|  |  |
| --- | --- |
| A picture of a winding road and trees  Coursework  Mobile Application Design & Development | Tran Dinh Minh  ID UOG: 001197133  ID Student: GCD18633  22 November 2021  COMP1786  Greenwich University |

Table of Contents

[I. Hybrid App (React-Native): 1](#_Toc87615501)

[1. Table of features list: 1](#_Toc87615502)

[2. Bugs and weaknesses: 1](#_Toc87615503)

[3. Strengths: 1](#_Toc87615504)

[4. Screen shots demonstrating each of the features: 2](#_Toc87615505)

[5. Evaluation: 23](#_Toc87615506)

[5.1. Human Computer Interaction (HCI): 23](#_Toc87615507)

[5.2. Security: 24](#_Toc87615508)

[5.3. Maintainability: 24](#_Toc87615509)

[5.4. Improve the app in the future: 25](#_Toc87615510)

[II. Android Native App: 27](#_Toc87615511)

[1. Table of features list: 27](#_Toc87615512)

[2. Bugs and weaknesses: 28](#_Toc87615513)

[3. Strengths: 28](#_Toc87615514)

[4. Screen shots demonstrating each of the features: 28](#_Toc87615515)

[5. Evaluation: 35](#_Toc87615516)

[5.1. Human Computer Interaction (HCI): 35](#_Toc87615517)

[5.2. Maintainability: 36](#_Toc87615518)

[5.3. Improve the app in the future: 36](#_Toc87615519)

[III. Conclusion: 37](#_Toc87615520)

Table of Tables

[Table 1: Features list React-Native. 1](#_Toc87615526)

[Table 2: Features list Android. 27](#_Toc87615527)

Table of Figures

[Figure 1: Login and register page. 3](#_Toc87615531)

[Figure 2: Validation form register. 4](file:///D:\ASM%20Greenwich%20university\Top_Up\Mobile%20Application%20Design%20&%20Develop\COMP1786_CW_TranDinhMinh_GCD18633.docx#_Toc87615532)

[Figure 3: Create account and login. 5](file:///D:\ASM%20Greenwich%20university\Top_Up\Mobile%20Application%20Design%20&%20Develop\COMP1786_CW_TranDinhMinh_GCD18633.docx#_Toc87615533)

[Figure 4: View data page. 6](#_Toc87615534)

[Figure 5: Input screen. 7](#_Toc87615535)

[Figure 6: Validation all required field. 8](#_Toc87615536)

[Figure 7: Validation field required. 9](file:///D:\ASM%20Greenwich%20university\Top_Up\Mobile%20Application%20Design%20&%20Develop\COMP1786_CW_TranDinhMinh_GCD18633.docx#_Toc87615537)

[Figure 8: Select time date. 10](#_Toc87615538)

[Figure 9: Price validation required. 10](#_Toc87615539)

[Figure 10: Select monthly rent price field with number. 11](#_Toc87615540)

[Figure 11: Validation price value 0 and a negative number. 11](#_Toc87615541)

[Figure 12: Validation price greater than 10000. 11](file:///D:\ASM%20Greenwich%20university\Top_Up\Mobile%20Application%20Design%20&%20Develop\COMP1786_CW_TranDinhMinh_GCD18633.docx#_Toc87615542)

[Figure 13: Validation Notes. 12](#_Toc87615543)

[Figure 14: Validation required name of reporter. 12](#_Toc87615544)

[Figure 15: Validation required name of reporter. 13](file:///D:\ASM%20Greenwich%20university\Top_Up\Mobile%20Application%20Design%20&%20Develop\COMP1786_CW_TranDinhMinh_GCD18633.docx#_Toc87615545)

[Figure 16: Confirm data. 14](#_Toc87615546)

[Figure 17: View and delete data. 15](file:///D:\ASM%20Greenwich%20university\Top_Up\Mobile%20Application%20Design%20&%20Develop\COMP1786_CW_TranDinhMinh_GCD18633.docx#_Toc87615547)

[Figure 18: Check duplicate in database. 16](file:///D:\ASM%20Greenwich%20university\Top_Up\Mobile%20Application%20Design%20&%20Develop\COMP1786_CW_TranDinhMinh_GCD18633.docx#_Toc87615548)

[Figure 19: Search by Property type. 17](#_Toc87615549)

[Figure 20: Search by Bedrooms. 18](#_Toc87615550)

[Figure 21: Search by Name of the reporter. 19](#_Toc87615551)

