MINISTRY OF EDUCATION & TRAINING

HOCHIMINH CITY UNIVERSITY OF TECHNOLOGY

BASE PROJECT

Discipline: **INFORMATION TECHNOLOGY**

Major: **INFORMATION TECHNOLOGY**

Instructor: Huynh Danh Chieu Phu

Student's name :

1. Nguyen Hoang An Student's ID: 2080600144 Class: 20DTHQB1

Ho Chi Minh City, 2023

TABLE OF CONTENTS

INSTRUCTOR’S COMMENT

I/ INTRODUCTION

1. Needs of project

Nowadays, due to the amount and pressure of work, there is a need for convenient entertainment to reduce pressure and relax the mind after tiring working days.

For people who focus most of their time on working, then a light, relaxing entertained application is exactly what whey need.

This is why I create this novel reading application, to help people who have strictly time schedules can be

My application is called NovelNest, an mobile app let you read your favorite novels, stories,…

Don’t you want to experience fantasy, dramatic and diverse stories? Don’t you want to be updated with the latest and most outstanding works of domestic and foreign authors? Then you can’t refuse NovelNest.

My novel reading application is an online story reading software. You can read stories anytime, anywhere, as long as you have an internet connection. NovelNest provides you with rich and attractive story genres, such as science fiction, time travel, action, romance, and many other genres.

2. Requirements

- Can choose and read whatever novels that user like.
- A media player that support low music combine with reading can optimize relaxation.
- User need to sign up to create an account to store information includes followed novels, read history, …

II/ SOLUTION

2.1/ Theoretical basis

Before going to next part, these are some tools that I used to built the NovelNest project.

2.1.1/ Android Studio

The official Integrated Development Environment (IDE) for creating Android apps is called Android Studio. Android Studio, which is built on top of IntelliJ IDEA's robust code editor and developer tools, provides additional features to boost your productivity when creating Android apps, like:

- An adaptable build system based on Gradle
- A quick and robust emulator
- A single, integrated environment for Android development
- Use Live Edit to instantly update components in emulators and actual devices.
- You may import sample code and create typical app features with the aid of code templates and GitHub integration.
- Extensive frameworks and tools for testing
- Lint tools for identifying issues with usability, performance, version compatibility, and other issues
- Support for NDK and C++
- Integrated support for Google Cloud Platform, which simplifies the integration of App Engine and Google Cloud Messaging

Android Studio features:

Project structure:

One or more modules with source code files and resource files are included in every Android Studio project. The many kinds of modules consist of:

- Android app modules
- Library modules
- Google App Engine modules

Project files are visible in the Android project window by default in Android Studio, as demonstrated. The main source files for your project are easily accessible thanks to the module-based organization of this view. Under Gradle Scripts at the top level, you can see all of the build files.

Gradle build system:

The build system for Android Studio is based on Gradle, with additional features unique to Android offered by the Android Gradle plugin. This build method operates separately from the command line and as an integrated tool accessible through the Android Studio menu.

Build variants:

You can use the build system to generate many app versions from a single project. This is helpful if your software has both a paid and a free version, or if you wish to release several APKs on Google Play for various device setups.

Multiple APK support:

You may effectively produce numerous APKs based on screen density or ABI with multiple APK support. For instance, you can make different APKs of the same app for hdpi and mdpi screen densities, but treat them as one version and allow them to share ProGuard settings, javac, dx, and test APKs.

2.1.2/ Firebase

With Firebase, you can create and expand popular apps and games that users like. endorsed by Google and relied upon by millions of companies worldwide.

Firebase features:

Authentication: Users can sign into their apps with ease and security thanks to Firebase. Developers can support Google Sign-In, Facebook Sign-In, email and password login, and more using Firebase Authentication.

Realtime database: A cloud-hosted NoSQL database called Firebase Realtime Database enables businesses to store and sync data in real time across all of their consumers' devices. Because of this, creating apps that stay current even when users are not online is made simple.

Cloud Messaging: Businesses can send messages to their users' smartphones with Firebase Cloud Messaging (FCM), even while they aren't using the app. FCM allows developers to change app content, issue push notifications, and do a lot more.

Crashlytics: Firebase Crashlytics is a solution that assists companies in monitoring and resolving app crashes. In-depth crash reports are provided by Crashlytics, enabling prompt root cause analysis and problem resolution.

Performance Monitoring: Firebase Crashlytics helps businesses track and fix app crashes. Crashlytics offers comprehensive crash reports that facilitate rapid root cause analysis and issue resolution.

2.1.3/ Java Programming Language

Java is a popular programming language, created in 1995.

It is owned by Oracle, and more than 3 billion devices run Java.

It is used for:

- Mobile applications (specially Android apps)
- Desktop applications
- Web applications
- Web servers and application servers
- Games
- Database connection
- And much, much more!

Why Use Java?

- Numerous operating systems, including Windows, Mac, Linux, Raspberry Pi, etc., support Java.
- It is among the most widely used programming languages worldwide, and the employment market today is highly demanding for those who know it.
- It is simple to use and straightforward to understand.
- It is free and open-source.
- It is quick, strong, and safe.
- There are tens of millions of developers who support it.

2.2/ Process diagram

Below is a diagram illustrate the process from the beginning when user install and open the app for the first time.

