MINISTRY OF EDUCATION & TRAINING

HOCHIMINH CITY UNIVERSITY OF TECHNOLOGY

MAJOR PROJECT

TRIP PLANNER

Discipline: **INFORMATION TECHNOLOGY**

Major: **INFORMATION TECHNOLOGY**

Supervisor: Vo Tan Dung

Student's name :

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

Ho Chi Minh City, 2023

Table Of Contents

Catogery of pictures

Chapter 1: Overview

1.1: Introduction

Travel is one of the most interesting and common activity that everybody does on special holidays, summer vacation, spring break,…. But due to Covid-19 from 2019, many people had been worried and were not outside of the house for a long time. Till 2022, COVID-19 in remission, that people have no concern to go on a trip. With high technology nowadays, we can search any information that we need to know about the trip such as hotel, interesting places, restaurant, ….

While we can search anything from everywhere, and book at different website, then this application I am going to show you will have everything you want in one simple app.

1.2: Research

Because of the demand, you can find a lot of online booking app, website, tour guide, …. But following that, there are not many actually quality apps on the marketplace. Some apps will be very expensive to book although the trip doesn't cost that much, some will just suggest normal and not so interesting places of where you are going to visit,…

That is why I created this application, to introduce many quality and suitable to your finance ability.

1.3: Task of project

- Learn about Android programming, and apply it to create an application.
- Learn about SQLite, a database that can stores customer's information, places, hotels,…
- Research about the demand of customer.

1.4: Project's structure

When you lauch this application, you will see a simple welcome screen to choose login or sign up.

Since this is the first time you access the system, you need to create an account with username and password.

If you choose login without an account, there is an option below let you return to the sign up screen. Here you will need to provide an username and a password to access the application.

After you finish created account or logged in, you will get to home screen, this screen provides many interesting places, along with price of the trip.

When you tap a place you like, this lead you to details screen which have description about this trip.

If you decide to book a trip, the system will go to history screen where all of your trip books are show here.

Chapter 2: Theoretical basis

2.1: About Android Studio

The official Integrated Development Environment (IDE) for creating Android apps is called Android Studio. Android Studio, which is based on the robust code editor and developer tools from IntelliJ IDEA, offers additional features that increase your efficiency when creating Android apps.

Figure 2.1.1: Official logo of Android Studio.

Android Studio employs a Gradle-based build system, an Android emulator, code templates, and GitHub integration to assist application development for the Android OS. In Android Studio, every project contains one or more modalities that include source code and resource files. These modalities consist of Google App Engine modules, Library modules, and modules for Android apps.

Android Studio 3.0 or later supports Kotlin, as well as "all Java 7 language features and a subset of Java 8 language features that vary by platform version." It also supports all the programming languages that IntelliJ (and CLion) offer, including Java, C++, and others with extensions, such as Go.Some Java 9 features are backported by external projects. Although IntelliJ claims that Android Studio supports all currently available Java versions, including Java 12, it is unclear to what extent Java versions up to Java 12 are

supported by Android Studio (the documentation cites Java 8 support in part). Android can use at least some of the new language features up to Java 12.

Project structure:

One or more modules with source code files and resource files can be found in each project in Android Studio. The various module types consist of:

- Android app modules.
- Library modules.
- Google app engine modules.

