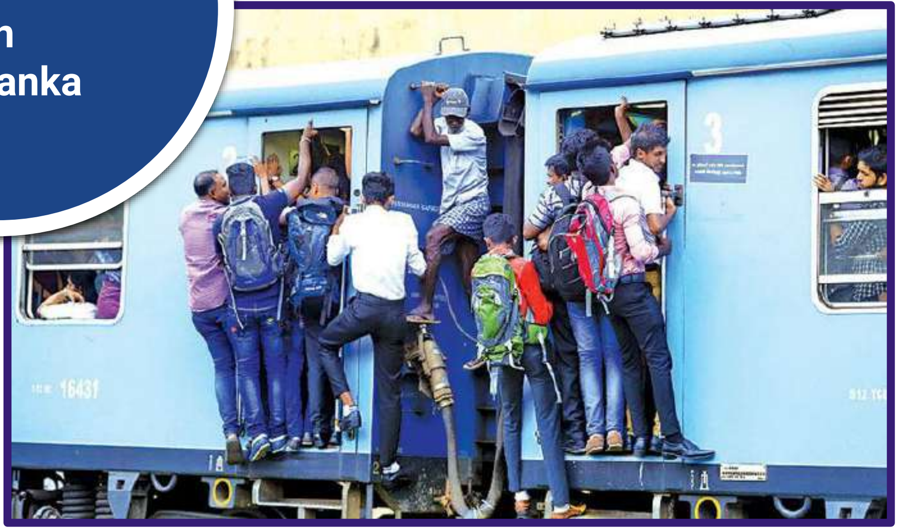
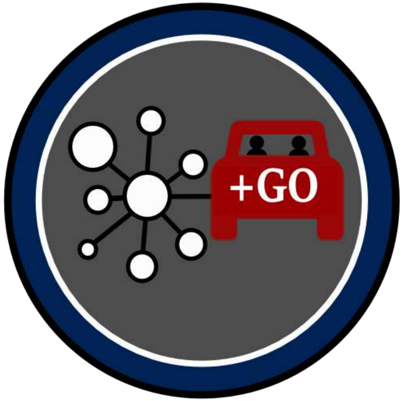
Public Transportation in Sri Lanka

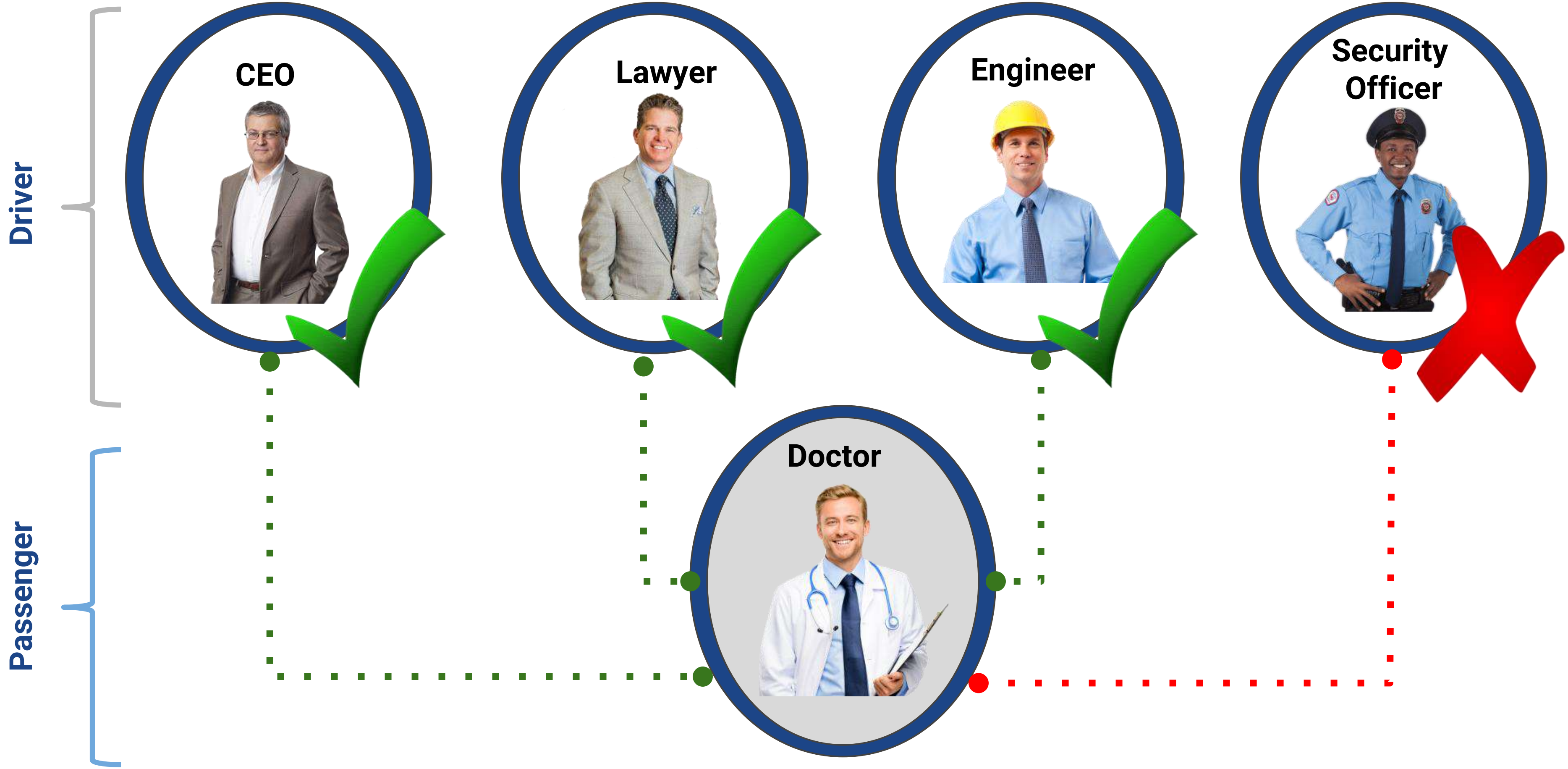




Plus Go

Intelligent Complementary Ride-Sharing System

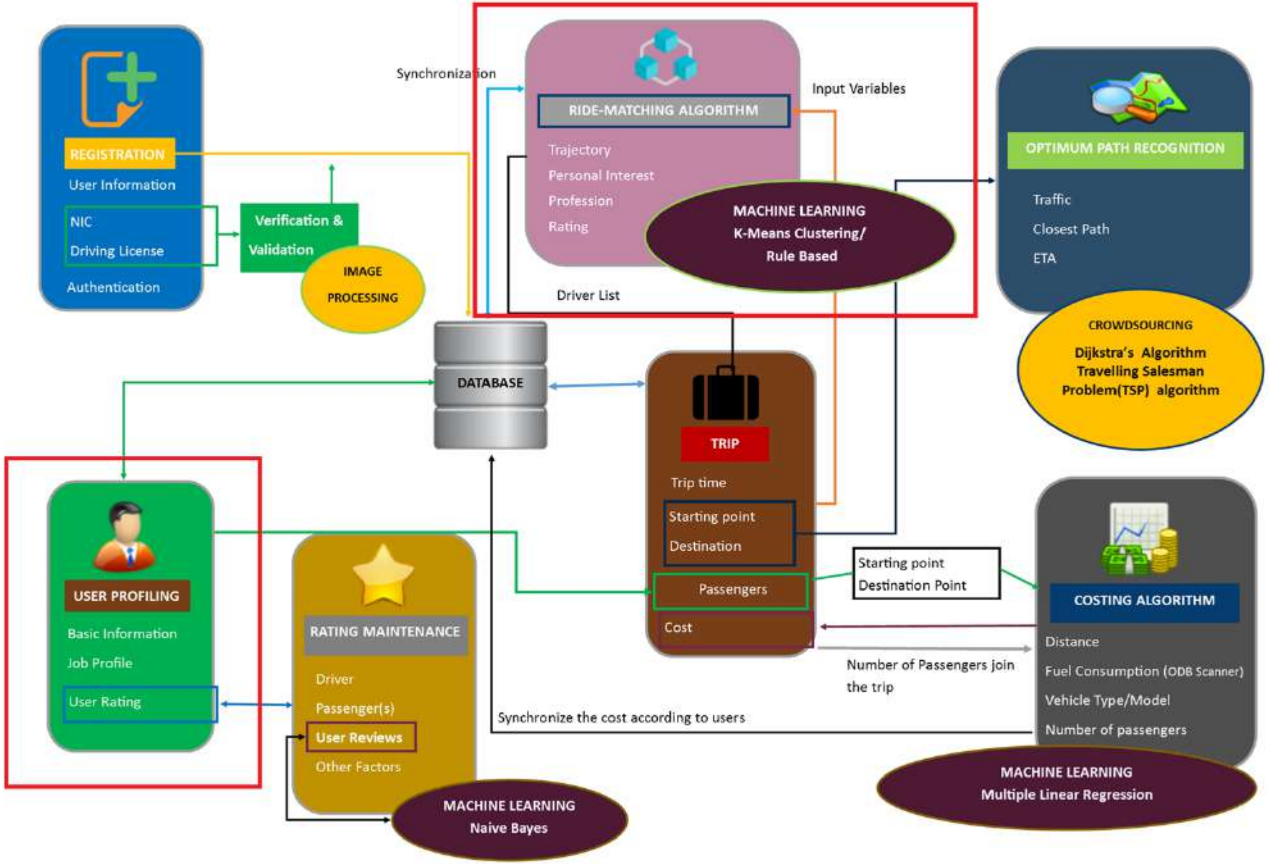
Mechanism of Suggesting Most Suitable Drivers



DEMO

User Profile Management

High Level Diagram



Objectives

- Suggest most suitable drivers for a passenger based on the Profession and Preferences of the passenger. Thereby no random suggestions are given.

  Preferences Include:

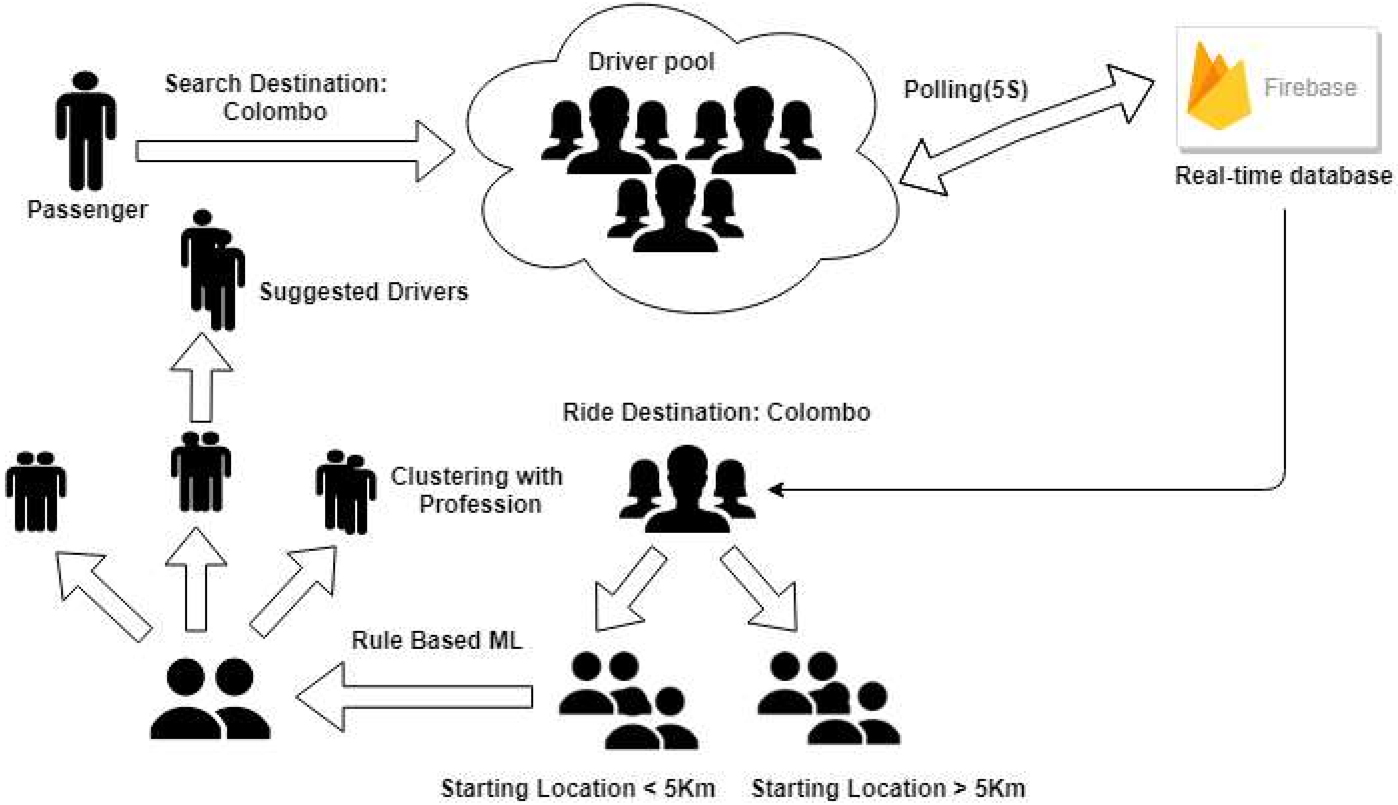
  - Smoking Condition
  - Music Lover
  - Language Preference
  - Motion Sickness
  - Gender Preference
  - Like Quietness

- Spouse/Guardian can see trip history of passenger and report any suspicious drivers.

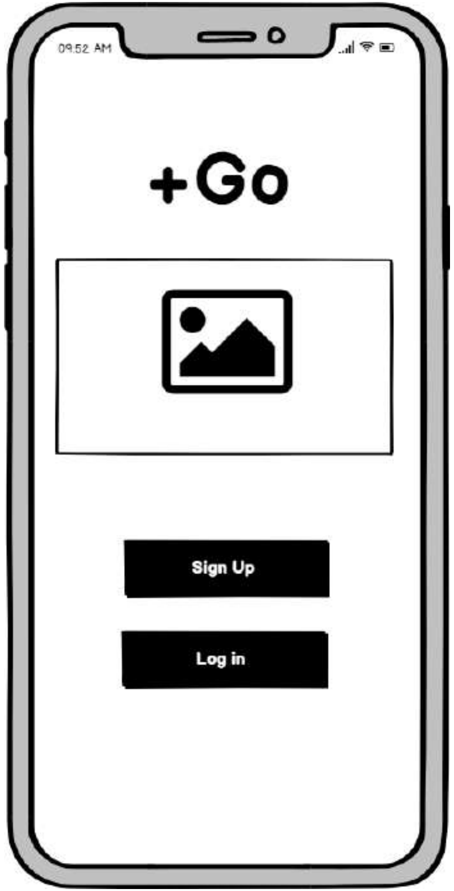
Research Gap

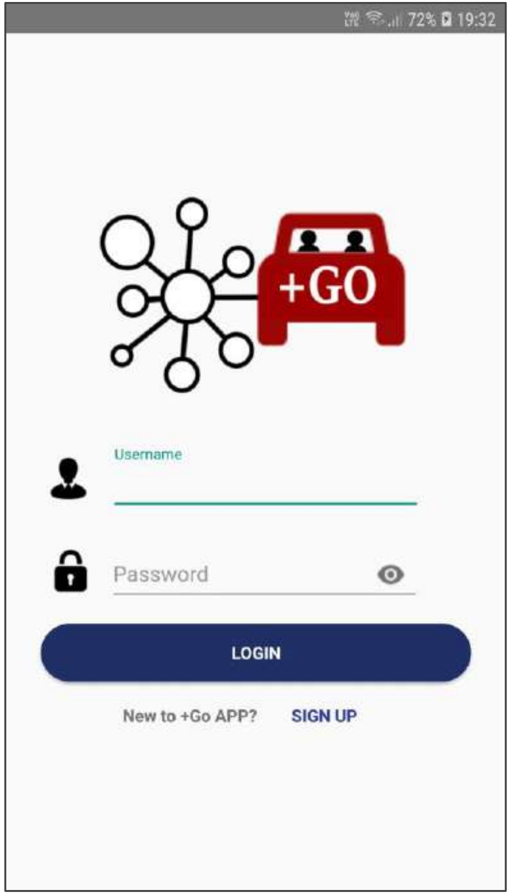
Features	UBER	UDIO	Carpooling.lk	RideShare.lk	+GO
Focused only on professionals	✗	✗	✗	✗	✓
Matching the passengers' profile with drivers	✗	✗	✗	✗	✓
Allow the spouse/guardian to check the passenger’s trip details	✗	✗	✗	✗	✓
Focus on Gender Preference to provide more customize suggestions	✗	✓	✗	✗	✓

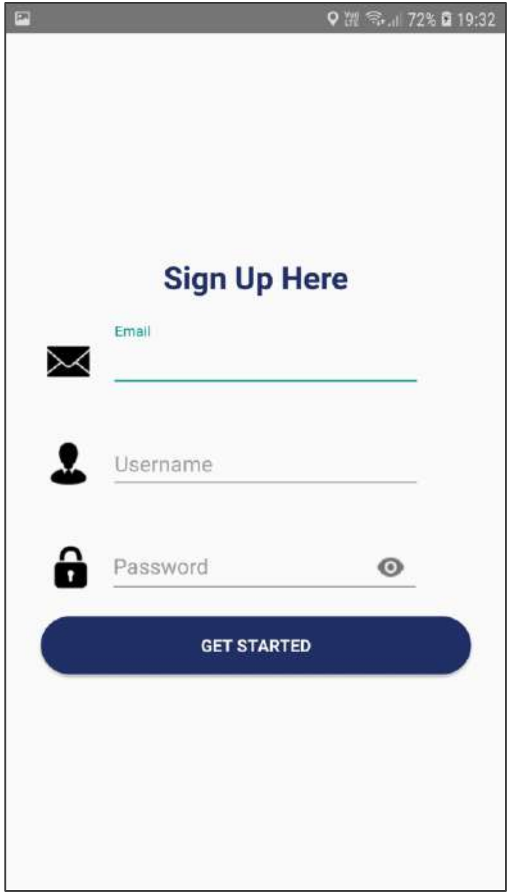
Flow of the User Profile Management



Prototype vs Implementation

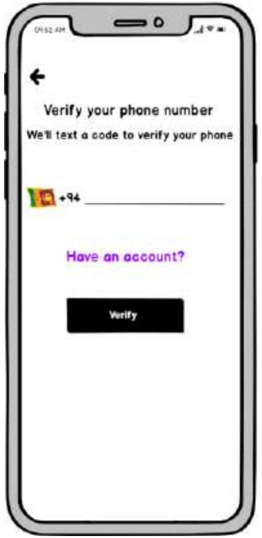






Prototype vs Implementation

Mobile Number Verification



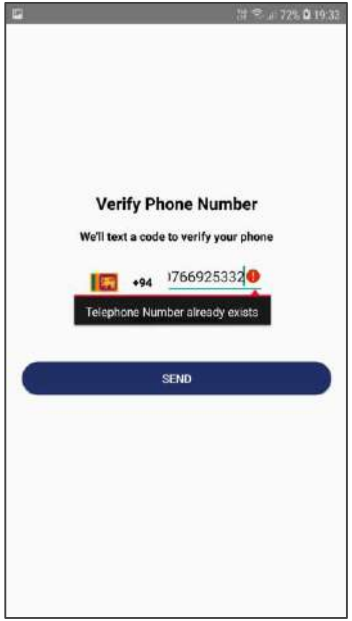
Verify your phone number

We'll text a code to verify your phone

+94

Have an account?

Verify



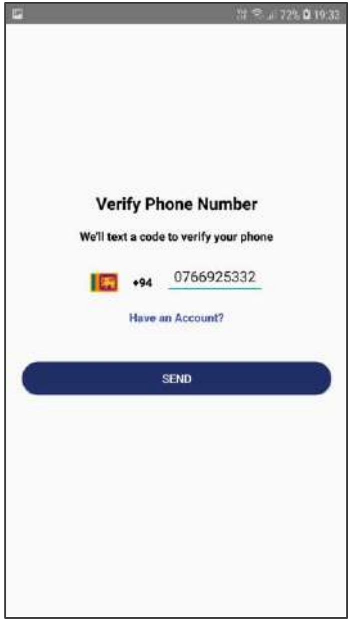
Verify Phone Number

We'll text a code to verify your phone

+94 0766925332

Telephone Number already exists

SEND



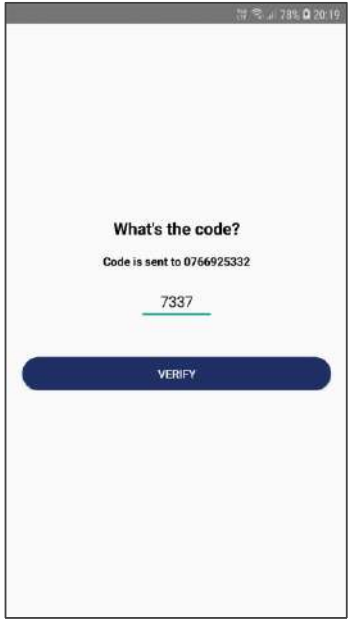
Verify Phone Number

We'll text a code to verify your phone

+94 0766925332

Have an Account?

SEND



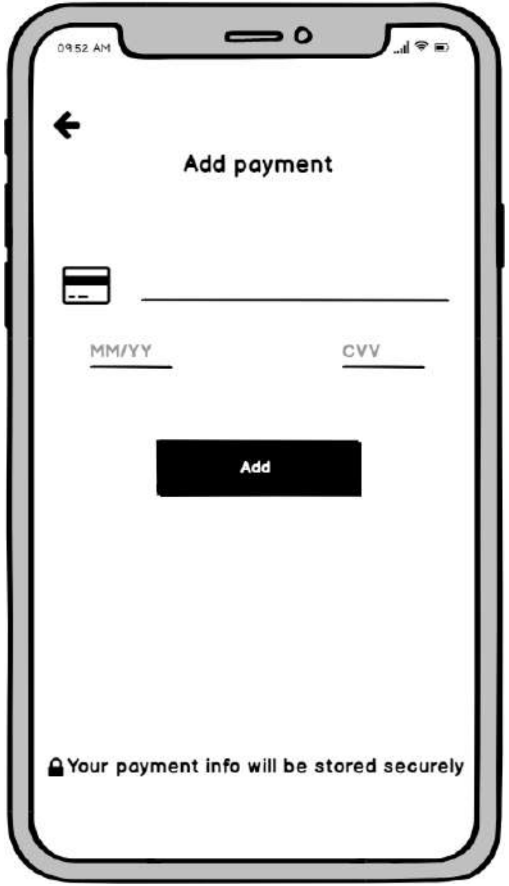
What's the code?

Code is sent to 0766925332

7337

VERIFY

Prototype vs Implementation



09:52 AM

Add payment

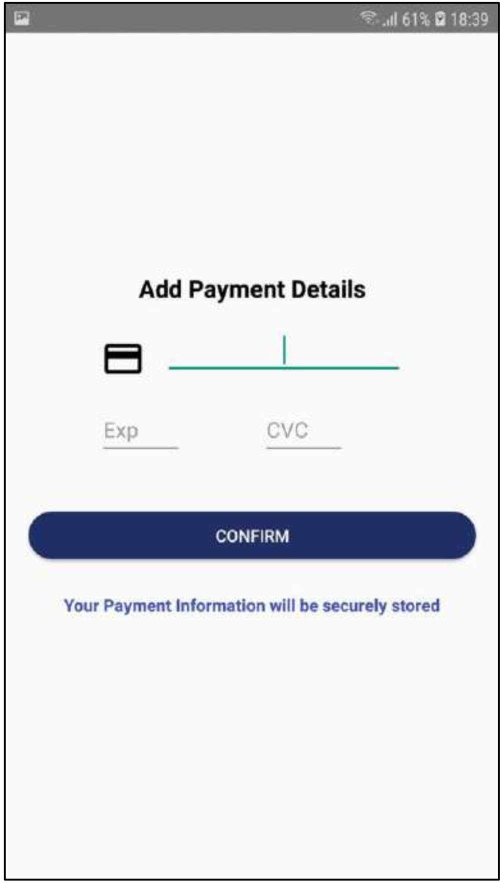
MM/YY

CVV

Add

Your payment info will be stored securely

Adding Payment Method



61% 18:39

Add Payment Details

Exp

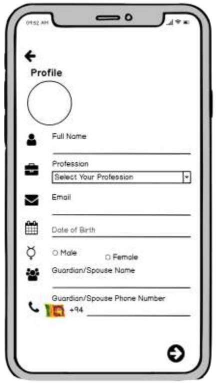
CVC

CONFIRM

Your Payment Information will be securely stored

Prototype vs Implementation

Profile Creation



Profile

Full Name

Profession

Select Your Profession

Email

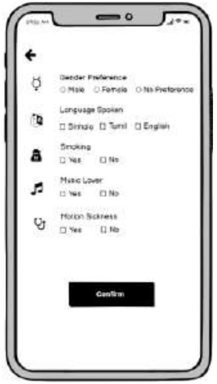
Date of Birth

○ Male ○ Female

Guardian/Spouse Name

Guardian/Spouse Phone Number

+94



Gender Preference

○ Male ○ Female ○ No Preference

Language Spoken

□ Sinhala □ Tamil □ English

Smoking

□ Yes □ No

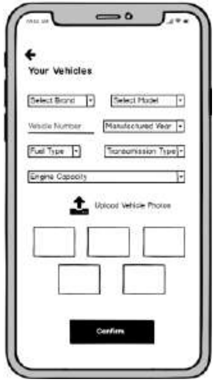
Music Lover

□ Yes □ No

Motion Sickness

□ Yes □ No

Confirm



Your Vehicles

Select Brand

Select Model

Vehicle Number

Manufactured Year

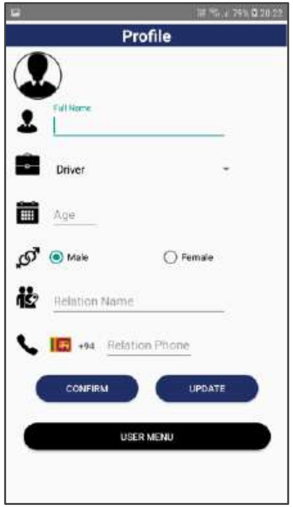
Fuel Type

Transmission Type

Engine Capacity

Upload Vehicle Photos

Confirm



Profile

Full Name

Driver

Age

● Male ○ Female

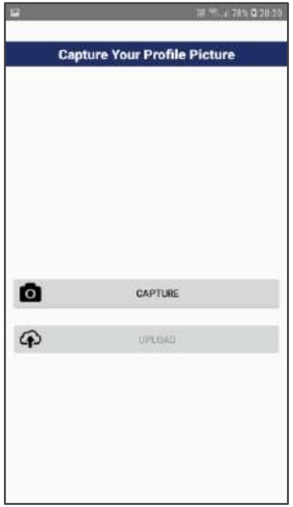
Relation Name

+94 Relation Phone

CONFIRM

UPDATE

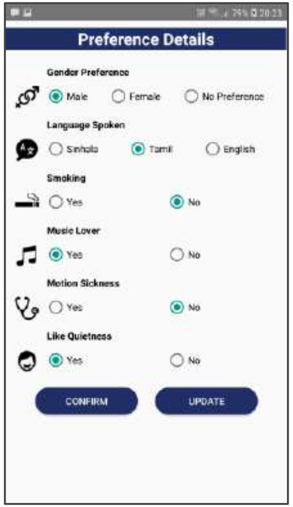
USER MENU



Capture Your Profile Picture

CAPTURE

UPLOAD



Preference Details

Gender Preference

● Male ○ Female ○ No Preference

Language Spoken

○ Sinhala ● Tamil ○ English

Smoking

○ Yes ● No

Music Lover

● Yes ○ No

Motion Sickness

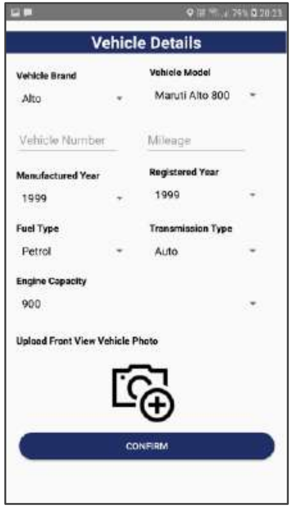
○ Yes ● No

Like Quietness

● Yes ○ No

CONFIRM

UPDATE



Vehicle Details

Vehicle Brand: Alto

Vehicle Model: Maruti Alto 800

Vehicle Number

Mileage

Manufactured Year: 1999

Registered Year: 1999

Fuel Type: Petrol

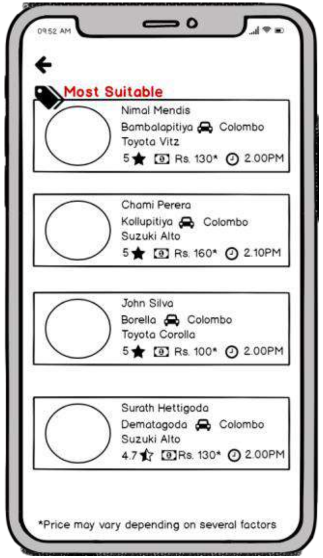
Transmission Type: Auto

Engine Capacity: 900

Upload Front View Vehicle Photo

CONFIRM

Prototype vs Implementation



Most Suitable

Nimal Mendis
Bambalapitiya Colombo
Toyota Vitz
5 ★ Rs. 130* 2.00PM

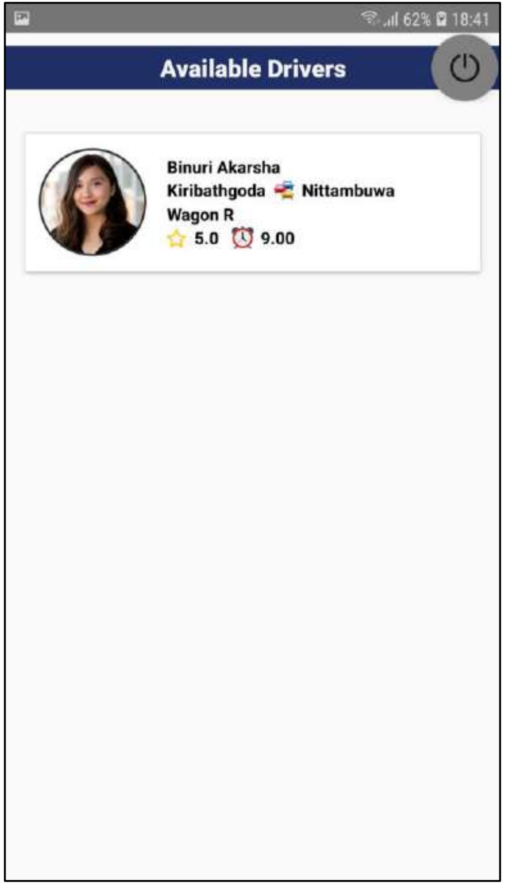
Chami Perera
Kollupitiya Colombo
Suzuki Alto
5 ★ Rs. 160* 2.10PM

John Silva
Borella Colombo
Toyota Corolla
5 ★ Rs. 100* 2.00PM

Surath Hettigoda
Dematagoda Colombo
Suzuki Alto
4.7 ★ Rs. 130* 2.00PM

*Price may vary depending on several factors

Suggesting Driver List



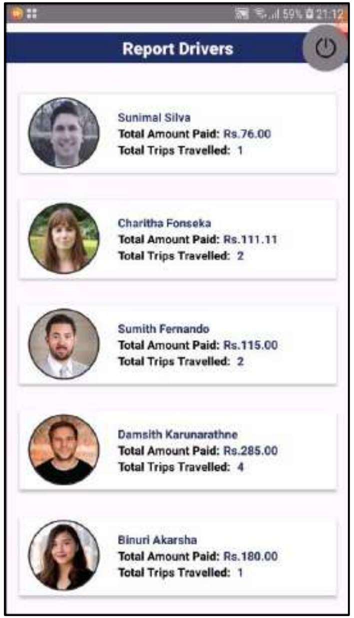
62% 18:41

Available Drivers

Binuri Akarsha
Kiribathgoda Nittambuwa
Wagon R
☆ 5.0 9.00

Prototype vs Implementation

Spouse/Guardian Reporting



Report Drivers

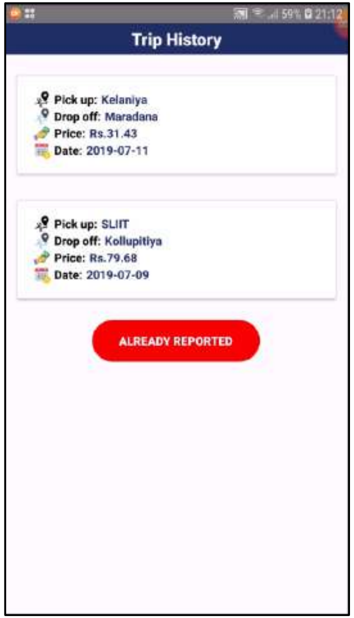
Sunimal Silva
Total Amount Paid: Rs.76.00
Total Trips Travelled: 1

Charitha Fonseka
Total Amount Paid: Rs.111.11
Total Trips Travelled: 2

Sumith Fernando
Total Amount Paid: Rs.115.00
Total Trips Travelled: 2

Damsith Karunarathne
Total Amount Paid: Rs.285.00
Total Trips Travelled: 4

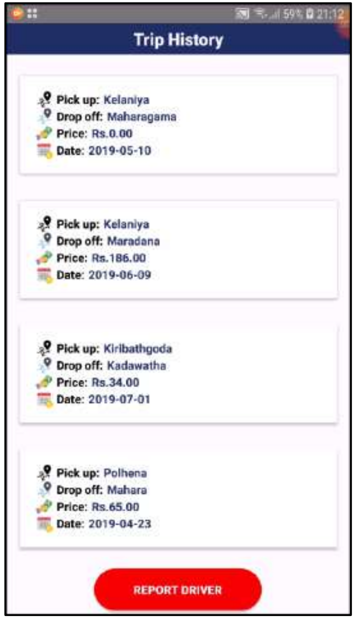
Binuri Akarsha
Total Amount Paid: Rs.180.00
Total Trips Travelled: 1



Trip History

Pick up: Kelaniya
Drop off: Maradana
Price: Rs.31.43
Date: 2019-07-11

Pick up: SLIIT
Drop off: Kollupitiya
Price: Rs.79.68
Date: 2019-07-09



Trip History

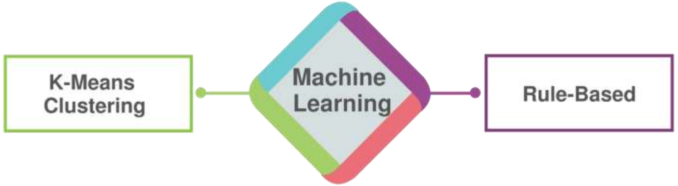
Pick up: Kelaniya
Drop off: Maharagama
Price: Rs.0.00
Date: 2019-05-10

