**UECS3253/UECS3463 WIRELESS APPLICATION**

**DEVELOPMENT**

**MAY 2022**

**Group Info**

| Name | Student ID | Practical Group |
| --- | --- | --- |
| Kang Shu Yi | 1903794 | 2 |
| Lim Yen Khai | 2005211 | 1 |
| Loke Yee Ching | 2004231 | 2 |
| Wong Jing Jun | 2004189 | 2 |
| Wong Sheng Ann | 1903376 | 2 |

**Marks breakdown**

**Marks Allocation**

| **PART** | | **Criteria** | **Marks Allocation** |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |
| A | 1 | Screens & Layout | **5** |  |  |
| 2 | Use of Appropriate Components | **5** |  |  |
| 3 | Reusability (Custom Components)  inserts/updates | **5** |  |  |
| 4 | Implementation of React-Navigation: Stack, Drawer, Tab Navigation | **15** |  |  |
| **TOTAL** | | | **30** |  |  |
| B | 1 | Appropriate Usage of Data  Persistence | **5** |  |  |
| 2 | Implementation of Data Persistence | **10** |  |  |
| 3 | CRUD Operations & input validation | **15** |  |  |
| **TOTAL** | | | **30** |  |  |
| C | 3 | Appropriate Usage of Cloud  Connectivity | **15** |  |  |
| 4 | Implementation of Cloud  Connectivity | **15** |  |  |
| 5 | Presentation | **10** |  |  |
| **TOTAL** | | | **40** |  |  |

**Table of Contents**

| No | Title | Page Number |
| --- | --- | --- |
| 1 | Introduction |  |
| 2 | Part A |  |
| 3 | Part B |  |
| 4 | Part C |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 |  |  |
| 11 | References |  |

*\*Add any contents if necessary*

**1.0 List of key features of Karotsell**

1. Sign up, login and logout to Karotsell
2. View product
3. Search product
4. Purchase product
5. Edit user profile
6. Add new product
7. Delete product

**1.1 Sign up, login and logout to Karotsell**

When a user first opens the app, based on the previously saved login state, the user will either be directed to the login page or directly to the home page. If the user is not logged in, he/she will be greeted with a login form asking for an email address and password, with a “login” button to allow them to confirm their login. If the user does not have an account, they can tap the highlighted register text at the bottom of the screen which will bring them to a registration page. After registering successfully, the user will be redirected to the login page where they can then proceed to login. Once logged in, the user can log out using the “logout” button in the profile tab, which will clear the session token and set login state to false, then redirect the user to the login page.

**1.2 View product**

In the search page or the home page, the list of products are displayed. When a user taps on the product item on the search page or the home page, it leads the user to the “View Product” page. In the “View Product” page, details of the product item will be displayed including the photo of the product item, the name, price, number of likes, condition, category, brand, delivery method, meetup location and the seller of the item. Users may view more images of the product by swiping across the photos of the product. At the bottom of the page, there is a “Buy” button to allow users to make a purchase.

**1.3 Search product**

Once the users have logged into their respective accounts, they will be able to navigate to a search screen using the bottom navigation bar. After clicking into the search page the users will be able to clearly see a search bar on the top of the screen where they can type in individual product names to search for. After users typed in the name of the products, users can click on the search button to search for the specific product from the database. The search results will be queried through the database and the results will be retrieved and displayed at the container that shows the products. After querying the product out from the database, users can click on the image of the product to be prompted to view the individual product screen whereby all information of the displayed product will be displayed.

**1.4 Purchase product**

In the home screen and search screen users will be able to click on the images of the displayed products to see more information about the product, then on the bottom of the screen there is a “Buy” button for users to click if they are interested in purchasing the product. Once the users click on the “Buy” button, the users will be brought to a confirmation page with the images of the product selected. In this page they will be prompted to enter their address. After the user enters their address they can click on the “Confirm Order” button to confirm their order.

**1.5 Edit user profile**

To edit the user’s user profile, the user needs to access the profile page and tap on the “Edit profile” button. Tapping on the button leads the user to a page which prompts the user to enter the new name, email and phone number to be updated. Once the user enters all the information, the user has to tap on the “Save Changes” button to commit the updates. After that, the user will be able to view the updated changes on the profile page.

**1.6 Add new product**

This “add new product” feature allows users to place items up for sale in the Karotsell phone application. By tapping on the “Sell” button from the bottom navigation bar, users will be prompted to enter the details of the items that they want to sell. Here the picture of the item is chosen using the “Image Picker” component. Image picker components allow users to take a picture by using the camera and the picture will be displayed. The users are required to enter the items details for listing in the sell screen. After that, the users are prompted to view the item to be sold for the final time before the user can confirm the sale of that particular item. The item will be inserted into the database once the

**1.7 Delete product**

If the user wants to delete their own product, the user can go to the profile and click on “My Product” to show their product that has been listed on the market. Then the product will be retrieved from the database by using their own userID. When the user clicks on the delete button, the product will be deleted from the database by using its productID. The “My Product” page will only display the other product that users have in the database.

**2.0 Implementation of Stack, Tab, Drawer navigators**

**2.1 Highest Level Stack Navigation Container**

The navigation container at the highest level consists of a stack navigator that leads to the bottom tab navigator or the authentication stack. The authentication stack consists of the login and register page, while the bottom tab navigator leads to the rest of the application. The user is automatically directed to either of these 2 stacks based on their login state stored in asynchronous storage.

**2.1.1 Home Stack Navigator**

Home stack consists of three screens: a home screen,, a view product screen and a purchase confirmation screen. The initial route is linked to the home screen. Whenever the user lands on the stack, they will be led to the home screen. Stack navigator is applied to transition from the search screen to the view product screen and then from the view product screen to the confirmation of purchase screen which is triggered when the user taps on the “Buy” button.

