) execution. So, in future, we should create more threads to handle these heavy tasks.

6. Plan the time schedule well before the project start, so you will not waste any time in the project.

7. Firebase data items are not stored in chronological order, so the order of data obtained by notification querying the database is unknown. According to user habits, notifications should be arranged in reverse chronological order, so the latest notification records are displayed at the top, but the current notification list items are

in accordance with The storage of the database is arranged in reverse order, so it needs to continue to be improved in the future to improve the user experience.

8. The notification page should provide more permissions to parent users to implement operations such as modification of notification list items.

9. In addition, the notification function of this machine will continue to be added in the future, so that users can also see the newly added notification items in the background.

10. Add more security-related information monitoring to protect users, such as limiting the validity of qr codes, authentication after login and so on.

11. Because of the limitation of Google Firebase Realtime Database and the design of database structure, the title of task will be limited, users cannot use some specific characters in the task title. Redesign the database structure may solve the limitation, for example, do not use the title as a part of path in the database.

In addition, in the continued development of future projects, we also need to continuously optimize the UI design to make it look more comfortable and user-friendly according to the usage habits and feedback of different users.

We will also add some new functions such as user birthday surprises, regular reminders of daily tasks, etc.

For the development method of the project, we can also develop more new functional modules to meet the different needs of more users. We should adopt the form of more detailed division of labor development for these requirements, and optimize the code structure to reduce the coupling degree of each part in the project development process. We will also optimize the way project and user data are accessed to reduce the impact of data growth on application response time.

Finally, we should also maintain the application on a regular basis to ensure the data security of the database and the normal operation of the program when the user is using the application.

REFERENCES

DingTalk. (2021). DingTalk, Make Work and Study Easy. Retrieved March 24, 2022, from https://www.dingtalk.com/en

DingTalk. (n.d.). DingTalk Lite helps high school build up an online education platform. Retrieved March 24, 2022, from https://page.dingtalk.com/wow/dingtalk/act/en-case3?spm=a213l2.14889004.8312382473.6.10eb2726vhCAsa&wh_biz=tm&acm=lb-zebra-676952-8441325.1003.4.7994477&scm=1003.4.lb-zebra-676952-8441325.OTHER_15875790261861_7994477

Google LLC. (2022). Google tasks - apps on Google Play. Retrieved March 24, 2022, from https://play.google.com/store/apps/details?id=com.google.android.apps.tasks

Microsoft. (2022). To do list and task management app: Microsoft to do. Retrieved March 24, 2022, from https://www.microsoft.com/en-us/microsoft-365/microsoft-to-do-list-app?rtc=1