Pick up: Kelaniya
Drop off: Maradana
Price: Rs.186.00
Date: 2019-06-09

Pick up: Kiribathgoda
Drop off: Kadawatha
Price: Rs.34.00
Date: 2019-07-01

Pick up: Polhena
Drop off: Mahara
Price: Rs.65.00
Date: 2019-04-23

The Specialized Area and the Use of Technologies

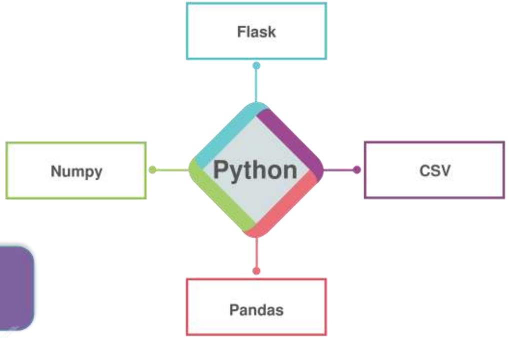


K-Means Clustering

Machine Learning

Rule-Based

Polling



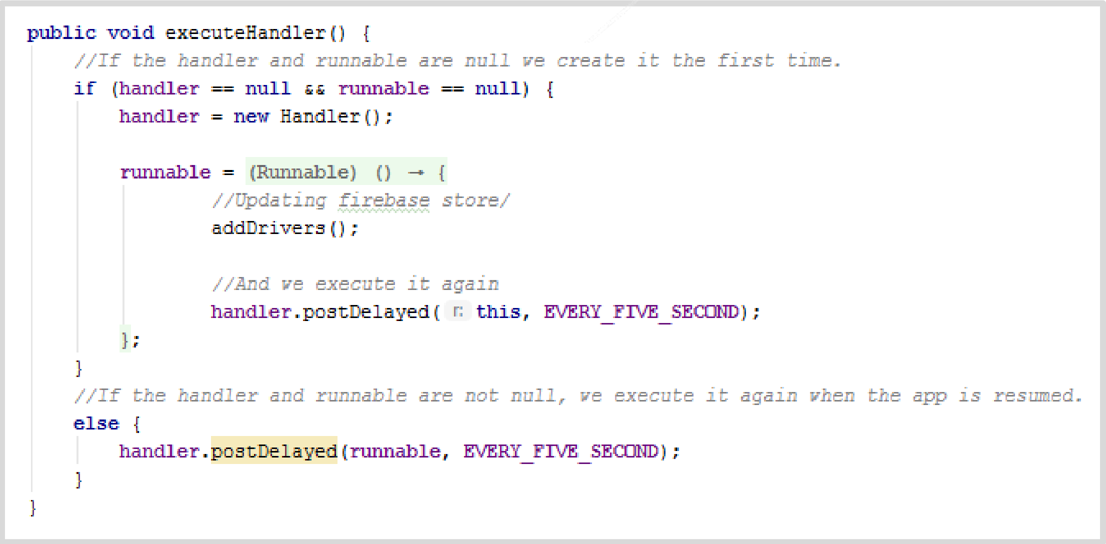
Flask

Numpy

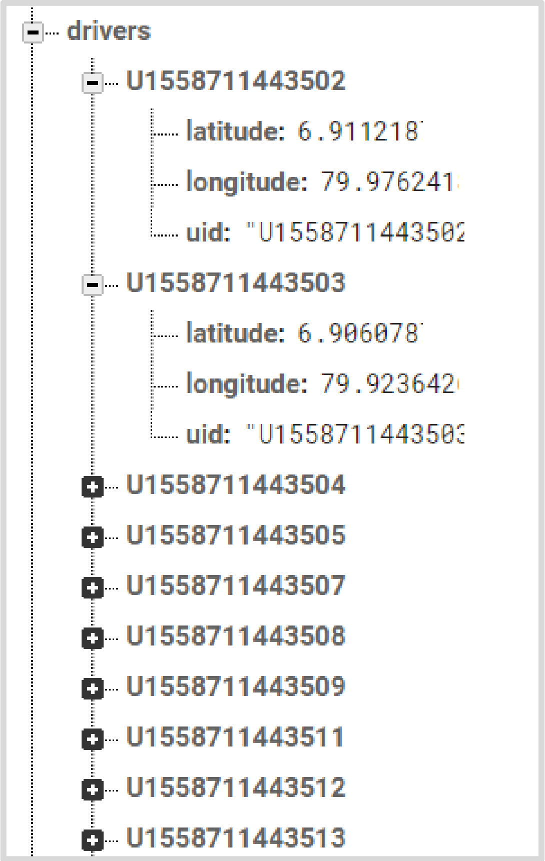
Python

CSV

Pandas

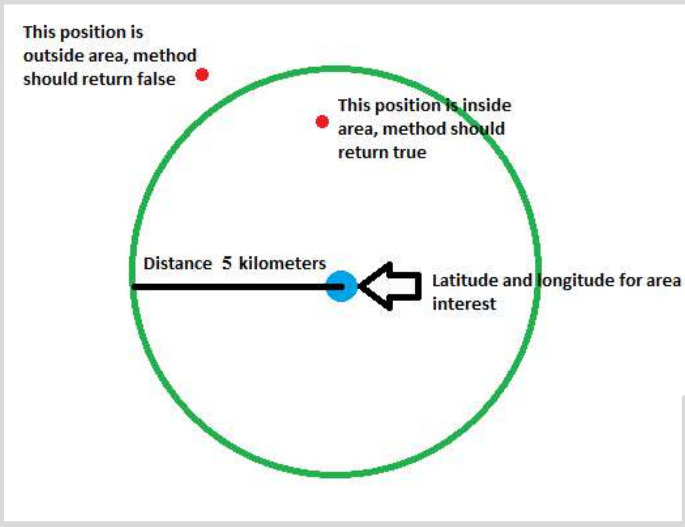


```
public void executeHandler() {
    //If the handler and runnable are null we create it the first time.
    if (handler == null && runnable == null) {
        handler = new Handler();

        runnable = (Runnable) () → {
                //Updating firebase store/
                addDrivers();

                //And we execute it again
                handler.postDelayed( r: this, EVERY_FIVE_SECOND);
        };
    }
    //If the handler and runnable are not null, we execute it again when the app is resumed.
    else {
        handler.postDelayed(runnable, EVERY_FIVE_SECOND);
    }
}
```

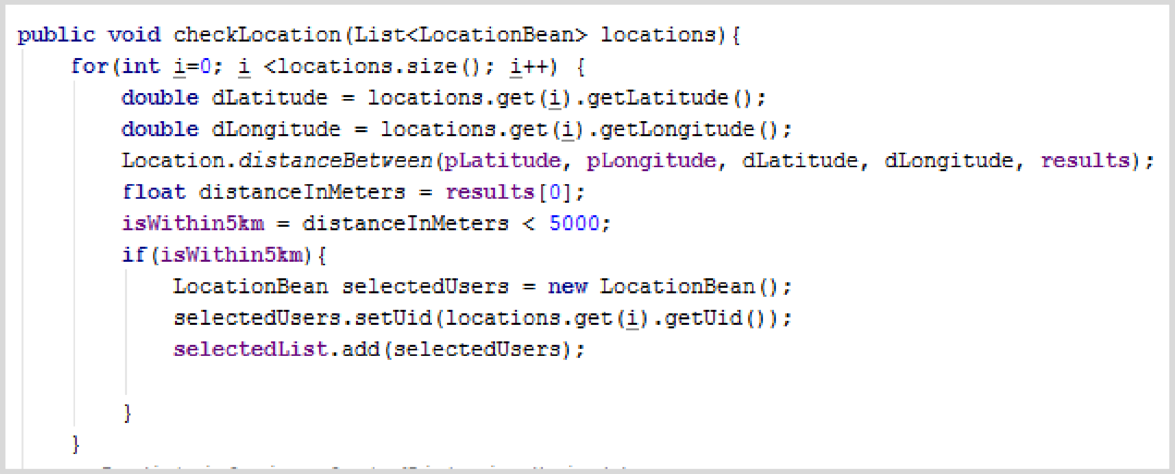


```
drivers
    U1558711443502
        latitude: 6.911218
        longitude: 79.976241
        uid: "U1558711443502
    U1558711443503
        latitude: 6.906078
        longitude: 79.923642
        uid: "U1558711443503
    U1558711443504
    U1558711443505
    U1558711443507
    U1558711443508
    U1558711443509
    U1558711443511
    U1558711443512
    U1558711443513
```

The Specialized Area and the Use of Technologies

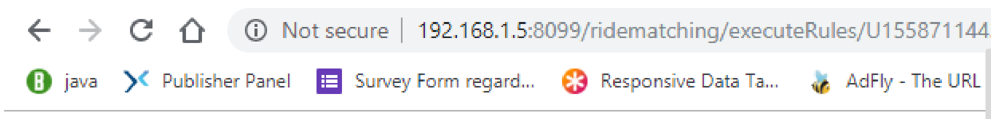

This position is outside area, method should return false
This position is inside area, method should return true
Distance 5 kilometers
Latitude and longitude for area of interest

Location Tracker

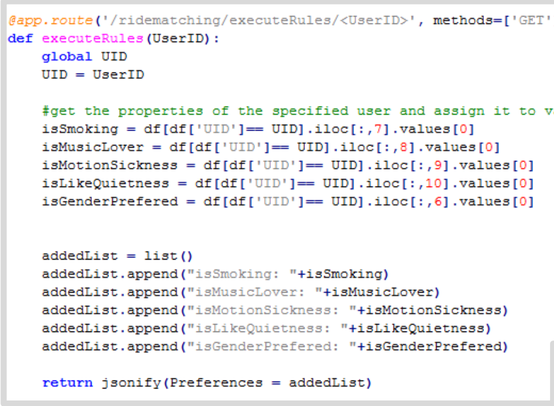


```
public void checkLocation(List<LocationBean> locations){
    for(int i=0; i <locations.size(); i++) {
        double dLatitude = locations.get(i).getLatitude();
        double dLongitude = locations.get(i).getLongitude();
        Location.distanceBetween(pLatitude, pLongitude, dLatitude, dLongitude, results);
        float distanceInMeters = results[0];
        isWithin5km = distanceInMeters < 5000;
        if(isWithin5km){
            LocationBean selectedUsers = new LocationBean();
            selectedUsers.setUid(locations.get(i).getUid());
            selectedList.add(selectedUsers);

        }
    }
```

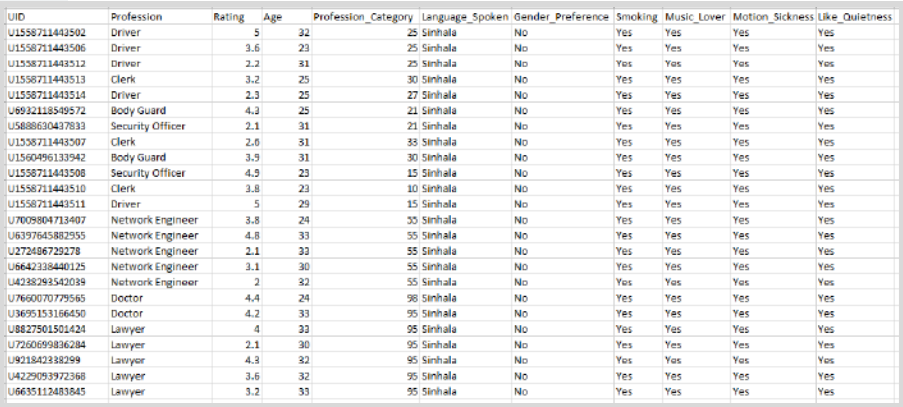
The Specialized Area and the Use of Technologies



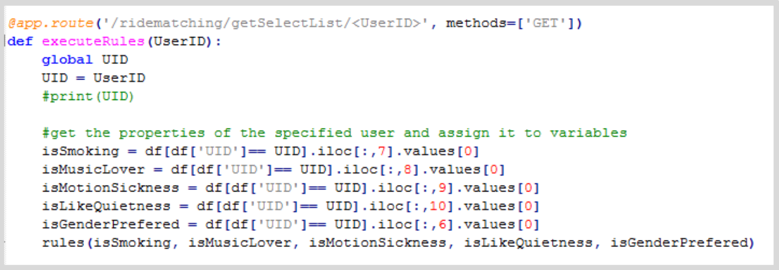
```
{
  "Preferences": [
    "isSmoking: Yes",
    "isMusicLover: Yes",
    "isMotionSickness: Yes",
    "isLikeQuietness: Yes",
    "isGenderPrefered: No"
  ]
}
```



```
@app.route('/ridematching/executeRules/<UserID>', methods=['GET'])
def executeRules(UserID):
    global UID
    UID = UserID

    #get the properties of the specified user and assign it to variables
    isSmoking = df[df['UID']== UID].iloc[:,7].values[0]
    isMusicLover = df[df['UID']== UID].iloc[:,8].values[0]
    isMotionSickness = df[df['UID']== UID].iloc[:,9].values[0]
    isLikeQuietness = df[df['UID']== UID].iloc[:,10].values[0]
    isGenderPrefered = df[df['UID']== UID].iloc[:,6].values[0]


    addedList = list()
    addedList.append("isSmoking: "+isSmoking)
    addedList.append("isMusicLover: "+isMusicLover)
    addedList.append("isMotionSickness: "+isMotionSickness)
    addedList.append("isLikeQuietness: "+isLikeQuietness)
    addedList.append("isGenderPrefered: "+isGenderPrefered)

    return jsonify(Preferences = addedList)
```

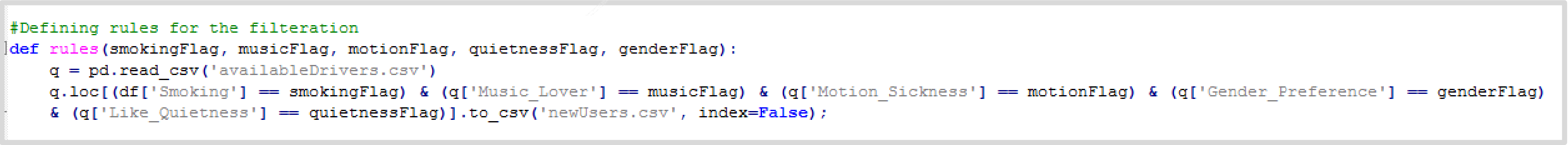


UID	Profession	Rating	Age	Profession_Category	Language_Spoken	Gender_Preference	Smoking	Music_Lover	Motion_Sickness	Like_Quietness
U1558711443502	Driver	5	32	25	Sinhala	No	Yes	Yes	Yes	Yes
U1558711443506	Driver	3.6	23	25	Sinhala	No	Yes	Yes	Yes	Yes
U1558711443512	Driver	2.2	31	25	Sinhala	No	Yes	Yes	Yes	Yes
U1558711443513	Clerk	3.2	25	30	Sinhala	No	Yes	Yes	Yes	Yes
U1558711443514	Driver	2.3	25	27	Sinhala	No	Yes	Yes	Yes	Yes
U6932118549572	Body Guard	4.3	25	21	Sinhala	No	Yes	Yes	Yes	Yes
U5888630437833	Security Officer	2.1	31	21	Sinhala	No	Yes	Yes	Yes	Yes
U1558711443507	Clerk	2.6	31	33	Sinhala	No	Yes	Yes	Yes	Yes
U1560496133942	Body Guard	3.9	31	30	Sinhala	No	Yes	Yes	Yes	Yes
U1558711443508	Security Officer	4.9	23	15	Sinhala	No	Yes	Yes	Yes	Yes
U1558711443510	Clerk	3.8	23	10	Sinhala	No	Yes	Yes	Yes	Yes
U1558711443511	Driver	5	29	15	Sinhala	No	Yes	Yes	Yes	Yes
U7009804713407	Network Engineer	3.8	24	55	Sinhala	No	Yes	Yes	Yes	Yes
U6397645882955	Network Engineer	4.8	33	55	Sinhala	No	Yes	Yes	Yes	Yes
U272486729278	Network Engineer	2.1	33	55	Sinhala	No	Yes	Yes	Yes	Yes
U6642338440125	Network Engineer	3.1	30	55	Sinhala	No	Yes	Yes	Yes	Yes
U4238293542039	Network Engineer	2	32	55	Sinhala	No	Yes	Yes	Yes	Yes
U7660070779565	Doctor	4.4	24	98	Sinhala	No	Yes	Yes	Yes	Yes
U3695153166450	Doctor	4.2	33	95	Sinhala	No	Yes	Yes	Yes	Yes
U8827501501424	Lawyer	4	33	95	Sinhala	No	Yes	Yes	Yes	Yes
U7260699836284	Lawyer	2.1	30	95	Sinhala	No	Yes	Yes	Yes	Yes
U921842338299	Lawyer	4.3	32	95	Sinhala	No	Yes	Yes	Yes	Yes
U4229093972368	Lawyer	3.6	32	95	Sinhala	No	Yes	Yes	Yes	Yes
U6635112483845	Lawyer	3.2	33	95	Sinhala	No	Yes	Yes	Yes	Yes



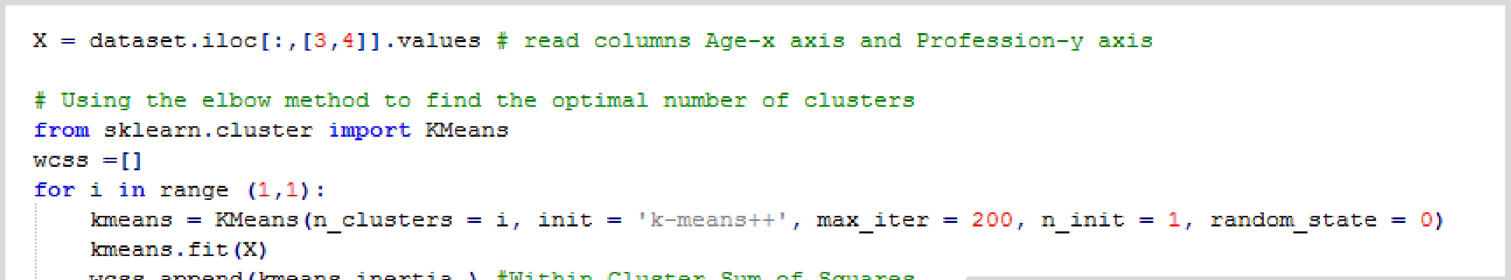
```
@app.route('/ridematching/getSelectList/<UserID>', methods=['GET'])
def executeRules(UserID):
    global UID
    UID = UserID
    #print(UID)

    #get the properties of the specified user and assign it to variables
    isSmoking = df[df['UID']== UID].iloc[:,7].values[0]
    isMusicLover = df[df['UID']== UID].iloc[:,8].values[0]
    isMotionSickness = df[df['UID']== UID].iloc[:,9].values[0]
    isLikeQuietness = df[df['UID']== UID].iloc[:,10].values[0]
    isGenderPrefered = df[df['UID']== UID].iloc[:,6].values[0]
    rules(isSmoking, isMusicLover, isMotionSickness, isLikeQuietness, isGenderPrefered)
```

Rule-Based Machine Learning



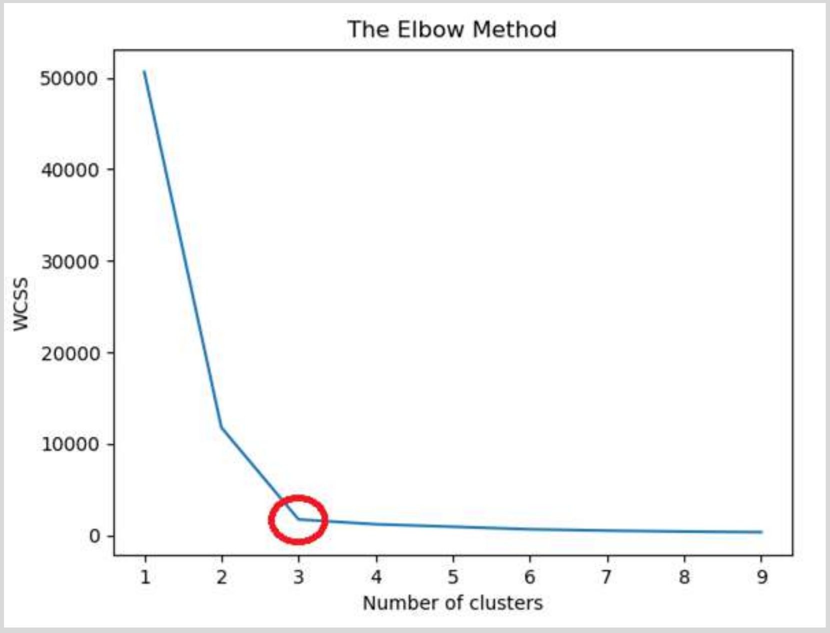
```
#Defining rules for the filteration
def rules(smokingFlag, musicFlag, motionFlag, quietnessFlag, genderFlag):
    q = pd.read_csv('availableDrivers.csv')
    q.loc[(df['Smoking'] == smokingFlag) & (q['Music_Lover'] == musicFlag) & (q['Motion_Sickness'] == motionFlag) & (q['Gender_Preference'] == genderFlag)
    & (q['Like_Quietness'] == quietnessFlag)].to_csv('newUsers.csv', index=False);
```

The Specialized Area and the Use of Technologies

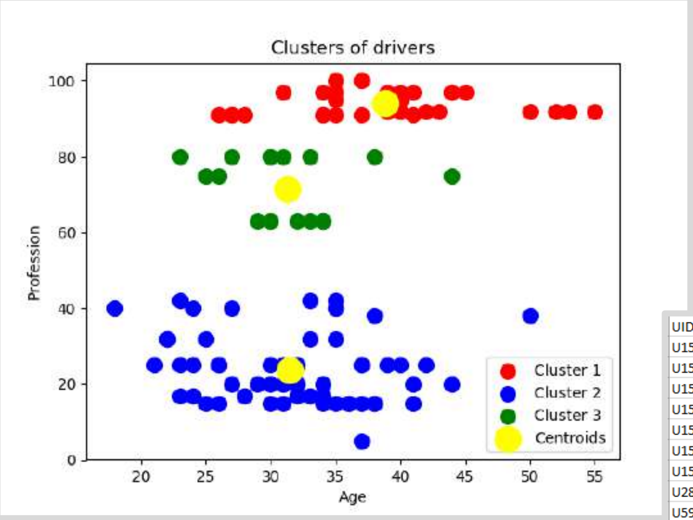


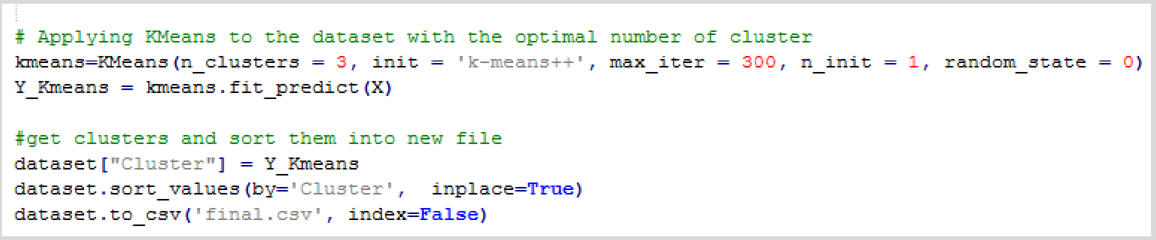
```
X = dataset.iloc[:,[3,4]].values # read columns Age-x axis and Profession-y axis

# Using the elbow method to find the optimal number of clusters
from sklearn.cluster import KMeans
wcss =[]
for i in range (1,1):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', max_iter = 200, n_init = 1, random_state = 0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_) #Within Cluster Sum of Squares
```

Elbow Method

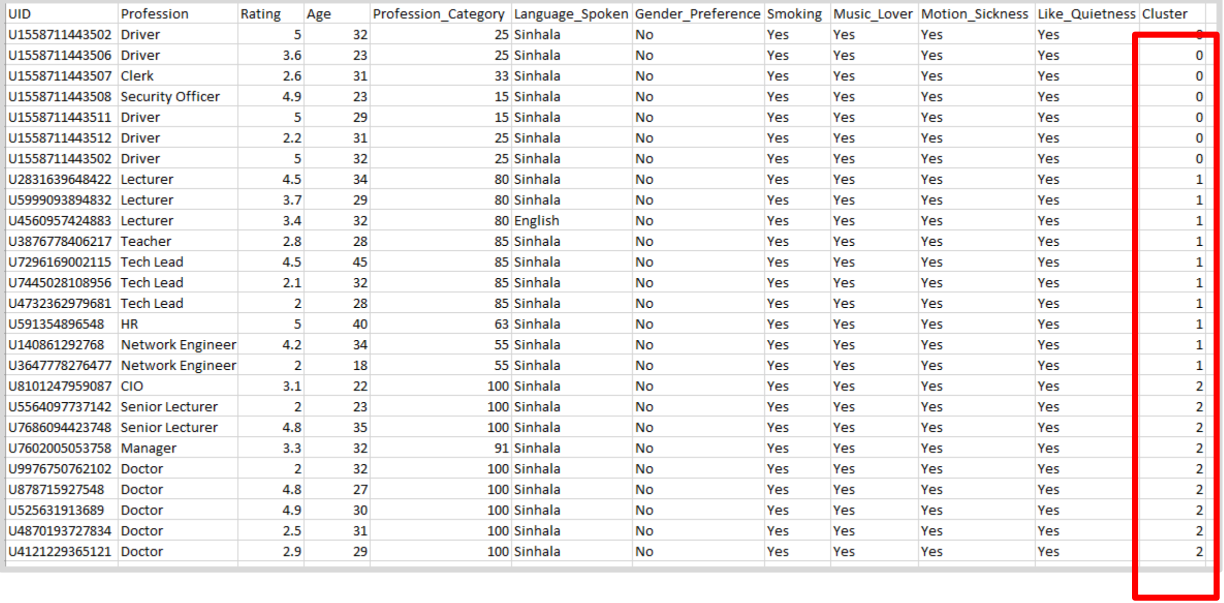


The Specialized Area and the Use of Technologies





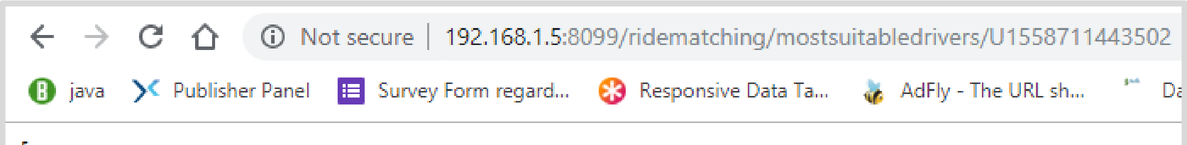
```
# Applying KMeans to the dataset with the optimal number of cluster
kmeans=KMeans(n_clusters = 3, init = 'k-means++', max_iter = 300, n_init = 1, random_state = 0)
Y_Kmeans = kmeans.fit_predict(X)

#get clusters and sort them into new file
dataset["Cluster"] = Y_Kmeans
dataset.sort_values(by='Cluster',  inplace=True)
dataset.to_csv('final.csv', index=False)
```



UID	Profession	Rating	Age	Profession_Category	Language_Spoken	Gender_Preference	Smoking	Music_Lover	Motion_Sickness	Like_Quietness	Cluster
U1558711443502	Driver	5	32	25	Sinhala	No	Yes	Yes	Yes	Yes	0
U1558711443506	Driver	3.6	23	25	Sinhala	No	Yes	Yes	Yes	Yes	0
U1558711443507	Clerk	2.6	31	33	Sinhala	No	Yes	Yes	Yes	Yes	0
U1558711443508	Security Officer	4.9	23	15	Sinhala	No	Yes	Yes	Yes	Yes	0
U1558711443511	Driver	5	29	15	Sinhala	No	Yes	Yes	Yes	Yes	0
U1558711443512	Driver	2.2	31	25	Sinhala	No	Yes	Yes	Yes	Yes	0
U1558711443502	Driver	5	32	25	Sinhala	No	Yes	Yes	Yes	Yes	0
U2831639648422	Lecturer	4.5	34	80	Sinhala	No	Yes	Yes	Yes	Yes	1
U5999093894832	Lecturer	3.7	29	80	Sinhala	No	Yes	Yes	Yes	Yes	1
U4560957424883	Lecturer	3.4	32	80	English	No	Yes	Yes	Yes	Yes	1
U3876778406217	Teacher	2.8	28	85	Sinhala	No	Yes	Yes	Yes	Yes	1
U7296169002115	Tech Lead	4.5	45	85	Sinhala	No	Yes	Yes	Yes	Yes	1
U7445028108956	Tech Lead	2.1	32	85	Sinhala	No	Yes	Yes	Yes	Yes	1
U4732362979681	Tech Lead	2	28	85	Sinhala	No	Yes	Yes	Yes	Yes	1
U591354896548	HR	5	40	63	Sinhala	No	Yes	Yes	Yes	Yes	1
U140861292768	Network Engineer	4.2	34	55	Sinhala	No	Yes	Yes	Yes	Yes	1
U3647778276477	Network Engineer	2	18	55	Sinhala	No	Yes	Yes	Yes	Yes	1
U8101247959087	CIO	3.1	22	100	Sinhala	No	Yes	Yes	Yes	Yes	2
U5564097737142	Senior Lecturer	2	23	100	Sinhala	No	Yes	Yes	Yes	Yes	2
U7686094423748	Senior Lecturer	4.8	35	100	Sinhala	No	Yes	Yes	Yes	Yes	2
U7602005053758	Manager	3.3	32	91	Sinhala	No	Yes	Yes	Yes	Yes	2
U9976750762102	Doctor	2	32	100	Sinhala	No	Yes	Yes	Yes	Yes	2
U878715927548	Doctor	4.8	27	100	Sinhala	No	Yes	Yes	Yes	Yes	2
U525631913689	Doctor	4.9	30	100	Sinhala	No	Yes	Yes	Yes	Yes	2
U4870193727834	Doctor	2.5	31	100	Sinhala	No	Yes	Yes	Yes	Yes	2
U4121229365121	Doctor	2.9	29	100	Sinhala	No	Yes	Yes	Yes	Yes	2

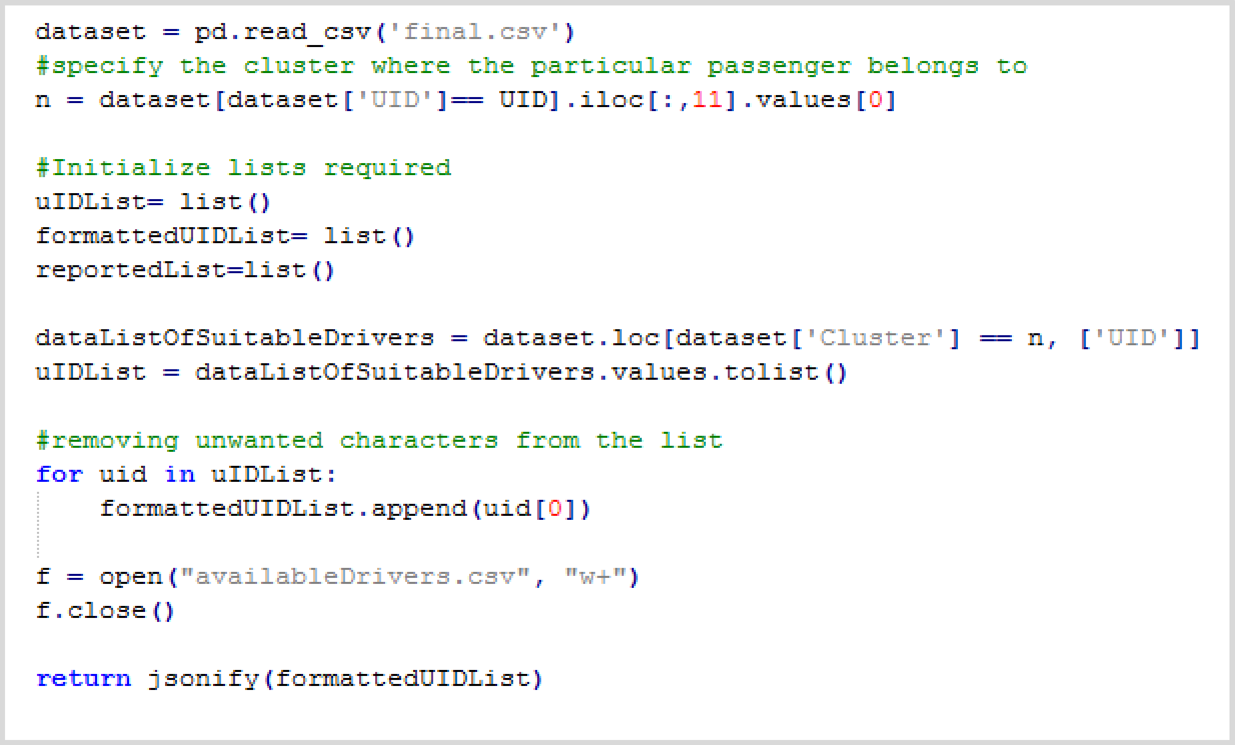
K-Means Clustering

The Specialized Area and the Use of Technologies



192.168.1.5:8099/ridematching/mostsuitabledrivers/U1558711443502

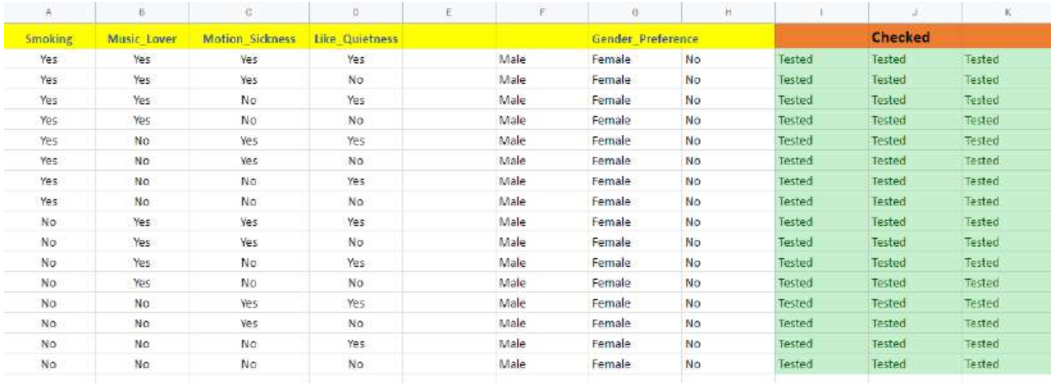
```
[
  "U1558711443502",
  "U1558711443508",
  "U1560496133942",
  "U1558711443507",
  "U5888630437833",
  "U6932118549572",
  "U1558711443514",
  "U1558711443513",
  "U1558711443512",
  "U1558711443506",
  "U1558711443510",
  "U1558711443511"
]
```



```
dataset = pd.read_csv('final.csv')
#specify the cluster where the particular passenger belongs to
n = dataset[dataset['UID']== UID].iloc[:,11].values[0]

#Initialize lists required
uIDList= list()
formattedUIDList= list()
reportedList=list()

dataListOfSuitableDrivers = dataset.loc[dataset['Cluster'] == n, ['UID']]
uIDList = dataListOfSuitableDrivers.values.tolist()

#removing unwanted characters from the list
for uid in uIDList:
    formattedUIDList.append(uid[0])

f = open("availableDrivers.csv", "w+")
f.close()

return jsonify(formattedUIDList)
```