**2.1.2 Search Stack Navigator**

Search stack consists of three screens: a search screen, a view product screen and a purchase confirmation screen. The initial route is linked to the search screen. Whenever the user lands on the stack, they will be led to the search screen. Stack navigator is applied to transition from the search screen to the view product screen and then from the view product screen to the confirmation of purchase screen which is triggered when the user taps on the “Buy” button.

**2.1.3 Sell Stack Navigator**

Sell stack consists of two screens : a sell screen for uploading the item and a preview item screen for the user to preview the item before inserting it into the database. The initial route for the sell stack is the sell screen and therefore the app will redirect the user to the sell screen when the user select the sell stack at the button tab navigator. The app will redirect user to the preview screen after pressing the preview button in the sell screen.

**2.1.3 Profile Stack Navigator**

Profile stack consists of three screens: a profile screen, an edit screen and a myProduct screen. Stack navigator is applied to transition from the profile screen to the edit profile screen and from the profile screen to the myProduct screen which allows the user to delete the products.

**2.2 Bottom Tab navigator**

In Karotsell, there is a bottom tab on every page to allow users to navigate between different screens on the app. The bottom tab is a nested navigator where it consists of the Home stack, the Search stack, the Sell stack and Profile stack. A nested navigator is used to allow the nesting of multiple screens in a navigator.

**3.0 Implementation of Data Persistence in Karotsell**

**3.1 SQLite**

Karotsell operates on SQLite database. In the database, there are three tables: Users, Orders and Item. In the user table, the primary key is user\_id, along with other keys including name, email, password and phone\_num . In the Item table, the id for each item is the primary key and each item consists of attributes such as sellerID, name of item, date\_posted, condition of the item, item category, price, brand, description, delivery, meetup, the url of the item image, and also review of the product. For the order table however, the primary key is the order\_id and it’s attributes are product\_id, buyer\_id, delivery\_address, and order\_date.

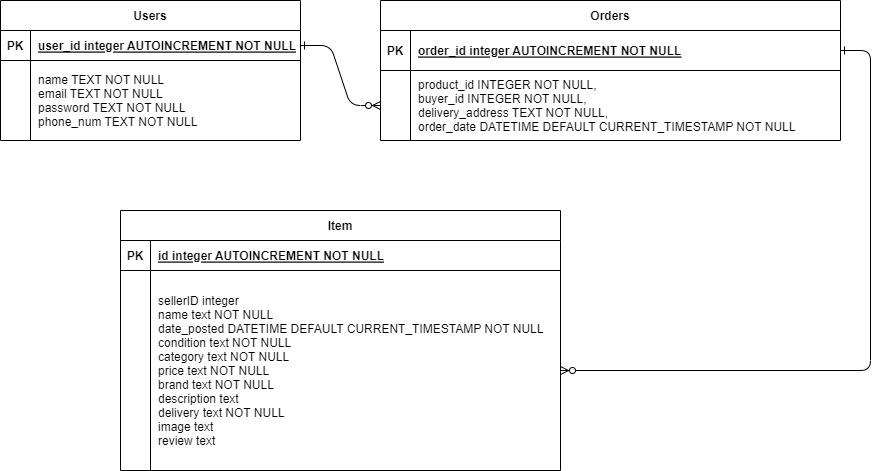


Figure 1 shows the Entity Relationship Diagram of the tables in the SQLite database for Karotsell.

**3.1.2 View item**

When the user enters the “View Item” page, data will be retrieved from the database to be displayed. Data retrieved include: name, price, date\_posted, condition, category, brand, description, delivery, meetup, image and review.

**3.1.3 Search item**

When the user enters the name of the item in the search bar, it will be passed as a parameter to the query to retrieve the item with the same name in the item table. If the item of the same name exists in the item table, it will be retrieved from the table and displayed in the search result. Else, the search result will be empty.

**3.1.4 Purchase item**

When users are in the purchase screen they will be able to see the image of the product that they are buying which is queried from the image column from the item database. After users filled in the address and clicks confirm, the application will insert the order data into the order table and remove the sold product from the item table

**3.1.5 Edit user profile**

In the edit user profile screen, users will be able to change their information such as name, email address, and phone number by getting the user id through the async token. Then, the query to update the database will be sent to the SQLite database through the flask API.

**3.1.6 Add new item**

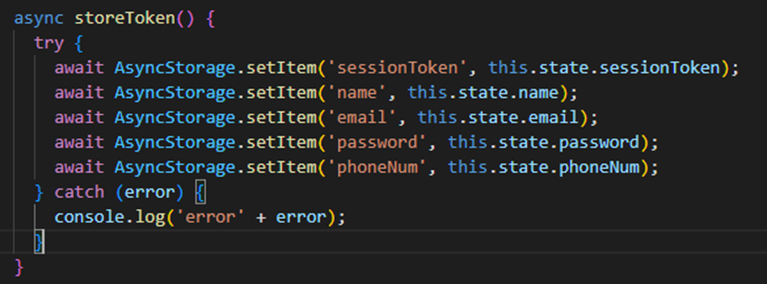
In the add new item module for sale, first of all it will prompt the user to enter information of the product to be sold on the application such as item name, price, brand, category and description. Then a preview screen will show the product information entered by the user so that users can have a final confirmation before posting the item for sale. After that, a SQL query is used to insert the new item into the database.

