Public Transportation in Sri Lanka

Plus Go

Intelligent Complementary Ride-Sharing System

Mechanism of Suggesting Most Suitable Drivers

Driver

Passenger

CEO

Lawyer

Engineer

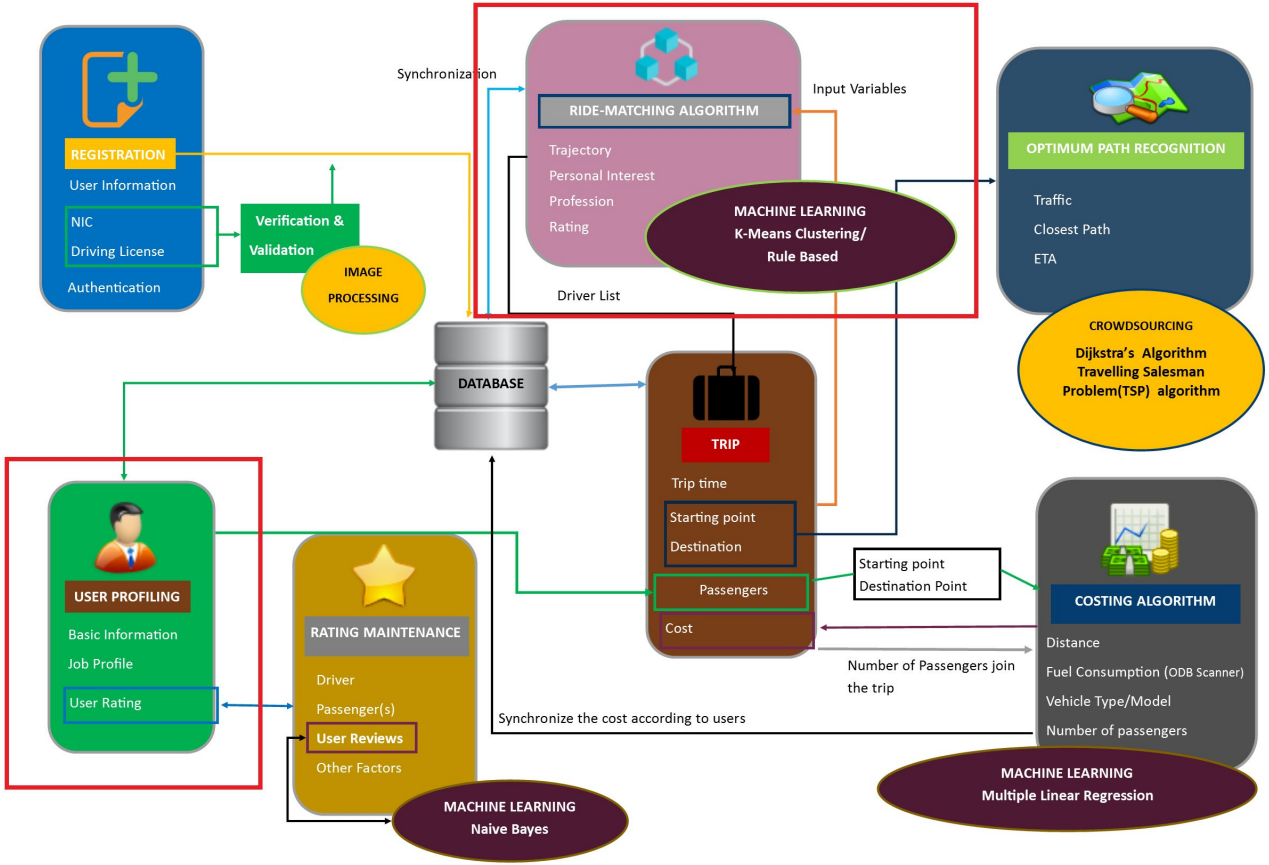
Security Officer

Doctor

User Profile Management

IT16030190

High Level Diagram



Objectives

- Suggest most suitable drivers for a passenger based on the Profession and Preferences of the passenger.

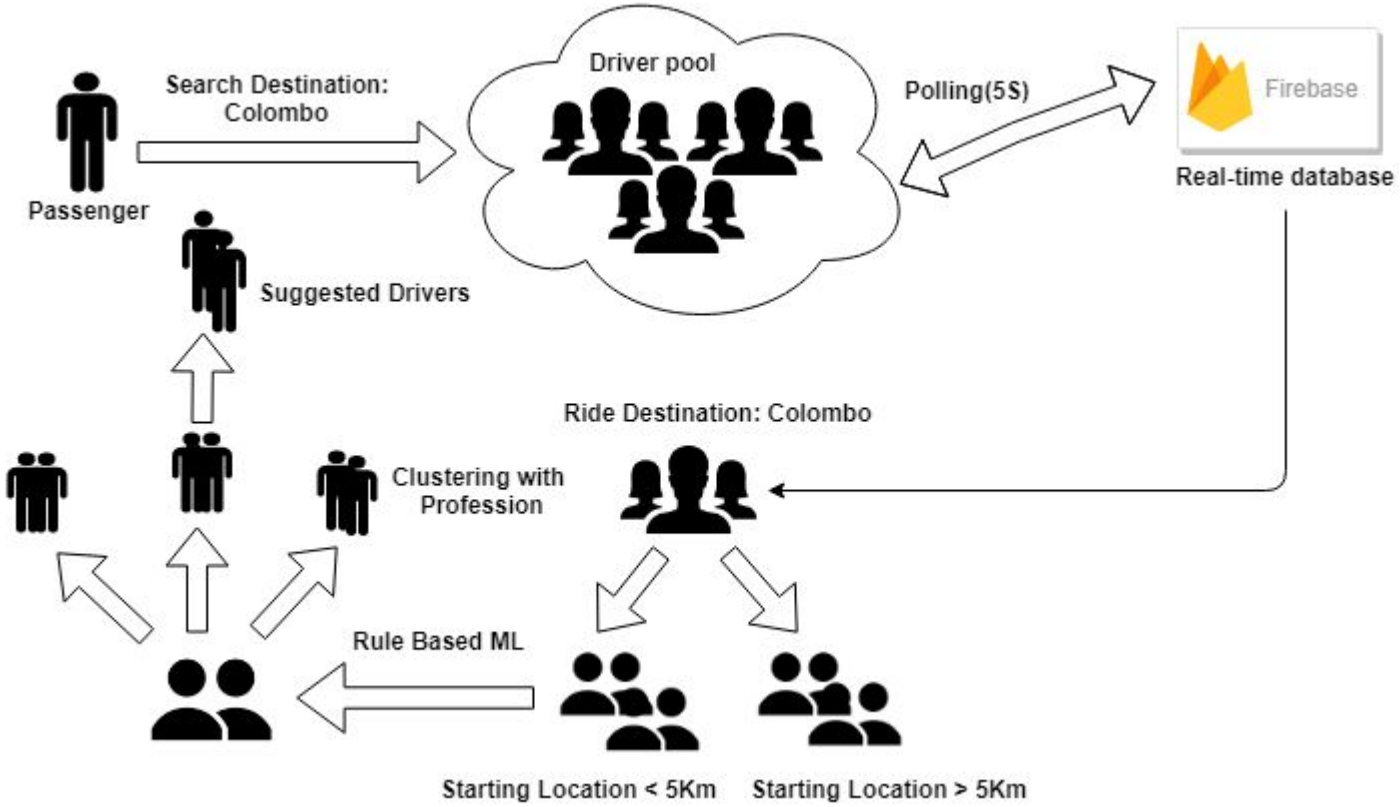
  Preferences Include:
  - Smoking Condition
  - Music Lover
  - Language Preference
  - Motion Sickness
  - Gender Preference
  - Like Quietness

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

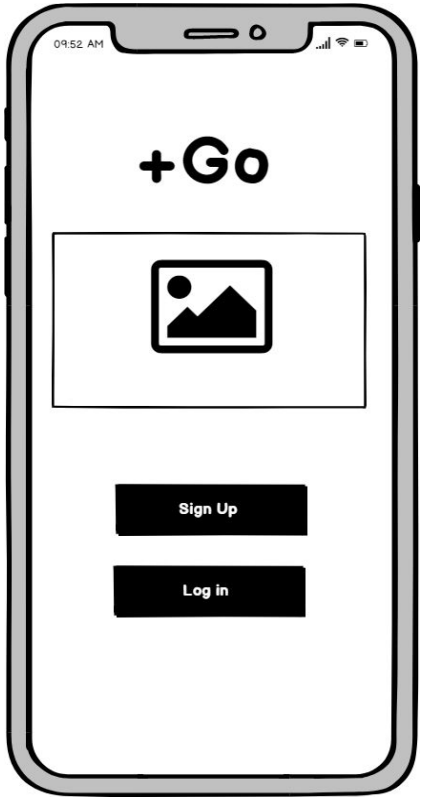
Research Gap

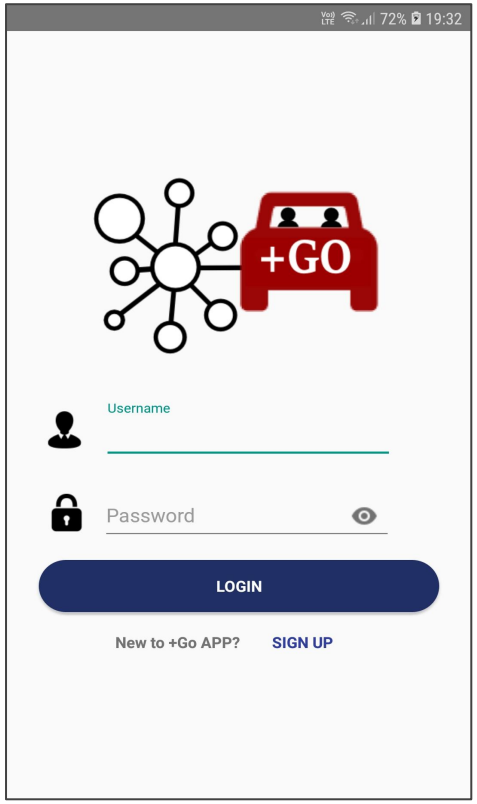
Features	UBER	UDIO	Carpooling.lk	RideShare.lk	+GO
Focused only on professionals	✗	✗	✗	✗	✓
Matching the passengers' profile with drivers	✗	✗	✗	✗	✓
Allow the spouse/guardian to check the passenger's trip details	✗	✗	✗	✗	✓
Focus on Gender Preference to provide more customize suggestions	✗	✓	✗	✗	✓

