Technological and Higher Education Institute of Hong Kong

Faculty of Design and Environment

BSc (Hons) in Information and Communications Technology

Final Year Project Report

< APP Making – ShoPair >

Student Name: Iu Cheuk Ying

Student No.: 160426110

Advisor: Mr. Rex Lau

Date: 18/05/2020

Content Page

1. Introduction
    a. Background
    b. Literature Review
    c. Project Objectives

2. System Requirements
    a. Scope of the proposed system
    b. Description of functions provided
    c. Data processed by the system
    d. Decision of hardware / software tools with reasons

3. System Analysis
    a. Use Case
    b. Class diagram
    c. Sequence diagram

4. System Testing
    a. Test Plan
    b. API Test
    c. App UI Test
    d. Website UI Test

5. Critical Evaluation
    a. Problems / difficulties encountered
    b. Any changes in project schedule
    c. Limitations of the proposed system
    d. Future Improvement

6. Conclusion

7. References

8. User Menu

1. Introduction

    a. Background
    In this project, an app for item price comparing will be made. The name of the app is called ShoPair. It is including partly of the words searching and compare. There will be a website for the admin to edit and add the information in the App so to provide a more accurate information to the client.

    Below are some App and website are having similar function for other category:

    Expedia is a comparing app and web for people to comparing the hotel and flight price. However, it is only forces on traveling, not for local. Shopping and Travel is also important for people. Especially shopping, there are many things that people need to be needed in their daily life. So, a shopping comparing app will be as important as a travel comparing app.

    Price.com is one of the similar products that are launching on the app store now. It is also a shopping comparing app. However, the category of the app is not comprehensive, the things that people will use in their daily life, like food, drink, tissue e.c.t, is not included in it. At the same time, all of the information from the app is come from the shop owner and not all the shop is including in the app. User could not find the prices from the small shop that near them, they may miss the real cheapest price from the shop.

    Handy Lah is another similar product that are launching on the app store now. However, it is only comparing in 6 supermarkets. The information is also come from the shop owner. User still could not find the prices from the small shop that near them, they may miss the real cheapest price from the shop.

    People will like to know the price of the product in different store by walking and finding it out by themselves so they need a platform to including all the price of the product they want to buy.

Information from the small shops may not be included in the database so updating prices by the user themselves can help other to have the most update information from big or small shops.

That is the reason why this project should be done to let people having an App platform to compare the price before buying things. Helping them to buy the product in the cheapest price.

b. Literature Review

		App
1		big big shop - You can buy everything y…
2	淘	手机淘宝
3	C	Carousell: Snap-Sell, Chat-Buy
4		SHKP Malls
5	$	Price香港格價網

According to the play store from android, price.com is in the top 10 of the shopping app ranking. (SimilarWeb,2020)

		App
1	淘	手机淘宝 - 淘到你说好
2		HKTVmall – 網上購物
3	C	Carousell
4		big big shop - 睇得到, 買得到
5		MoneyBack
6	S	SHEIN-Fashion Shopping Online
7	特价	淘宝特价版
8	$	Price.com.hk 香港格價網

According to the app store from iOS, price.com is also in the top 10 of the shopping app ranking. (SimilarWeb,2020)

It shows that although Hong Kong people are asking for staying at home in this few month, they are still caring on the price on the product that they will

like to buy. That the reason why price.com is still keeping in top 10 of the shopping category.

Hong Kong people is the most connected consumers in the world, they love buying things a lot. Consumers in nowadays Hong Kong mainly looking for high-quality products and competitive prices. (Societe Generale, 2020) It can show that Hong Kong people is really in love with shopping especially the product they think is in a reasonable price while comparing.

c. Project Objectives

The objective of this Project:
➢ Make an App to solve people need
➢ Using the knowledge that have been learn on class
➢ Learn the new technology to enrich the project

The objective of the App (ShoPair):
➢ To help the people to compare the price of the product they want to buy
➢ To give a platform for the user to share the updated price for helping each other to have the newest information, especially the information from the small shops

The objective of this report:
➢ Finally app introduction
➢ Testing of the app
➢ Reflection on the whole fyp project

2. System Requirements

   a. Scope of the proposed system

   Client Part:
   The App are making as a shopping comparing system. It including a main page which including the slider for new information, some promotion poster, type searching button and all item search button.

   Searching page is set as a separate page for user to find the product that they want easily. It including the searching system which is search by keyword. They can use keyword to searching the item, the type of item or the information of the item that they want to find.

   Page of the product detail is a page that showing the address and price of the product that user find after searching from the searching page. There is not only one address for the user, there can be a lot which to help them to choose the price that they want for that item and know where they can buy it.

   Creating address page is for the user to create the new address and price that they can see for the item. When they see the price, which is lower and they can add the information on the app, other can know about it immediately.

   Login will be need for identity on the price compare given.

   Admin Part:
   A website was made for admin(staff) to get update on the information of the database.

   Page for create like item, item information is for the admin to create a new data for improve the data of the app.

   Page for list like member, item, item information, staff, app information is for the admin to update the previous data which made have wrong data or new updated data. It can make sure that the app has the newest data for the client.

   b. Description of functions provided
   Client Part (App):
   ➢ Login system for users
   ➢ Searching system
   ➢ Create address information for the database

Admin Part (Website):
- ➢ Login system for admin
- ➢ Create and update information for the database

Other Part:
- ➢ Database for saving the product information
- ➢ Server for managing the App and website
- ➢ API from the server

c. Data processed by the system
   - ➢ Login
     The user needs to sign up for helping to updating the recent cheapest price of the product. The simple information of the user like username, password and other information will be need to saved down in the database

   - ➢ Create for client users
     The shop name, address and also price will help to create by the client who using the app. That can help to enhance the quantity of the data from the database.

   - ➢ Create and update for admin users
     More data will be creating by the admin staff. More confirmed information will be inputted after the admin has checked in real. The information from the client may have wrong information or not updated information, admin will help to updated it so to make sure the data in the app has a high correction.

   - ➢ Searching
     Using key word to search the data in the database. The key word can help to find the item name, item type or item information which can search more data to let client to use. After user find the item they want, the sever will give all the information of the item to the UI and let client to see.

d. Decision of hardware / software tools with reasons
   - ➢ React Native (Android App)
     React Native is using for app making which using API from the server to add into the UI design and show the data in the database to the client. React Native gives simple interface which can help to reduces the loading time of UI. It is more stable in app which the data binding

process is more simplifying. It can use in cross platform like web, iOS and android at the same time.

➢ HBS (Website)
HBS names handlebars which is using to make the admin website. It is a file same as the html but it can embed with handlebars expressions.

➢ Mongodb
Mongodb is for the database use for saving all the data. The single object structure is clear to use. It is easier to scale on. It is fast to update.

➢ Node.js
Node.js is using to hold a server. It is easy to learn which one of the lessons in this semester is also teaching it. It also gives a full-stack to connecting the client side and server side.

3. System Analysis

   a. Use Case
      ➤ Users (Client)
        Users is member that has been register and they need to login by email
        and password to use the App. They can use the searching function by
        typing in the keyword or pressing the type button in the main page.
        They can also help to update the price of the product and address of
        where it can buy to help to improve the database of the App.

      ➤ Admin (Staff)
        Admin is the staff of the app. They already have the staff_id and
        password to login the website of the admin use. They can use the
        website to create and update the data of the database. At the same
        time, they can see the data of the database at the list page to know
        what data are already input in it.

      ➤ Database
        Database is mainly for the saving and updating.

        Client Part:
        For the searching, the database will help to check the record of the
        product. If there are record, the database will send back the product
        name, product type, product information, product prices and where can
        it buy to the App page and show it to the user. If there are no record,
        the database needs to reply no data to the app so the server will show
        nothing to the user.

        For the login function, the database is main helping on saving the login
        email, password and also some information of the user. When the user
        registers the information, the server will be sent the information of what
        they type into the database. The database will save the information and
        ready for login checking. When the user login to the App, the database
        needs to help checking the email and password. If it is correct, the user
   can login. If not, the login will be failed.

        For product, as the users can help to update the prices, places name and
        the places address of the product that the App have. The new data will
        be sent back to the database and save it down. It will send it back to the
        App when the product is searching by the other user.

        Admin Part:
        For the login function, the database is main helping on saving the login
        staff_id, password and also some information of the staff. The staff do

not need to register the account as the company will help to create the account in the database so it has already been ready for login checking for the admin. When the admin login to the website, the database needs to help checking the staff_id and password. If it is correct, the staff can login. If not, the login will be failed.

