EasyChair Preprint

№ 2755

Ready Rescue – A platform which connects mechanics with customers in need of Roadside assistance

Yohan Silva

February 22, 2020

Ready Rescue – A platform which connects mechanics with customers in need of Roadside assistance

Yohan Silva
dept. of Science and Technology
Informatics Institute of Technology (of Affiliation) University of Westminster
Colombo, Sri Lanka
yohan.2016235@iit.ac.lk

Abstract*— In this fast-moving world facing a problem where the traveling vehicle breakdowns will cause a problem to the person and put the person into a great difficulty to recover from this situation. This problem might get worse depending on the place of breakdowns such as being on a highway or an unknown area. The "Ready Rescue" application will provide the user the ability to select the type breakdown which has occurred and request the closest mechanics in the area to come and provide assistance. This application has 2 main sections which are the User's portal and the Mechanic's portal. Both the users will require to register or login before using the application. By using GPS, the user could pin the location of the breakdown and the mechanic could see this location through the application and proceed there. Considering the Mechanic's section, the mechanic could register to the application by selecting the services which he desires to provide and once he has registered to the system the mechanic data will be updated in the Firebase database. The mechanic will not be able to use the application until the application admin which will be using a web-based system provides access to the mechanic. The Mechanic will be able to find the location of the user and provide the user his live location using Google Maps API. By using the GeoLocation feature with Firebase the mechanic's and user's location could be shared amongst all in Realtime. The User will receive the Closest Mechanic who is registered for the service requested. The users can furthermore request a mechanic to become a favorite and the user could later straightaway request for him. Furthermore, users could take quizzes on vehicle maintenance during the time taken by the mechanic to arrive and gain Points that could be redeemed later.

Index Terms*— Google Map API, Realtime Database, Firebase, List of Favorites, Mechanic Services, Redeemable Points, BI tools.

I. INTRODUCTION

Vehicles are an essential mode of transportation, and the demand for vehicles keep on increasing every day. According to a government statistical survey of All types of vehicles registered from 2013 to 2017 have increases from 5,203,678 to 7,247,122 (Refer Annexure 3). This is clear evidence that the demand and the number of vehicles has increased along the years (Traffic, Department of Motor, 2018). With the use of the increase in the use of these vehicles it is evident that at some point a vehicle will go through a breakdown even with utmost care and proper service it could at least experience a tire puncture on the road. Therefore, a vehicle breakdown is unpredictable, and we may never know when or where it would happen. "A breakdown can be a stressful experience – and no matter how old or new the car, every car can be susceptible to a breakdown under the right conditions. While all cars have their differences there are a few common reasons that all cars tend to breakdown" (Colorado, 2018) At the face of an actual breakdown the inefficiency in reach out for roadside assistance could be considered as the main problem for the drivers.

According to the distribution of top 20 occupations, where the greatest number of employees quitted from 2015 to June 2017. It is observed that employees who are already working in this sector are quitting their job which is 440 in 2015, 870 in 2016 and 415 in 2017 (Department of Census and Statistics, 2017). From this we can understand that there is a considerable number of people leaving the industry. All the people here who quitted will be unemployed at this situation. Due to this instability of the availability of vehicle mechanics the mechanic a customer expects from a service center might not be available the next day the customer requires a service.

There might be many reasons why a person might drive a vehicle or be driven by it, some reasons are such as going to work daily, going grocery shopping, going on trips, etc. at these instances mainly in driving on trips the user might drive into an area who is unknow and completely new to. "A person driving this vehicle is in a new neighborhood where they have no clue about any repairing centers in their vicinity or they themselves know nothing about fixing their vehicles." (Philip, Nayak,

Patel, & Devashrayee, 2018). Accordingly, when a vehicle breakdown in an unknown region it is hard to contact a mechanic and request for assistance.

