TRIBHUWAN UNIVERSITY

Institute of Science and Technology

"Mechanic Finder"

A Project Report

Submitted to:

Office of the Dean

Institute of Science and Technology, Tribhuwan University

Kirtipur, Nepal

In the partial fulfillment of the requirement for the bachelor's Degree in computer

Science and Information Technology

Submitted By:

Bamdev K.C. (TU Roll No. 10903)

Subash Nepal (TU Roll No.10933)

Sudarshan Shrestha (TU Roll No. 10934)

Ujwal Pandey (TU Roll No.10940)

ACKNOWLEDGEMENT

We would like to express our deepest appreciation to all those who provided us the possibility to complete this project. However, it would not have been possible without the help and support from our college administration and our supervisor. We would like to extend our sincere thanks to all of them.

We are highly indebted to our supervisor and coordinator Mr. Chakra Rawal Sir for his valuable suggestion and constant supervision to help us take our entire project forward. His guidance was huge support for our project.

Likewise, we are thankful towards Mr. Surya Bam Sir, Principal Er. Anil Lal Amatya Sir and all the faculty members including our teachers who have helped us with our project.

Last but not least, we would like to thank all our friends who have directly or indirectly helped us in our project.

Thanking you,

Bamdev K.C. (TU Roll No. 10903)

Subash Nepal (TU Roll No.10933)

Sudarshan Shrestha (TU Roll No. 10934)

Ujwal Pandey (TU Roll No.10940)

ABSTRACT

Mechanic Finder is an android application for On Road Vehicle Breakdown Assistance that can search a mechanic basis on the user’s location. On-road vehicle breakdown assistance is designed for the user to get quick service at the event on any vehicle breakdown. Many android users can access this application. This application will help to reduce wasting user time for finding a proper mechanic.

The proposed system developed is an Android based application which can run on any of the compatible Android devices, be it a Tablet or a Smartphone. The app will enable the vehicle owner to search and communicate with any service center’s mechanic. The user gets access to the mechanic’s location and the mechanic in return gets access to the user’s location, through the application using the Google map facility thereby saving a lot of time in such an complicated situation Henceforth, the proposed system goes hand in hand with the new age technology and characterizes – user friendliness, in formativeness and time saving.

Keywords: *mobile, android, mechanic, admin, user, vehicle breakdown, vehicle assistance*

Table of Contents

List of Figures

List of Tables

List of Abbreviation

CFD	Context Flow Diagram
DFD	Data Flow Diagram
ER	Entity-Relationship
FR	Functional Requirement
IEEE	Institute of Electrical and Electronics Engineers
NFR	Non- Functional Requirement
USD	Use Case Diagram

CHAPTER ONE

INTRODUCTION

1.1 Introduction

Travelling along the highways is some experience every one of us will experience and cherish in our lifetime. We might also plan and make all the necessary precautions to make the journey safe and smooth. However, in the unfortunate & unforeseen event of a breakdown or road accidents what is we need is an immediate solution so as to avoid unnecessary complications present with being stranded in the middle of the road. Our goal is to ensure that one gets the prompt assistance they deserve during such a situation [1]. To make this possible we develop an application called Mechanic Finder.

Mechanic Finder is an application that helps to find mechanics easily and quickly when a user's vehicle breaks down on a certain journey or travel. Vehicle breakdowns are a daily occurrence. Whenever such happens, the services of Mechanics are vital through which roadside assistance is possible. This system helps to overcome vehicle breakdown by providing mechanic details in one click. Here the locator allows users to search mechanics from different locations. This online mechanic locator reduces work and can easily find the mechanics from various nearest locations.

A vehicle breakdown is the mechanical failure of a motor vehicle in such a way that the underlying problem prevents the vehicle from being operated at all, or impedes the vehicle's operation so much, that it is very difficult, nearly impossible, or else dangerous to operate [2]. Vehicle breakdowns have various causes. Depending on the nature of the problem, the vehicle may or may not need to be towed to an automobile repair shop.

Roadside assistance is the attempt by a service professional to make minor mechanical repairs and adjustments in an effort to make a vehicle drivable again [3]. Roadside assistance covers many problems a user might experience on the roadway. The problems on vehicle which cannot be solved by the users can be solved with the help of the roadside assistance or the mechanic.

In the context of Nepal, with poor roads and unmanaged vehicles, the breakdown of a vehicle might take at any place. With the help of this system, any vehicle users can search for the mechanics near to their locations, through which the user’s vehicle can be repaired at the spot or on road with the help of the mechanics. Along with this, users can search for the type of mechanic they are searching for to repair their vehicle. For e.g. to repair the motorbike, a motorbike specialized mechanic can be requested by the users.

1.2 Problem Definition

Vehicle breakdowns are a daily event, it might occur due to various conditions and reasons for any kind or type of users. No matter how expensive, powerful the vehicle is, the vehicle will be affected and must be repaired. Currently, when a vehicle is broken down in the middle of the journey users seek the nearest garage and try to repair their vehicles but if there are no signs of any garage users often call the garage center which they knew and request for the mechanics. The arrival of the mechanics might take days considering various factors like distance, time, etc. The mechanics then arrived at the spot might not be of the quality the users are searching for. Along with this the mechanics might not be trustworthy and may not have good knowledge of the specific vehicle and they might further damage the vehicle which will create further problems for the user.

In general, whenever a vehicle is broken down at the middle way to the specific journey it is hard for users to search and call the mechanics that have got the capabilities to repair their vehicle at minimum time.

1.3 Objectives

The objective of this study is to build a model that can be used by users to easily find the mechanic. And, hence road assistance can be possible so that broken vehicles can be repaired quickly. The main objective can be listed as,

1. To design an application that can identify a suitable mechanic.

1.4 Scope

As of present, the scope of the application is only limited to android platform. If user have android phone and are using this application, user can find suitable mechanic within the few minutes based on their current location.

1.5 Limitation

Every system has demerits along with the merits. So, like every other system Mechanic Finder also has some limitations. The limitations can be listed as follows:

1. Since this is a mobile application users and mechanics need to be connected to internet in order to use it.
2. There is not a payment method in this Mechanic Finder app.
3. This application works only on android mobile.

1.6 Report Organization

The report is organized as follows: Chapter 1 explains the introduction, problem statement and proposed system, and scopes and limitations. Chapter 2 reviews existing system, requirement analysis and feasibility study of our project including use-case diagram. Chapter 3 specifies the system design of the project and describes the process model. Chapter 4 explains the system implementation. And finally, Chapter 5 discusses the conclusion

CHAPTER TWO

REQUIREMENT ANALYSIS AND FEASIBILITY ANALYSIS

2.1 Literature Review

Akhila V Khanapuri proposed that there has been an exponential increase in the number of vehicles on road, number of road accidents and vehicle breakdown cases recorded [4]. Finding effective ways to achieve maximum fuel efficiency without hampering the internal structure of these vehicles along with providing a response system to combat mishaps is a challenging task. In this paper, android application is proposed which monitors parameters like Engine RPM, fuel status, throttle position through an On board Diagnostics (OBD-II) being able to help amateur drivers with gear changing and provide assistance in case of vehicle breakdown. In these kinds of situation the need of a roadside assistance plays a vital role.

There are different web and mobile applications available at the internet that works for the Roadside Assistance. These applications generally provide insurance to the vehicle by providing various services or assistance to the vehicle if the vehicle is broken down during the journey. In most of these applications a membership card is to be generated to get the feature of the on road assistance.

IAE Roadside Assistance Service has built a web application to help thousands of customers to fix their vehicle problem on road over Nepal [5]. They provide an extensive list of services including: break down coverage, flat tires, recovery, journey continuation, towing, fuel delivery, battery replacement, lock up services and many more. IAE offers various membership plans to students and professionals.

The All State [6] application has shown that it offers fast, track able and reliable roadside service when a users vehicle has been broken down. This application provides service only when the users have a membership card and then it provides insurance to the vehicle.

The Car Insurance Comparison [7] application states that those users who routinely travel on difficult roads, old vehicles should considered using the roadside assistance. At any time the vehicles might be broken down and it will be hard for them to repair soon so using on road assistance is the best way to keep the vehicle repaired.

2.2 Requirement Analysis

Requirement analysis results in the specification of software’s operational characteristics indicates software’s interface with other system elements and establish constraints that soft- ware must meets. The requirements analysis task is a process of discovery, refinement, modeling and specification. The scope, initially established by us and refined during project planning, is refined in details. Model of the required data, information and control flow and operations behavior are created.

