**Pit Locator**

**A Minor Project II**

Submitted in partial fulfillment of the requirements

for the degree of

**Bachelor of Technology**

**(Computer Science & Engineering)**

*by*

**Akash Kumar Kanojiya, Dipanshu Mishra, Anmol Sahu**

**0225CS191008 0225CS191023, 0225CS191016**

*Under the guidance of*

**Prof. Saurabh Sharma**

**Department of Computer Science & Engineering**



**Global Nature Care Sangathan’s Group of Institutions, Jabalpur (M.P.)**

*under*

**Rajiv Gandhi Prodyogiki Vishwavidyalaya, Bhopal (M.P.)**

**Apr-2022**

**Global Nature Care Sangathan’s Group of Institutions, Jabalpur (M.P.)**

Department of Computer Science & Engineering

**Certificate**

This is to certify that the Major Project report entitled “**Pit Locator”** submitted by **Akash Kumar Kanojiya, Dipanshu Mishra and Anmol Sahu** has been carried out under my guidance & supervision. The project report is approved for submission towards partial fulfillment of the requirement for the award of degree of **Bachelor of Engineering** in **Computer Science & Engineering** from “**Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P).**

**Prof. Saurabh Sharma Prof. Saurabh Sharma**

Project Incharge HOD

Dept. of CSE

**Global Nature Care Sangathan’s Group of Institutions, Jabalpur (M.P.)**

Department of Computer Science & Engineering

**Certificate**

This is to certify that the Major Project report entitled “**Pit Locator”** is submitted by **Akash Kumar Kanojiya, Dipanshu Mishra and Anmol Sahu** for the partial fulfillment of the requirement for the award of degree of **Bachelor of Engineering** in **Computer Science & Engineering** from **Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P).**

Internal Examiner External Examiner

Date : Date :

**ABSTRACT**

Pothole Tracking System which assists the driver in avoiding pot-holes on the roads, investigates an application of mobile detecting and reporting the surface conditions of roads. It describes a system to monitor this important civil infrastructure using an android based smart phone. The pothole tracking system uses the inherent mobility of the participating smart phone by the citizen, opportunistically gathering data from image clicking from an android based smart phone which is GPRS enabled, and processing the data to assess road surface conditions. Using a simple geotagging technique which is a feature of android OS, it show that we are able to identify potholes and other severe road surface anomalies from images clicked by the citizens and uploaded by the same application on the server. This project report presents the architecture of a Global Positioning System (GPS) based approach for reporting thoroughfare problems for road maintenance management environment. To increase accuracy and efficiency, GPS can be used as it enables the tracking and tracing of the three figures of a GPS receiver’s coordinates namely longitude, latitude and altitude.

While uploading the images, our app will take care of the Geo-location of the images, with the help of Geo- Tagging. Then, the images will be distributed to different city administrative officers. The additional feature in our app is that it will tell the user best possible route with pit location on map for their convenience. There is a fragment in our app with range and condition bar for the user to describe the adversity of the pothole. There is another fragment in our app which will record the number of complaints ward wise in a sorted manner for the government to prioritize and work on. Administrative officer can have a look on the report and take decisions accordingly

**Declaration**

I / We hereby declare that the project entitled “**Pit Locator”** which is being submitted in partial fulfillment of the requirement for award of the Degree of Bachelor of Engineering in Computer Science and Engineering to **“RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL (M.P.)”** is an authentic record of our own work done under the guidance of **Prof. Saurabh Sharma**, Department Computer Science & Engineering, **GLOBAL ENGINEERING COLLEGE, JABALPUR**..

The matter reported in this Project has not been submitted earlier for the award of any other degree.

**Dated : Akash Kumar Kanojiya**

**Place : 0225CS191008**

**Dipanshu Mishra**

**0225CS191023**

**Anmol Sahu**

**0225CS191016**

**Acknowledgment**

We sincerely express indebtedness to esteemed and revered guide “Prof. Saurabh Sharma”, in department of CSE for his invaluable guidance, supervision and encouragement throughout the work. Without his kind patronage and guidance the project would not have taken shape.