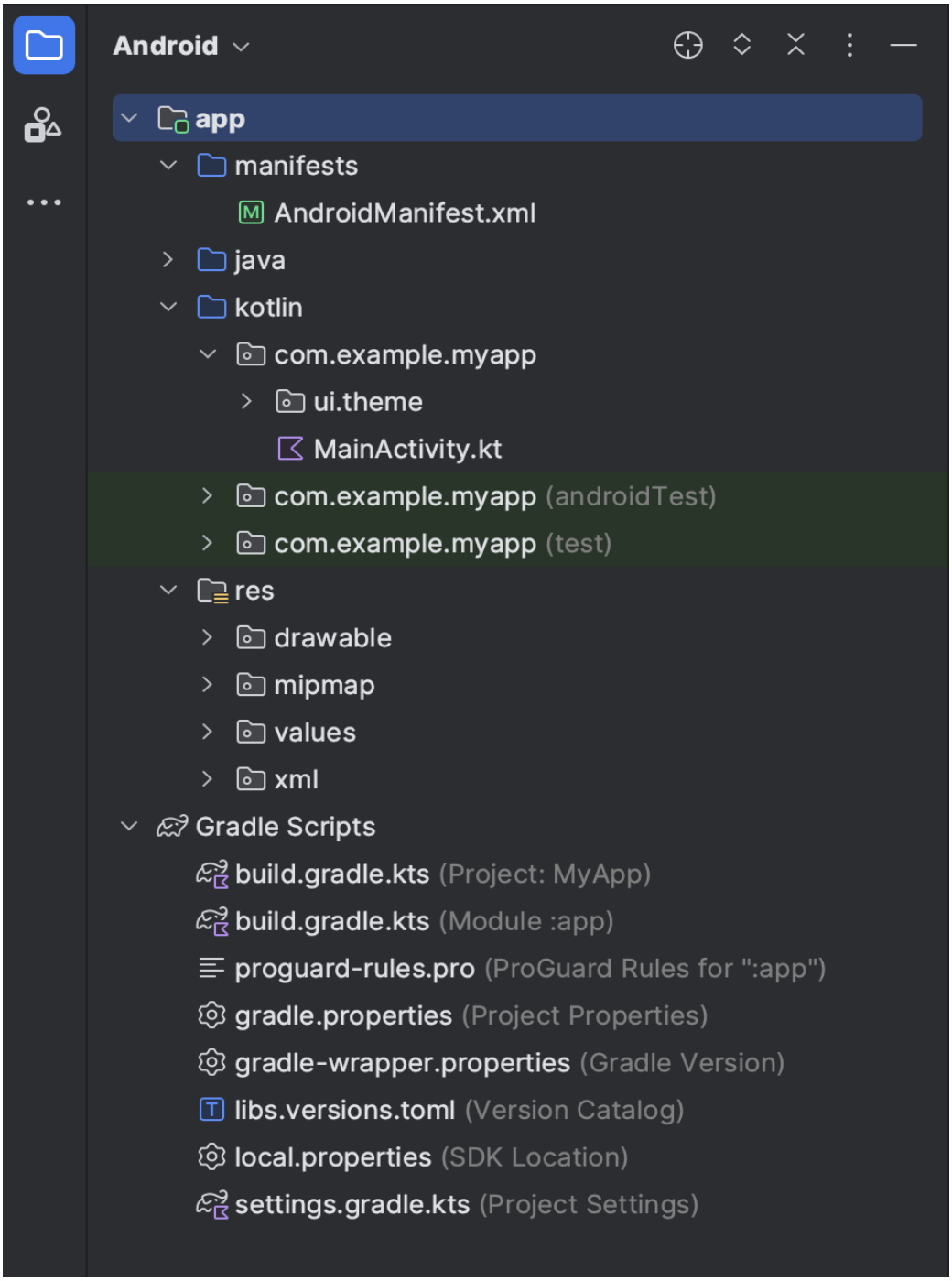

Figure 2.1.2: The project files in Android projects

Figure 2.1.2 illustrates how Android Studio by default presents your project files in the Android project window. The important source files for your project are easily accessible thanks to the module-based organization of this view. At the top level, under Gradle Scripts, all build files are accessible.

Each app module contains the following folders:

- manifests: Contains the AndroidManifest.xml file.
- java: Contains the Kotlin and Java source code files, including JUnit test code.
- res: Contains all non-code resources such as UI strings and bitmap images.

The Android project structure on disk differs from this flattened representation. To see the actual file structure of the project, select Project instead of Android from the Project menu.

Gradle build system:

The build mechanism of Android Studio is based on Gradle, with the Android Gradle plugin providing additional Android-specific features. This build mechanism may be launched both independently from the command line and as an integrated tool from the Android Studio menu. The following activities can be carried out using the build system's features:

- Customize, configure, and extend the build process.
- Create multiple APKs for your app with different features, using the same project and modules.
- Reuse code and resources across source sets.

When using Kotlin (which is advised), the build files for Android Studio are named build.gradle.kts, while Groovy build files are named build.gradle. They are simple text files that setup the build with elements from the Android Gradle plugin using the Kotlin or Groovy syntax. One top-level build file for the entire project and separate module-level build files for each module are present in every project. Android Studio creates the necessary build files automatically when you import an existing project.

Build varians:

You can use the build system to produce various iterations of the same app from a single project. This is helpful if you want to offer numerous APKs on Google Play for various device setups or if your software has both a free and a premium edition.

Multiple APK support:

Multiple APK support lets you efficiently create multiple APKs based on screen density or ABI. For example, you can create separate APKs of an app for the hdpi and mdpi screen densities, while still considering them a single variant and letting them share test APK, javac, dx, and ProGuard settings.

Resource Shrinking:

With Android Studio's resource shrinking feature, unwanted resources from your packed app and library dependencies are automatically removed. For instance, if you are not currently using Google Sign-In and your app uses Google Play services to access Google Drive functionality, resource shrinking may eliminate the various drawable components for the SignInButton buttons.

Manage dependencies:

Your project's dependencies are listed by name in the module-level build script. Dependencies are discovered by Gradle and made accessible in your build. In your build, you can specify module dependencies, external binary requirements, and local binary dependencies.File gradle.kts.

By default, Android Studio sets up projects to use the Maven Central Repository. The project's top-level build file contains this configuration.

Debug and profile tools:

With the aid of inline debugging and performance analysis tools, Android Studio aids in the testing and optimization of your code.

Inline debugging:

Improve your code walkthroughs in the debugger view by using inline debugging to check references, expressions, and variable values as you go.
Inline debug information includes:

- Inline variable values
- Objects that reference a selected object
- Method return values
- Lambda and operator expressions
- Tooltip values

Performance profilers:

The performance profilers offered by Android Studio make it simple to monitor the memory and CPU utilization of your app, identify deallocated objects, uncover memory leaks, improve graphics performance, and examine network requests.

Android Studio advantages and disadvantages:

- Advantages:
    - Faster Coding

      The ability to code more quickly is one of Android Studio's primary benefits. This is because capabilities like live rendering, which let you see changes you make to your code as they happen, are available.
    - Fast Emulator

      Given that the emulator's sluggishness used to be one of the things developers complained about the most, Android Studio 3.0's new emulator is thought to be extremely quick, which is a significant gain. You can test your software on a variety of devices thanks to the new version, which is touted to start up in less than 6 seconds.
    - Solid Testing

      When developing a mobile app, testing is crucial. Thankfully, Android Studio gives developers all the tools they require to test their apps across a variety of devices.
    - Firebase Support and Integrated Cloud

      Both cloud messaging and support for Firebase are built into Android Studio. This is a huge benefit because it makes it simple for developers to incorporate push notifications and other similar capabilities into their apps.
    - Collaborative

      The fact that Android Studio promotes team cooperation is yet another significant benefit. For instance, it makes it simple for engineers to share and access code snippets. Additionally, they can combine edits from several team members without worrying about back-and-forth emails and texts.
- Disadvantages:
    - High Hardware Requirements

      The fact that Android Studio uses a lot of RAM and CPU resources to function effectively is one of its key issues. This might be a significant drawback,

particularly for programmers on a tight budget or those using outdated computers.