2.2.1 Functional Requirement

A Functional Requirement (FR) is a description of the service that the software must offer. It describes a software system or its component.

- The developed system should able to find the mechanic according to user’s location.
- User and Mechanic Registration should be valid.
- User should able to request the service for their vehicles.
- Mechanic should able to response the user request.

2.2.2 Use case Diagram

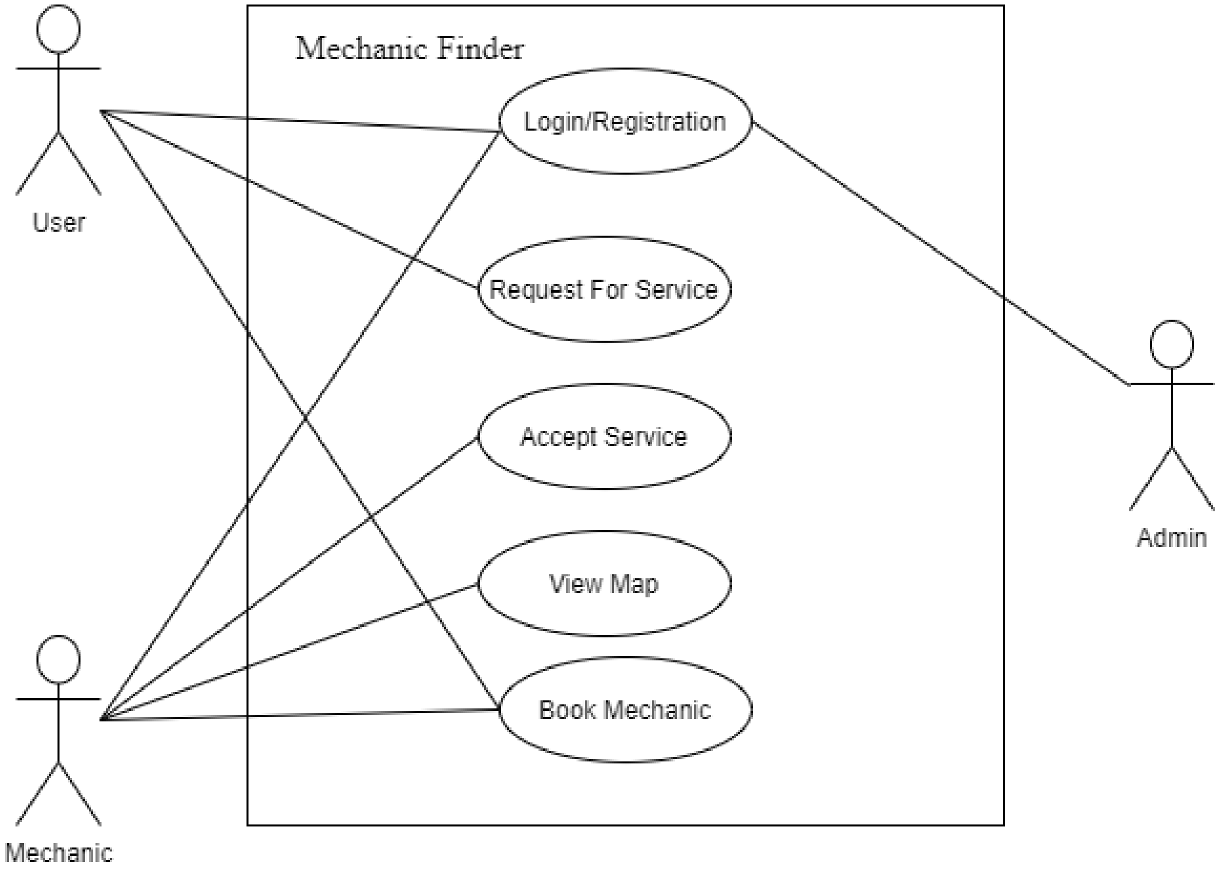


Fig 2.1: Use Case Diagram of Mechanic Finder

In Figure 2.1, the use case of the proposed system is shown and in the system, the user and the mechanic needs to register. User request for the service while mechanic accepts the service. Admin validates the users and mechanic registration as well as login.

2.2.3 Non-Functional requirement

Non-Functional Requirement (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. The non-functional requirements used in Mechanic Finder are as follows:

Performance: Performance defines how fast a software system or its particular piece responds to certain users' actions under certain workload. In this application Mechanics are fetched according to user's location to provide service to the users.

Usability: Usability defines how difficult it will be for a user to learn and operate the system. This system can be easily be operated by any user who knows how to operate the android phones.

Functionality: Functionality describes what function the system does. This software will deliver on the functional requirements.

Availability: Availability describes how likely the system is accessible for a user at a given point in time. This system will work only if users and mechanics registration are valid.

Portability: Portability defines how a system or its element can be launched on one environment or another. The system should work on all android mobiles.

2.3 Feasibility study

The feasibility study is the initial design stage of the project that indicates if the proposed project is possible or not. It determines if the proposed project is useful or not and defines the practicality of the project or a system. The study also determines if the system can be built efficiently on required time with the available resources. It gives answer to the following two questions. Will the project possess the required resources and technology and Will the project receive sufficiently high return on its investment?

There are various types of feasibility study we have done during the course of the project.

2.3.1 Technical Feasibility

Technical feasibility defines whether there exist required resources and technologies which will be used for project development.

All the tools and software product required to construct this project is easily available in the web. It do not require special environment to execute. It needs an IDE. All these aspects are easily affordable. The application can be accessed from anywhere at any time with an internet connection. Thus, it is technically feasible.

2.3.2 Operational Feasibility

Operational feasibility is a measure of how well a proposed system solves the problems, and how it satisfies the requirements identified in the requirements analysis phase of system development. It asks if the project is affordable, maintainable and reliable.

Mechanic Finder is an android based application. It is based on client-server architecture and needs the internet connection to access information. All of the technology required by the application are available and can access freely making it operationally feasible.

2.3.3 Economic Feasibility

Economic feasibility attempts to weigh the costs of developing and implementing a new system, against the benefits that would accrue from having the new system in place. The cost of developing the application is minimal. The cost associated while building the application was:

- Internet usage
- Development time

Since the technical requirements for our project were not costly, it required minimum amount to complete this project.

2.3.4 Schedule Feasibility

A project is feasible in term of schedule since no any process requires time longer than predicted. Figure below displays the time schedule for this project.

ID	Task Name	Duration	2020/21				
			Nov	Dec	Jan	Feb	Mar
1	Study and Analysis	2w	■	■	■		
2	Implementation	3w		■	■	■	
3	Testing and Analysis	2w				■	■
4	Documentation	3w				■	■
5	Review	1w					■
6	Presentation and Submission	1d					■

CHAPTER 3

SYSTEM DESIGN

3.1 System Architecture

The architecture consists of user and mechanic. Users and Mechanics both are registered to use the system. When users request a service mechanics are to response to Users. Mechanics then are provided the user's locations and they are able to find the path to the users using the Vincenty Algorithm.

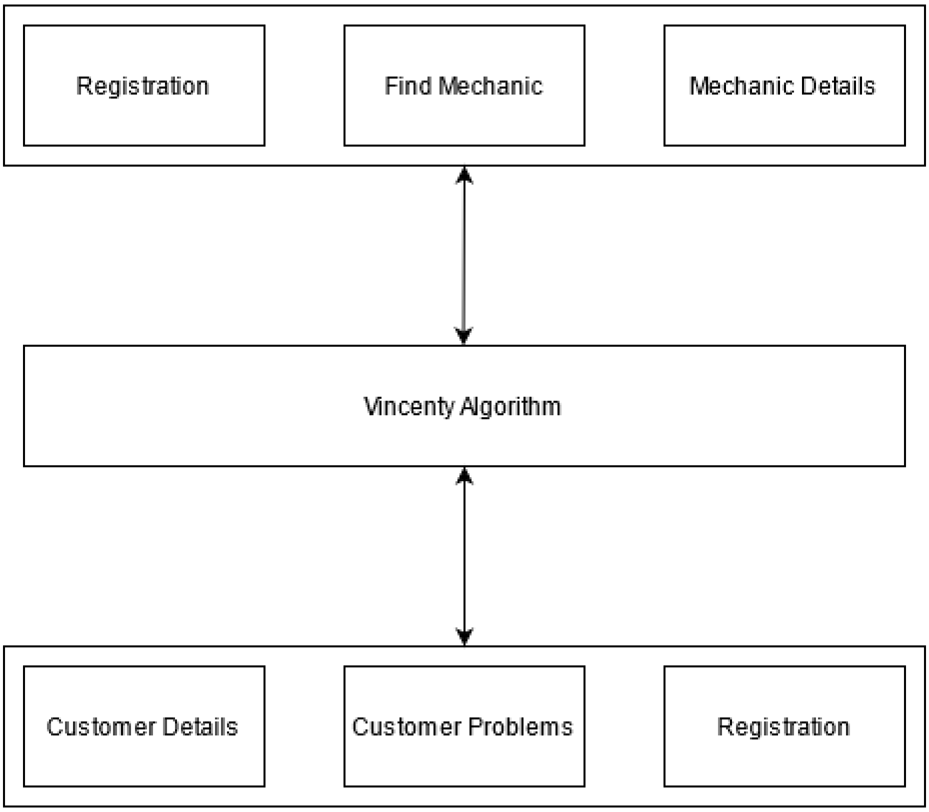


Fig 3.1: Architecture diagram of Mechanic Finder

3.2 Module

The application is divided into three modules.