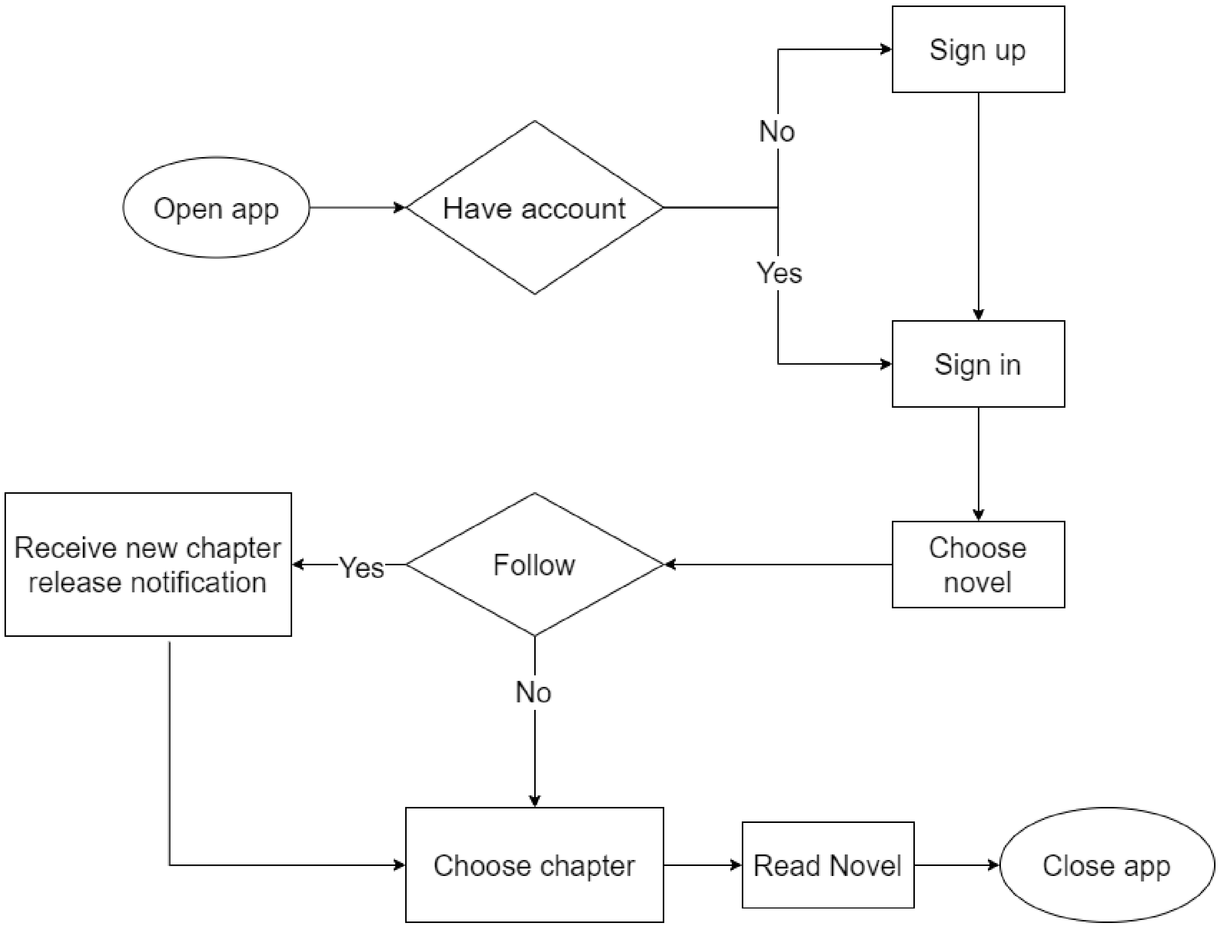


Figure 2.1: Process diagram

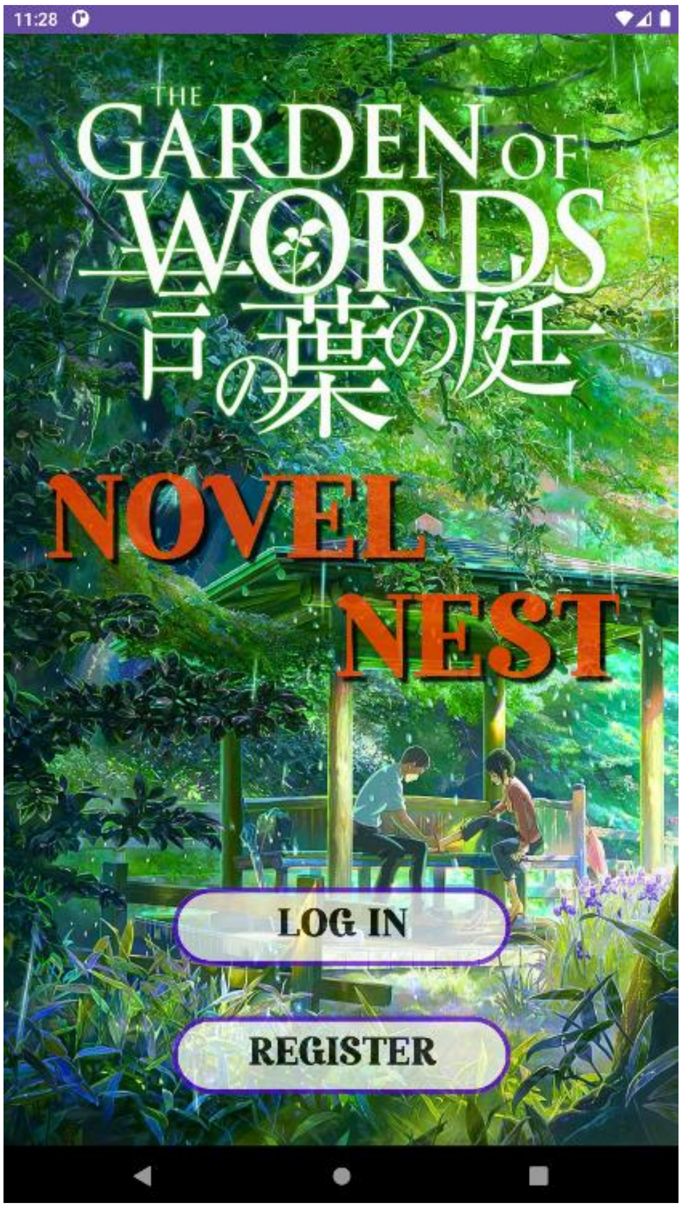
Starting with the Sign up screen, first time user need to register, then sign in with the account just created. After sign in user will choose novel they want to read in the main screen. Then the novel detail screen will be opened, there is a button to follow if user want to receive new chapter update notification, finally just choose chapter that user desire to read and enjoy.

III/ FINAL RESULT

3.1/ Describe product

As mentien before, my application is NovelNest, a mobile platform software for novel reading with special features. I will explain specifically the details of NovelNest below.

3.1.1/ Welcome (Splash screen) activity



This is a splash screen, or welcome screen, the first screen that appear when open the app.

There is a application’s name in the center of screen – NovelNest.

Below that are 2 button, “LOG IN” button will lead to the login screen, while “REGISTER” button leads to the register screen.