○ System Lag

Other issues that some developers have with Android Studio include system latency, which can result in events and slow down the development process.

○ Uses a Lot of RAM

The fact that Android Studio consumes a lot of RAM is another drawback. This can be particularly problematic for programmers using older systems or those with less RAM.

○ Slow Installation

For developers who are eager to get started, the lengthy installation process for Android Studio can be a hassle.

2.2: About Java programming language

Programming languages like Java are frequently used to create web apps. With millions of Java programs in use today, it has been a well-liked option among developers for more than 20 years. Java is a network-centric, multi-platform, object-oriented language that may also be used as a platform by itself. It is a quick, safe, and dependable programming language for creating anything from big data applications to server-side technologies to mobile apps and enterprise software.

Figure 2.2.1: Official logo of Java

Most uses of Java:

- Game Development

  Java is used to create a lot of well-known video games, computer, and mobile games. Java technology is used to create even contemporary video games that incorporate cutting-edge hardware like virtual reality or machine learning.

- Cloud computing

  Java is frequently described as WORA, or Write Once and Run Anywhere, which makes it ideal for distributed cloud-based applications. Java is the language of choice for cloud providers to run applications on a variety of underlying platforms.

- Big Data

  Java is utilized for data processing engines that can handle enormous amounts of real-time data and complex data sets.

- Artificial Intelligence

  Java is a machine learning library powerhouse. For the development of artificial intelligence applications like deep learning and natural language processing, its stability and speed make it ideal.

- Internet of Things

  Java has been used to program hardware and sensors in edge devices that can connect to the internet on their own.

Java is well-liked because it was made to be simple to use. Some factors motivating engineers to stick with Java over competing programming languages are as follows:

- High quality learning resources

  Since Java has been around for a while, new programmers have access to a wealth of learning materials. Developers are helped through the learning curve through extensive books, classes, and detailed documentation. Additionally, novice programmers can begin writing code in Core Java before switching to Advanced Java.

- Inbuilt functions and libraries

Developers don't have to create each new function from scratch while utilizing Java. Instead, Java offers a large ecosystem of built-in libraries and functions to create a variety of applications.

- Active community support

Java has a large, active user base and a supportive community that may help developers when they run into coding problems. The Java platform software is also regularly updated and maintained.

- High-quality development tools

For automated editing, debugging, testing, deployment, and change management, Java provides a number of tools. Java programming is time and money efficient because to these technologies.

- Platform Independent

Java code doesn't need to be rewritten to operate on any underlying platform, including Windows, Linux, iOS, and Android. In the modern setting, where we wish to execute programs on various devices, this makes it very powerful.

- Security

Untrusted Java code can be downloaded by users over a network and run in a safe environment where it cannot cause any harm. Untrusted code is unable to read or write to the host system's hard drive or infect it with a virus. Java's security settings and limitations have a lot of configuration options.

Every programming language is a form of machine communication. Hardware on computers only reacts to electrical communication. Human language and hardware language are connected by high-level programming languages like Java. A developer needs to comprehend two concepts in order to use Java:

- Java language and APIs

The Java programming language's syntax and semantics are defined by Java. This covers the fundamental terminologies and guidelines for writing algorithms, such as primitive data types, if/else statements, loops, etc.

APIs are crucial pieces of software that come with the Java Platform. These are pre-written Java applications that can be used in your own code to plug and play pre-

existing functionality. Java APIs can be used, for instance, to alter text, execute mathematical operations, and retrieve the date and time.

Any Java application code created by a developer will often incorporate both fresh and previously written code from Java libraries and APIs.

- Java Virtual Machine

The Java platform and the machine hardware underneath are separated from one another by the Java Virtual Machine. Only computers with the JVM installed on them can run Java source code. The history of programming provides the reason for the necessity of the Java Virtual Machine.

Because Java is platform-independent and hence works on all platforms, it is ideally suited for developing Android apps. Java features a separate API and runtime environment called Java Runtime Environment. Using a Java Virtual Machine (JVM), the Java program was the first programming language to incorporate the two methodologies mentioned above. The Java Virtual Machine is the name given to the Java code compiler. To create bytecode, any Java file must first be compiled. Only the JVM supports running Java bytecode. The bytecode is subsequently translated by the JVM in order to be executed on the underlying hardware platform. Because of this, the JVM will interpret the program for Windows if it is executing on a Windows machine. On the other hand, the JVM will interpret it for Linux if it is operating on an open-source platform like Linux. Java, one of the most popular programming languages on GitHub, serves as the foundation for a sizable number of Android apps.

Java programming language advantages and disadvantages:

- Advantages:
    - Java is Simple

      A straightforward programming language is one that is simple to learn and comprehend. Java is one of the easiest programming languages to learn and use thanks to its basic and understandable codes.

      Java also simplifies the implementation of programs by eliminating all the difficult C and C++ elements like pointers, structures, and unions.

    - Java is an Object-Oriented Programming Language

Java's ability to be an object-oriented programming language is one of its key benefits. We are all aware of how difficult and complicated procedural languages are to use. Java is more simpler to implement and much more secure when using the OOPs approach. OOPs ideas assist Java in resolving practical issues. By splitting up enormous code into smaller, labeled chunks, it also makes maintenance easier.

○ Java is a Secure Language

Pointers were used in languages like C and C++ to access memory locations. Given that pointers might result in illegal memory access, this poses a security issue. Java also made use of OOPs ideas like inheritance, encapsulation, and abstraction, which boosts security and prevents unauthorized user access.