3.2.1 Module 1

The first module in the application is the login. In the login process, the user and mechanic can input their email and password if he/she has already got an account; else the user should sign up by putting various forms.

3.2.2 Module 2

The second module in the application is the accessibility of the main features of the application after login. The user can view various services provided by the system and then user can request the service as they wanted to mechanic.

3.2.3 Module 3

The third module in the application allows mechanic to use user's location to find the path to reach out to users.

3.3 System Diagram

3.3.1 Entity Relationship Diagram

An Entity-Relationship (ER) Diagram is a type of flowchart that illustrates how "entities" such as people, objects or concepts relate to each other within a system.

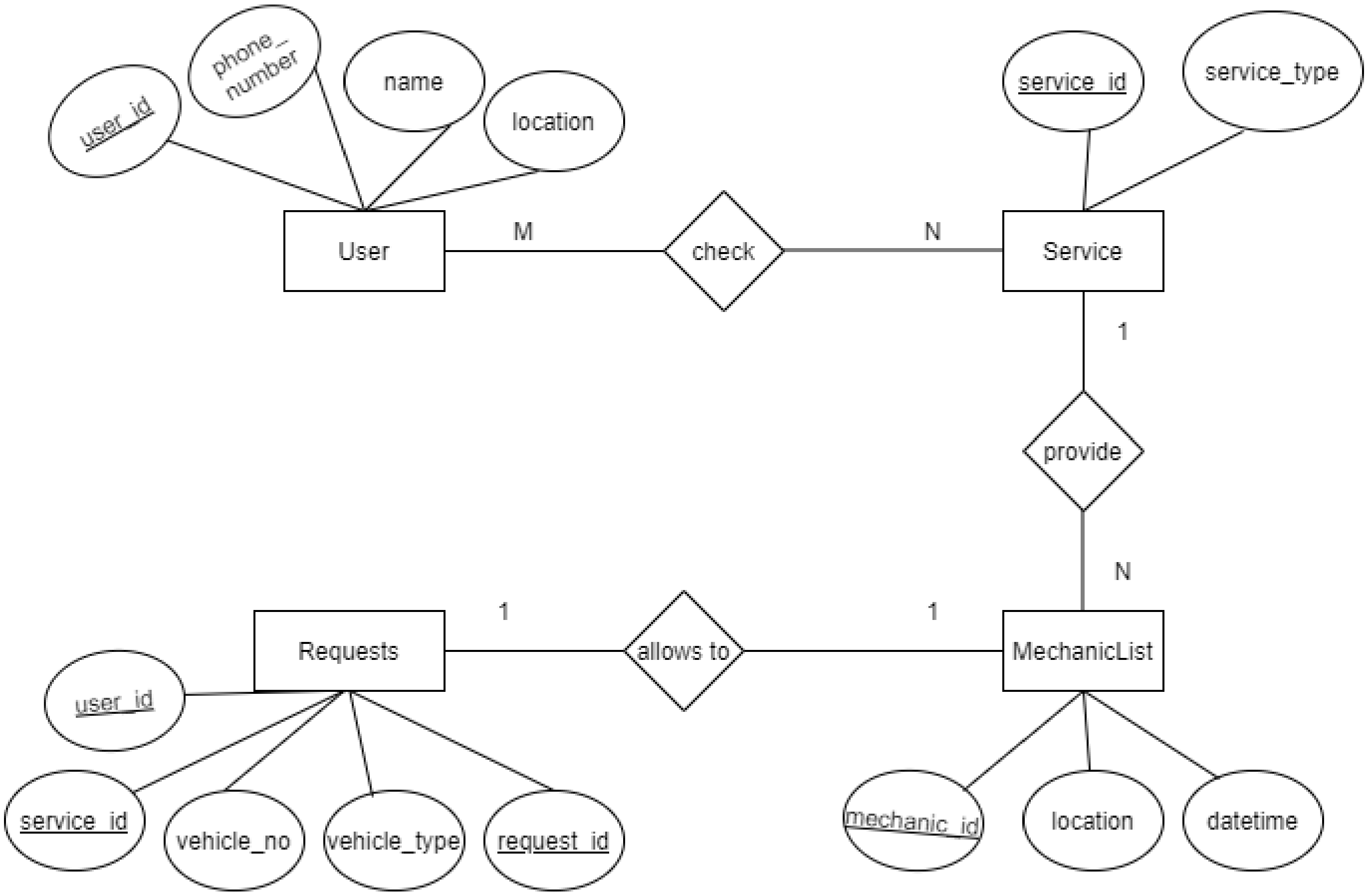


Fig 3.2: ER Diagram of User in Mechanic Finder

In the case of this project, there are two main entities; User and Mechanic. The User entity has various attributes like name, location, phone number. Users check the service available in the system. The service has the attribute service_ type. When a user selects one of the service type various mechanics are recommend to user. The mechanic list has the attribute of mechanic id, location and date time. Then after selecting the nearest mechanic user can send request which contains the attribute vehicle number and vehicle type. Then the request is send to the mechanic database.

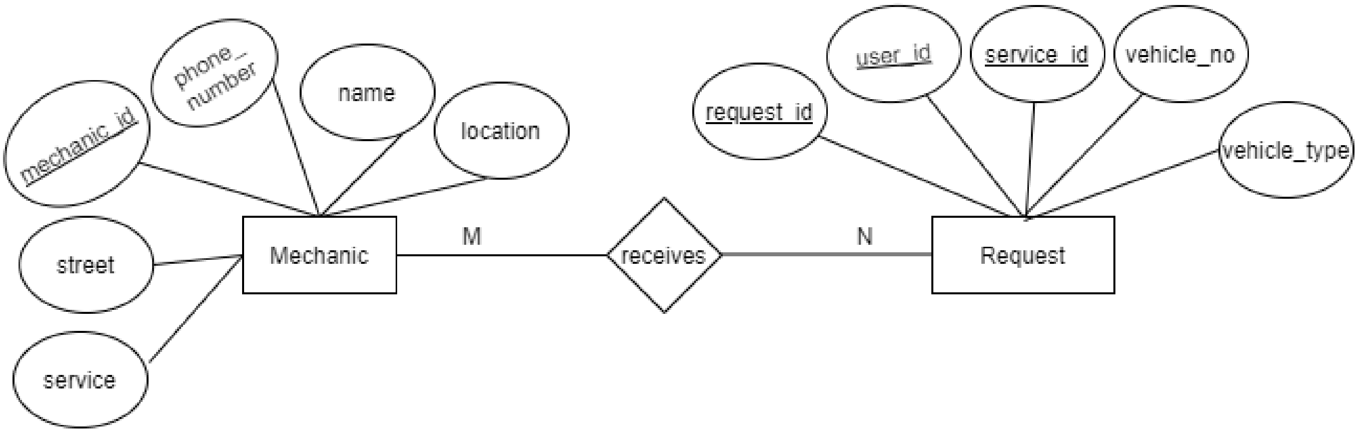


Fig 3.3: ER Diagram of Mechanic in Mechanic Finder

The Mechanic entity has attributes like name, mechanic id, phone number, service, location and street. Mechanics receives various requests send by the Users. Mechanic then can confirm or delete the request sent by the users.

3.3.2 Context Flow Diagram

The Context Flow Diagram (CFD) shows the system under consideration as a single high-level process and then shows the relationship that the system has with other external entities.