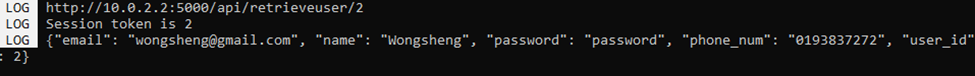
**3.1.7 Delete item**

To delete an item on sale in the application, a user would need to navigate to the “my products” screen in the “Profile” page. In this “my products” screen this page displays all the items that are put on sale by this individual user by using a select query to retrieve all items that have the user’s ID. Then, if the seller wishes to delete the item on sale the “Delete” button will prompt a SQL query that deletes the item from the database

**3.2 Asynchronous storage**

This application uses asynchronous storage as a “session handler”. By default, the sessionToken variable is set to “NONE”, and the isLoggedIn variable is set to false. On successful login, parameters of the current user obtained through the flask web server from the database is stored in asynchronous storage







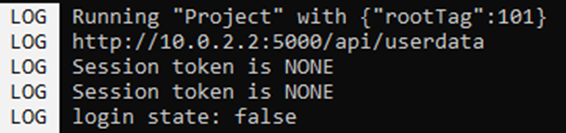
the sessionToken is initialized to the current user id and the variable isLoggedIn is set to true. To ensure users cannot navigate directly to the app without success login to the system, in the highest-level navigation container, a ternary operator was used to return either the authentication stack containing login and register page, or the tab stack that returns the rest of the app.



Since the user id is stored in the asynchronous storage upon login, all other modules of the application that require the information about the current user can obtain it from the async storage using the “getItem” function.



On logout, the sessionToken variable and the token in asynchronous storage will be set back to the default value of “NONE”, and variable isLoggedIn will be set back to false.



This will navigate the user back to the authentication stack containing the login and register pages.

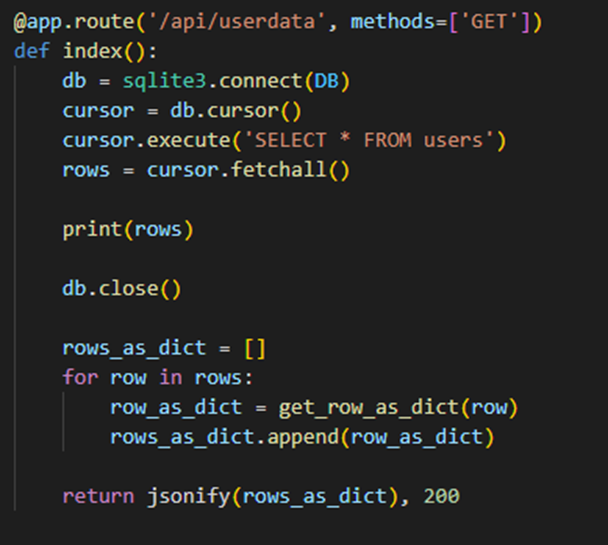
**4.0 Connection to Cloud and its Usage**

Flask Web Server (User Authentication)

This application uses a python flask web server to communicate with a sqlite database to store and authenticate user data. First, we write a python script to create the database, tables, and prepopulate it with some data. Instead of directly performing sqlite queries with react native commands, we use react native code to invoke database functions from the python flask, using GET and POST methods. This is because direct sqlite queries through sqlite will cause changes in the database locally in the app, but the changes will not be committed to the database file. Through the flask, we can perform CRUD functions on the user table that will cause changes to the sqlite file due to the python script being able to commit changes. Unfortunately due to time constraints, we were unable to implement the flask web server to commit CRUD operations for the user table, while other data is handled with direct sqlite queries.

Functions

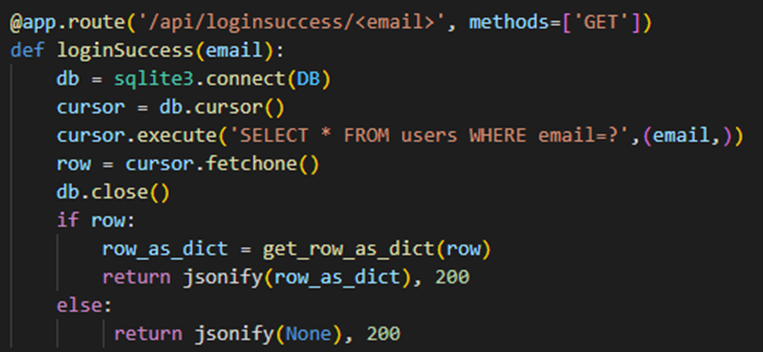
index() - retrieve all data from the user table



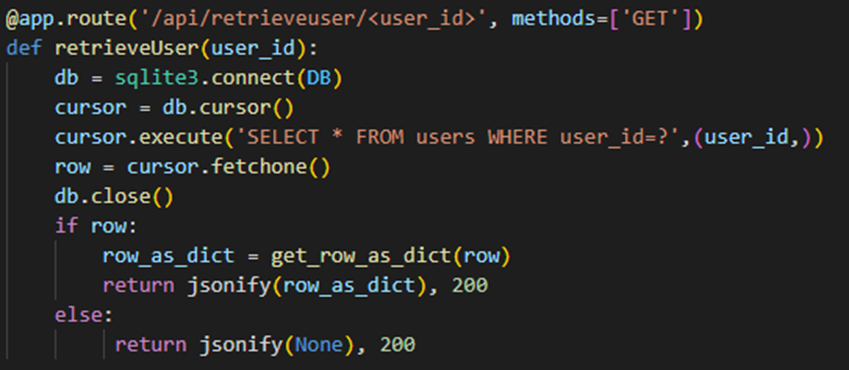
login() - verify user input with data in the sqlite database