[Figure 22: Add notes. 20](file:///D:\ASM%20Greenwich%20university\Top_Up\Mobile%20Application%20Design%20&%20Develop\COMP1786_CW_TranDinhMinh_GCD18633.docx#_Toc87615552)

[Figure 23: Add note successful. 21](#_Toc87615553)

[Figure 24: Sign out. 22](file:///D:\ASM%20Greenwich%20university\Top_Up\Mobile%20Application%20Design%20&%20Develop\COMP1786_CW_TranDinhMinh_GCD18633.docx#_Toc87615554)

[Figure 25: Impressions lead to interactive curiosity. 23](#_Toc87615555)

[Figure 26: Symbol meaning. 23](#_Toc87615556)

[Figure 27: Easy-to-understand validation message. 24](#_Toc87615557)

[Figure 28: File clear. 25](#_Toc87615558)

[Figure 29: Improve map. 26](#_Toc87615559)

[Figure 30: Improve UI. 27](#_Toc87615560)

[Figure 31: Input screen. 29](#_Toc87615561)

[Figure 32: Validation all required field. 30](#_Toc87615562)

[Figure 33: Validation required field. 31](file:///D:\ASM%20Greenwich%20university\Top_Up\Mobile%20Application%20Design%20&%20Develop\COMP1786_CW_TranDinhMinh_GCD18633.docx#_Toc87615563)

[Figure 34: Date time. 32](file:///D:\ASM%20Greenwich%20university\Top_Up\Mobile%20Application%20Design%20&%20Develop\COMP1786_CW_TranDinhMinh_GCD18633.docx#_Toc87615564)

[Figure 35: Monthly rent price validation. 32](#_Toc87615565)

[Figure 36: Validation price. 33](#_Toc87615566)

[Figure 37: Validation Notes. 33](#_Toc87615567)

[Figure 38: Validation Name. 34](file:///D:\ASM%20Greenwich%20university\Top_Up\Mobile%20Application%20Design%20&%20Develop\COMP1786_CW_TranDinhMinh_GCD18633.docx#_Toc87615568)

[Figure 39: Validation successful. 34](file:///D:\ASM%20Greenwich%20university\Top_Up\Mobile%20Application%20Design%20&%20Develop\COMP1786_CW_TranDinhMinh_GCD18633.docx#_Toc87615569)

[Figure 40: Easy-to-understand validation message. 35](#_Toc87615570)

[Figure 41: Impressions lead to interactive curiosity. 35](#_Toc87615571)

[Figure 42: File clear. 36](#_Toc87615572)

[Figure 43: Connect SQLite. 37](#_Toc87615573)

# I. Hybrid App (React-Native):

In this coursework, I decided to use React-Native to design and develop the Rental Apartments Finder (RentalZ) application. The reason I use React-Native is because it can work almost like a native app built on either iOS or Android platforms. Moreover, React-Native is also supported by the ReactJS library to make it easy for me to design both front end and backend of mobile apps. The application will have features from a to g based on the task required by the coursework.

## 1. Table of features list:

|  |  |
| --- | --- |
| Feature | Implementation |
| Design app screens - Basic details input screen | Fully implemented |
| Implement forms validation | Fully implemented |
| Store, view and delete the basic details and check for duplicate events | Fully implemented |
| Search | Implemented and the application can search by property type, bedrooms, and name of the reporter. |
| Add a note input screen | Implemented, I have created user input according to previously entered Notes field and data will be stored if user update according to their old Notes field. |
| Add additional features | I have developed more registration function, login before entering the application, they can also log out of the application. |

Table 1: Features list React-Native.

## 2. Bugs and weaknesses:

* After creating more user data to go to the data view page, sometimes it can't be reloaded immediately, but it must wait about 5 seconds, making the user experience less than optimal.
* The application interface is quite simple.

## 3. Strengths:

Firebase Authentication:

* I use Firebase Authentication service with user account created function with email and password with end-to-end identity solution. Firebase Authentication makes it easy for me to authenticate and create an account on the fly without having to create an API. Firebase Authentication is also supported with automated validations with comprehensive security and easy cross-platform login.

Cloud Firestone:

* The RentalZ application will use the NoSQL Cloud Firestore database in conjunction with React-Native. Cloud Firestore is considered as the perfect match for mobile app development, it doesn't need a complete server to create API and use extra database to prototype or build MVP for app just create a collection database that is supported by google and stored in the cloud. Furthermore, Cloud Firestore is offline for both Apple and Android clients. It is worth noting that my RentalZ application only needs one collection of Users so Cloud Firestore will make adding, modifying, deleting functions in the database easy and not difficult like other databases.