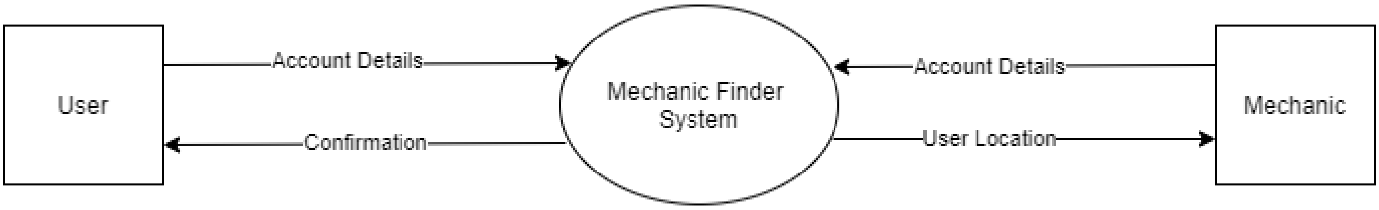


Fig 3.4: Context Flow Diagram of Mechanic Finder

Mechanic Finder System is the primary process which is connected to the External entities User and Mechanic. User and Mechanic provide their account details to the system. User’s location is provided to mechanic while users confirms for the mechanic.

3.3.3 Data Flow Diagram

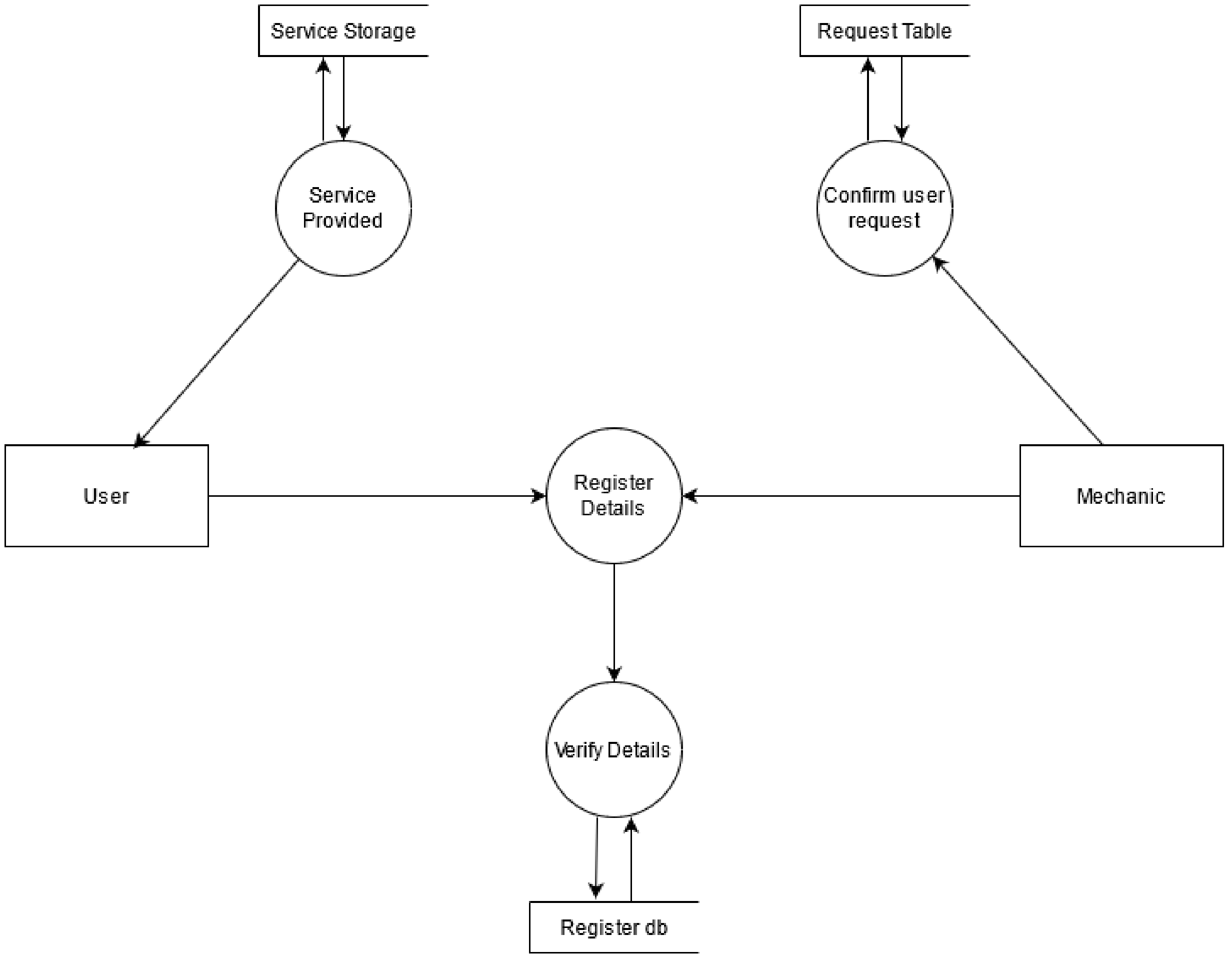


Fig 3.5: Level 1 DFD of Mechanic Finder

The level 1 data flow diagrams discusses about detail working principle of the application. User and Mechanic register details are stored on Database. User is provided the service included in mechanic finder. Mechanic confirms any request user had sent related to the service included in the app.

CHAPTER FOUR

IMPLEMENTATION

4.1 Implementation

The proposed system has two major divisions. One is the Mechanic portal and the other is the User/Customer Portal.

The mechanic portal allows the mechanic to initially register into the system and after logging in the application, the mechanic will be able to find any available customer looking for a service to be done. If the offer is accepted by a mechanic, location of the customer will be sent to mechanic and intimated to the customer also.

The Customer portal mainly focuses on recording customer complaints and posts it to nearby mechanics available in that locality and if the service request is accepted by the mechanic, routing will be carried over to the location.

The working starts from the customer's portal where a complaint or service request will be lodged for the vehicle and details of the service repair will be recorded and then stored. Furthermore, the details of the customer will be sent to the nearby mechanics in that locality. In the mechanic portal after initially registering into the application, a feed of customer complaints will be displayed. The proposed system will used the Vincenty formula to find the path between user and mechanic.

4.2 Algorithms Used

Vincenty formula is the main algorithm used in mechanic finder to map user and the mechanic given their latitude and longitude.

Vincenty formula was developed by Thaddeus Vincenty for calculating geodesic distances between a pair of latitude/longitude points on an ellipsoidal model of the Earth.

The formula uses an iterative method and assumes the Earth is an ellipsoid. This formula includes a direct and an inverse method:

- **Direct Method**: It computes the location of a point that is a given distance and angle from another point.
- **Inverse Method**: It computes the geographical distance and angle between two given points.

Distance/bearing between two points:

Given the coordinates of the two points (Φ_1, L_1) and (Φ_2, L_2), the inverse problem finds the angles α_1, α_2 and the ellipsoidal distance s.

Calculate U_1, U_2 and L, and set initial value of $\lambda = L$. Then iteratively evaluate the following equations until λ converges

$$\sin \sigma = \sqrt{[\,(\cos U_2 \cdot \sin\lambda)^2 + (\cos U_1 \cdot \sin U_2 - \sin U_1 \cdot \cos U_2 \cdot \cos \lambda)^2\,]}$$

$$\cos \sigma = \sin U_1 \cdot \sin U_2 + \cos U_1 \cdot \cos U_2 \cdot \cos \lambda$$

$$\sigma = \mathrm{atan}(\sin \sigma / \cos \sigma)$$

$$\sin \alpha = \cos U_1 \cdot \cos U_2 \cdot \sin \lambda / \sin \sigma$$

$$\cos^2 \alpha = 1 - \sin^2 \alpha$$

$$\cos 2\sigma_m = \cos \sigma - 2 \cdot \sin U_1 \cdot \sin U_2 / \cos^2 \alpha$$

$$C = f/16 \cdot \cos^2 \alpha \cdot [4 + \mathrm{f} \cdot (4 - 3 \cdot \cos^2 \alpha)]$$

$$\lambda = L + (1 - C) \cdot f \cdot \sin \alpha \cdot \{\sigma + C \cdot \sin \sigma \cdot [\cos 2\sigma_m + C \cdot \cos \sigma \cdot (-1 + 2 \cdot \cos^2 2\sigma_m)]\}$$

