Mobile Application Development - Project Report

SOFE 4640U

Student Name	Student Number
Dhanushga Lionel	100616831
Allan Santosh	100557518
Mohammed Hameeduddin	100587893
Ayesha Farkhundah	100517461

A report submitted in fulfillment of the group project of Mobile Application Development in the Faculty of Engineering and Applied Science.

Professor: Dr. Anwar Abdalbari

Submitted to the University of Ontario Institute of Technology

December 2019

Table of Contents

Introduction

As technology progresses and get more widespread, so to does distributed systems grow. This is a vast and challenging field, but one with many benefits[1]. The end goal of a distributed system is to get a group of computing devices, work together, so as to appear as a single computer.

A distributed system contains multiple nodes, which are physically separate, but are used together in a network [2]. Each node communicates to other nodes in the system, and handle processes [2].

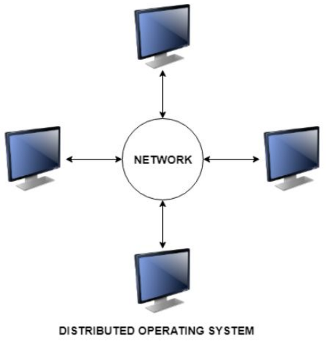


Figure 1: Distributed System

There are 2 types of distributed systems, client/server systems, or peer-to-peer systems [2].

Client/Server Systems

In this type of system, a client will request for a resource from the server. A server can have multiple clients, but the inverse is not true, a client can only have one server. Both the client and server communicate through the network, which makes this a distributed system[2].

Peer-to-Peer Systems

In this type of system, all nodes are equal partners who share all resources. The tasks are equally divided between all the nodes[2].

Advantages

There are many advantages of distributed systems. These include scalability, as more nodes can be added. Distributed systems are also reliable, as a failure of a node does not affect the whole system, nodes can still communicate with each other. Distributed systems are also fast.

Disadvantages

Security is a huge issue in distributed systems, since each node needs to be secured. Another disadvantage is that data might get lost in transmission. There are also more components, which means more points for failure. Overloading can also occur if every node tries to send data at the same time.

Background & Related Work

Existing solutions do exist [1]. However, these solutions are not fully fleshed out, they lack the ability to perform analysis on the data available. Existing solutions also don't allow for automatic trend detection. The main focus of this project will be data analysis, finding business trends based on time of year and geographic location of sales. The project will get continuous data over the internet from stores part of the organization's network.

Cars4U aims to eliminate this problem by providing students another route for transportation, while making money at the side. With Car4U, students can search their cities for car rentals for cheap being rented out by other students too. This app allows the user to search through distinct filters, so they don't have to settle. The criteria include: location, time, model, type, seats, and max price. Similarly, if a student wants to make money they can simply list their car with the information on the app and wait for a notification that their car has been booked.

For Car4U, it was based on elements of Uber, AirBnB and Turo. With Uber and AirBnB, we see the dual nature of the app, one for users to make money, and another for users to spend money. This application is very much like AirBnb, because users can rent out or rent a car, instead of houses.