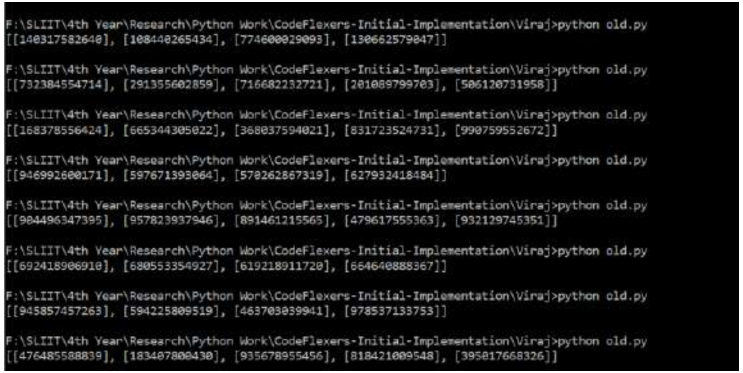
Most Suitable Driver List

Testing and Results



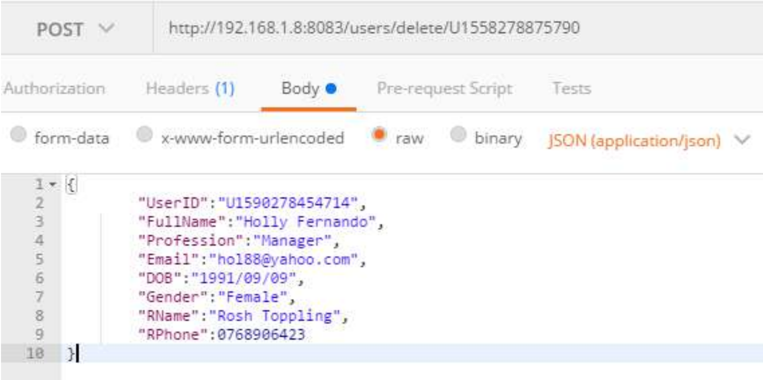
A	B	C	D	E	F	G	H	I	J	K
Smoking	Music_Lover	Motion_Sickness	Like_Quietness			Gender_Preference		Checked		
Yes	Yes	Yes	Yes		Male	Female	No	Tested	Tested	Tested
Yes	Yes	Yes	No		Male	Female	No	Tested	Tested	Tested
Yes	Yes	No	Yes		Male	Female	No	Tested	Tested	Tested
Yes	Yes	No	No		Male	Female	No	Tested	Tested	Tested
Yes	No	Yes	Yes		Male	Female	No	Tested	Tested	Tested
Yes	No	Yes	No		Male	Female	No	Tested	Tested	Tested
Yes	No	No	Yes		Male	Female	No	Tested	Tested	Tested
Yes	No	No	No		Male	Female	No	Tested	Tested	Tested
No	Yes	Yes	Yes		Male	Female	No	Tested	Tested	Tested
No	Yes	Yes	No		Male	Female	No	Tested	Tested	Tested
No	Yes	No	Yes		Male	Female	No	Tested	Tested	Tested
No	Yes	No	No		Male	Female	No	Tested	Tested	Tested
No	No	Yes	Yes		Male	Female	No	Tested	Tested	Tested
No	No	Yes	No		Male	Female	No	Tested	Tested	Tested
No	No	No	Yes		Male	Female	No	Tested	Tested	Tested
No	No	No	No		Male	Female	No	Tested	Tested	Tested

Backend Test Plan



```
F:\SLIIT\4th Year\Research\Python Work\CodeFlexers-Initial-Implementation\Viraj>python old.py
[[140317582640], [108440265434], [774600029093], [130662579047]]

F:\SLIIT\4th Year\Research\Python Work\CodeFlexers-Initial-Implementation\Viraj>python old.py
[[732384554714], [291355602859], [716682232721], [201089799703], [506120731958]]

F:\SLIIT\4th Year\Research\Python Work\CodeFlexers-Initial-Implementation\Viraj>python old.py
[[168378556424], [665344305022], [368037594021], [831723524731], [990759552672]]

F:\SLIIT\4th Year\Research\Python Work\CodeFlexers-Initial-Implementation\Viraj>python old.py
[[946992600171], [597671393064], [570262867319], [627932418484]]

F:\SLIIT\4th Year\Research\Python Work\CodeFlexers-Initial-Implementation\Viraj>python old.py
[[904496347395], [957823937946], [891461215565], [479617555363], [932129745351]]

F:\SLIIT\4th Year\Research\Python Work\CodeFlexers-Initial-Implementation\Viraj>python old.py
[[692418906910], [680553354927], [619218911720], [664640888367]]

F:\SLIIT\4th Year\Research\Python Work\CodeFlexers-Initial-Implementation\Viraj>python old.py
[[945857457263], [594225809519], [463703039941], [978537133753]]

F:\SLIIT\4th Year\Research\Python Work\CodeFlexers-Initial-Implementation\Viraj>python old.py
[[476485588839], [183407800430], [935678955456], [818421009548], [395017668326]]
```

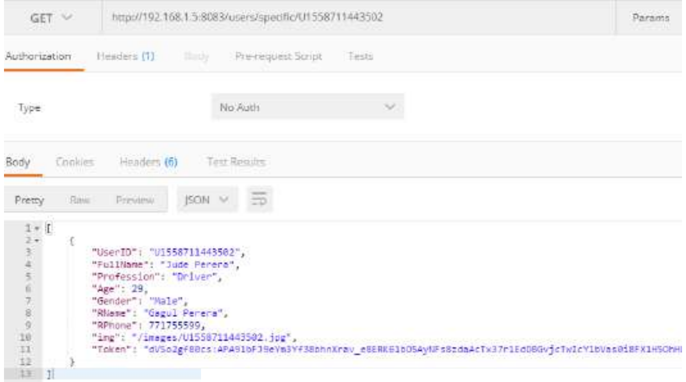
Backend Test Results



POST http://192.168.1.8:8083/users/delete/U1558278875790

```
{
        "UserID":"U1590278454714",
        "FullName":"Holly Fernando",
        "Profession":"Manager",
        "Email":"hol88@yahoo.com",
        "DOB":"1991/09/09",
        "Gender":"Female",
        "RName":"Rosh Toppling",
        "RPhone":0768906423
}
```

GET Request Testing



GET http://192.168.1.5:8083/users/specific/U1558711443502

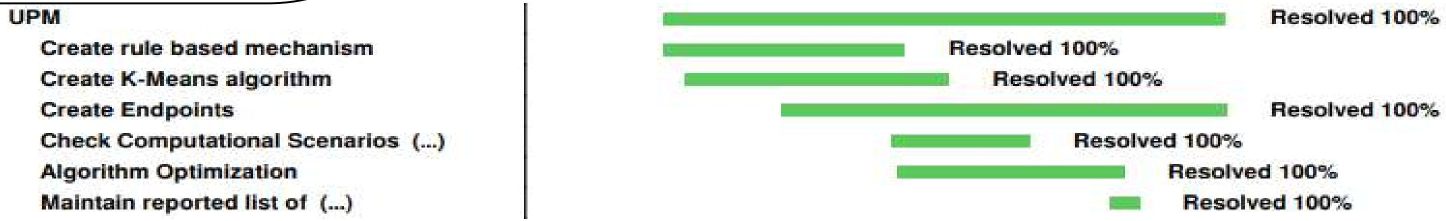
Type: No Auth

```
[
    {
        "UserID": "U1558711443502",
        "FullName": "Jude Perera",
        "Profession": "Driver",
        "Age": 29,
        "Gender": "Male",
        "RName": "Gagul Perera",
        "RPhone": 771755599,
        "img": "/images/U1558711443502.jpg",
        "Token": "dVSo2gf80cs:APA91bFJ9eYm3Yf38hnXrav_e8ERK61bO5AyNFs8zdaAcTx37r1Ed0BGvjcTwIcY1bVas0i8FX1HSOhH
    }
]
```

POST Request Testing

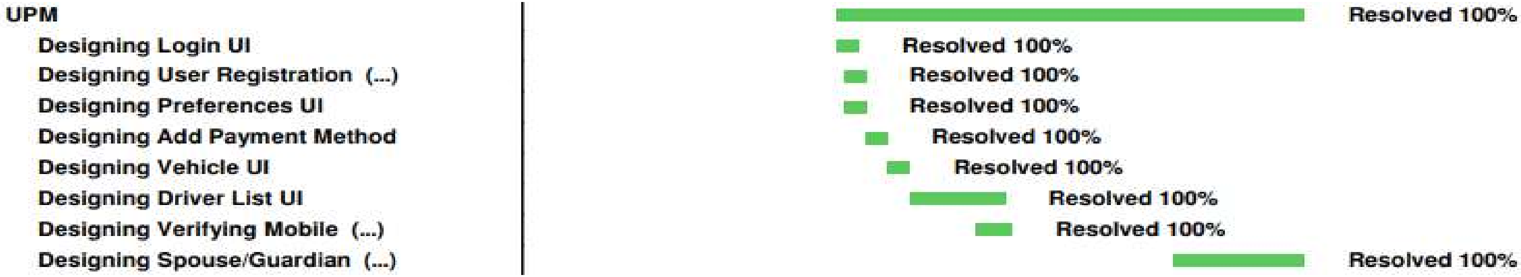
Work Progress

Back End Development



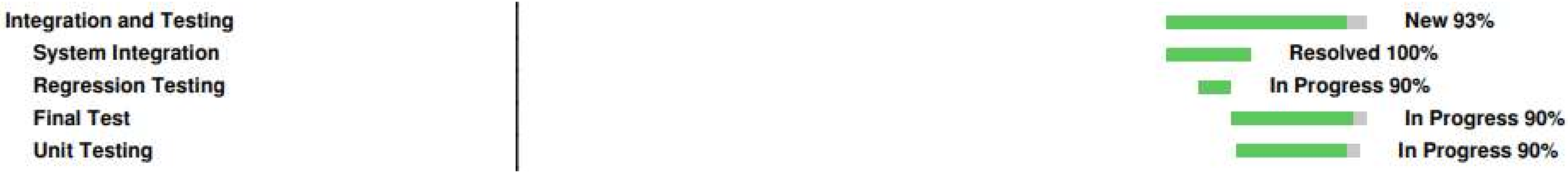
UPM	Resolved 100%
Create rule based mechanism	Resolved 100%
Create K-Means algorithm	Resolved 100%
Create Endpoints	Resolved 100%
Check Computational Scenarios (...)	Resolved 100%
Algorithm Optimization	Resolved 100%
Maintain reported list of (...)	Resolved 100%

Front End Development



UPM	Resolved 100%
Designing Login UI	Resolved 100%
Designing User Registration (...)	Resolved 100%
Designing Preferences UI	Resolved 100%
Designing Add Payment Method	Resolved 100%
Designing Vehicle UI	Resolved 100%
Designing Driver List UI	Resolved 100%
Designing Verifying Mobile (...)	Resolved 100%
Designing Spouse/Guardian (...)	Resolved 100%

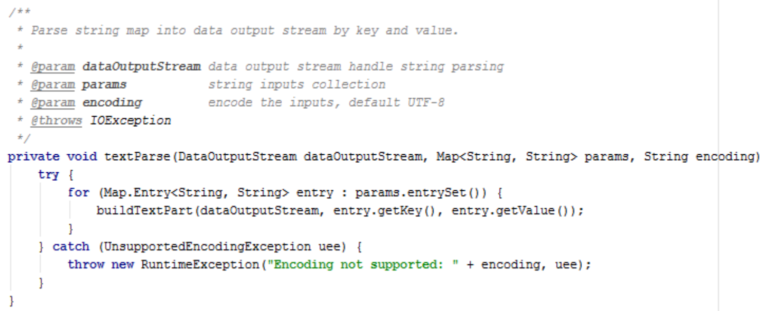
Testing



Integration and Testing	New 93%
System Integration	Resolved 100%
Regression Testing	In Progress 90%
Final Test	In Progress 90%
Unit Testing	In Progress 90%

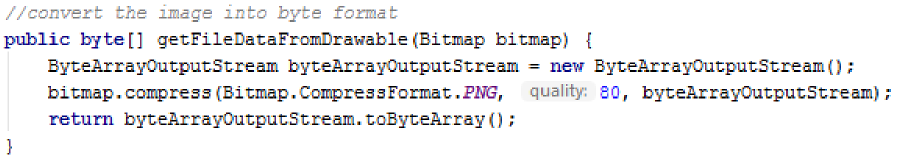
Standards and Best Practices

Clean code and use of Comments



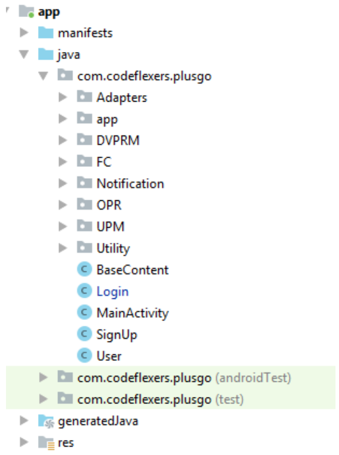
```
/**
 * Parse string map into data output stream by key and value.
 *
 * @param dataOutputStream data output stream handle string parsing
 * @param params           string inputs collection
 * @param encoding         encode the inputs, default UTF-8
 * @throws IOException
 */
private void textParse(DataOutputStream dataOutputStream, Map<String, String> params, String encoding)
    try {
        for (Map.Entry<String, String> entry : params.entrySet()) {
            buildTextPart(dataOutputStream, entry.getKey(), entry.getValue());
        }
    } catch (UnsupportedEncodingException uee) {
        throw new RuntimeException("Encoding not supported: " + encoding, uee);
    }
}
```

Standard Naming Conventions



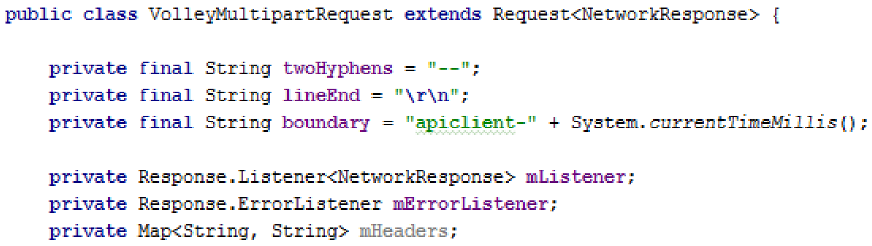
```
//convert the image into byte format
public byte[] getFileDataFromDrawable(Bitmap bitmap) {
    ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream();
    bitmap.compress(Bitmap.CompressFormat.PNG,  quality: 80, byteArrayOutputStream);
    return byteArrayOutputStream.toByteArray();
}
```

File and Folder Organization



```
app
  manifests
  java
    com.codeflexers.plusgo
      Adapters
      app
      DVPRM
      FC
      Notification
      OPR
      UPM
      Utility
      BaseContent
      Login
      MainActivity
      SignUp
      User
    com.codeflexers.plusgo (androidTest)
    com.codeflexers.plusgo (test)
  generatedJava
  res
```

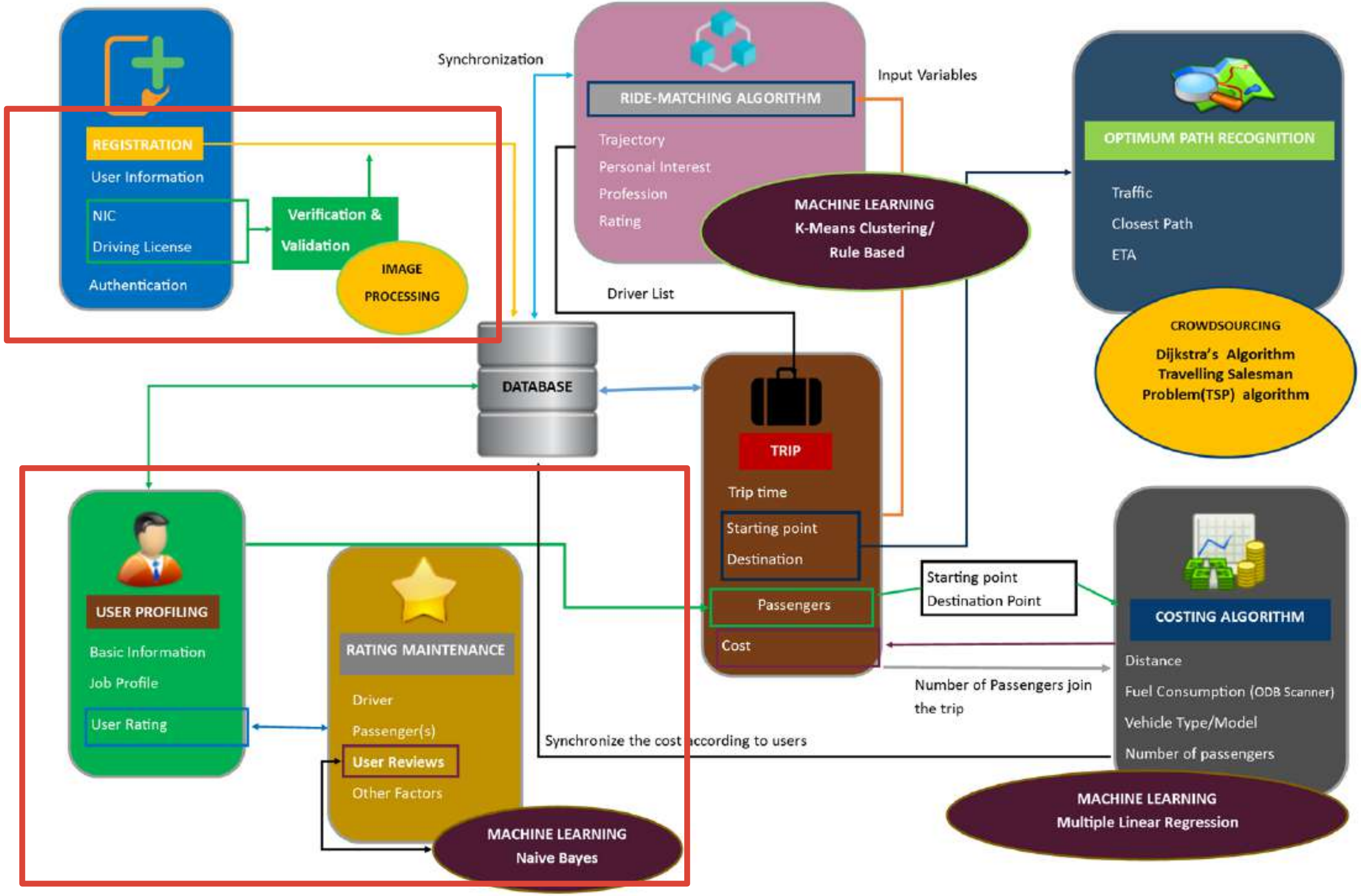
Use of Object Oriented Concepts



```
public class VolleyMultipartRequest extends Request<NetworkResponse> {

    private final String twoHyphens = "--";
    private final String lineEnd = "\r\n";
    private final String boundary = "apiclient-" + System.currentTimeMillis();

    private Response.Listener<NetworkResponse> mListener;
    private Response.ErrorListener mErrorListener;
    private Map<String, String> mHeaders;
```

Document Validation and Profile Rating Maintenance

High Level Diagram



Objective

Document Validation

Validate the driving license and NIC cards, and identify the NIC number and expiration dates using an image processing algorithm and minimize the risk of fake profiles getting registered in the system.

- Non-Electronic NIC
- Electronic NIC
- License

Profile Rating Maintenance

Identify the response of the drivers and passengers regarding their travelling experience, and rate the users and the platform accordingly.

- Keyword Identification
- Sentiment Analysis
- Unwanted Driver Blocking

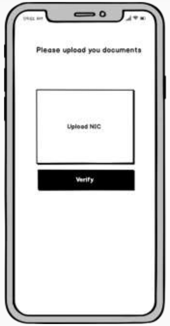
The ultimate goal is to provide a better experience by increasing the reliability and security of the proposed ride sharing platform

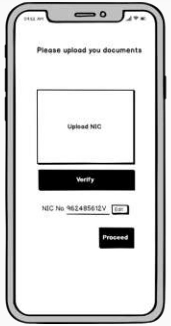
Research Gap

Features	UBER	UDIO	Carpooling.lk	RideShare.lk	+GO
Validating the user by processing and comparing the images of both NIC and license in real time	✗	✗	✗	✗	✓
Analyze the reviews given by users based on their severity and categorizing them	✗	✗	✗	✗	✓
Allowing the passengers to rate the driver, vehicle and co-passengers separately at the end of trip.	✗	✗	✗	✗	✓
Allowing the passengers to block particular driver in future suggestions	✗	✗	✗	✗	✓

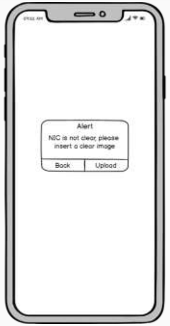
Prototype vs Implementation

NIC card validation

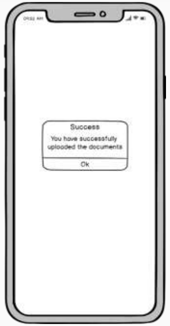

Please upload you documents


Please upload you documents

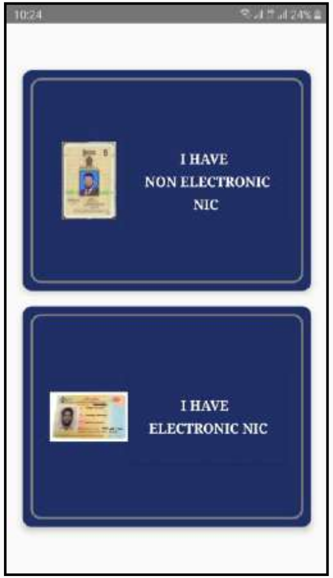
NIC No 962485612V

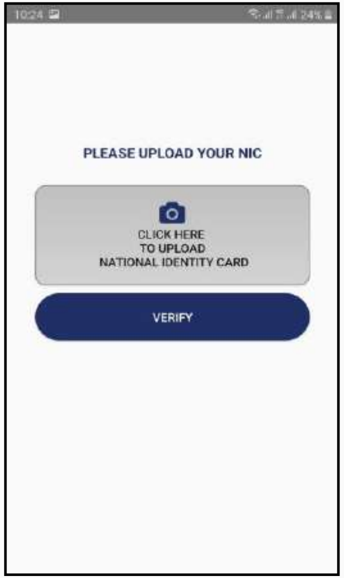

Alert

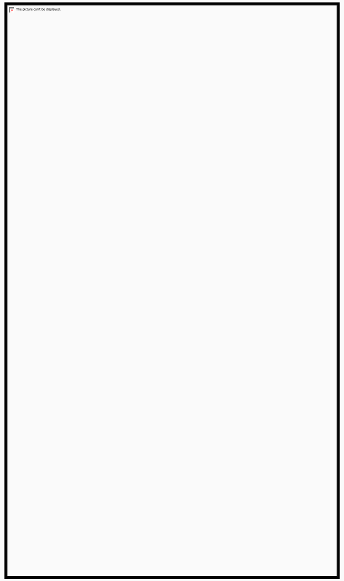
NIC is not clear, please insert a clear image

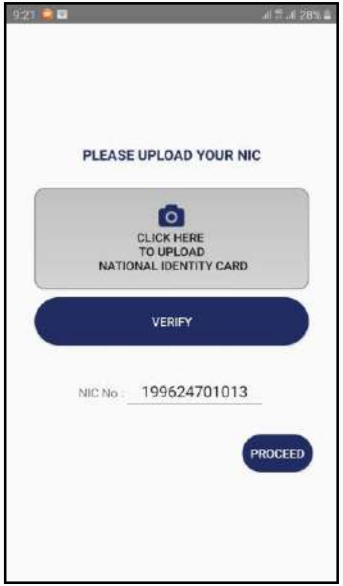

Success

You have successfully uploaded the documents





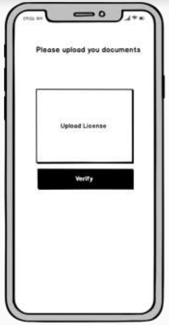

The picture can't be displayed.

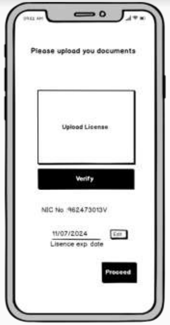


NIC No : 199624701013

Prototype vs Implementation cont...

License card validation

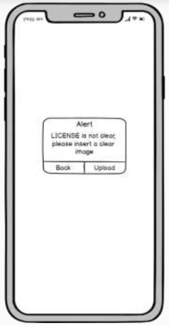

Please upload you documents


Please upload you documents

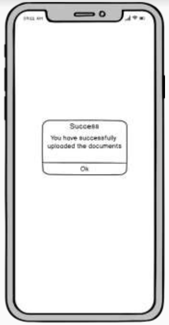
NIC No :962473013V

11/07/2024

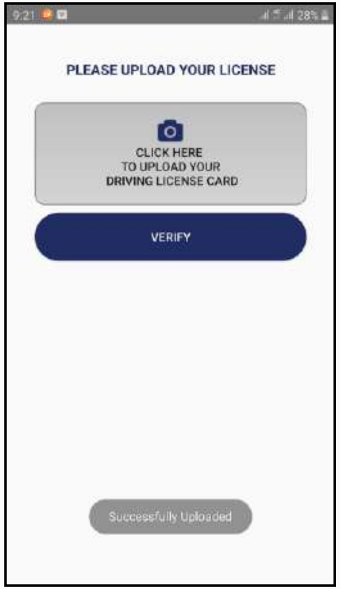
Lisence exp date


Alert

LICENSE is not clear, please insert a clear image

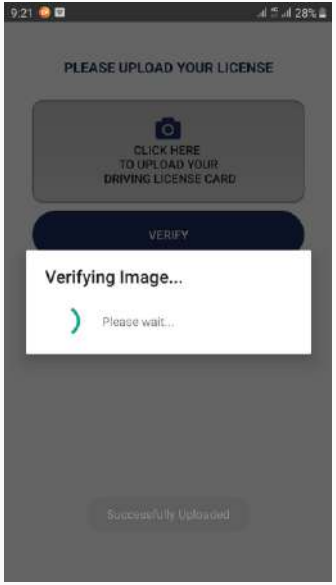

Success

You have successfully uploaded the documents


PLEASE UPLOAD YOUR LICENSE

Successfully Uploaded

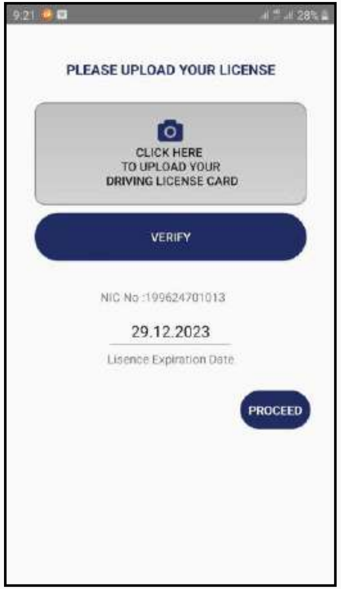
The picture can't be displayed.


PLEASE UPLOAD YOUR LICENSE

Verifying Image...

Please wait...

Successfully Uploaded

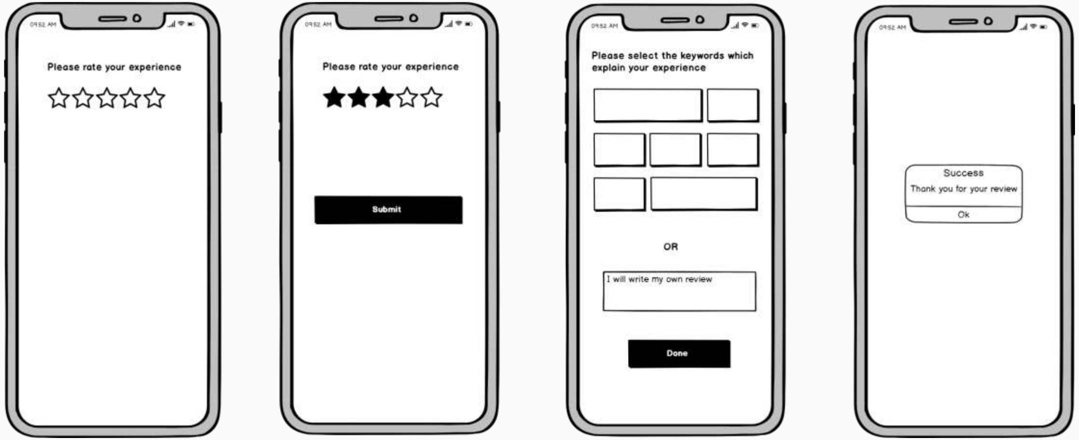

PLEASE UPLOAD YOUR LICENSE

NIC No :199624701013

29.12.2023

Lisence Expiration Date

Prototype vs Implementation cont...



Please rate your experience

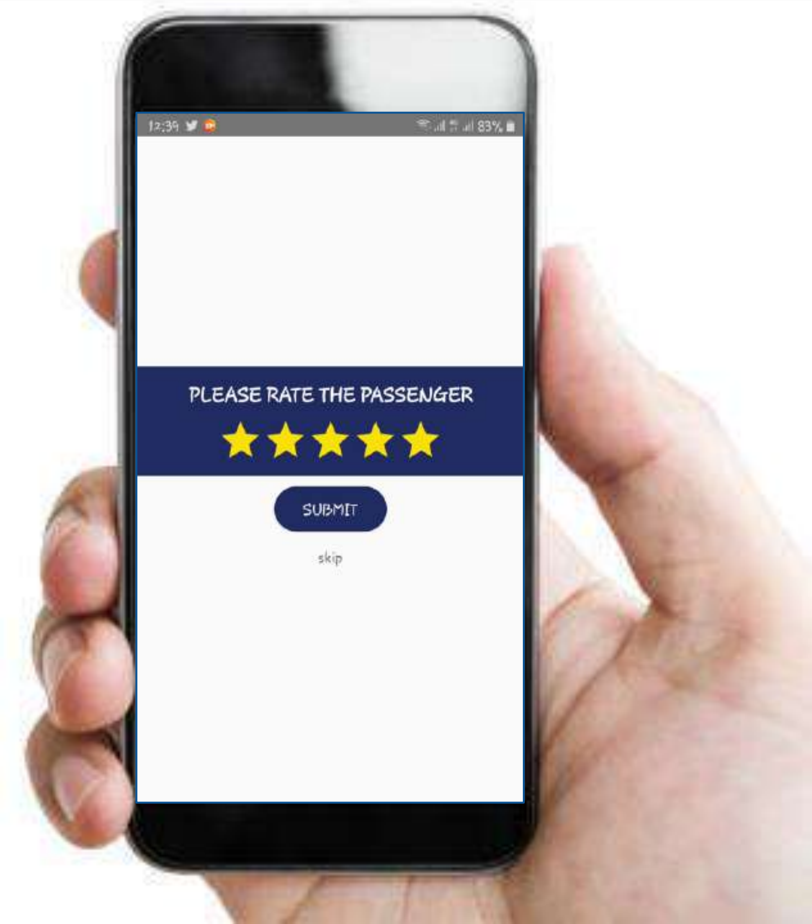
Please rate your experience

Please select the keywords which explain your experience

I will write my own review

Success

Thank you for your review

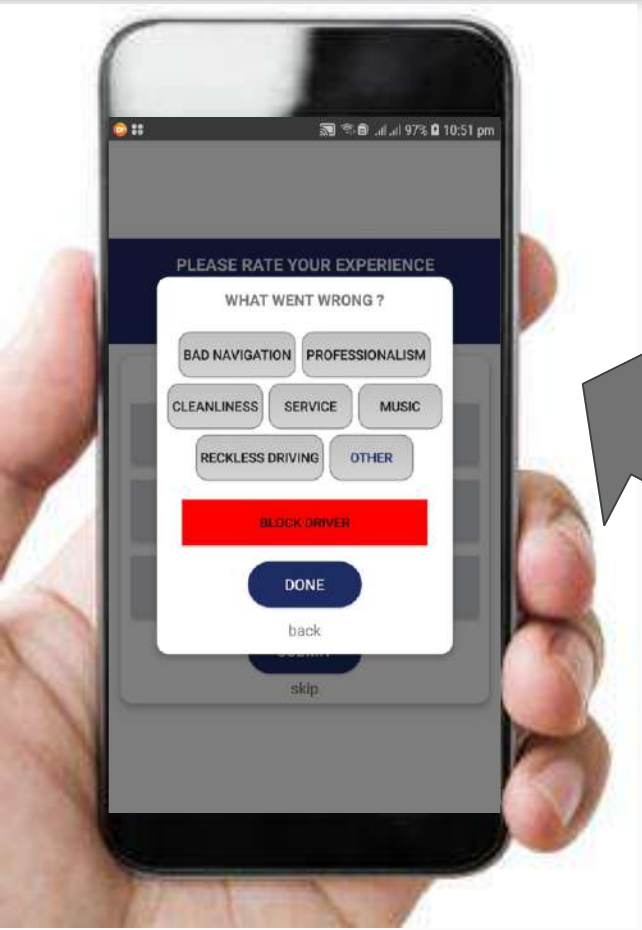


PLEASE RATE THE PASSENGER

Rating by Driver

Prototype vs Implementation cont...

Rating by Passenger



PLEASE RATE YOUR EXPERIENCE

WHAT WENT WRONG ?

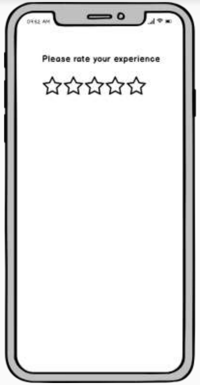
BAD NAVIGATION | PROFESSIONALISM | CLEANLINESS | SERVICE | MUSIC | RECKLESS DRIVING | OTHER

BLOCK DRIVER

DONE

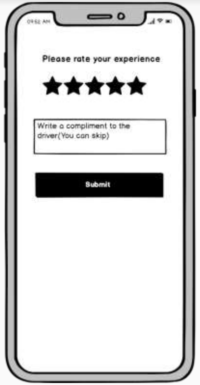
back

skip



Please rate your experience

☆☆☆☆☆

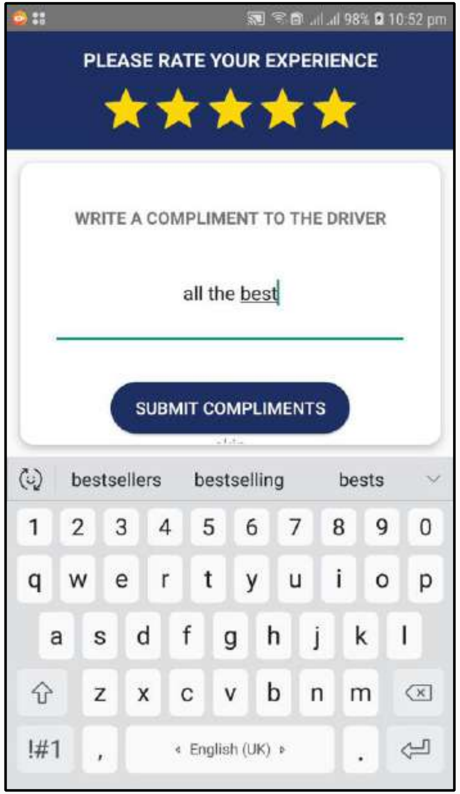


Please rate your experience

★★★★★

Write a compliment to the driver(You can skip)

Submit



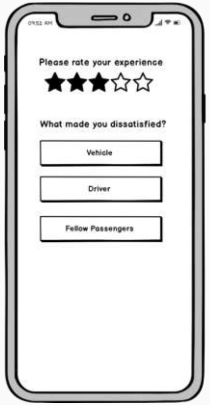
PLEASE RATE YOUR EXPERIENCE

★★★★★

WRITE A COMPLIMENT TO THE DRIVER

all the best

SUBMIT COMPLIMENTS



Please rate your experience

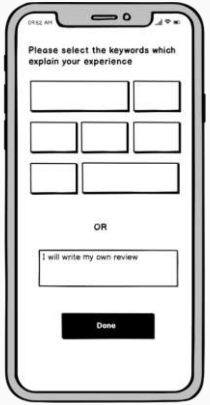
★★★☆☆

What made you dissatisfied?

Vehicle

Driver

Fellow Passengers

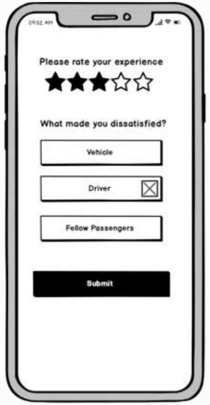


Please select the keywords which explain your experience

OR

I will write my own review

Done



Please rate your experience

★★★☆☆

What made you dissatisfied?

