woohoo

Documentation

By Qinghang Hong, Jiajun Li, Haochen Yang, Yiwei Zhang

图片包含 白板

描述已自动生成

**Part One**

**Design Process**

**1.Tools and Technologies**

Language: Java

Development Tool: Android Studio

Backend: Firebase

Problem Set:

* Name
* Number of problems
* Problems: Each question has its answers and correct answer index.

User:

* Email
* Friend List
* Name
* Icon
* Pending Friend List: Other users’ information, including name, id and icon.

Current Room Set:

* Room Id
* Name of Problem Set
* Quiz Index (current index of questions)
* Timer Per Question
* Host Id
* Player List: It includes each player’s user id and score.
* Start Flag (if the game start)

**2.Stages of Design**

Stage 1: The beginning

We decided to first use Kotlin for the following reasons:

1. It is a novel and concise language other than Java.
2. After using Kotlin, the app will become easier and cheaper to maintain.

Stage 2: After the First Meeting

We switched from Kotlin to Java for the following reasons:

1. There are more resources implemented by Java.
2. We have more experience in Java because of other CS courses.

We used Firebase as backend database for the following reasons:

1. There are more tutorials about firebase online.
2. It syncs the real-time data across all the devices in Android.
3. It is flexible for developers to store values.

We generated a general idea and divided it into several parts:

1. Create & Find Room
2. Quiz & Settlement Page
3. Login & Logout Page
4. Upload Question Page
5. Friend Page

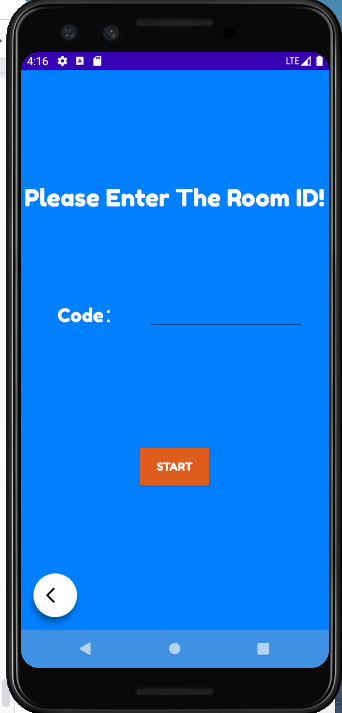
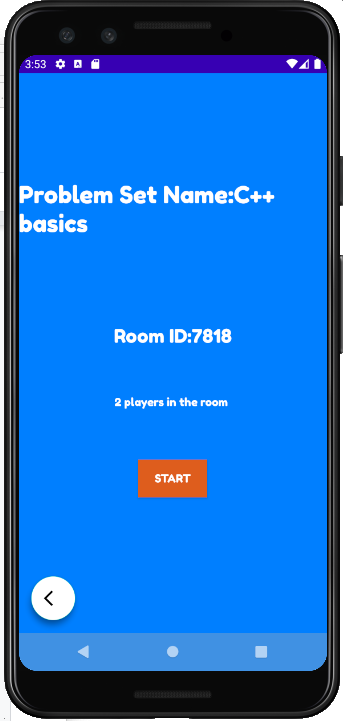
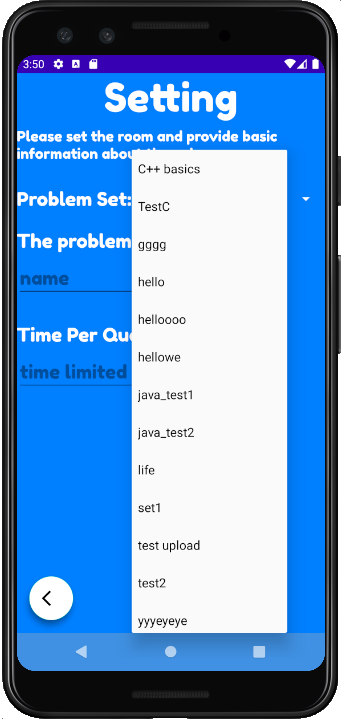
Stage 3: Detailed implementation

1. Create & Find Room

We created class structures which can store the user’s information, and quiz room’s information like room ID. Then we connected our application to firebase, so that we can store the information about users and quiz rooms in firebase. Through firebase, after users enter the same room ID, they can enter the same quiz room.