When λ has converged to the desired degree of accuracy

$$u^2 = \cos^2 \alpha \cdot (a^2 - b^2)/b^2$$

$$A = 1 + u^2/16384 \cdot \{4096 + u^2 \cdot [-768 + u^2 \cdot (320 - 175 \cdot u^2)]\}$$

$$B = u^2/1024 \cdot \{256 + u^2 \cdot [-128 + u^2 \cdot (74 - 47 \cdot u^2)]\}$$

$\Delta\sigma = B \cdot \sin\sigma \cdot \{\cos 2\sigma_m + B/4 \cdot [\cos\sigma \cdot (-1 + 2 \cdot \cos^2 2\sigma_m) - B / 6 \cdot \cos 2\sigma_m \cdot (-3 + 4 \cdot \sin^2 \sigma) \cdot (-3 + 4 \cdot \cos^2 2\sigma_m)]\}$

$s = b \cdot A \cdot (\sigma - \Delta\sigma)$

$\alpha_1 = \text{atan}(\cos U_2 \cdot \sin\lambda / \cos U_1 \cdot \sin U_2 - \sin U_1 \cdot \cos U_2 \cdot \cos\lambda)$

$\alpha_2 = \text{atan}(\cos U_1 \cdot \sin\lambda / -\sin U_1 \cdot \cos U_2 + \cos U_1 \cdot \sin U_2 \cdot \cos\lambda$ [8]

Where:

- a is length of semi-major axis of the ellipsoid (radius at equator)
- f is flattening of the ellipsoid
- U1,U2 are reduced latitude
- L1, L2 are longitude of the points
- Φ1, Φ2 are latitude of the points
- σ is angular separation between points
- σ1 is angular separation between the point and the equator
- s is the geodesic distance along the surface of the ellipsoid (in the same units as a & b)
- α1 is the initial bearing, or forward azimuth
- α2 is the final bearing

Between two nearly antipodal points, the iterative formula may fail to converge; this will occur when the first guess at λ as computed by the equation above is greater than π in absolute value.

The final value α1 and α2 will be the azimuth which then is used for mapping between the User and the mechanic.

4.3 Tools Used

We have used following tools in order to create our project.

Dart and Flutter

We have used Dart programming language for the development of this system. Dart is a programming language designed for client development, such as for the web and mobile apps. It is developed by Google and can also used to build server and desktop applications. Dart is the programming language used to code Flutter apps.

Flutter is a UI toolkit for building fast, beautiful, natively compiled applications for mobile, web, and desktop with one programming language and single codebase. It is free and open-source.

Visual Studio Code

We have used Visual Studio Code as our IDE. Visual Studio Code is a freeware source-code editor made by Microsoft which is a lightweight but powerful source code editor which runs on desktop and is available for Windows, macOS and Linux.

Visual Studio Code comes with built-in support for JavaScript, Typescript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).

Firebase

We have used Firebase as a database which is basically designed for mobile applications. Firebase is a Backend-as-a-Service, and it is a real-time database which is basically designed for mobile applications. Firebase is a platform from Google with powerful features for developing, handling, and enhancing applications.

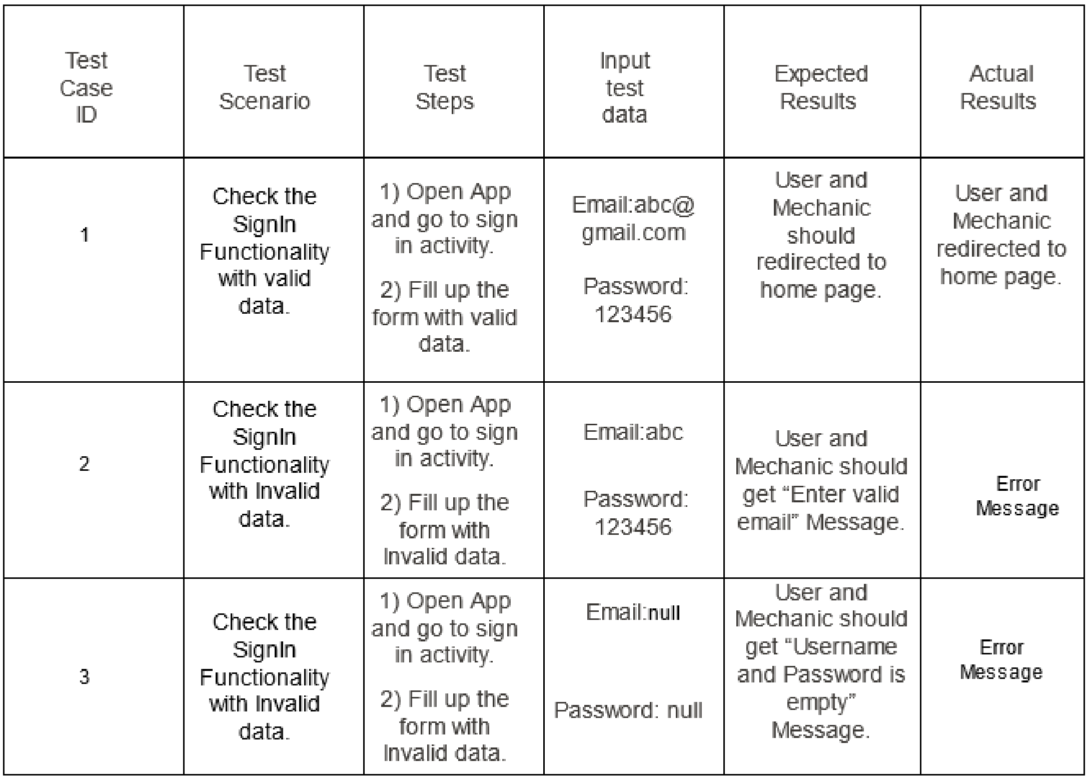
4.4 Testing

4.4.1 Unit Testing

Unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use. Unit test was done by feeding the system with several activities. The activity like login in, registration, send request, customer details, mechanic details were tested individually. All the units testing have been done using the ARD (android Real device).

Test case For Sign In

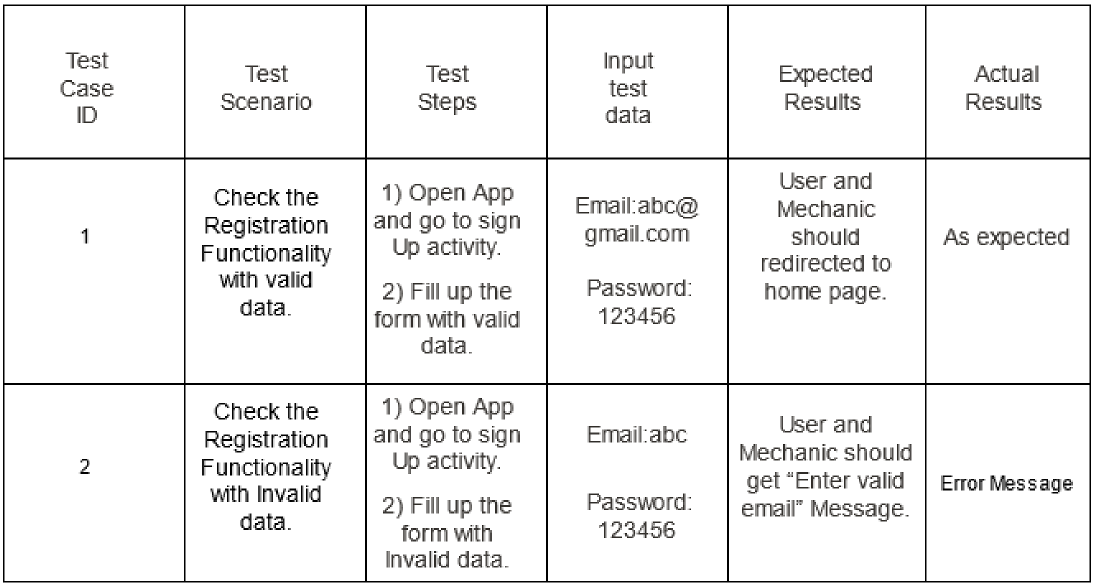
Table 1: Test case For Sign In



Test Case ID	Test Scenario	Test Steps	Input test data	Expected Results	Actual Results
1	Check the SignIn Functionality with valid data.	1) Open App and go to sign in activity. 2) Fill up the form with valid data.	Email:abc@ gmail.com Password: 123456	User and Mechanic should redirected to home page.	User and Mechanic redirected to home page.
2	Check the SignIn Functionality with Invalid data.	1) Open App and go to sign in activity. 2) Fill up the form with Invalid data.	Email:abc Password: 123456	User and Mechanic should get "Enter valid email" Message.	Error Message
3	Check the SignIn Functionality with Invalid data.	1) Open App and go to sign in activity. 2) Fill up the form with Invalid data.	Email:null Password: null	User and Mechanic should get "Username and Password is empty" Message.	Error Message