Figure 3.1: Splash (Welcome) activity

3.1.2/ Log in activity

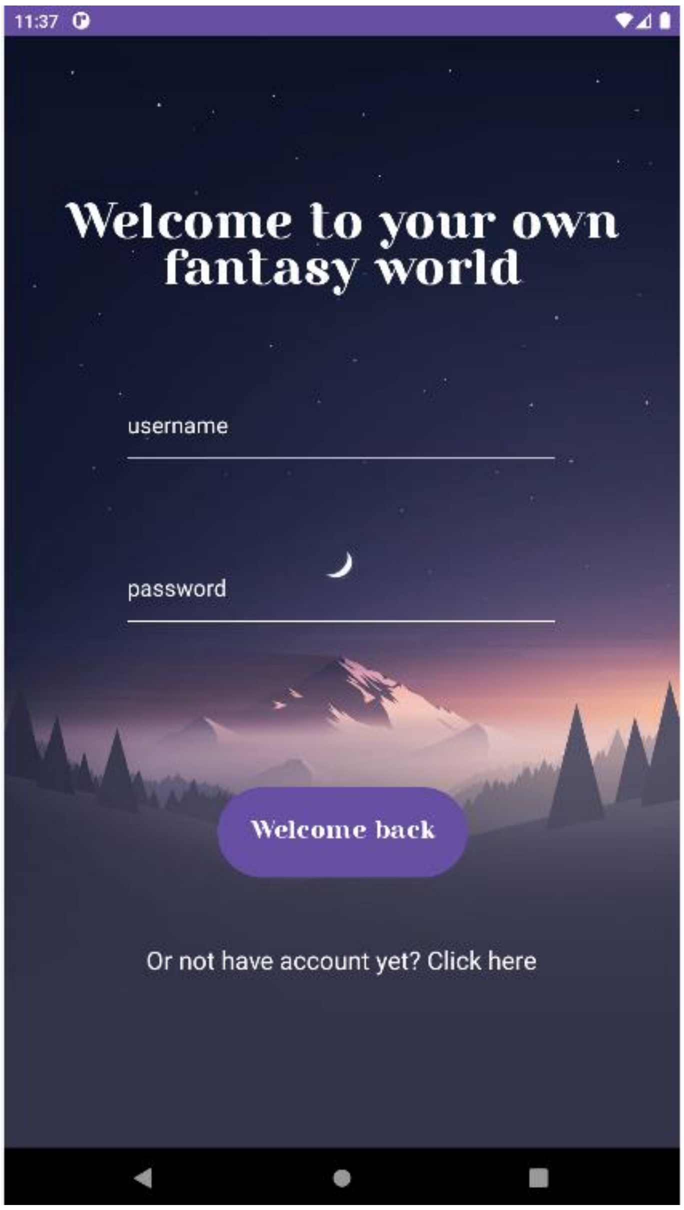

11:37
Welcome to your own fantasy world
username
password
Welcome back
Or not have account yet? Click here

Figure 3.2: Log in screen

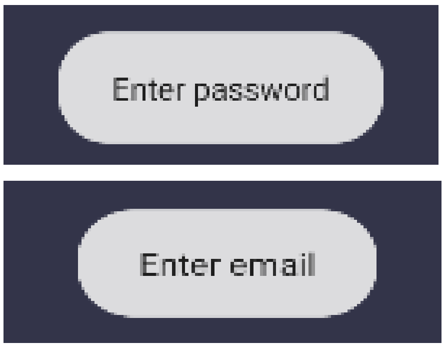


Figure 3.3: Missing toast

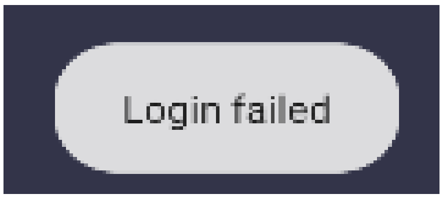


Figure 3.4: Login failed

On this screen, user will have to input 2 field, the username is the email that user use when create account at first time, next field is password of account. When 1 of 2 field is missing when user clicked on the "Welcome back" button (Login button), a small toast will display to ask user to input that field (figure3.3). If the user input wrong username or password, a toast will show that login failed (figure 3.4). And there's an option for user to sign up if user has no account yet by clicking line "Or not have account yet? Click here" (figure 3.5).

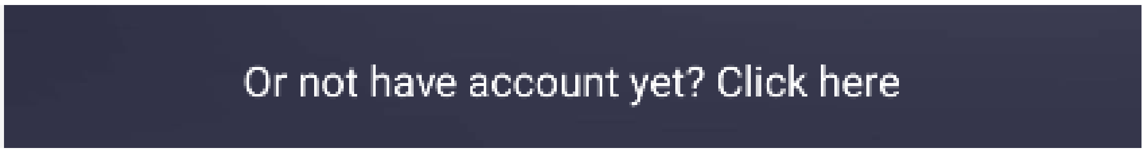


Figure 3.5: Go to sign up navigator

And instead of using local database like SQLite, MySQL, …, I use realtime database by Firebase Authentication to handle the login and register activity. For example this project allow email and password to sign up or sign in (username end with @gmail.com, @yahoo.com, …)

Firebase has available structures to implement into project as below:

```
mAuth.signInWithEmailAndPassword(email, password)
        .addOnCompleteListener(new OnCompleteListener<AuthResult>()
{
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if (task.isSuccessful()) {
                    // Sign in success, update UI with the signed-in
user’s information
                    Toast.makeText(Login.this, “Login success”,
Toast.LENGTH_SHORT).show();
                    Intent intent = new
Intent(getApplicationContext(), MainActivity.class);
                    startActivity(intent);
                } else {
                    // If sign in fails, display a message to the
user.
                    Toast.makeText(Login.this, “Login failed”,
Toast.LENGTH_SHORT).show();
                }
            }
        });
```