II. METHODOLOGY

All these vehicles broken down at some point on the road will need roadside assistance to recover from the situation. This is where roadside assistance comes into action to provide the broken-down vehicles the assistance they need. Roadside assistance can come in various meanings, here it is used for the definition of the assistance provided by a vehicle mechanic to a customer who is in need in a situation of a vehicle breakdown. For a customer to a connect with a mechanic Firstly the mechanic should set his state as working which will trigger a flag in the firebase database as to true. After that the mechanic will start receiving requests from the customer. Once a customer has experienced a vehicle breakdown the person could select the type of service which are Tire Change, Battery Jump Start, Lock Smith Service, Fuel Delivery, Engine Issues, Towing Services. After they have selected the required services require which they need they could pin their location and request for a mechanic this request will be updated in the database. Once a mechanic accepts the request the request The Customer location will be updated on the mechanic end as well as the mechanic location will be updated in the customer end this will be done using the Geo Fire feature which updates the firebase with the longitude and latitude. The location will be displayed on the device with the use of Google Maps API and GPS. Furthermore, a customer could request mechanics to be a favorite which will be saved in the database. Using the time taken by mechanics to come to their location customers could take a game quiz to increase their knowledge on how to maintain vehicles and service them according to how they score they will receive points.

Furthermore, other than the customer and the mechanic there will be a system admin which will be using a web-based system and will receive all the registrations made by mechanics to a separate page. Here the admin could verify the mechanic and grant him access to use the app. There will be a dashboard using BI tools where the system admin could monitor the progress and various details about the application.

III. SYSTEM ARCHITECTURE

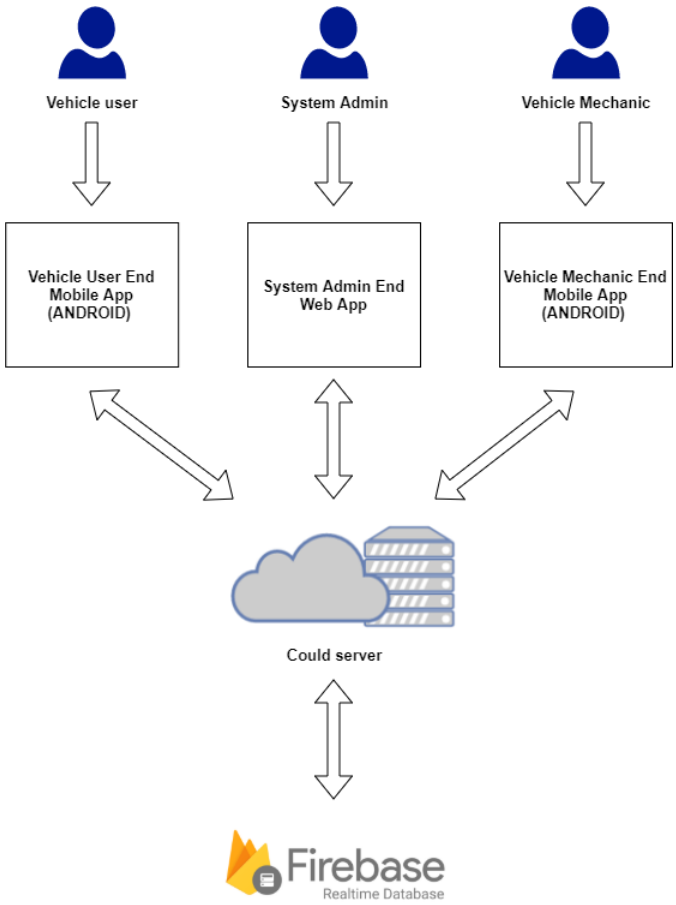

Fig. 1. System Architecture of the whole system

The mechanic and the user will be using an android platform for their system and this will share the same database using firebase. All the registration details, location info. Etc. could be easily share across the whole platform using a single firebase project and multiple applications embedded in the project. Accordingly, the system admin too could refer the same database and access all the data. Google analytics are embedded in the firebase itself which will be helpful in providing stats for the admin.

IV. SYSTEM OVERVIEW

A. Mechanic Section

The Mechanic section will be separate application for the mechanic to be installed on his device. This separate device will be linked with the common Firebase project.