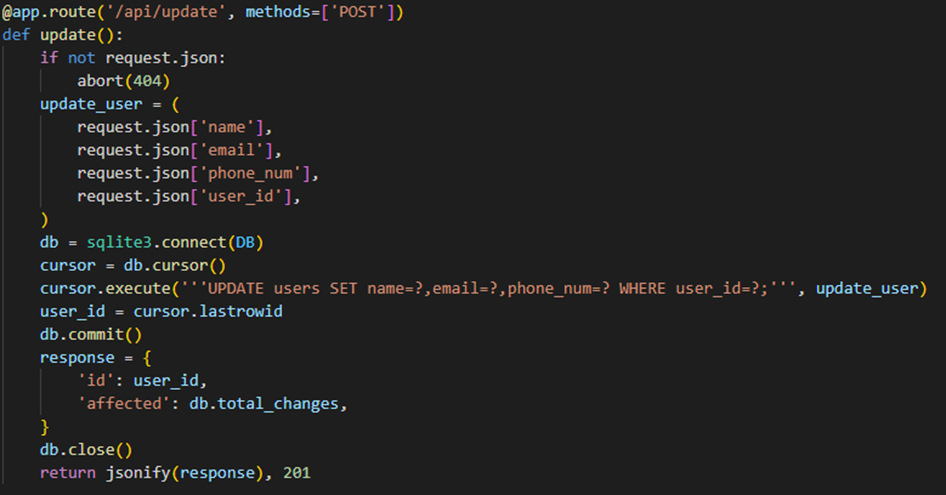
loginSuccess() - invoked by the front end to retrieve data of user who logged in successfully



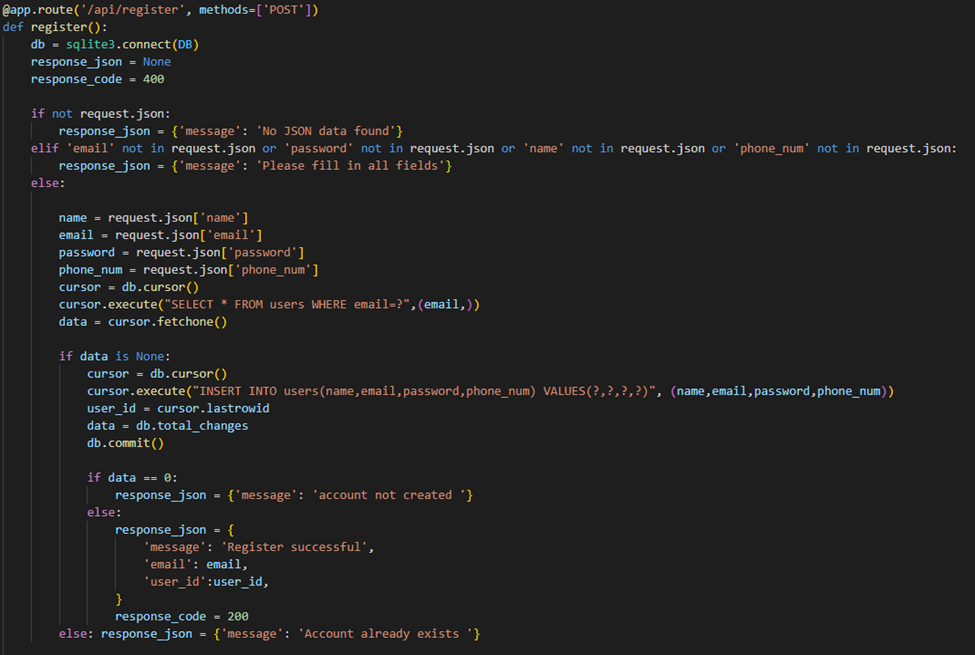
retrieveuser() - retrieve a row from user table with specific user id



update() - allow current user to change profile information

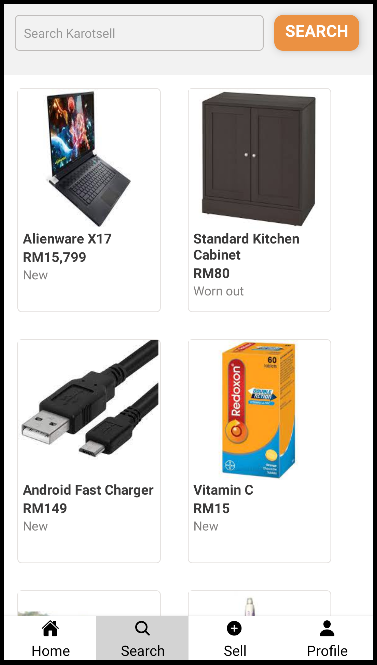
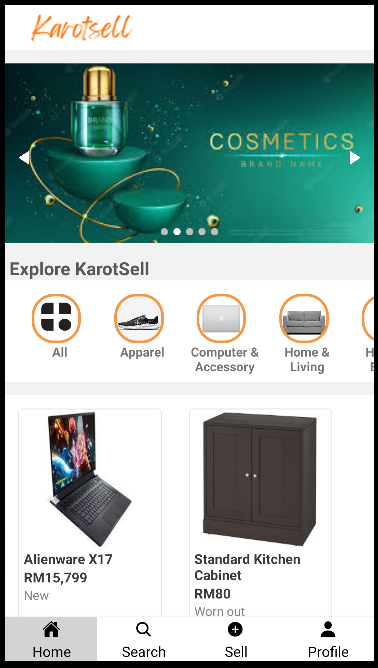


register() - verify if user already exists in database, and allow entry of new user in the database if user does not exist.