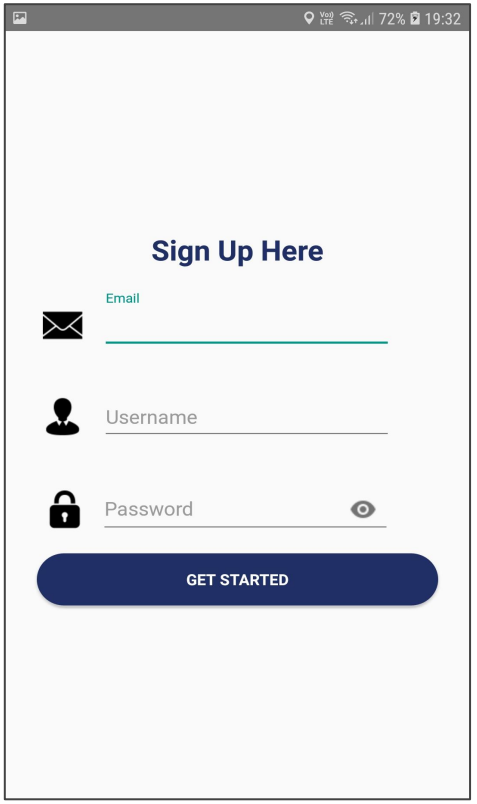
Flow of the User Profile Management



Prototype vs Implementation





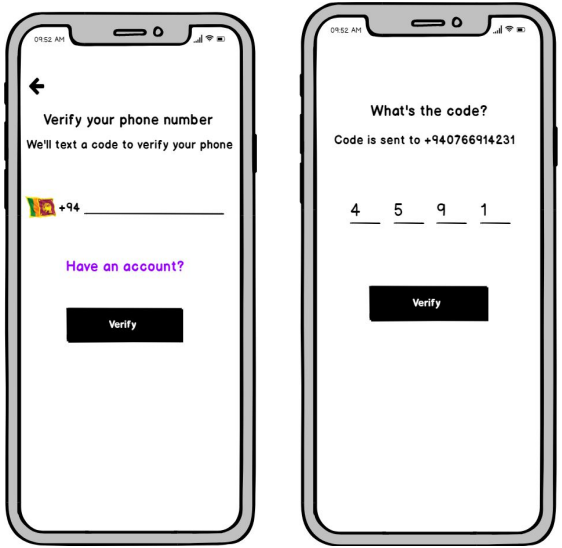


Prototype

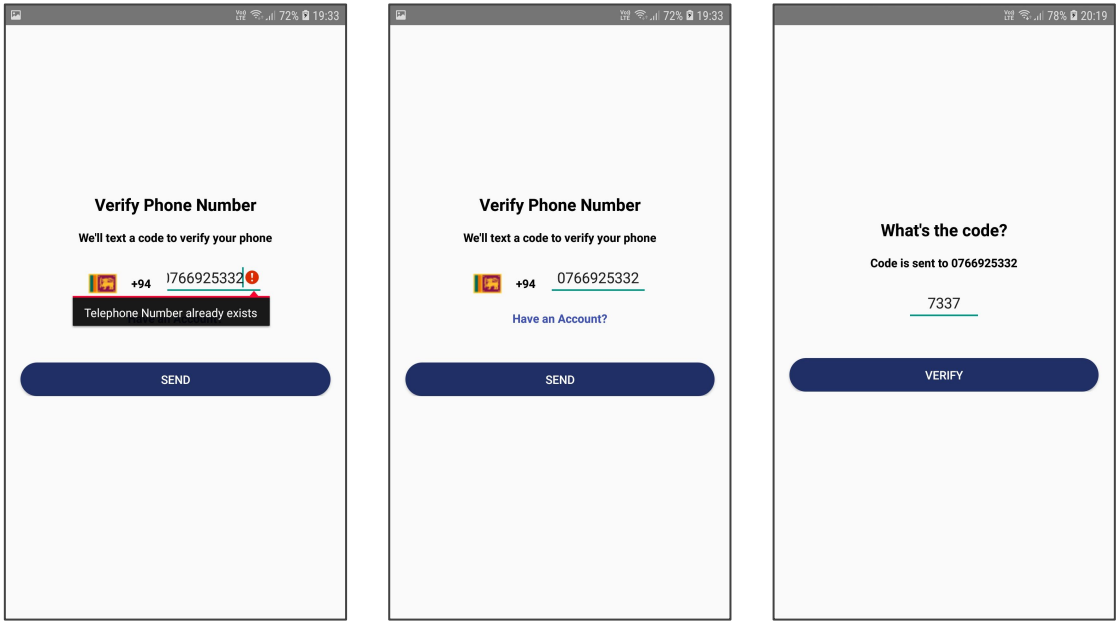
Application

Prototype vs Implementation

Mobile Number Verification



Prototype

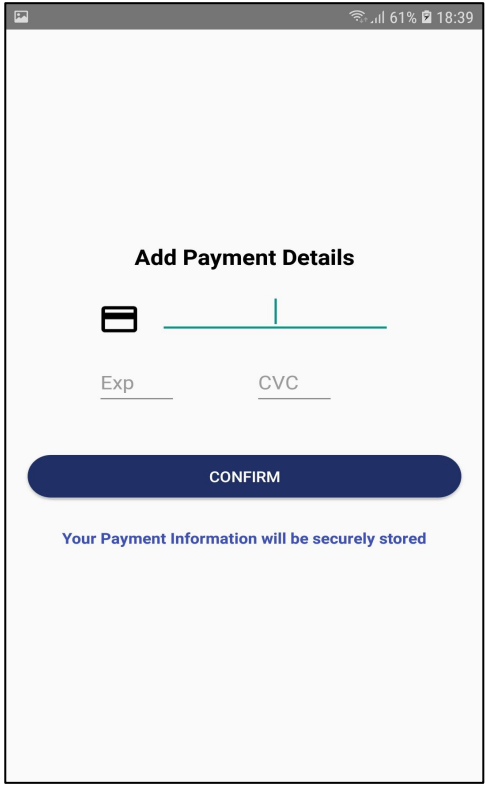


Application

Prototype vs Implementation

Prototype

Adding Payment Method

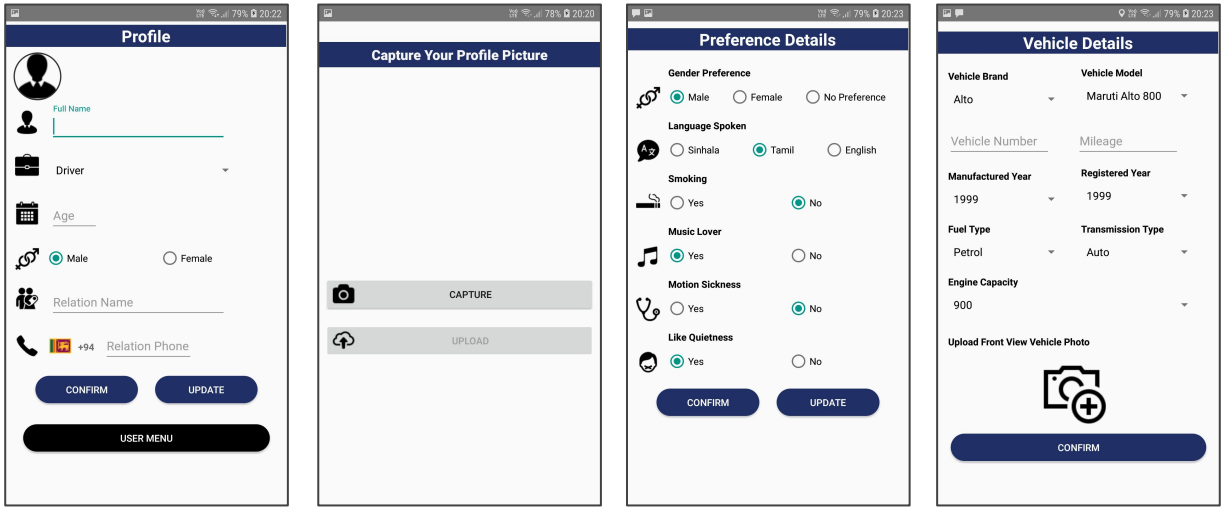


Application

Prototype vs Implementation

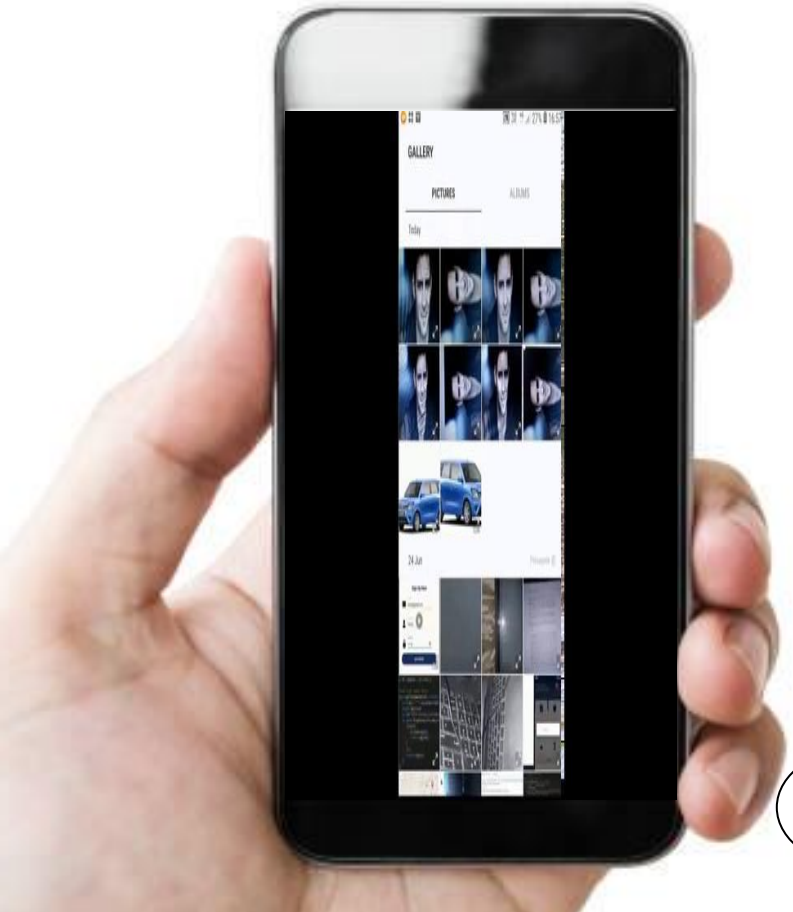
Prototype

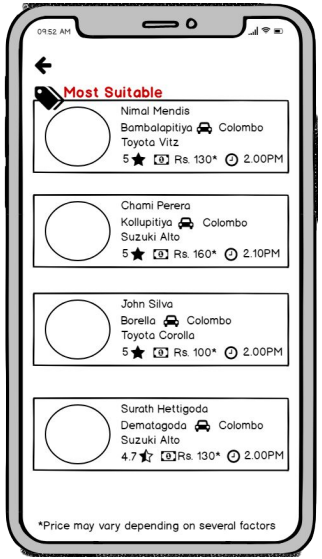
Profile Creation



Application

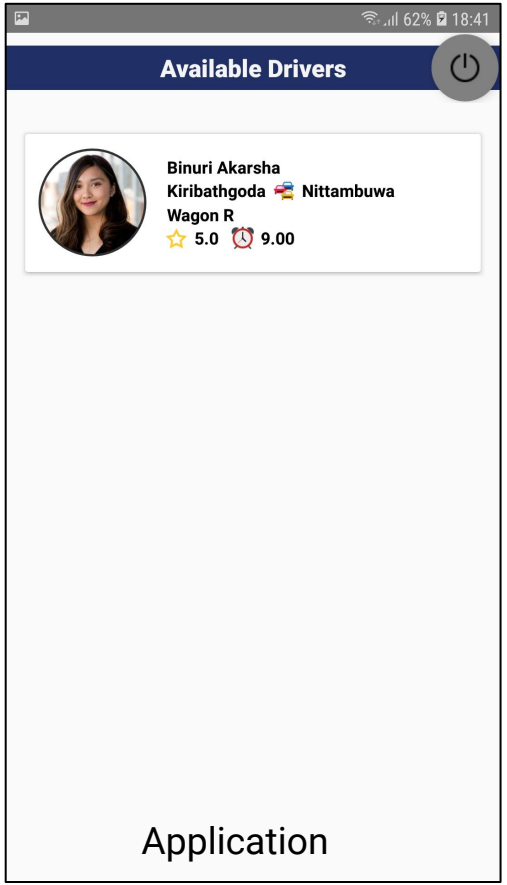
Prototype vs Implementation





Prototype

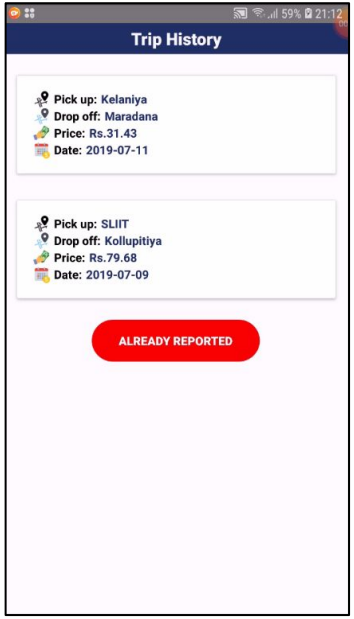
Suggesting Driver List

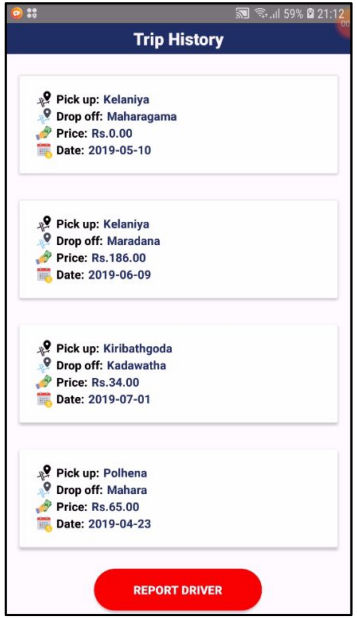


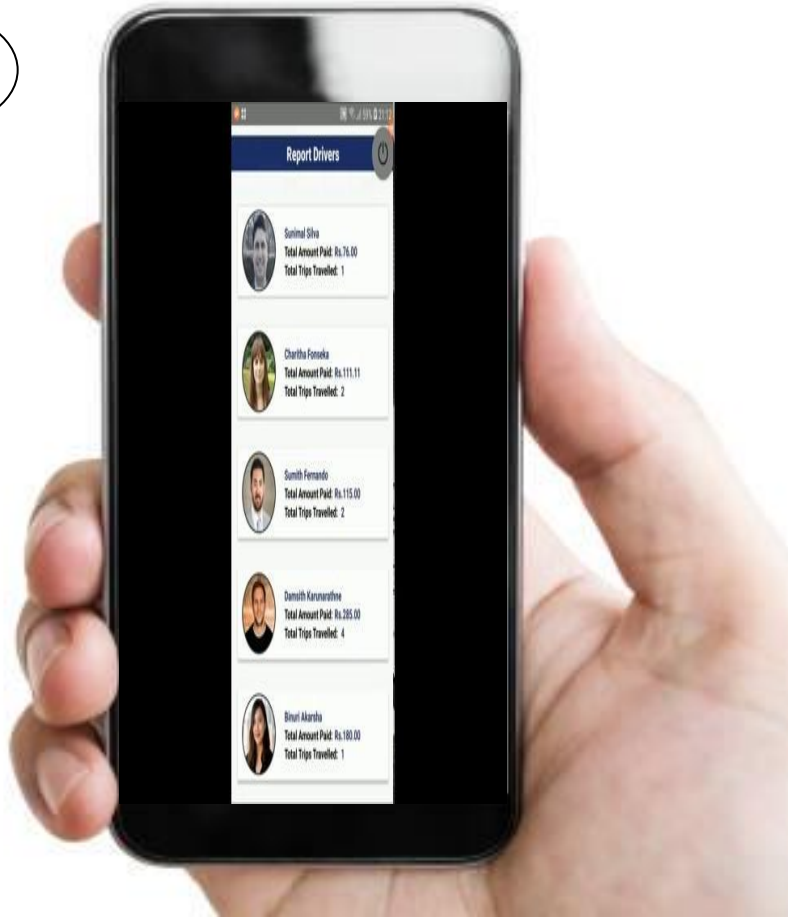
Application

Prototype vs Implementation

Spouse/Guardian Reporting

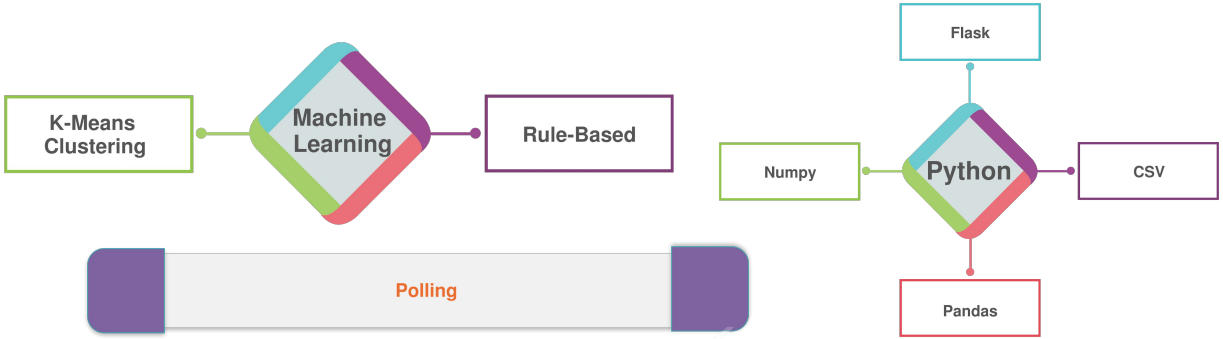




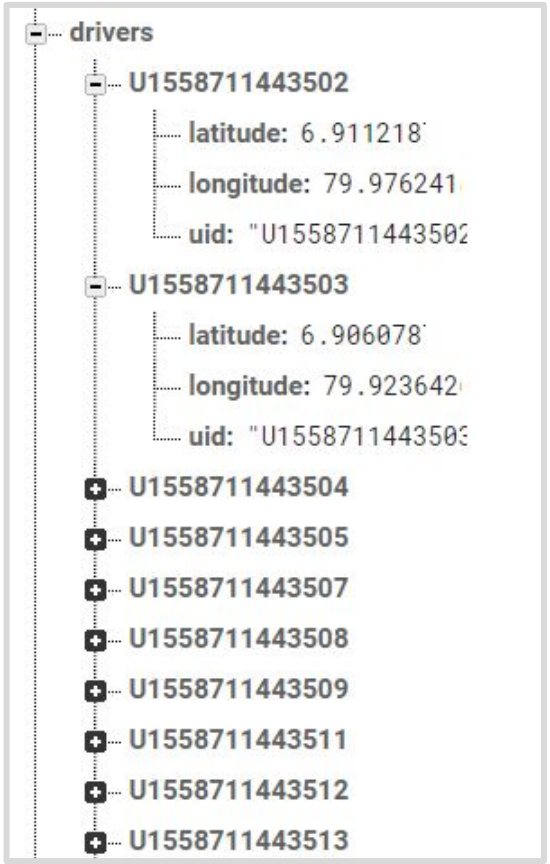


Final Output - Suggesting Driver List

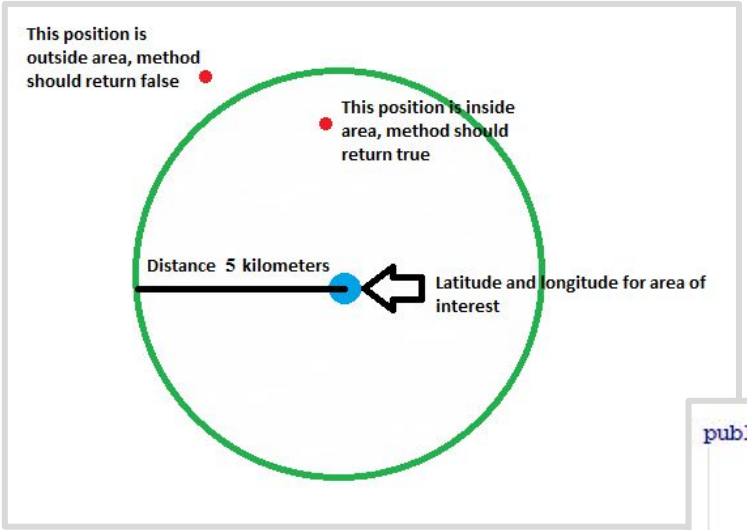
The Specialized Area and the Use of Technologies




```
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( 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);
    }
}
```



The Specialized Area and the Use of Technologies



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

Not secure | 192.168.1.5:8099/ridematching/executeRules/U1558711443502

java | Publisher Panel | Survey Form regard... | Responsive Data Ta... | AdFly - The URL


```
{
  "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)
```

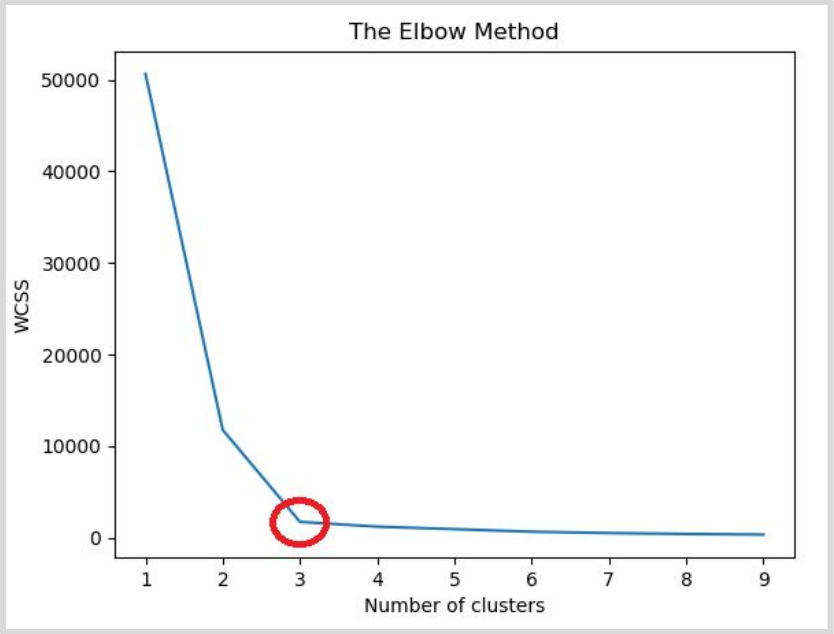
Rule-Based Machine Learning

```
#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



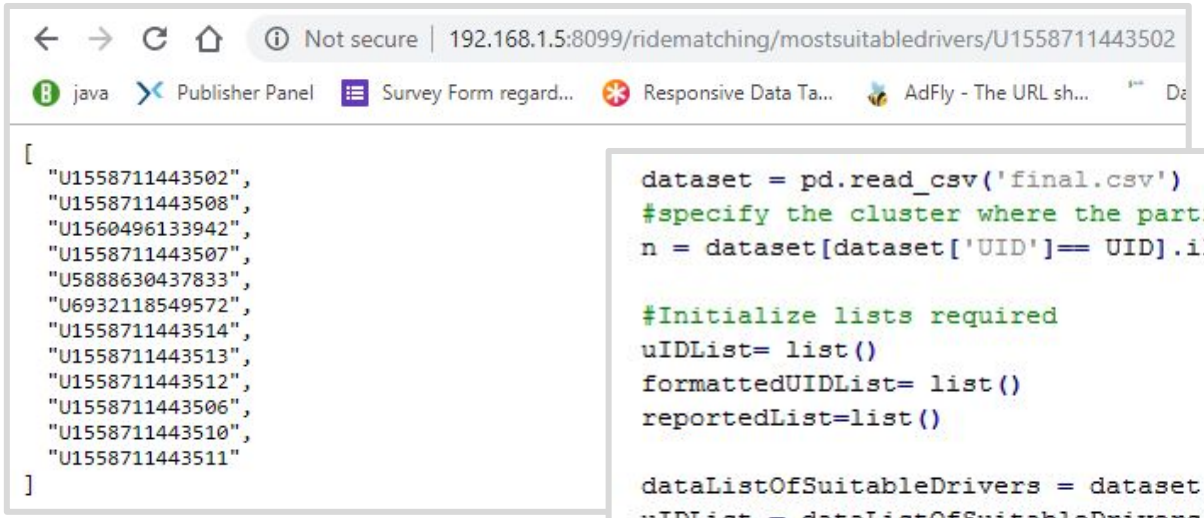
The Specialized Area and the Use of Technologies


```
# 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



Most Suitable Driver List

```
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)
```

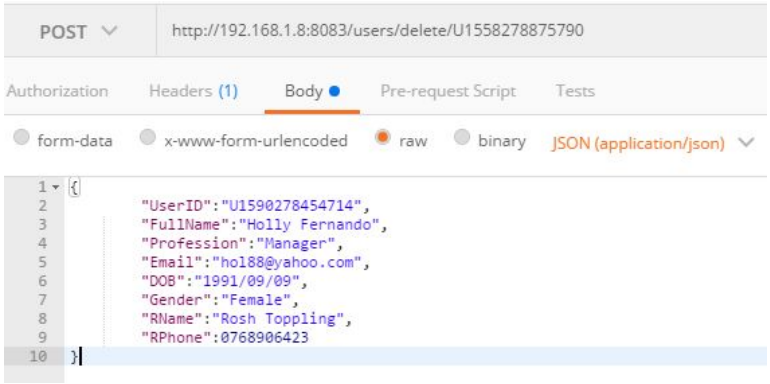
Testing Plan

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

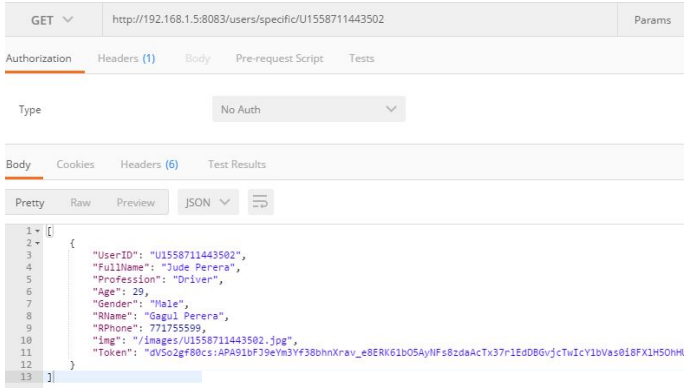
Backend Test Plan

```
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



GET Request Testing



POST Request Testing

Results

Accuracy of Algorithm

```
---Testing For the Accuracy of Ride-Matching Algorithm with Parameter Count---

Total Dataset Records            : 679
Testing UserID                   : U1558711443502
Paramater Count                  : 1
User's Cluster                   : 2

Total No.of Entries in the Cluster: 340
Entries match with U1558711443502 : 232
                    --------------
Accuracy of Exact Matches[%]      : 68
```

```
---Testing For the Accuracy of Ride-Matching Algorithm with Parameter Count---

Total Dataset Records            : 679
Testing UserID                   : U1558711443502
Paramater Count                  : 2
User's Cluster                   : 2

Total No.of Entries in the Cluster: 280
Entries match with U1558711443502 : 199
                                  --------------
Accuracy of Exact Matches[%]      : 71
```

```
---Testing For the Accuracy of Ride-Matching Algorithm with Parameter Count---

Total Dataset Records            : 679
Testing UserID                   : U1558711443502
Paramater Count                  : 6
User's Cluster                   : 2