○ Java is cheap and economical to maintain

Java is inexpensive and simple to develop as a result of its straightforward structure. Java can operate on any computer, independent of the hardware, which drastically lowers the cost of development.

○ Java is platform-independent

Java adheres to the WORA (Write Once, Run Anywhere) functionality. Any other system with Java installed can run the Java programs created on one system. Because Java's compatibility is not based on the operating system or hardware, it is platform-independent and incredibly adaptable.

○ Java is a High-Level Programming Language

Human language, a high-level language, is used to create Java programs. With a few straightforward and simple to understand syntaxes, it is comparable to English. Java features an interpreter that converts the code to a language that a computer can understand.

○ Java supports portability feature

Java is a language that is very adaptable. This is due to Java's platform independence and the fact that it doesn't need any specialized hardware to function. Java is now practically compatible with all devices thanks to this.

○ Java supports Multithreading

The tiniest component of a process is a thread. Multithreading is a crucial element to get optimal CPU utilization. Programming language Java allows for multithreading. Java allows us to run several threads simultaneously. To

improve the application's effectiveness and performance, they share a single memory. Each thread operates separately from the others.

○ Java is stable

To fix issues, Java is updated frequently. Java is one of the most reliable programming languages available as a result. Through updates, almost all bugs are instantly fixed. Because of this, it's crucial to update Java frequently.

- Disadvantages:

○ Slow and Poor Performance

When compared to native programming languages like C and C++, Java uses more memory. Due to the interpreter's added work in translating the code into machine language, Java is also slower than them. The JVM conducts a number of backend tasks that slow down the program. Java provides automatic garbage collection, so it runs continuously in the background, hurting performance.

○ Poor GUI

When it comes to GUI, Java is well behind. Java's GUI builder is inadequate and unable to create complicated user interfaces. For designing GUIs, Java offers a variety of frameworks, including Swing, SWT, JavaFX, JSF, and others. However, these frameworks are not sufficiently advanced to create intricate GUIs. The GUI builders in contemporary languages like Python, R, C#, etc. are better.

○ No backup facility

Java has virtually no functionality for user data backup. It primarily focuses on data storage, but there is no backup facility for those data.

○ Significant memory space required

Compared to other programming languages like C and C++, Java uses more memory. Java has terrible memory management. Java has a garbage collector, however it negatively affects performance.

2.3: About SQLite

Figure 2.3.1: Official logo of SQLite.

A self-contained, serverless, zero-configuration, transactional SQL database engine is implemented by SQLite, an in-process library. Since SQLite's source code is in the public domain, it may be used for any objective, whether public or private. The most extensively used database in the world, SQLite has more applications than we can list, including a number of well-known initiatives.

SQLite is an embedded SQL database engine. SQLite lacks a dedicated server process, unlike the majority of other SQL databases. SQLite directly reads and writes to common disk files. One disk file can store an entire SQL database, including all of its tables, indices, triggers, and views. The database file format is cross-platform, allowing you to freely copy databases between big-endian and little-endian architectures as well as between 32-bit and 64-bit platforms. These characteristics make SQLite a well-liked option for application file formats. The US Library of Congress suggests storing data in SQLite database files. Consider SQLite to be a substitute for fopen() rather than Oracle.

SQLite is a compact library. With all features enabled, the library size can be less than 750KiB, depending on the target platform and compiler optimization settings. (64-bit code is larger. And some compiler optimizations such as aggressive function inlining and loop unrolling can cause the object code to be much larger.) There is a tradeoff between memory usage and speed. SQLite generally runs faster the more memory you give it. Nevertheless, performance is usually quite good even in low-memory environments. Depending on how it is used, SQLite can be faster than direct filesystem I/O.SQLite may be quicker than direct filesystem I/O depending on how it is utilized.

Prior to every release, SQLite is thoroughly tested, and it has a solid reputation for dependability. Only testing and verification-related code makes up the majority of the SQLite source code. A 100% branch test coverage is attained using an automated test

suite that performs hundreds of millions of test cases and hundreds of millions of individual SQL statements. Errors in disk I/O and memory allocation are handled gently by SQLite. Even when a system crash or power outage interrupts a transaction, it still remains ACID. Automated tests using specialized test harnesses that mimic system failures are used to validate all of this. Of course, there are still bugs despite all of this testing. However, SQLite is transparent and honest about all flaws and gives bug lists and minute-by-minute chronologies of code changes, unlike some comparable projects (especially commercial rivals).

SQLite advantages and disadvantages:

- Advantages:
    - Lightweight

      Due to its low weight, SQLite is suitable for usage as embedded software in a variety of electrical devices, including televisions, mobile phones, cameras, and other household appliances.
    - Better Performance

      For SQLite databases, reading and writing activities happen relatively quickly. Compared to File system, it is almost 35% faster.

      Instead of reading the entire file and storing it in memory, it merely loads the necessary data.

      Small edits only affect the portions of the file that were altered.
    - No Installation Needed

      Learning SQLite is really simple. It does not require installation or configuration. Your computer will be prepared to create the database when you download the SQLite libraries.
    - Reliable

      Because it updates your content continuously, little to no effort is lost in the event of a crash or power outage.

      Compared to specially created file I/O codes, SQLite is less prone to errors. Because SQLite queries are smaller than comparable procedural codes, the likelihood of problems is low.
    - Portable

      All 32-bit and 64-bit operating systems, as well as big- and little-endian architectures, are compatible with SQLite.

Multiple processes can read from and write to the same application file without interfering with one another.

Without any compatibility problems, it can be used with all programming languages.