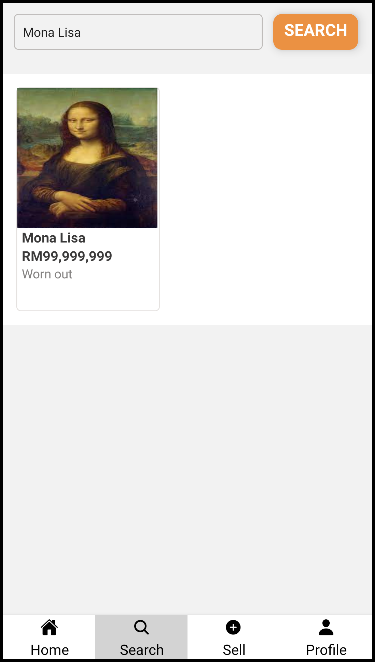
**5.0 Demonstration of CRUD Operations**

**5.1 View item**

****

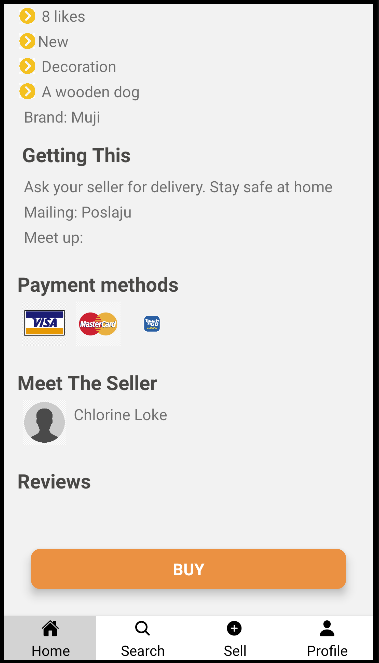
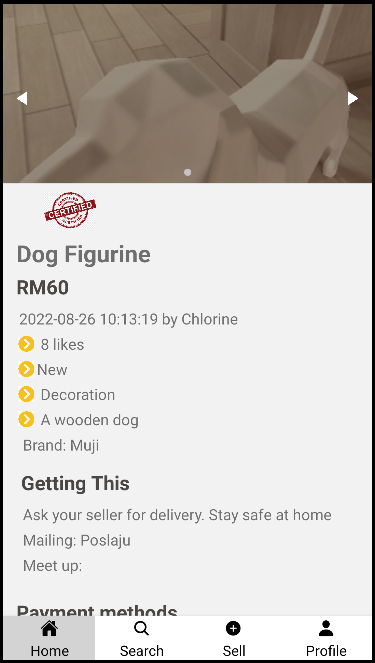
The query to retrieve all the items for sale from the database is SELECT \* FROM item. Here we retrieve and display all items in the home screen and search screen of Karotsell.

**5.2 Search item**

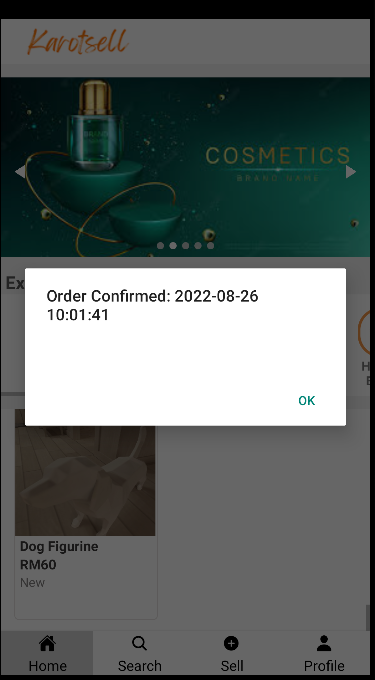
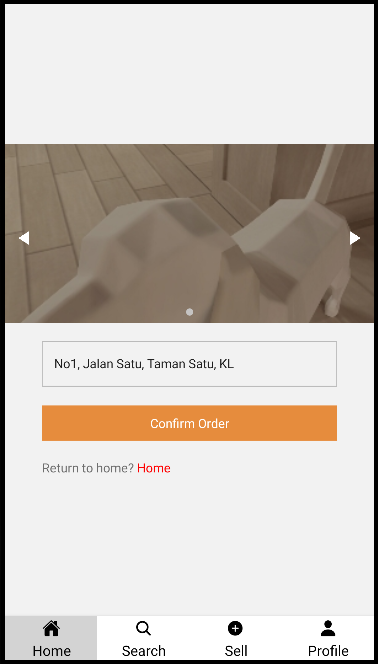
****

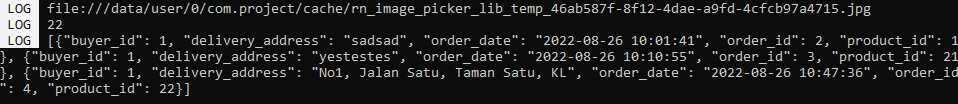
In the search screen, users can specifically search for individual items by their name, we achieved this by using the SQL query of selecting items according to their name. The query is “SELECT \* from item WHERE name LIKE ?”

**5.3 Purchase product**

****

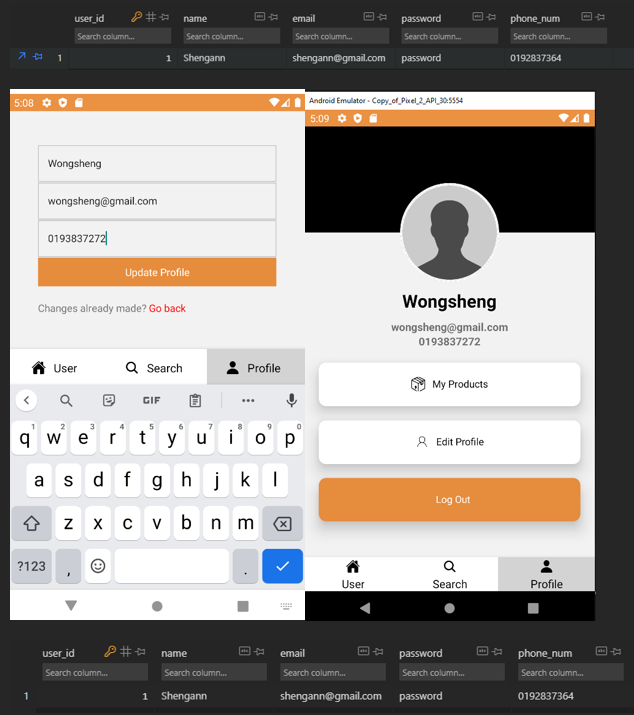
When users click into the individual product screens they can buy the product using the “Buy” button. Then they will be prompted to a confirmation screen.