Total No.of Entries in the Cluster: 120
Entries match with U1558711443502 : 118
                                  --------------
Accuracy of Exact Matches[%]      : 98
```

Results

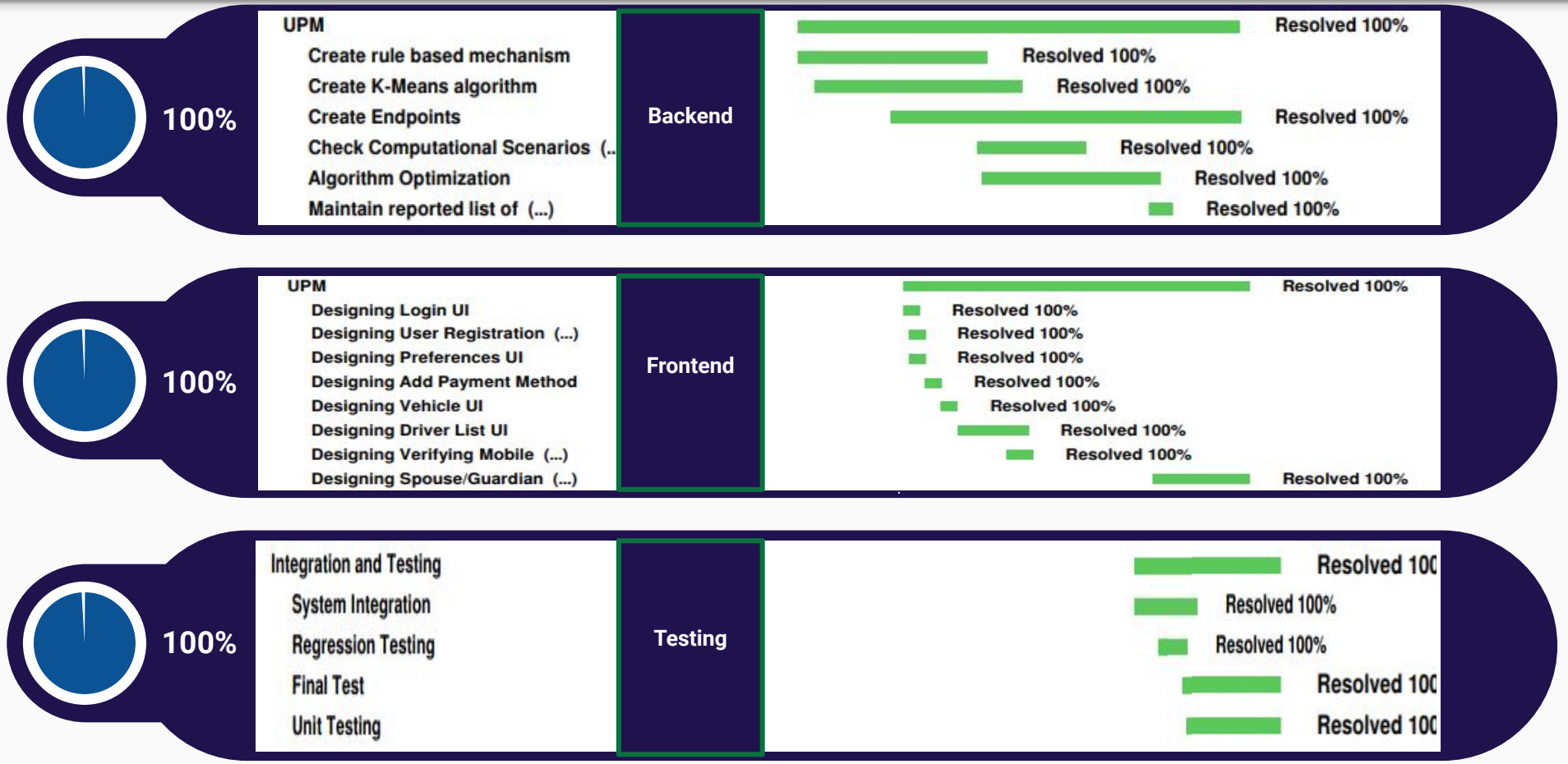
Execution time of Algorithm

```
---Time Taken to execute Ride-Matching Algorithm---

Parameters/Rules Used : 1
Execution Time        : 0.61980453653091
```

```
---Time Taken to execute Ride-Matching Algorithm---

Parameters/Rules Used : 3
Execution Time        : 0.7466500123201
```

```
---Time Taken to execute Ride-Matching Algorithm---

Parameters/Rules Used : 6
Execution Time        : 0.8437609672546387 seconds
```

Work Progress



Standards and Best Practices

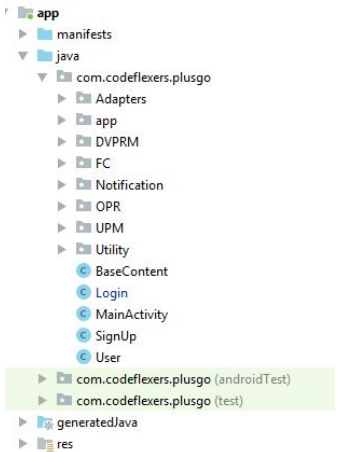
Clean code and use of Comments

```
/**
 * 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



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

IT16025936

Objective

(1) 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

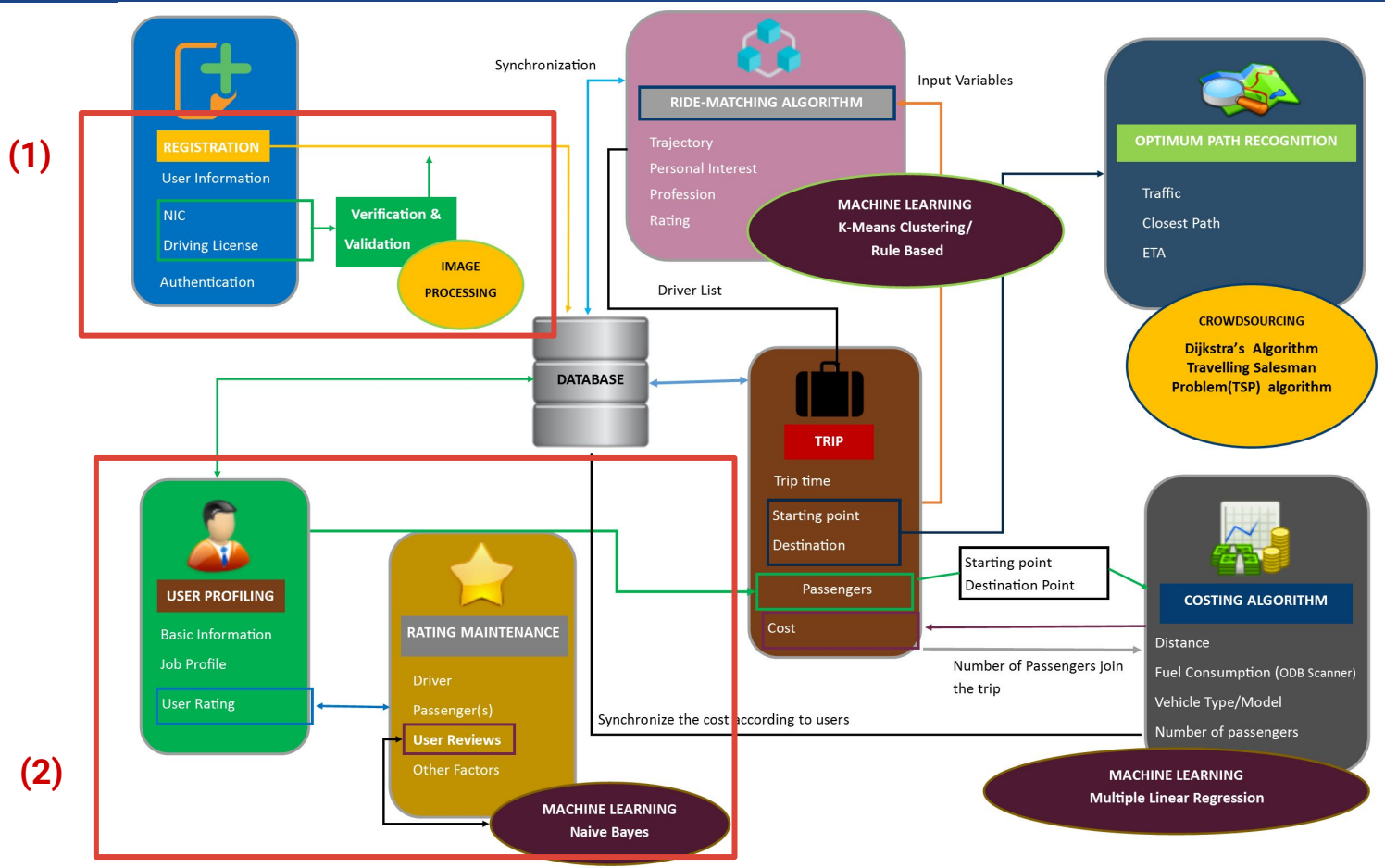
(2) Profile Rating Maintenance

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

High Level Diagram



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	✗	✗	✗	✗	✓

(1) Document Validation

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

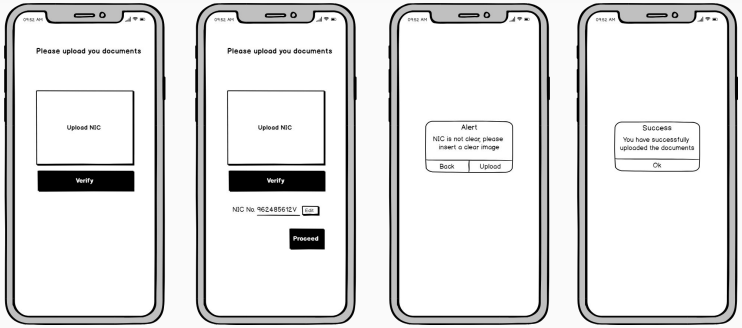
Raw Image → Gray Scale Image → Binary Image

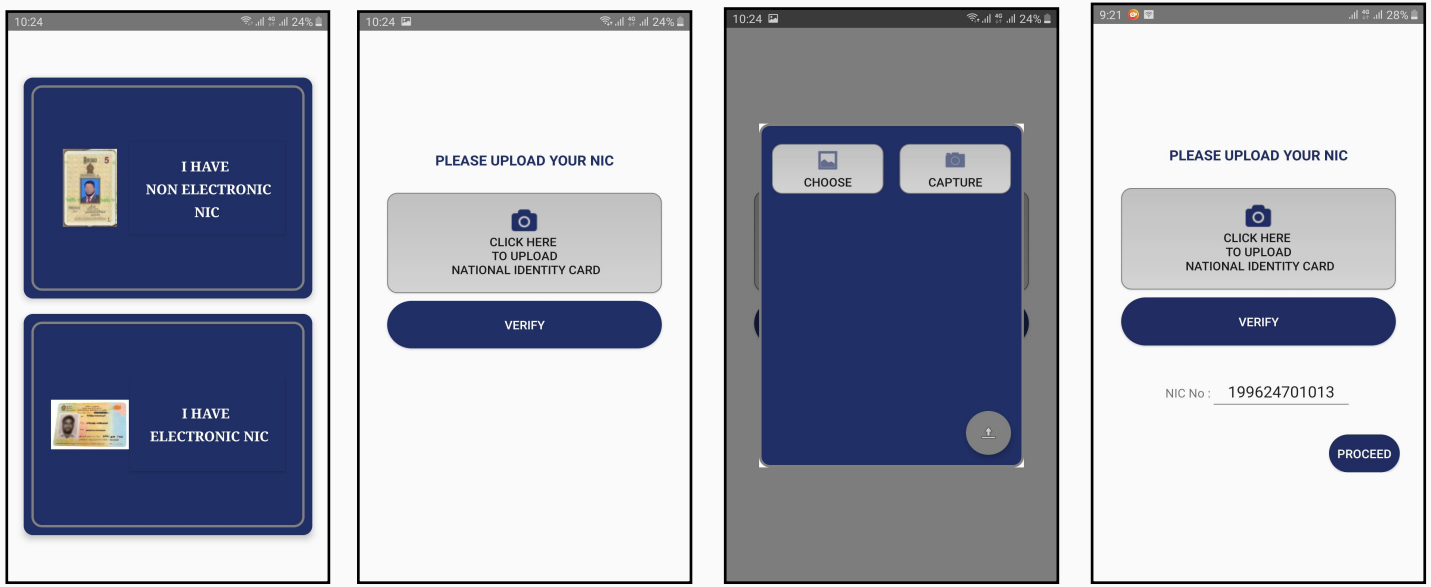
Document Validation - NIC (Image Processing)

Tesseract
Flask
Python
Open Cv
Android
PHP

Prototype vs Implementation

NIC card validation





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

Haar-cascade algorithm

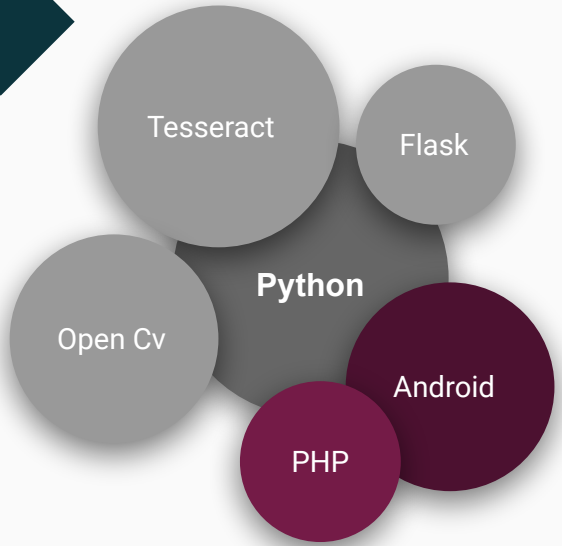
- Sharpening
- Adjust Brightness & Contrast

- 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

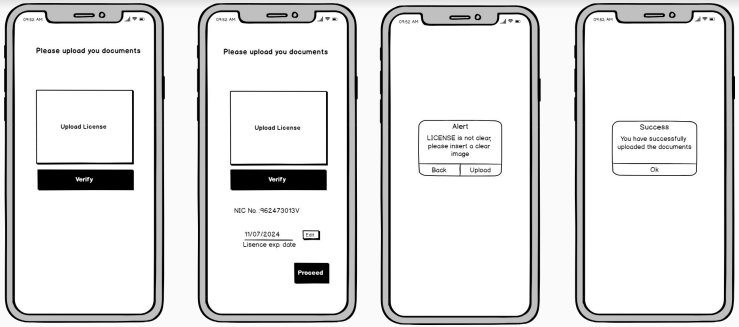
- Raw Image
- Grayscale Image
- Binary Image

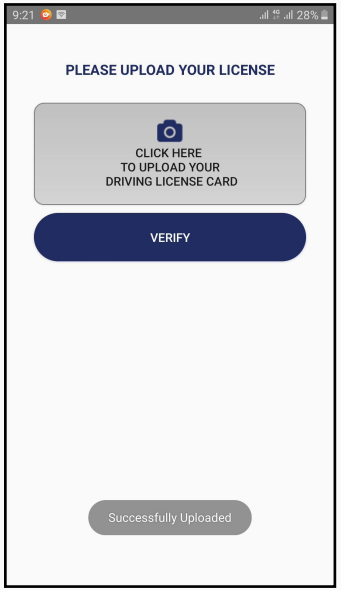
Document Validation - License (Image Processing)

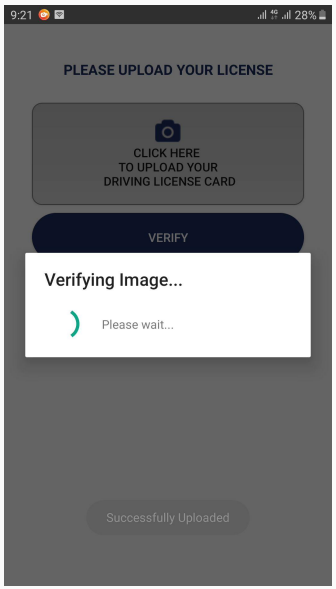


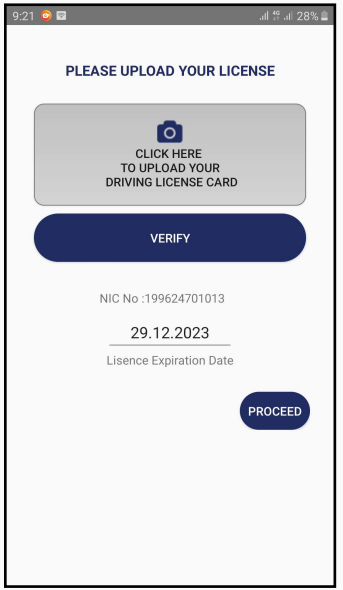
Prototype vs Implementation cont...