○ Accessible

Many different third-party tools can access the SQLite database.

The content of a lost SQLite database is more likely to be recoverable. Code expires faster than data.

○ Reduce Cost and Complexity

Because short SQL queries may access and update content rather than complex and error-prone procedural queries, it lowers the cost of the program.

Simply simply adding new tables and/or columns, SQLite can be readily expanded in upcoming releases. Additionally, it maintains backwards compatibility.

- Disadvantages:
  - ○ SQLite is used to handle low to medium traffic HTTP requests.
  - ○ Database size is restricted to 2GB in most cases.

Chapter 3: Experimental results

3.1: Dataflow diagram

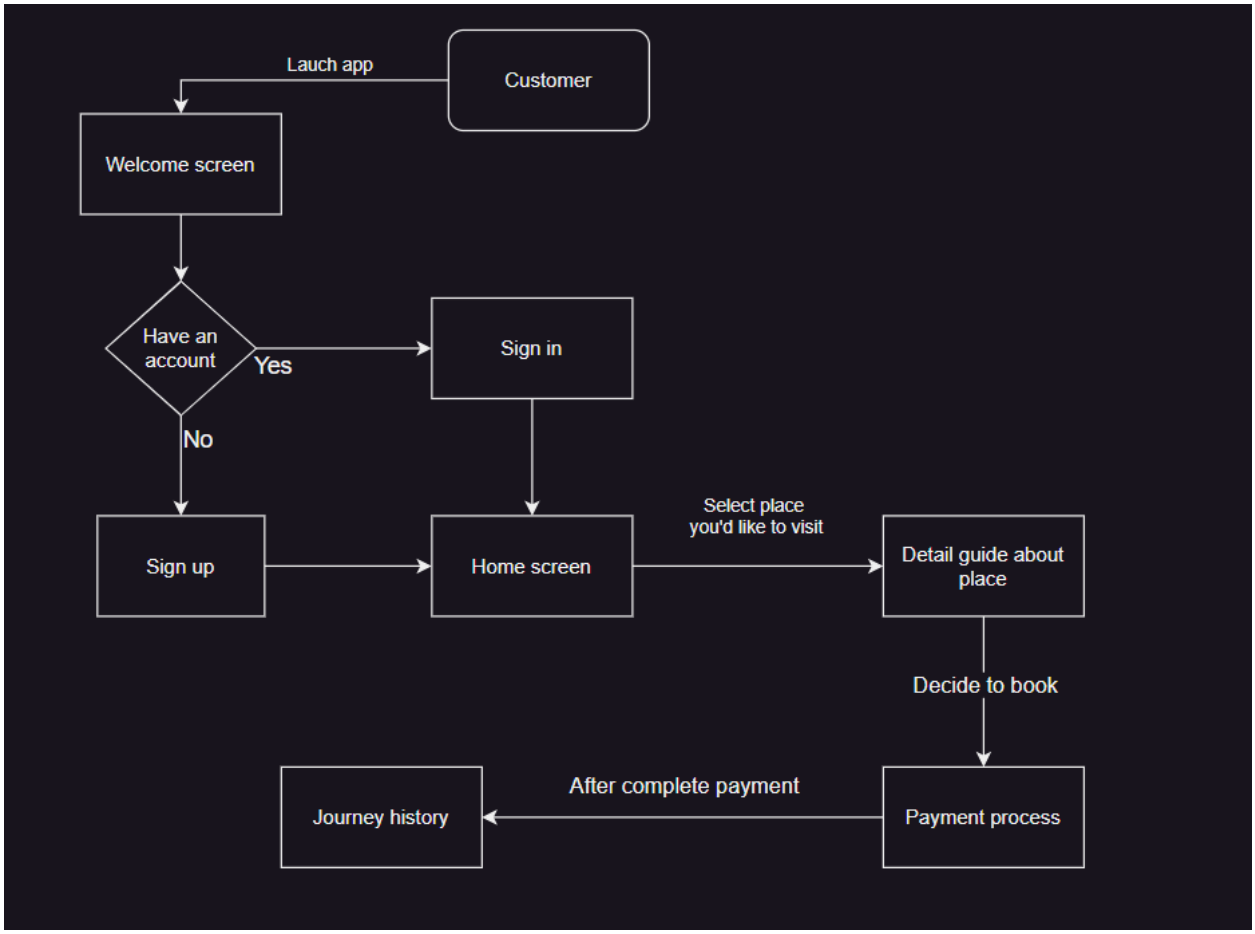

Figure 3.1: Dataflow diagram of Trip planner project.

According to the diagram, when the customer lauch this app, the Welcome screen will show up, let the customer choose if they want to sign in or sign up. If this is the first time customer use the app, they have to create an account to access the system.

After the logged in or signed up, the app will lead to Home screen where suggest a lot of interesting places around the world, with a full trip guide in it.

Then if the customer choose the trip, they will go to the Payment screen which they have to provide information and confirm the booking.

Finally when the booking is confirmed, the screen will change to Journey history, where they can see all of the trip they had booked.

3.2: User guide with illustrations

When the customer lauch the application, the first thing shows up will be the Welcome screen:

Figure 3.2.1: The Welcome screen.

As you can see, there are two options in this screen, “SIGN UP” and “OR SIGN IN BY EXISTING ACCOUNT”. If this is the first time using the app, the customer will have to create an account, which mean they have to choose the “SIGN UP” option, for the customer who has already had an account, they will choose another option.