Proposed Solution

Architecture

Site Structure

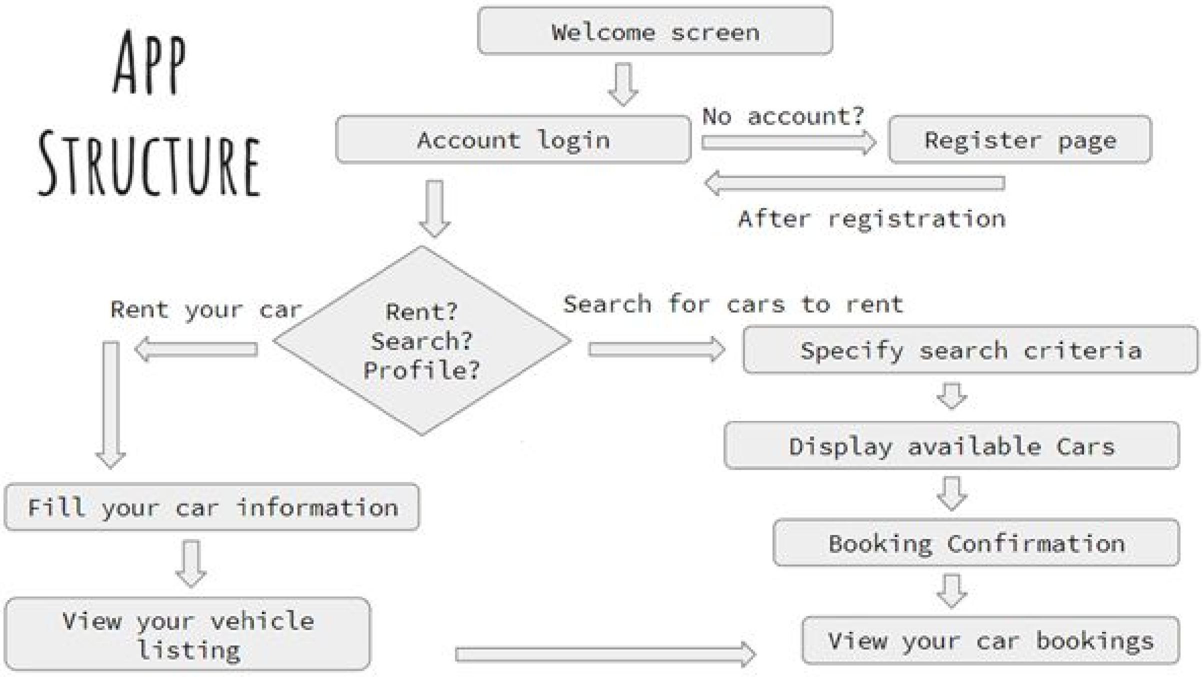


Figure 2: App structure

As shown in the figure above, there are 2 basic ways to use this app, renting out your car, or renting a car. First a user will have to login or register. Once a user logs in, they are able to rent out a car, or search for cars to rent.

If a user want to rent out their cars, they first have to fill out car information, such as the make, model, number of seats, price, and images to boost cars. After that they can confirm, and the car will be submitted for review, which they can view.

If a user needs to rent out a car, they can first specify criteria, such as the make,model, date and price. After criteria specification, cars that match that query will be displayed in an organized manner. The user can select and confirm a booking. After that they can view their booking.

Objectives

Cars4U aims to eliminate this problem by providing students another route for transportation, while making money at the side. With Car4U, students can search their cities for car rentals for cheap being rented out by other students too. This app allows the user to search through distinct filters, so they don't have to settle. The criteria include: location, time, model, type, seats, and max price. Similarly, if a student wants to make money they can simply list their car with the information on the app and wait for a notification that their car has been booked.

Components

The following components were used in creating Car4U.

User Interface:

We used android studio's Layout Editor to build a UI for our mobile app. Utilizing minimalist design to increase usability. The inclusion of a dashboard like page once your log on, is also set up to make navigation easier. The search result also provides the user with a picture of a car, to increase reliability.

Multiple Activities:

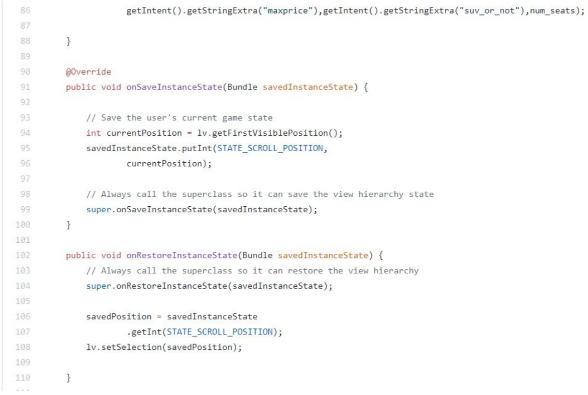
We had multiple Activities as pages for the users to be drawn to representing their request. The Activities were: booking_confirmation, car_search,confirmation_page,login_page,main,mybookingsand_listings, register_page, rent_my_car, search_results,selection_page,splash_screen,dialog,light_layout_search_results,listing_layouts.