License card validation









Final Output

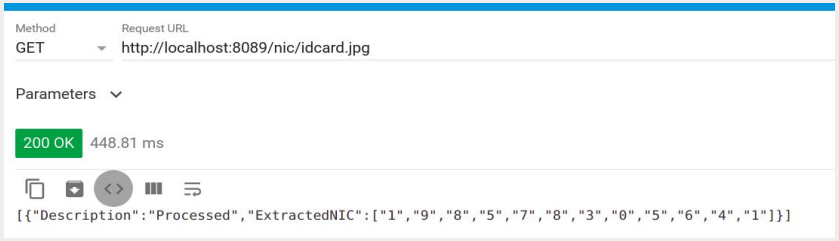
Screen Recording

Actual Scenario

Results

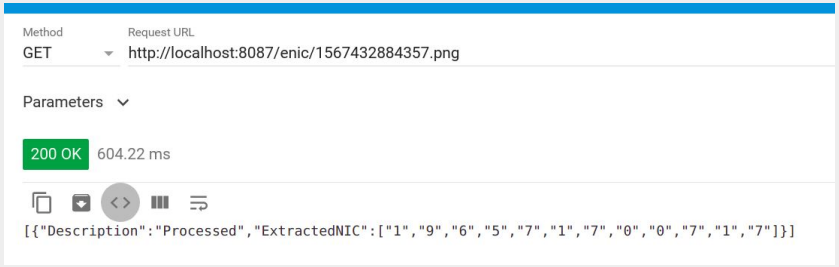
Non Electronic NIC

```
 * 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']
```



Electronic NIC

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



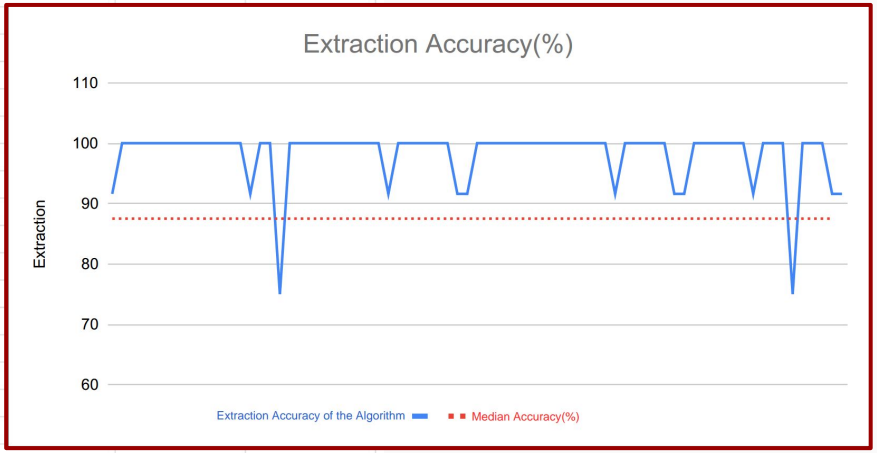
License

```
 * 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 -
```

Testing and Results

	A	B	C	D	E	F	G	H	I
1	Actual NIC number	Expected Output	Actual Output	Extraction Accuracy(%)	Type	Comments		Average Accuracy	Median Accuracy
2	967652792V	199676502792	199976502792	91.6	Non Electronic			98.10133333	87.5
3	987071087V	199870701087	199870701087	100	License				
4	987071087V	199870701087	199870701087	100	License	Size change			
5	987071087V	199870701087	199870701087	100	License	Brighness			
6	987071087V	199870701087	199870701087	100	License	Size & Brightness			
7	987071087V	199870701087	199870701087	100	Non Electronic				
8	970300295V	199703000295	199703000295	100	License				
9	970300295V	199703000295	199703000295	100	License	Size Change			
10	970300295V	199703000295	199703000295	100	License	Size Change			
11	970300295V	199703000295	199703000295	100	License	Size and Brightness			
12	970300295V	199703000295	199703000295	100	Non Electronic				
13	968664166V	199686604166	199686604166	100	License				
14	967912603V	199679102603	199679102603	100	License				
15	967912603V	199679102603	199679102603	100	Non Electronic				
16	967652792V	199676502792	199676502798	91.6	License				
17	965570390V	199655700390	199655700390	100	Non Electronic				
18	965313850V	199653103850	199653103850	100	Non Electronic				
19	942914210V	199429104210	199627106210	75	License				
20	962471013V	199624701013	199624701013	100	Non Electronic				
21	962471013V	199624701013	199624701013	100	License				
22	962230245V	199622300245	199622300245	100	License				
23	962230245V	199622300245	199622300245	100	Non Electronic				
24	961650275V	199616500275	199616500275	100	Non Electronic				
25	960710223V	199607100223	199607100223	100	License	Size			
26	960710223V	199607100223	199607100223	100	License	Brightness			
27	960710223V	199607100223	199607100223	100	License				

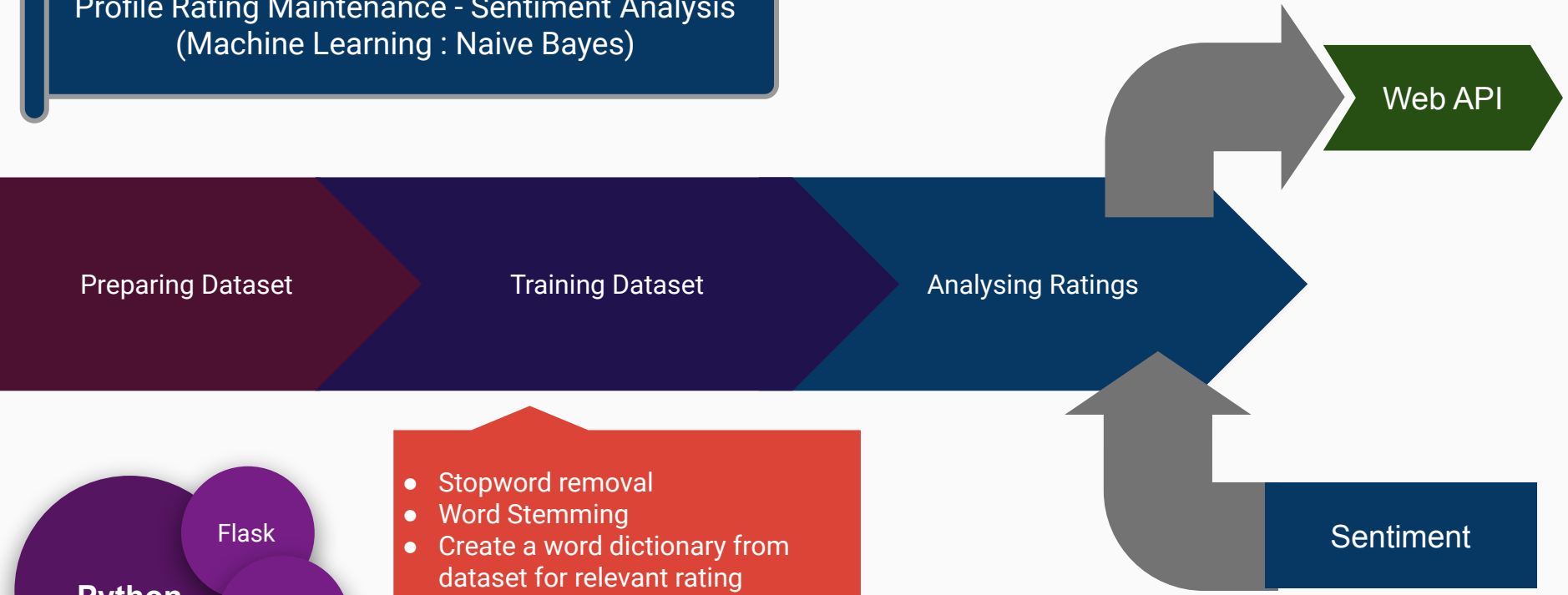
Median accuracy of 87.5%



(2) Profile Rating Maintenance

The Specialized Area and the Use of Technologies

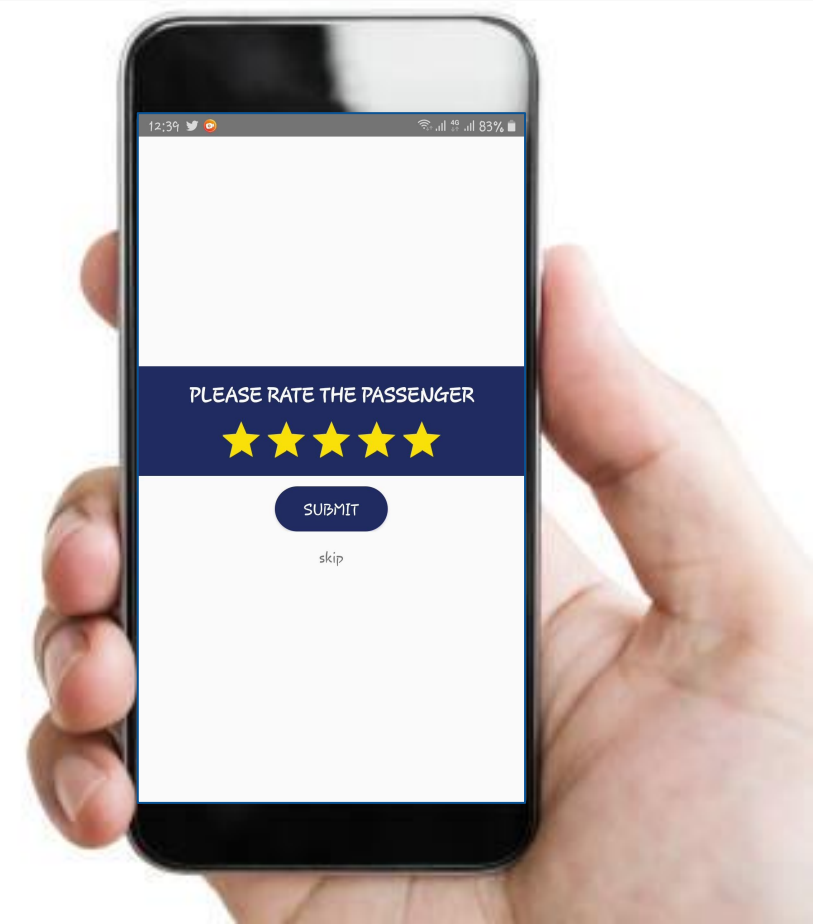
Profile Rating Maintenance - Sentiment Analysis
(Machine Learning : Naive Bayes)



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

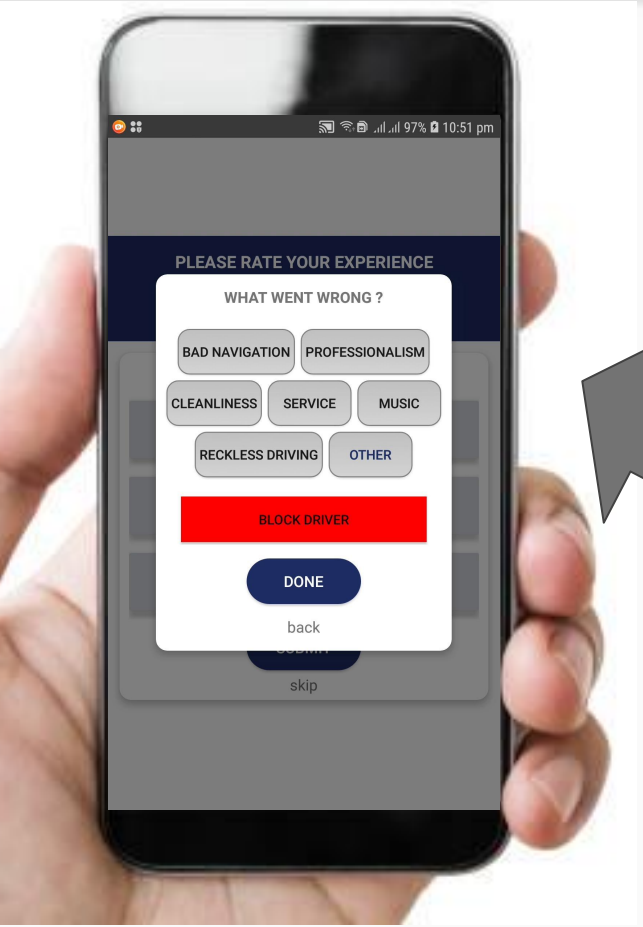
Prototype vs Implementation cont...

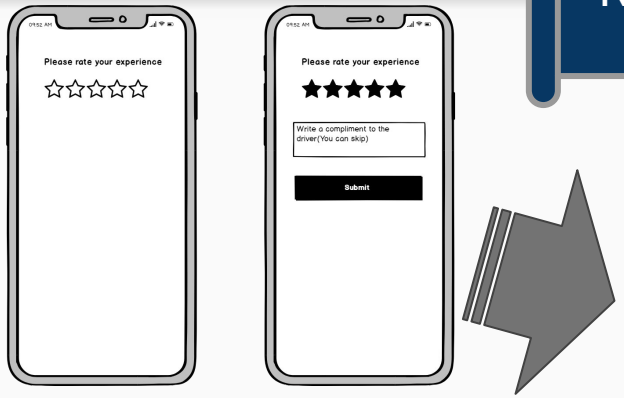
Rating by Driver

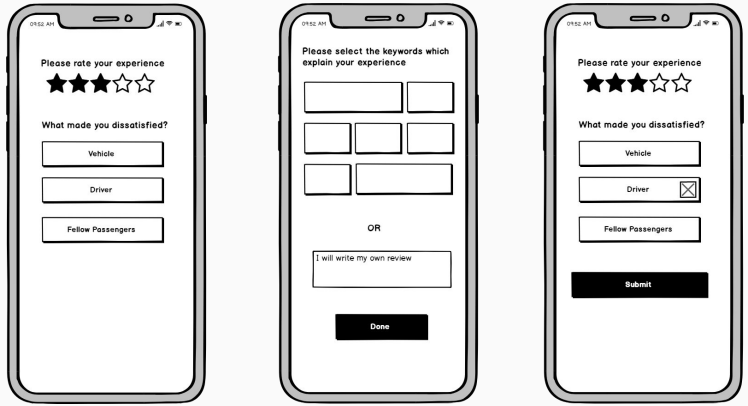


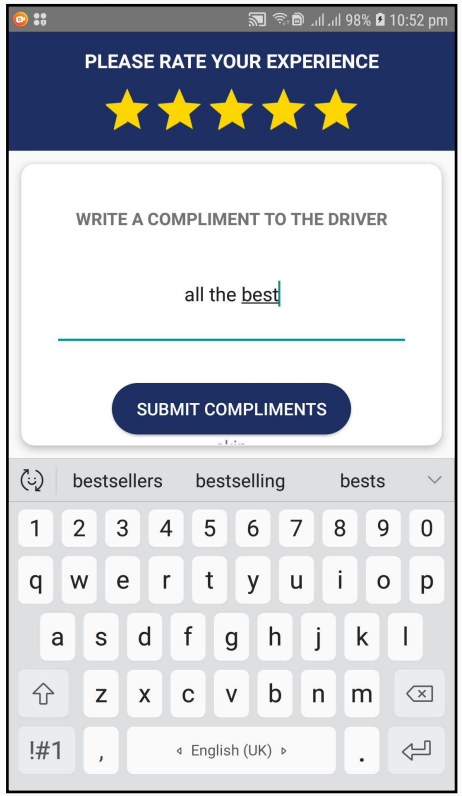
Prototype vs Implementation cont...

Rating by Passenger









Final Output

Driver

Passenger

Results

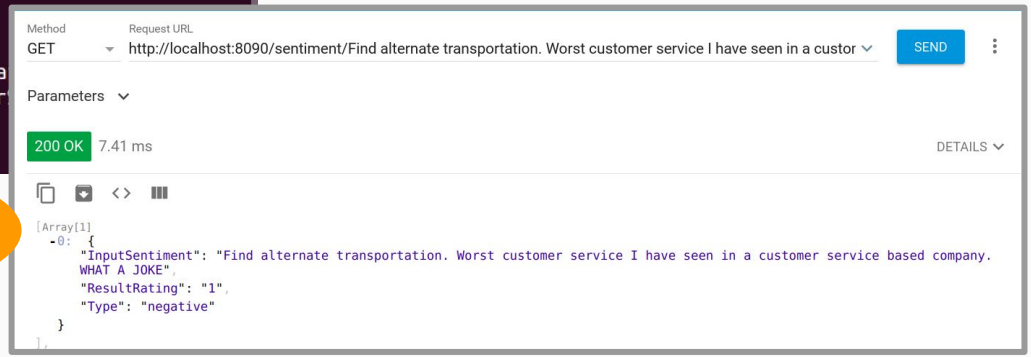
Sentiment Analysis
(Machine Learning : Naive Bayes)


```
great service will definitely use again
{'1': 6.870463190966089e-06, '2': 1.3999915390680115e-06, '3': 2.170232099759973e-06,
'4': 2.647855055164963e-06, '5': 1.9442101536428057e-05}
positive
5
127.0.0.1 - - [11/Nov/2019 11:00:56] "GET /sentiment/great%20service%20will%20definite
ly%20use%20again HTTP/1.1" 200 -
```

Positive Sentiment

```
 * 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%20alterna
on.Worst%20customer%20service%20I%20have%20seen%20in%20a%20customer
20company.What%20A%20JOKE???? HTTP/1.1" 200 -
```

Negative Sentiment



Testing and Results

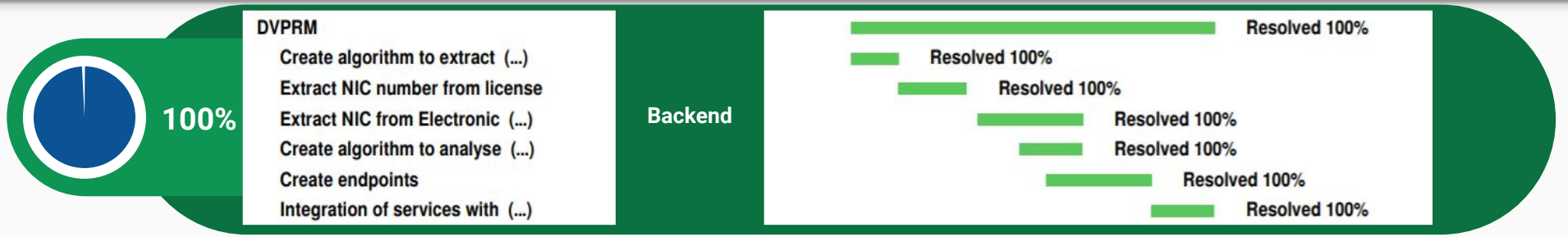
Sentiment Analysis
(Machine Learning : Naive Bayes)