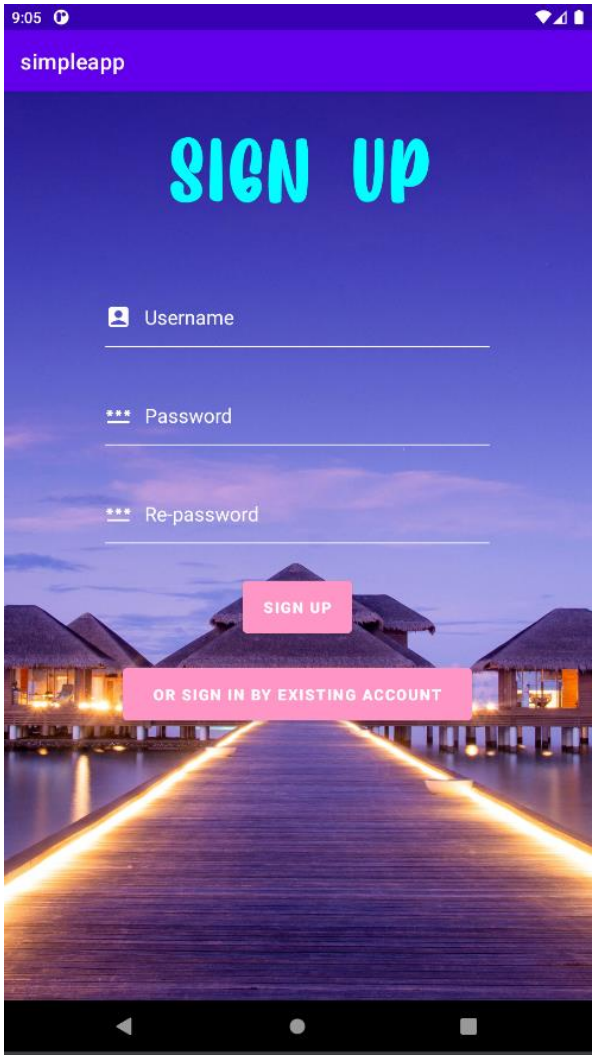

Figure 3.2.2: Sign up screen.

If you tap the “SIGN UP” button on the welcome screen, figure 3.2.2 is what you will see. There are 3 field that you need to provide, first is username, should be unique, and the password, after typed password you need to type one more in “Re-password” field, when you have finished just tap the button “SIGN UP”, the system will read and write your information into database. Below is some errors when you didn’t input valid information.

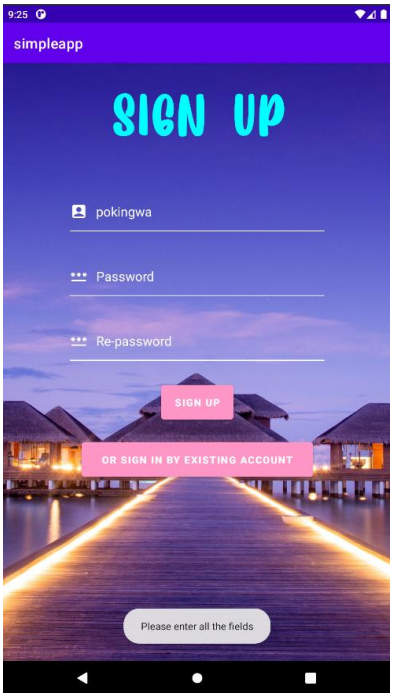

Figure 3.2.3: Error notification when User didn't input "Password" and "Re-password" fields.

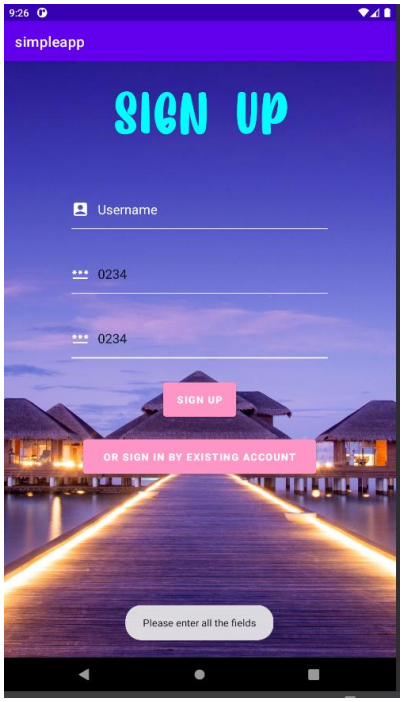

Figure 3.2.4: Error notification when User didn't input "Username" field.

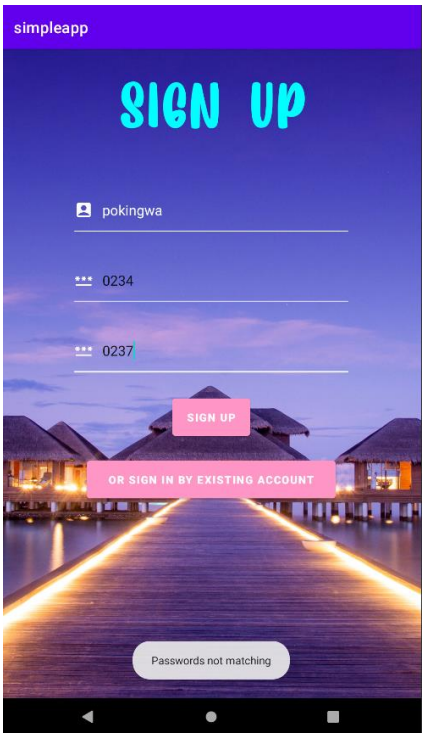

Figure 3.2.5: Error notification when User didn't input "Re-password" field matching "Password" field.