For product, as the staff will be the one who create the item and item information and update the member, item, item information, staff, app information. The new data or updated data will be sent back to the database and save it down. It will be sent it back to the member, item, item information staff or app information list in the website to show back the data that are in the database when the staff want to check it.
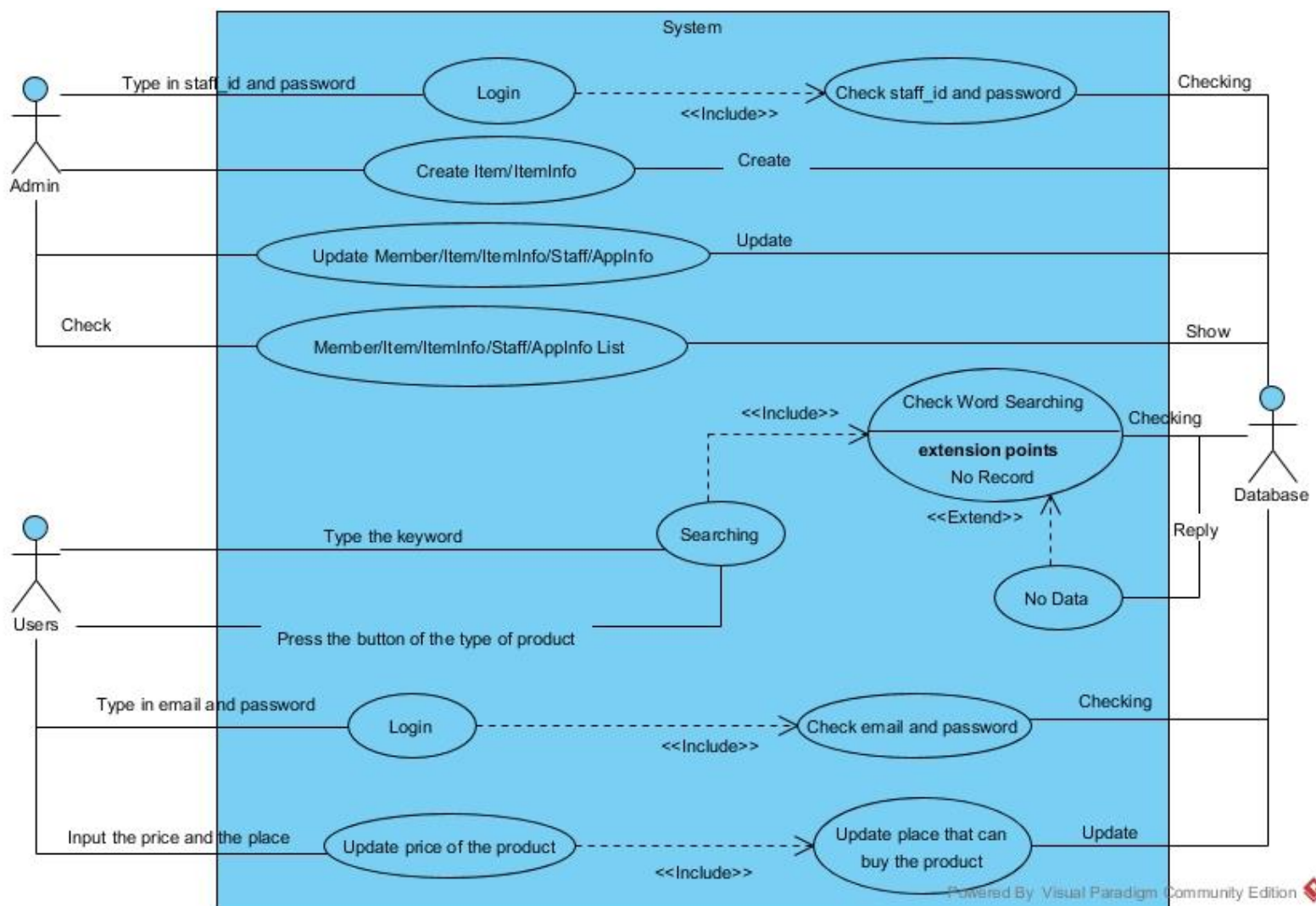
b. Class diagram

Client side is including 19 classes for provide services for the App. Login is the first page which linked with register and menu.
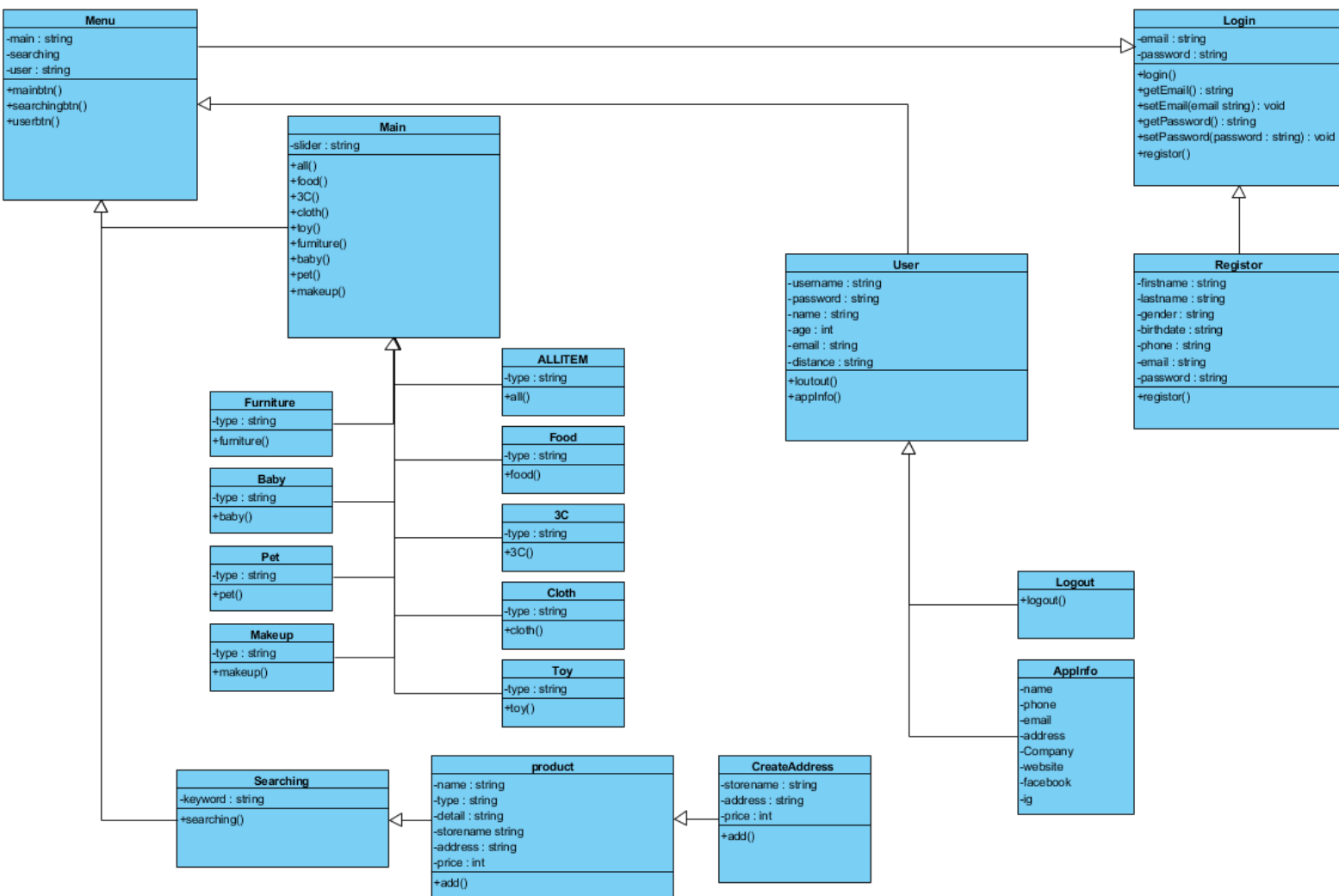When it is in menu, it is linkage with main, searching and user.
For Main, it links to 9 types of item, all item, food, 3C, cloth, toy, furniture, baby, pet and makeup.
For user, it links to app information and logout.
For searching, it links to product and sub link to create address.
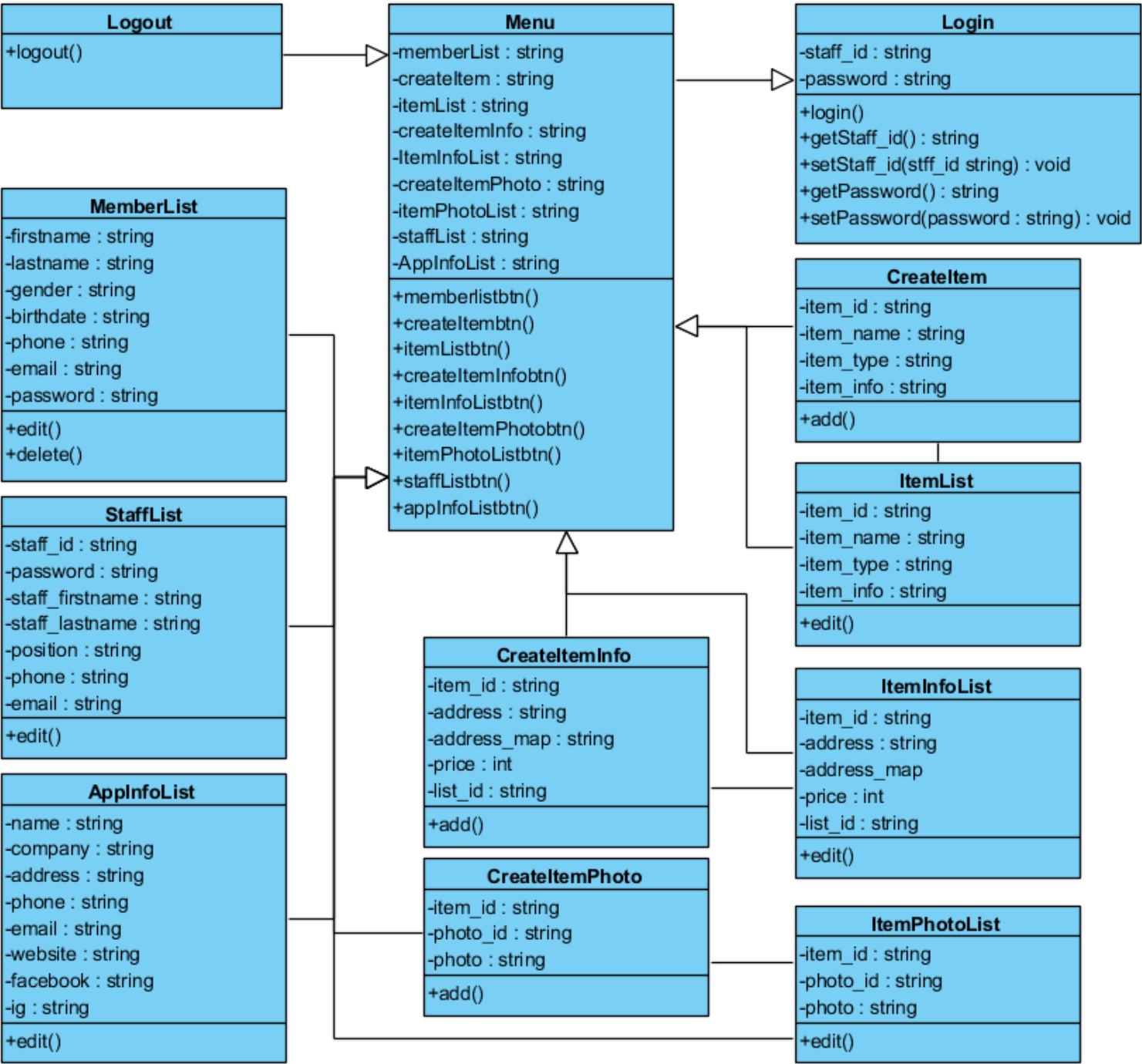
Client Part:

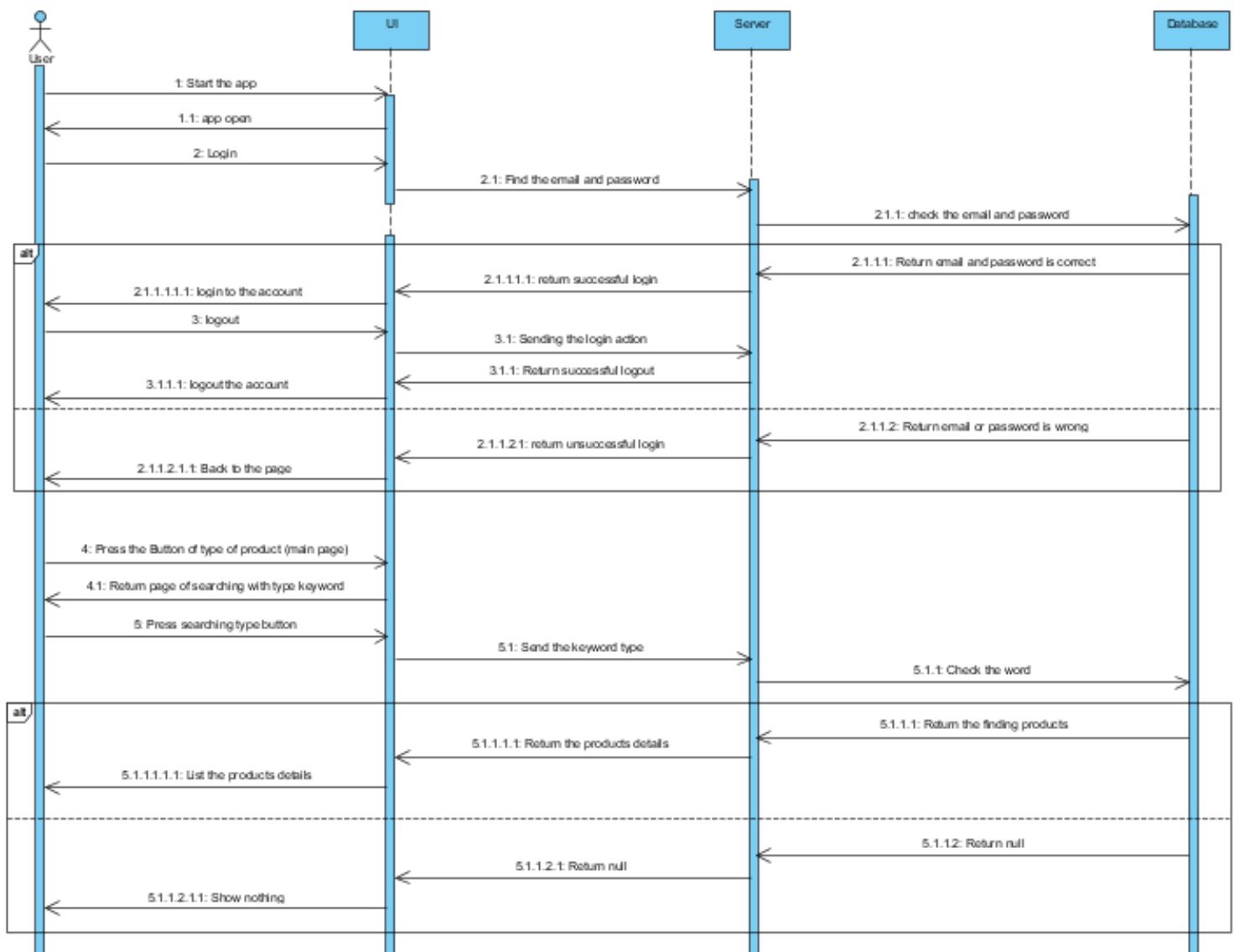Admin side is including 12 classes for providing the service for the website.
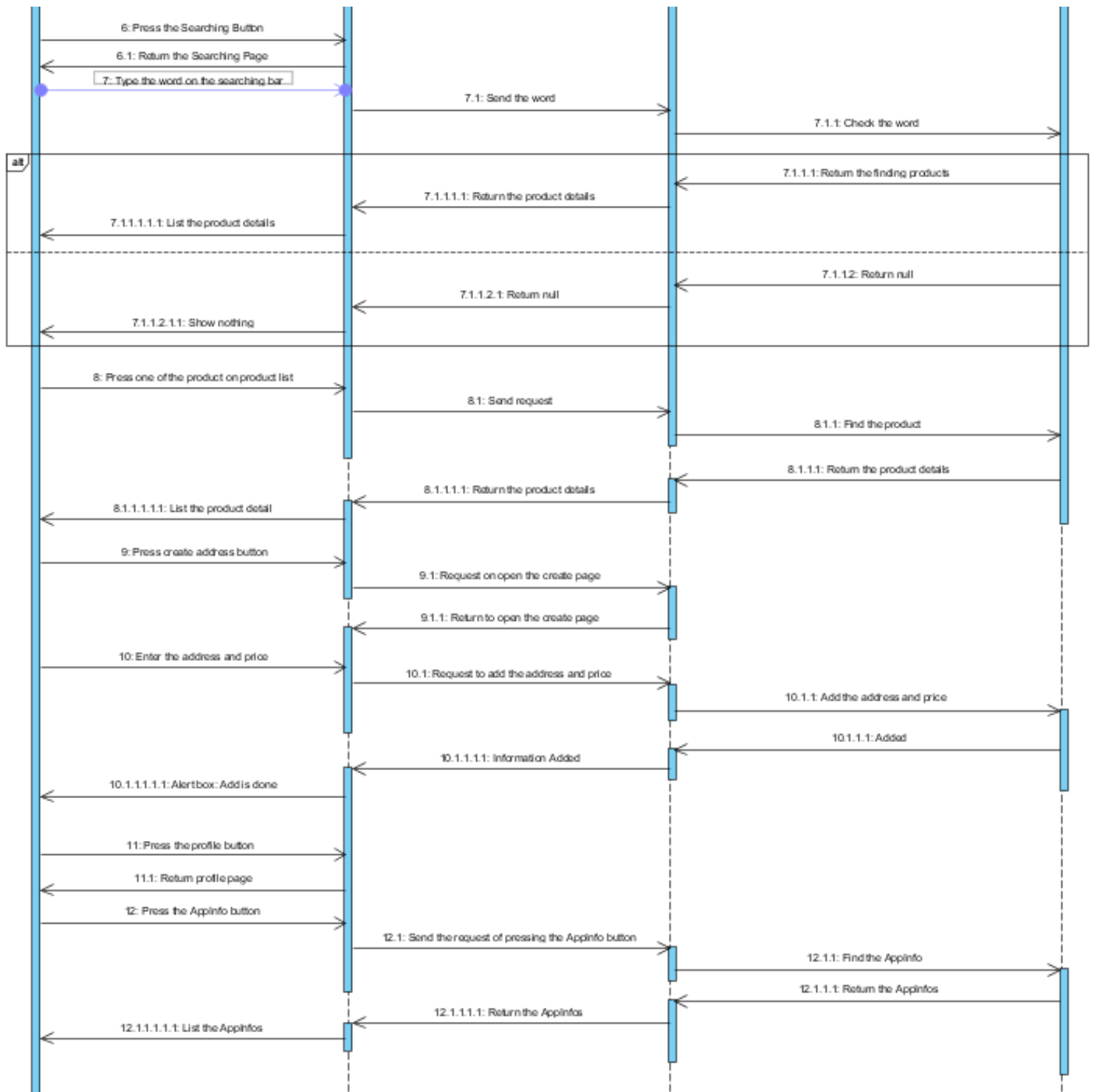Login is also the first page which linked with and menu.
Menu is linked with all other classes, logout, memberList, staffList, appinfoList, createItem, itemList, createItemInfo, itemInfoList, createItemPhoto and itemPhotoList.