	A	B	C	D	E
1	Test case	Resulting Probability	Result Type		Final Rating
2	1	[(5.962671252544677e-18, '1'), (4.373536229055597e-18, '2'), (1.8748184305513526e-18, '3'), (1.5490402546737354e-21, '4'), (9.376271772881629e-23, '5')]	Negative		1
3	2	[(6.898305572456868e-21, '1'), (1.7083048862944496e-21, '2'), (1.2895089956211236e-21, '3'), (2.0474696515306334e-24, '4'), (2.4052318339944445e-25, '5')]	Negative		1
4	3	[(5.588691552558338e-10, '1'), (5.514826317767379e-10, '2'), (3.071598028085042e-10, '3'), (1.7212893261138856e-10, '4'), (1.5851390565804216e-10, '5')]	Negative		1
5	4	[(3.438405175919485e-26, '1'), (1.4411624851650009e-20, '2'), (1.4415760961443014e-24, '3'), (2.2864910447481222e-25, '4'), (6.323872392725927 4e-27, '5')]	Negative		2
6	5	[(7.512837817358848e-25, '1'), (1.5437014061052798e-20, '2'), (8.953630728247569e-25, '3'), (5.97493096097148e-28, '4'), (1.0218085814933863e-31, '5')]	Negative		2
7	6	[(3.671229960231486e-39, '1'), (1.8140340217915 04e-32, '2'), (9.431955555096896e-37, '3'), (4.6581772513389274e-43, '4'), (8.22529271417799e-46, '5')]	Negative		2
8	7	[(6.25346903453 5338e-31, '1'), (4.022969868801098e-25, '2'), (4.4773071979618 49e-32, '3'), (6.60864398154342e-32, '4'), (8.460769505839564e-36, '5')]	Negative		2
9	8	[(1.727727381767436 9e-103, '1'), (1.5902900175811767e-85, '2'), (1.1781508285233013e-99, '3'), (1.3335023075522012e-112, '4'), (2.0215236672612 18e-122, '5')]	Negative		2
10	9	[(3.168116873646999e-179, '1'), (6.031898445032029e-152, '2'), (1.7663006785155443e-173, '3'), (2.0158305840504515e-189, '4'), (2.2666183905080 05e-214, '5')]	Negative		2
11	10	[(1.2145378422166083e-116, '1'), (2.370980019785004e-94, '2'), (6.028514770850162e-115, '3'), (8.626556342446438e-131, '4'), (1.7324463320369158e-150, '5')]	Negative		2
12	11	[(1.2642211848491124e-116, '1'), (6.924624764173099e-90, '2'), (6.759020997714 71e-105, '3'), (1.237325440961051e-118, '4'), (1.8761600964514 85e-138, '5')]	Negative		2
13	12	[(3.175642358404346e-22, '1'), (3.02452599564308 7e-19, '2'), (1.22519051198476e-23, '3'), (5.0652665094438125e-25, '4'), (5.068794074222687e-28, '5')]	Negative		2
14	13	[(1.4296900000865428e-44, '1'), (9.724605868009125e-38, '2'), (4.968721722100074e-43, '3'), (7.943981071898768e-48, '4'), (2.5206747944718 78e-53, '5')]	Negative		2
15	14	[(4.4755291057985e-44, '1'), (1.2364675282611392e-38, '2'), (1.0391948738690784e-46, '3'), (5.9062521964506 58e-47, '4'), (2.1283518296635325e-51, '5')]	Negative		2
16	15	[(5.2022200469555124e-51, '1'), (8.75095800753960 9e-45, '2'), (4.6596618446853407e-51, '3'), (1.8144636328579465e-56, '4'), (6.799572346600288e-60, '5')]	Negative		2
17	16	[(8.326276896565601e-71, '1'), (5.4934498854965 44e-60, '2'), (6.1376155037581 06e-66, '3'), (6.958472139212199e-72, '4'), (1.5491678109184 08e-76, '5')]	Negative		2
18	17	[(9.366105909011263e-179, '1'), (5.1451173455000706e-148, '2'), (5.392635354534663e-180, '3'), (2.5129950565311805e-190, '4'), (4.783697120398264e-208, '5')]	Negative		2
19	18	[(1.2649763155554014e-32, '1'), (3.367502640571003e-28, '2'), (1.3422780349071244e-32, '3'), (8.626451473559118e-37, '4'), (1.0098670379413143e-40, '5')]	Negative		2
20	19	[(9.89446280503574e-122, '1'), (4.636656577169887e-102, '2'), (3.741667453980016 5e-121, '3'), (1.8459512843406425e-137, '4'), (1.1752541196988275e-155, '5')]	Negative		2
21	20	[(9.493145553208207e-220, '1'), (3.554300506950632e-184, '2'), (1.2564123039233031e-212, '3'), (9.855074850503718e-228, '4'), (2.0051401450883133e-252, '5')]	Negative		2
22	21	[(4.344795293126701e-42, '1'), (9.073854476607 41e-37, '2'), (6.749918374436325e-41, '3'), (2.890453689945905e-46, '4'), (7.309868098605174e-49, '5')]	Negative		2
23	22	[(3.535691589243922e-74, '1'), (2.142815398361456e-61, '2'), (1.9885297612372847e-73, '3'), (3.9981784642750214e-80, '4'), (3.864024106190053e-89, '5')]	Negative		2
24	23	[(3.599757797363786e-28, '1'), (6.096788890990757e-25, '2'), (4.350158070887702e-26, '3'), (5.107491370700 48e-29, '4'), (5.754849760226616e-33, '5')]	Negative		2
25	24	[(7.676359719465796e-54, '1'), (1.2972871903513135e-42, '2'), (4.083124563303299 6e-49, '3'), (2.013171636137663 8e-56, '4'), (1.2367470417004807e-65, '5')]	Negative		2
26	25	[(2.300391734231407e-63, '1'), (6.2430561815040 51e-55, '2'), (4.916224255643061 3e-60, '3'), (4.3344471706577 57e-65, '4'), (9.570110228478585e-71, '5')]	Negative		2
27	26	[(3.598256186962277e-26, '1'), (3.776196041390 5127e-22, '2'), (5.6905502082345 64e-25, '3'), (2.655219951949827 3e-27, '4'), (1.1051764315461001e-29, '5')]	Negative		2
28	27	[(9.806125752802563e-21, '1'), (2.188370924669046e-18, '2'), (7.005971970797 98e-20, '3'), (6.258748745635434e-22, '4'), (1.5173352766328223e-23, '5')]	Negative		2
29	28	[(1.1616649171887 47e-19, '1'), (7.037506144199893e-19, '2'), (1.4583618391258705e-19, '3'), (1.0557404894030044e-20, '4'), (1.2691996228518 39e-20, '5')]	Negative		2
30	29	[(2.289127012555 4314e-45, '1'), (1.5996192328816902e-39, '2'), (4.826926180458973 6e-45, '3'), (2.791149889102423 6e-49, '4'), (6.523944031808186e-55, '5')]	Negative		2
31	30	[(1.2486340747644432e-12, '1'), (4.746072936827621e-11, '2'), (1.836296752869 78e-11, '3'), (1.828953312834889e-12, '4'), (6.962136800171 29e-13, '5')]	Negative		2
32	31	[(2.300421675404689e-25, '1'), (5.297710550566453e-22, '2'), (2.033426574979232 3e-23, '3'), (9.686938931341956e-27, '4'), (2.0854030550461746e-29, '5')]	Negative		2
33	32	[(8.655961698911415e-53, '1'), (1.1214114724350308e-47, '2'), (1.7646283009785 23e-53, '3'), (2.243074935449642 6e-58, '4'), (2.7833310607532477e-63, '5')]	Negative		2
34	33	[(1.3468038967519944e-27, '1'), (9.277140728966769e-25, '2'), (1.5342126178492234e-25, '3'), (2.188783990706569 5e-30, '4'), (1.7028858275752 27e-31, '5')]	Negative		2
35	34	[(1.2861666625903779e-36, '1'), (3.388668209410422e-30, '2'), (5.529547691923579e-36, '3'), (4.288711531175726 6e-41, '4'), (1.4858575545269 96e-47, '5')]	Negative		2
36	35	[(2.3355171219255377e-26, '1'), (5.782634626281131e-22, '2'), (1.5755354966838319e-25, '3'), (1.5204456895565209e-25, '4'), (8.243818106186289e-27, '5')]	Negative		2
37	36	[(2.405417628573952 7e-35, '1'), (1.3043308733306631e-27, '2'), (2.892837843448504 5e-33, '3'), (2.292510824253329 7e-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



Standards and Best Practices

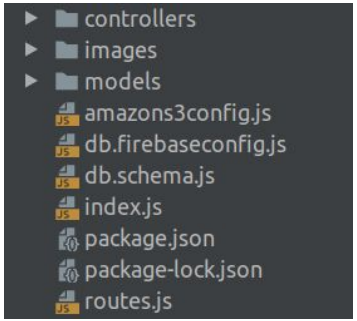
Clean code and use of Comments

```
/**
 * 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
```