1. *Login or Register*

The First thing a mechanic will be directed to once the application is loaded is the login page. If the mechanic already has an account, he could just simply login to the system, if he

does not have an account, he could register for a new account by selecting the services he wishes to provide. Once the user registers He will not receive any requests from customers since he can't start working. This will be enabled only after the admin has given access to the mechanic to use the app. This process is handled by setting a default flag in the DB Authorization as False. Once the mechanic has registered or logged into the system the user does not need to login every time, he opens the app. It will identify that the use has already logged in and direct to the mechanic's working page.

```
mAuth = FirebaseAuth.getInstance();

firebaseAuthListner = new FirebaseAuth.AuthStateListener() {
    @Override
    public void onAuthStateChanged(@NonNull FirebaseAuth
firebaseAuth) {
        FirebaseUser user =
FirebaseAuth.getInstance().getCurrentUser();
        if (user!=null){
            Intent intent = new
Intent(MechanicRegisterActivity.this,
MechanicMapActivity.class);
            startActivity(intent);
            finish();
            return;
        }
    }
};

mAuth.createUserWithEmailAndPassword(email,
password).addOnCompleteListener(MechanicRegisterActivity.thi
s, new OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(@NonNull Task<AuthResult> task) {
        if(!task.isSuccessful()){

Toast.makeText(MechanicRegisterActivity.this,"sign up
error", Toast.LENGTH_LONG).show();
        }
        else{
            user_id = mAuth.getCurrentUser().getUid();
            DatabaseReference current_user_db =
FirebaseDatabase.getInstance().getReference().child("Users")
.child("Mechanics").child(user_id);
            current_user_db.setValue(true);
            DatabaseReference userName =
FirebaseDatabase.getInstance().getReference().child("Users")
.child("Mechanics").child(user_id).child("Name");
            userName.setValue(name);
            DatabaseReference number =
FirebaseDatabase.getInstance().getReference().child("Users")
.child("Mechanics").child(user_id).child("Phone");
            number.setValue(phone);

            setMechanicServices();

            DatabaseReference auth =
FirebaseDatabase.getInstance().getReference().child("Users")
.child("Mechanics").child(user_id).child("Authorization");
            auth.setValue(false);

            DatabaseReference working =
FirebaseDatabase.getInstance().getReference().child("Users")
.child("Mechanics").child(user_id).child("WorkingState");
            working.setValue(false);

            DatabaseReference customerAccept =
FirebaseDatabase.getInstance().getReference().child("Users")
.child("Mechanics").child(user_id).child("CustomerAccepted")
;
            customerAccept.setValue(false);
        }
```

2. *Mechanic could start working*

Once the Mechanic has Logged into the system, he could set his status as working or stopped working. This is a flag which is set in the database and on the click of the button the flag will be set to true or false. Once it is set to true the firebase "onDataChange()" method will be triggered and it will find if any customer has requested for a service which the mechanic has subscribed for. If there is a customer found requesting for a service, the request would popup on the screen Once the mechanic accepts the request there is a flag for the mechanic which is CustomerAccepted in the database which will be set to true this will prevent from customer requesting a mechanic who has already accepted a request.

```
userID =
FirebaseAuth.getInstance().getCurrentUser().getUid();
DatabaseReference workingRef =
FirebaseDatabase.getInstance().getReference().child("Users")
.child("Mechanics").child(userID);
workingRef.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {
        if (dataSnapshot.exists()){
            Map<String, Object> map = (Map <String, Object>)
dataSnapshot.getValue();
            workingState =
map.get("WorkingState").toString();
            authorization =
map.get("Authorization").toString();
            startWorking();
        }
    }
```