Intents:

Intenets were used to give user a passageway to navigate through the multiple activities presented above. They were also used to send information to values to pages. Such as search criterias, and user_name.

Save state mechanism:

While you change orientations the program will save its state. The forms you filled out will have been refilled by the application. The program sends the values via post method in php and then it runs the query using the php files and outputs the json file. The json file is then read to display the results. Bundles were also used to save in-app progress.

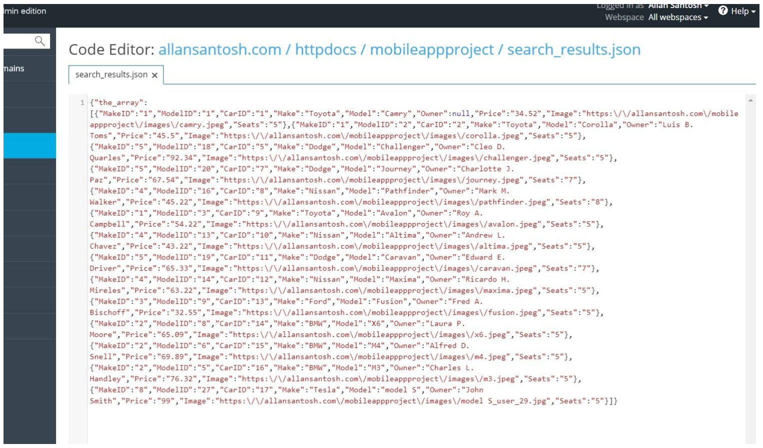


```
                getIntent().getStringExtra("maxprice"),getIntent().getStringExtra("suv_or_not"),num_seats);

    }

    @Override
    public void onSaveInstanceState(Bundle savedInstanceState) {

        // Save the user's current game state
        int currentPosition = lv.getFirstVisiblePosition();
        savedInstanceState.putInt(STATE_SCROLL_POSITION,
                currentPosition);

        // Always call the superclass so it can save the view hierarchy state
        super.onSaveInstanceState(savedInstanceState);
    }

    public void onRestoreInstanceState(Bundle savedInstanceState) {
        // Always call the superclass so it can restore the view hierarchy
        super.onRestoreInstanceState(savedInstanceState);

        savedPosition = savedInstanceState
                .getInt(STATE_SCROLL_POSITION);
        lv.setSelection(savedPosition);

    }
```

Figure 3: Save state example

The usage of internet resources:

We simply utilized jason. It was used to transmit data across clients.