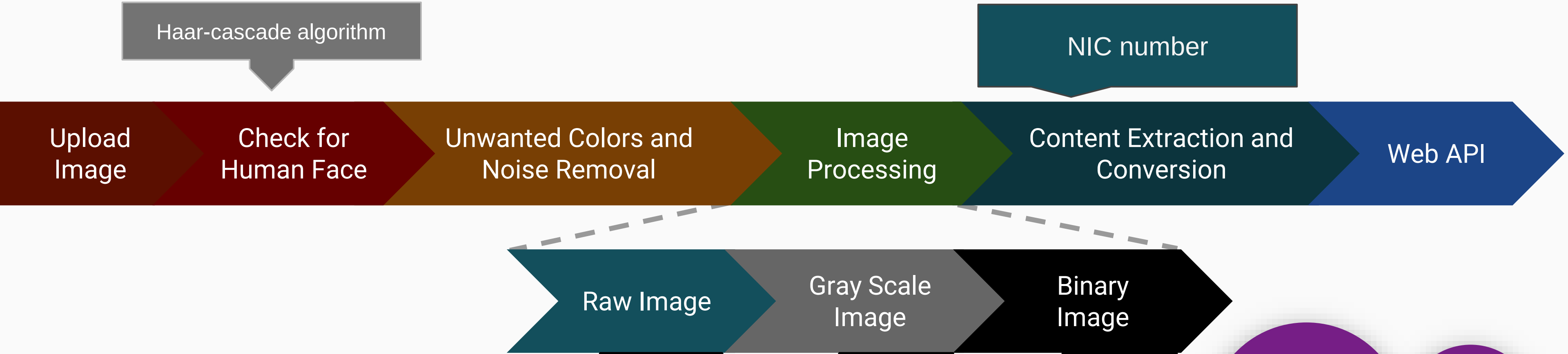
Vehicle

Driver ☒

Fellow Passengers

Submit

The Specialized Area and the Use of Technologies



Haar-cascade algorithm

NIC number

Upload Image

Check for Human Face

Unwanted Colors and Noise Removal

Image Processing

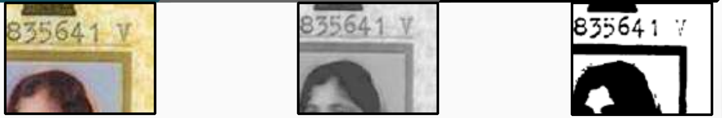
Content Extraction and Conversion

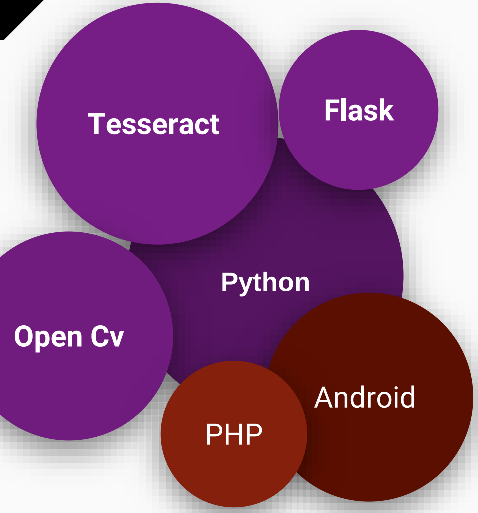
Web API

Raw Image

Gray Scale Image

Binary Image





Tesseract

Flask

Python

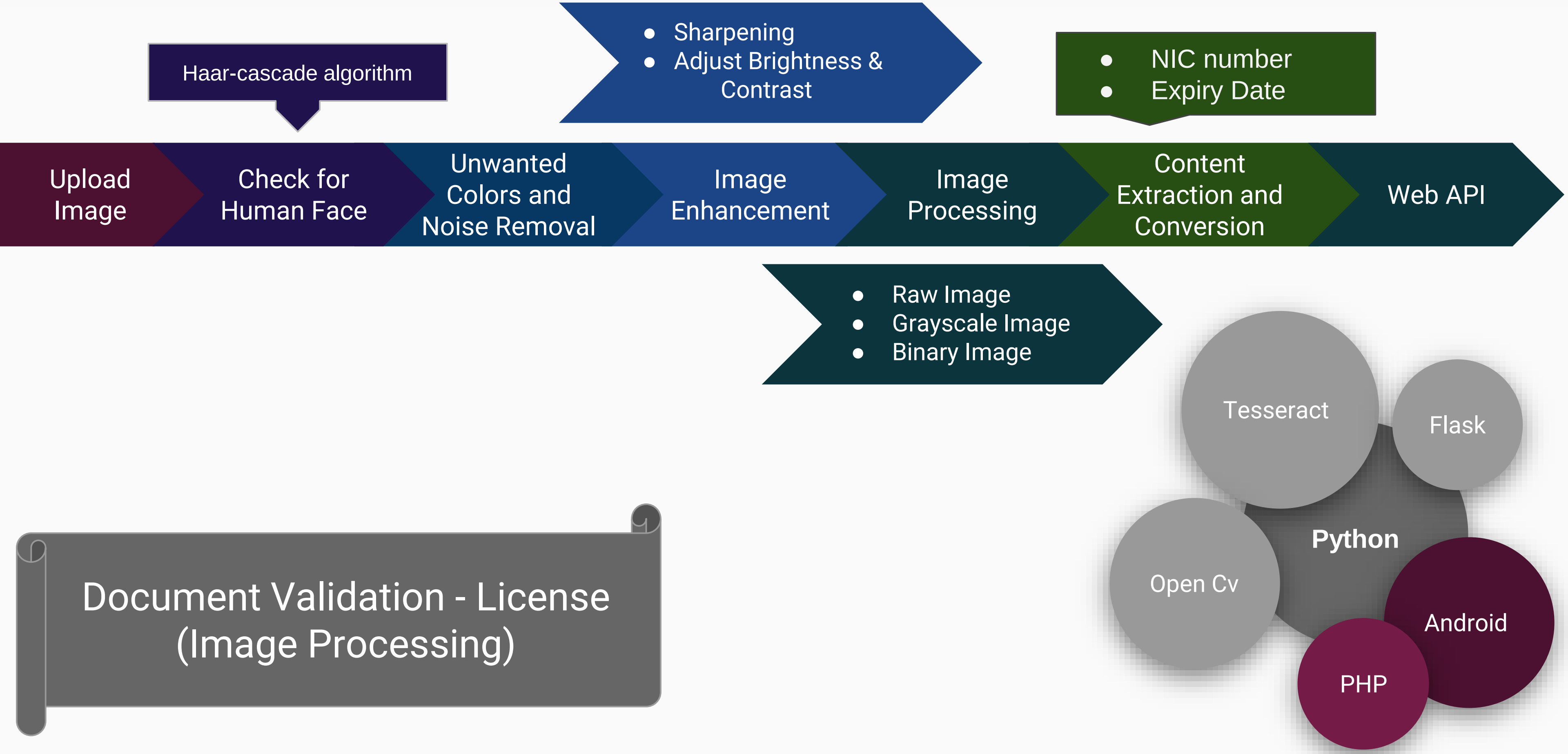
Open Cv

Android

PHP

Document Validation - NIC (Image Processing)

The Specialized Area and the Use of Technologies cont...



Haar-cascade algorithm

- Sharpening
- Adjust Brightness & Contrast

- NIC number
- Expiry Date

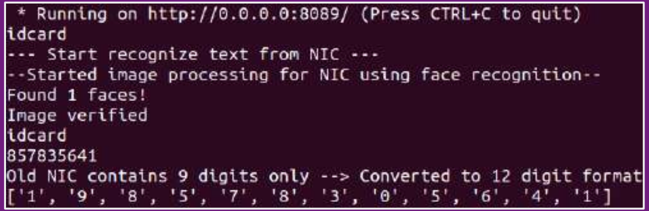
Upload Image → Check for Human Face → Unwanted Colors and Noise Removal → Image Enhancement → Image Processing → Content Extraction and Conversion → Web API

- Raw Image
- Grayscale Image
- Binary Image

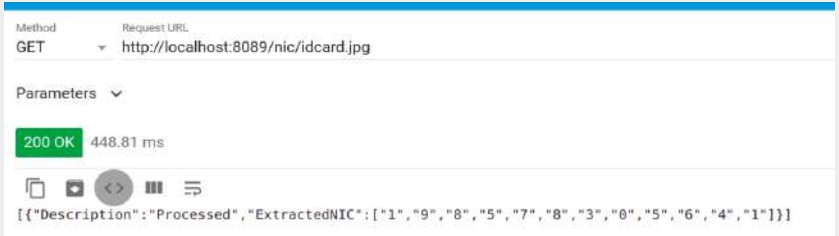
Document Validation - License (Image Processing)

Results

Non Electronic NIC



```
 * Running on http://0.0.0.0:8089/ (Press CTRL+C to quit)
idcard
--- Start recognize text from NIC ---
--Started image processing for NIC using face recognition--
Found 1 faces!
Image verified
idcard
857835641
Old NIC contains 9 digits only --> Converted to 12 digit format
['1', '9', '8', '5', '7', '8', '3', '0', '5', '6', '4', '1']
```

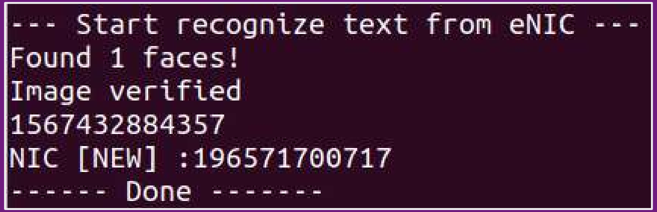


Method: GET

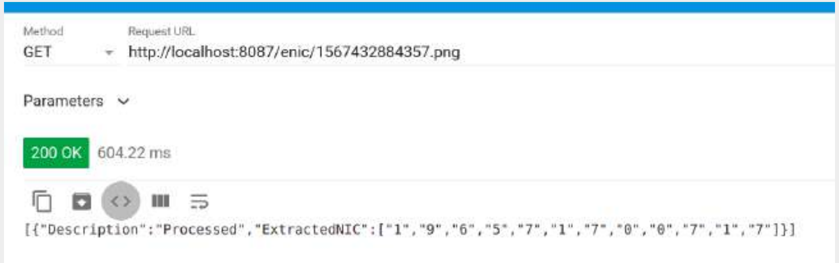
Request URL: http://localhost:8089/nic/idcard.jpg

```
[{"Description":"Processed","ExtractedNIC":["1","9","8","5","7","8","3","0","5","6","4","1"]}]
```

Electronic NIC



```
--- Start recognize text from eNIC ---
Found 1 faces!
Image verified
1567432884357
NIC [NEW] :196571700717
------ Done -------
```

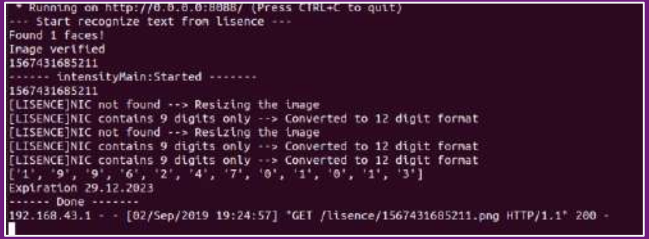


Method: GET

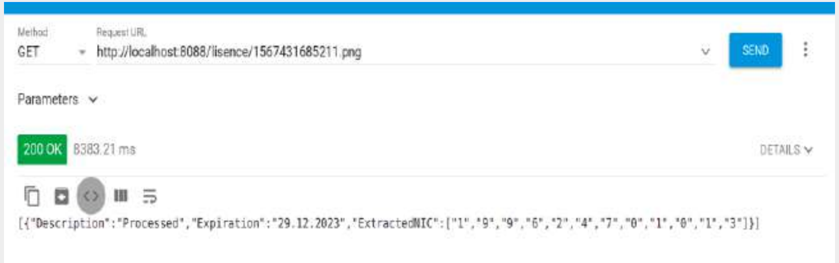
Request URL: http://localhost:8087/enic/1567432884357.png

```
[{"Description":"Processed","ExtractedNIC":["1","9","6","5","7","1","7","0","0","7","1","7"]}]
```

License



```
 * Running on http://0.0.0.0:8088/ (Press CTRL+C to quit)
--- Start recognize text from lisence ---
Found 1 faces!
Image verified
1567431685211
------ intensityMain:Started -------
1567431685211
[LISENCE]NIC not found --> Resizing the image
[LISENCE]NIC contains 9 digits only --> Converted to 12 digit format
[LISENCE]NIC not found --> Resizing the image
[LISENCE]NIC contains 9 digits only --> Converted to 12 digit format
[LISENCE]NIC contains 9 digits only --> Converted to 12 digit format
['1', '9', '9', '6', '2', '4', '7', '0', '1', '0', '1', '3']
Expiration 29.12.2023
------ Done -------
192.168.43.1 - - [02/Sep/2019 19:24:57] "GET /lisence/1567431685211.png HTTP/1.1" 200 -
```



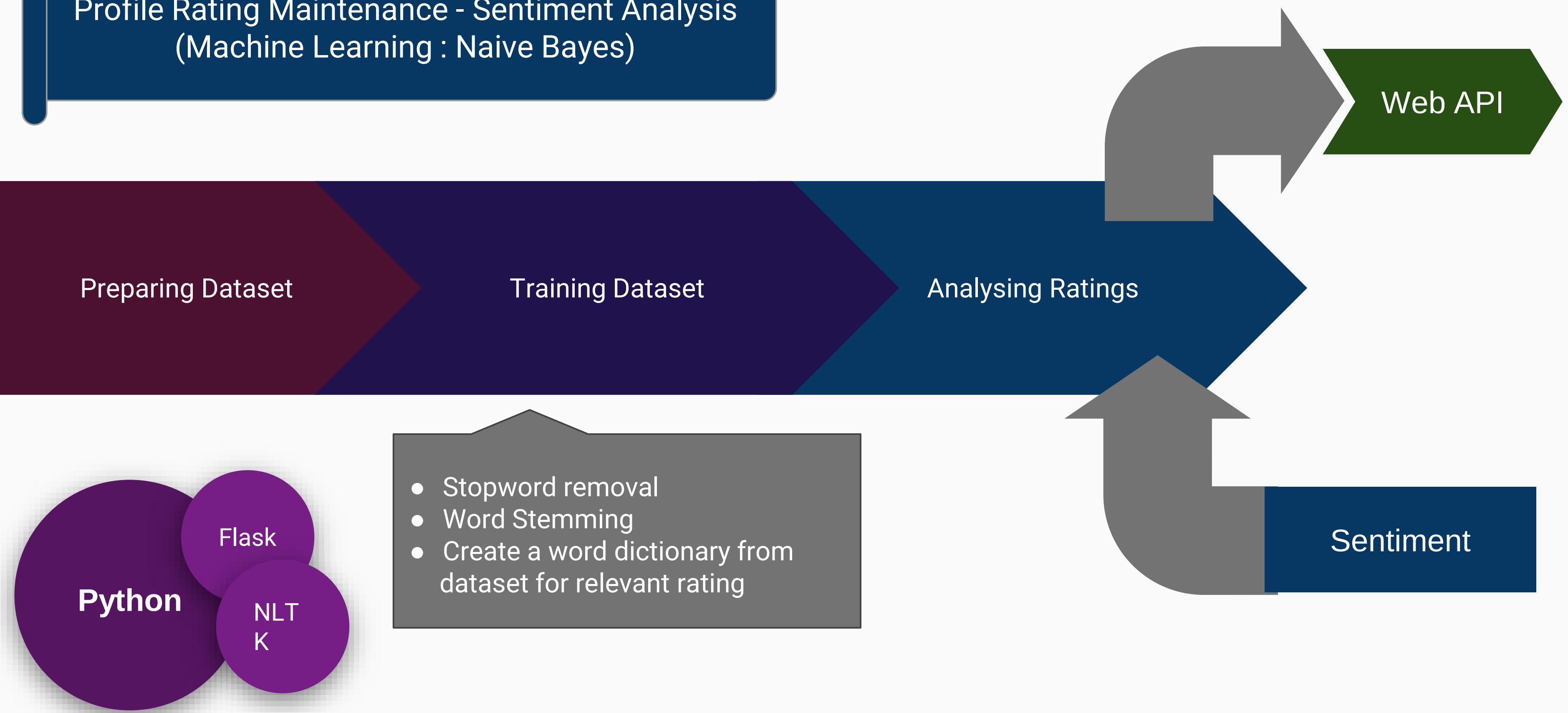
Method: GET

Request URL: http://localhost:8088/lisence/1567431685211.png

```
[{"Description":"Processed","Expiration":"29.12.2023","ExtractedNIC":["1","9","9","6","2","4","7","0","1","0","1","3"]}]
```

The Specialized Area and the Use of Technologies

Profile Rating Maintenance - Sentiment Analysis
(Machine Learning : Naive Bayes)



Preparing Dataset

Training Dataset

Analysing Ratings

Web API

Sentiment

- Stopword removal
- Word Stemming
- Create a word dictionary from dataset for relevant rating

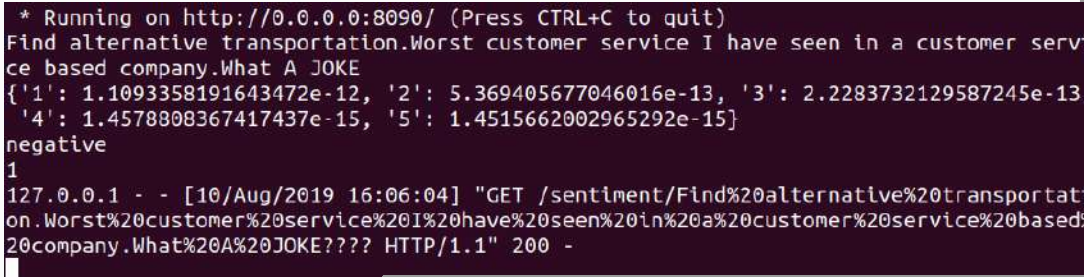
Python

Flask

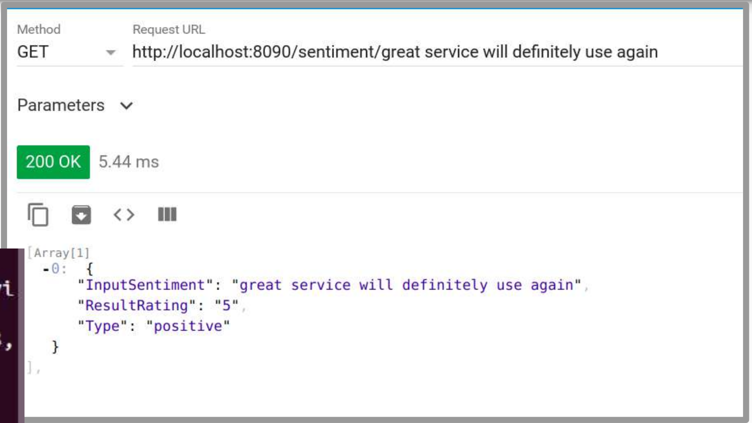
NLT
K

Results

Sentiment Analysis
(Machine Learning : Naive Bayes)



```
 * Running on http://0.0.0.0:8090/ (Press CTRL+C to quit)
Find alternative transportation.Worst customer service I have seen in a customer servi
ce based company.What A JOKE
{'1': 1.1093358191643472e-12, '2': 5.369405677046016e-13, '3': 2.2283732129587245e-13,
 '4': 1.4578808367417437e-15, '5': 1.4515662002965292e-15}
negative
1
127.0.0.1 - - [10/Aug/2019 16:06:04] "GET /sentiment/Find%20alternative%20transportati
on.Worst%20customer%20service%20I%20have%20seen%20in%20a%20customer%20service%20based%
20company.What%20A%20JOKE???? HTTP/1.1" 200 -
```

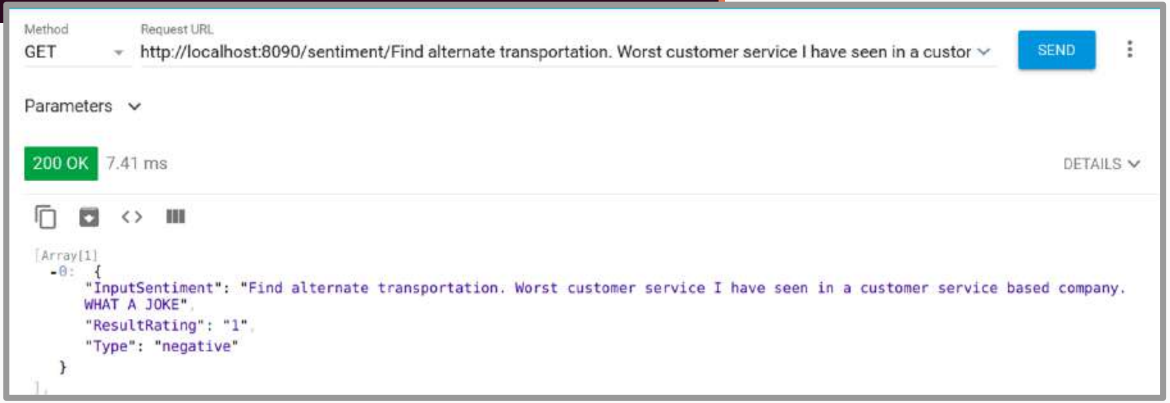


Method: GET

Request URL: http://localhost:8090/sentiment/great service will definitely use again

200 OK 5.44 ms

```
[Array[1]
 -0: {
      "InputSentiment": "great service will definitely use again",
      "ResultRating": "5",
      "Type": "positive"
    }
],
```



Method: GET

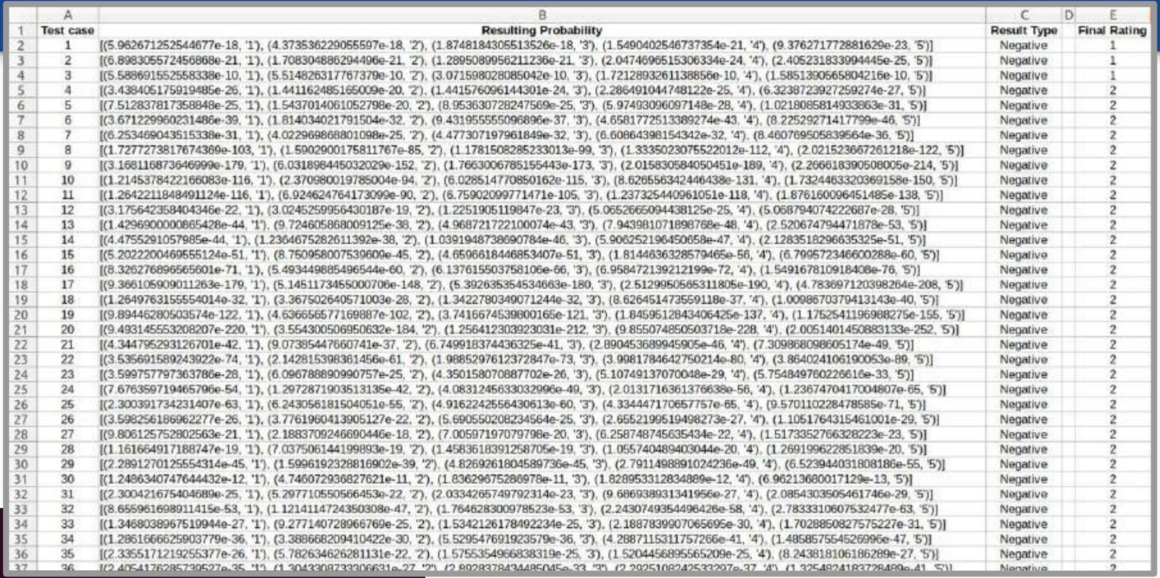
Request URL: http://localhost:8090/sentiment/Find alternate transportation. Worst customer service I have seen in a custor

200 OK 7.41 ms

```
[Array[1]
 -0: {
      "InputSentiment": "Find alternate transportation. Worst customer service I have seen in a customer service based company.
      WHAT A JOKE",
      "ResultRating": "1",
      "Type": "negative"
    }
],
```

Testing and Results

Sentiment Analysis
(Machine Learning : Naive Bayes)



Test case	Resulting Probability	Result Type	Final Rating
1	[(5.962671252544677e-18, '1'), (4.373536229055597e-18, '2'), (1.8748184305513526e-18, '3'), (1.5490402546737354e-21, '4'), (9.376271772881629e-23, '5')]	Negative	1
2	[(6.898305572456868e-21, '1'), (1.708304886294496e-21, '2'), (1.2895089956211236e-21, '3'), (2.0474696515306334e-24, '4'), (2.405231833994445e-25, '5')]	Negative	1
3	[(5.588691552558338e-10, '1'), (5.514826317767379e-10, '2'), (3.071598028085042e-10, '3'), (1.7212893261138856e-10, '4'), (1.5851390565804216e-10, '5')]	Negative	1
4	[(3.438405175919485e-26, '1'), (1.441162485165009e-20, '2'), (1.441576096144301e-24, '3'), (2.286491044748122e-25, '4'), (6.323872392725274e-27, '5')]	Negative	2
5	[(7.512837817358848e-25, '1'), (1.5437014061052798e-20, '2'), (8.953630728247569e-25, '3'), (5.97493096097148e-28, '4'), (1.0218085814933863e-31, '5')]	Negative	2
6	[(3.671229960231486e-39, '1'), (1.814034021791504e-32, '2'), (9.431955555096896e-37, '3'), (4.6581772513389274e-43, '4'), (8.22529271417799e-46, '5')]	Negative	2
7	[(6.253469043515338e-31, '1'), (4.022969868801098e-25, '2'), (4.477307197961849e-32, '3'), (6.60864398154342e-32, '4'), (8.460769505839564e-36, '5')]	Negative	2
8	[(1.7277273817674369e-103, '1'), (1.5902900175811767e-85, '2'), (1.1781508285233013e-99, '3'), (1.3335023075522012e-112, '4'), (2.021523667261218e-122, '5')]	Negative	2
9	[(3.1681168736469996e-179, '1'), (6.031898445032029e-152, '2'), (1.7663006785155443e-173, '3'), (2.015830584050451e-189, '4'), (2.266618390508005e-214, '5')]	Negative	2
10	[(1.2145378422166083e-116, '1'), (2.370980019785004e-94, '2'), (6.028514770850162e-115, '3'), (8.626556342446438e-131, '4'), (1.7324463320369158e-150, '5')]	Negative	2
11	[(1.264221184849 1124e-116, '1'), (6.924624764173099e-90, '2'), (6.75902099771471e-105, '3'), (1.237325440961051e-118, '4'), (1.876160096451485e-138, '5')]	Negative	2
12	[(3.175642358404346e-22, '1'), (3.0245259956430187e-19, '2'), (1.2251905119847e-23, '3'), (5.0652665094438125e-25, '4'), (5.068794074222687e-28, '5')]	Negative	2
13	[(1.4296900000865428e-44, '1'), (9.724605868009125e-38, '2'), (4.968721722100074e-43, '3'), (7.943981071898768e-48, '4'), (2.520674794471878e-53, '5')]	Negative	2
14	[(4.4755291057985e-44, '1'), (1.2364675282611392e-38, '2'), (1.0391948738690784e-46, '3'), (5.906252196450658e-47, '4'), (2.1283518296635325e-51, '5')]	Negative	2
15	[(5.2022200469555124e-51, '1'), (8.750958007539609e-45, '2'), (4.6596618446853407e-51, '3'), (1.8144636328579465e-56, '4'), (6.799572346600288e-60, '5')]	Negative	2
16	[(8.326276896565601e-71, '1'), (5.493449885496544e-60, '2'), (6.137615503758106e-66, '3'), (6.958472139212199e-72, '4'), (1.549167810918408e-76, '5')]	Negative	2
17	[(9.366105909011263e-179, '1'), (5.1451173455000706e-148, '2'), (5.392635354534663e-180, '3'), (2.5129950565311805e-190, '4'), (4.783697120398264e-208, '5')]	Negative	2
18	[(1.2649763155554014e-32, '1'), (3.367502640571003e-28, '2'), (1.3422780349071244e-32, '3'), (8.626451473559118e-37, '4'), (1.0098670379413143e-40, '5')]	Negative	2
19	[(9.89446280503574e-122, '1'), (4.636656577169887e-102, '2'), (3.7416674539800165e-121, '3'), (1.8459512843406425e-137, '4'), (1.1752541196988275e-155, '5')]	Negative	2
20	[(9.493145553208207e-220, '1'), (3.554300506950632e-184, '2'), (1.256412303923031e-212, '3'), (9.855074850503718e-228, '4'), (2.005140145088 3133e-252, '5')]	Negative	2
21	[(4.344795293126701e-42, '1'), (9.07385447660741e-37, '2'), (6.749918374436325e-41, '3'), (2.890453689945905e-46, '4'), (7.309868098605174e-49, '5')]	Negative	2
22	[(3.535691589243922e-74, '1'), (2.142815398361456e-61, '2'), (1.9885297612372847e-73, '3'), (3.9981784642750214e-80, '4'), (3.864024106190053e-89, '5')]	Negative	2
23	[(3.599757797363786e-28, '1'), (6.096788890990757e-25, '2'), (4.350158070887702e-26, '3'), (5.10749137070048e-29, '4'), (5.754849760226616e-33, '5')]	Negative	2
24	[(7.676359719465796e-54, '1'), (1.2972871903513135e-42, '2'), (4.0831245633032996e-49, '3'), (2.0131716361376638e-56, '4'), (1.2367470417004807e-65, '5')]	Negative	2
25	[(2.300391734231407e-63, '1'), (6.243056181504051e-55, '2'), (4.9162242556430613e-60, '3'), (4.334447170657757e-65, '4'), (9.570110228478585e-71, '5')]	Negative	2
26	[(3.598256186962277e-26, '1'), (3.7761960413905127e-22, '2'), (5.690550208234564e-25, '3'), (2.6552199519498273e-27, '4'), (1.105176431546 1001e-29, '5')]	Negative	2
27	[(9.806125752802563e-21, '1'), (2.1883709246690446e-18, '2'), (7.00597197079798e-20, '3'), (6.258748745635434e-22, '4'), (1.5173352766328223e-23, '5')]	Negative	2
28	[(1.161664917188747e-19, '1'), (7.037506144199893e-19, '2'), (1.4583618391258705e-19, '3'), (1.055740489403044e-20, '4'), (1.269199622851839e-20, '5')]	Negative	2
29	[(2.2891270125554314e-45, '1'), (1.5996192328816902e-39, '2'), (4.8269261804589736e-45, '3'), (2.7911498891024236e-49, '4'), (6.523944031808186e-55, '5')]	Negative	2
30	[(1.2486340747644432e-12, '1'), (4.746072936827621e-11, '2'), (1.83629675286978e-11, '3'), (1.828953312834889e-12, '4'), (6.96213680017129e-13, '5')]	Negative	2
31	[(2.300421675404689e-25, '1'), (5.297710550566453e-22, '2'), (2.0334265749792314e-23, '3'), (9.686938931341956e-27, '4'), (2.0854303505461746e-29, '5')]	Negative	2
32	[(8.655961698911415e-53, '1'), (1.1214114724350308e-47, '2'), (1.764628300978523e-53, '3'), (2.2430749354496426e-58, '4'), (2.7833310607532477e-63, '5')]	Negative	2
33	[(1.3468038967519944e-27, '1'), (9.277140728966769e-25, '2'), (1.5342126178492234e-25, '3'), (2.1887839907065695e-30, '4'), (1.7028850827575227e-31, '5')]	Negative	2
34	[(1.2861666625903779e-36, '1'), (3.388668209410422e-30, '2'), (5.529547691923579e-36, '3'), (4.2887115311757266e-41, '4'), (1.485857554526996e-47, '5')]	Negative	2
35	[(2.3355171219255377e-26, '1'), (5.782634626281131e-22, '2'), (1.5755354966838319e-25, '3'), (1.5204456895565209e-25, '4'), (8.243818106186289e-27, '5')]	Negative	2
36	[(2.4054176285739527e-35, '1'), (1.3043308733306631e-27, '2'), (2.8928378434485045e-33, '3'), (2.2925108242533297e-37, '4'), (1.3254824183728489e-41, '5')]	Negative	2

```
-----Testing for the exact value-----
No. of correct matches of values   : 803
No. of incorrect matches of values : 197
                                  ------------------------
Accuracy of exact matches [%]      : 80.30000000000001

---Testing for the positive/negative sentiment identification---
No. of correct matches for sentiment classification   : 895
No. of incorrect matches for sentiment classification : 105
                                                      -----------
Accuracy of sentiment classification [%]              : 89.5

---Testing completed---
```

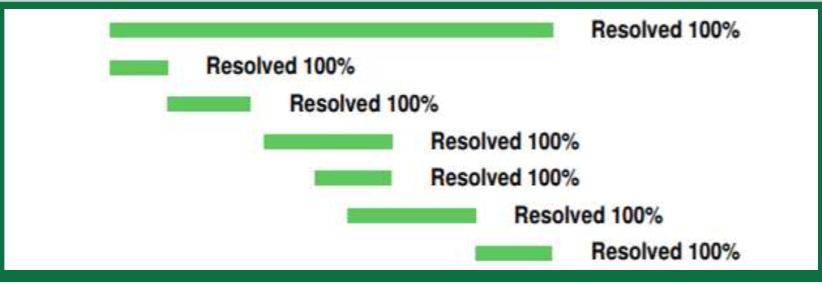
Sample test result as a report

Work Progress

100%

DVPRM	Resolved 100%
Create algorithm to extract (...)	Resolved 100%
Extract NIC number from license	Resolved 100%
Extract NIC from Electronic (...)	Resolved 100%
Create algorithm to analyse (...)	Resolved 100%
Create endpoints	Resolved 100%
Integration of services with (...)	Resolved 100%

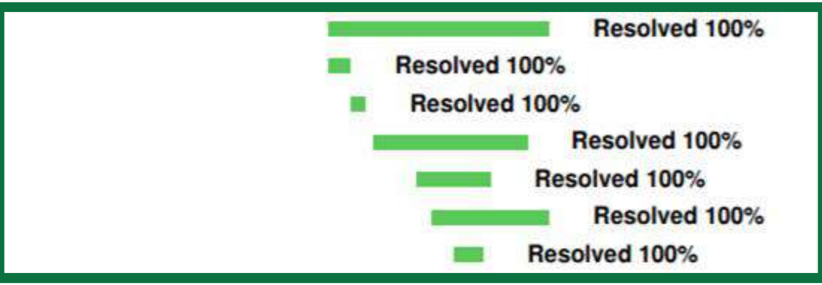
Backend



100%

DVPRM	Resolved 100%
Design Rating Interface of (...)	Resolved 100%
Designing rating activity (...)	Resolved 100%
Creation of UI with passenger (...)	Resolved 100%
Creation of UI to capture (...)	Resolved 100%
Creation of UI to capture (...)	Resolved 100%
Design Rating Interfaces for (...)	Resolved 100%

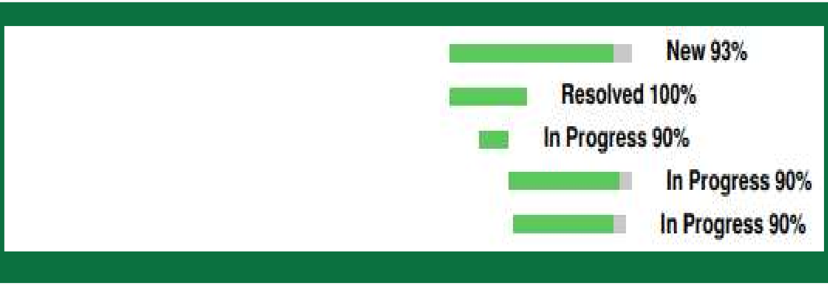
Frontend



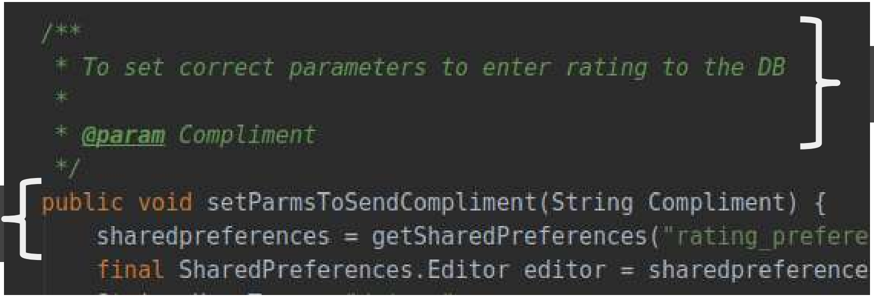
93%

Integration and Testing	New 93%
System Integration	Resolved 100%
Regression Testing	In Progress 90%
Final Test	In Progress 90%
Unit Testing	In Progress 90%