3.1.3/ Register activity

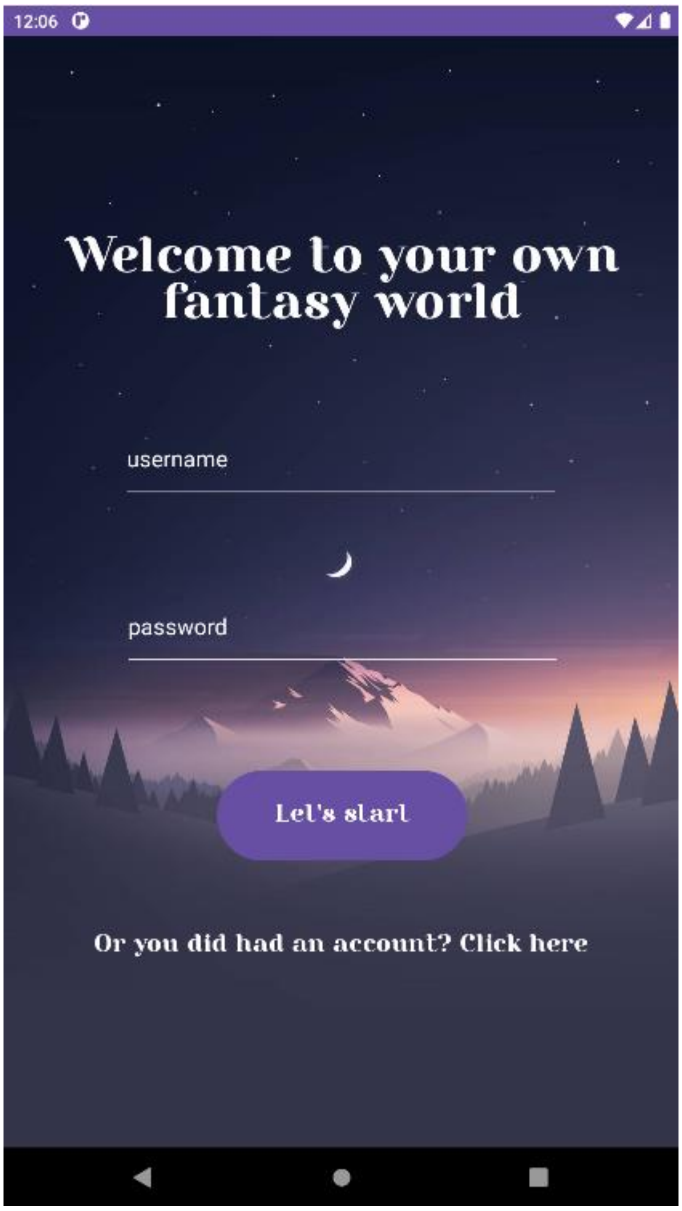

12:06
Welcome to your own fantasy world
username
password
Let's start
Or you did had an account? Click here

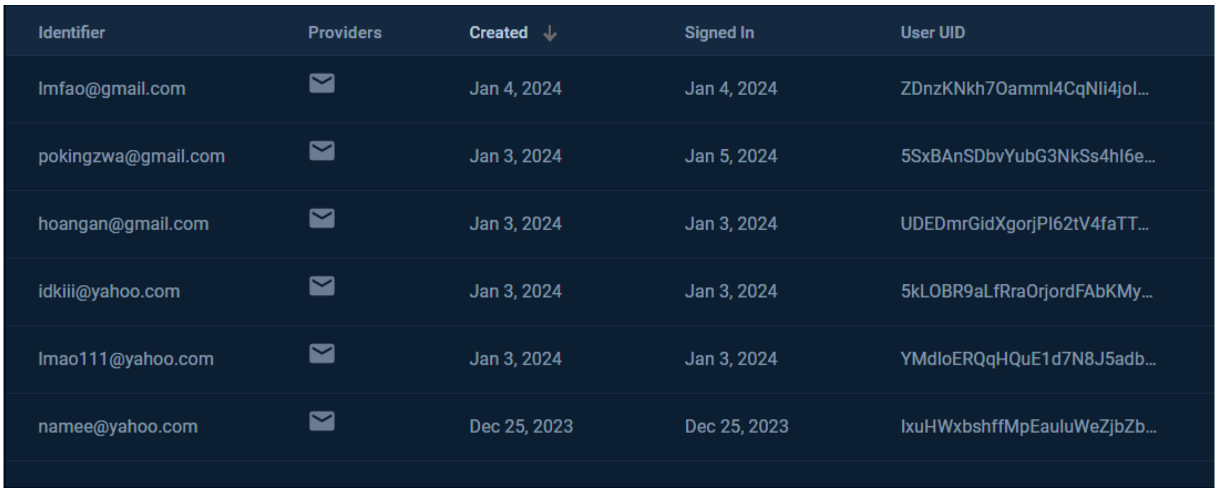
Figure 3.6: Register activity

Almost have the same features as Login activity, this activity also show toast when user don't input 1 or both fields, when the input got wrong email format, and an option to go back to Login activity by clicking the line "Or you did had an account? Click here".

But in stead of login, it create a new account for new users, also implement code structure from Firebase:

```
mAuth.createUserWithEmailAndPassword(email, password)
        .addOnCompleteListener(new OnCompleteListener<AuthResult>()
{
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if (task.isSuccessful()) {
                    // Sign in success, update UI with the signed-in
user's information
                    Toast.makeText(Register.this, "Register
Success.",
                            Toast.LENGTH_SHORT).show();
                    Intent intent = new
Intent(getApplicationContext(), Login.class);
                    startActivity(intent);
                } else {
                    // If sign in fails, display a message to the
user.
                    Toast.makeText(Register.this, "Register
failed.",
                            Toast.LENGTH_SHORT).show();
                }
            }
        });
```

After register succesfully, user will go again to the login activity and login by the account they just created. Below is accounts stored on Firebase database:



Identifier	Providers	Created ↓	Signed In	User UID
lmfao@gmail.com	✉	Jan 4, 2024	Jan 4, 2024	ZDnzKNkh7Oamml4CqNIi4jol...
pokingzwa@gmail.com	✉	Jan 3, 2024	Jan 5, 2024	5SxBAnSDbvYubG3NkSs4hl6e...
hoangan@gmail.com	✉	Jan 3, 2024	Jan 3, 2024	UDEDmrGidXgorjPI62tV4faTT...
idkiii@yahoo.com	✉	Jan 3, 2024	Jan 3, 2024	5kLOBR9aLfRraOrjordFAbKMy...
lmao111@yahoo.com	✉	Jan 3, 2024	Jan 3, 2024	YMdloERQqHQuE1d7N8J5adb...
namee@yahoo.com	✉	Dec 25, 2023	Dec 25, 2023	IxuHWxbshffMpEauluWeZjbZb...