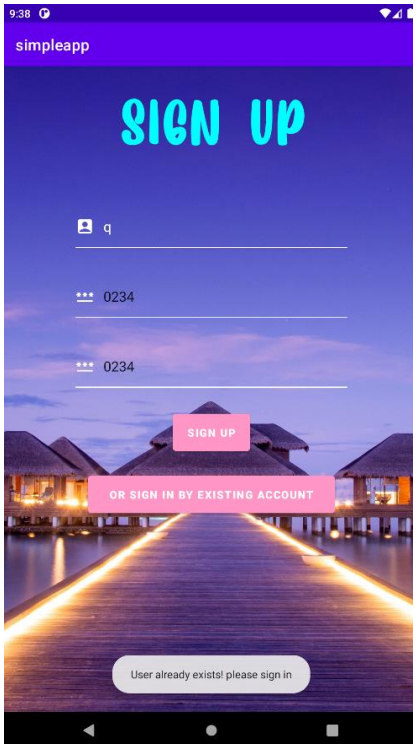

Figure 3.2.6: Error notification when User input the username that already had in the system.

On the other hand, if customer already had an account, they should choose the “OR SIGN IN BY EXISTING ACCOUNT” option. Here is the Sign in screen looks like

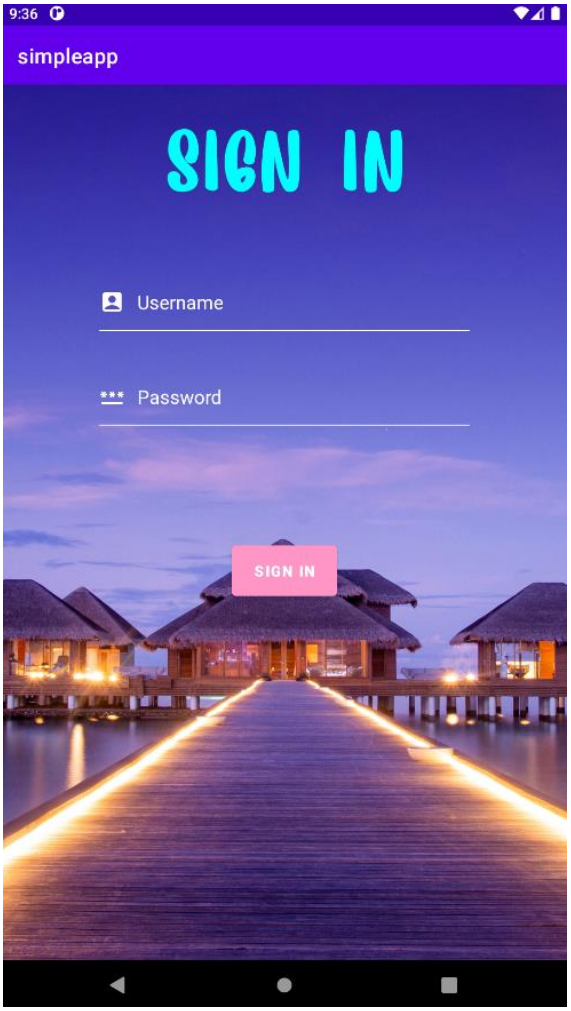

Figure 3.2.7: The Sign in screen.

If the system have thier account, then they just need to input the correct information into those two fields. Below is the notification when they didn’t input correctly or leave it blank.

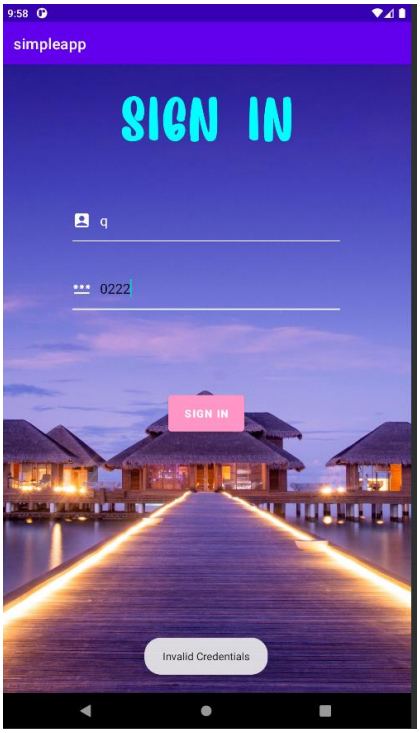

Figure 3.2.8: Error notification when user didn't input correctly account's Information.

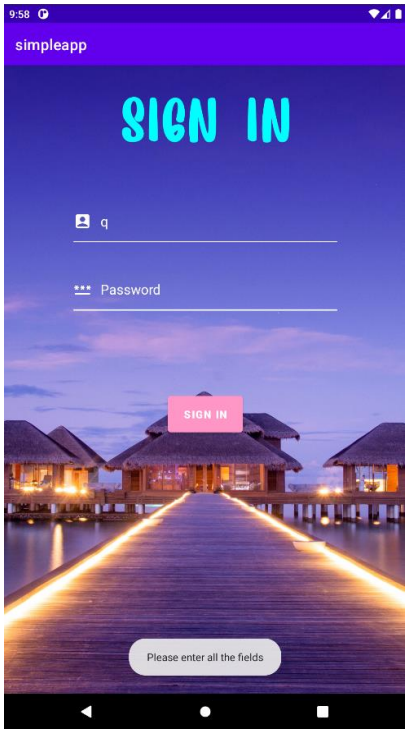

Figure 3.2.9: Error notification when user leave black one of the fields.

After finished create account or logged in, there are going to have a notification that the sign in or sign up process are successful and lead directly to the Home screen.

Figure 3.2.10: Notification after signed in.

Figure 3.2.11: Notification after signed up.