Admin Part:

Logout
+logout()

Menu
-memberList : string
-createItem : string
-itemList : string
-createItemInfo : string
-ItemInfoList : string
-createItemPhoto : string
-itemPhotoList : string
-staffList : string
-AppInfoList : string

+memberlistbtn()
+createItembtn()
+itemListbtn()
+createItemInfobtn()
+itemInfoListbtn()
+createItemPhotobtn()
+itemPhotoListbtn()
+staffListbtn()
+appInfoListbtn()

Login
-staff_id : string
-password : string

+login()
+getStaff_id() : string
+setStaff_id(stff_id string) : void
+getPassword() : string
+setPassword(password : string) : void

MemberList
-firstname : string
-lastname : string
-gender : string
-birthdate : string
-phone : string
-email : string
-password : string

+edit()
+delete()

CreateItem
-item_id : string
-item_name : string
-item_type : string
-item_info : string

+add()

ItemList
-item_id : string
-item_name : string
-item_type : string
-item_info : string

+edit()

StaffList
-staff_id : string
-password : string
-staff_firstname : string
-staff_lastname : string
-position : string
-phone : string
-email : string

+edit()

CreateItemInfo
-item_id : string
-address : string
-address_map : string
-price : int
-list_id : string

+add()

ItemInfoList
-item_id : string
-address : string
-address_map
-price : int
-list_id : string

+edit()

AppInfoList
-name : string
-company : string
-address : string
-phone : string
-email : string
-website : string
-facebook : string
-ig : string

+edit()

CreateItemPhoto
-item_id : string
-photo_id : string
-photo : string

+add()

ItemPhotoList
-item_id : string
-photo_id : string
-photo : string

+edit()

c. Sequence diagram

6: Press the Searching Button

6.1: Return the Searching Page

7: Type the word on the searching bar

7.1: Send the word

7.1.1: Check the word

7.1.1.1: Return the finding products

7.1.1.1.1: Return the product details

7.1.1.1.1.1: List the product details

7.1.1.2: Return null

7.1.1.2.1: Return null

7.1.1.2.1.1: Show nothing

8: Press one of the product on product list

8.1: Send request

8.1.1: Find the product

8.1.1.1: Return the product details

8.1.1.1.1: Return the product details

8.1.1.1.1.1: List the product detail

9: Press create address button

9.1: Request on open the create page

9.1.1: Return to open the create page

10: Enter the address and price

10.1: Request to add the address and price

10.1.1: Add the address and price

10.1.1.1: Added

10.1.1.1.1: Information Added

10.1.1.1.1.1: Alertbox: Add is done

11: Press the profile button

11.1: Return profile page

12: Press the AppInfo button

12.1: Send the request of pressing the AppInfo button

12.1.1: Find the AppInfo

12.1.1.1: Return the AppInfos

12.1.1.1.1: Return the AppInfos

12.1.1.1.1.1: List the AppInfos

4. System
    a. API

Method	get/post	Description	File
login/	get	Render the page to login/login.hbs.	loginController.js
login/	post	Checking the staff_id and pwd in the admin website login.	loginController.js
login/login	post	Checking the email and password in the client app login.	loginController.js
member/	post	Update the member record from the admin website and input to the members database.	memberController.js
member/signup	post	Create the member record from the client app and input to the members database.	memberController.js
member/list	get	Get all the member record from the database.	memberController.js
member/:id	get	Get the member record by id from the database.	memberController.js
item/	get	Render the page to item/add_edit.hbs.	itemController.js
item/	post	Check if it is insert or update record. Create the item record from the admin website and input to the items database. Update the item record from the admin website and input to the items database.	itemController.js
item/find	get	Get all the item record from the database.	itemController.js
item/searching	post	Find the item record by keyword and check with the	itemController.js

		fill item_name, item_info and item_type from the database.	
item/type	post	Find the item record by type and check with the fill item_type from the database.	itemController.js
item/name	post	Find the item record by the item_name from the database.	itemController.js
item/list	get	Get all the item record from the database.	itemController.js
item/:id	get	Get the item record by id from the database.	itemController.js
itemInfo/	get	Render the page to itemInfo/add_edit.hbs.	itemInfoController.js
itemInfo/	post	Check if it is insert or update record. Create the item information record from the admin website and input to the itemInfos database. Update the item information record from the admin website and input to the itemInfos database.	itemInfoController.js
itemInfo/find	post	Find the item information record by id from the database.	itemInfoController.js
itemInfo/add	post	Create the item information record from the client app and input to the itemInfos database.	itemInfoController.js
itemInfo/list	get	Get all the item information record from the database.	itemInfoController.js

itemInfo/:id	get	Get the item information record by id from the database.	itemInfoController.js
staff/	post	Update the staff record from the admin website and input to the staffs database.	staffController.js
staff/list	get	Get all the staff record from the database.	staffController.js
staff/:id	get	Get the staff record by id from the database.	staffController.js
appInfo/	post	Update the app information record from the admin website and input to the appInfos database.	appInfoController.js
appInfo/appInfo	get	Get all the app information record from the database for the client app side.	appInfoController.js
appInfo/list	get	Get all the app information record from the database for the admin website.	appInfoController.js
appInfo/:id	get	Get the app information record by id from the database.	appInfoController.js

    b. Test Plan

In the testing, it will be divided into three parts. API will be including into App UI Testing as it is using axios to get the API. App UI Test is for checking all the button workable and create function in the App is usable or not. Website UI Test is for create and update function in the website is usable or not.

    c. App UI Test

Login

  ➢ Check:

Is the login workable or not?

  ➢ Result:

Successful Login

(email = apple@gmail.com   Password = apple)
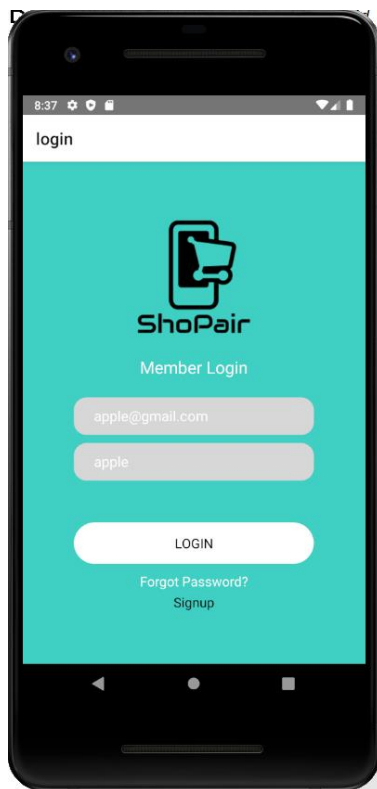
[Sun May 17 2020 16:38:07.687]   LOG   apple@gmail.com
[Sun May 17 2020 16:38:07.696]   LOG   apple
[Sun May 17 2020 16:38:07.698]   LOG   false
[Sun May 17 2020 16:38:07.923]   LOG   {"__v": 0, "_id": "5
000Z", "createdAt": "2020-05-13T16:16:08.001Z", "email": "ap
lastname": "LamLam", "member_id": 0, "password": "apple", "ph
Sun May 17 2020 16:38:08 780]   LOG   true

{
  _id: 5ebc1d47da95f87ba0dd3265,
  member_id: 0,
  firstname: 'AppleApple',
  lastname: 'LamLam',
  gender: 'F',
  birthdate: 2020-05-07T00:00:00.000Z,
  phone: '23456789',
  email: 'apple@gmail.com',
  password: 'apple',
  createdAt: 2020-05-13T16:16:08.001Z,
  updatedAt: 2020-05-17T08:18:43.917Z,
  __v: 0
}

Unsuccessful Login
(email = apple   Password = apple)
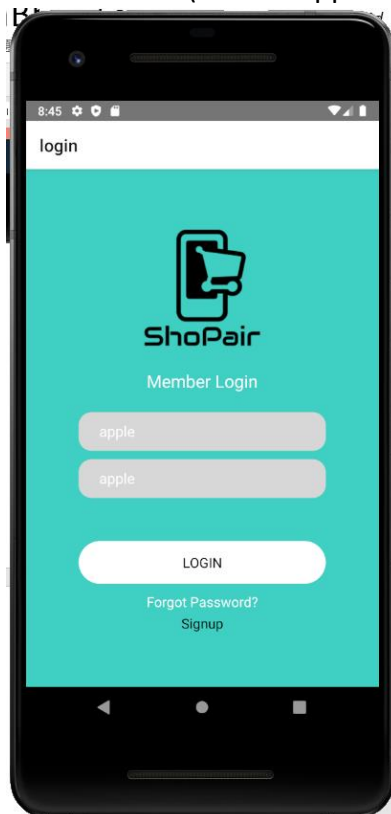

```
null
Login failure. Wrong account or password
```