3. *Mechanic Map*

The mechanic map will identify the current location of the mechanic and using Google Maps API and the GeoFire function of Firebase the data will be updated to a child in the database called MechanicAvailable and this child of the DB will store the longitude and the altitude of the Mechanic. This location will be used by the customer to locate the closest mechanic to send the request. Once the mechanic has accepted the request this will be deleted from the MechanicAvaialble child and sent to a new child called MecahanicWorking so that other customers will not be able to see the Mechanic when requesting for a service. After a service is completed the customer could rate the mechanic on the service provided.

```
LocationCallback mLocationCallback = new LocationCallback(){
    @Override
    public void onLocationResult(LocationResult
locationResult) {

        connectDriver();
        for (Location location :
locationResult.getLocations()){
            mLastLocation = location;
```

```
            LatLng latlng = new
LatLng(location.getLatitude(),location.getLongitude());

mMap.moveCamera(CameraUpdateFactory.newLatLng(latlng));

mMap.animateCamera(CameraUpdateFactory.zoomTo(15));


            String userID =
FirebaseAuth.getInstance().getCurrentUser().getUid();
            DatabaseReference availableRef =
FirebaseDatabase.getInstance().getReference("MechanicAvailab
le");
            DatabaseReference workingRef =
FirebaseDatabase.getInstance().getReference("MechanicsWorkin
g");

            GeoFire geoFireAvailable = new
GeoFire(availableRef);
            GeoFire geoFireWorking = new
GeoFire(workingRef);

            switch (customerId){
                case "":
                    geoFireWorking.removeLocation(userID,
new GeoFire.CompletionListener() {
                        @Override
                        public void onComplete(String key,
DatabaseError error) {}
                    });
                    geoFireAvailable.setLocation(userID, new
GeoLocation(location.getLatitude(),
location.getLongitude()), new GeoFire.CompletionListener(){
                        @Override
                        public void onComplete(String key,
DatabaseError error) { }
                    });
                    break;

                default:
                    if(customerRequestAccepted) {

geoFireAvailable.removeLocation(userID, new
GeoFire.CompletionListener() {
                            @Override
                            public void onComplete(String
key, DatabaseError error) {}
                        });
                        geoFireWorking.setLocation(userID,
new GeoLocation(location.getLatitude(),
location.getLongitude()), new GeoFire.CompletionListener() {
                            @Override
                            public void onComplete(String
key, DatabaseError error) {}
                        });
                    }
                    if (!customerRequestAccepted){

geoFireWorking.removeLocation(userID, new
GeoFire.CompletionListener() {
                            @Override
                            public void onComplete(String
key, DatabaseError error) {}
                        });
                        geoFireAvailable.setLocation(userID,
new GeoLocation(location.getLatitude(),
location.getLongitude()), new GeoFire.CompletionListener(){
                            @Override
                            public void onComplete(String
key, DatabaseError error) { }
                        });
                    }
```

4. *Mechanic Favorite list*

The mechanic favorite list is formed by requests sent by customers sending requests for mechanics once the mechanic accepts the request it will be saved child as FavoriteCustomers the mechanic could always view the list of customers and if needed remove any when he requires. If the mechanics state is as working the customer could straightaway request a mechanic to come and assist the customer.

5. *Mechanic Dashboard*

The Mechanic dashboard will display important details which the mechanic will need to identify his progress on how he has performed during the past services he has provided. This will include details such as total number of services, total earnings, most provided service, most demanding area, etc. all these details will be filterable to the mechanics desire.

B. Customer Section

The Customer section will be separate application for the customers to installed. This will be linked with the common Firebase project to be shared with the Mechanics and Admin.

1. *Login or Register*

The First time a customer opens the application the login page is displayed. If the customer already has an account, the user could simply login to the system, if not the user could register for a new account. Once the mechanic has registered or logged into the system the user does not need to login every time, he opens the app. Same as the Mechanic it will identify that the use has already logged in and direct to the request services page of the customer.

```
mAuth.createUserWithEmailAndPassword(email,
password).addOnCompleteListener(CustomerRegisterActivity.thi
s, new OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(@NonNull Task<AuthResult> task) {
        if(!task.isSuccessful()){

Toast.makeText(CustomerRegisterActivity.this,"sign up
error", Toast.LENGTH_SHORT).show();
        }
        else{
            String user_id =
mAuth.getCurrentUser().getUid();
            DatabaseReference current_user_db =
FirebaseDatabase.getInstance().getReference().child("Users")
.child("Customers").child(user_id);
            current_user_db.setValue(true);
            DatabaseReference userName =
FirebaseDatabase.getInstance().getReference().child("Users")
.child("Customers").child(user_id).child("Name");
            userName.setValue(name);
            DatabaseReference number =
FirebaseDatabase.getInstance().getReference().child("Users")
.child("Customers").child(user_id).child("Phone");
            number.setValue(phone);
        }
    }
});
```