3.1.4/ Home activity

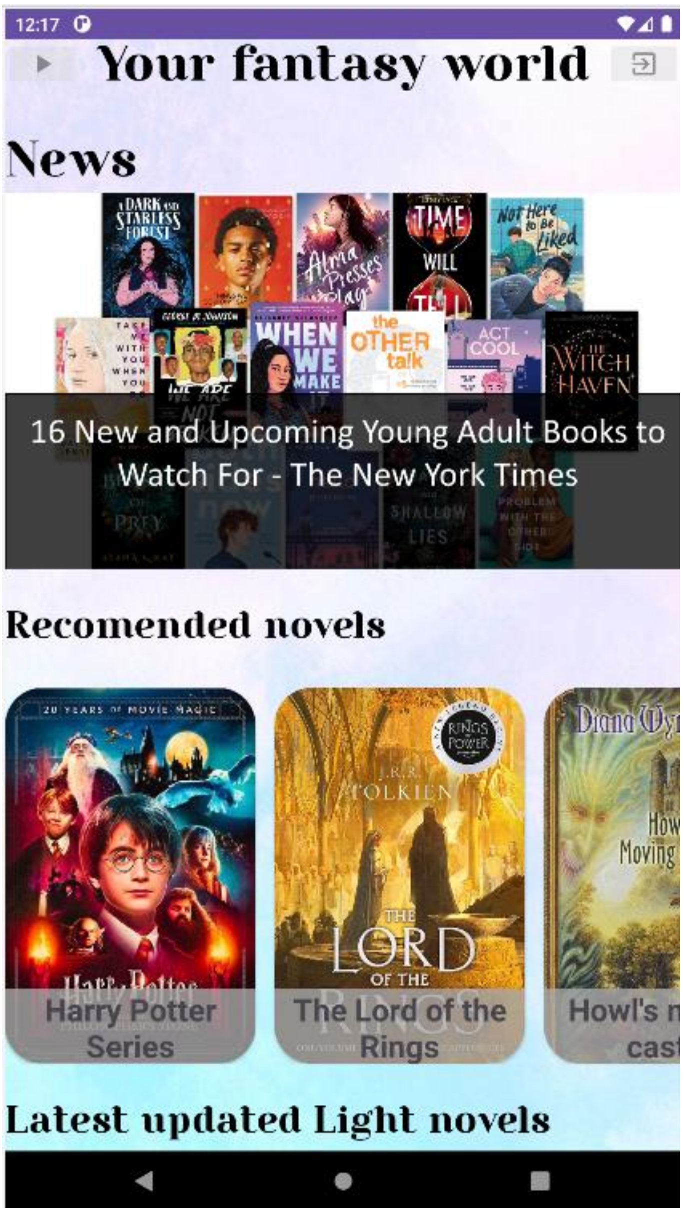

12:17
Your fantasy world
News
16 New and Upcoming Young Adult Books to Watch For - The New York Times
Recomended novels
Harry Potter Series
The Lord of the Rings
Howl's m
cast
Latest updated Light novels

Figure 3.7: Home activity (1)

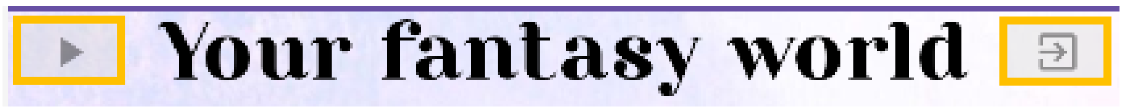


Figure 3.8: Top bar

After user loged in, the home activity will start.

On the top of screen there are 2 buttons, the one on the left is for play and pause the melody that attach to the app, the right one is for log out of the current account.

For the Music button, I use MediaPlayer service of android to play and pause the music whenever user press on this button.

```
music.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (isPlaying) {
            pauseMusic();
        } else {
            playMusic();
        }
    }
});
private void playMusic() {
    if (!mediaPlayer.isPlaying()) {
        mediaPlayer.start();
        isPlaying = true;
        music.setImageResource(R.drawable.baseline_pause_24);
    }
}
private void pauseMusic() {
    if (mediaPlayer.isPlaying()) {
        mediaPlayer.pause();
        isPlaying = false;
        music.setImageResource(R.drawable.baseline_play_arrow_24);
    }
}
```

About the Logout button, Firebase provides a simple line of code to logout of the current user.

```
mAuth.signOut();
```

In this activity, the first feature is News, which shows the news of anything about novels, books, like best sellings, most read books, top novel for age groups, new release in future, update schedule, ….

For this field I use Linear Layout and ImageView to display a slideshow (array) of image contain news.

At the second field, NovelNest will provide user a series of famous novels that many people have read. I use recycler view and card view to show array of books that contain thumbnail and title.