Testing



Standards and Best Practices

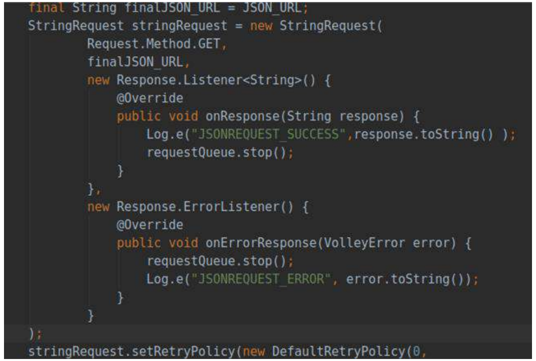


```
/**
 * To set correct parameters to enter rating to the DB
 *
 * @param Compliment
 */
public void setParmsToSendCompliment(String Compliment) {
    sharedpreferences = getSharedPreferences("rating_prefere
    final SharedPreferences.Editor editor = sharedpreference
```

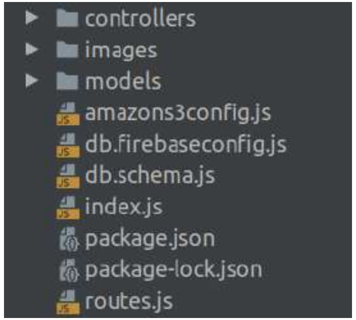
Clean code and use of Comments

Standard Naming Conventions

Consistent indentation

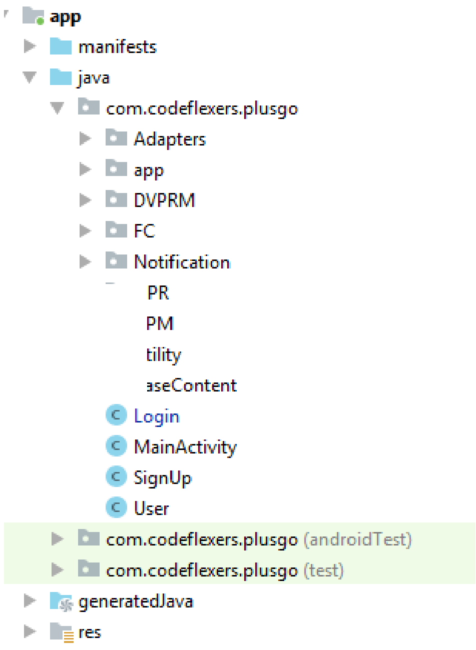


```
final String finalJSON_URL = JSON_URL;
StringRequest stringRequest = new StringRequest(
        Request.Method.GET,
        finalJSON_URL,
        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                Log.e("JSONREQUEST_SUCCESS",response.toString() );
                requestQueue.stop();
            }
        },
        new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                requestQueue.stop();
                Log.e("JSONREQUEST_ERROR", error.toString());
            }
        }
);
stringRequest.setRetryPolicy(new DefaultRetryPolicy(0,
```



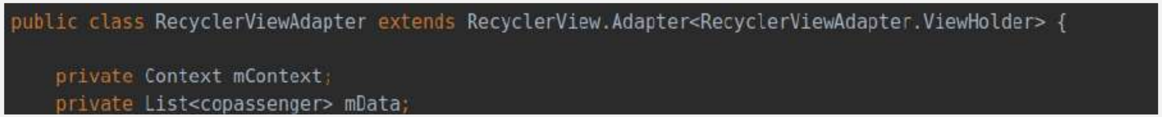
```
▶ controllers
▶ images
▶ models
  amazons3config.js
  db.firebaseconfig.js
  db.schema.js
  index.js
  package.json
  package-lock.json
  routes.js
```

File and Folder Organization



```
app
  ▶ manifests
  ▼ java
      ▼ com.codeflexers.plusgo
          ▶ Adapters
          ▶ app
          ▶ DVPRM
          ▶ FC
          ▶ Notification
              PR
              PM
              tility
              aseContent
            Login
            MainActivity
            SignUp
            User
      ▶ com.codeflexers.plusgo (androidTest)
      ▶ com.codeflexers.plusgo (test)
  ▶ generatedJava
  ▶ res
```

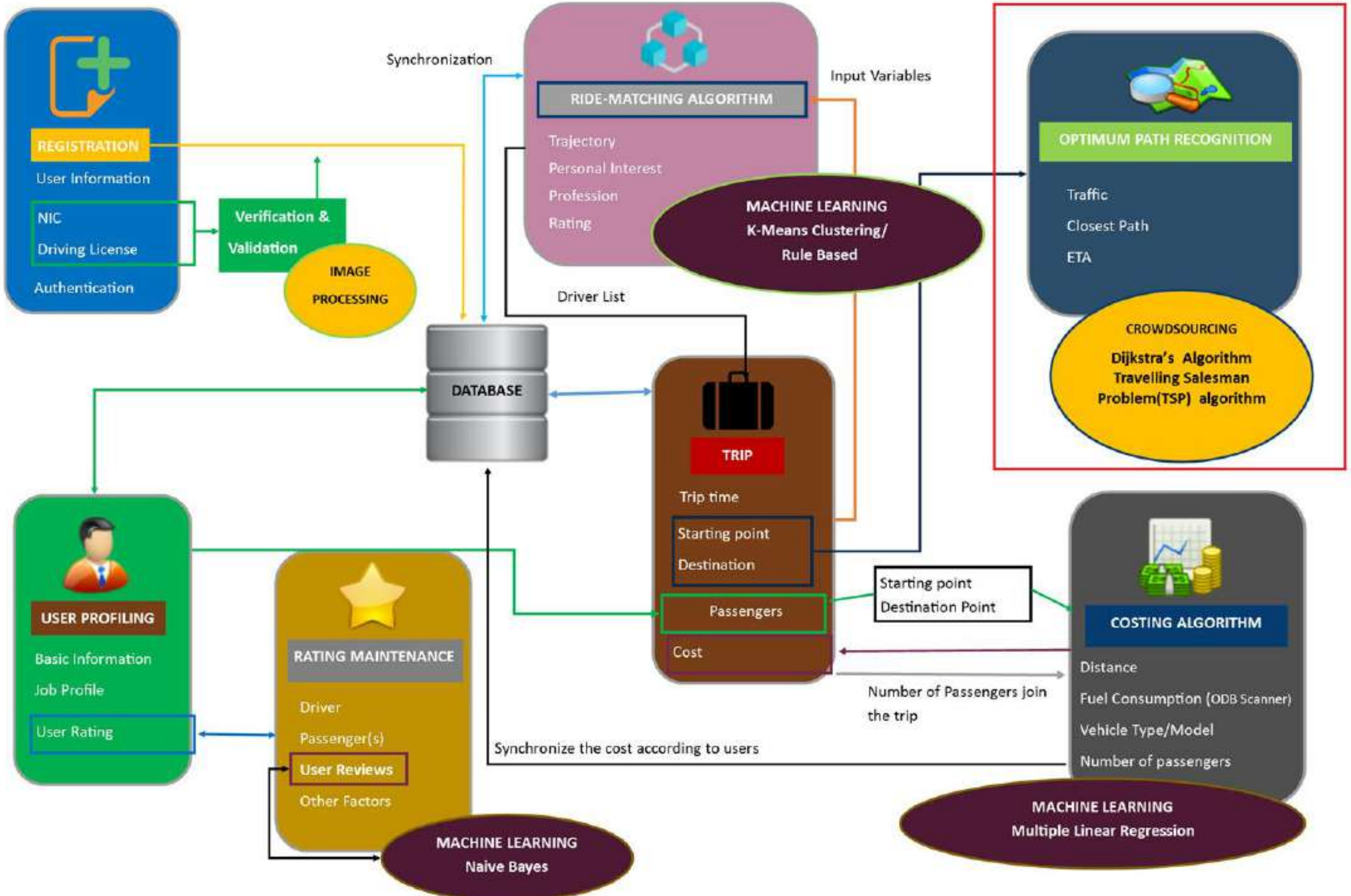
Use of Object Oriented Concepts



```
public class RecyclerViewAdapter extends RecyclerView.Adapter<RecyclerViewAdapter.ViewHolder> {

    private Context mContext;
    private List<copassenger> mData;
```

Optimum Path Recognition

High Level Diagram



REGISTRATION
User Information
NIC
Driving License
Authentication

Verification & Validation

IMAGE PROCESSING

Synchronization

RIDE-MATCHING ALGORITHM
Trajectory
Personal Interest
Profession
Rating

MACHINE LEARNING
K-Means Clustering/
Rule Based

Input Variables

OPTIMUM PATH RECOGNITION
Traffic
Closest Path
ETA

CROWDSOURCING
Dijkstra's Algorithm
Travelling Salesman Problem(TSP) algorithm

Driver List

DATABASE

TRIP
Trip time
Starting point
Destination
Passengers
Cost

Starting point
Destination Point

COSTING ALGORITHM
Distance
Fuel Consumption (ODB Scanner)
Vehicle Type/Model
Number of passengers

Number of Passengers join the trip

Synchronize the cost according to users

MACHINE LEARNING
Multiple Linear Regression

USER PROFILING
Basic Information
Job Profile
User Rating

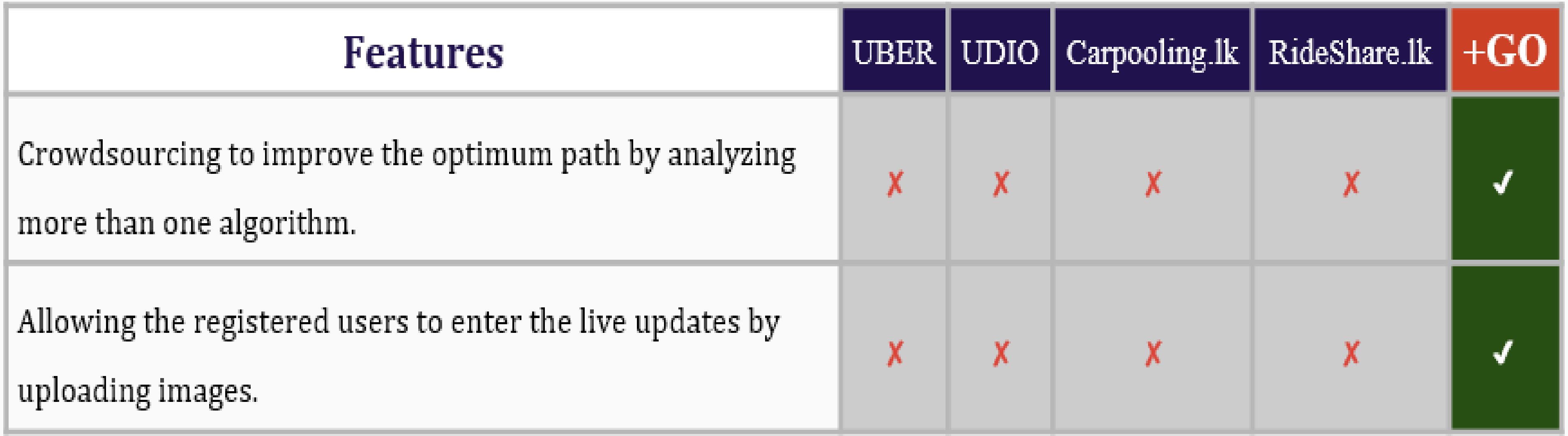
RATING MAINTENANCE
Driver
Passenger(s)
User Reviews
Other Factors

MACHINE LEARNING
Naive Bayes

Objective

- **In Order to display the route,finding the shortest path which connect starting point and destination ,while tracking the position and order of intermediary locations by using crowdsourcing technology.**

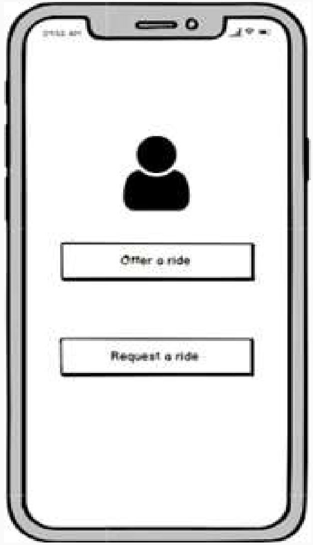
Research Gap

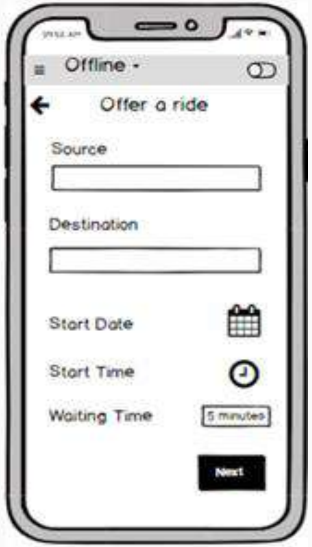


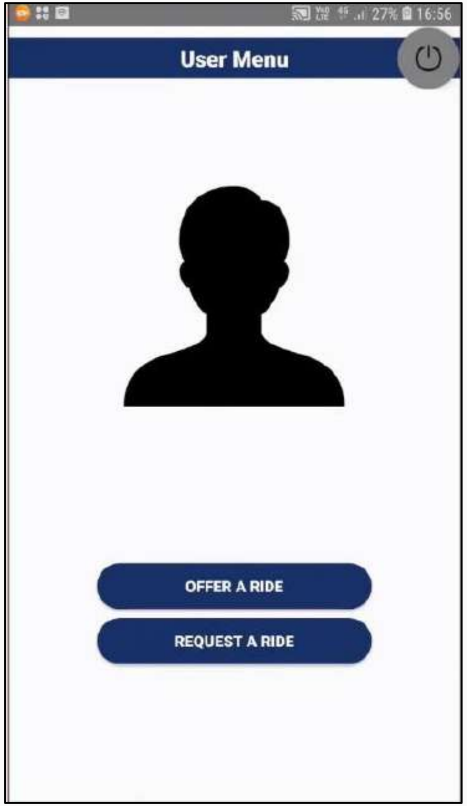
Features	UBER	UDIO	Carpooling.lk	RideShare.lk	+GO
Crowdsourcing to improve the optimum path by analyzing more than one algorithm.	✗	✗	✗	✗	✓
Allowing the registered users to enter the live updates by uploading images.	✗	✗	✗	✗	✓

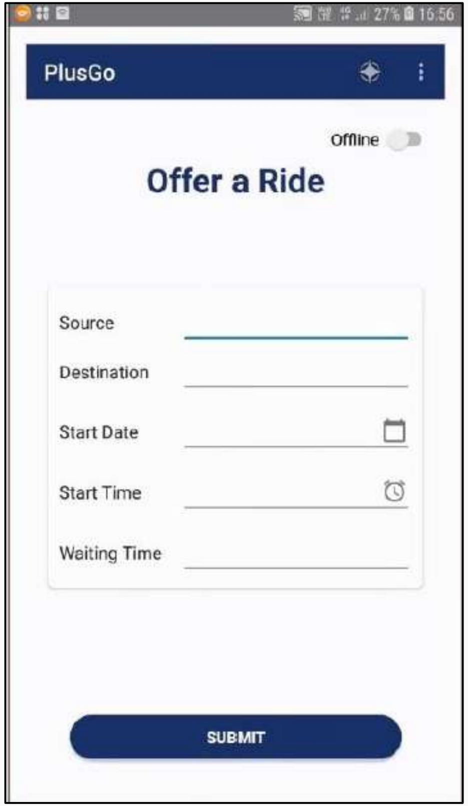
Prototype vs Implementation

DRIVER OFFER RIDE



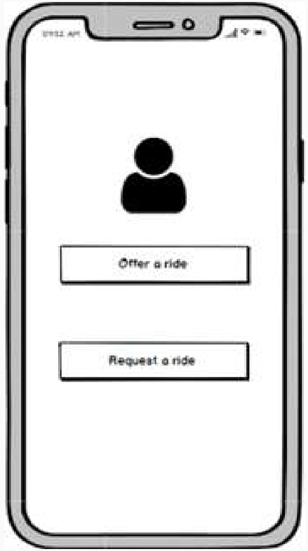


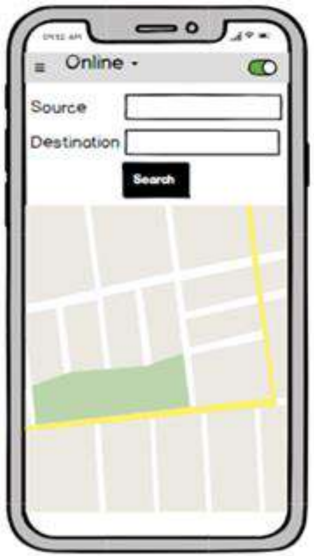


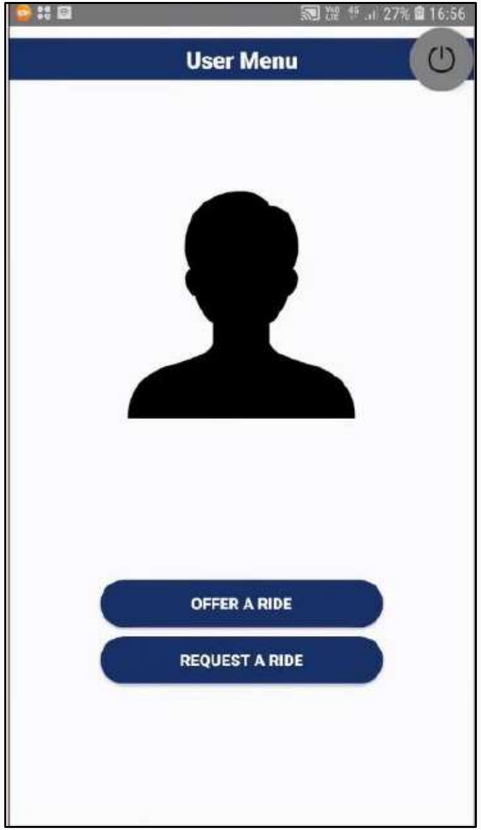


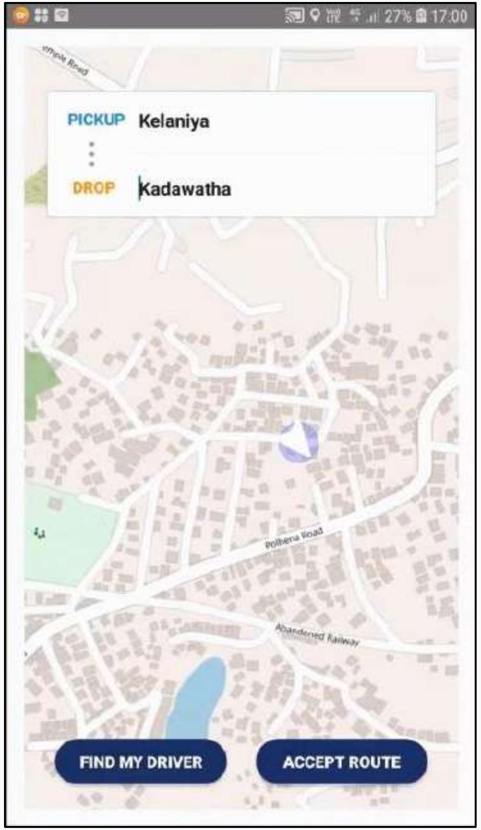
Prototype vs Implementation cont...

PASSENGER FINDS A RIDE



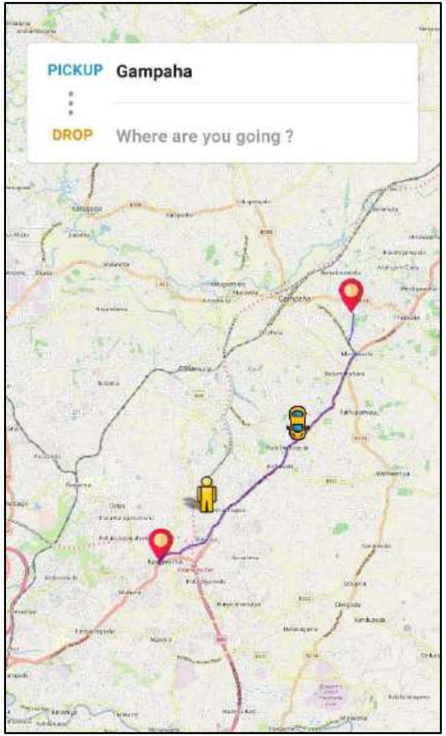






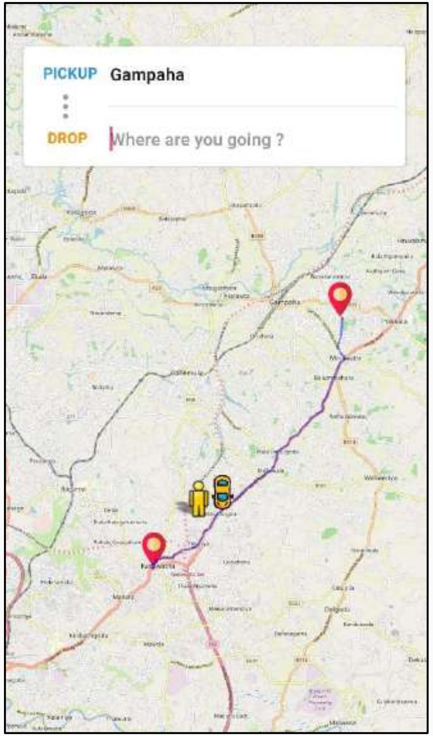
Prototype vs Implementation cont...

DRIVER NAVIGATE TO PASSENGER LOCATION



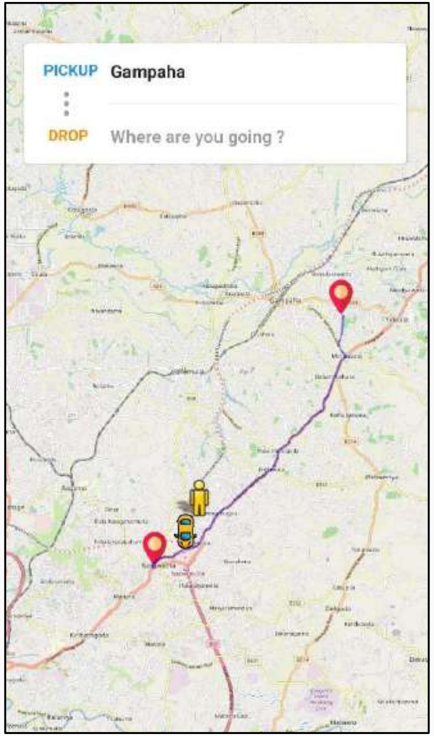
PICKUP Gampaha

DROP Where are you going ?



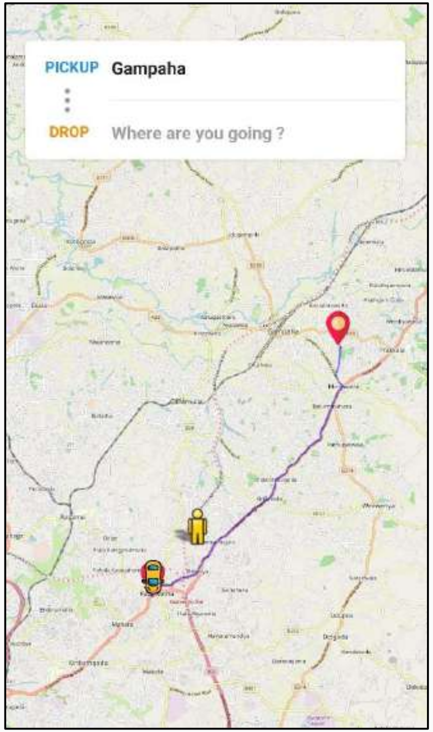
PICKUP Gampaha

DROP Where are you going ?



PICKUP Gampaha

DROP Where are you going ?

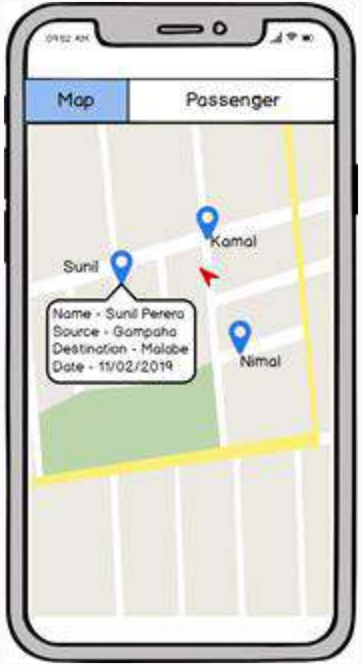


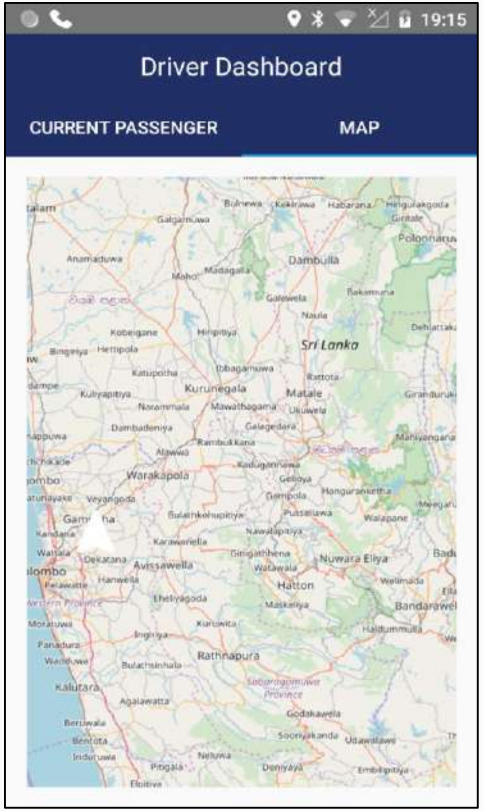
PICKUP Gampaha

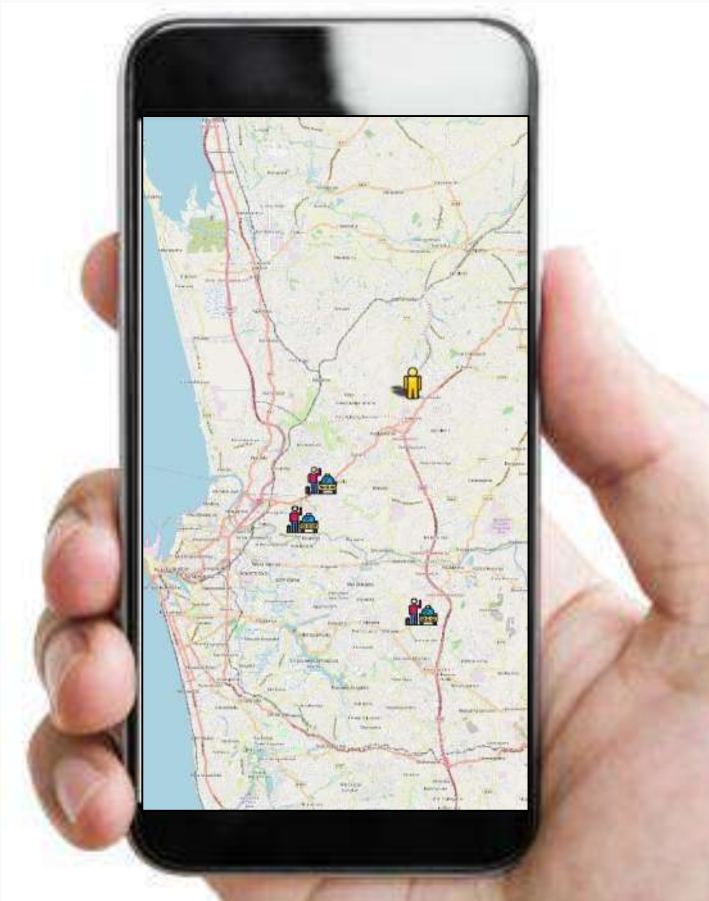
DROP Where are you going ?

Prototype vs Implementation

VIEW CURRENT PASSENGERS

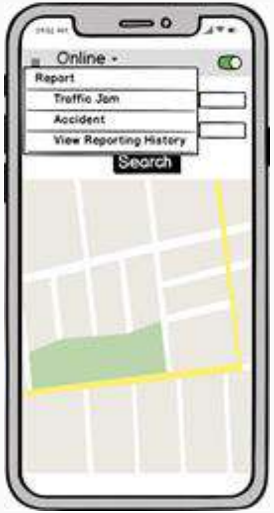




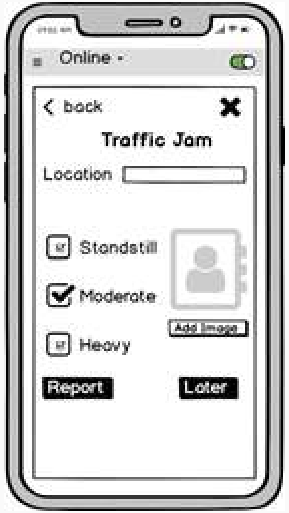


Prototype vs Implementation cont...

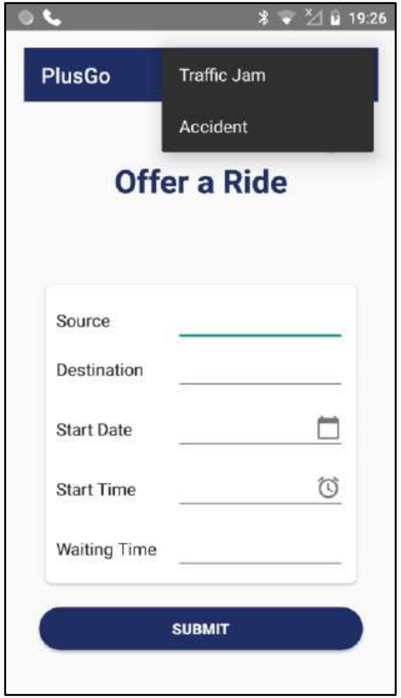
REPORT TRAFFIC JAM



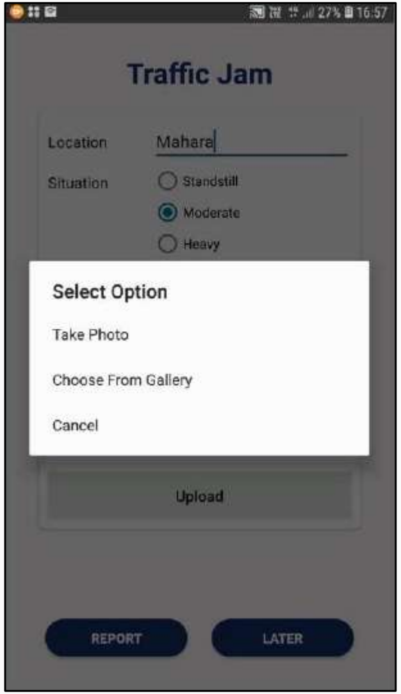
Online
Report
Traffic Jam
Accident
View Reporting History
Search



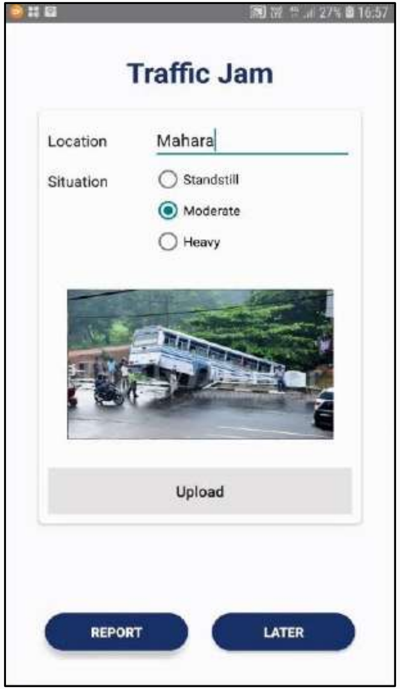
Online
back
Traffic Jam
Location
☑ Standstill
☑ Moderate
☑ Heavy
Add Image
Report
Later



19:26
PlusGo
Traffic Jam
Accident
Offer a Ride
Source
Destination
Start Date
Start Time
Waiting Time
SUBMIT



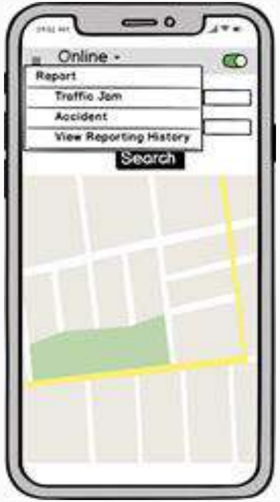
27% 16:57
Traffic Jam
Location Mahara
Situation ○ Standstill
● Moderate
○ Heavy
Select Option
Take Photo
Choose From Gallery
Cancel
Upload
REPORT
LATER



27% 16:57
Traffic Jam
Location Mahara
Situation ○ Standstill
● Moderate
○ Heavy
Upload
REPORT
LATER

Prototype vs Implementation cont...