2. *Select Services to be requested*

Once a customer needs a service, the user could select the service or multiple services if needed and add any special notes such as the severity or any special requirements through this. Once these details are completed the customer could proceed to the next part of the request which is selecting the location and requesting the mechanic to come and assist.

```
    String userId = FirebaseAuth.getInstance().getUid();

    DatabaseReference tireChange =
FirebaseDatabase.getInstance().getReference("Users").child("
Customers").child(userId).child("Services").child("TireChang
e");
    if (mTire.isChecked())
        tireChange.setValue(true);
    else
        tireChange.setValue(false);

    DatabaseReference batteryChange =
FirebaseDatabase.getInstance().getReference("Users").child("
Customers").child(userId).child("Services").child("BatteryJu
mpStart");
    if (mBattery.isChecked())
        batteryChange.setValue(true);
    else
        batteryChange.setValue(false);

    DatabaseReference lockSmithService =
FirebaseDatabase.getInstance().getReference("Users").child("
Customers").child(userId).child("Services").child("LockSmith
Service");
    if (mLockSmith.isChecked())
        lockSmithService.setValue(true);
    else
        lockSmithService.setValue(false);

    DatabaseReference FuelDelivery =
FirebaseDatabase.getInstance().getReference("Users").child("
Customers").child(userId).child("Services").child("FuelDeliv
ery");
    if (mFuel.isChecked())
        FuelDelivery.setValue(true);
    else
        FuelDelivery.setValue(false);

    DatabaseReference EngineIssues =
FirebaseDatabase.getInstance().getReference("Users").child("
Customers").child(userId).child("Services").child("EngineIss
ues");
    if (mEngine.isChecked())
        EngineIssues.setValue(true);
    else
        EngineIssues.setValue(false);

    DatabaseReference towingServices =
FirebaseDatabase.getInstance().getReference("Users").child("
Customers").child(userId).child("Services").child("TowingSer
vices");
    if (mTowing.isChecked())
        towingServices.setValue(true);
    else
        towingServices.setValue(false);
```

3. *Customer Map*

Once the customer selects the services needed the next step is using Google Maps API, pining the location and sending the request to the mechanic to accept the request. The request of the customers location will be updated as a child in the DB as CustomerRequest. This location will be able to be displayed by the mechanic when he accepts the request on his map. Once the mechanic accepts the request and after the mechanic location is shifted to the child MecahanicWorking the customer will be able to see the mechanics location on application map. The mechanism used to find the closest mechanic will be used by the Geo Location quarries. Here in order to send the request to the closest mechanic there are few constraints. It will use the event listener “addGeoQueryEventListener()” and the method “onKeyEntered()” which will be triggered once a mechanic is found this by using a radius of 1 which will increment by 1 each and every time a mechanic is not found. This will consider the closest mechanic, if the mechanic is working and that he has not accepted any other customers request. After a service is completed the customer could rate the mechanic on the service provided.

```
private int radius = 1;
private Boolean mechanicFound = false;
private String mechanicFoundID;

private void getClosestMechanic() {

    DatabaseReference mechanicLocation =
FirebaseDatabase.getInstance().getReference().child("Mechani
cAvailable");

    GeoFire geoFire = new GeoFire(mechanicLocation);
    GeoQuery geoQuery = geoFire.queryAtLocation(new
GeoLocation(mPickupLocation.latitude,
mPickupLocation.longitude), radius);
    geoQuery.removeAllListeners();

    geoQuery.addGeoQueryEventListener(new
GeoQueryEventListener() {
        @Override
        public void onKeyEntered(String key, GeoLocation
location) {
            if
(!mechanicFound&&validateMechanicService(key)&&mechanicWorki
ng(key)) {
                mechanicFound = true;
                mechanicFoundID = key;

                DatabaseReference mechanicRef =
FirebaseDatabase.getInstance().getReference().child("Users")
.child("Mechanics").child(mechanicFoundID);
                HashMap map = new HashMap();
                map.put("CustomerID", userId);
                mechanicRef.updateChildren(map);

                services = serviceDetails.dataSnapshot;
                if
(Boolean.parseBoolean(services.child("Mechanics").child(key)
.child("CustomerAccepted").getValue().toString())) {

                    getMechanicLocation();
                    mRequest.setText("Looking for a
Mechanic");
                }
                else
                    getClosestMechanic();
            }
        }
```

```
        @Override
        public void onKeyExited(String key) {   }

        @Override
        public void onKeyMoved(String key, GeoLocation
location) { }

        @Override
        public void onGeoQueryReady() {
            if (!mechanicFound){
                radius++;
                getClosestMechanic();
            }
        }

        @Override
        public void onGeoQueryError(DatabaseError error) {
}
    });
}
```

4. Customer Favorites list

The customer favorite list is formed by requesting mechanics to become of their favorites. If the mechanic accepts it will be saved child in the customer's section as FavoriteMechanics and as FavoriteCustomers in the mechanic's section. The customer could always view the list of mechanics and if needed remove any when it is required. If the mechanics state is as working the customer could straightaway request a mechanic.