Signup
➢ Check:
  Is the create member workable or not?
➢ Result:
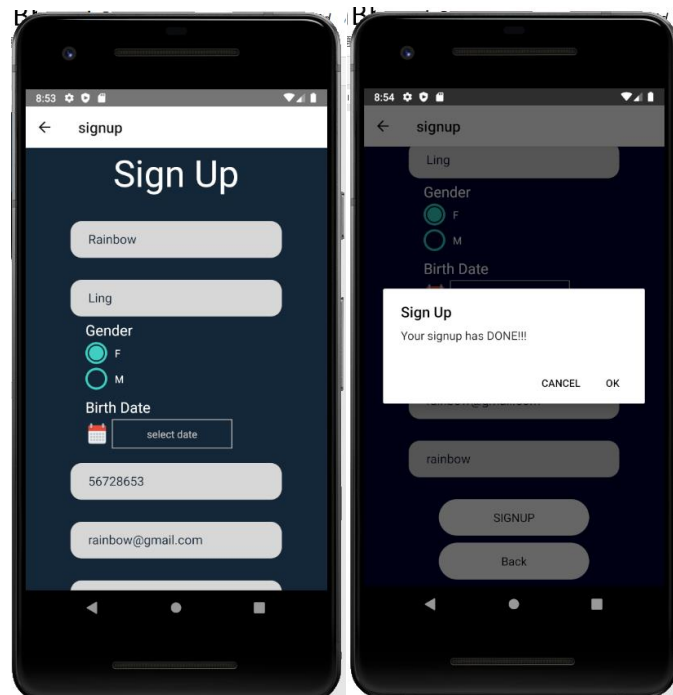
0	Rainbow	Ling	F	Wed May 06 2020 08:00:00 GMT+0800 (GMT+08:00)	56728653	rainbow@gmail.com	rainbow	Edit

```
_id: ObjectId("5ec0fba8d39a651b8837d07f")
member_id: 0
firstname: "Rainbow"
lastname: "Ling"
gender: "F"
birthdate: 2020-05-06T00:00:00.000+00:00
phone: "56728653"
email: "rainbow@gmail.com"
password: "rainbow"
createdAt: 2020-05-17T08:54:00.031+00:00
updatedAt: 2020-05-17T08:54:00.031+00:00
__v: 0
```

Create Item information
➢ Check:
  Is the create item information workable or not?
➢ Result:
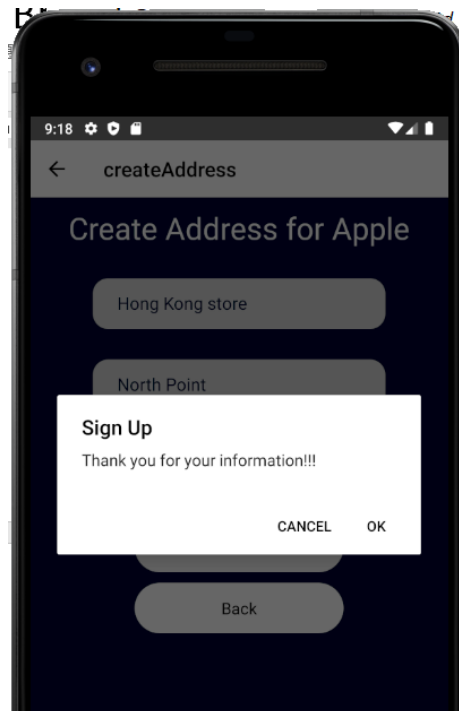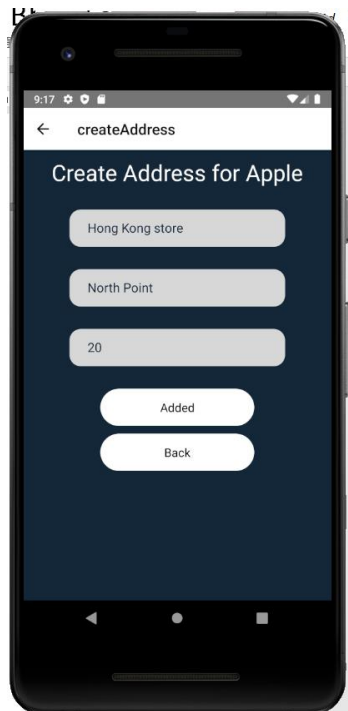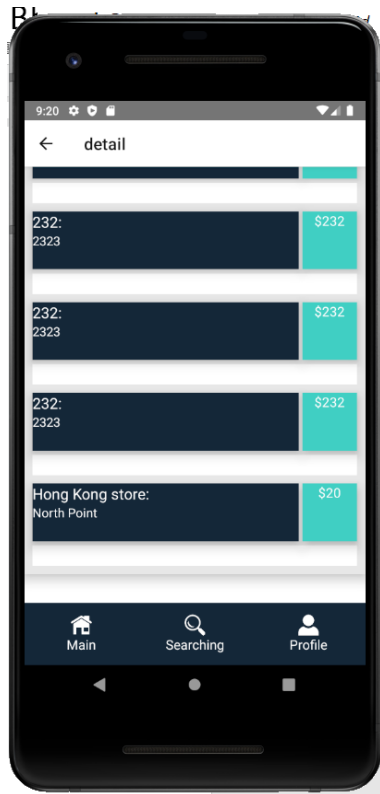
_id: ObjectId("5ec1017dd39a651b8837d080")
item_id: 1
address: "Hong Kong store"
address_map: "North Point"
price: 20
__v: 0

1		Hong Kong store	North Point		20	Edit

d. Website UI Test
   Login
   ➢ Check:
      Is the login workable or not?
   ➢ Result:

      Successful Login
      (Staff_id = 3   Password = admin12345)

MongoDB Connection Succeeded.
{
  _id: 5eb6c8f474b3147220305f49,
  staff_id: '3',
  name: '',
  pwd: 'admin12345'
}

Coming soon!

Unsuccessful Login
(Staff_id = 4   Password = admin)

null
Login failure. Wrong account or password

Update Member
➢   Check:
      Is the update member workable or not?
➢   Result:

Member ID	First Name	Last Name	Gender	Date of Birth	Phone No.	Email	Password	
0	Apple	Lam	F	Tue Apr 14 2015 08:00:00 GMT+0800 (GMT+08:00)	23456789	apple@gmail.com	apple	Edit

First name: Apple → First name: AppleApple
Last name: Lam → Last name: Lam Lam

First name:

AppleApple

Last name:

LamLam

Member ID	First Name	Last Name
0	AppleApple	LamLam

_id: ObjectId("5ebc1d47da95f87ba0dd3265")
member_id: 0
firstname: "AppleApple"
lastname: "LamLam"
gender: "F"
birthdate: 2020-05-07T00:00:00.000+00:00
phone: "23456789"
email: "apple@gmail.com"
password: "apple"
createdAt: 2020-05-13T16:16:08.001+00:00
updatedAt: 2020-05-17T08:18:43.917+00:00
__v: 0

Create Item
➢ Check:
   Is the create item workable or not?
➢ Result:

Item ID	Item Name	Item Type	Item Info	
1	Apple	Food	Apple	Edit
2	TV	furniture	TV	Edit
3	Orange	Food	Orange	Edit

Item_ID: 4
Item name: Dog Food
Item Type: Pet
Item Info: For Dog use

Item_ID:

4

Item name:

Dog Food

Item Type:

Pet

Item Info:

For Dog use

Item ID	Item Name	Item Type	Item Info	
1	Apple	Food	Apple	Edit
2	TV	furniture	TV	Edit
3	Orange	Food	Orange	Edit
4	Dog Food	Pet	For Dog use	Edit

_id: ObjectId("5ec0ec79faf74344cc273389")
item_id: 4
item_name: "Dog Food"
item_type: "Pet"
item_info: "For Dog use"
__v: 0

5. Critical Evaluation

   a. Problems / difficulties encountered
      ➢ Time
        The project is already working for a year. However, the time is still a
        limitation in it. There are many things are planned to do but work is not
        catching up the time. The time management still need a great
        improvement and to be tight and follow straightly. It is the hardest
        things in this project.

      ➢ Knowledge
        This is the first time to use react native to do the App. There is limited
        knowledge on it. More information searching need to be done to help
        on this project. There will be difficulty to learn a new knowledge in the
        limited time.

   b. Any changes in project schedule

Date (Month)	Work
September	➢ Idea Thinking
October	➢ Proposal
November	➢ App Making 　i. User Interface 　ii. AI object recognize ➢ Report
December	➢ Mid-term present ➢ App Making 　i. User Interface
January	➢ App Making 　i. Database 　ii. Own API of the product 　iii. Server
February	➢ App Making 　i. Database 　ii. Server 　iii. Own API of the product

March	➢ App Making 　　i. Database 　　ii. Server
April	➢ App Testing 　　i. Bug Checking ➢ Debug
May	➢ Debug ➢ Report
June	➢ Final Present

c. Limitations of the proposed system

As the database is set as new, the data inside is enter by hand so the data is not as much as other. More and more data will be need to store in the database. It will be using a lot of time on it to build up this database.

d. Future Improvement
- ➢ AI object recognition (App)
- ➢ Mapping (App)
- ➢ Top Item (App)
- ➢ Recommendation (App)
- ➢ History (App)
- ➢ Image Improvement (App)
- ➢ User Profile (App)
- ➢ Dashboard (Website)
- ➢ API (Database)

6. Conclusion

In conclude, a app with price comparing system and website for admin is done. Hong Kong People loved shopping. Everyone would like to buy in the cheapest price. ShoPair can help people to have the price and address information of the item sharing on the same platform which at the same time they can find the cheapest price of the item that they want to buy. Between the project, there are many problem appeared, learning to solve and do what I should do is important for my future. Although something that I prefer is not done at last, I still learn a lot from this project.