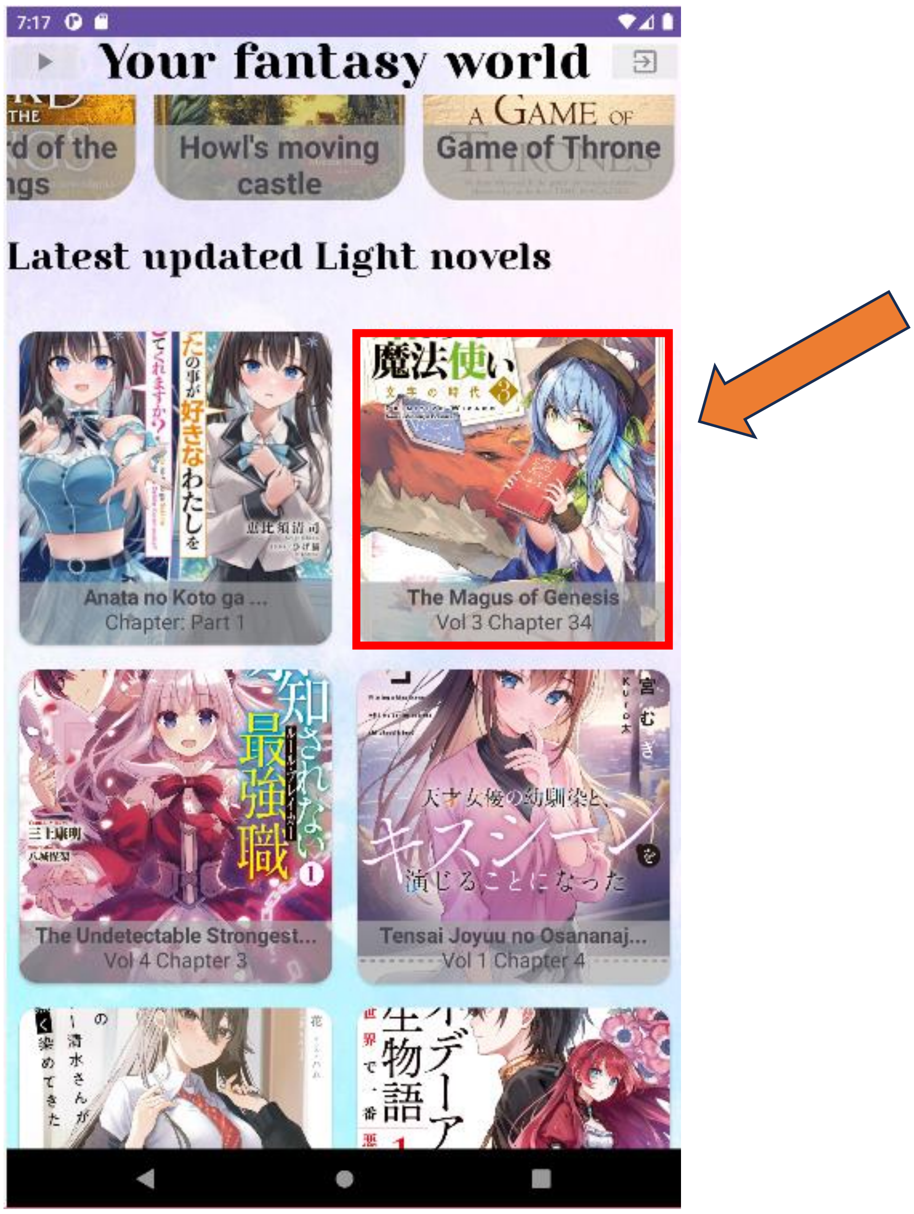
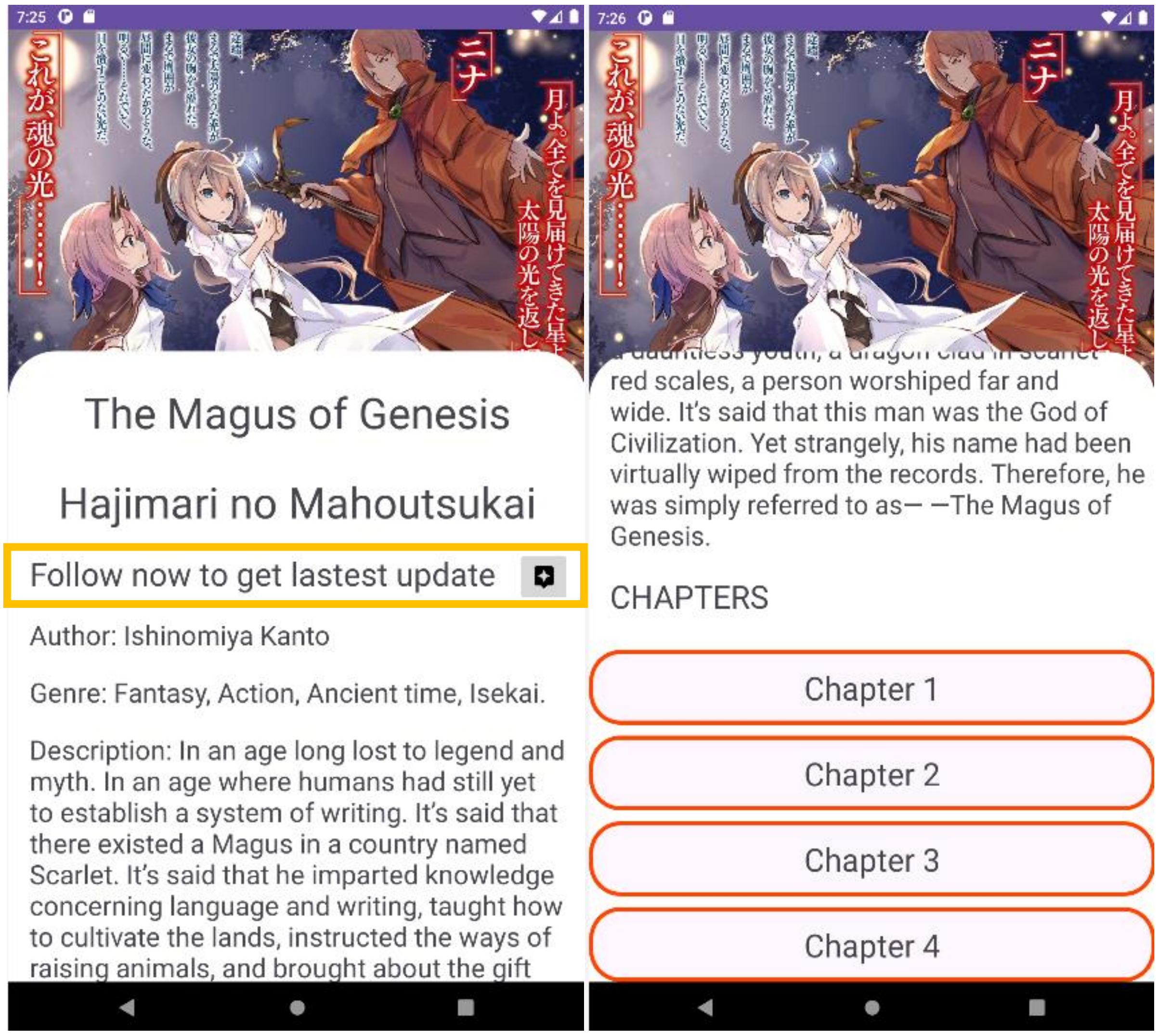


Figure 3.9: Home activity (2)

When user scroll down the home screen, they will see the next field is latest updated Light novels, which uses Grid layout and card view to show covers and titles of books.

3.1.5/ Novel details activity



The Magus of Genesis

Hajimari no Mahoutsukai

Follow now to get lastest update

Author: Ishinomiya Kanto

Genre: Fantasy, Action, Ancient time, Isekai.