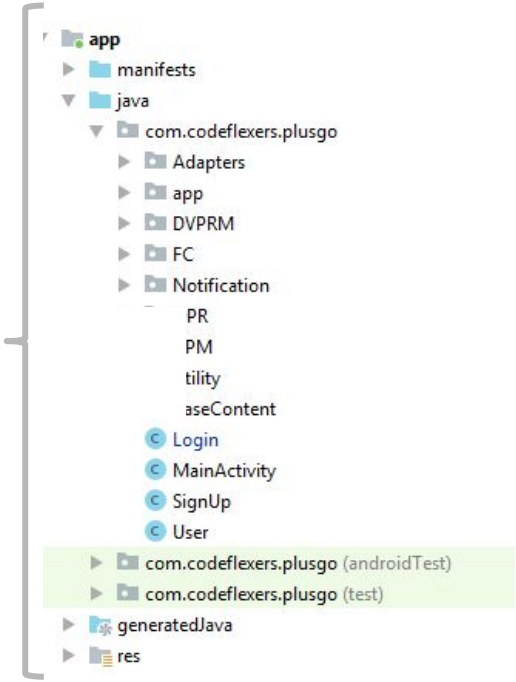
Standard Naming Conventions

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,
```



File and Folder Organization



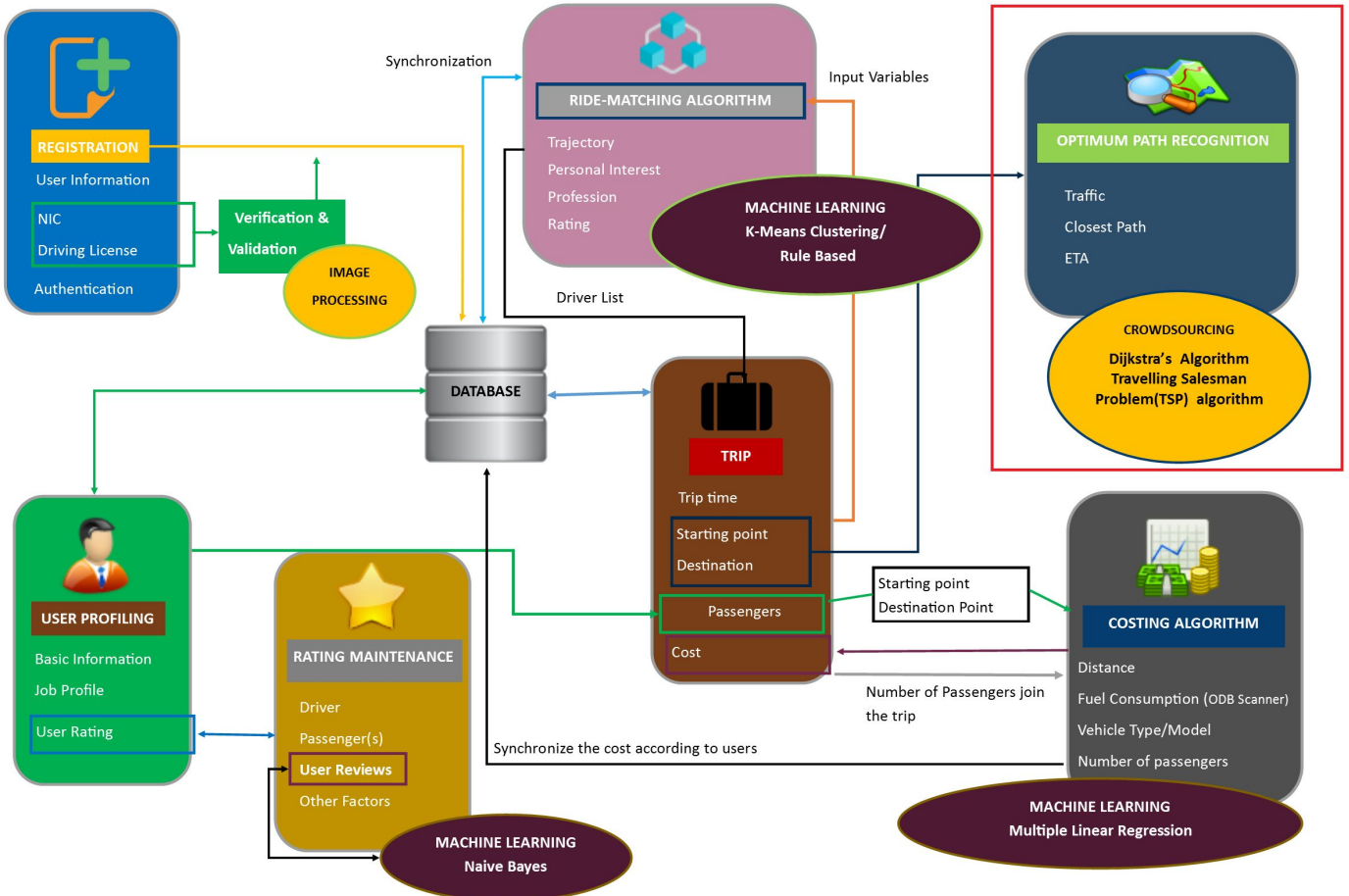
Use of Object Oriented Concepts

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

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

Optimum Path Recognition

IT16033474

High Level Diagram



Objective

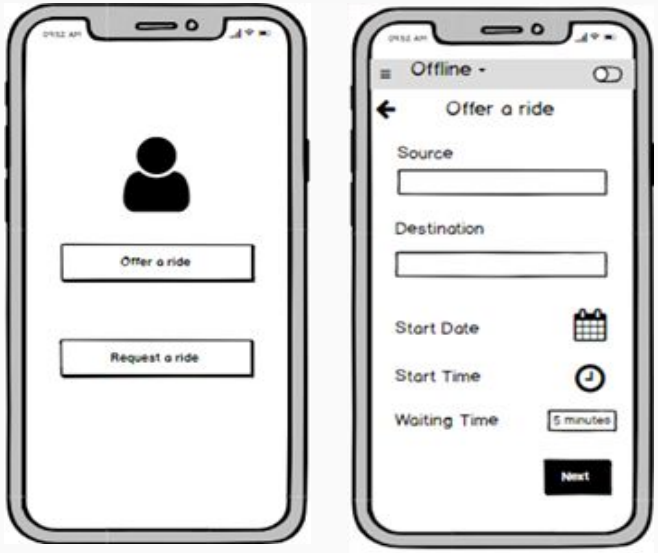
- In Order to display the route,finding the shortest path which connect starting point and destination ,while tracking the position and order of intermediary locations by using crowdsourcing technology.

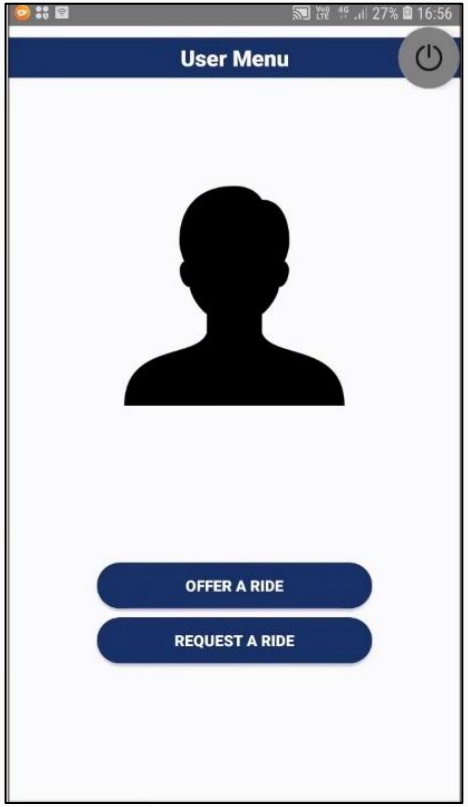
Research Gap

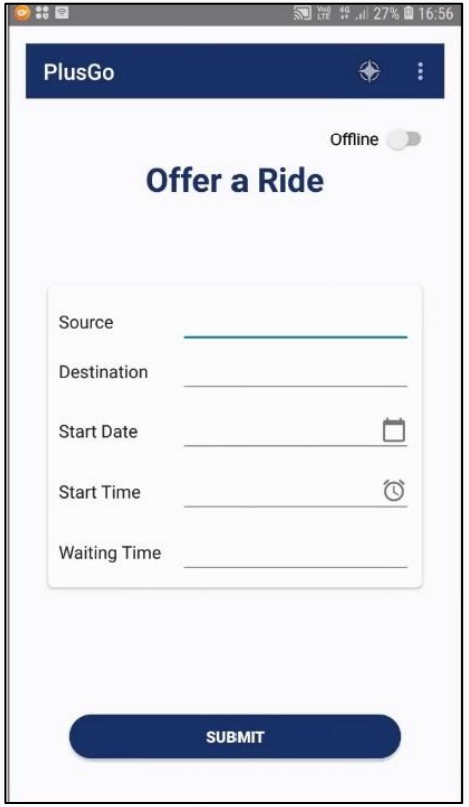
Features	UBER	UDIO	Carpooling.lk	RideShare.lk	+GO
Crowdsourcing to improve the optimum path by analyzing more than one algorithm.	✗	✗	✗	✗	✓
Allowing the registered users to enter the live updates by uploading images.	✗	✗	✗	✗	✓

Prototype vs Implementation

DRIVER OFFER RIDE

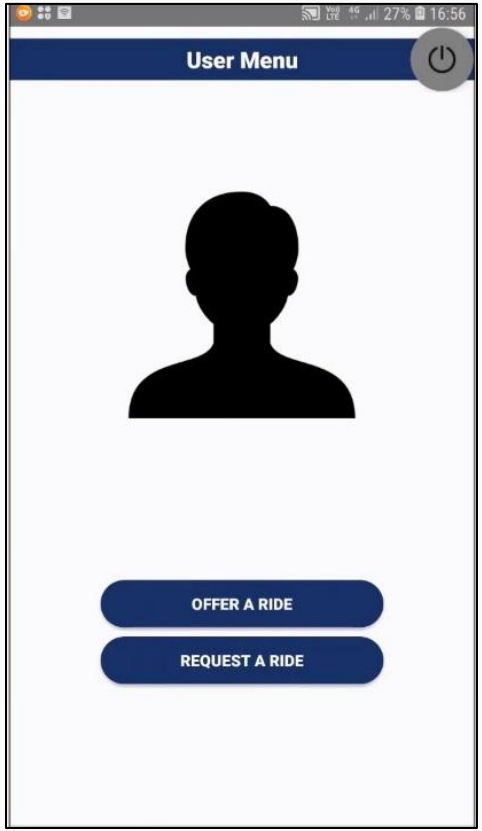


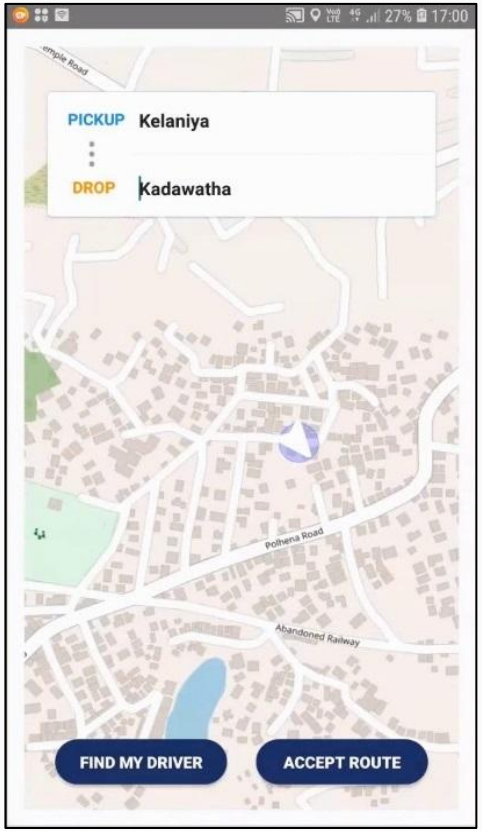




Prototype vs Implementation cont...

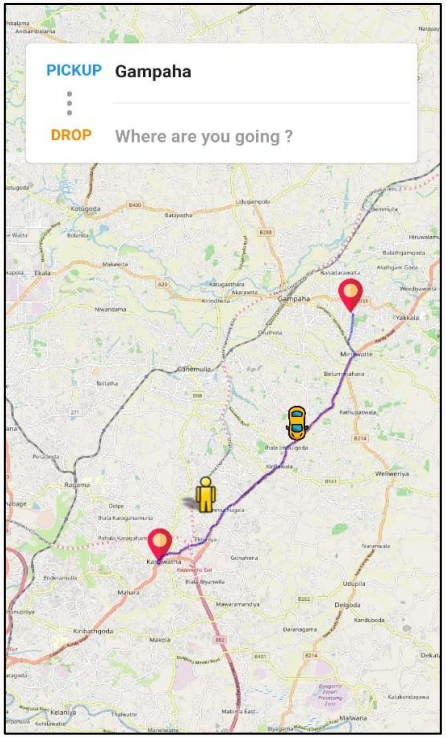
PASSENGER FINDS A RIDE

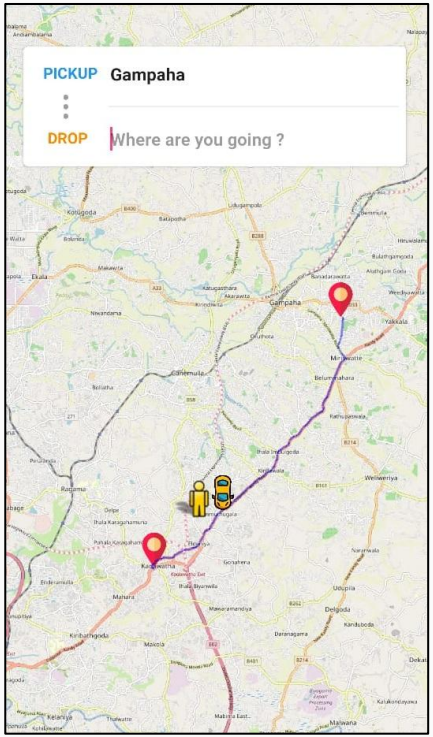


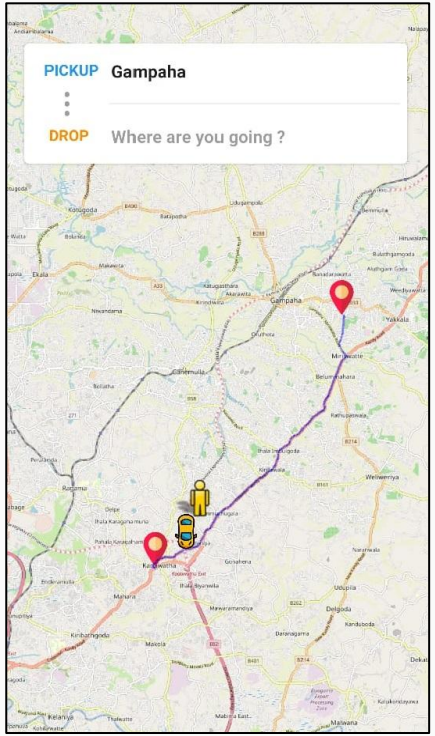


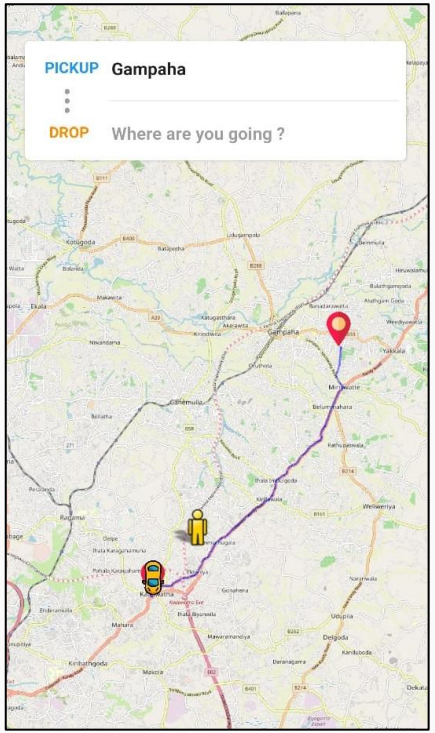
Prototype vs Implementation cont...

DRIVER NAVIGATE TO PASSENGER LOCATION



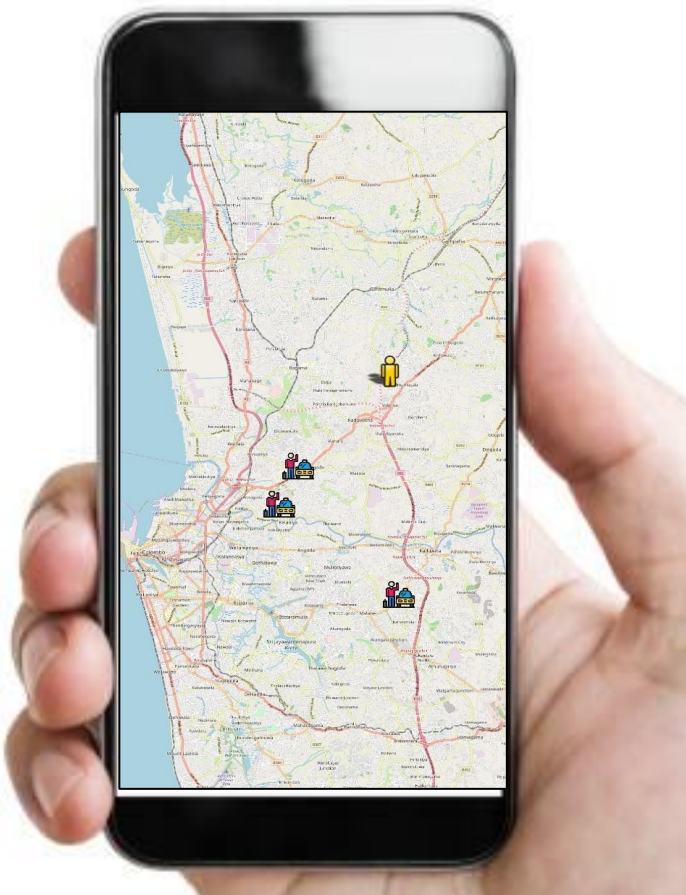






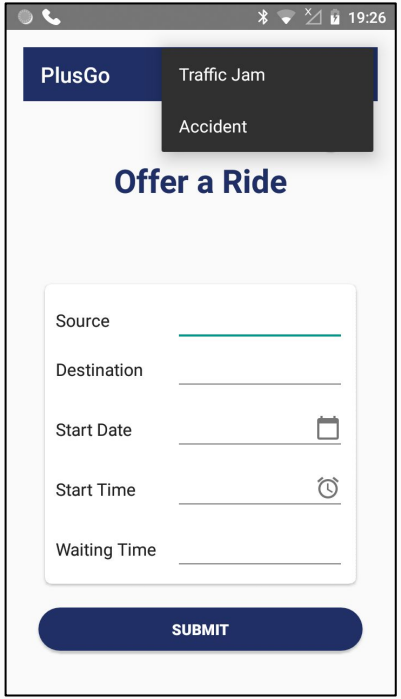
Prototype vs Implementation

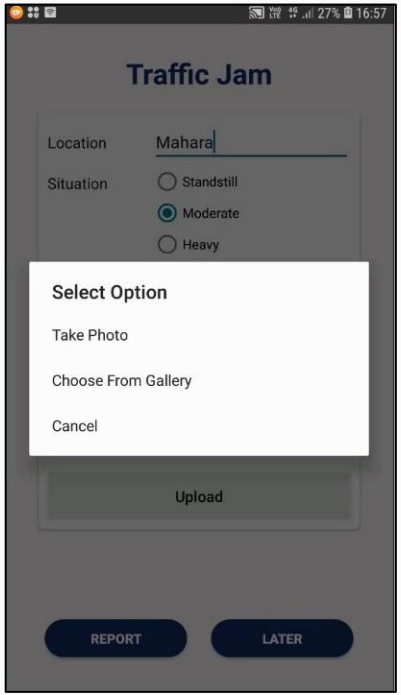
VIEW CURRENT PASSENGERS

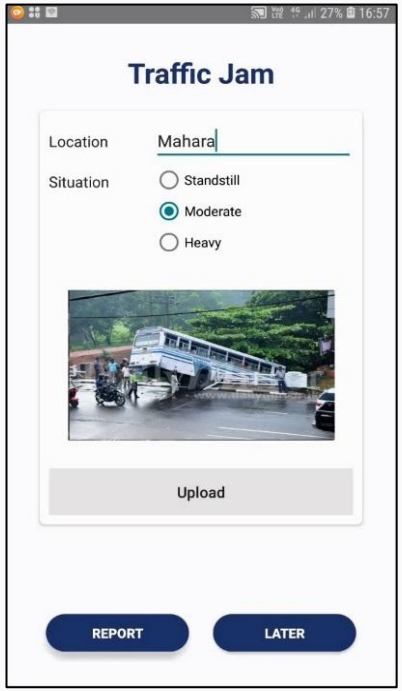


Prototype vs Implementation cont...

REPORT TRAFFIC JAM

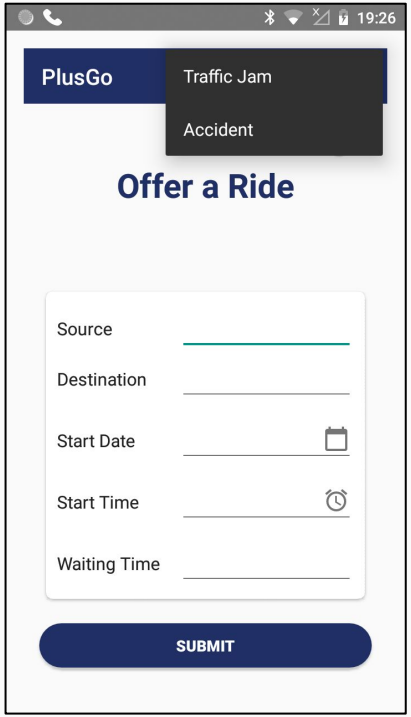


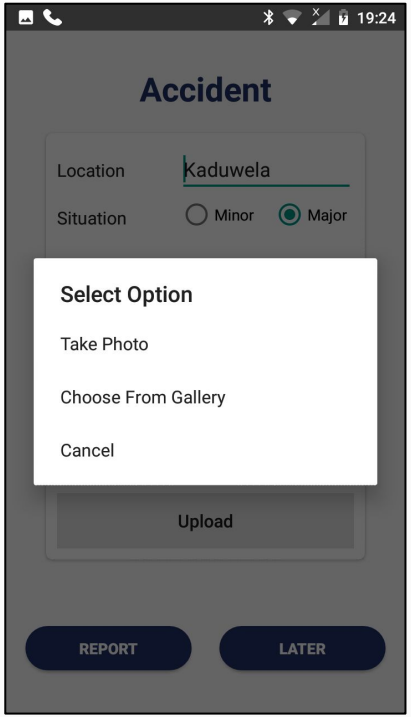


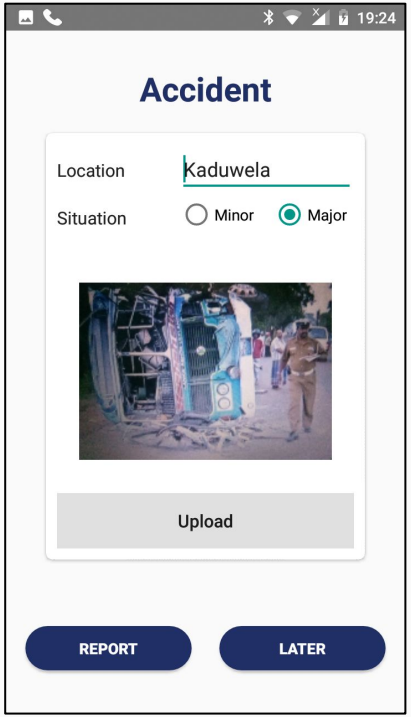


Prototype vs Implementation cont...

REPORT ACCIDENT

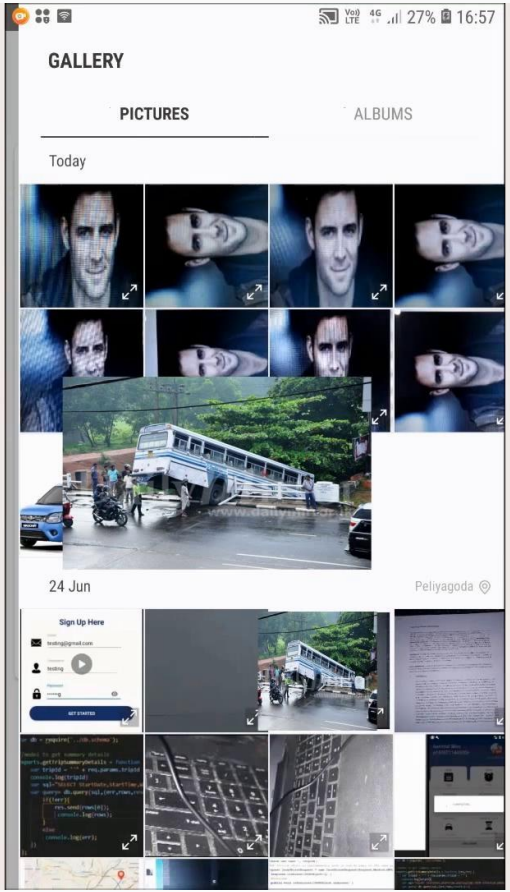






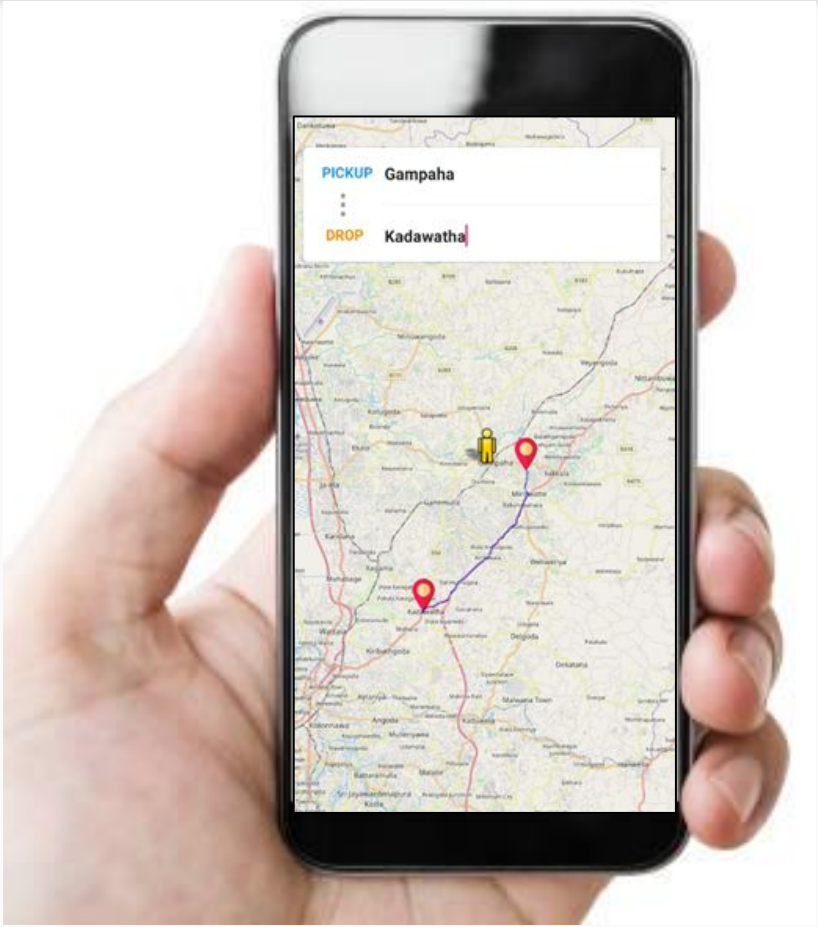
Final Output - Report Traffic Jam

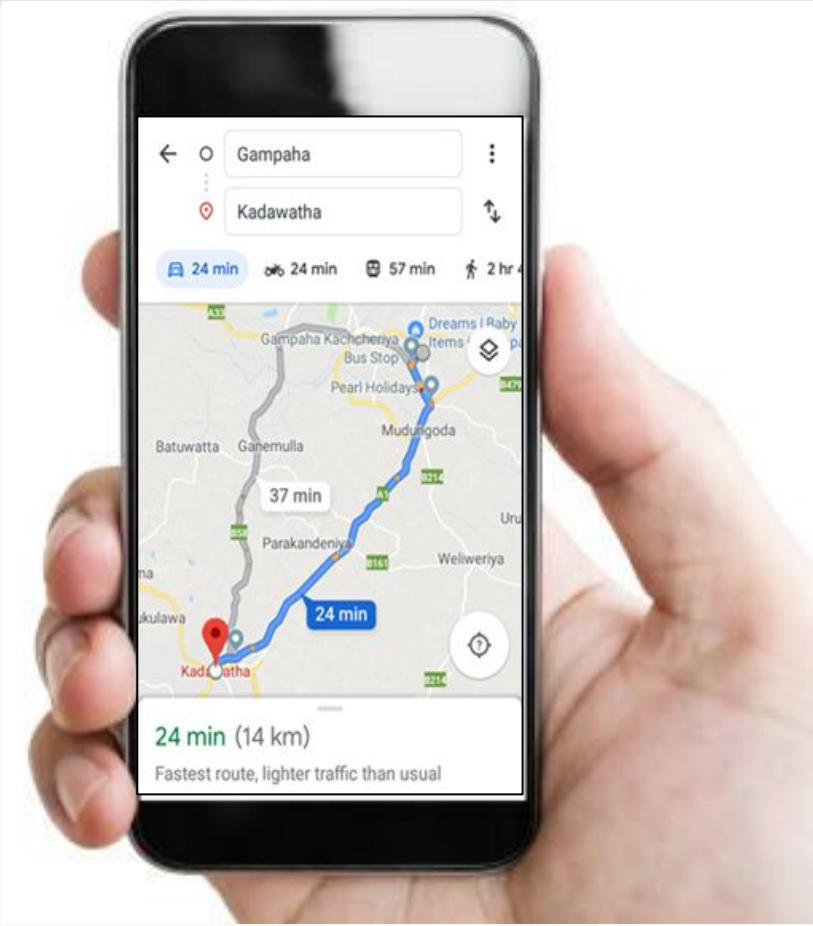
Registered Users



Utilization of Crowdsourcing platform

RESULTS


PICKUP Gampaha
DROP Kadawatha


Gampaha
Kadawatha
24 min
24 min
57 min
2 hr
Gampaha Kachcheriya Bus Stop
Pearl Holidays
Mudungoda
Batuwatta
Ganemulla
37 min
Parakandeniya
Weliweriya
24 min
24 min (14 km)
Fastest route, lighter traffic than usual

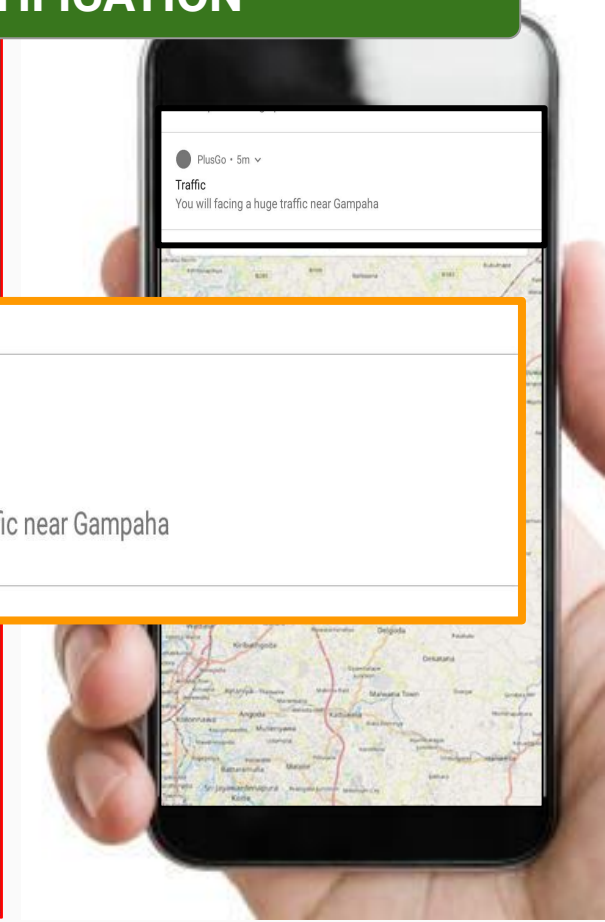
Final Demo - Display real route

RESULTS

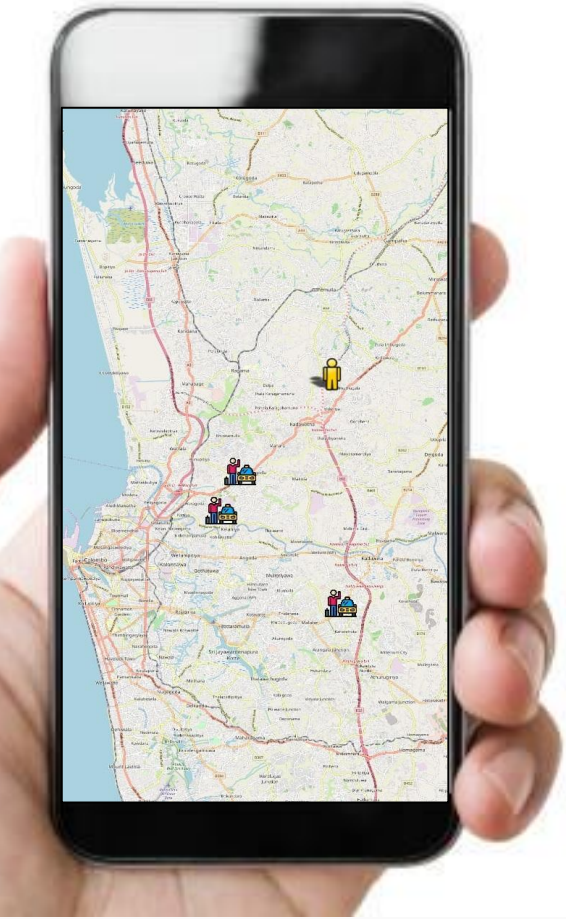
CROWDSOURCING INFORMATION NOTIFICATION