7. References

➢ Expedia. (2019). Retrieved November 29, 2019, from https://www.expedia.com.hk/

➢ Follow the leaders: highest ranking Apps in Google Play Store, Hong Kong. (2020). Retrieved from https://www.similarweb.com/apps/top/google/store-rank/hk/shopping/top-free

➢ Follow the leaders: highest ranking Apps in Apple App Store, Hong Kong. (2020). Retrieved from https://www.similarweb.com/apps/top/apple/store-rank/hk/shopping/top-free/iphone

➢ HONG KONG MARKET: CONSUMER (2020). Retrieved from https://import-export.societegenerale.fr/en/country/hong-kong/market-consumer

➢ Price.com.hk. (2019). Retrieved November 29, 2019, from https://www.price.com.hk/

8. User Menu:
   - Server
     - I.   Start the mongodb and pug in the data back to the database.
     - II.   cd to the file folder in the cmd
     - III. type in node server.js in the cmd
     - IV.   Then the server will be started
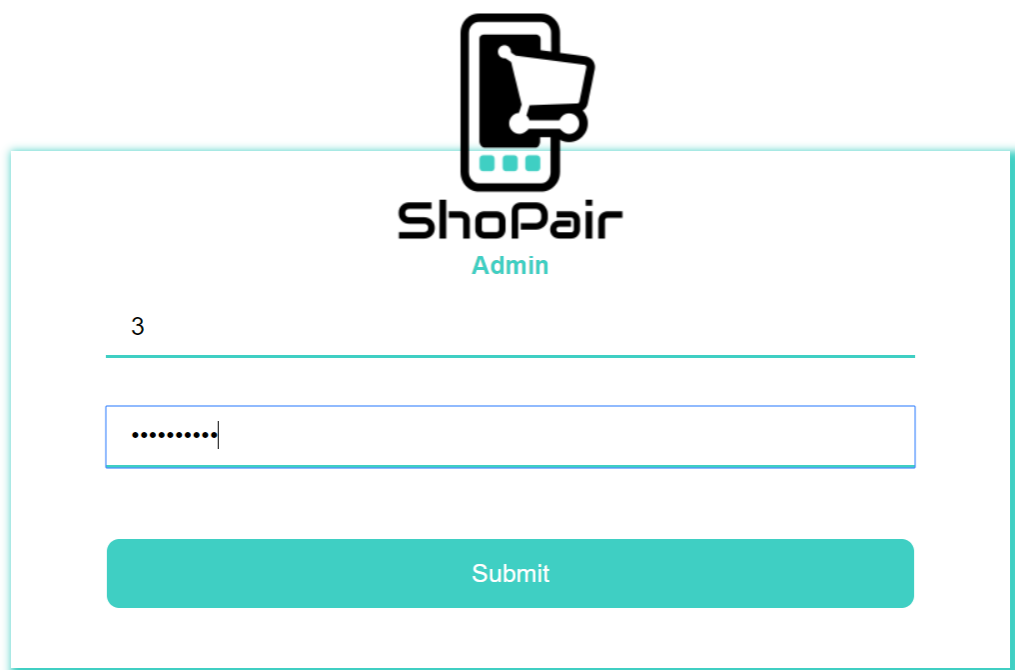
   - Admin Website
     Type http://localhost:3000/login (if is not in the same computer use ip address of the server computer instead of localhost)

   Login:
   - I.   Type in the staff_id and password in the login page.

II. If successful, it will jump to the main page.
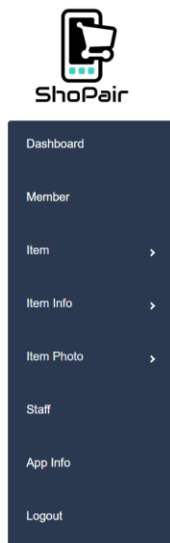
Create Item/Item Info:
I. Go to the Create page

II. Input the information to the input box

Item_ID:

| 5 |

Item name:

| Lego |

Item Type:

| Toy |

Item Info:

| Lego |

III. Press Submit

IV. It will show in the List

Item ID	Item Name	Item Type	Item Info	
1	Apple	Food	Apple	Edit
2	TV	furniture	TV	Edit
3	Orange	Food	Orange	Edit
4	Dog Food	Pet	For Dog use	Edit
5	Lego	Toy	Lego	Edit

Update Member/Item/Item Info/Staff/App Info:
I. Go to the list page

App Info List

Name	Company	Phone No.	Email	Address	Website	Facebook	IG	
ShoPair	ShoPair Comapny	23456789	shopair@gmail.com	123456	123456	ShoPair	ShoPair	Edit

II. Press edit button

Edit AppInfo

Name:

ShoPair

Company:

ShoPair Comapny

Phone No.:

23456789

Email:

shopair@gmail.com

Address:

123456

III. Input the information to input box
Example of Change Data:
Address:Mongkok
Website:http://shopair.com

Address:

Mongkok

Website:

http://shopair.com

IV. Press Submit

V. It will show in the List

Name	Company	Phone No.	Email	Address	Website	Facebook	IG	
ShoPair	ShoPair Comapny	23456789	shopair@gmail.com	Mongkok	http://shopair.com	ShoPair	ShoPair	Edit

➢ Client App (**API need to change back to the server ip)
Start the App

Login:

I.   Type in the email and password in the login page.
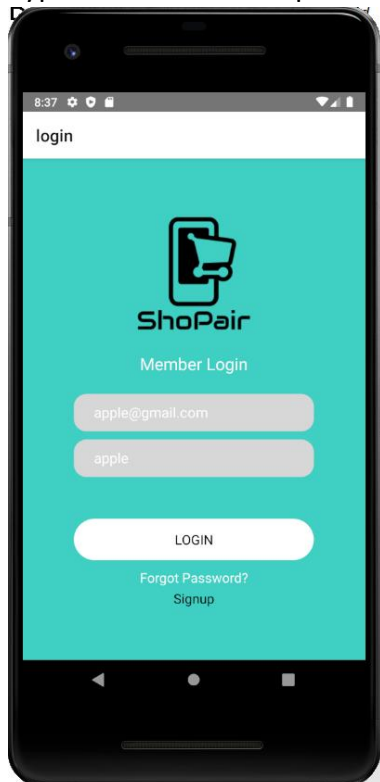
II. If successful, it will jump to the main page.

Sign Up:

I. Press the sign up button under the login page
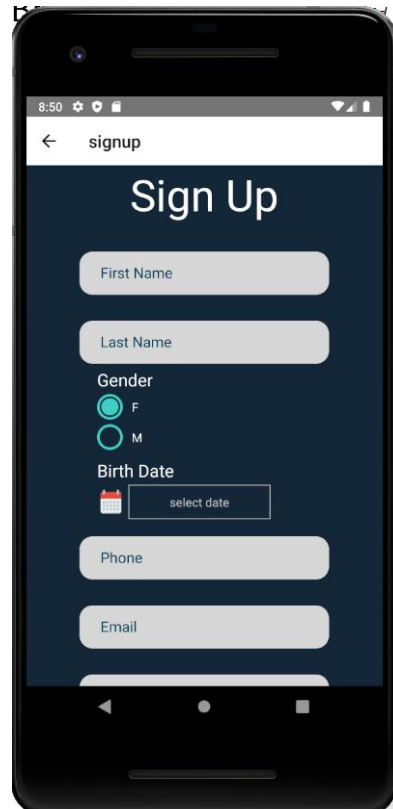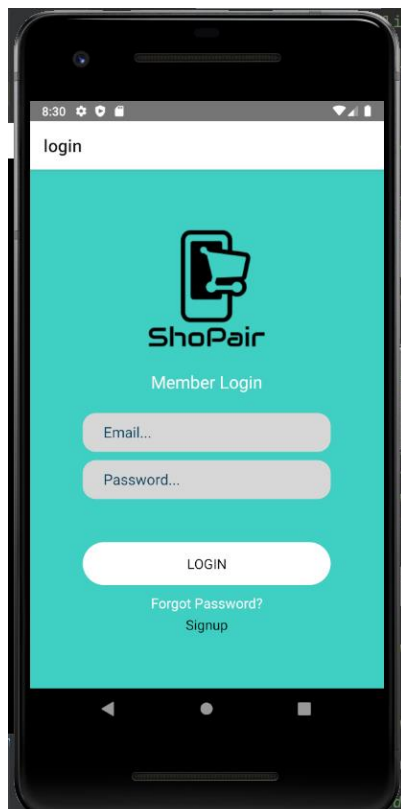
II. Fill in the input box

III. Press submit

IV. Alert box will show if sign up successful

V.   Press OK

VI.   It will jump back to the login page

Type Button

I.   In the main page, roll down to the bottom

II.   There is some circle button with picture represent different type of product
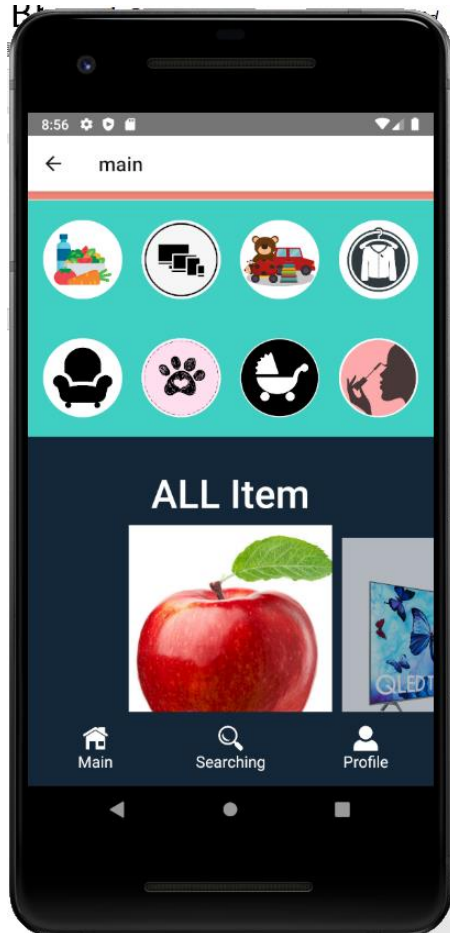
III. Press on one of the circle buttons