The data input fields are designed for user with 4 input types: Dropdown, Text Input, Currency Input, and Datetime picker.

* Dropdown: It is used for Property Type, Bedrooms, Furniture field so that users can easily choose their options.
* Text Input: It is used for Notes and Reporter name.
* Currency Input: It is used for Monthly Rent input. Its purpose is only for the user to enter numbers and the user will not be allowed to type in letters.
* Date and time picker: Its purpose is to make UI more beautiful with picker that matches the design.

## 4. Screen shots demonstrating each of the features:

First, the registration and login page will display. User can login account if user have created before, if user has not created account before they must enter email, password and, click “Register” to create account.

Graphical user interface, application

Description automatically generated

Figure 1: Login and register page.

Graphical user interface, text, application, chat or text message

Description automatically generatedEmail and password are also validated if they have not entered the data but clicking “Register” it will display a message that the request does enter data. Great that the validation feature is supported by Firebase Authentication.

Graphical user interface, text, application, chat or text message

Description automatically generated

Figure 2: Validation form register.

Graphical user interface, application

Description automatically generatedGraphical user interface, text, application, chat or text message

Description automatically generatedIf the user has passed the validations and created the account successfully, the user clicks “Login” to enter the RentalZ application.

Figure 3: Create account and login.

After the user successfully registers and logs in, the user's RentalZ homepage will be displayed. The homepage includes functions to sign out, create a rental listing, delete data, add notes, and search data.

Graphical user interface, application, Teams

Description automatically generated

Figure 4: View data page.

Next, the user clicks “CREATE A RENTAL LISTING” they will go to the application data generation form page. This is the form of RentalZ application, if the user enters data and passes the validations, the data will be saved to firebase. RentalZ application provides a form with 7 specific fields Property type, Bedrooms, Date Time, Monthly rent price, Furniture types, Notes, Name of the reporter which the user must enter. In addition, Furniture types and Notes field are not required to enter, and the remaining fields require the user to enter data.

Graphical user interface, application

Description automatically generated

Figure 5: Input screen.

First, if all fields are not entered, the application will display a message asking the user to enter all data.

Graphical user interface, text, application

Description automatically generated

Figure 6: Validation all required field.

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generatedAfter selecting the property type field and clicking submit, if the BedRooms field has not been selected, it will ask the user to choose, continue if the Date Time field is not selected, the application will also notify the user to enter the Date Time data.

Figure 7: Validation field required.

In addition, the Date Time field is only allowed for the user to select data from the present to the past, not for the user to select data larger than the current time.

Calendar

Description automatically generated

Figure 8: Select time date.

Next, the Monthly rent price field is required to enter the data and only allow the user to enter numeric data.

Graphical user interface, text, application

Description automatically generated

Figure 9: Price validation required.

Calendar

Description automatically generated

Figure 10: Select monthly rent price field with number.

A picture containing text, saw, tool

Description automatically generatedIn the Monthly rent price field, the user cannot enter a value 0 and a negative number.

Graphical user interface, text, application

Description automatically generated

Figure 11: Validation price value 0 and a negative number.

Graphical user interface, text, application

Description automatically generatedFurthermore, the user cannot enter a value greater than 10000.

*Table

Description automatically generated*

A picture containing graphical user interface

Description automatically generated

Figure 12: Validation price greater than 10000.

Next, the Furniture types and Notes field is optional, but I also validate the Notes field and only allow users to enter less than 30 characters.

Graphical user interface, text, application

Description automatically generated

Figure 13: Validation Notes.

The last, name of the reporter field is also a required field for the user to enter data. It only allows users to enter from 3 characters and less than 20 characters.

Graphical user interface, application

Description automatically generated

Figure 14: Validation required name of reporter.

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Figure 15: Validation required name of reporter.

After the user passes all the data fields, the user will click the “Submit” button and the application will show all the data to the user again. If the user wants to change any field they want, they will click “No” to update the data again. If the user has agreed with the data they have entered, they will click “Yes” to enter the data into the app.

Graphical user interface, text, application

Description automatically generated

Figure 16: Confirm data.

After the user has successfully created the form, the application will redirect back to the data view page. If user wants to delete data, user just need to click “DELETE” button and the application will notify user that the data has been deleted successfully.