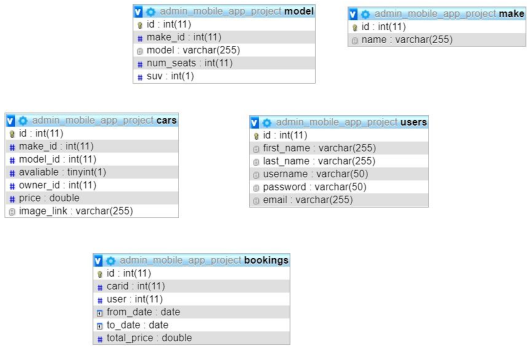

min edition
Logged in as Allan Santosh
Webspace All webspaces
Help
mains
Code Editor: allansantosh.com / httpdocs / mobileappproject / search_results.json
search_results.json ×

```
{"the_array":
[{"MakeID":"1","ModelID":"1","CarID":"1","Make":"Toyota","Model":"Camry","Owner":null,"Price":"34.52","Image":"https:\/\/allansantosh.com\/mobile
appproject\/images\/camry.jpeg","Seats":"5"},{"MakeID":"1","ModelID":"2","CarID":"2","Make":"Toyota","Model":"Corolla","Owner":"Luis B.
Toms","Price":"45.5","Image":"https:\/\/allansantosh.com\/mobileappproject\/images\/corolla.jpeg","Seats":"5"},
{"MakeID":"5","ModelID":"18","CarID":"5","Make":"Dodge","Model":"Challenger","Owner":"Cleo D.
Quarles","Price":"92.34","Image":"https:\/\/allansantosh.com\/mobileappproject\/images\/challenger.jpeg","Seats":"5"},
{"MakeID":"5","ModelID":"20","CarID":"7","Make":"Dodge","Model":"Journey","Owner":"Charlotte J.
Paz","Price":"67.54","Image":"https:\/\/allansantosh.com\/mobileappproject\/images\/journey.jpeg","Seats":"7"},
{"MakeID":"4","ModelID":"16","CarID":"8","Make":"Nissan","Model":"Pathfinder","Owner":"Mark M.
Walker","Price":"45.22","Image":"https:\/\/allansantosh.com\/mobileappproject\/images\/pathfinder.jpeg","Seats":"8"},
{"MakeID":"1","ModelID":"3","CarID":"9","Make":"Toyota","Model":"Avalon","Owner":"Roy A.
Campbell","Price":"54.22","Image":"https:\/\/allansantosh.com\/mobileappproject\/images\/avalon.jpeg","Seats":"5"},
{"MakeID":"4","ModelID":"13","CarID":"10","Make":"Nissan","Model":"Altima","Owner":"Andrew L.
Chavez","Price":"43.22","Image":"https:\/\/allansantosh.com\/mobileappproject\/images\/altima.jpeg","Seats":"5"},
{"MakeID":"5","ModelID":"19","CarID":"11","Make":"Dodge","Model":"Caravan","Owner":"Edward E.
Driver","Price":"65.33","Image":"https:\/\/allansantosh.com\/mobileappproject\/images\/caravan.jpeg","Seats":"7"},
{"MakeID":"4","ModelID":"14","CarID":"12","Make":"Nissan","Model":"Maxima","Owner":"Ricardo H.
Mireles","Price":"63.22","Image":"https:\/\/allansantosh.com\/mobileappproject\/images\/maxima.jpeg","Seats":"5"},
{"MakeID":"3","ModelID":"9","CarID":"13","Make":"Ford","Model":"Fusion","Owner":"Fred A.
Bischoff","Price":"32.55","Image":"https:\/\/allansantosh.com\/mobileappproject\/images\/fusion.jpeg","Seats":"5"},
{"MakeID":"2","ModelID":"8","CarID":"14","Make":"BMW","Model":"X6","Owner":"Laura P.
Moore","Price":"65.09","Image":"https:\/\/allansantosh.com\/mobileappproject\/images\/x6.jpeg","Seats":"5"},
{"MakeID":"2","ModelID":"6","CarID":"15","Make":"BMW","Model":"M4","Owner":"Alfred D.
Snell","Price":"69.89","Image":"https:\/\/allansantosh.com\/mobileappproject\/images\/m4.jpeg","Seats":"5"},
{"MakeID":"2","ModelID":"5","CarID":"16","Make":"BMW","Model":"M3","Owner":"Charles L.
Handley","Price":"76.32","Image":"https:\/\/allansantosh.com\/mobileappproject\/images\/m3.jpeg","Seats":"5"},
{"MakeID":"8","ModelID":"27","CarID":"17","Make":"Tesla","Model":"model S","Owner":"John
Smith","Price":"99","Image":"https:\/\/allansantosh.com\/mobileappproject\/images\/model S_user_29.jpg","Seats":"5"}]}
```

Figure 4:Json File

Cloud database transactions:

The application uses a cloud, SQL database. The database stored information such as the username, password, email. Then as each user adds their car for rent it also gets saved in a different table. For the search function.