REPORT ACCIDENT



Online

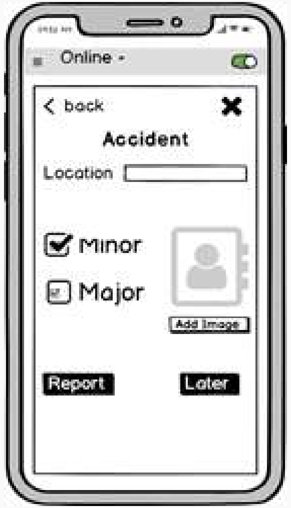
Report

Traffic Jam

Accident

View Reporting History

Search



Online

< back

Accident

Location

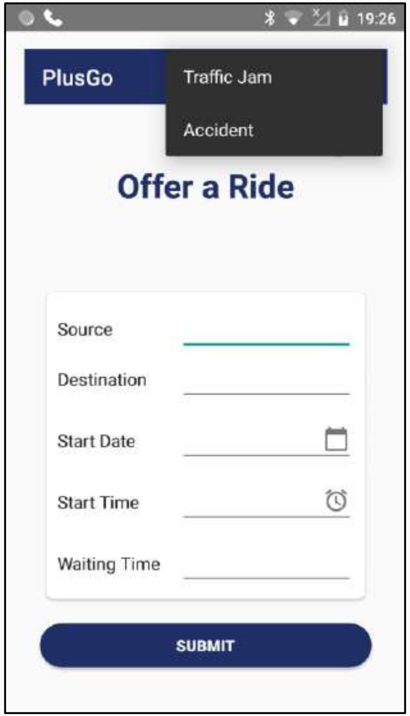
☑ Minor

☐ Major

Add Image

Report

Later



19:26

PlusGo

Traffic Jam

Accident

Offer a Ride

Source

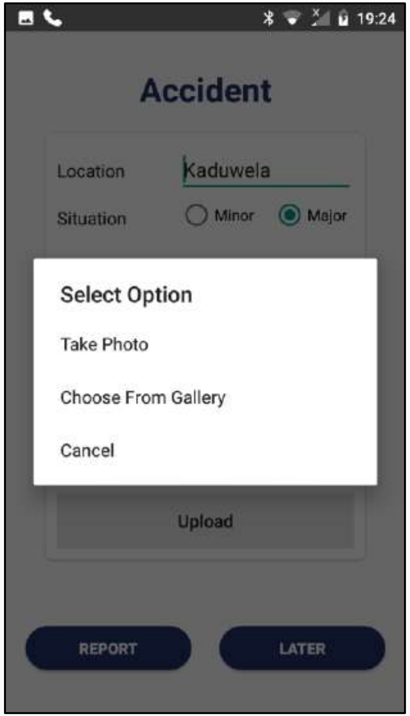
Destination

Start Date

Start Time

Waiting Time

SUBMIT



19:24

Accident

Location: Kaduwela

Situation: ○ Minor ● Major

Select Option

Take Photo

Choose From Gallery

Cancel

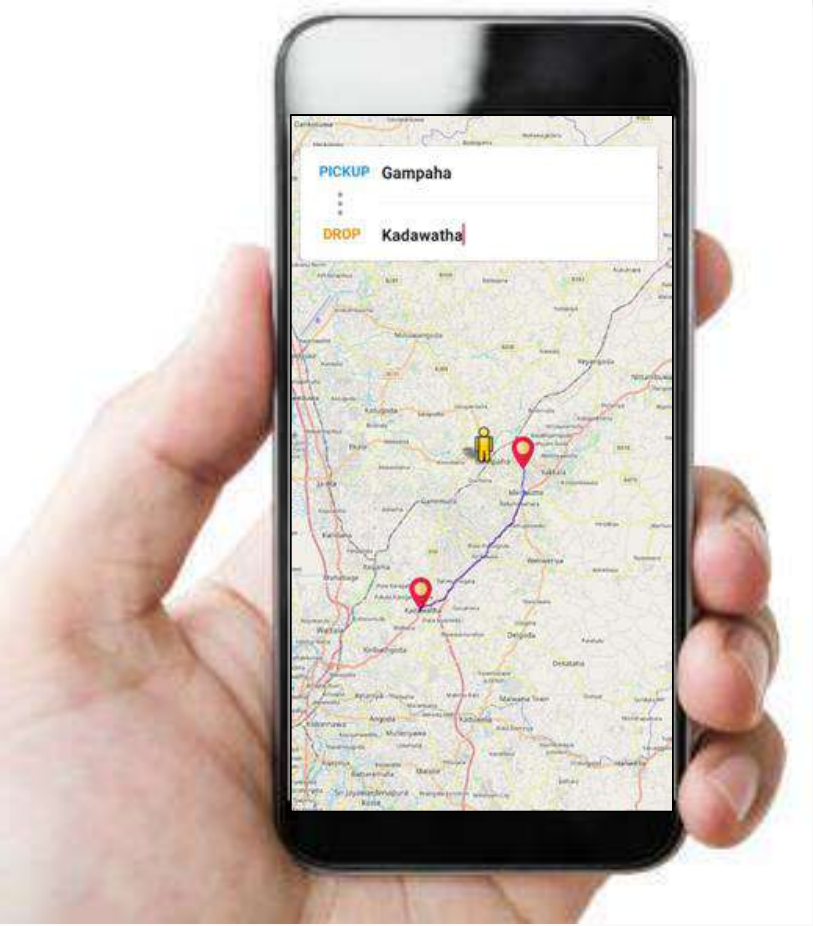
Upload

REPORT

LATER

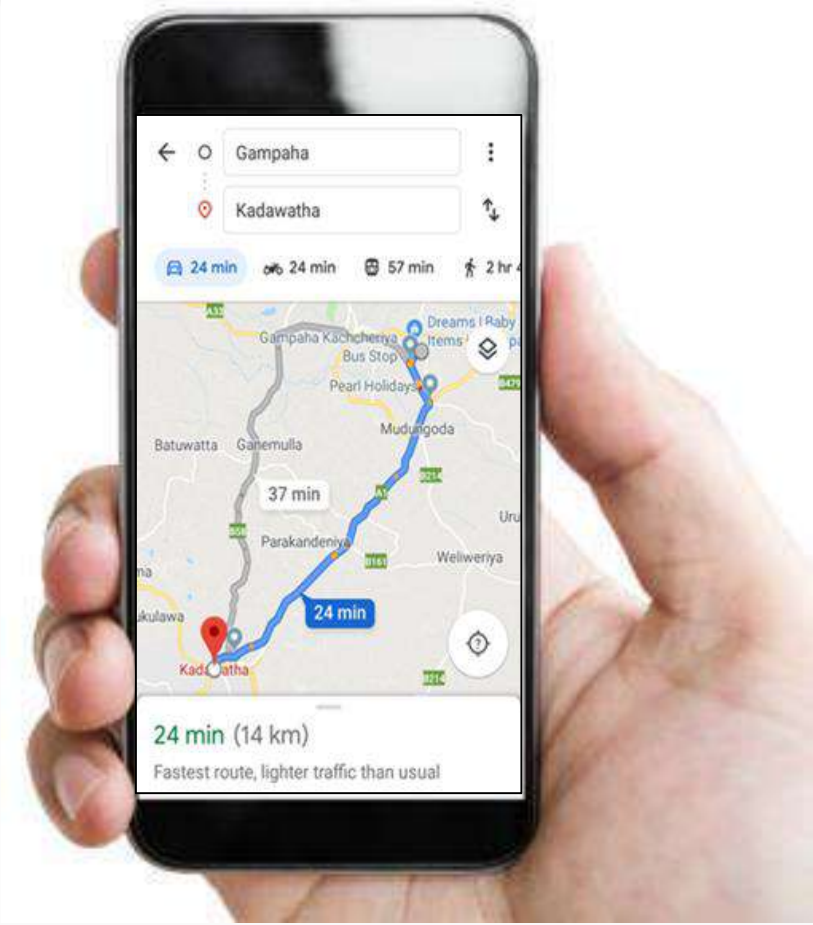
The picture can't be displayed.

RESULTS



PICKUP Gampaha

DROP Kadawatha



Gampaha

Kadawatha

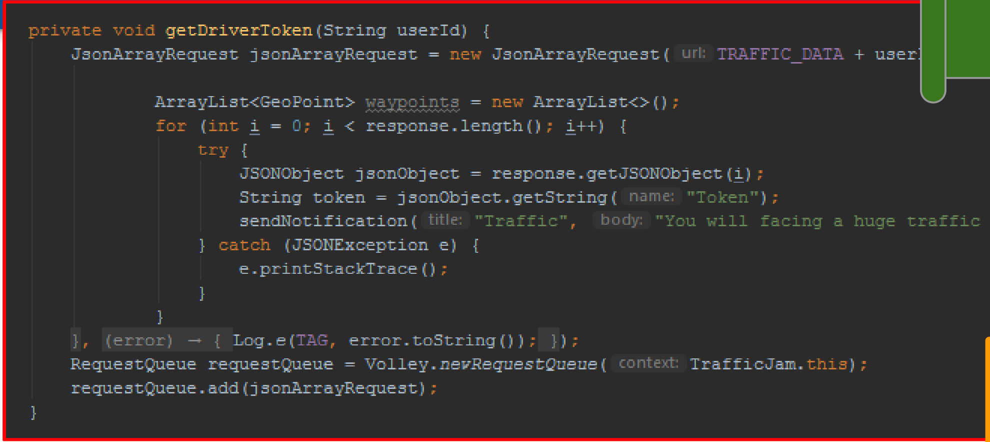
24 min | 24 min | 57 min | 2 hr

24 min (14 km)

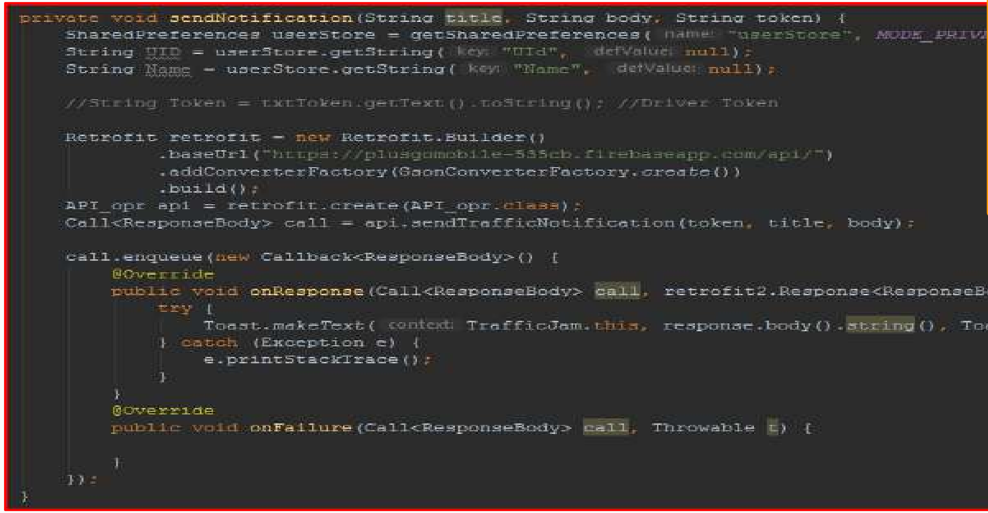
Fastest route, lighter traffic than usual

RESULTS

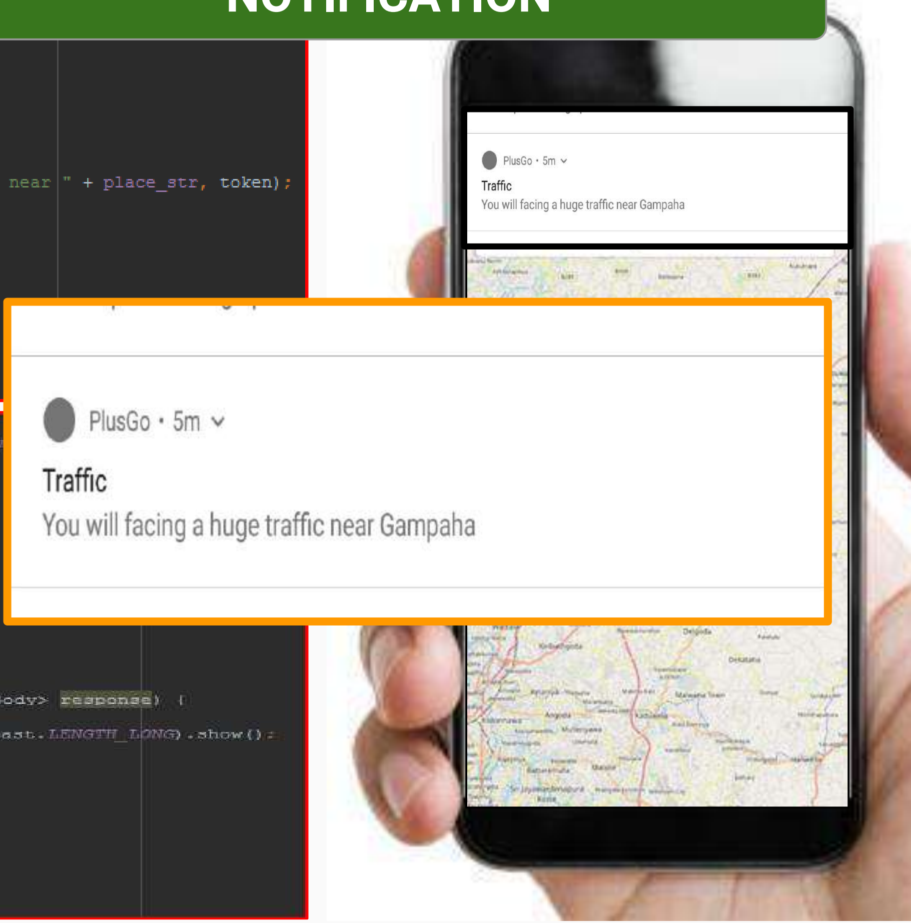
CROWDSOURCING INFORMATION NOTIFICATION



```
private void getDriverToken(String userId) {
    JsonArrayRequest jsonArrayRequest = new JsonArrayRequest( url: TRAFFIC_DATA + user

            ArrayList<GeoPoint> waypoints = new ArrayList<>();
            for (int i = 0; i < response.length(); i++) {
                try {
                    JSONObject jsonObject = response.getJSONObject(i);
                    String token = jsonObject.getString( name: "Token");
                    sendNotification( title: "Traffic",  body: "You will facing a huge traffic near " + place_str, token);
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
    }, (error) → { Log.e(TAG, error.toString()); });
    RequestQueue requestQueue = Volley.newRequestQueue( context: TrafficJam.this);
    requestQueue.add(jsonArrayRequest);
}
```



```
private void sendNotification(String title, String body, String token) {
    SharedPreferences userStore = getSharedPreferences( name: "userStore", MODE_PRIV
    String UID = userStore.getString( key: "UId",  defValue: null);
    String Name = userStore.getString( key: "Name",  defValue: null);

    //String Token = txtToken.getText().toString(); //Driver Token

    Retrofit retrofit = new Retrofit.Builder()
            .baseUrl("https://plusgomobile-535cb.firebaseapp.com/api/")
            .addConverterFactory(GsonConverterFactory.create())
            .build();
    API_opr api = retrofit.create(API_opr.class);
    Call<ResponseBody> call = api.sendTrafficNotification(token, title, body);

    call.enqueue(new Callback<ResponseBody>() {
        @Override
        public void onResponse(Call<ResponseBody> call, retrofit2.Response<ResponseBody> response) {
            try {
                Toast.makeText( context: TrafficJam.this, response.body().string(), Toast.LENGTH_LONG).show();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        @Override
        public void onFailure(Call<ResponseBody> call, Throwable t) {

        }
    });
}
```



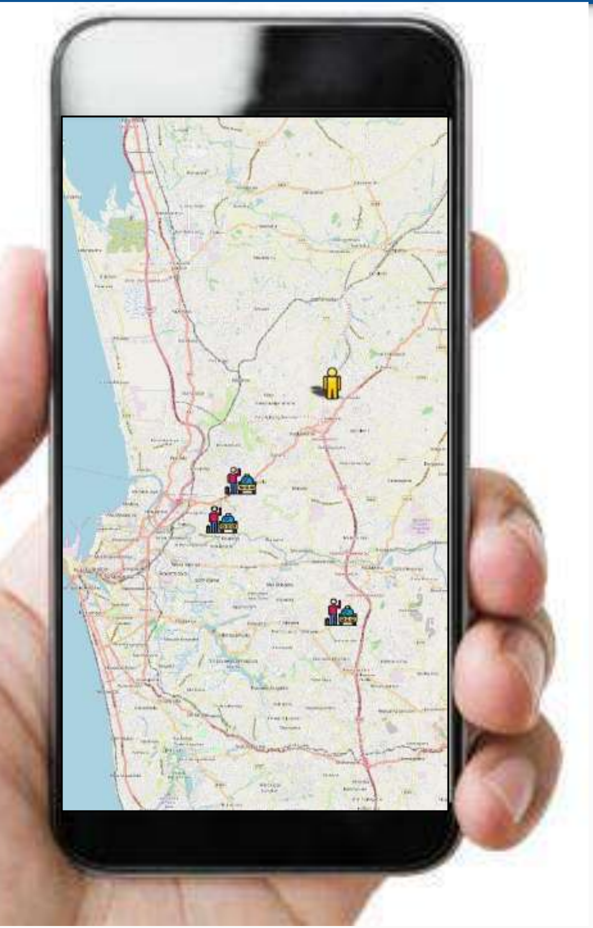
PlusGo • 5m

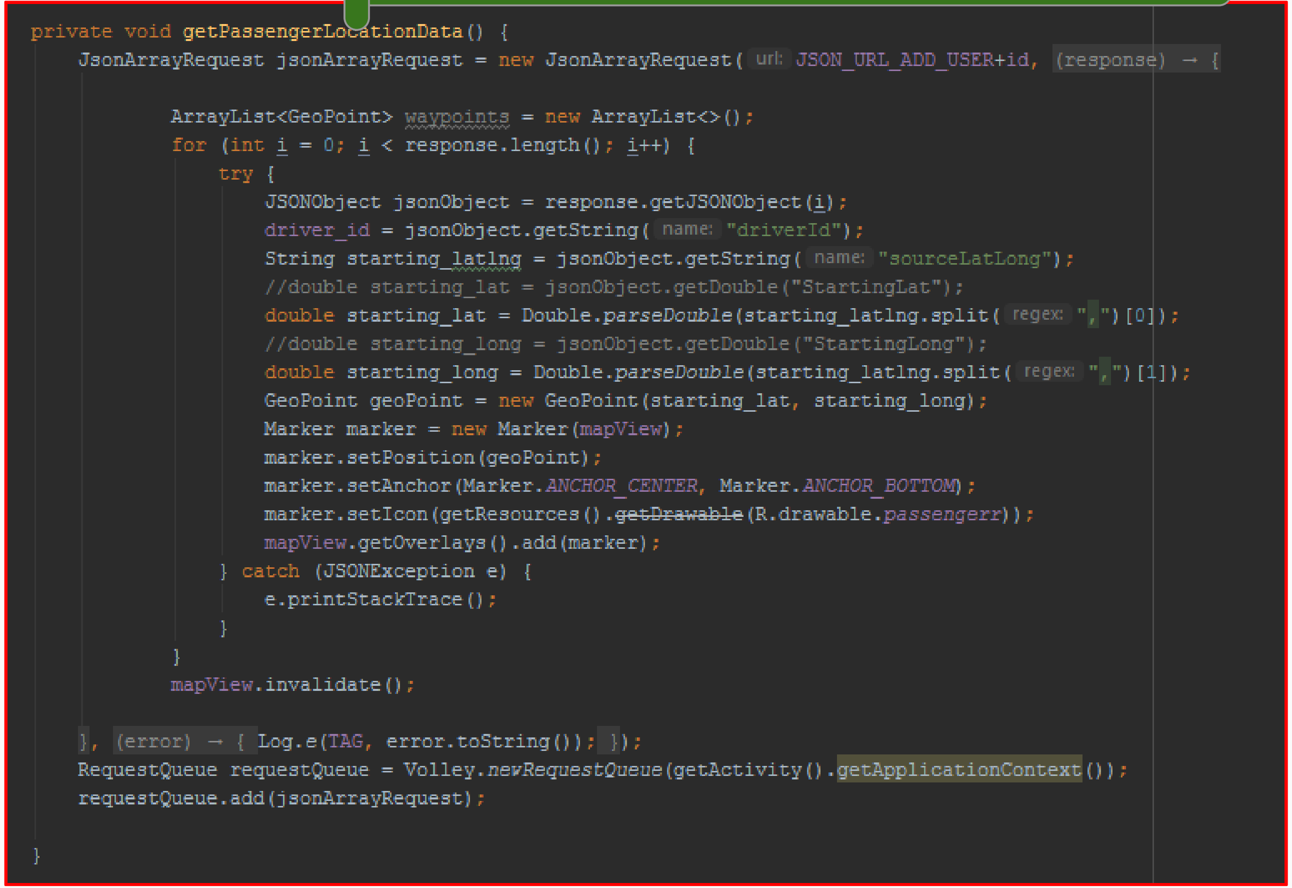
Traffic

You will facing a huge traffic near Gampaha

RESULTS

DISPLAY CURRENT PASSENGERS TO DRIVER

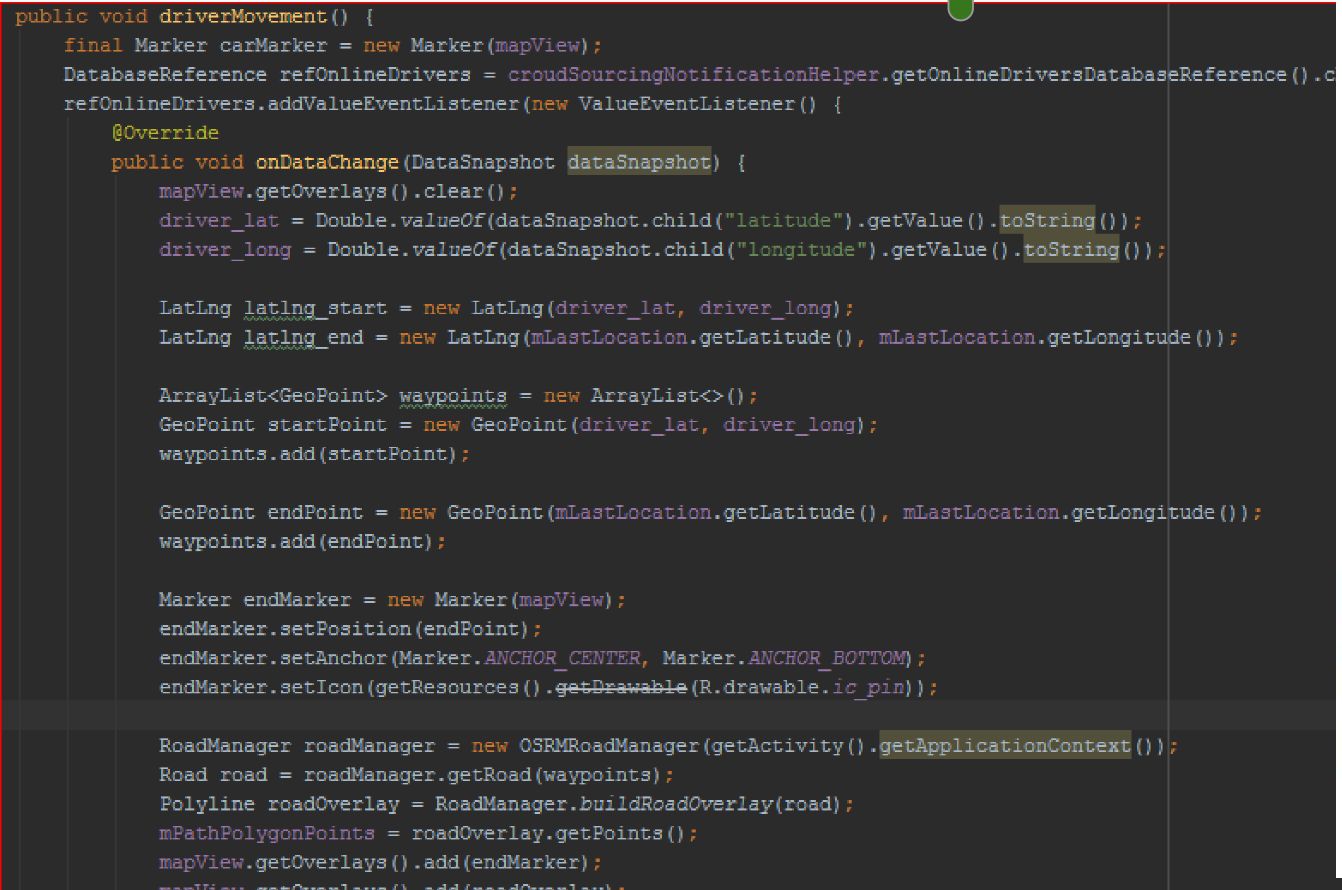




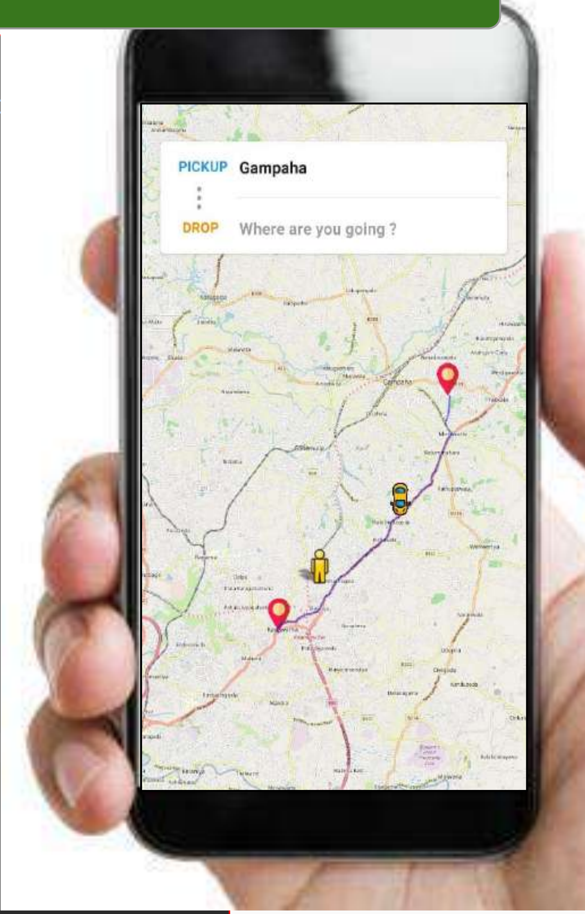
```
private void getPassengerLocationData() {
    JsonArrayRequest jsonArrayRequest = new JsonArrayRequest( url: JSON_URL_ADD_USER+id, (response) → {

            ArrayList<GeoPoint> waypoints = new ArrayList<>();
            for (int i = 0; i < response.length(); i++) {
                try {
                    JSONObject jsonObject = response.getJSONObject(i);
                    driver_id = jsonObject.getString( name: "driverId");
                    String starting_latlng = jsonObject.getString( name: "sourceLatLong");
                    //double starting_lat = jsonObject.getDouble("StartingLat");
                    double starting_lat = Double.parseDouble(starting_latlng.split( regex: ",")[0]);
                    //double starting_long = jsonObject.getDouble("StartingLong");
                    double starting_long = Double.parseDouble(starting_latlng.split( regex: ",")[1]);
                    GeoPoint geoPoint = new GeoPoint(starting_lat, starting_long);
                    Marker marker = new Marker(mapView);
                    marker.setPosition(geoPoint);
                    marker.setAnchor(Marker.ANCHOR_CENTER, Marker.ANCHOR_BOTTOM);
                    marker.setIcon(getResources().getDrawable(R.drawable.passengerr));
                    mapView.getOverlays().add(marker);
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
            mapView.invalidate();

    }, (error) → { Log.e(TAG, error.toString()); });
    RequestQueue requestQueue = Volley.newRequestQueue(getActivity().getApplicationContext());
    requestQueue.add(jsonArrayRequest);

}
```

RESULTS

DISPLAY DRIVER NAVIGATION PATH

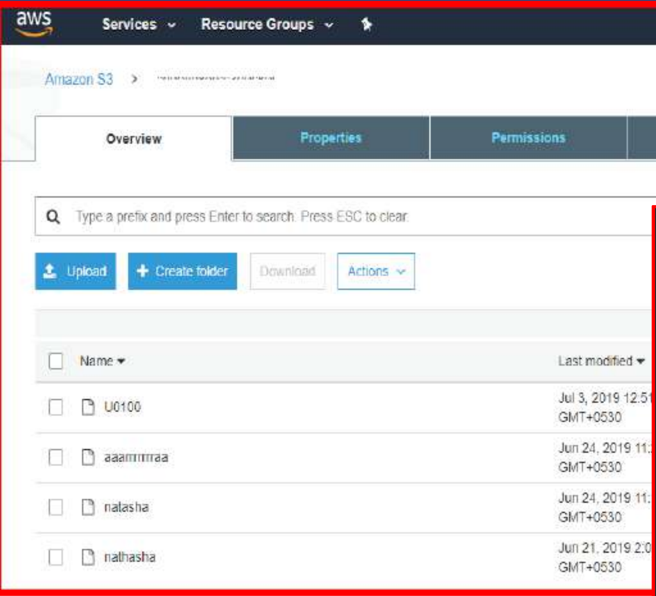


```java
public void driverMovement() {
    final Marker carMarker = new Marker(mapView);
    DatabaseReference refOnlineDrivers = croudSourcingNotificationHelper.getOnlineDriversDatabaseReference().c
    refOnlineDrivers.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            mapView.getOverlays().clear();
            driver_lat = Double.valueOf(dataSnapshot.child("latitude").getValue().toString());
            driver_long = Double.valueOf(dataSnapshot.child("longitude").getValue().toString());

            LatLng latlng_start = new LatLng(driver_lat, driver_long);
            LatLng latlng_end = new LatLng(mLastLocation.getLatitude(), mLastLocation.getLongitude());

            ArrayList<GeoPoint> waypoints = new ArrayList<>();
            GeoPoint startPoint = new GeoPoint(driver_lat, driver_long);
            waypoints.add(startPoint);

            GeoPoint endPoint = new GeoPoint(mLastLocation.getLatitude(), mLastLocation.getLongitude());
            waypoints.add(endPoint);

            Marker endMarker = new Marker(mapView);
            endMarker.setPosition(endPoint);
            endMarker.setAnchor(Marker.ANCHOR_CENTER, Marker.ANCHOR_BOTTOM);
            endMarker.setIcon(getResources().~~getDrawable~~(R.drawable.ic_pin));

            RoadManager roadManager = new OSRMRoadManager(getActivity().getApplicationContext());
            Road road = roadManager.getRoad(waypoints);
            Polyline roadOverlay = RoadManager.buildRoadOverlay(road);
            mPathPolygonPoints = roadOverlay.getPoints();
            mapView.getOverlays().add(endMarker);
            mapView.getOverlays().add(roadOverlay);
```

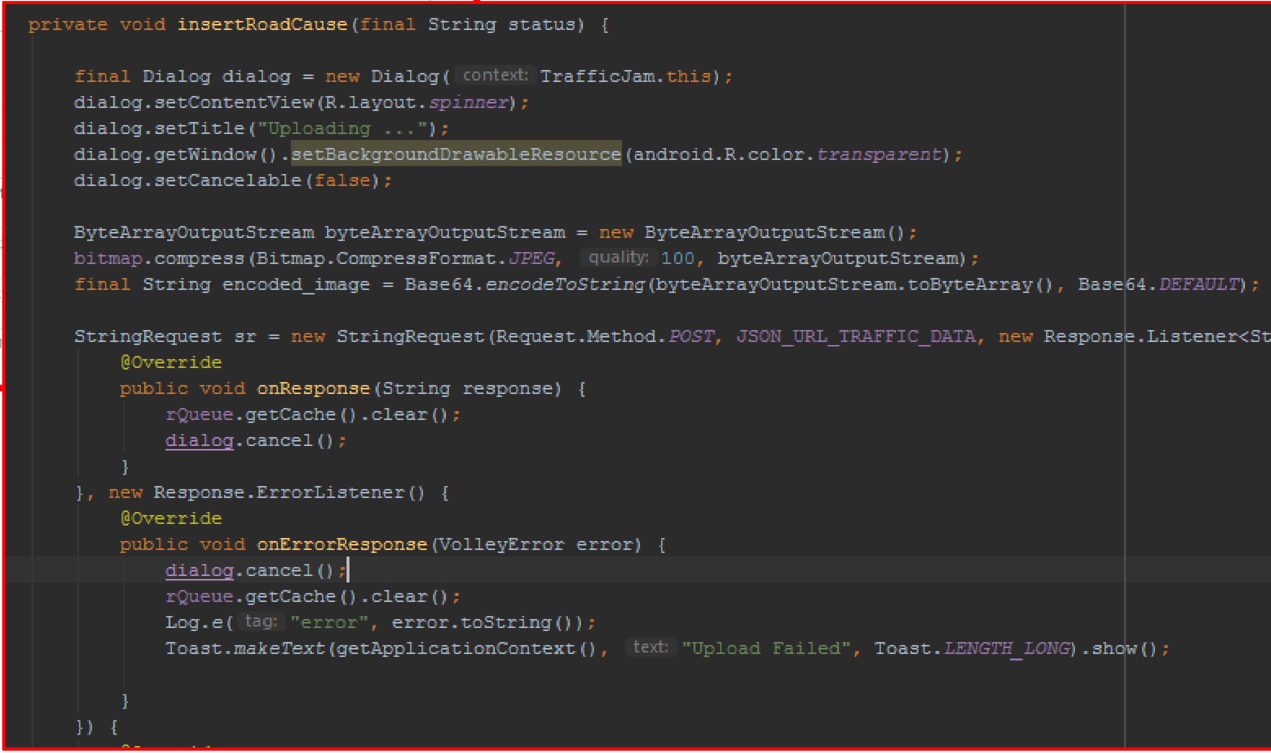


RESULTS

STORE REPORTED IMAGES USING AWS S3 BUCKET



	Name	Last modified
□	U0100	Jul 3, 2019 12:51 GMT+0530
□	aaammraa	Jun 24, 2019 11: GMT+0530
□	natasha	Jun 24, 2019 11: GMT+0530
□	nathasha	Jun 21, 2019 2:0 GMT+0530



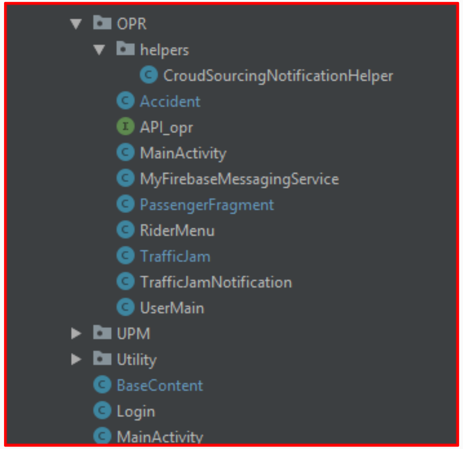
```
private void insertRoadCause(final String status) {

    final Dialog dialog = new Dialog( context: TrafficJam.this);
    dialog.setContentView(R.layout.spinner);
    dialog.setTitle("Uploading ...");
    dialog.getWindow().setBackgroundDrawableResource(android.R.color.transparent);
    dialog.setCancelable(false);

    ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream();
    bitmap.compress(Bitmap.CompressFormat.JPEG,  quality: 100, byteArrayOutputStream);
    final String encoded_image = Base64.encodeToString(byteArrayOutputStream.toByteArray(), Base64.DEFAULT);

    StringRequest sr = new StringRequest(Request.Method.POST, JSON_URL_TRAFFIC_DATA, new Response.Listener<St
        @Override
        public void onResponse(String response) {
            rQueue.getCache().clear();
            dialog.cancel();
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            dialog.cancel();
            rQueue.getCache().clear();
            Log.e( tag: "error", error.toString());
            Toast.makeText(getApplicationContext(),  text: "Upload Failed", Toast.LENGTH_LONG).show();

        }
    }) {
```

PYTHON

AWS

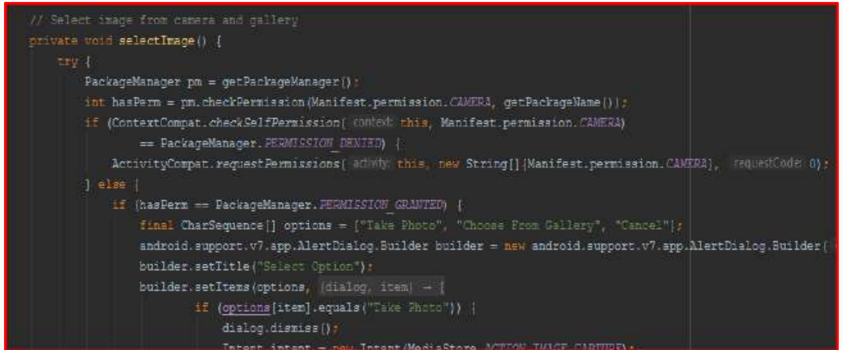
Best Practices

File and folder structure



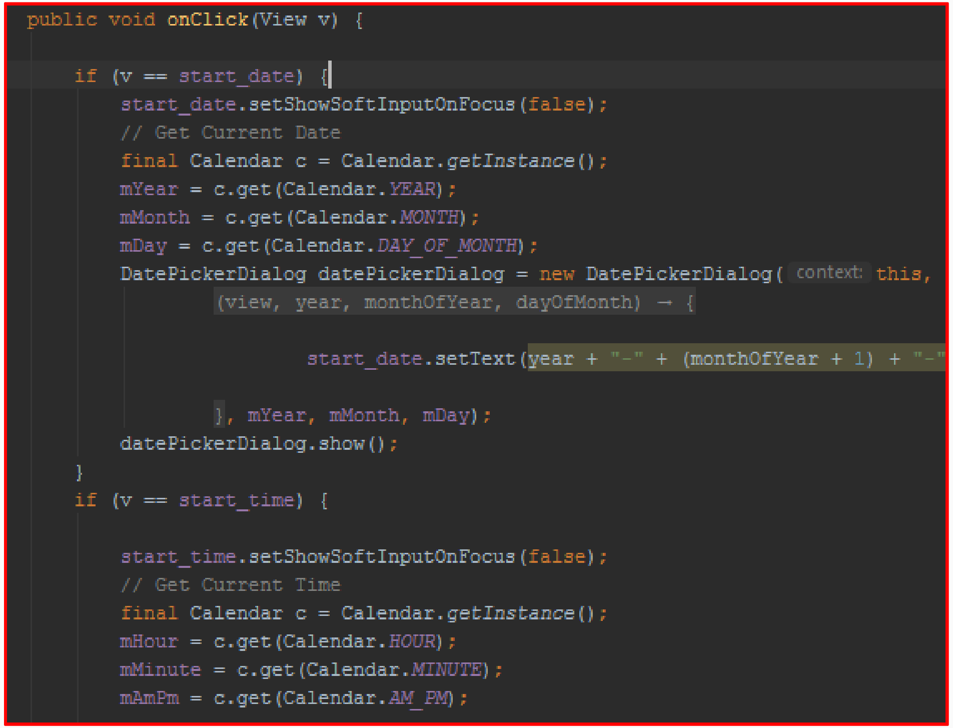
```
▼ OPR
    ▼ helpers
          CroudSourcingNotificationHelper
      Accident
      API_opr
      MainActivity
      MyFirebaseMessagingService
      PassengerFragment
      RiderMenu
      TrafficJam
      TrafficJamNotification
      UserMain
▶ UPM
▶ Utility
  BaseContent
  Login
  MainActivity
```