Graphical user interface, text, application, Teams

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Figure 17: View and delete data.

Graphical user interface, application

Description automatically generatedGraphical user interface, text, application

Description automatically generatedIn addition, the application also has check for duplicate events in the database, if the user data enters all the data that already exists in the database before, they will not be able to create another field like that because the application check all duplicate data, user can only enter new data then new data can be added to the app.

Figure 18: Check duplicate in database.

Users can also search data by entering the information based on the three fields Property type, Bedrooms, and Name of the reporter.

Graphical user interface, text, application

Description automatically generated

Figure 19: Search by Property type.

Graphical user interface, text, application

Description automatically generated

Figure 20: Search by Bedrooms.

Graphical user interface, text, application, Teams

Description automatically generated

Figure 21: Search by Name of the reporter.

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generatedNext, users can overwrite the old note, they just need to click “ADD NOTES MORE”. The application will go to the add note page and will display the old data for the user to enter. Note, the note field is only 50 characters maximum.

Figure 22: Add notes.

Graphical user interface, text, application

Description automatically generated

Figure 23: Add note successful.

Graphical user interface, application

Description automatically generatedGraphical user interface, application, Teams

Description automatically generatedFinally, when the user has done all the functions of RentalZ application, they will click “SIGN OUT” to perform the logout function to exit the application.

Figure 24: Sign out.

## 5. Evaluation:

### 5.1. Human Computer Interaction (HCI):

User-friendly experience is always a difficult question and should be solved with any mobile app. Therefore, I have always put human-computer interaction at the heart of RentalZ application development. Here are some points that aim to be user-friendly:

**Impressions lead to interactive curiosity:**

* Any mobile application needs to have noticeable highlights. Therefore, I used outstanding color and capital letters to accentuate the user experience. The goal is to help users impress functions through special colors and avoid functions that are overlooked by factors that do not attract attention. With impressive colors, users will be more curious and interactive with the application.

Graphical user interface, text, application

Description automatically generated

Figure 25: Impressions lead to interactive curiosity.

**Symbol meaning:**

* In the search form in the data view page, I used the search and cancel icons to make it easy for users to see what the purpose of this event is without them having to read the text "Search by".



Figure 26: Symbol meaning.

**Easy-to-understand validation message:**

* For any data form a validation form is required. Therefore, in my RentalZ app I have put the validation factor on top with each field having a different validation message. Messages are validated with clear content to help users not be confused because there are some applications with confusing feedback validation content.

Graphical user interface, text, application

Description automatically generated

Figure 27: Easy-to-understand validation message.

### 5.2. Security:

Security is an essential and imperative issue for RentalZ app. I have also created registration and login functionality to go with validation forms to give the app a professional look. Each account will be created only once, and strangers will not be able to know the created user account. However, my app still doesn't have the permissions section and I will improve it in the future. Furthermore, I also needed to add a confirm password field for the old password. With these factors will help RentalZ complete and secure in managing user accounts.

### 5.3. Maintainability:

Maintenance is a necessity for any future application. Therefore, I have also divided the folders into separate pages, for the firebase connection page I will use the database folder, the pages with the search functions, add rental list, view data, and so on, it will be in the folder Screens. This is necessary because it makes it easier for me to maintain in the future if functions may arise or fail, I can find the functions in any file and easily improve them that function. Furthermore, dividing files into the correct folders avoids unnecessary risks to the RentalZ application.

Graphical user interface, application

Description automatically generated

Figure 28: File clear.

### 5.4. Improve the app in the future:

In the future, I will develop more map access feature with GPS. This is a function that helps users easily find the location of the apartment they want to rent because the psychology of tourists now is that they want to have many services around their apartment such as supermarkets, restaurants, water shop, and so on. This will bring a lot of profit to the business in the future.