5. Customer Game and points

The customer could play a game which will be a quiz regarding vehicle servicing and knowledge on vehicles. The customer could play this during the time taken to the mechanic to come to the location of the breakdown. The points earned by the game will be added to the child points under customer of the DB. These points could be later redeemable by the customer when using services later.

C. System Admin Section

The system admin will be able to monitor all the activities of the mechanics and the customers. The mechanic will have specific page to see all the registered mechanics, once the authentication process of the mechanic is complete the admin could grant access for the mechanic to start providing services. The system admin could furthermore view all the statistics of the application such as user count, total revenue, total services provided, most provided service and many more by using a BI tool which is power BI.

V. OUTPUT

A. Mechanic Section

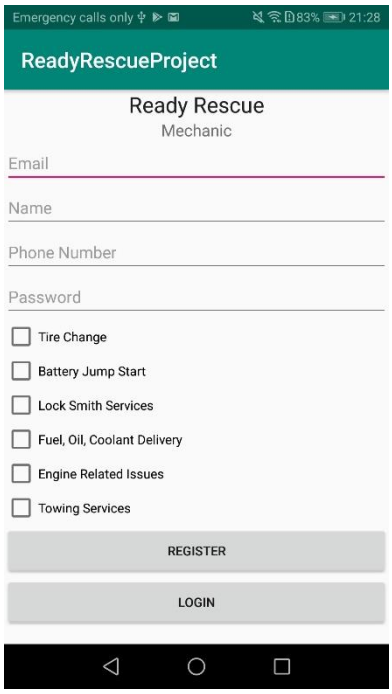

Fig. 2. Registration page of Mechanic

Fig. 3. Start and stop working page of Mechanic

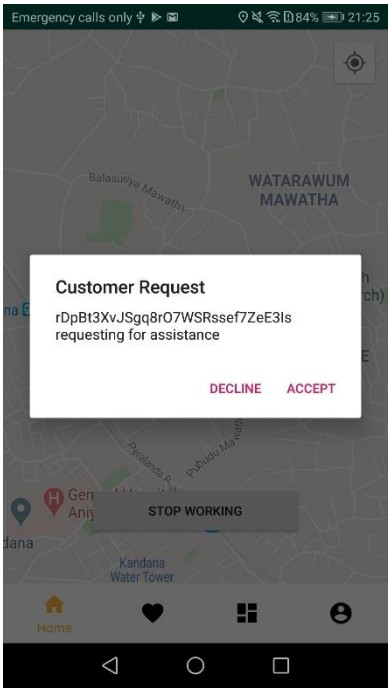

Fig. 4. Customer Request Pop Up for Mechanic

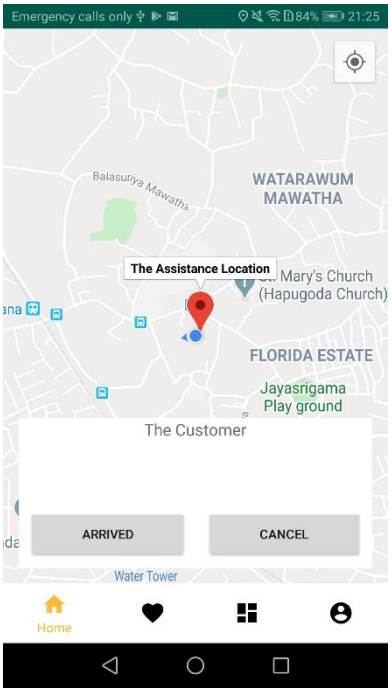

Fig. 5. Customer location on Mechanic Map

B. Customer Section

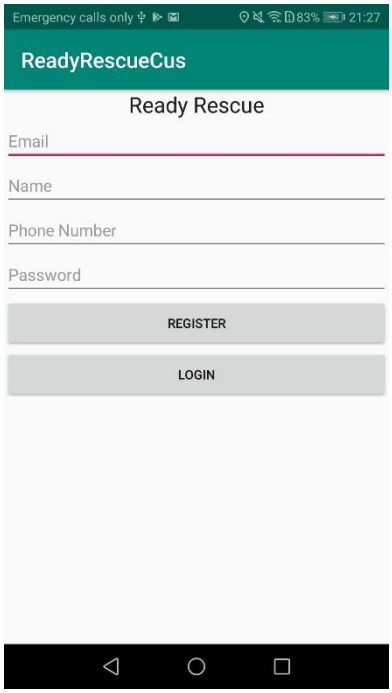

Fig. 6. Customer Registration page

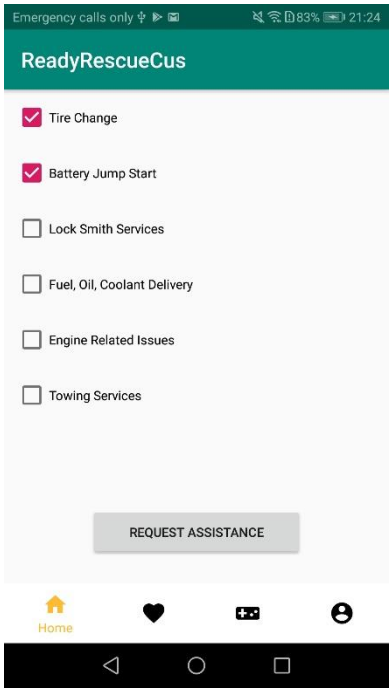

Fig. 7. Customer Service Selection page

Fig. 8. Customer Confirm request location page

Fig. 8. Customer Confirm request location page

VI. FUTURE SCOPE

For future implementations a recombination and suggestion system will be implemented where if the system identifies that a user is doing the same repair for a multiple time within a short period of time, it will suggest the user to give more attention to the issue. Another improvement will be where the system will identify the area which the user travels the most and recommend mechanics to be added to their favorite list.