Commenting



```
// Select image from camera and gallery
private void selectImage() {
    try {
        PackageManager pm = getPackageManager();
        int hasPerm = pm.checkPermission(Manifest.permission.CAMERA, getPackageName());
        if (ContextCompat.checkSelfPermission( context: this, Manifest.permission.CAMERA)
                == PackageManager.PERMISSION_DENIED) {
            ActivityCompat.requestPermissions( activity: this, new String[]{Manifest.permission.CAMERA},  requestCode: 0);
        } else {
            if (hasPerm == PackageManager.PERMISSION_GRANTED) {
                final CharSequence[] options = {"Take Photo", "Choose From Gallery", "Cancel"};
                android.support.v7.app.AlertDialog.Builder builder = new android.support.v7.app.AlertDialog.Builder(
                builder.setTitle("Select Option");
                builder.setItems(options, (dialog, item) → {
                        if (options[item].equals("Take Photo")) {
                            dialog.dismiss();
                            Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
```

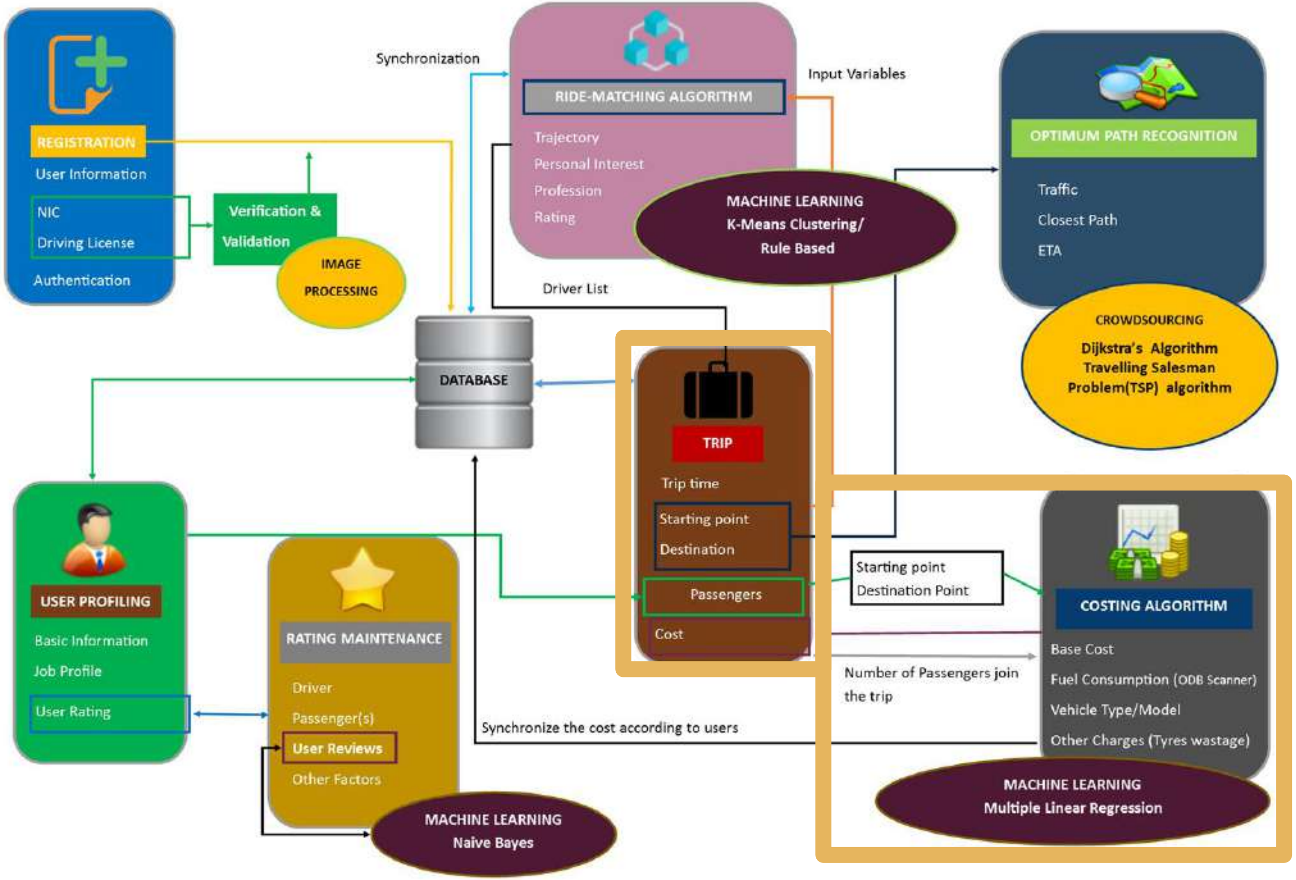
Consistent Indentation



```
public void onClick(View v) {

    if (v == start_date) {
        start_date.setShowSoftInputOnFocus(false);
        // Get Current Date
        final Calendar c = Calendar.getInstance();
        mYear = c.get(Calendar.YEAR);
        mMonth = c.get(Calendar.MONTH);
        mDay = c.get(Calendar.DAY_OF_MONTH);
        DatePickerDialog datePickerDialog = new DatePickerDialog( context: this,
                (view, year, monthOfYear, dayOfMonth) → {

                        start_date.setText(year + "-" + (monthOfYear + 1) + "-"

                }, mYear, mMonth, mDay);
        datePickerDialog.show();
    }
    if (v == start_time) {

        start_time.setShowSoftInputOnFocus(false);
        // Get Current Time
        final Calendar c = Calendar.getInstance();
        mHour = c.get(Calendar.HOUR);
        mMinute = c.get(Calendar.MINUTE);
        mAmPm = c.get(Calendar.AM_PM);
```

Fare Calculation

High Level Diagram



Objectives

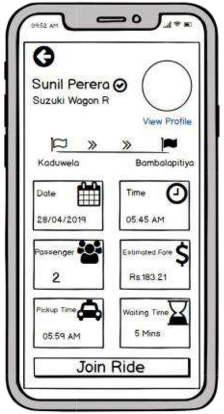
Main objective is to examine how fare calculation is divided among passengers and communicate both drivers and passengers via firebase push notifications related about trip.

Research Gap

Features	UBER	UDIO	Carpooling.lk	RideShare.lk	+GO
The system will decide the estimated fare before joining the trip.	✓	✓	✗	✗	✓
Vehicle fuel consumption calculated according to the condition of the vehicle	✗	✗	✗	✗	✓
Passengers can get off in any place where is the between source and destination because the fare will calculate according to the fuel consumption of the vehicle	✗	✗	✗	✗	✓

Prototype vs Implementation

Interfaces related to Request trip for Passenger

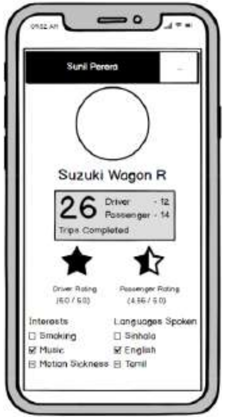


Sunil Perera
Suzuki Wagon R
View Profile

Kaduwela
Bambalapitiya

Date 28/04/2019
Time 05.45 AM
Passenger 2
Estimated Fare Rs.183.21
Pickup Time 05.59 AM
Waiting Time 5 Mins

Join Ride



Sunil Perera

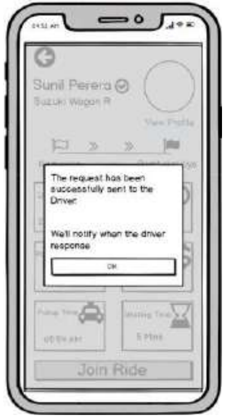
Suzuki Wagon R

26 Driver - 12
Passenger - 14
Trips Completed

Driver Rating (5.0 / 5.0)
Passenger Rating (4.56 / 5.0)

Interests
□ Smoking
☑ Music
☒ Motion Sickness

Languages Spoken
□ Sinhala
☑ English
☒ Tamil



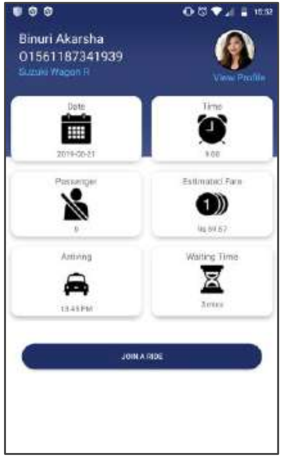
Sunil Perera
Suzuki Wagon R
View Profile

The request has been successfully sent to the Driver.

We'll notify when the driver response

OK

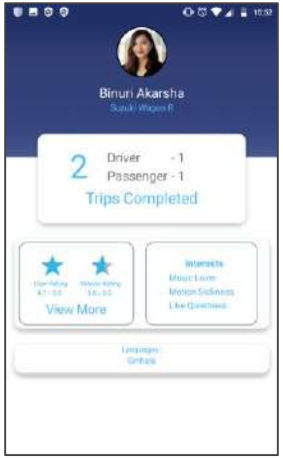
05.59 AM
5 Mins
Join Ride



Binuri Akarsha
01561187341939
Suzuki Wagon R
View Profile

Date 2019-05-21
Time 9:00
Passenger 0
Estimated Fare Rs 59.67
Arriving 13.45 PM
Waiting Time 3mins

JOIN A RIDE



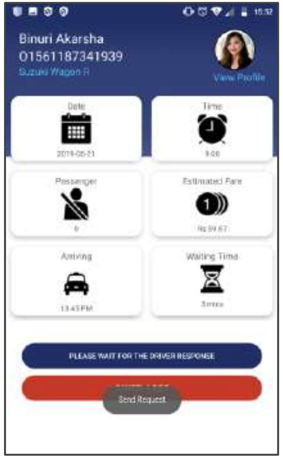
Binuri Akarsha
Suzuki Wagon R

2 Driver - 1
Passenger - 1
Trips Completed

User Rating 4.7 / 5.0
Vehicle Rating 3.6 / 5.0
View More

Interests
Music Lover
Motion Sickness
Like Quietness

Languages
Sinhala



Binuri Akarsha
01561187341939
Suzuki Wagon R
View Profile

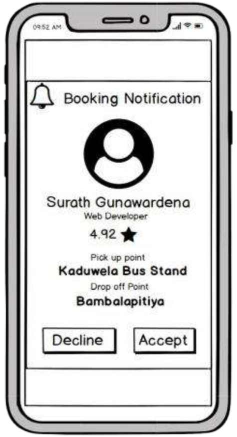
Date 2019-05-21
Time 9:00
Passenger 0
Estimated Fare Rs 59.67
Arriving 13.45PM
Waiting Time 3mins

PLEASE WAIT FOR THE DRIVER RESPONSE

Send Request

Prototype vs Implementation

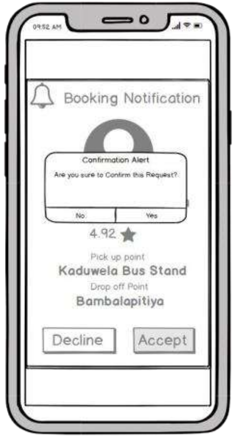
Request trip
Related Interfaces
for
Driver



Booking Notification

Surath Gunawardena
Web Developer
4.92 ★
Pick up point
Kaduwela Bus Stand
Drop off Point
Bambalapitiya

Decline | Accept

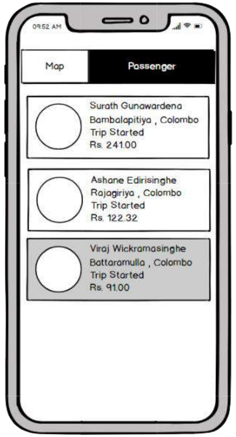


Booking Notification

Confirmation Alert
Are you sure to Confirm this Request?
No | Yes

4.92 ★
Pick up point
Kaduwela Bus Stand
Drop off Point
Bambalapitiya

Decline | Accept

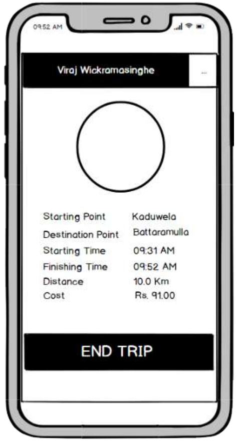


Map | Passenger

Surath Gunawardena
Bambalapitiya , Colombo
Trip Started
Rs. 241.00

Ashane Edirisinghe
Rajagiriya , Colombo
Trip Started
Rs. 122.32

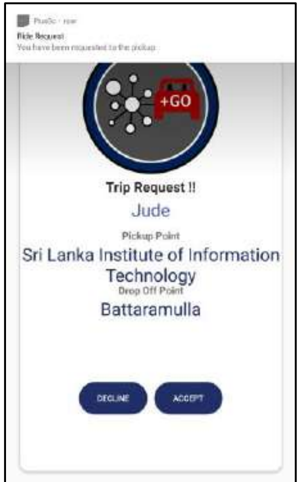
Viraj Wickramasinghe
Battaramulla , Colombo
Trip Started
Rs. 91.00



Viraj Wickramasinghe

Starting Point	Kaduwela
Destination Point	Battaramulla
Starting Time	09:31 AM
Finishing Time	09:52 AM
Distance	10.0 Km
Cost	Rs. 91.00

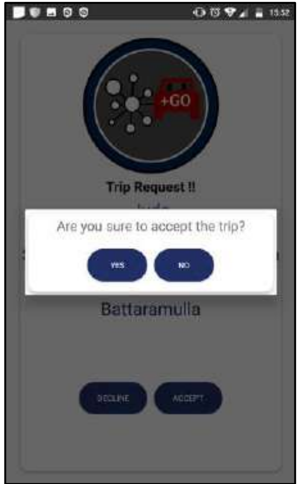
END TRIP



PlusGo · now
Ride Request
You have been requested to the pickup

Trip Request !!
Jude
Pickup Point
Sri Lanka Institute of Information Technology
Drop Off Point
Battaramulla

DECLINE | ACCEPT

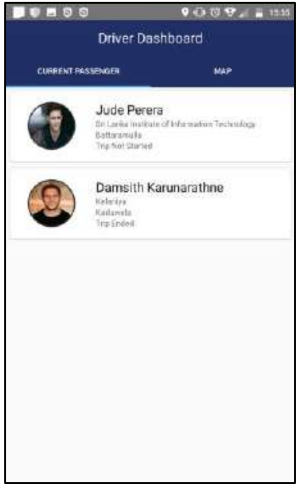


Trip Request !!

Are you sure to accept the trip?
YES | NO

Battaramulla

DECLINE | ACCEPT

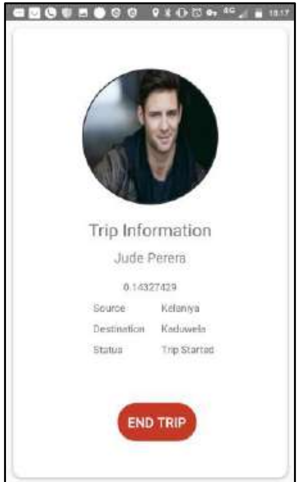


Driver Dashboard

CURRENT PASSENGER | MAP

Jude Perera
Sri Lanka Institute of Information Technology
Battaramulla
Trip Not Started

Damsith Karunarathne
Kelaniya
Kaduwela
Trip Ended



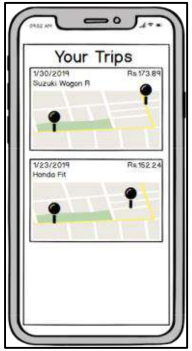
Trip Information
Jude Perera
0.14327429

Source	Kelaniya
Destination	Kaduwela
Status	Trip Started

END TRIP

Prototype vs Implementation

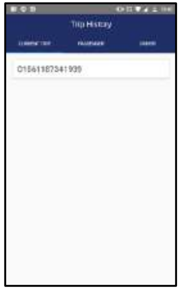
Trip History
Related Interface



Your Trips

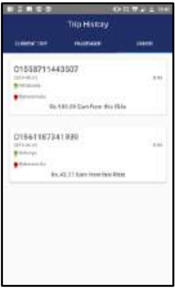
1/30/2019 Rs.173.89
Suzuki Wagon R

1/23/2019 Rs.152.24
Honda Fit



Trip History

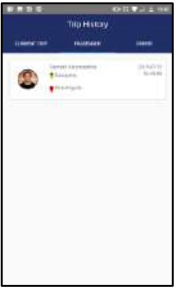
01561187341939



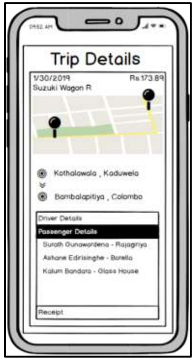
Trip History

01558711443507

01561187341939



Trip History



Trip Details

1/30/2019 Rs.173.89
Suzuki Wagon R

Kothalawala , Kaduwela

Bambalapitiya , Colombo

Driver Details

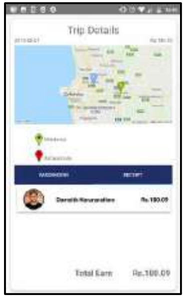
Passenger Details

Surath Gunawardena - Rajagiriya

Ashane Edirisinghe - Borella

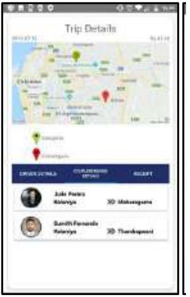
Kalum Bandara - Glass House

Receipt

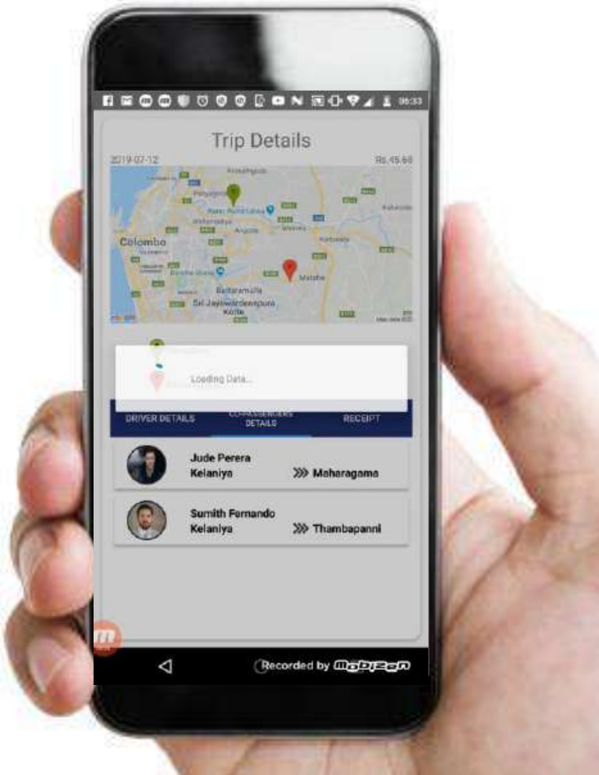


Trip Details

Total Earn Rs.180.09



Trip Details



Trip Details

2019-07-12 Rs.45.68

Loading Data...

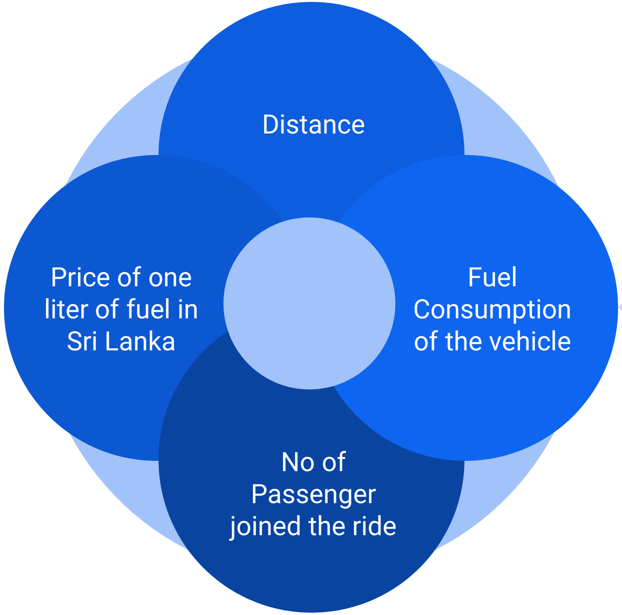
DRIVER DETAILS | CO-PASSENGERS DETAILS | RECEIPT

Jude Perera
Kelaniya
Maharagama

Sumith Fernando
Kelaniya
Thambapanni

The Specialized Area and the Use of Technologies

Estimated fare calculation before Join the Ride


Distance
Price of one liter of fuel in Sri Lanka
Fuel Consumption of the vehicle
No of Passenger joined the ride

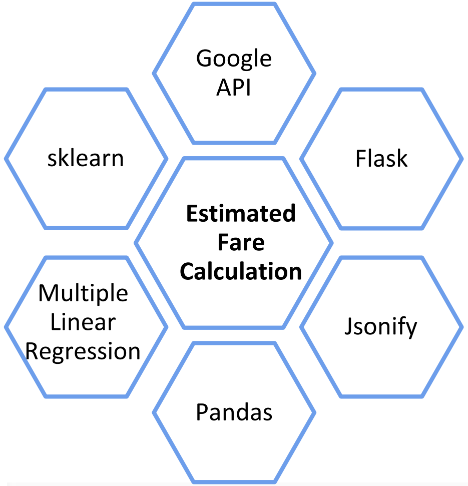
Attributes of Selected Driver's Vehicle

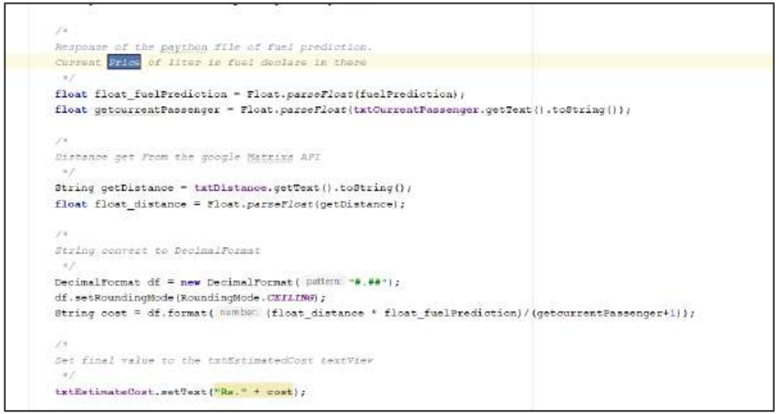
- Manufacture Year
- Registered Year
- No of Cylinders
- Fuel Type
- Engine Capacity
- Engine Power
- Mileage

The Specialized Area and the Use of Technologies

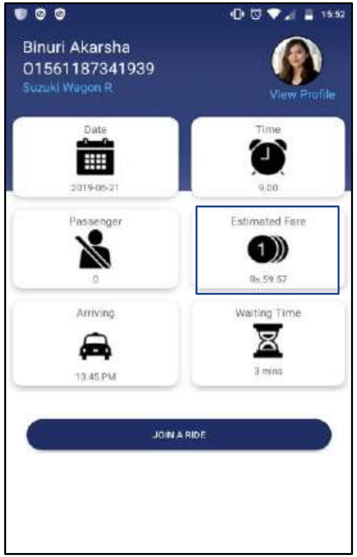
Estimated fare calculation before Join the Ride

$$Estimated\ fare\ Calculation = \frac{\left(\frac{Price\ of\ one\ liter\ of\ fuel}{fuel\ Consumption}\right) * distance}{No\ of\ passengers}$$



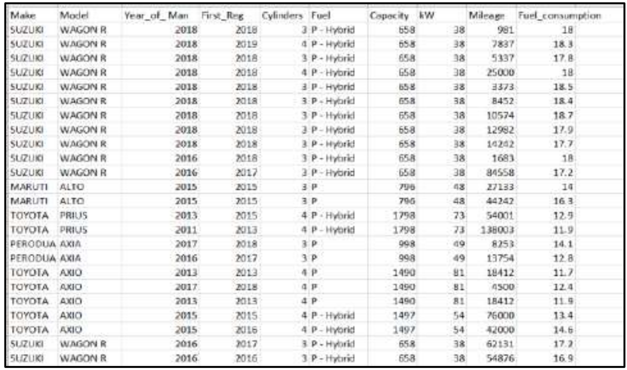


```
/*
Response of the paython file of fuel prediction.
Current Price of liter is fuel declare in there
 */
float float_fuelPrediction = Float.parseFloat(fuelPrediction);
float getcurrentPassenger = Float.parseFloat(txtCurrentPassenger.getText().toString());

/*
Distance get From the google Matrixs API
 */
String getDistance = txtDistance.getText().toString();
float float_distance = Float.parseFloat(getDistance);

/*
String convert to DecimalFormat
 */
DecimalFormat df = new DecimalFormat( pattern: "#.##");
df.setRoundingMode(RoundingMode.CEILING);
String cost = df.format( number: (float_distance * float_fuelPrediction)/(getcurrentPassenger+1));

/*
Set final value to the txtEstimatedCost textView
 */
txtEstimateCost.setText("Rs." + cost);
```

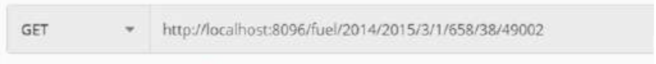


Binuri Akarsha
01561187341939
Suzuki Wagon R

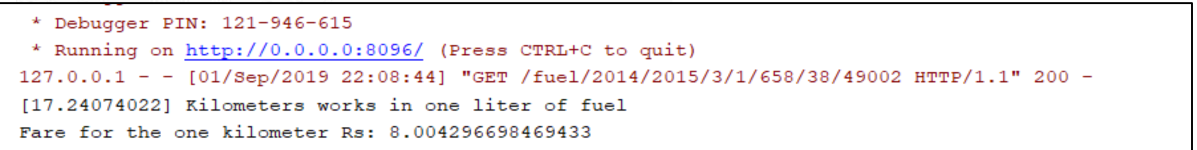
Date: 2019-06-21
Time: 9.00
Passenger: 0
Estimated Fare: Rs.59.57
Arriving: 13.45 PM
Waiting Time: 3 mins



Make	Model	Year_of_Man	First_Reg	Cylinders	Fuel	Capacity	kW	Mileage	Fuel_consumption
SUZUKI	WAGON R	2018	2018	3	P - Hybrid	658	38	981	18
SUZUKI	WAGON R	2018	2019	4	P - Hybrid	658	38	7837	18.3
SUZUKI	WAGON R	2018	2018	3	P - Hybrid	658	38	5337	17.8
SUZUKI	WAGON R	2018	2018	4	P - Hybrid	658	38	25000	18
SUZUKI	WAGON R	2018	2018	3	P - Hybrid	658	38	3373	18.5
SUZUKI	WAGON R	2018	2018	3	P - Hybrid	658	38	8452	18.4
SUZUKI	WAGON R	2018	2018	3	P - Hybrid	658	38	10574	18.7
SUZUKI	WAGON R	2018	2018	3	P - Hybrid	658	38	12982	17.9
SUZUKI	WAGON R	2018	2018	3	P - Hybrid	658	38	14242	17.7
SUZUKI	WAGON R	2016	2018	3	P - Hybrid	658	38	1683	18
SUZUKI	WAGON R	2016	2017	3	P - Hybrid	658	38	84558	17.2
MARUTI	ALTO	2015	2015	3	P	796	48	27133	14
MARUTI	ALTO	2015	2015	3	P	796	48	44242	16.3
TOYOTA	PRIUS	2013	2015	4	P - Hybrid	1798	73	54001	12.9
TOYOTA	PRIUS	2011	2013	4	P - Hybrid	1798	73	138003	11.9
PERODUA	AXIA	2017	2018	3	P	998	49	8253	14.1
PERODUA	AXIA	2016	2017	3	P	998	49	13754	12.8
TOYOTA	AXIO	2013	2013	4	P	1490	81	18412	11.7
TOYOTA	AXIO	2017	2018	4	P	1490	81	4500	12.4
TOYOTA	AXIO	2013	2013	4	P	1490	81	18412	11.9
TOYOTA	AXIO	2015	2015	4	P - Hybrid	1497	54	76000	13.4
TOYOTA	AXIO	2015	2016	4	P - Hybrid	1497	54	42000	14.6
SUZUKI	WAGON R	2016	2017	3	P - Hybrid	658	38	62131	17.2
SUZUKI	WAGON R	2016	2016	3	P - Hybrid	658	38	54876	16.9



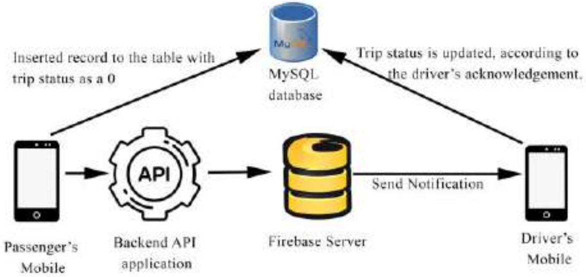
```
GET   http://localhost:8096/fuel/2014/2015/3/1/658/38/49002
```



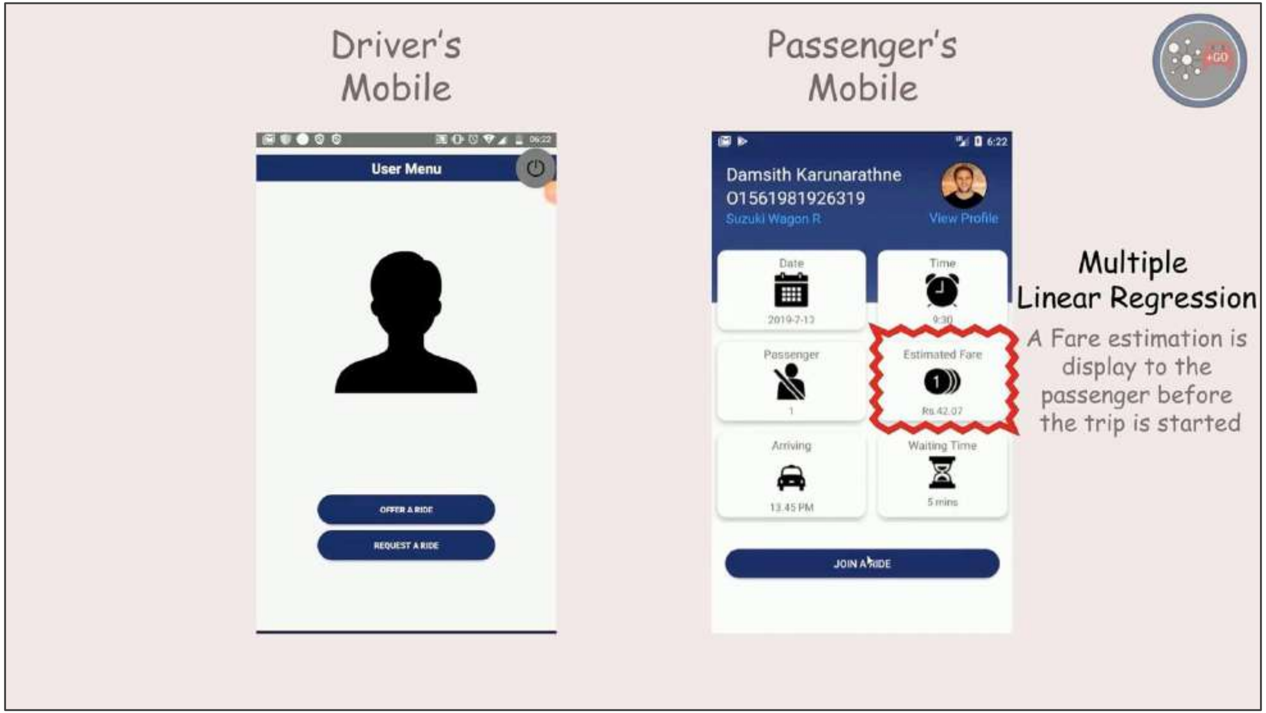
```
 * Debugger PIN: 121-946-615
 * Running on http://0.0.0.0:8096/ (Press CTRL+C to quit)
127.0.0.1 - - [01/Sep/2019 22:08:44] "GET /fuel/2014/2015/3/1/658/38/49002 HTTP/1.1" 200 -
[17.24074022] Kilometers works in one liter of fuel
Fare for the one kilometer Rs: 8.004296698469433
```

The Specialized Area and the Use of Technologies

Join a Ride



Inserted record to the table with trip status as a 0
MySQL database
Trip status is updated, according to the driver's acknowledgement.
API
Send Notification
Passenger's Mobile
Backend API application
Firebase Server
Driver's Mobile


Driver's Mobile

Passenger's Mobile

Damsith Karunarathne
01561981926319
Suzuki Wagon R

Date: 2019-7-13

Time: 9:30

Passenger: 1

Estimated Fare: Rs.42.07

Arriving: 13.45 PM

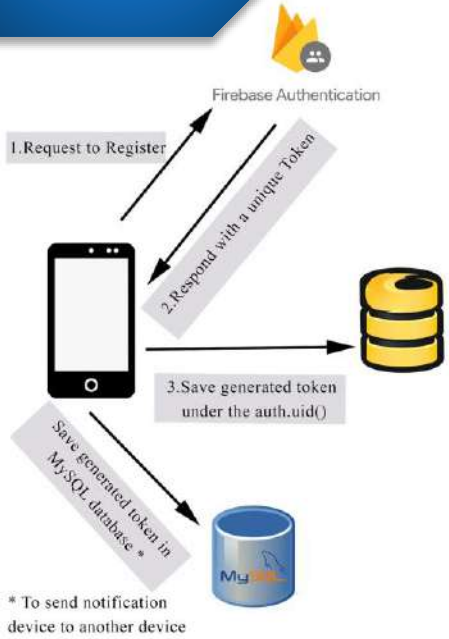
Waiting Time: 5 mins

Multiple Linear Regression

A Fare estimation is display to the passenger before the trip is started

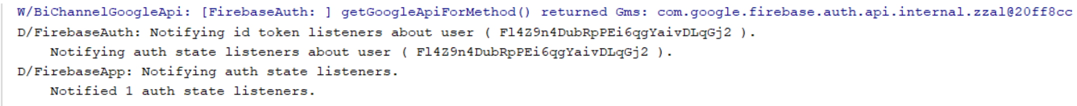
The Specialized Area and the Use of Technologies

Firebase Notification


Firebase Authentication
1.Request to Register
2.Respond with a unique Token
3.Save generated token under the auth.uid()
Save generated token in MySQL database *
MySQL

* To send notification device to another device

User Authenticates to a firebase application

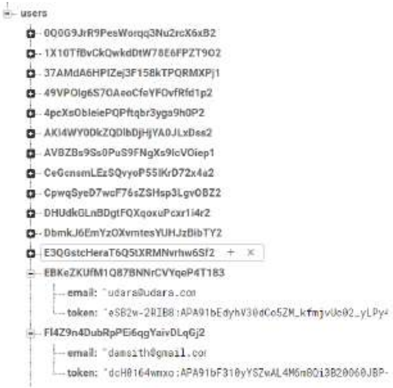


```
W/BiChannelGoogleApi: [FirebaseAuth: ] getGoogleApiForMethod() returned Gms: com.google.firebase.auth.api.internal.zzal@20ff8cc
D/FirebaseAuth: Notifying id token listeners about user ( Fl4Z9n4DubRpPEi6qgYaivDLqGj2 ).
    Notifying auth state listeners about user ( Fl4Z9n4DubRpPEi6qgYaivDLqGj2 ).
D/FirebaseApp: Notifying auth state listeners.
    Notified 1 auth state listeners.
```

Generate Firebase Cloud Messaging Token for Device

Stored in Firebase Real-time database

Stored in MySQL Database



```
users
  0Q0G9JrR9PesWorqq3Nu2rcX6xB2
  1X10TfBvCkQwkdDtW78E6FPZT9O2
  37AMdA6HPIZej3F158kTPQRMXPj1
  49VPOlg6S7OAeoCfeYFOvfRfd1p2
  4pcXsObleiePQPftqbr3yga9h0P2
  AKI4WY0DkZQDlbDjHjYA0JLxDss2
  AVBZBs9Ss0PuS9FNgXs9lcVOiep1
  CeGcnsmLEzSQvyoP55IKrD72x4a2
  CpwqSyeD7wcF76sZSHsp3LgvOBZ2
  DHUdkGLnBDgtFQXqoxuPcxr1i4r2
  DbmkJ6EmYzOXvmtesYUHJzBibTY2
  E3QGstcHeraT6Q5tXRMNvrhw6Sf2  +  ×
  EBKeZKUfM1Q87BNNrCVYqeP4T183
    email: "udara@udara.com
    token: "eSB2w-2RIB8:APA91bEdyhV30dCo5ZM_kfmjvUc02_yLPy
  Fl4Z9n4DubRpPEi6qgYaivDLqGj2
    email: "damsith@gmail.cor
    token: "dcH0164wmxo:APA91bF310yYSZwAL4M6m0Oi3B2006OJBP-
```

The Specialized Area and the Use of Technologies

Actual Fare Calculation

Fuel Consumption

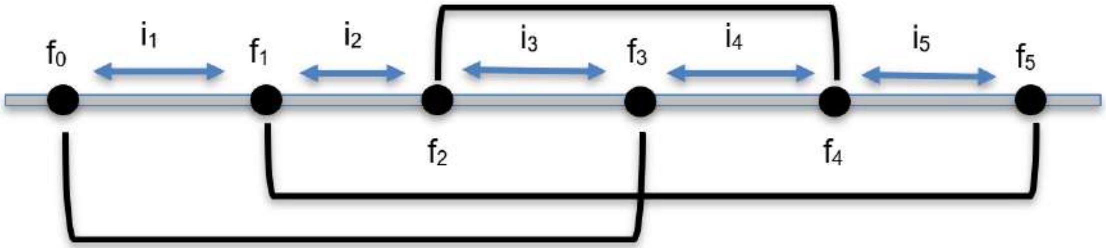
No of Passengers

Current price of the fuel

What is the Segment ?