Description: In an age long lost to legend and myth. In an age where humans had still yet to establish a system of writing. It's said that there existed a Magus in a country named Scarlet. It's said that he imparted knowledge concerning language and writing, taught how to cultivate the lands, instructed the ways of raising animals, and brought about the gift

red scales, a person worshiped far and wide. It's said that this man was the God of Civilization. Yet strangely, his name had been virtually wiped from the records. Therefore, he was simply referred to as— —The Magus of Genesis.

CHAPTERS

Chapter 1

Chapter 2

Chapter 3

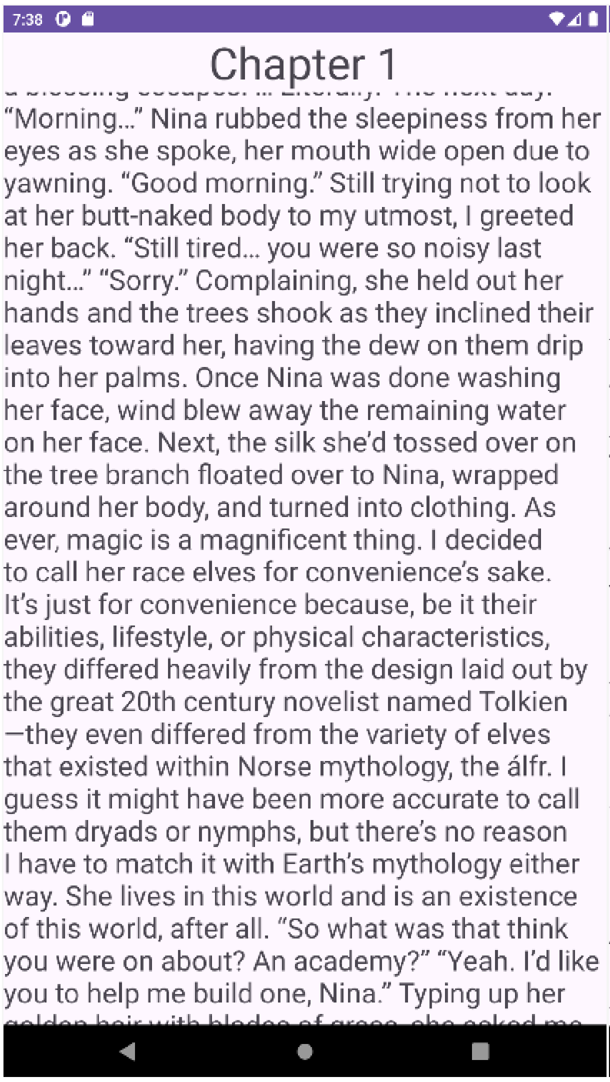
Chapter 4

Figure 3.9 and 3.10: Novel details activity

After user decide what they want to read, just click on the card and the novel details activity will start. This illustrate will be the novel with title "The Magnus of Genesis".

In this screen, user will be introduced about the novel with information like the title and the other names of that novel, the author, genres, and a short description.And when user scroll down there will be a field for user to choose which chapter they would like to read.

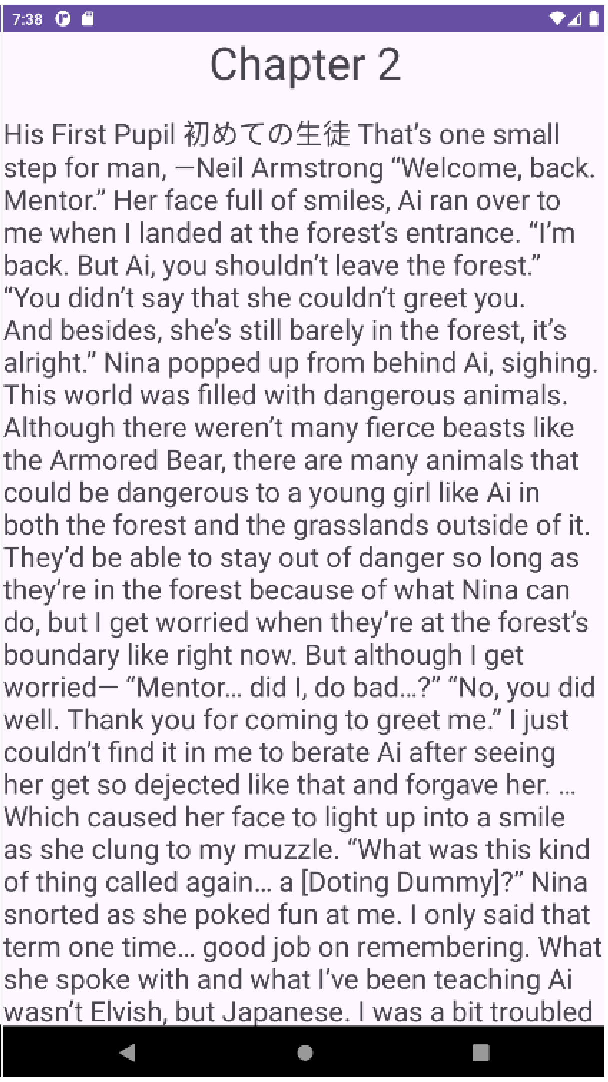
3.1.6/ Chapter activity



7:38

Chapter 1

"Morning..." Nina rubbed the sleepiness from her eyes as she spoke, her mouth wide open due to yawning. "Good morning." Still trying not to look at her butt-naked body to my utmost, I greeted her back. "Still tired... you were so noisy last night..." "Sorry." Complaining, she held out her hands and the trees shook as they inclined their leaves toward her, having the dew on them drip into her palms. Once Nina was done washing her face, wind blew away the remaining water on her face. Next, the silk she'd tossed over on the tree branch floated over to Nina, wrapped around her body, and turned into clothing. As ever, magic is a magnificent thing. I decided to call her race elves for convenience's sake. It's just for convenience because, be it their abilities, lifestyle, or physical characteristics, they differed heavily from the design laid out by the great 20th century novelist named Tolkien —they even differed from the variety of elves that existed within Norse mythology, the álfr. I guess it might have been more accurate to call them dryads or nymphs, but there's no reason I have to match it with Earth's mythology either way. She lives in this world and is an existence of this world, after all. "So what was that think you were on about? An academy?" "Yeah. I'd like you to help me build one, Nina." Typing up her