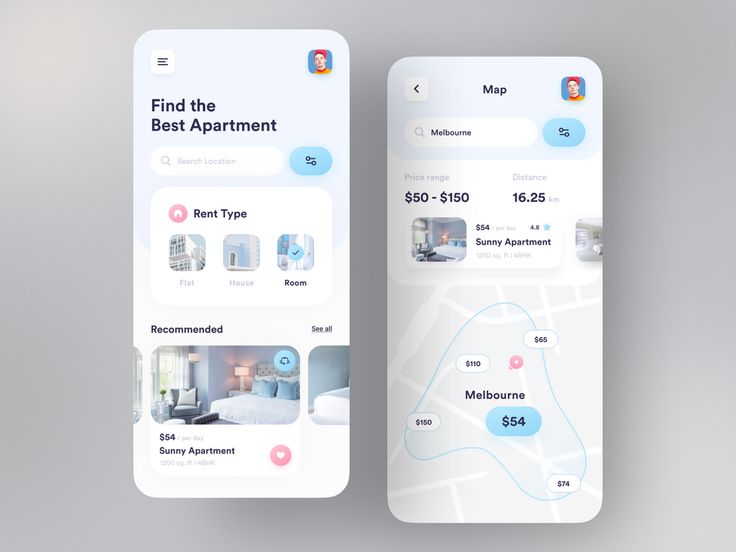


Figure 29: Improve map.

Source: [https://i.pinimg.com/736x/61/17/fa/6117fa3813d6666e509bfa5823ac66ab.jpg]

Moreover, I also need to improve the user interface to give them the best experience. The interfaces to login, register, view data, create data will also be explored and updated by me in the future, with the aim of bringing RentalZ success.

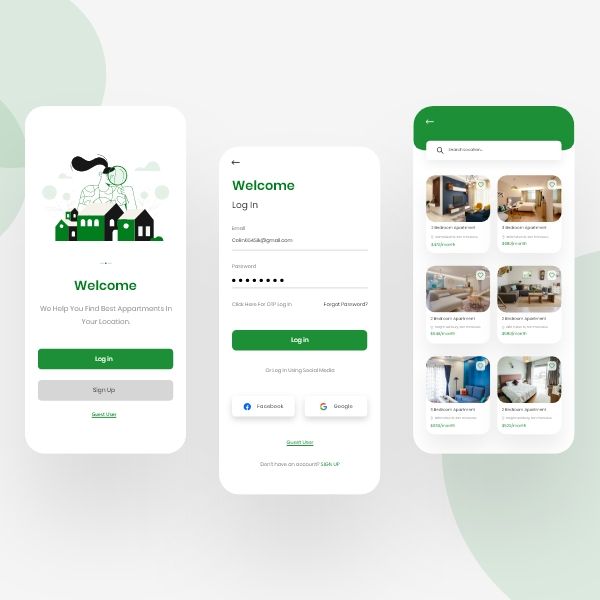


Figure 30: Improve UI.

Source: [https://i.pinimg.com/736x/fe/3d/72/fe3d72c26669df0c1799ac0ec605f302.jpg]

# II. Android Native App:

In this section, I created RentalZ app with native Android app version. With simpler functions and only implements the 2 required features a) Design app screens - Basic details input screen and b) Implement forms validation.

## 1. Table of features list:

|  |  |
| --- | --- |
| Feature | Implementation |
| Design app screens - Basic details input screen | Fully implemented |
| Implement forms validation | Fully implemented |
| Add additional features | No additional features implemented |

Table 2: Features list Android.

## 2. Bugs and weaknesses:

* My implementation functions are only defined in one MainActivity class.
* The application interface is quite simple.
* The data has not been stored in the database.

## 3. Strengths:

The data input fields are designed for user with 4 input types: Radio Button, Text Input, Currency Input, and Datetime picker.

* Radio Button: It is used for Property Type, Bedrooms, Furniture field so that users can easily choose their options.
* Text Input: It is used for Notes and Reporter name.
* Currency Input: It is used for Monthly Rent input. Its purpose is only for the user to enter numbers and the user will not be allowed to type in letters.
* Date and time picker: Its purpose is to make UI more beautiful with Android picker that matches the design of Samsung, Oppo, and so on phones.

Moreover, I also carefully validate the form with appropriate messages for each field.

## 4. Screen shots demonstrating each of the features:

RentalZ application provides a form with 7 specific fields Property type, Bedrooms, Date Time, Monthly rent price, Furniture types, Notes, Name of the reporter which the user must enter. In addition, Furniture types and Notes field are not required to enter, the remaining fields require the user to enter data.

Graphical user interface, text, application, email

Description automatically generated

Figure 31: Input screen.

First, if all fields are not entered, the application will display a message asking the user to enter all data.

Graphical user interface, text, application, email

Description automatically generated

Figure 32: Validation all required field.

Graphical user interface, text, application, email

Description automatically generatedAfter selecting the property type field and clicking submit, if the BedRooms field has not been selected, it will ask the user to choose, continue if the Date Time field is not selected, the application will also notify the user to enter the Date Time data.