The host has the right to start the quiz. As the host clicks the start button, the application will send the “start” value to firebase. Then each member in the same room will check the “start” value in the firebase. After that, their quiz will also start.

Thus, it has the following functionalities: Host can create a room with unique 4-digit ID, then he/she goes to the main waiting room; Players can join in the room with this ID, then he/she goes to the student waiting room; Both waiting rooms show the current number of players after joining the room, but only main waiting room has the create button to start the quiz game.



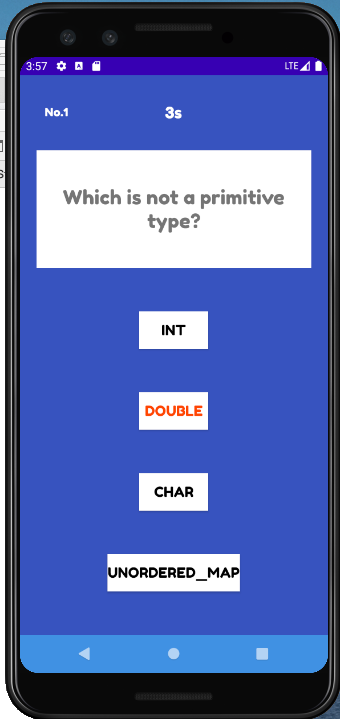
1. Quiz & Settlement Page

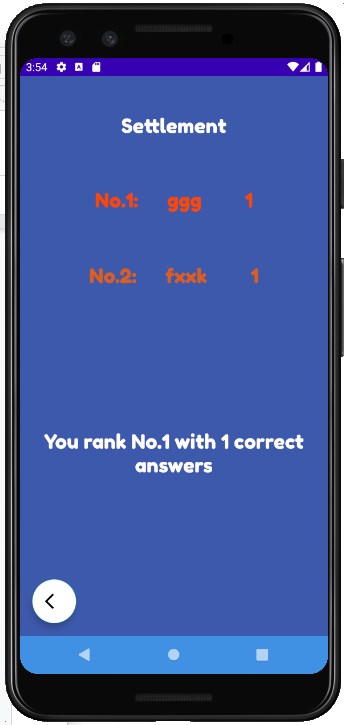
Before entering the Quiz Page, we create a game object that contains quiz descriptions, current index of quiz, time per question, and multiple answers. Then we download all this information from firebase and put it into the game object.

In the Quiz page, for every question, these information are extracted from game objects and shown on the screen. There is a timer built in the Quiz Page so that it will be directed to the Answer Page at the end of countdown.

In the Answer Page, for every question, quiz descriptions and correct answers are displayed. Also, we built a toast so that players can know whether their answer is right or not in the last page. Also, we built two transitions, from Answer Page to Quiz Page and from Answer Page to Settlement Page when the timer finishes. Then, we update the current index contained in the game object so that it can decide which transition it should go.

If we check the current question index is the same as the number of total questions in the Answer Page, then it will be directed to the Settlement Page. Then, we connected to the firebase and set a waiting toast until all the players had finished their questions. After that, it will read the data from the firebase and rank them based on their scores. Then, players can see the top 3 players and their ranking in the Settlement Page.

电子设备的屏幕

描述已自动生成

1. Login & Logout Page

We used Firebase Authentication to register users. We only allowed email-password registration.

When creating a user, we stored the user's basic information including nickname, UID, e-mail into the real-time database so that we can fetch user’s data easily.

By checking all existing users' nicknames, we avoid users registering using the same nickname. Users don’t have to enter their nickname when logging in.

屏幕的手机截图

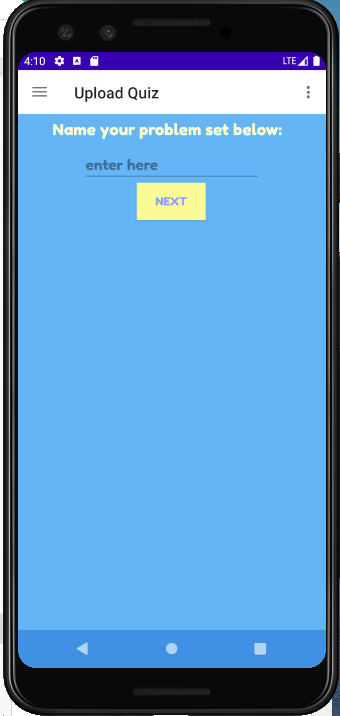
描述已自动生成图形用户界面, 应用程序, Teams

描述已自动生成

1. Upload Question Page

We prompt the user to enter the name of the problem set he/she wants. After entering each question, the “next” button appends the problem to the problem set in the view model and clears all fields to prompt the user to enter the next problem.