REFERENCES

- Bhatt, P., Gupta, S., Singh, P., & Dhiman, P. (2017). *Accident and Road Quality Assessment using.* IEEE.
- Chowdhury, A., Banerjee, T., Chakravarty, T., & Balamuralidhar, P. (2015). *Smartphone Based Sensing Enables Automated.* IEEE.
- Communit, A. e. (2017). *What are the main problems of mobile apps today.* androiddeveloper.
- Department of Census and Statistics. (2017). *Labour Demand Survey.* Ministry of National Policies and Economic Affaires.
- JMango. (2019). *Mobile App versus Mobile Website Statistics: 2018 and beyond.*
- KAI, O. J. (2019). *DEVELOPING A REAL TIME DIGITAL EMERGENCY PERSONAL SAFETY APP TO ASSIST IN EMERGENCY SITUATIONS USING BEST TIME.* Kampar: Universiti Tunku Abdul Rahman.
- Khanapuri, A., D'souza, G., D'souza, S., & Shastri, A. (2015). *On Road: A car assistant application.* IEEE.
- Kumar, K., Bose, J., & Kumar, S. (2017). *A Generic Visualization Framework based on a Data.* IEEE.
- Mahajan, S., Parekh, M., Patel, H., & Patil, S. (2017). BRB Dashboard: A Web-based Statistical Dashboard.
- Mullet, J. (2019). *Goodyear As-A-Service Project Report.* Williams Honors College, Honors Research.
- Philip, J., Nayak, S., Patel, S., & Devashrayee, Y. (2018). *Mobile Mechanic – An innovative step towards Digital Automobile Service.* IEEE.
- Programiz. (2019). *Learn Java (Introduction and Tutorials to Java Programming).*
- Ramamurthy, K., Singh, M., Davis, M., Kevern, J., Klein, U., & Peran, M. (2015). *Identifying Employees for Re-Skilling using an Analytics-Based Approach.* IEEE.
- Traffic, Department of Motor. (2018). *Number of motor vehicles by type.* Department of Census and Statistics.
- Wijesiri, L. (2020). *Web-based taxi services: Regulation needed.* Daily News.