IV.   It will jump to the type searching page
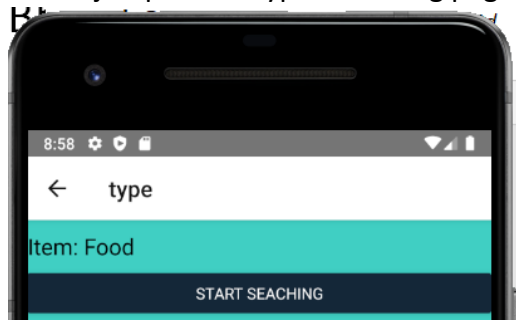
V.   Press the Start searching button

VI.   The record of all the food type will be appeared
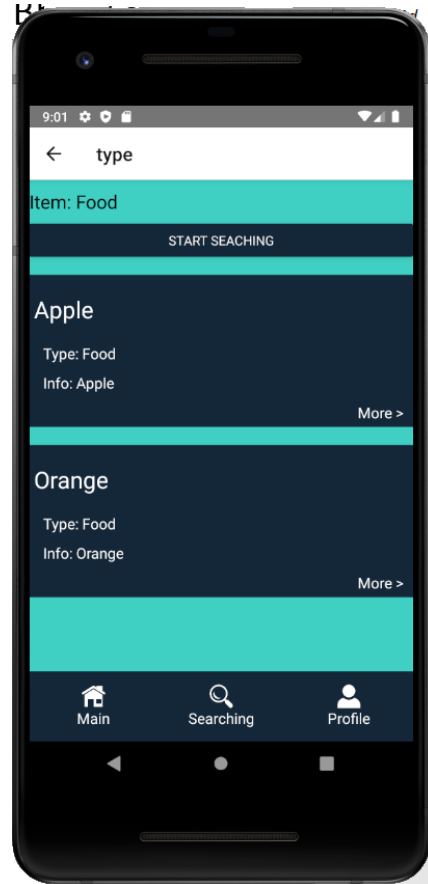
VII. Press one of the items you like
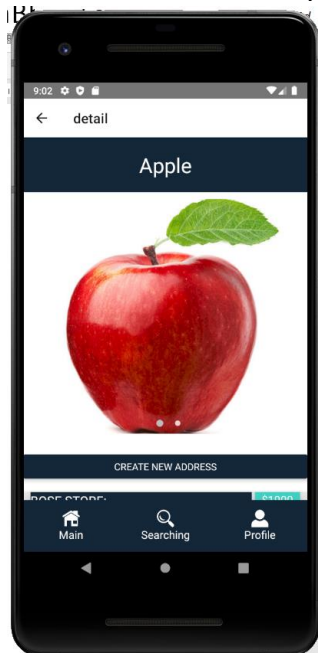
VIII. It will jump to the detail page of the item

Searching
I.   In the main page, roll down to the bottom

II.   There is ALL Item button with picture represent all product is shown
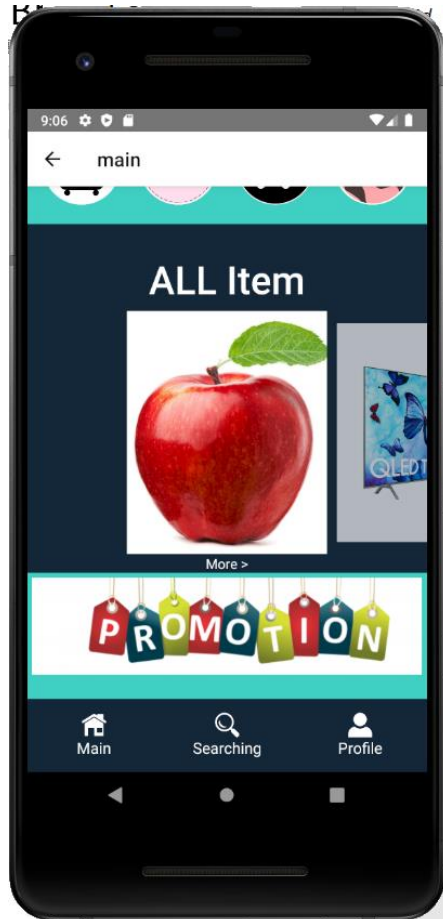
III.   Press the ALL Item button

IV.   It will jump to the type searching page
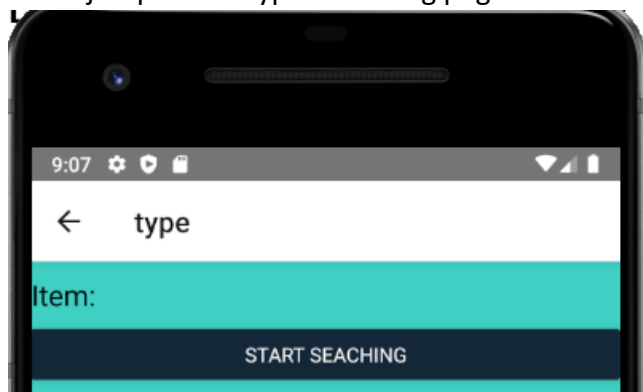
V.   Press the Start searching button

VI. The record of all the food type will be appeared
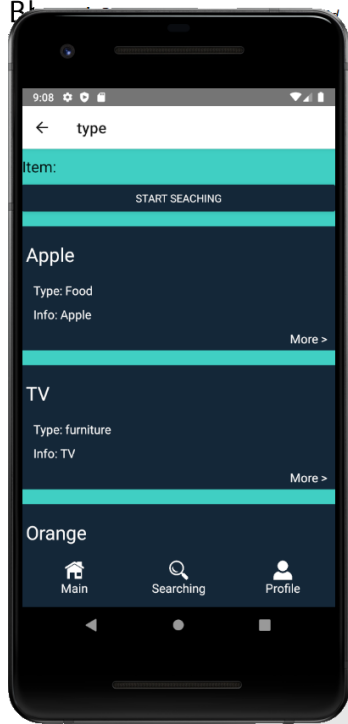
VII. Press one of the items you like
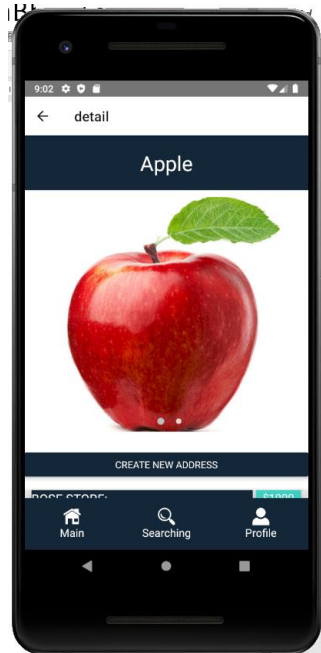
VIII. It will jump to the detail page of the item