We take this opportunity to express deep sense of gratitude to “Prof. Saurabh Sharma”, Head of “Department of Computer Science & Engineering”for his encouragement and kind approval. Also we thank him in providing the computer lab facility. We would like to express our sincere regards to him for advice and counseling from time to time.

We owe sincere thanks to all the lecturers in “Department of Computer Science & Engineering*”* for their advice and counseling time to time.

**Dated : Akash Kumar Kanojiya**

**Place : 0225CS191008**

**Dipanshu Mishra**

**0225CS191023**

**Anmol Sahu**

**0225CS191016**

**Table of Contents**

1. **Chapter 1 : Introduction**
   1. Background
   2. Objective
   3. Purpose, Scope and Applicability
      1. Purpose
      2. Scope
      3. Applicability
2. **Chapter 2: Survey of Technologies**
3. **Chapter 3: Requirement and Analysis**
   1. Problem Definition
   2. Identification of Need
   3. Feasibility Study
   4. Project Planning
   5. Software Requirement Specification ( SRS)
   6. Software Engineering Paradigm applied.
4. **Chapter 4 : System Design**
   1. Basic Modules
   2. Data Design
      1. Database Design
   3. Diagrams
      1. Data Flow Diagram
      2. Use Case Diagram
   4. Use Interface Design
5. **Chapter 5 : Implementation**
   1. Coding Details
6. **Chapter 6 :Testing Approach**
   1. Testing techniques and testing strategies
      1. Unit Testing
      2. Integrated Testing
      3. Compatibility Testing
7. **Chapter 7 : SWOT Analysis**
   1. Strength
   2. Weakness
8. **Chapter 8 : Conclusions**
   1. Limitations of the System
   2. Future Scope and Further Enhancement of the Project

REFERENCES

**Chapter 1**

***Introduction***

* + - 1. **INTRODUCTION**
  1. **Background**

Roads are the dominant means of transportation in India today. Over the last 20 years, there has been an incredible increase within the vehicle population. This increase in the population of vehicles has led to several problems. One of the major problems is Potholes. Potholes are usually caused due to the heavy traffic and water on roads. The condition of road becomes worst when the roads are submerged under water after a single rainfall. Also because of the bad conditions of road, fuel consumption of the vehicle also increases, causing wastage of such precious fuel. There are many instances where people lost their lives due to a Pothole. Several studies conducted in cities like Chandigarh and Mumbai point to the shortage of a correct system and weak proportioning of aggregates for construction as major reasons for pothole. Because of the current manual system citizens are fed up of the long Queues of Municipal Corporation for their complaints regarding potholes on road and highways.

Road Accidents are increasing now days due to unfitness of roads everywhere. Contractors who take up road and highway contracts and maintenance contract does not do their task as stated in the tenders they file, so to have foolproof evidence and also to form them notice of their work done on roads. Keeping in mind all these problems we are developing an Android app named Pit Locator. Which will help citizens to lodge complain of a Pothole, locate Potholes in their path through map for safer journey, in addition this data will be shared to the municipal corporation so that the Potholes can be fixed.

**1.2 Objective**

Our objective is to create a mobile application through which the user can click pictures of the potholes on the road and, upload the pothole locations to a remote database that stores pothole location data, the data will be sent to the Administrative panel, from where the concerned authorities can fetch the data and send it to municipal corporation so that the information can be shared to the contractor to fix the road. Also, the locations of the potholes will be visible to the user through geo-locaters for convenience in route selection which will result in safer journey.

**1.3 Purpose, Scope and Applicability**

**1.3.1 Purpose**

Potholes are dangerous, damaging obstacles on the road for drivers. Not only are they jolting and uncomfortable, but they are a large cause of vehicular accidents every year. There are many instances where people lost their lives due to a Pothole. There is a need for a way to detect the potholes and report them to the appropriate authorities, So that proper actions can be taken.

* + 1. **Scope**

The scope of Pit Locator is global i.e. it should be able to be accessed from anywhere through their android device with active internet connection i.e. users must be able to click and upload the images to the app by verifying their mobile number with the help of OTP.