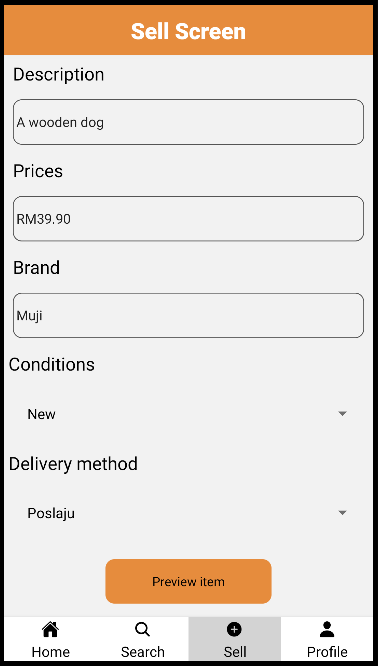
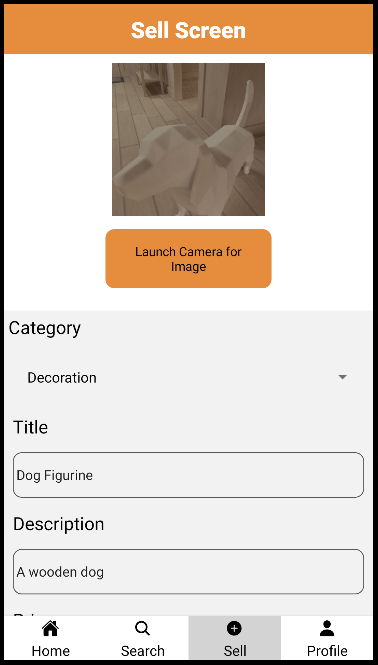
In this confirmation screen users would have to key in their address and press the “Confirm Order” button to submit their order and their orders will be saved into the order table in the database. Here we check if the order has been added by using the console to display the order table. In the code, we achieve this by using the “Insert to table” query.

**5.4 Edit user profile**

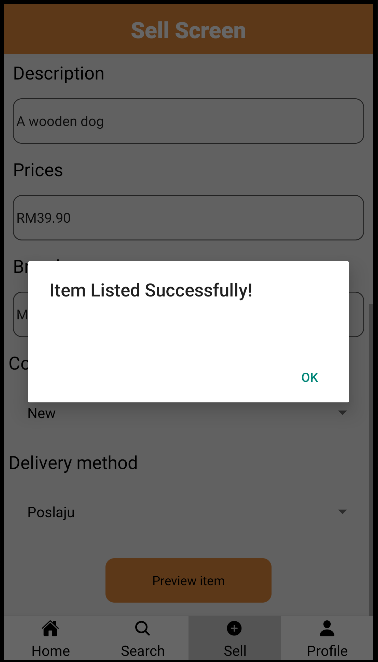
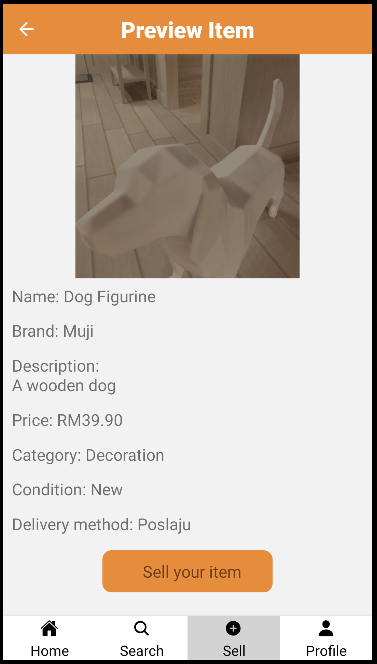
****

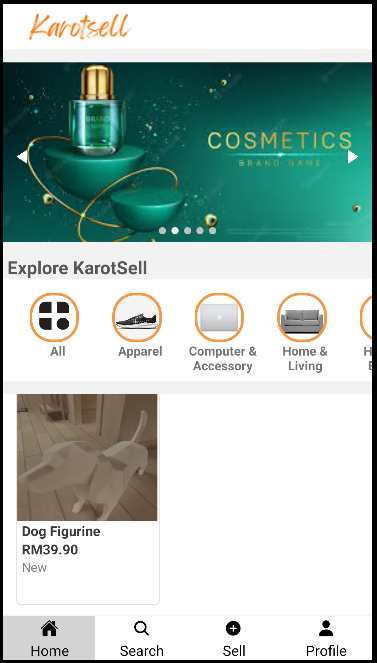
In the edit profile page, users would be prompted to enter their new profile name, email address and also phone number. Then the user tables will be updated to the new values. Here we are using the “Update table” SQL query to achieve this update effect.

**5.5 Add new product**

****

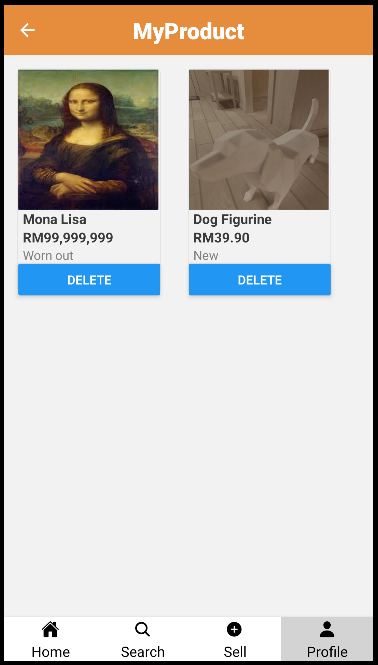
Once users finish entering information for the product to be posted for sale, they would need to click on the preview item button and click confirm to put the product up for sale.