IX. The address and price will show under
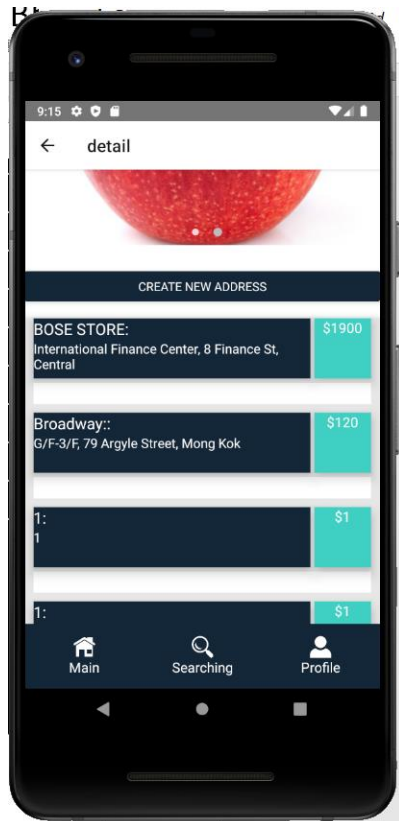
Searching:

I. Press the Searching button at the menu bar

II. It will jump to the searching page
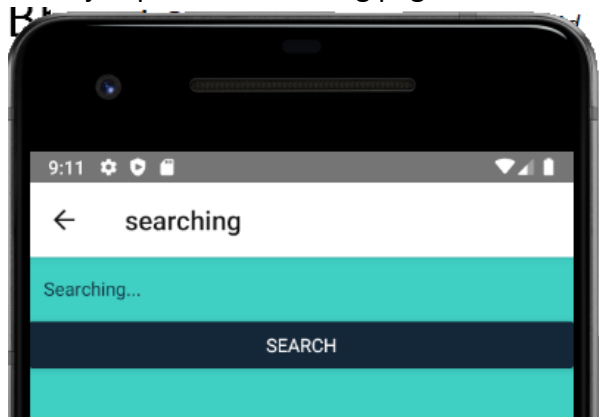
III. Type the keyword

IV. Press search

V. It will show all the item with that word
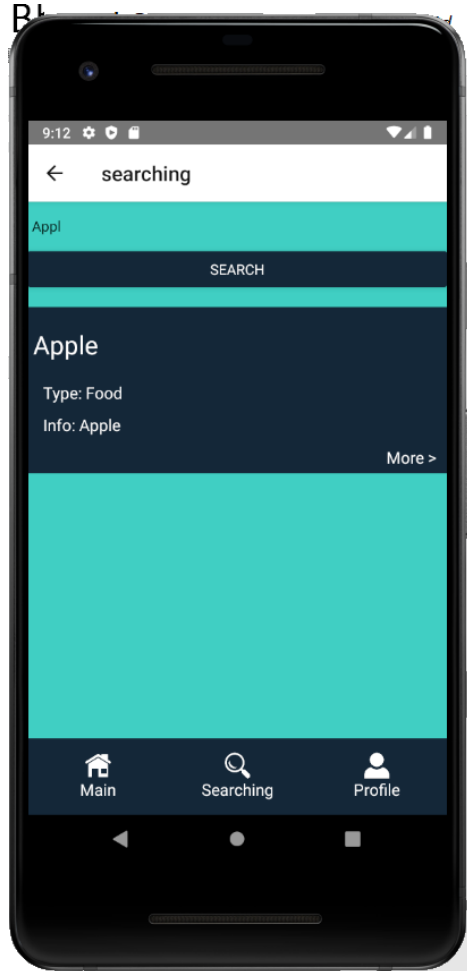
VI. Press one of the items you like
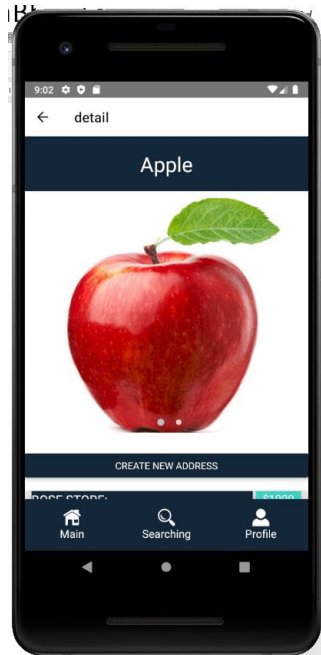
VII. It will jump to the detail page of the item

X. The address and price will show under
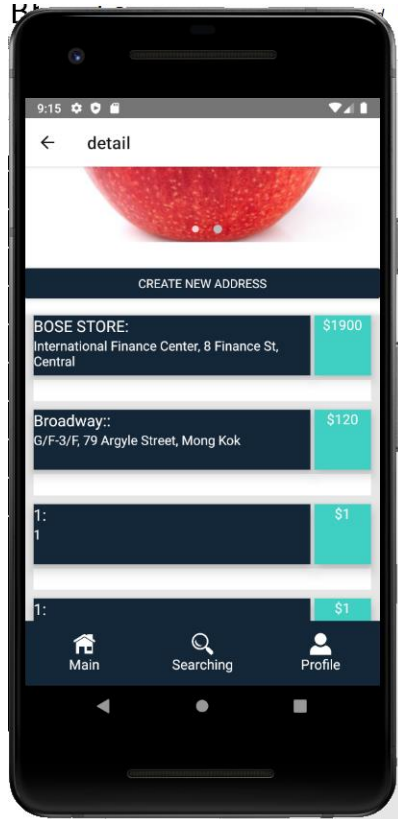
Create Address:

I. In the item detail page, press the create new address button
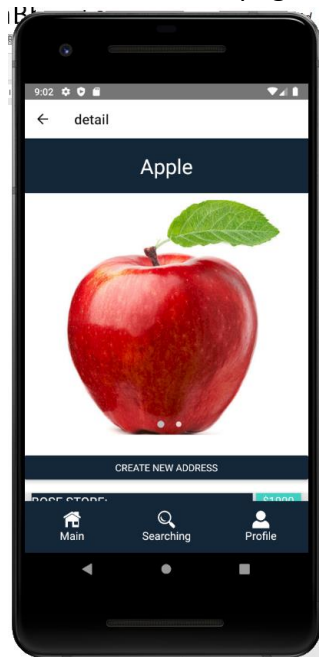
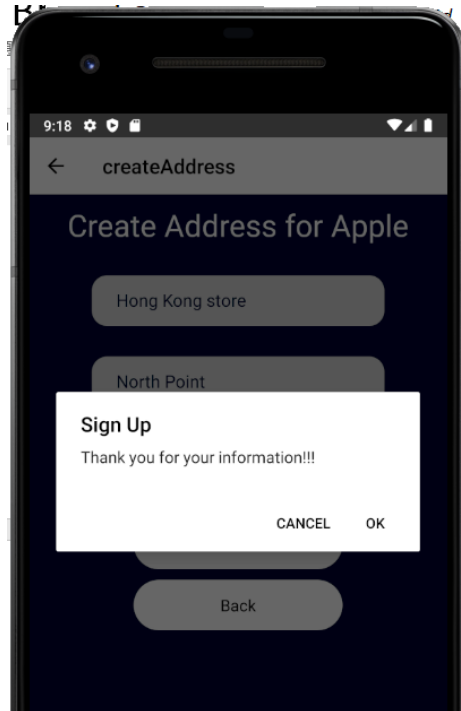II. It will jump to the create address page
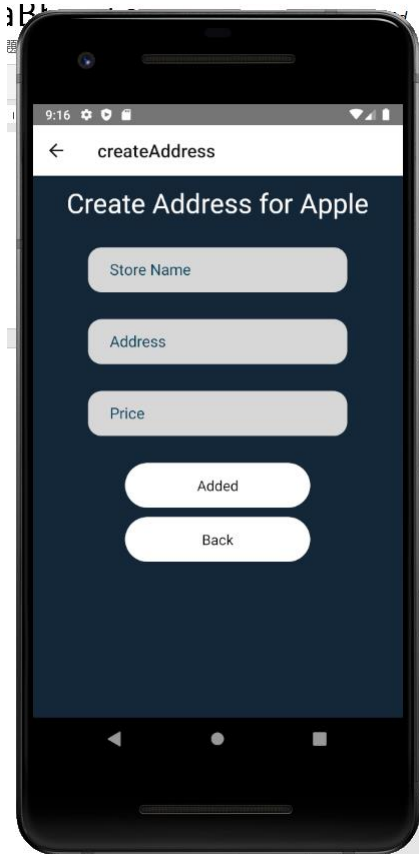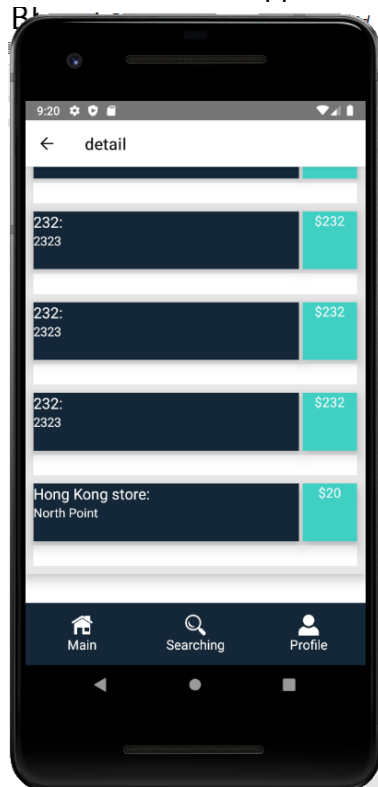
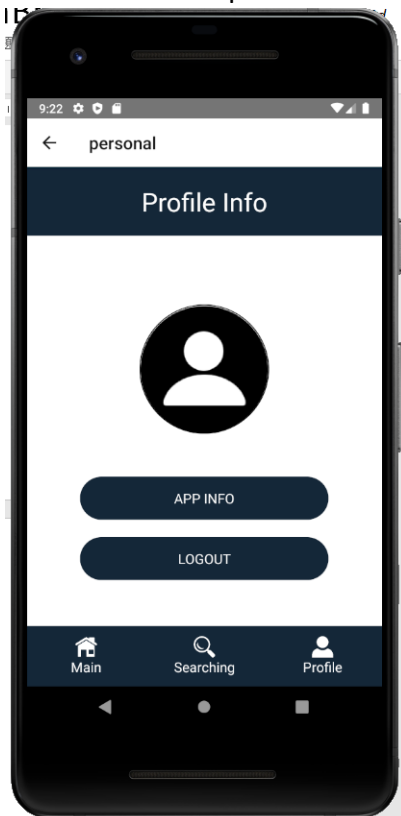III. Enter the input box
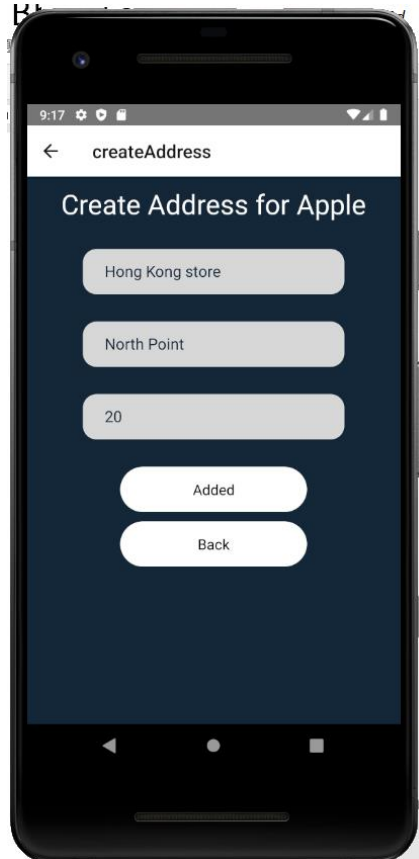
IV.   Press added

V.   Alert box is shown

VI.   Press OK

VII.   The data will appear in the app when you search the item again

App Information:

I. Press the button profile in the menu bar

II. Press the button App Info

III. The information will appear in a box

IV. Press back and it will back to the profile page