Graphical user interface, text, application, email

Description automatically generated

Figure 33: Validation required field.

In addition, the Date Time field is only allowed for the user to enter data from the present to the past, not for the user to enter data larger than the current time

Calendar

Description automatically generatedA screenshot of a clock

Description automatically generated with medium confidence

Figure 34: Date time.

Next, the Monthly rent price field is required to enter the data and only allow the user to enter numeric data.

Calendar

Description automatically generated

Figure 35: Monthly rent price validation.

In the Monthly rent price field, the user cannot enter a value 0 and a negative number. Furthermore, the user cannot enter a value greater than 10000.

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Figure 36: Validation price.

Next, furniture types and Notes field is optional, but I also validate the Notes field and only allow users to enter less than 30 characters.

Graphical user interface, text, application, chat or text message, email

Description automatically generated

Figure 37: Validation Notes.

The last, name of the reporter field is also a required field for the user to enter data and only allow users to enter less than 20 characters.

Graphical user interface, text, application, email, website

Description automatically generatedGraphical user interface, text, application, chat or text message

Description automatically generated

Figure 38: Validation Name.

Graphical user interface, application

Description automatically generatedGraphical user interface, text, application, email

Description automatically generatedIf the user has entered all the data and passed the validation, the user only needs to click submit to use the application.

Figure 39: Validation successful.

## 5. Evaluation:

### 5.1. Human Computer Interaction (HCI):

In the process of designing the interaction of the RentalZ application with the Android version, I also put HCI as a top priority with the following design principles:

**Easy-to-understand validation message:**

* During the data entry process, if the user enters inappropriate data, the system will display the validations with appropriate warning messages to the user. With the goal of the application getting the right data for each field data.

Calendar

Description automatically generated

Figure 40: Easy-to-understand validation message.

**Impressions lead to interactive curiosity:**

* In this version of Android, I have also used highlight colors and capital letters to highlight the user experience. The aim is to help users impress functions through distinctive colors and avoid functions that are overlooked by unattractive elements. With impressive colors, users will be curious and interact with the application more.



Figure 41: Impressions lead to interactive curiosity.

### 5.2. Maintainability:

I have also explicitly split the files between the client and the server into separate files. This is necessary because it makes it easier for me to maintain in the future if functions may arise or fail, I can find functions in any file and easily improve it. Furthermore, dividing files into correct folders avoids unnecessary risks to the RentalZ application.

Graphical user interface, text

Description automatically generated

Figure 42: File clear.

### 5.3. Improve the app in the future:

In the future, the RentalZ Android app will be used by me with a database connection for the purpose of saving user data. I think I will use SQLite to design the database for this release.

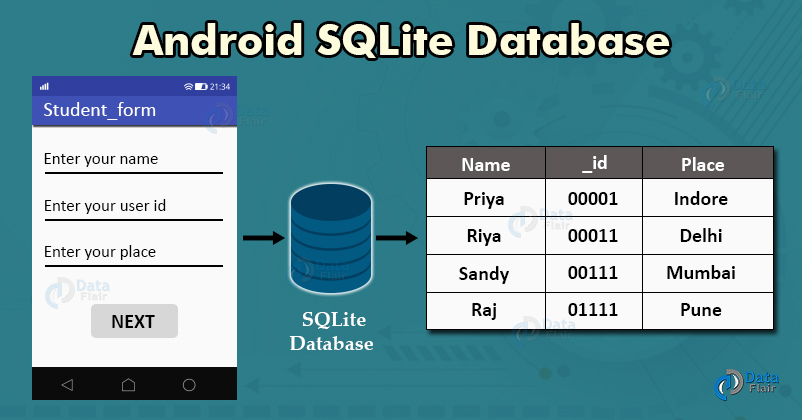


Figure 43: Connect SQLite.

Source: [https://data-flair.training/blogs/wp-content/uploads/sites/2/2020/05/Android-SQLite-Database.jpg]

Moreover, I will also decentralize with two permissions admin and user, only user can create data and admin will view user's data. The purpose is to help increase the security of the system.

# III. Conclusion:

In this coursework, I have designed RentalZ application with 2 different versions. Although the process was not long, I overcame the pressure to finish the course with the functions required by the assignment. I learned many new technologies like firebase, react-native, and some libraries. Finally, I would like to thank teacher Phan Thanh Tra for enthusiastically helping me in the process so that I could successfully complete this subject.