If new passenger joins the trip or passenger end up the trip, new segment will be created.

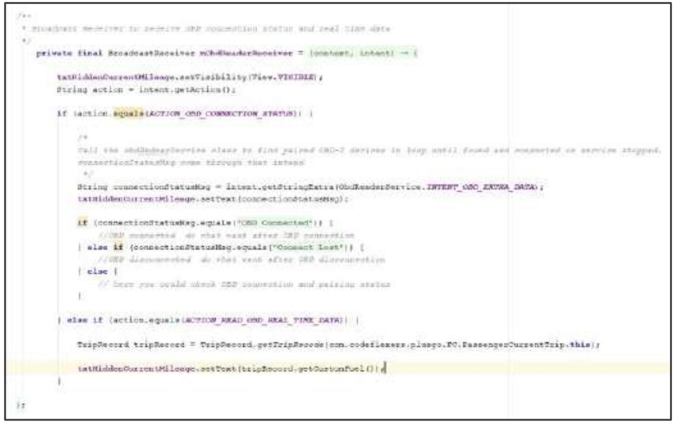
$$C\ i\ \text{(Total fare for the i th segment)} = \frac{\text{Fuel Consumption} * \text{price}}{\sum \text{passenger}}$$



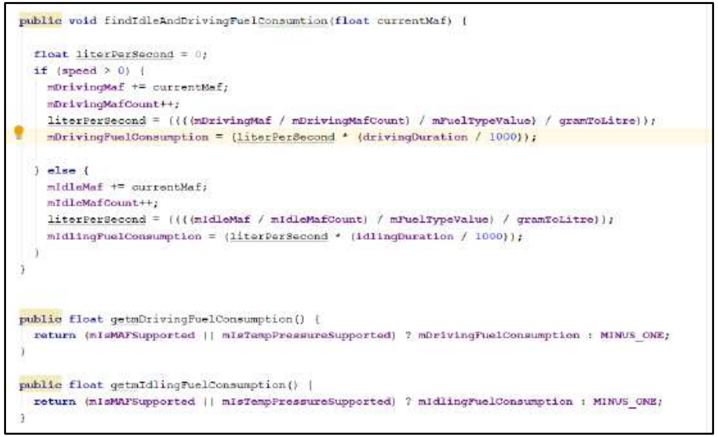
$$\text{Total Fare of the Ride} = \sum_{i=Start\ point}^{end\ point} (\text{C i})$$

The Specialized Area and the Use of Technologies

How OBD Reads Fuels consumption

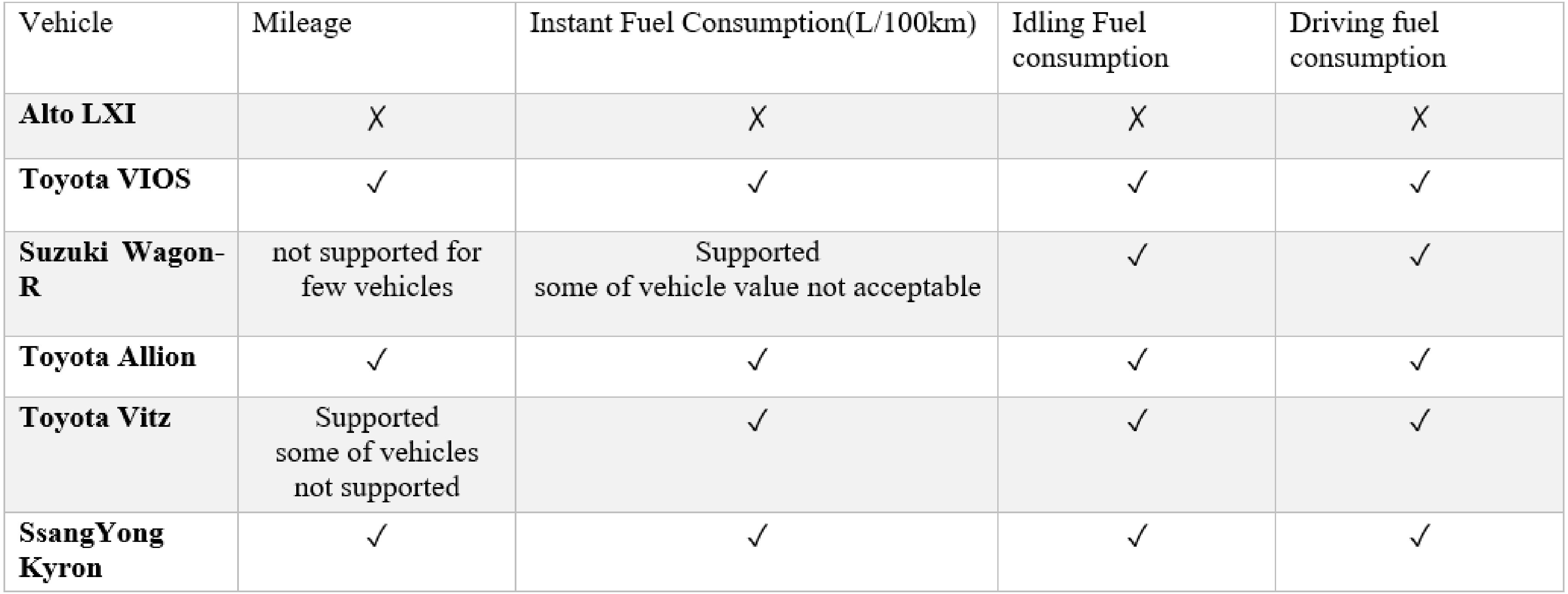


```
/**
 * Broadcast Receiver to receive OBD connection status and real time data
 */
    private final BroadcastReceiver mObdReaderReceiver = (context, intent) → {

        txtHiddenCurrentMileage.setVisibility(View.VISIBLE);
        String action = intent.getAction();

        if (action.equals(ACTION_OBD_CONNECTION_STATUS)) {

            /*
            Call the obdReaderService class to find paired OBD-2 devices in loop until found and connected or service stopped.
            connectionStatusMsg come through that intent
             */
            String connectionStatusMsg = intent.getStringExtra(ObdReaderService.INTENT_OBD_EXTRA_DATA);
            txtHiddenCurrentMileage.setText(connectionStatusMsg);

            if (connectionStatusMsg.equals("OBD Connected")) {
                //OBD connected  do what want after OBD connection
            } else if (connectionStatusMsg.equals("Connect Lost")) {
                //OBD disconnected  do what want after OBD disconnection
            } else {
                // here you could check OBD connection and pairing status
            }

        } else if (action.equals(ACTION_READ_OBD_REAL_TIME_DATA)) {

            TripRecord tripRecord = TripRecord.getTripRecode(com.codeflexers.plasgo.PC.PassengerCurrentTrip.this);

            txtHiddenCurrentMileage.setText(tripRecord.getCustomFuel());
        }

};
```



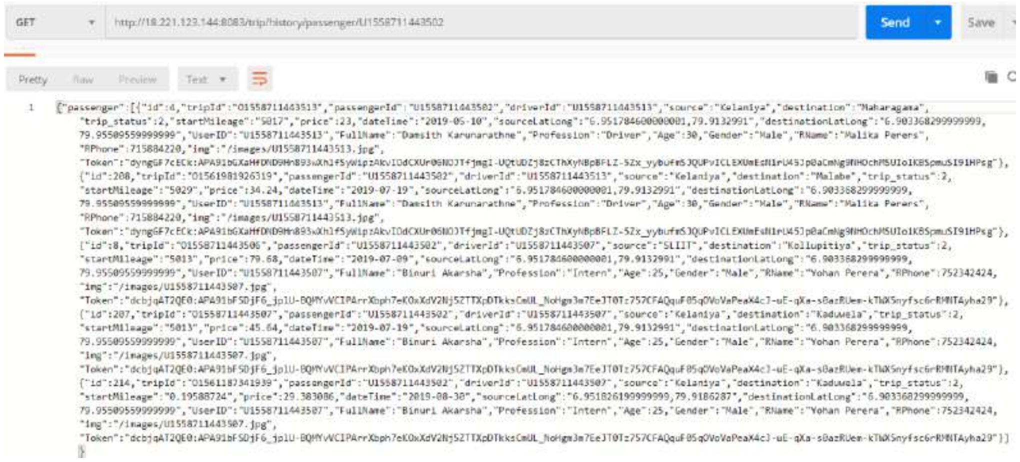
```
public void findIdleAndDrivingFuelConsumtion(float currentMaf) {

  float literPerSecond = 0;
  if (speed > 0) {
    mDrivingMaf += currentMaf;
    mDrivingMafCount++;
    literPerSecond = ((((mDrivingMaf / mDrivingMafCount) / mFuelTypeValue) / gramToLitre));
    mDrivingFuelConsumption = (literPerSecond * (drivingDuration / 1000));

  } else {
    mIdleMaf += currentMaf;
    mIdleMafCount++;
    literPerSecond = ((((mIdleMaf / mIdleMafCount) / mFuelTypeValue) / gramToLitre));
    mIdlingFuelConsumption = (literPerSecond * (idlingDuration / 1000));
  }
}


public float getmDrivingFuelConsumption() {
  return (mIsMAFSupported || mIsTempPressureSupported) ? mDrivingFuelConsumption : MINUS_ONE;
}

public float getmIdlingFuelConsumption() {
  return (mIsMAFSupported || mIsTempPressureSupported) ? mIdlingFuelConsumption : MINUS_ONE;
}
```

Testing and Results



Vehicle	Mileage	Instant Fuel Consumption(L/100km)	Idling Fuel consumption	Driving fuel consumption
Alto LXI	✗	✗	✗	✗
Toyota VIOS	✓	✓	✓	✓
Suzuki Wagon-R	not supported for few vehicles	Supported some of vehicle value not acceptable	✓	✓
Toyota Allion	✓	✓	✓	✓
Toyota Vitz	Supported some of vehicles not supported	✓	✓	✓
SsangYong Kyron	✓	✓	✓	✓

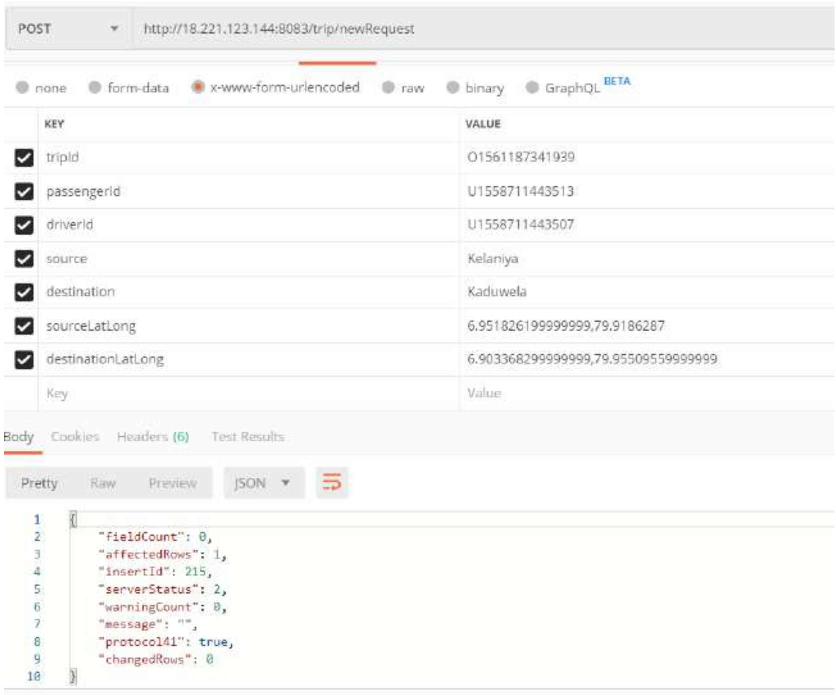
Testing and Results



http://18.221.123.144:8083/trip/history/passenger/U1558711443502

```
{"passenger":[{"id":4,"tripId":"O1558711443513","passengerId":"U1558711443502","driverId":"U1558711443513","source":"Kelaniya","destination":"Maharagama",
    "trip_status":2,"startMileage":"5017","price":23,"dateTime":"2019-05-10","sourceLatLong":"6.951784600000001,79.9132991","destinationLatLong":"6.903368299999999,
    79.95509559999999","UserID":"U1558711443513","FullName":"Damsith Karunarathne","Profession":"Driver","Age":30,"Gender":"Male","RName":"Malika Perers",
    "RPhone":715884220,"img":"/images/U1558711443513.jpg",
    "Token":"dyngGF7cECk:APA91bGXaHfDND9Hn893wXhlf5yWipzAkvIOdCXUr06NOJTfjmgI-UQtUDZj8zCThXyNBpBFLZ-5Zx_yybufm5JQUPvICLEXUmEsN1rU45Jp0aCmNg9NHOchM5UIo1KBSpmuSI91HPsg"},
    {"id":208,"tripId":"O1561981926319","passengerId":"U1558711443502","driverId":"U1558711443513","source":"Kelaniya","destination":"Malabe","trip_status":2,
    "startMileage":"5029","price":34.24,"dateTime":"2019-07-19","sourceLatLong":"6.951784600000001,79.9132991","destinationLatLong":"6.903368299999999,
    79.95509559999999","UserID":"U1558711443513","FullName":"Damsith Karunarathne","Profession":"Driver","Age":30,"Gender":"Male","RName":"Malika Perers",
    "RPhone":715884220,"img":"/images/U1558711443513.jpg",
    "Token":"dyngGF7cECk:APA91bGXaHfDND9Hn893wXhlf5yWipzAkvIOdCXUr06NOJTfjmgI-UQtUDZj8zCThXyNBpBFLZ-5Zx_yybufm5JQUPvICLEXUmEsN1rU45Jp0aCmNg9NHOchM5UIo1KBSpmuSI91HPsg"},
    {"id":8,"tripId":"O1558711443506","passengerId":"U1558711443502","driverId":"U1558711443507","source":"SLIIT","destination":"Kollupitiya","trip_status":2,
    "startMileage":"5013","price":79.68,"dateTime":"2019-07-09","sourceLatLong":"6.951784600000001,79.9132991","destinationLatLong":"6.903368299999999,
    79.95509559999999","UserID":"U1558711443507","FullName":"Binuri Akarsha","Profession":"Intern","Age":25,"Gender":"Male","RName":"Yohan Perera","RPhone":752342424,
    "img":"/images/U1558711443507.jpg",
    "Token":"dcbjqAT2QE0:APA91bFSDjF6_jp1U-BQMYvWCIPArrXbph7eKOxXdV2Nj5ZTTXpDTkksCmUL_NoHgm3m7EeJT0T:757CFAQquF05qOVoVaPeaX4cJ-uE-qXa-s0azRUem-kTWX5nyfsc6rRMNTAyha29"},
    {"id":207,"tripId":"O1558711443507","passengerId":"U1558711443502","driverId":"U1558711443507","source":"Kelaniya","destination":"Kaduwela","trip_status":2,
    "startMileage":"5013","price":45.64,"dateTime":"2019-07-19","sourceLatLong":"6.951784600000001,79.9132991","destinationLatLong":"6.903368299999999,
    79.95509559999999","UserID":"U1558711443507","FullName":"Binuri Akarsha","Profession":"Intern","Age":25,"Gender":"Male","RName":"Yohan Perera","RPhone":752342424,
    "img":"/images/U1558711443507.jpg",
    "Token":"dcbjqAT2QE0:APA91bFSDjF6_jp1U-BQMYvWCIPArrXbph7eKOxXdV2Nj5ZTTXpDTkksCmUL_NoHgm3m7EeJT0T:757CFAQquF05qOVoVaPeaX4cJ-uE-qXa-s0azRUem-kTWX5nyfsc6rRMNTAyha29"},
    {"id":214,"tripId":"O1561187341939","passengerId":"U1558711443502","driverId":"U1558711443507","source":"Kelaniya","destination":"Kaduwela","trip_status":2,
    "startMileage":"0.19588724","price":29.383086,"dateTime":"2019-08-30","sourceLatLong":"6.951826199999999,79.9186287","destinationLatLong":"6.903368299999999,
    79.95509559999999","UserID":"U1558711443507","FullName":"Binuri Akarsha","Profession":"Intern","Age":25,"Gender":"Male","RName":"Yohan Perera","RPhone":752342424,
    "img":"/images/U1558711443507.jpg",
    "Token":"dcbjqAT2QE0:APA91bFSDjF6_jp1U-BQMYvWCIPArrXbph7eKOxXdV2Nj5ZTTXpDTkksCmUL_NoHgm3m7EeJT0T:757CFAQquF05qOVoVaPeaX4cJ-uE-qXa-s0azRUem-kTWX5nyfsc6rRMNTAyha29"}]
    }
```

Backend Get Response



http://18.221.123.144:8083/trip/newRequest

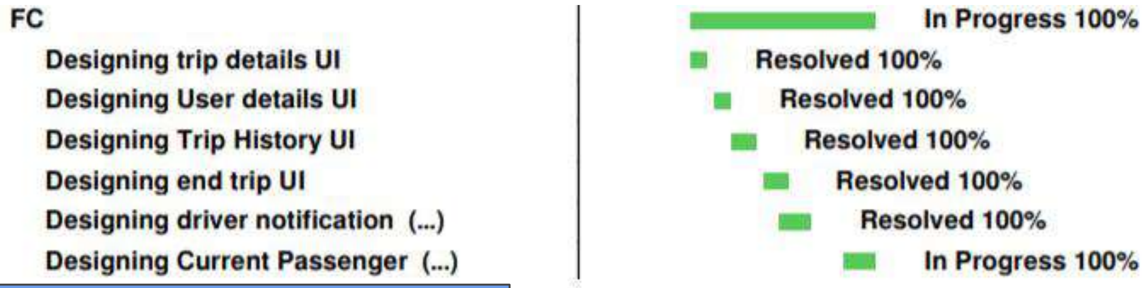
	KEY	VALUE
☑	tripId	O1561187341939
☑	passengerId	U1558711443513
☑	driverId	U1558711443507
☑	source	Kelaniya
☑	destination	Kaduwela
☑	sourceLatLong	6.951826199999999,79.9186287
☑	destinationLatLong	6.903368299999999,79.95509559999999
	Key	Value

```
{
    "fieldCount": 0,
    "affectedRows": 1,
    "insertId": 215,
    "serverStatus": 2,
    "warningCount": 0,
    "message": "",
    "protocol41": true,
    "changedRows": 0
}
```

Backend POST Request and Response

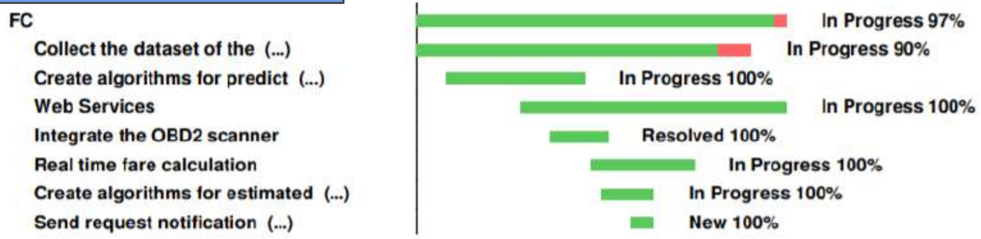
Work Progress

Front End Development



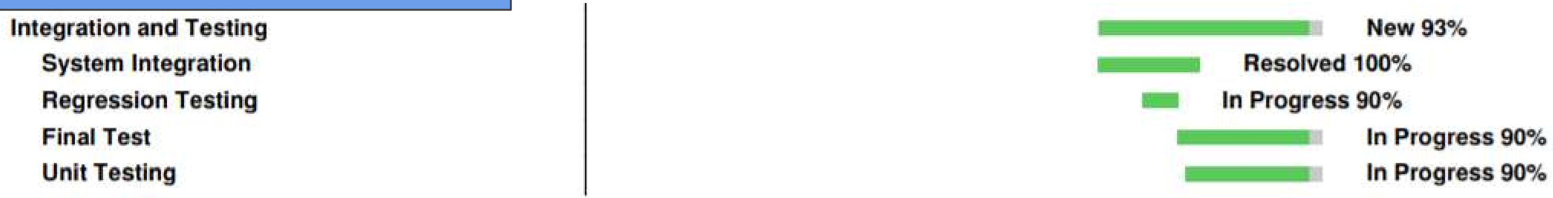
FC	In Progress 100%
Designing trip details UI	Resolved 100%
Designing User details UI	Resolved 100%
Designing Trip History UI	Resolved 100%
Designing end trip UI	Resolved 100%
Designing driver notification (...)	Resolved 100%
Designing Current Passenger (...)	In Progress 100%

Back End Development



FC	In Progress 97%
Collect the dataset of the (...)	In Progress 90%
Create algorithms for predict (...)	In Progress 100%
Web Services	In Progress 100%
Integrate the OBD2 scanner	Resolved 100%
Real time fare calculation	In Progress 100%
Create algorithms for estimated (...)	In Progress 100%
Send request notification (...)	New 100%

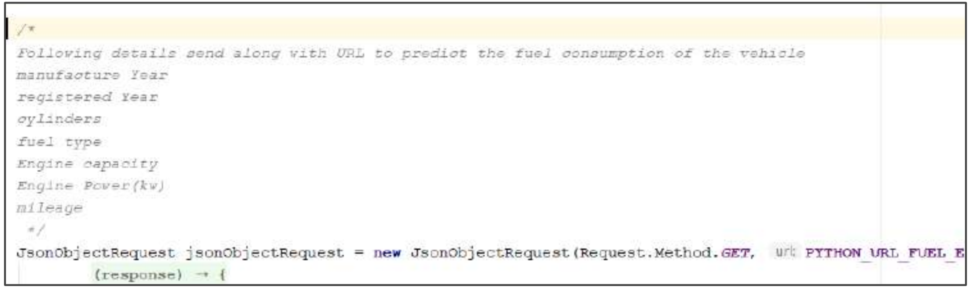
Testing



Integration and Testing	New 93%
System Integration	Resolved 100%
Regression Testing	In Progress 90%
Final Test	In Progress 90%
Unit Testing	In Progress 90%

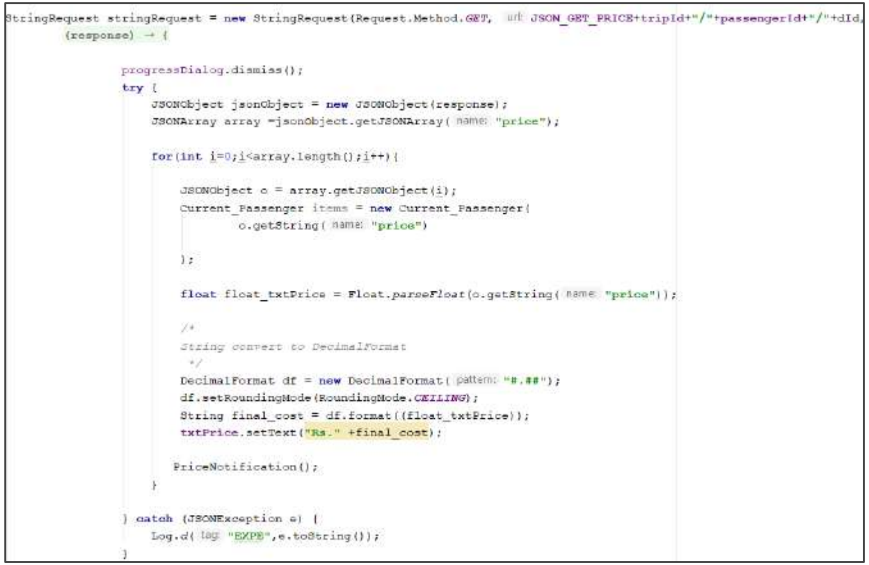
Standards and Best Practices

Clean code and use of Comments



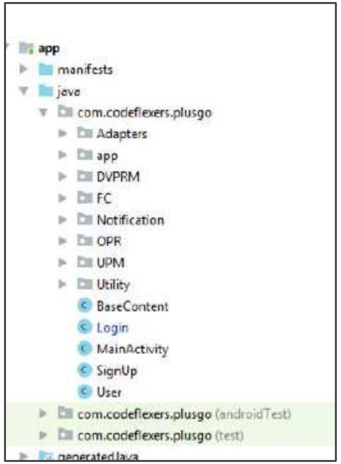
```
/*
Following details send along with URL to predict the fuel consumption of the vehicle
manufacture Year
registered Year
cylinders
fuel type
Engine capacity
Engine Power(kw)
mileage
 */
JsonObjectRequest jsonObjectRequest = new JsonObjectRequest(Request.Method.GET,  url: PYTHON_URL_FUEL_E
        (response) → {
```

Consistent Indentation



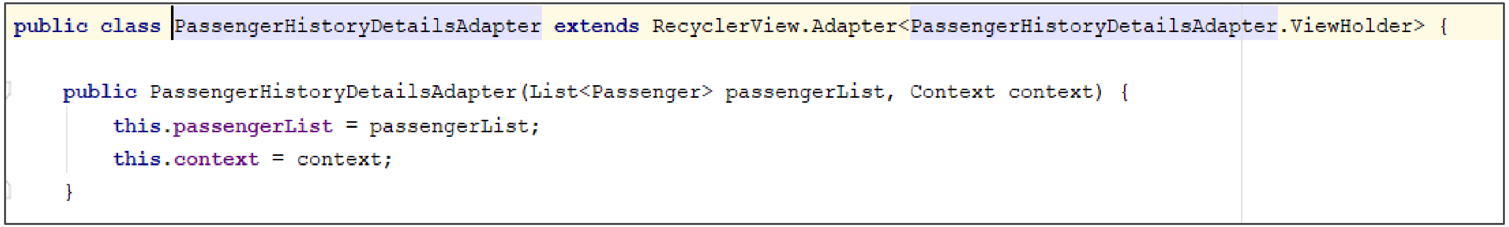
```
StringRequest stringRequest = new StringRequest(Request.Method.GET,  url: JSON_GET_PRICE+tripId+"/"+passengerId+"/"+dId,
        (response) → {

            progressDialog.dismiss();
            try {
                JSONObject jsonObject = new JSONObject(response);
                JSONArray array =jsonObject.getJSONArray( name: "price");

                for(int i=0;i<array.length();i++){

                    JSONObject o = array.getJSONObject(i);
                    Current_Passenger items = new Current_Passenger(
                            o.getString( name: "price")

                    );

                    float float_txtPrice = Float.parseFloat(o.getString( name: "price"));

                    /*
                    String convert to DecimalFormat
                     */
                    DecimalFormat df = new DecimalFormat( pattern: "#.##");
                    df.setRoundingMode(RoundingMode.CEILING);
                    String final_cost = df.format((float_txtPrice));
                    txtPrice.setText("Rs." +final_cost);

                  PriceNotification();
                }

            } catch (JSONException e) {
                Log.d( tag: "EXPE",e.toString());
            }
```

File and Folder Organization



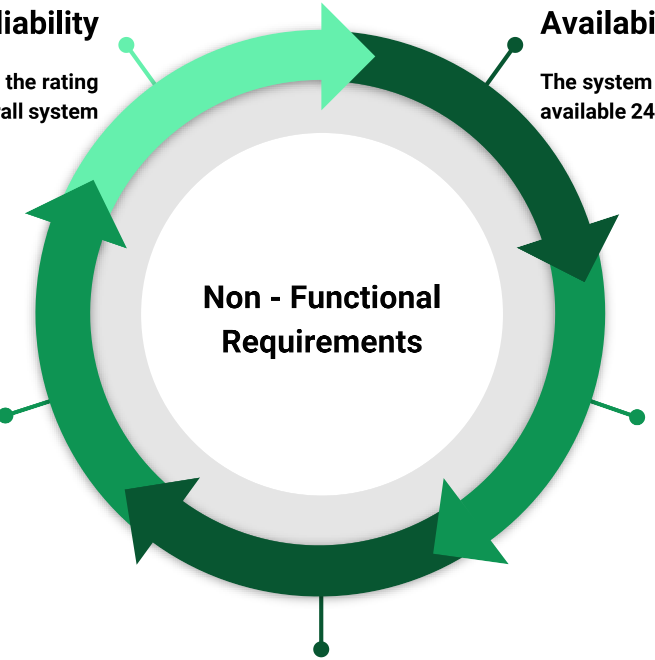
```
app
  manifests
  java
    com.codeflexers.plusgo
      Adapters
      app
      DVPRM
      FC
      Notification
      OPR
      UPM
      Utility
      BaseContent
      Login
      MainActivity
      SignUp
      User
    com.codeflexers.plusgo (androidTest)
    com.codeflexers.plusgo (test)
  generatedJava
```

Use of Object Oriented Concepts



```
public class PassengerHistoryDetailsAdapter extends RecyclerView.Adapter<PassengerHistoryDetailsAdapter.ViewHolder> {

    public PassengerHistoryDetailsAdapter(List<Passenger> passengerList, Context context) {
        this.passengerList = passengerList;
        this.context = context;
    }
```

Non Functional Requirements



Reliability

The document validation and the rating maintenance ensure reliability of overall system

Availability

The system is designed to be available 24x7

Usability

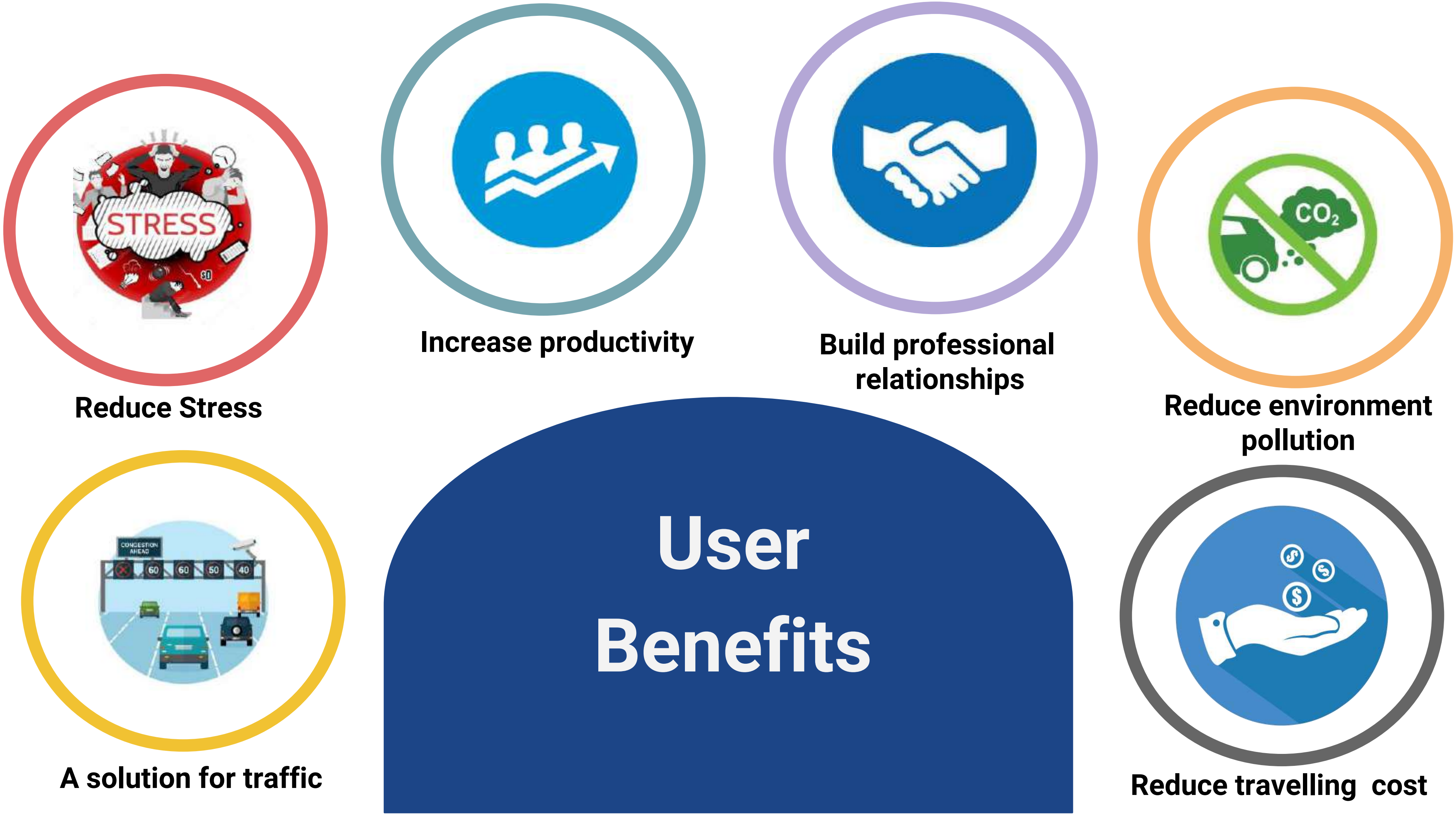
User interfaces are designed in simple and easy to use manner

Security

Confidential user details are protected and unwanted sensitive user data are automatically removed from system

Maintainability

The algorithms are developed to reduce high coupling



User Benefits

Reduce Stress

Increase productivity

Build professional relationships

Reduce environment pollution

A solution for traffic

Reduce travelling cost

Business Model

ABSOLUTELY FREE

Any user can download it for **Free**

REVENUE ON SALES

10% Charged on each User Ride

SUBSCRIPTION FOR VALUE ADDED SERVICES

Monthly Subscription on Selected services