Test case For Registration

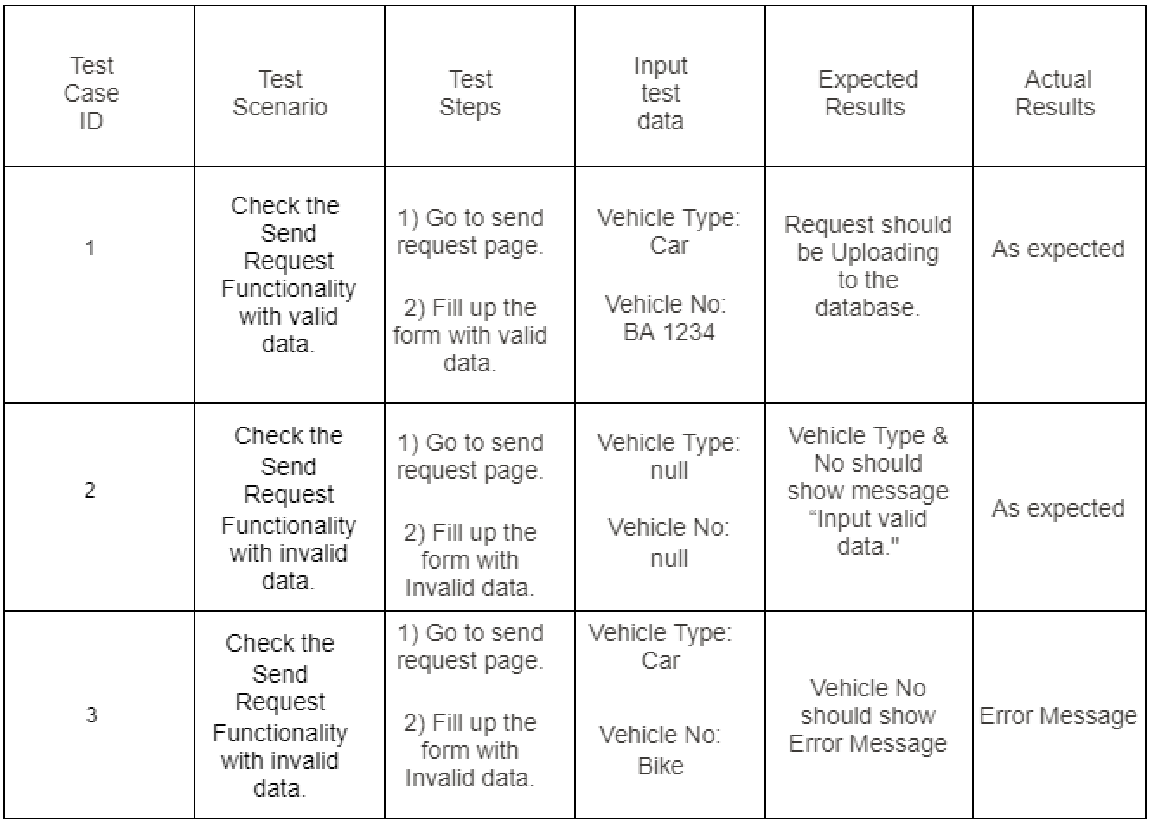
Table 2: Test case For Registration



Test Case ID	Test Scenario	Test Steps	Input test data	Expected Results	Actual Results
1	Check the Registration Functionality with valid data.	1) Open App and go to sign Up activity. 2) Fill up the form with valid data.	Email:abc@ gmail.com Password: 123456	User and Mechanic should redirected to home page.	As expected
2	Check the Registration Functionality with Invalid data.	1) Open App and go to sign Up activity. 2) Fill up the form with Invalid data.	Email:abc Password: 123456	User and Mechanic should get "Enter valid email" Message.	Error Message

Test case for Send Request

Table 3: Test case For Sending Request



Test Case ID	Test Scenario	Test Steps	Input test data	Expected Results	Actual Results
1	Check the Send Request Functionality with valid data.	1) Go to send request page. 2) Fill up the form with valid data.	Vehicle Type: Car Vehicle No: BA 1234	Request should be Uploading to the database.	As expected
2	Check the Send Request Functionality with invalid data.	1) Go to send request page. 2) Fill up the form with Invalid data.	Vehicle Type: null Vehicle No: null	Vehicle Type & No should show message "Input valid data."	As expected
3	Check the Send Request Functionality with invalid data.	1) Go to send request page. 2) Fill up the form with Invalid data.	Vehicle Type: Car Vehicle No: Bike	Vehicle No should show Error Message	Error Message

4.4.2 System Testing

The system testing part of a testing methodology involves testing the entire system for errors and bugs. This test is carried out by interfacing the hardware and software components of the entire system, and then testing it as a whole.

After testing individually, we tested the whole system for its functionality. The system testing has been performed by testing the whole application in real android device with the internet connection. The whole application runs smoothly and without error with the internet connection.

CHAPTER 5

CONCLUSION

5.1 Conclusion

While the chances of a properly maintained vehicle experiencing a breakdown are slim, it is never a possibility to predict when the user may experience a vehicular breakdown. The Android application developed here promises to make the life of a vehicle owner that much easier, as even in the probability of a breakdown, the vehicle owner is assured of the fact that he has a solution to the problem within a few steps of entering details in his smart phone and save himself from a major setback in such an undesirable situation.

The proposed system promises to act as a source of protection against the unpredictability of a vehicular breakdown and offers the owner of a particular some peace of mind in the event of operational failure of the user’s vehicle.

5.2 Future scope

Currently the users are requested to select those services which are available at the application. In future, users can request any service as they wanted. Similarly the application is now only available for android platform. In future, the application will also be provided to the ios platform.

REFERENCES

[1] B. S. S. S. H. Ajith Kumaar. D, "On Road Vehicle Service Finder," 4 2019. [Online]. Available: http://ijsrcseit.com/CSEIT1951137. [Accessed 6 March 2020].

[2] "Breakdown (vehicle)," [Online]. Available: https://desertoasisautorepair.com/. [Accessed 3 March 2020].

[3] "Roadside Assistance: What Is It, and Do I Need It?," [Online]. Available: https://www.valuepenguin.com/roadside-assistance-car-insurance. [Accessed 4 March 2020].

[4] A. &. S. &. D. &. D. Khanapuri, "on road: a car assistant application," 2015, p. 7.

[5] "Roadside Assistance Service," [Online]. Available: https://iaenepal.org/. [Accessed 18 02 2021].

[6] "Roadside Assistance," [Online]. Available: https://www.allstate.com/roadside-assistance.aspx. [Accessed 3 March 2020].

[7] "Breakdown Car Insurance," [Online]. Available: https://www.carinsurancecomparison.com/. [Accessed 4 March 2020].

[8] T. Vincenty, "Vincenty solutions of geodesics on the ellipsoid," [Online]. [Accessed 23 3 2021].