After the user clicks the “Finish” button, we will store the entire problem set in the firebase under the reference “ProblemSet”, with the name of the problem set as the key. The information includes the name, each problem with the question body, answer choices and correct answer index.

屏幕上有个手机

描述已自动生成

1. Friend Page

We created “friend” and “pending friends” references in the real-time database to fetch user’s friends and pending friends.

By adding a friend, the current user sends an invitation to the target user, which creates an item in the “pending friends” reference in the target user’s database. By this time, both users haven’t added each other yet. The target user has to approve the invitation in order to let both users successfully add each other. By checking the “pending friends” reference, the target user changed the needed fields in the database so that both users successfully add each other.

Users can delete friends. However, only current users can see the deletion. The current user deletes the target user in the current user’s “friend” reference. It will not delete the current user in the target user’s “friend” reference. The target user still has this user as a friend.

图形用户界面, 文本, 应用程序, 聊天或短信

描述已自动生成

Stage 4: UI Design

1. We have the following pages:

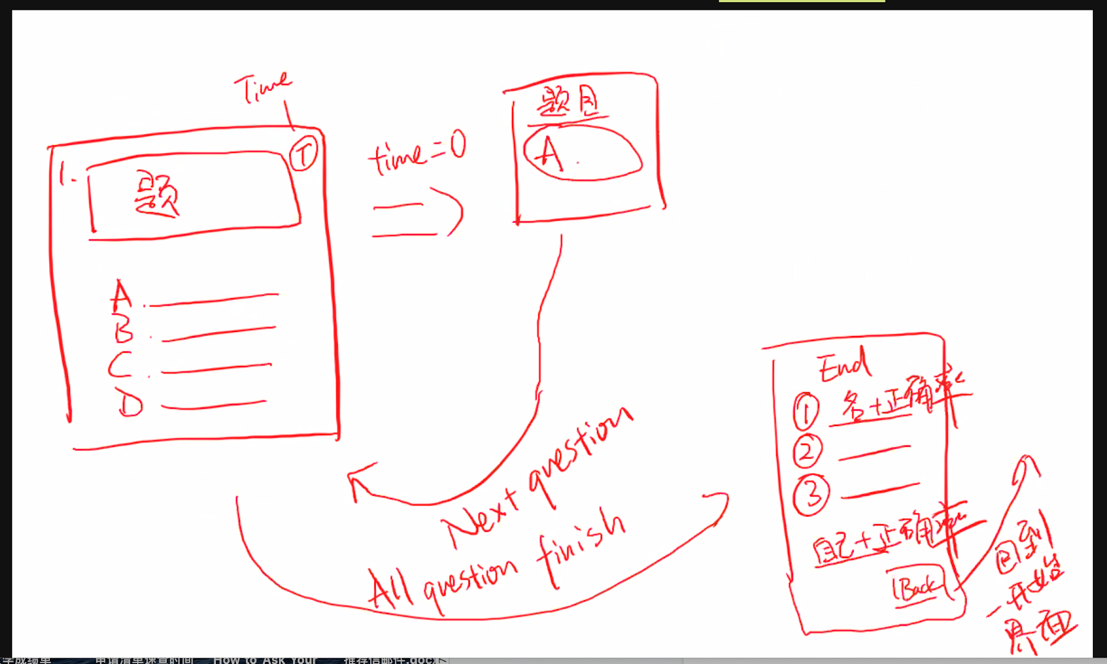
* Navigation Drawer Activity Template Page
* Login and Sign-Up Page
* Friend Page
* Create Room and Find Room Page
* Quiz Page
* Settlement Page
* Upload question Page