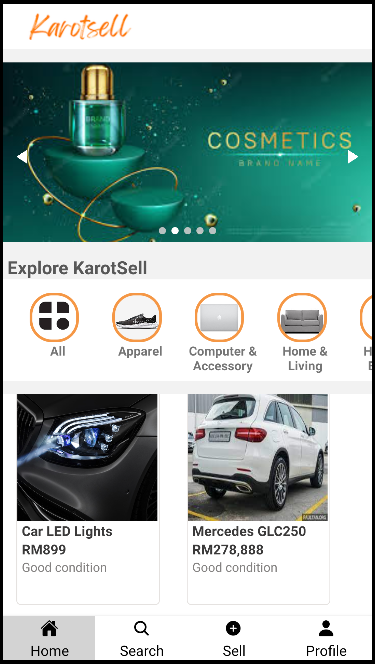
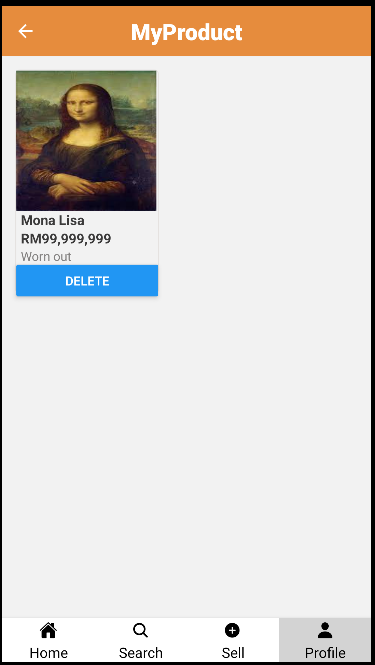


After that users will be able to see their products on sale from the home page of Karotsell. Here we are using the SQL insert query to add new items to the database.

**5.6 Delete product**

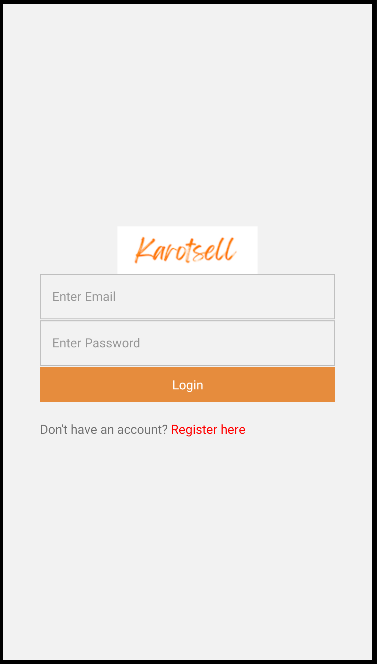
****

To delete a product, navigate to the “MyProduct” page in user profile and click on the “Delete” button.

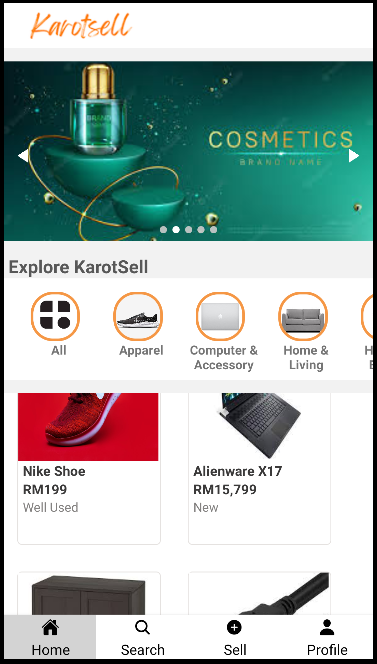


After that the item will be deleted from the “MyProduct” screen and also from the “HomePage” screen. SQL delete statements are used to remove individual items from the database.

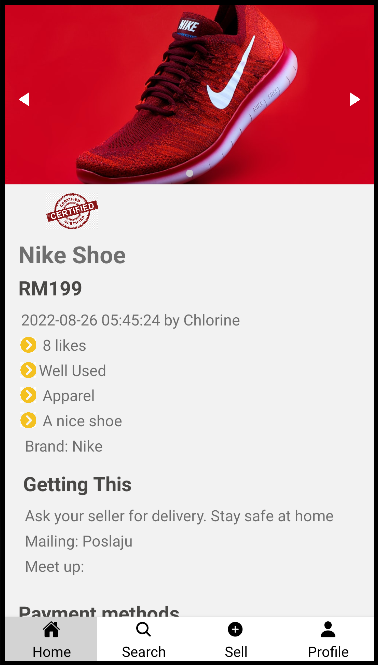
**Sample Screens Output**



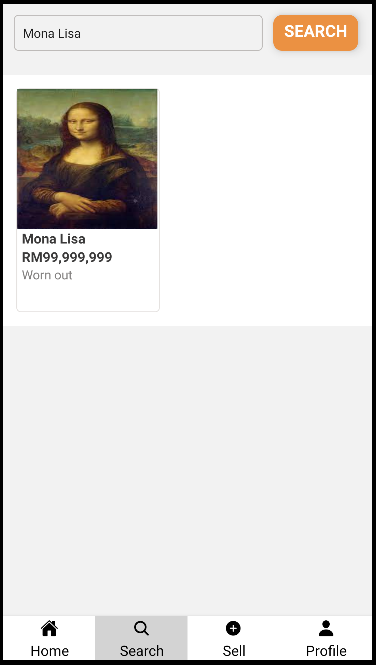
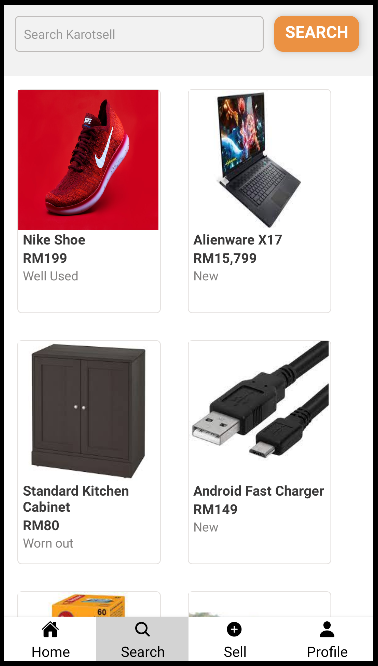
Login Page for Karotsell