APPENDICES

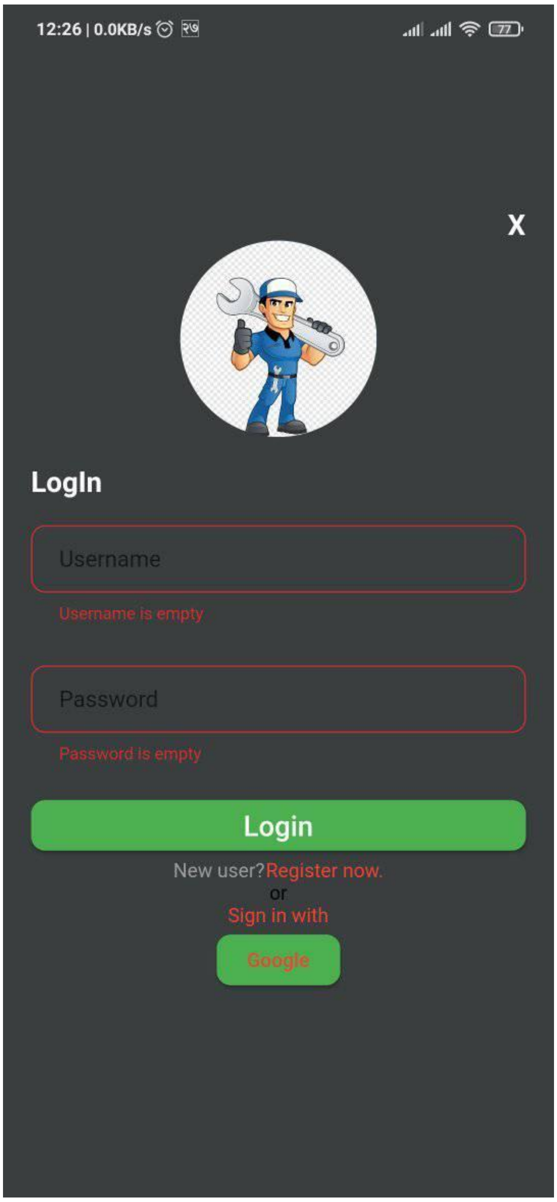
# Vincinety Algorithm Implementation

```
double getVincentyDistance(
     double lat1, double lon1, double lat2, double lon2) {
    double a = 6378137, b = 6356752.314245, f = 1 / 298.257223563;
    double L = 22 / 7 / 180 * (lon2 - lon1);
    double U1 = atan((1 - f) * tan(22 / 7 / 180 * (lat1)));
    double U2 = atan((1 - f) * tan(22 / 7 / 180 * (lat2)));
    double sinU1 = sin(U1), cosU1 = cos(U1);
    double sinU2 = sin(U2), cosU2 = cos(U2);
    double cosSqAlpha, sinSigma, cos2SigmaM, cosSigma, sigma;
    double lambda = L, lambdaP, iterLimit = 100;
    do {
      double sinLambda = sin(lambda), cosLambda = cos(lambda);
      sinSigma = sqrt((cosU2 * sinLambda) * (cosU2 * sinLambda) +
          (cosU1 * sinU2 - sinU1 * cosU2 * cosLambda) *
              (cosU1 * sinU2 - sinU1 * cosU2 * cosLambda));
      if (sinSigma == 0) return 0;
      cosSigma = sinU1 * sinU2 + cosU1 * cosU2 * cosLambda;
```

```
  sigma = atan2(sinSigma, cosSigma);
  double sinAlpha = cosU1 * cosU2 * sinLambda / sinSigma;
  cosSqAlpha = 1 - sinAlpha * sinAlpha;
  cos2SigmaM = cosSigma - 2 * sinU1 * sinU2 / cosSqAlpha;
  double C = f / 16 * cosSqAlpha * (4 + f * (4 - 3 * cosSqAlpha));
  lambdaP = lambda;
  lambda = L +
    (1 - C) *
      f *
      sinAlpha *
      (sigma +
        C *
          sinSigma *
          (cos2SigmaM +
            C * cosSigma * (-1 + 2 * cos2SigmaM * cos2SigmaM)));
} while ((lambda - lambdaP).abs() > 1e-12 && --iterLimit > 0);
if (iterLimit == 0) return 0;
double uSq = cosSqAlpha * (a * a - b * b) / (b * b);
double A = 1 + uSq / 16384 * (4096 + uSq * (-768 + uSq * (320 - 175 * uSq)));
```

```
  double B = uSq / 1024 * (256 + uSq * (-128 + uSq * (74 - 47 * uSq)));
  double deltaSigma = B *
    sinSigma *
    (cos2SigmaM +
      B /
        4 *
        (cosSigma * (-1 + 2 * cos2SigmaM * cos2SigmaM) -
          B /
            6 *
            cos2SigmaM *
            (-3 + 4 * sinSigma * sinSigma) *
            (-3 + 4 * cos2SigmaM * cos2SigmaM)));
  double s = b * A * (sigma - deltaSigma);
  return s;
}
```