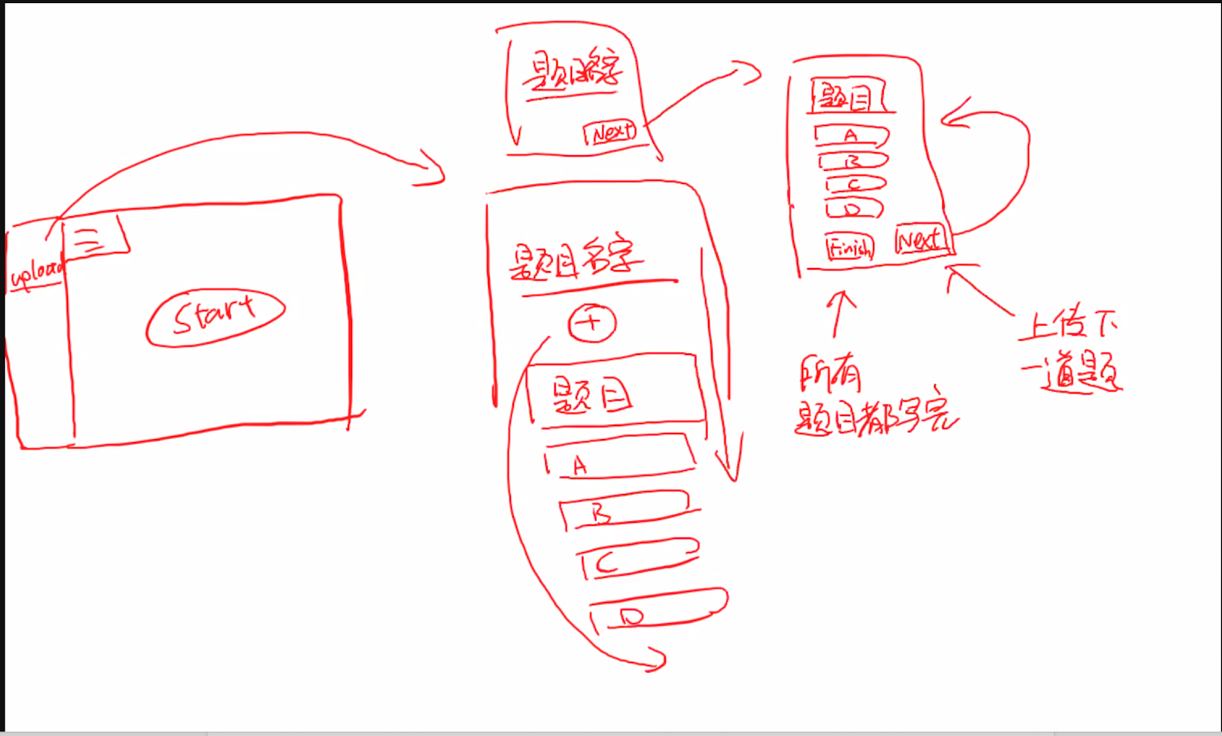
1. Scratch

At the beginning, we drew the UI picture on iPad.

文本, 白板

描述已自动生成

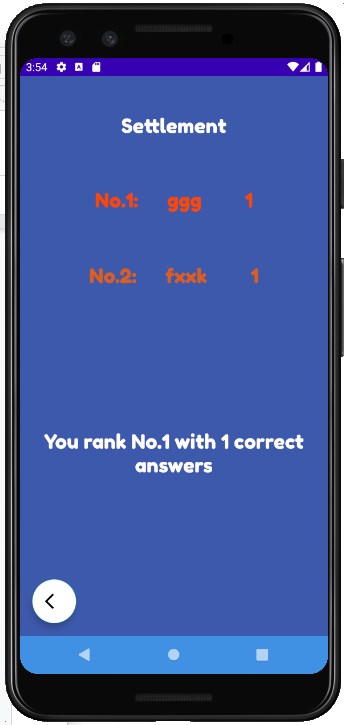
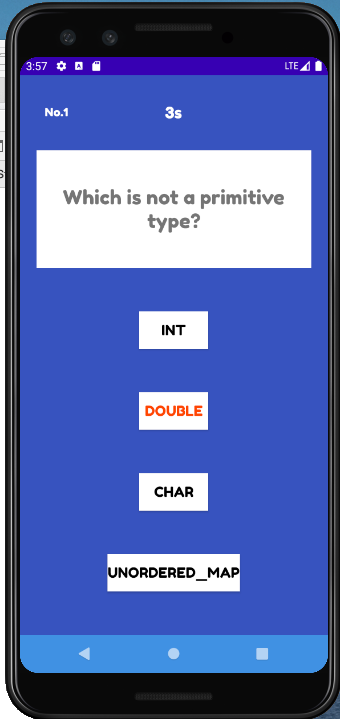




1. After Implementation

After Implementation, we think that a simple UI layout will benefit the user on glancing over. Thus, we decided to design a simple layout.