```
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) {

        }
    });
}
```



RESULTS



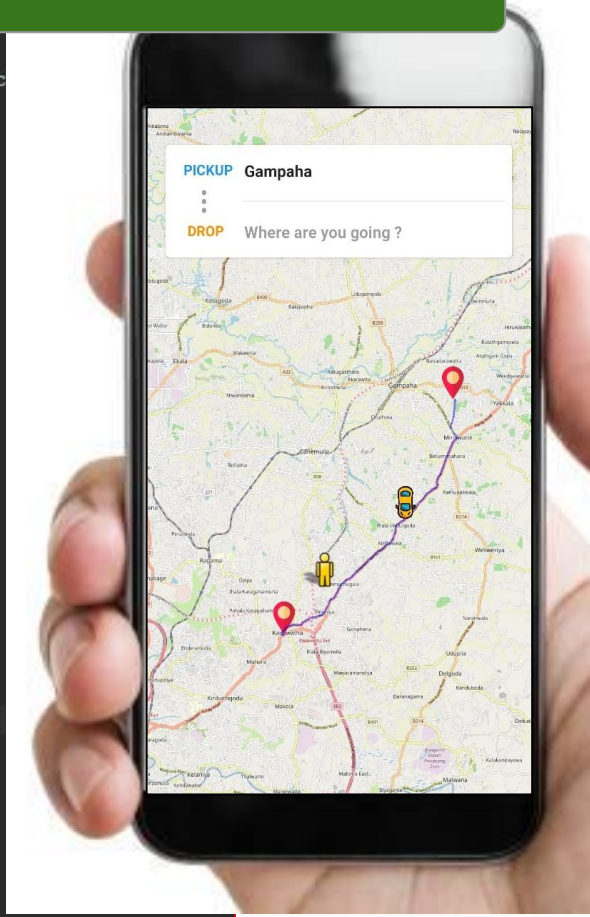
DISPLAY CURRENT PASSENGERS TO DRIVER

```
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

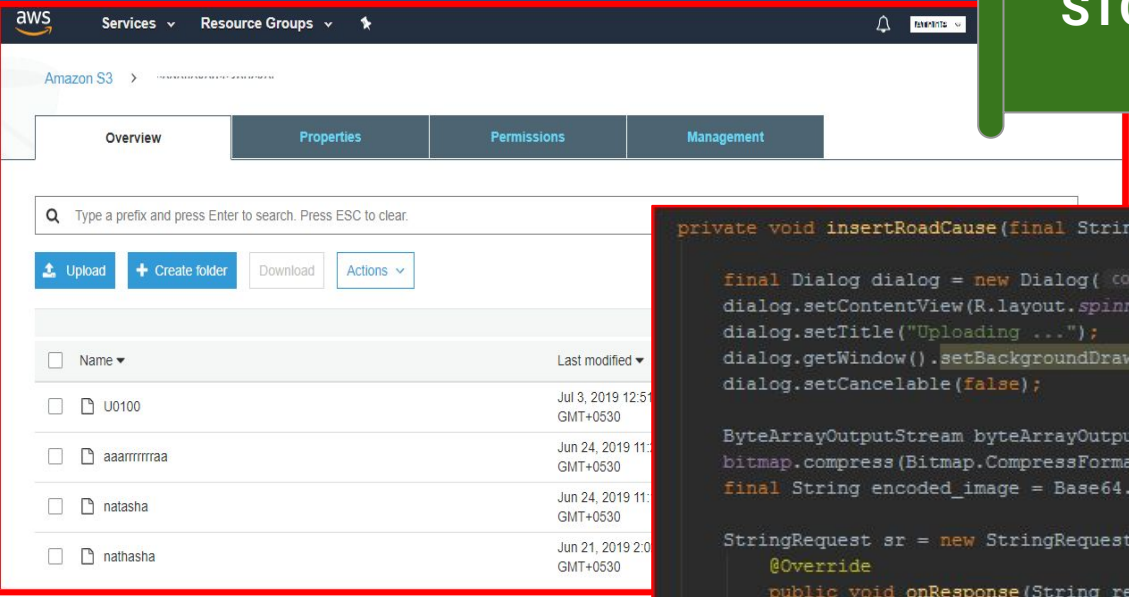
DISPLAY DRIVER NAVIGATION PATH

```
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



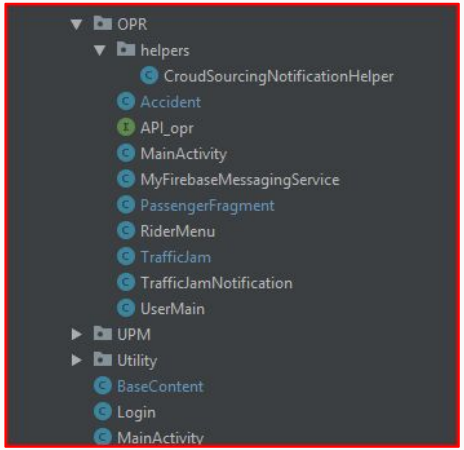

```
private void insertRoadCause(final String status) {

    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



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);
```

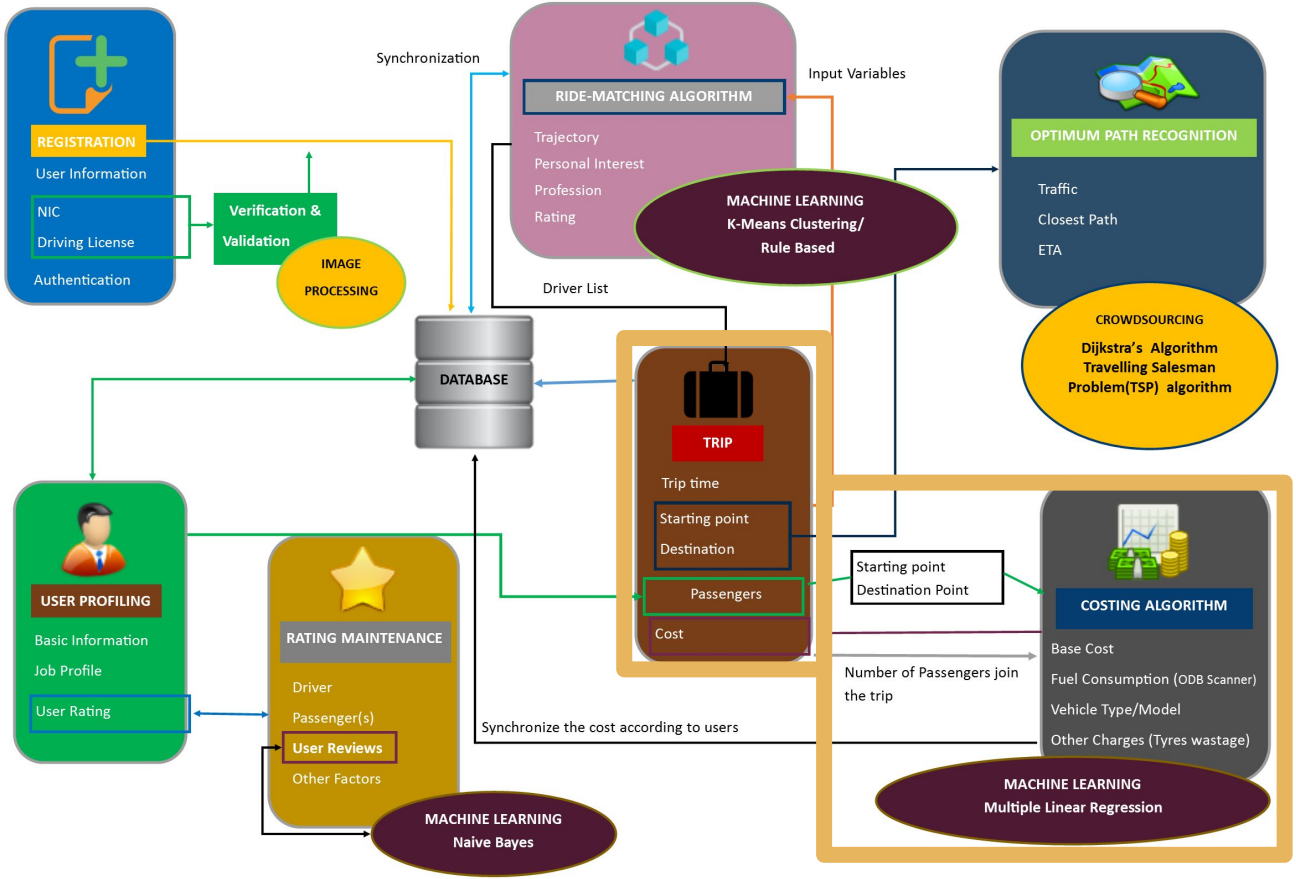
Consistent Indentation

```
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

IT16011380

High Level Diagram



Objectives

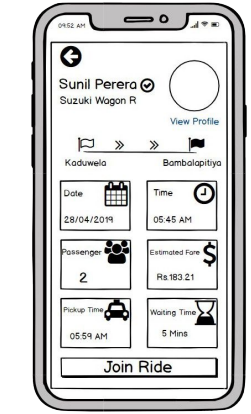
Main objectives are to examine how fare calculation is divided among passengers and to communicate between both drivers and passengers via firebase push notifications related about trip.

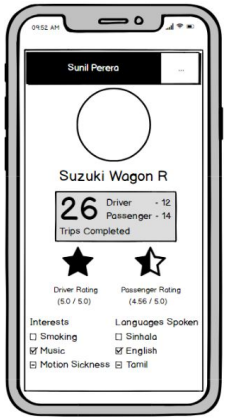
Research Gap

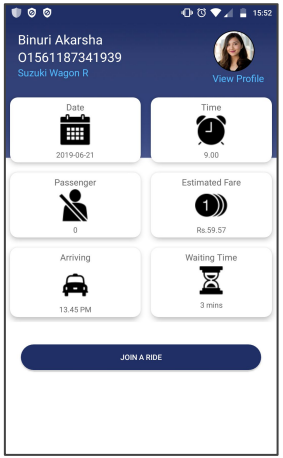
Features	UBER	UDIO	Carpooling.lk	RideShare.lk	+GO
The system will decide the estimated fare before joining the trip.	✓	✓	✗	✗	✓
Vehicle fuel consumption calculated according to the condition of the vehicle	✗	✗	✗	✗	✓
Passengers can get off in any place where is the between source and destination because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

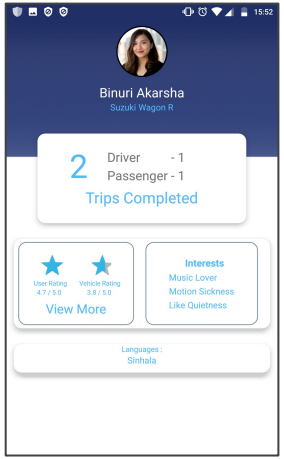
Prototype vs Implementation

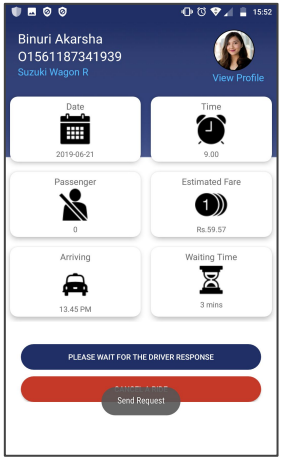
Interfaces related to Request trip for Passenger





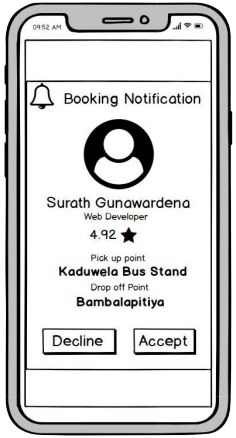


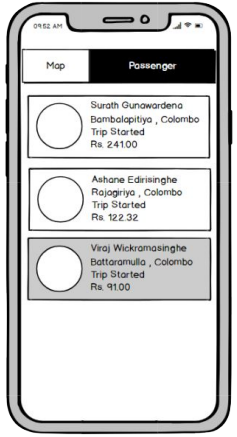


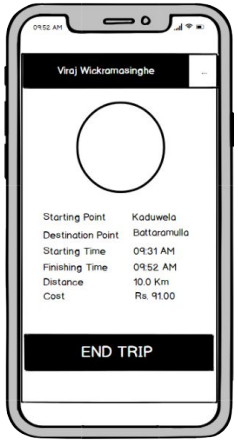


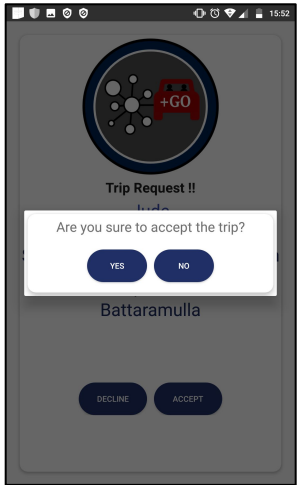
Prototype vs Implementation

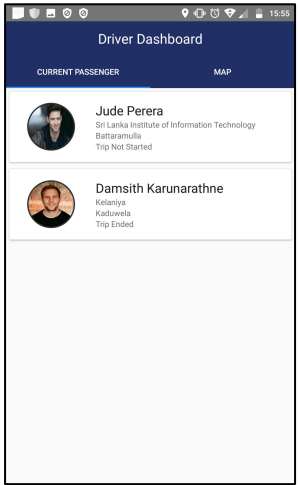
Request trip
Related Interfaces
for
Driver

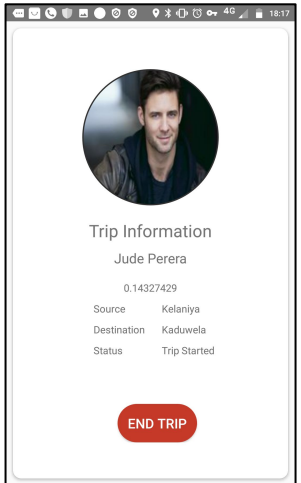






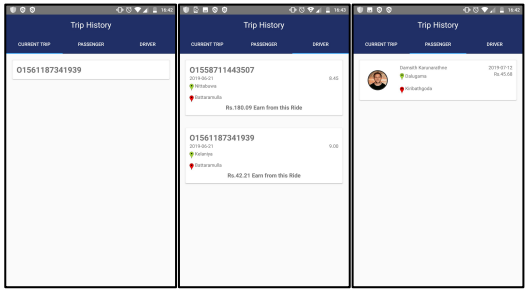


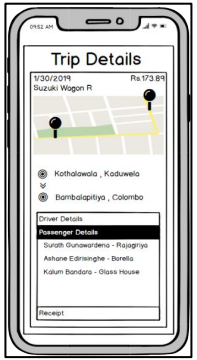


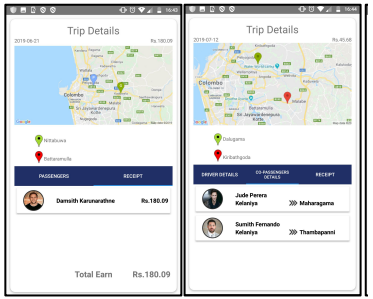


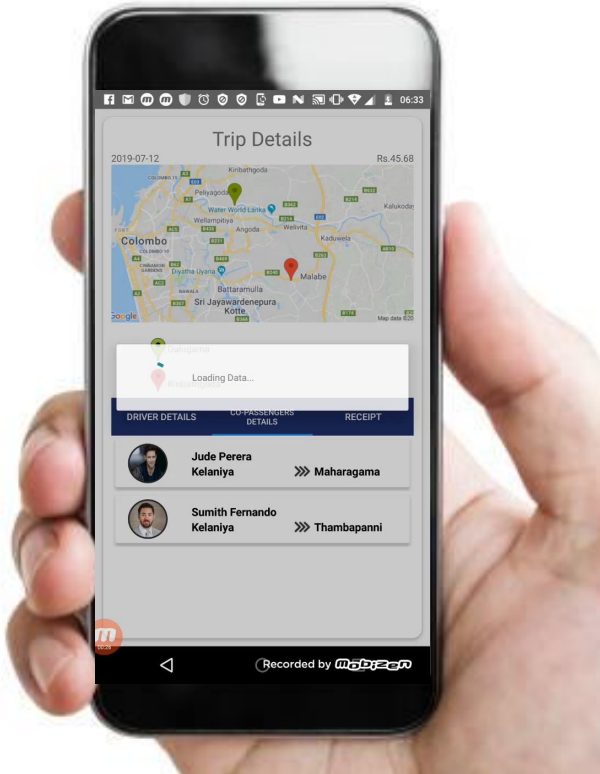
Prototype vs Implementation

Trip History
Related Interface



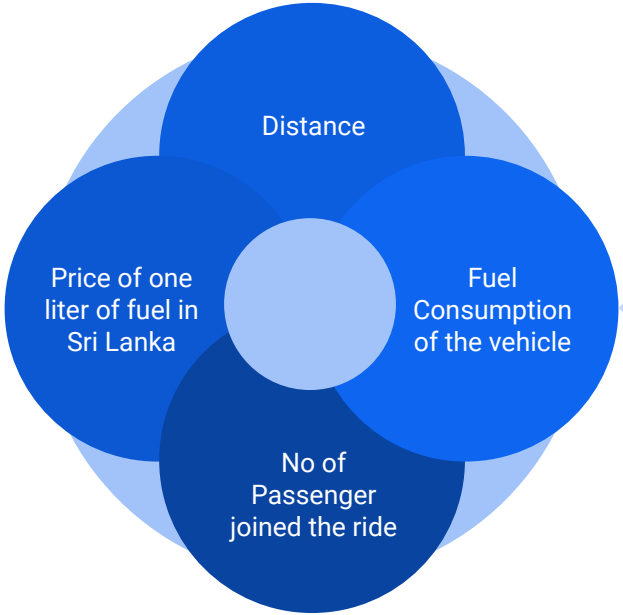






The Specialized Area and the Use of Technologies

Estimated fare calculation before Join the Ride



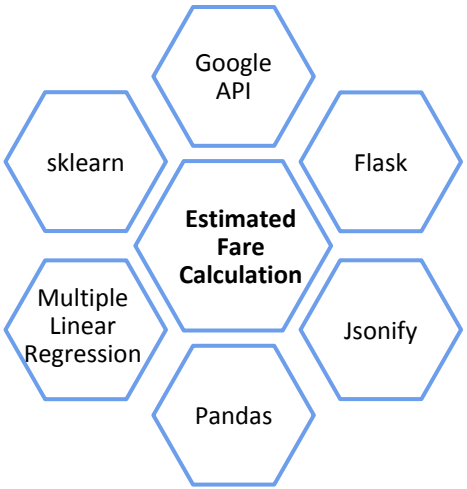
Attributes of Selected Driver's Vehicle

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

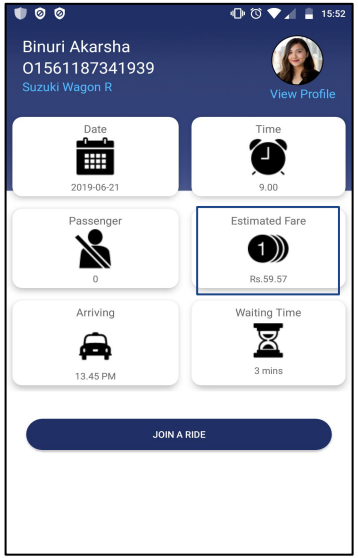
The Specialized Area and the Use of Technologies