#Screenshots



12:26 | 0.0KB/s

X

LogIn

Username

Username is empty

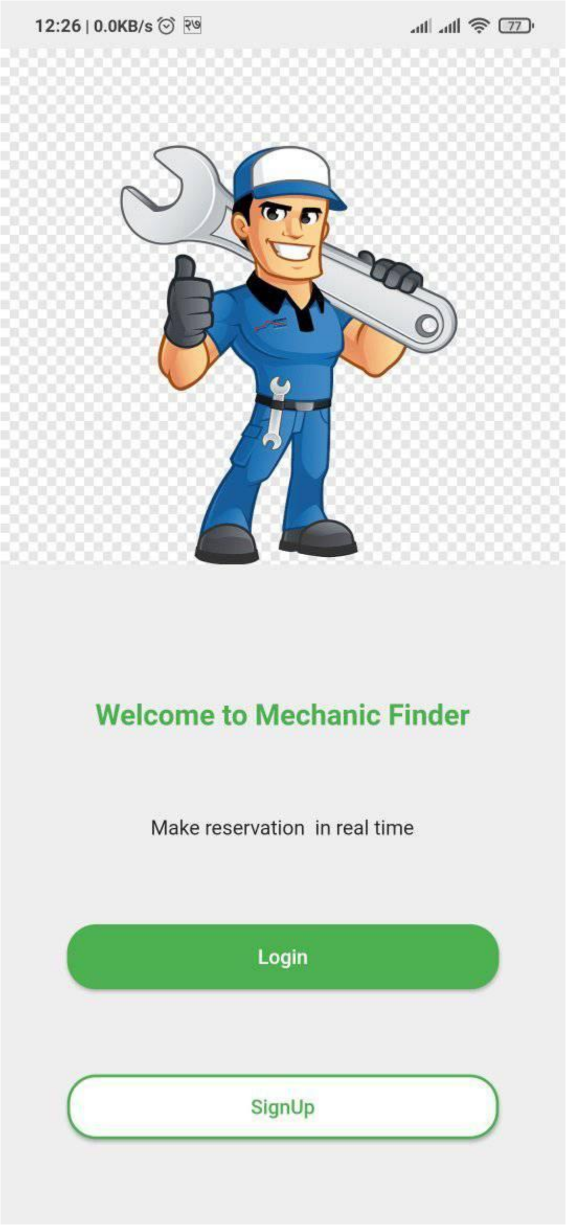
Password

Password is empty

Login

New user?Register now.
or
Sign in with

Google



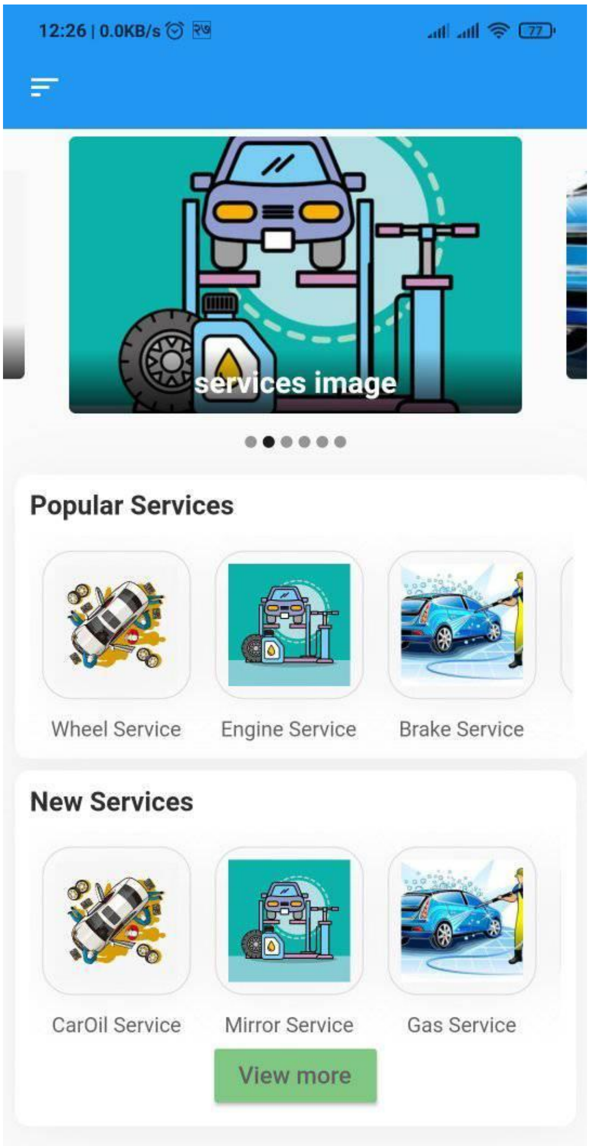
12:26 | 0.0KB/s

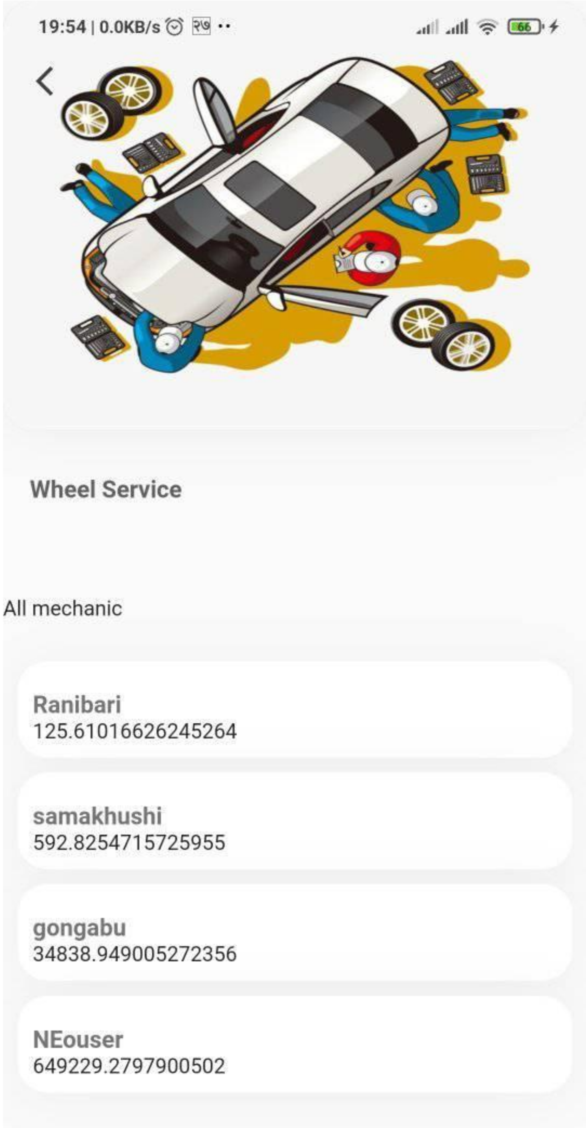
Welcome to Mechanic Finder

Make reservation in real time

Login

SignUp



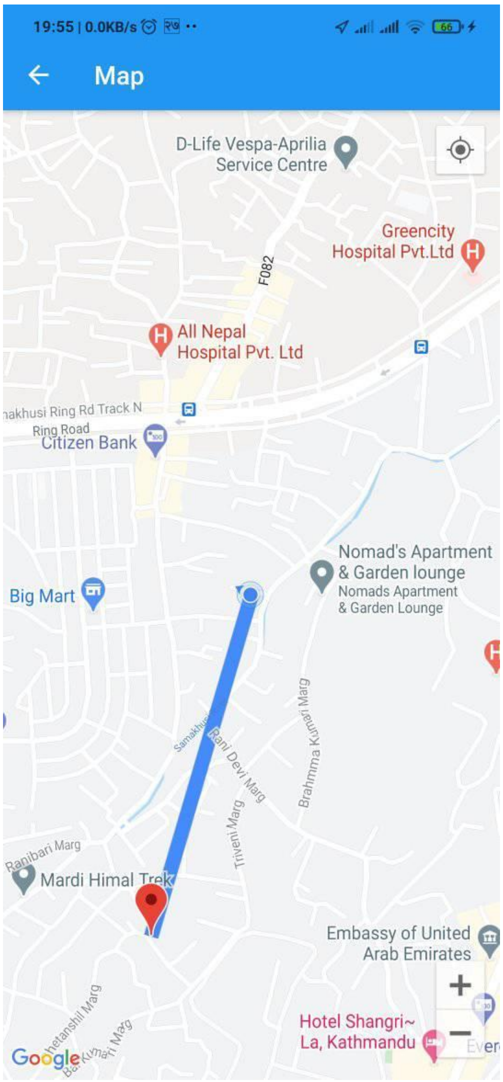


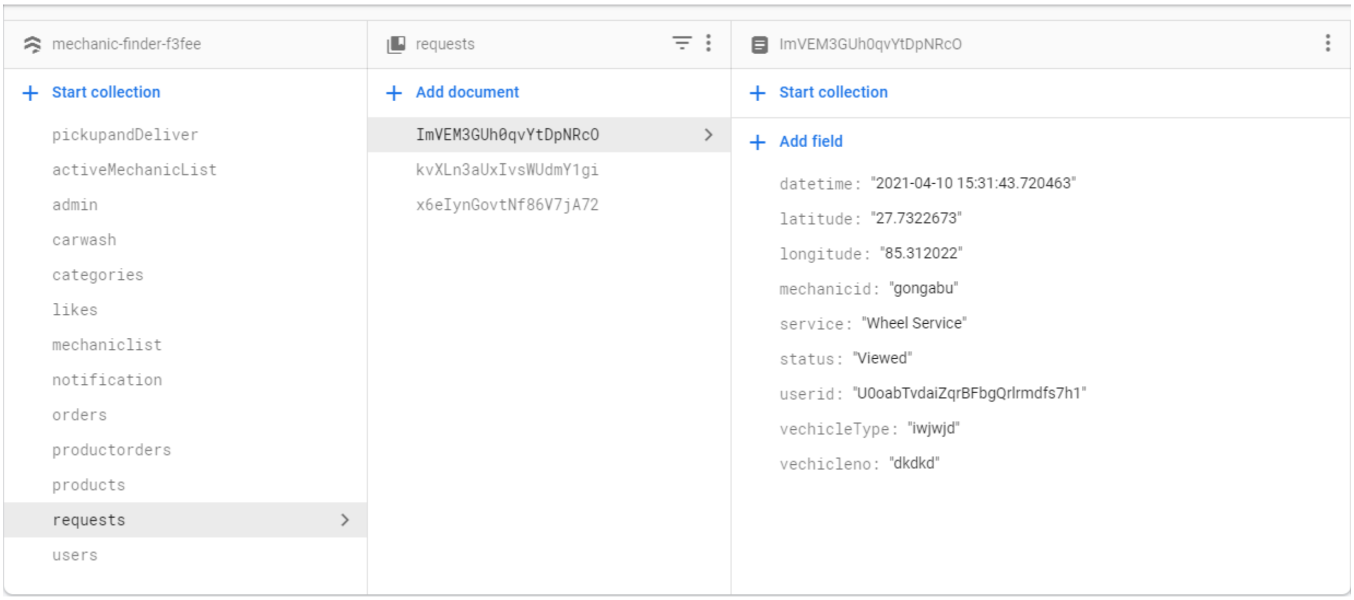
Ranibari
125.61016626245264

samakhushi
592.8254715725955

gongabu
34838.949005272356

NEouser
649229.2797900502


19:55 | 0.0KB/s
Map
D-Life Vespa-Aprilia Service Centre
Greencity Hospital Pvt.Ltd
F082
All Nepal Hospital Pvt. Ltd
akhusi Ring Rd Track N
Ring Road
Citizen Bank
Nomad's Apartment & Garden lounge
Nomads Apartment & Garden Lounge
Big Mart
Samakhusi
Rani Devi Marg
Brahmma Kumari Marg
Triveni Marg
Ranibari Marg
Mardi Himal Trek
Embassy of United Arab Emirates
Hotel Shangri~La, Kathmandu
Google



mechanic-finder-f3fee

+ Start collection

pickupandDeliver
activeMechanicList
admin
carwash
categories
likes
mechaniclist
notification
orders
productorders
products
requests >
users

requests

+ Add document

ImVEM3GUh0qvYtDpNRcO >
kvXLn3aUxIvsWUdmY1gi
x6eIynGovtNf86V7jA72

ImVEM3GUh0qvYtDpNRcO

+ Start collection

+ Add field

```
datetime: "2021-04-10 15:31:43.720463"
latitude: "27.7322673"
longitude: "85.312022"
mechanicid: "gongabu"
service: "Wheel Service"
status: "Viewed"
userid: "U0oabTvdaiZqrBFbgQrlrmdfs7h1"
vechicleType: "iwjwjd"
vechicleno: "dkdkd"
```