7:38

Chapter 2

His First Pupil 初めての生徒 That's one small step for man, —Neil Armstrong "Welcome, back. Mentor." Her face full of smiles, Ai ran over to me when I landed at the forest's entrance. "I'm back. But Ai, you shouldn't leave the forest." "You didn't say that she couldn't greet you. And besides, she's still barely in the forest, it's alright." Nina popped up from behind Ai, sighing. This world was filled with dangerous animals. Although there weren't many fierce beasts like the Armored Bear, there are many animals that could be dangerous to a young girl like Ai in both the forest and the grasslands outside of it. They'd be able to stay out of danger so long as they're in the forest because of what Nina can do, but I get worried when they're at the forest's boundary like right now. But although I get worried— "Mentor... did I, do bad...?" "No, you did well. Thank you for coming to greet me." I just couldn't find it in me to berate Ai after seeing her get so dejected like that and forgave her. ... Which caused her face to light up into a smile as she clung to my muzzle. "What was this kind of thing called again... a [Doting Dummy]?" Nina snorted as she poked fun at me. I only said that term one time... good job on remembering. What she spoke with and what I've been teaching Ai wasn't Elvish, but Japanese. I was a bit troubled

Figure 3.11 and 3.12: Chapter activity

This is what it looks like when user choose chapter they want to read. For example these are chapter 1 and chapter 2 of the novel mention above.

For this I use recycler view onItemClick to pass the data of chapters from the Novel details activity to each chapter in Chapter activity. The code to pass and receive of 2 activity will show below.

Pass data from Novel details activity:

```
@Override
public void onItemClick(int position) {
    Intent intent = new Intent(DetailMagnus.this,
chapters.class);
    intent.putExtra("Chap_number",
chaps.get(position).getChap_num());
    intent.putExtra("Chap_content",
chaps.get(position).getChap_content());
    startActivity(intent);
}
```

Catch or receive data of Chapter activity :

```
String chapnumber = getIntent().getStringExtra("Chap_number");
String chapcontent = getIntent().getStringExtra("Chap_content");

tvc1.setText(chapnumber);
tvcontent.setText(chapcontent);
```

3.1.7/ Notification

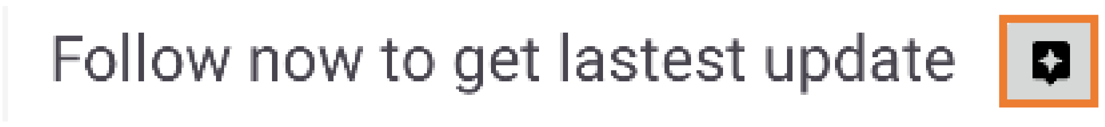
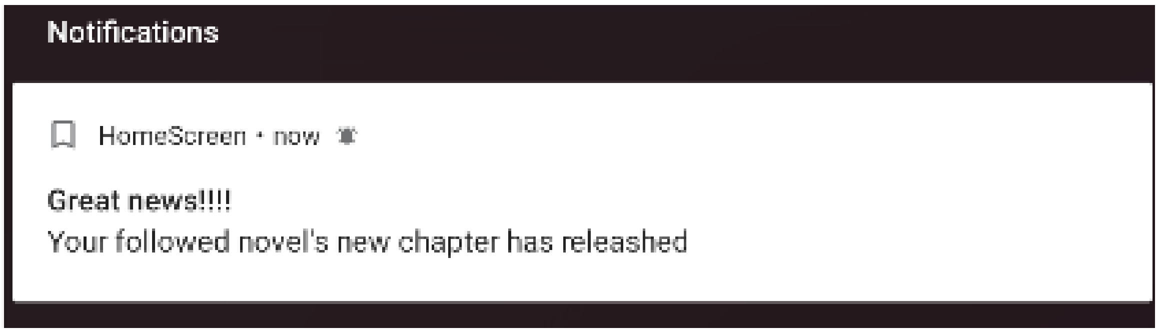


Figure 3.13: Follow

If user want to receive the notification about new chapter update in the future, they can press the button in figure 3.13. A notification will arrive to let user know that the new chapter of their followed novels has released (figure 3.14).


Notifications

HomeScreen • now

Great news!!!!
Your followed novel's new chapter has releashed

Figure 3.14: New chapter notification

To create the notification, I use Broadcast receiver and push notification.

First I created the channel:

```
private void createNotificationChannel(){
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O){
        CharSequence name = "Channel name?";
        int importance = NotificationManager.IMPORTANCE_DEFAULT;
        NotificationChannel channel = new
NotificationChannel("Followed", name, importance);

        NotificationManager notificationManager =
getSystemService(NotificationManager.class);
        notificationManager.createNotificationChannel(channel);

    }
}
```

Then to receive and create notification:

```
public void onReceive(Context context, Intent intent) {
        NotificationCompat.Builder builder = new
NotificationCompat.Builder(context, "Followed")
                .setSmallIcon(R.drawable.smallico)
                .setContentTitle("Great news!!!!")
                .setContentText("Your followed novel's new chapter
has releashed")
                .setPriority(NotificationCompat.PRIORITY_DEFAULT);

        NotificationManagerCompat notificationManager =
NotificationManagerCompat.from(context);
        notificationManager.notify(220, builder.build());
    }
}
```

3.2/ Conclusion

3.2.1/ Summary

NovelNest is a useful tool for people who love to read novels, stories, because my application will provide thousands of novels, great variety of genres, combines with low music to give user best reading experiences.

3.2.2/ Incomplete features

Because of lack of time, my project still has features not yet built in, includes:

- An admin process that allow admin to upload the chapters, novels, ….
- A user profile that contain read history, followed novels, ….
- Notification will arrive when the new chapter of user's followed novel released.

3.2.3/ Develop direction

In addition to the ones that mentioned above, this project will have many potential feature to be developed, contains:

- Reading community: connect with people who share interests about reading novels.
- Comment feature: allow user to comment to the novel they are reading.
- Buy E-novel: user can buy e-novel to read.

- Payment process: support buy e-novel feature.

And much more will be release in the near future.

IV/ REFERENCES

Android studio:
https://developer.android.com/studio?gclid=Cj0KCQiAy9msBhD0ARIsANbk0A-pODXusES0cbAcdw8ISbuQEcb5K9eX9UXcB5SElwbDGeI6zznnhFwaAv7sEALw_wcB&gclsrc=aw.ds

Firebase: https://firebase.google.com

Java programming language: https://www.w3schools.com/java/java_intro.asp