And the Home screen, provides many places that very interesting to explore, delecious food, fascinating activities, and much more.

Figure 3.2.12: Home screen with horizontal slide view of “Popular” tag.

Figure 3.2.13: Home screen with vertical scroll view in “Recently booked” tag.

In this screen, “Popular” tag will show places that visited, booked, higest rated from all around the world.

And the “Recently booked” tag shows trips that has just recently booked from other users.

When they tap on a card, it will lead to Details screen, provides information about the place they are interested about.

Figure 3.2.14: The Details screen

When they decide to book the trip, they will see the Payment screen.

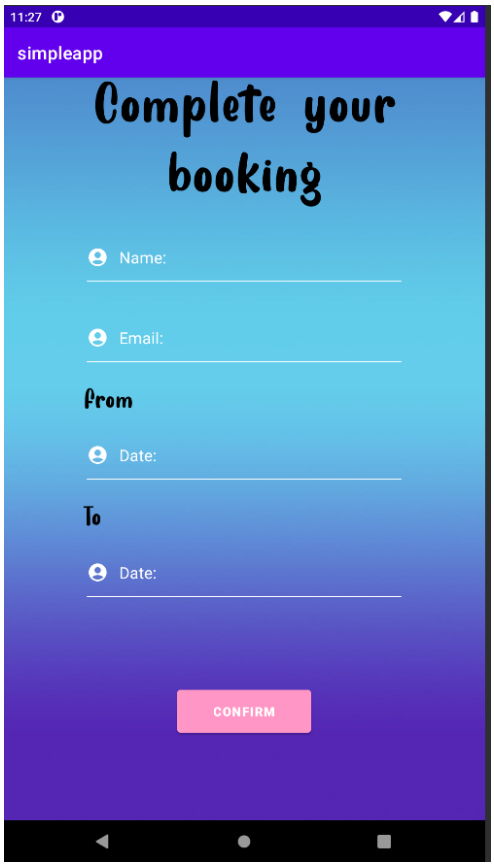

Figure 3.2.15: The Payment screen

They need to provide every information in this screen, name of the customer, email, booking trip from date to date. Finally, after confirmed payment, the screen will change to the Journey Diary with a nofitication that booked successfully, which is the history of trips they had booked.

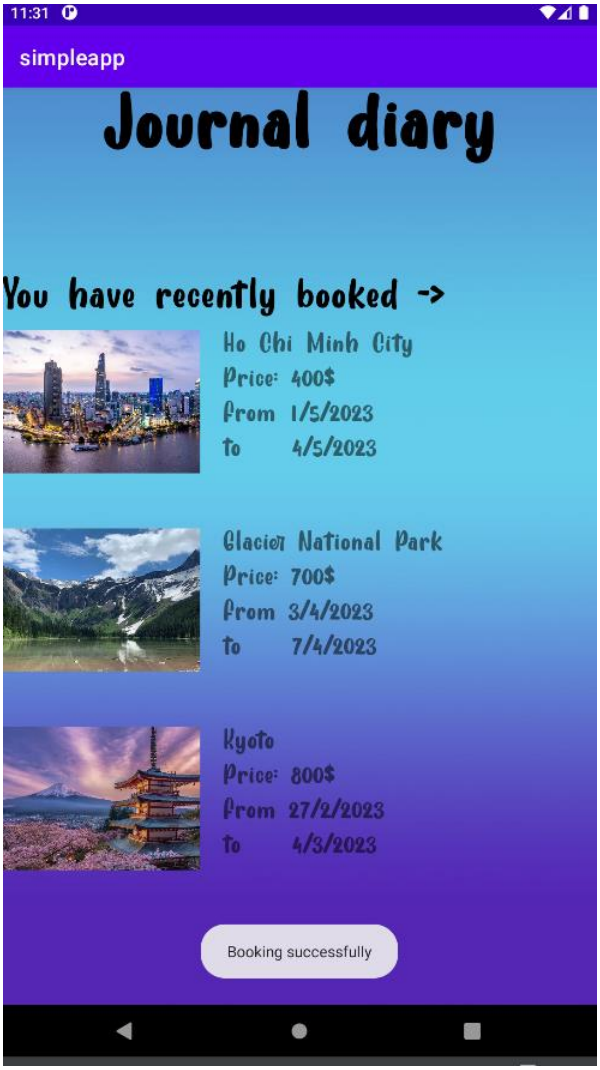

Figure 3.2.16: The Journey Diary screen.

Chapter 4: Conclusion and comments

4.1: Conclusion

In summary, this application will help you choose and enjoy your trips wisely, by showing you the whole picture of your destination.

4.2: Comment

For the last 2 months, I have learned how to create an application with Android Studio, cennect with database by using SQLite, and increased my experiences in programming with Java language.

Advantages of the project:

- User-friendly.
- Suitable for most of people.
- Easy to use.

Disadvantages of the project:

- Too simple.
- Doesn’t has navigations like bottom navigation and navigation drawer.
- Demo is uncompleted, still contains errors.
- The details of cards still very poor, doesn’t has much information.
- Cannot connect the details like pictures of place, description, … to database, have to do it manually.
- Doesn’t have verification.

References

Android Studio:

https://developer.android.com/studio?gclid=CjwKCAjwhJukBhBPEiwAniIcNQ8IrMparr2B_GLvX6CcBi_mhj4YpfAyFNpLPKsC6Ub45n-vyOWmrxoCIK8QAvD_BwE&gclsrc=aw.ds

Java programming language

https://en.wikipedia.org/wiki/Java_(programming_language)

https://www.w3schools.com/java/java_intro.asp

SQLIte

https://www.sqlite.org/index.html