手机的屏幕

描述已自动生成

1. Color

After Implementation all functionalities, it is time to try different color combinations on our product. After different trying, the plan for the quiz fragments is that we use blue as background color, and Orange and white as text colors. The orange over the blue will give users a freshness of vision. And white over blue makes the text seem orderly. For other pages like friends, login, and upload problem set, we used the same color scheme, like baby blue, cyan, and green.

1. Text and Minor Changes

After colored our application, we made some minor changes to the layout. For instance, text size is 20ps-36ps and the font family is fredoka one. These changes make users feel more comfortable when they are nervous in a quiz. More than that, we change the layout of friends, and the changes make the layout simpler and more uncluttered.

1. Selection Underline

To underline the selection for one question, we make the selection orange when the user clicks the selection. Orange and background blue are the contrasting colors. Therefore, orange will highlight the selection.

1. Icon

Instead of using a colorful image, we decide to use a simple and single-color icon. We chose a delicacy black pencil as our Icon. A good impression is a must in use.



**Part Two**

**Implementation Difficulties**

Summary:

Even though there are lots of difficulties and troubles during the implementation process, we always help each other. Especially when there is a weird bug, we will set up a meeting to solve it. Among several big problems we encountered, the following one difficulty is extremely hard.

Description:

One design of this application is to allow users to do the same quiz at the same time in the same online room. Therefore, synchronization information is necessary. However, there were always some cases that the application failed to receive real time data from firebase. For example, after one user sets a room and start quiz, the application still does not receive the quiz set from firebase. One more example is that ranking result could not be shown corrected for some users.

Solution:

To solve this weird problem, we discussed together and tried multiple ways. Finally, we solved it by creating a thread handler to wait until all firebase operations are completed before proceeding to render the pages with the data form the firebase. And we also design a dialogue window that appears on the screen until the firebase operation is done. Thus, these problems are solved perfectly.

**Part Three**

**External Resources**

Additional Libraries

<https://github.com/qstumn/BadgeView>

Firebase (Authentication & Realtime Database)

Additional Resources

Thread and Handler

Expandable List (Adapter)

Progress Dialog and Alert Dialog

References:

<https://blog.csdn.net/m0_37711172/article/details/80006872>

<https://www.cnblogs.com/yueshangzuo/p/8685810.html>

<https://www.cnblogs.com/xy95/p/5861421.html>

<https://stackoverflow.com/questions/1520887/how-to-pause-sleep-thread-or-process-in-android>

<https://blog.csdn.net/qq_35698774/article/details/79779238>

<https://stackoverflow.com/questions/4172940/how-to-set-background-color-of-a-button-in-java-gui>

https://stackoverflow.com/questions/432037/how-do-i-center-text-horizontally-and-vertically-in-a-textview

<https://developer.android.com/studio/?gclid=CjwKCAiA_eb-BRB2EiwAGBnXXv4C5dSYm5ZG6UovzFVsHtLsz22Hb4L6KXv9NwTQCfwfAuh49lhjkxoCtzEQAvD_BwE&gclsrc=aw.ds>

<https://www.rapidtables.com/web/color/RGB_Color.html>

<https://www.fesliyanstudios.com/sound-effects-search.php?q=incorrect>

**Part Four**

**Meeting Logs**

Anyone on the internet with this link can view.

<https://docs.google.com/document/d/1saCqx6yBPNgiQQpwlJ7tHy_U2oenwC5wBtdWeMqU69Y/edit?usp=sharing>

**Part Five**

**Analyze Competitors**

1. Kahoot

Kahoot is a free game-based learning platform.

1. Zoom

Zoom is our live lectures media. However, Hosts can only choose polls created by themselves. It will be nice to use our application at the same time when students are using zoom.

**Part Six**

**Future Improvement**

1. UI Design Improvement

Add some animations when one user interacts with the application.

1. Create group game

Allow groups of people to play against each other groups (e.g., 3vs3).

1. Send game invitation to friend

One user can invite friends to do a quiz (against the user or with the user).