Estimated fare calculation before Join the Ride

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



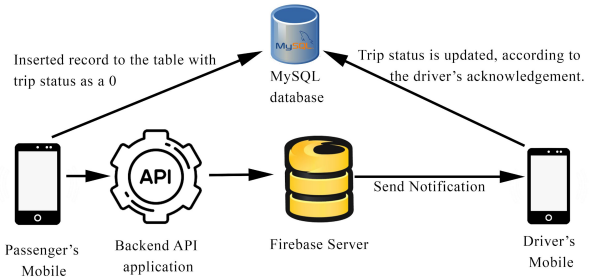

```
/*
Response of the paython file of fuel prediction.
Current Price of liter is fuel declare in there
 */
float float_fuelPrediction = Float.parseFloat(fuelPrediction);
float getcurrentPassenger = Float.parseFloat(txtCurrentPassenger.getText().toString());

/*
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);
```



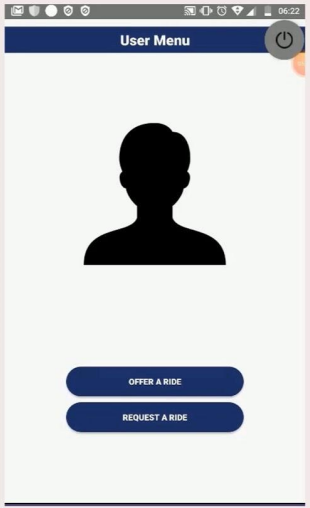
Make	Model	Year_of_ Man	First_Reg	Cylinders	Fuel	Capacity	kW	Mileage	Fuel_consumption
SUZUKI	WAGON R	2018	2018	3	P - Hybrid	658	38	981	18
SUZUKI	WAGON R	2018	2019	4	P - Hybrid	658	38	7837	18.3
SUZUKI	WAGON R	2018	2018	3	P - Hybrid	658	38	5337	17.8
SUZUKI	WAGON R	2018	2018	4	P - Hybrid	658	38	25000	18
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

The Specialized Area and the Use of Technologies

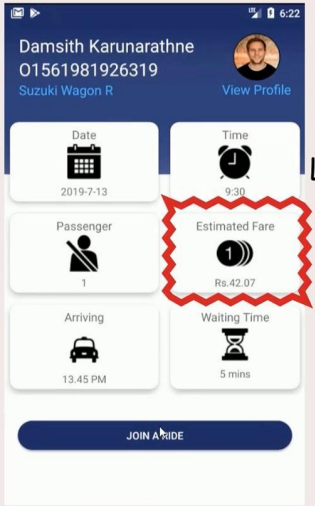
Join a Ride



Driver's Mobile



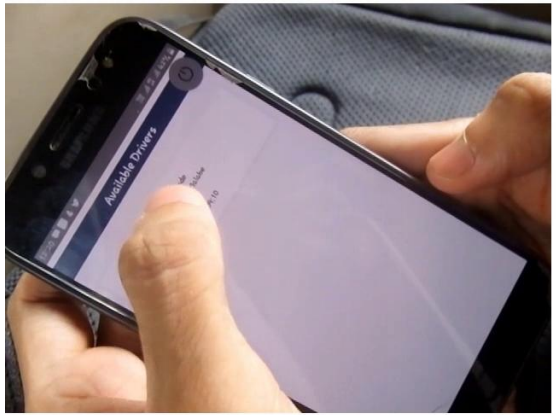
Passenger's Mobile

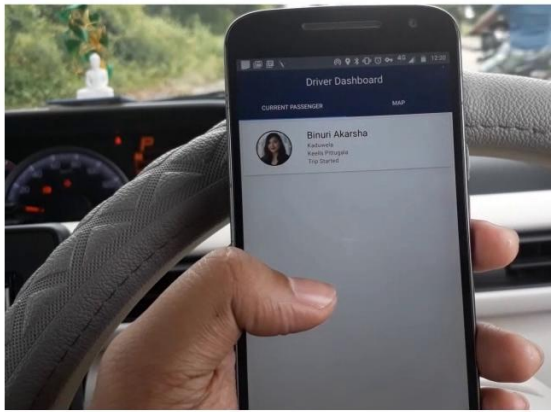


Multiple Linear Regression

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

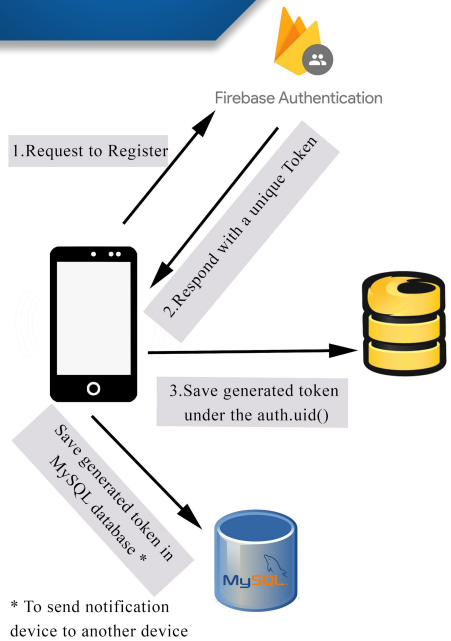
Final Output - Real Time Notification





The Specialized Area and the Use of Technologies

Firebase Notification



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

The Specialized Area and the Use of Technologies

Actual Fare Calculation

Real time fuel Consumption using OBD

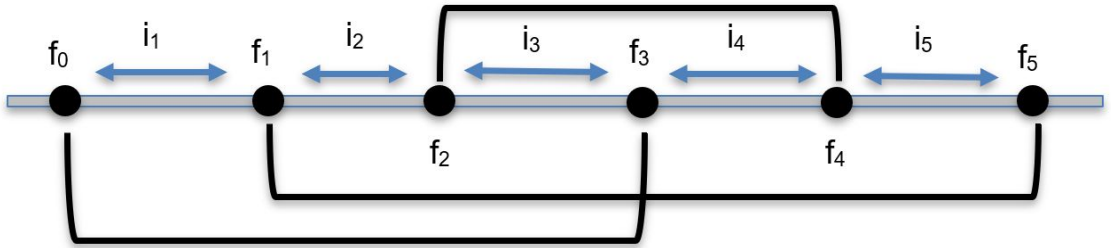
No of Passengers

Current price of one liter of the fuel

What is the Segment ?

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

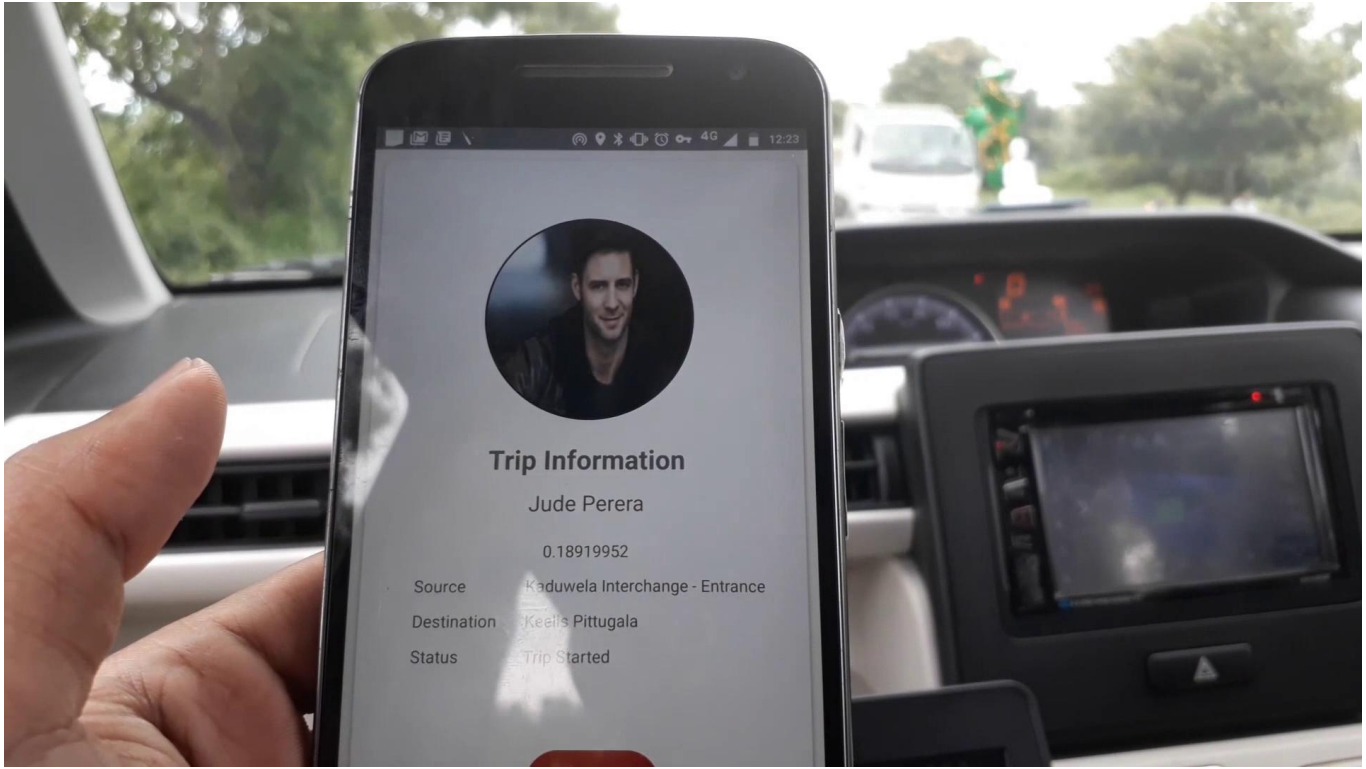
$$C\ i\ (\text{Total fare for the i th segment}) = \frac{\text{Fuel Consumption} * \text{price}}{\sum \text{passenger}}$$



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

Final Output

How OBD Reads Fuels consumption



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

GET http://18.221.123.144:8083/trip/history/passenger/U1558711443502 Send Save

Pretty Raw Preview Text


```
{"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:APA91bGXaHfDND9Hn893wXh1fSyWipzAkvIOdCXUr06NOJTfjmgI-UQtUDZj8zCThXyNBpBFLZ-5Zx_yybufmSJQUPvICLEXUmEsN1rU45Jp0aCmNg9NHOchMSUIoiKBSpmuSI91HPsg"},
{"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:APA91bGXaHfDND9Hn893wXh1fSyWipzAkvIOdCXUr06NOJTfjmgI-UQtUDZj8zCThXyNBpBFLZ-5Zx_yybufmSJQUPvICLEXUmEsN1rU45Jp0aCmNg9NHOchMSUIoiKBSpmuSI91HPsg"},
{"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-BQMYvVCIPArrXbph7eKOxXdV2Nj5ZTTXpDTkksCmUL_NoHgm3m7EeJT0Tz757CFAQquF05qOVoVaPeaX4cJ-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-BQMYvVCIPArrXbph7eKOxXdV2Nj5ZTTXpDTkksCmUL_NoHgm3m7EeJT0Tz757CFAQquF05qOVoVaPeaX4cJ-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-BQMYvVCIPArrXbph7eKOxXdV2Nj5ZTTXpDTkksCmUL_NoHgm3m7EeJT0Tz757CFAQquF05qOVoVaPeaX4cJ-uE-qXa-s0azRUem-kTWX5nyfsc6rRMNTAyha29"}]
}
```


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
```

Backend Get Response

POST http://18.221.123.144:8083/trip/newRequest

none form-data x-www-form-urlencoded raw binary GraphQL BETA

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

Body Cookies Headers (6) Test Results

Pretty Raw Preview JSON


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


Backend POST Request and Response

Work Progress

100%

- Designing trip details UI
- Designing User details UI
- Designing Trip History UI
- Designing end trip UI
- Designing driver notification (...)
- Designing Current Passenger (...)

Frontend

100%

- Collect the dataset of the (...)
- Create algorithms for predict (...)
- Web Services
- Integrate the OBD2 scanner
- Real time fare calculation
- Create algorithms for estimated (...)
- Send request notification (...)

Backend

100%

- Integration and Testing
  - System Integration
  - Regression Testing
  - Final Test
  - Unit Testing

Testing

Standards and Best Practices

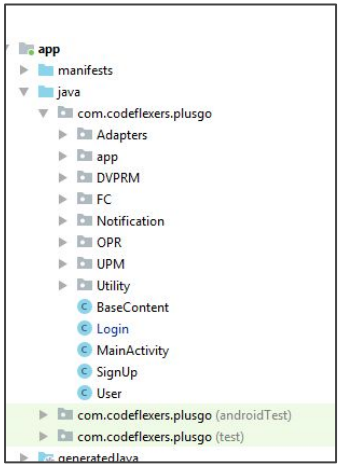
Clean code and use of Comments

```
/*
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());
            }
```

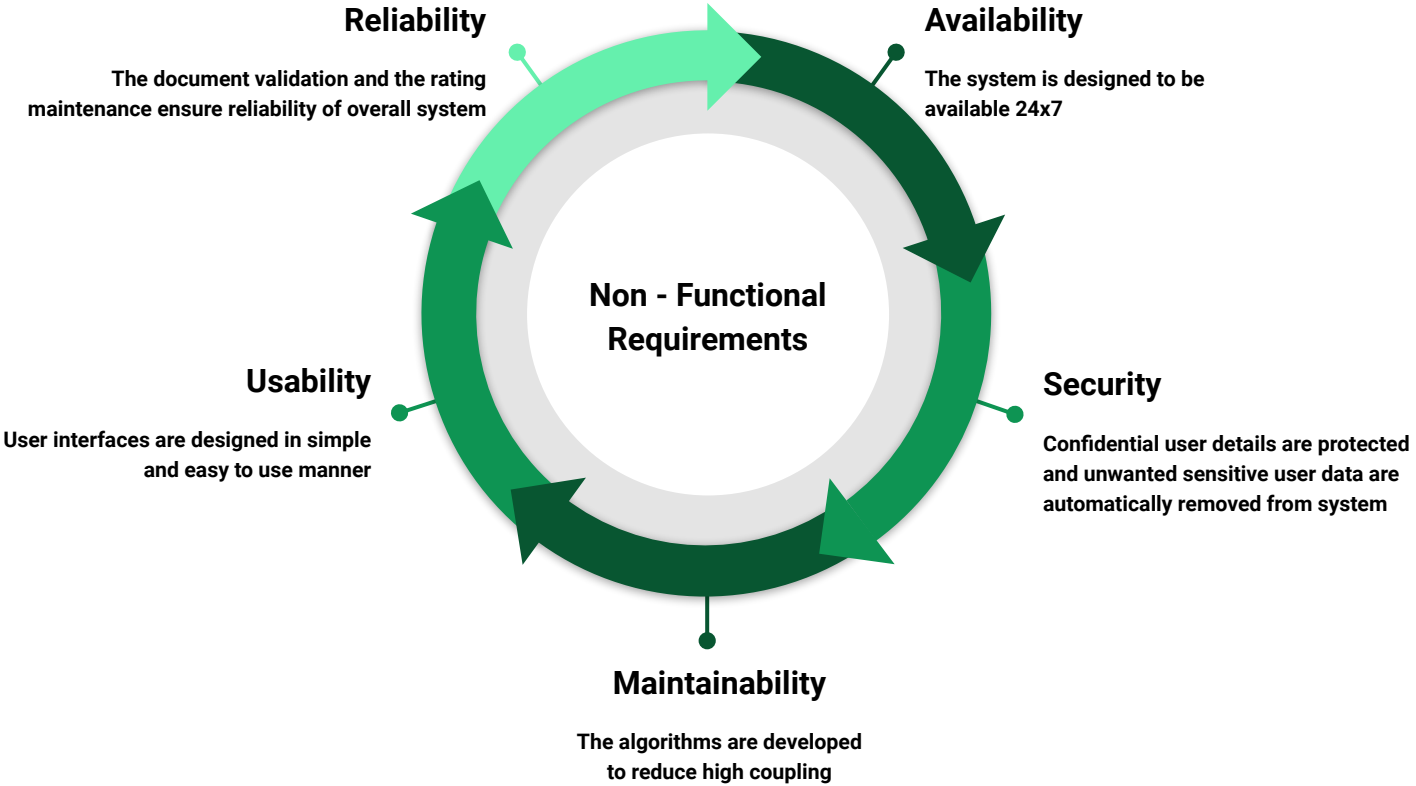
File and Folder Organization



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 of +Go



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

References

- https://www.newsfirst.lk/2017/03/16/rs-500m-loss-incurred-daily-traffic-congestion-transport-authorities-observe/
- https://indi.ca/2015/10/colombo-vehicle-statistics-2015/
- https://www.newsfirst.lk/2018/10/11/even-ministers-hate-the-rajagiriya-traffic/
- https://www.newsfirst.lk/2018/08/11/a-new-kelani-bridge-to-lessen-the-traffic/
- https://monkeylearn.com/sentiment-analysis/
- https://www.kdnuggets.com/2018/03/5-things-sentiment-analysis-classification.html
- https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/
- https://towardsdatascience.com/sentiment-analysis-with-python-part-1-5ce197074184
- https://www.datacamp.com/community/tutorials/simplifying-sentiment-analysis-python

Thank You