admin_mobile_app_project **model**
id : int(11)
make_id : int(11)
model : varchar(255)
num_seats : int(11)
suv : int(1)

admin_mobile_app_project **make**
id : int(11)
name : varchar(255)

admin_mobile_app_project **cars**
id : int(11)
make_id : int(11)
model_id : int(11)
avaliable : tinyint(1)
owner_id : int(11)
price : double
image_link : varchar(255)

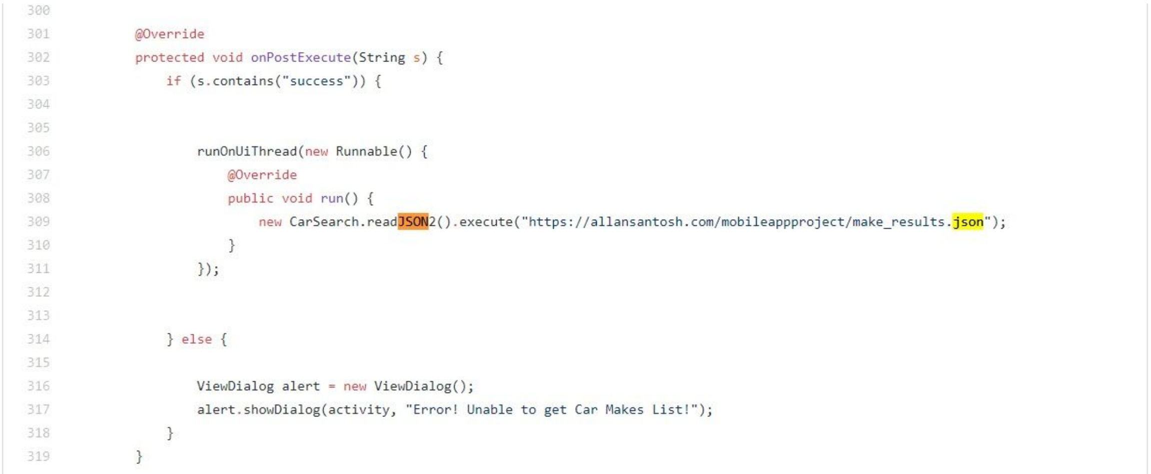
admin_mobile_app_project **users**
id : int(11)
first_name : varchar(255)
last_name : varchar(255)
username : varchar(50)
password : varchar(50)
email : varchar(255)

admin_mobile_app_project **bookings**
id : int(11)
carid : int(11)
user : int(11)
from_date : date
to_date : date
total_price : double

Figure 5: Schema

Multi-thread programming:

Multi-thread was used to call upon internet resources.



```

        @Override
        protected void onPostExecute(String s) {
            if (s.contains("success")) {


                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        new CarSearch.readJSON2().execute("https://allansantosh.com/mobileappproject/make_results.json");
                    }
                });


            } else {

                ViewDialog alert = new ViewDialog();
                alert.showDialog(activity, "Error! Unable to get Car Makes List!");
            }
        }
```

Figure 6: Runnable

Limitation

While our app is very close to being a complete solution, there are some things we can still make better. Firstly, we can make it a cross platform app. We plan to make the app cross-platform by making using flutter and dart, so we can cater to both Android and IOS markets. Secondly, we can use Google Firebase as our cloud database. This will make our app more responsive, and load faster. Thirdly, we can use a map functionality, instead of just typing a specific city, we can use Google maps for the car pickup locations. Fourthly, we can implement a way to send a confirmation code when signing up for an account, and set up 2-factor authentication to prevent bogus accounts. Finally we can implement a function such that the camera can be directly accessed from the app itself, rather than taking a picture, going to the app, and searching for the image. Taking a picture from the app itself, will make our app 35% more usable.

Conclusion:

To conclude, our app is a very big accomplishment, and our group is very happy with the final result, as was the professor. We implemented as many functionalities in a short amount of time, to make the app as complete as possible.