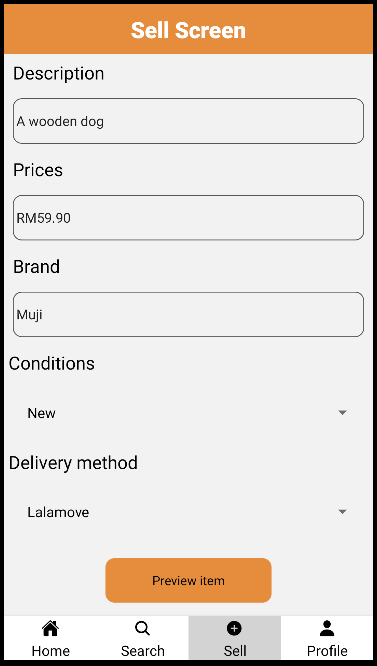
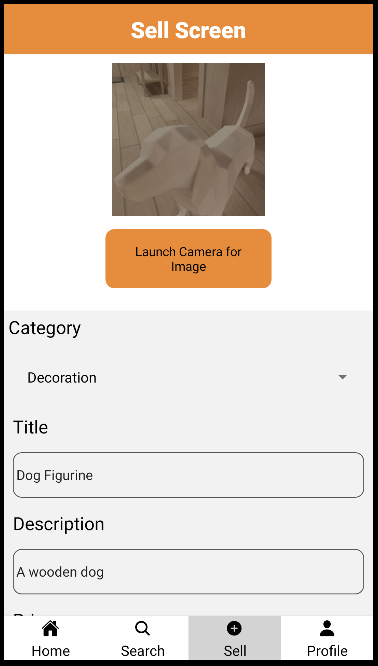
Home Page



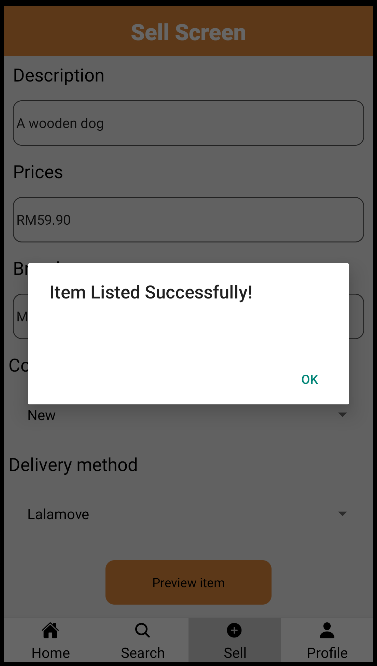
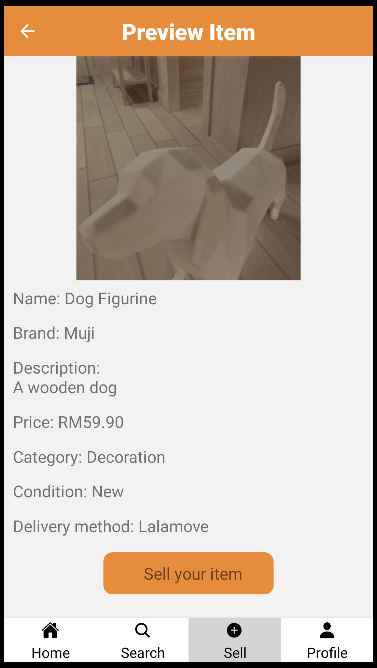
Screens containing information of individual items



Search Screen for Karotsell



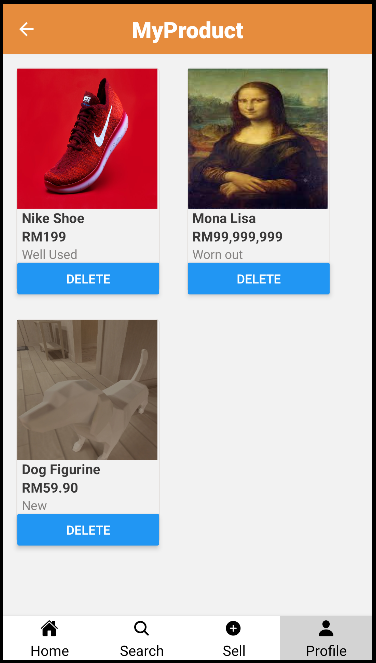
Sell screen of Karotsell



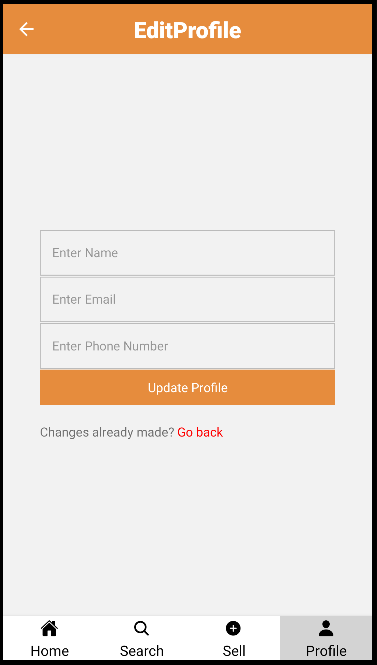
Preview Page of Karotsell before confirming to selling item



User profile page for Karotsell



MyProduct page inside user profile to show what are the items posted for sale by the user



Edit user profile page inside user profile