**1.3.3 Applicability**

Getting a pothole on your street should be as easy as sending a photo on your phone so a city roads for man can look it up on his android phone and get his crew working on it. Citizens are set free of the long queues. The mobile app is aimed at providing government with detailed statistics of areas with high numbers of potholes. No long procedures of complaints. It’s easy for the citizens to complaint the potholes and doesn’t need to go for long process of complaining. Corporations are having evidence to all the work done or doing.

**Chapter 2**

***Survey of Technologies***

**2. SURVEY OF TECHNOLOGIES**

* + - * 1. **Adobe XD:**

Adobe XD is the Adobe prototyping tool for user experience and interaction designers. Adobe XD features are used for creating wireframes, prototypes, and screen designs for digital products such as websites and mobile apps.

* + - * 1. **XML:**

The Extensible Markup Language (XML) is a simple text-based format derived from Standard Generalized Markup Language (SGML), for representing structured information: documents, data, configuration, books, transactions, invoices, and much more.  It defines a set of rules for encoding [documents](https://en.wikipedia.org/wiki/Electronic_document) in a format that is both [human-readable](https://en.wikipedia.org/wiki/Human-readable_medium) and [machine-readable](https://en.wikipedia.org/wiki/Machine-readable_data).

The design goals of XML emphasize simplicity, generality, and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, the language is widely used for the representation of arbitrary data structures such as those used in web services. It provides a standard method to access information, making it easier for applications and devices of all kinds to use, store, transmit, and display data.

* + - * 1. **Android Studio:**

Android Studio is the official Integrated Development Environment (IDE) designed for Android app development, based on IntelliJ IDEA. It is a unified environment where you can develop applications for all Android devices.  It is a replacement for the [Eclipse Android Development Tools](https://en.wikipedia.org/wiki/Eclipse_(software)#Android_Development_Tools) (E-ADT) as the primary IDE for native Android application development.

* + - * 1. **Java:**

Java is a programming language and computing platform first released by Sun Microsystems in 1995. It has evolved from humble beginnings to power a large share of today’s digital world, by providing the reliable platform upon which many services and applications are built. New, innovative products and digital services designed for the future continue to rely on Java, as well. Java is fast, secure, and reliable, therefore. It is widely used for developing Java applications in laptops, data centers, game consoles, scientific supercomputers, cell phones, etc. Java can be used to create complete applications that may run on a single computer or be distributed among servers and clients in a network. It can also be used to build a small application module or applet (a simply designed, small application) for use as part of a Web page

* + - * 1. **Google Maps API:**

Google Maps API is a set of application programming interfaces that lets developers talk to its services. It will allow developers to build simple apps to very sophisticated location-based apps for Web, iOS, and Android. The Maps JavaScript API lets you customize maps with your own content and imagery for display on web pages and mobile devices. The Maps JavaScript API features four basic map types (roadmap, satellite, hybrid, and terrain) which you can modify using layers and styles, controls and events, and various services and libraries.

* + - * 1. **Firebase database:**

The Firebase Realtime Database is a cloud-hosted database in which data is stored as JSON. The data is synchronized in real time to every connected client. By using a single API, the fire database provides the application with the current value of the data and updates to that data. Instead of typical HTTP requests, the Firebase Realtime Database uses data synchronization—every time data changes, any connected device receives that update within milliseconds. Provide collaborative and immersive experiences without thinking about networking code. The Firebase Realtime Database can be accessed directly from a mobile device or web browser; there’s no need for an application server. Security and data validation are available through the Firebase Realtime Database Security Rules, expression-based rules that are executed when data is read or written.

* + - * 1. **Cloudinary image database:**

Cloudinary is an end-to-end image- and video-management solution for websites and mobile apps, covering everything from image and video uploads, storage, manipulations, optimization to delivery. With Cloudinary, you can easily upload images and videos to the cloud and automate smart manipulations of those media without installing any other software. Cloudinary then seamlessly delivers your media through a fast content delivery network (CDN), optimized with the industry’s best practices. Additionally, Cloudinary offers comprehensive APIs and administration capabilities, which you can easily integrate with your web and mobile apps.

* + - * 1. **Geo-Tagging:**

Geotagging is the process of geographical identification [metadata](https://en.wikipedia.org/wiki/Metadata) to various media such as a [geotagged photograph](https://en.wikipedia.org/wiki/Geotagged_photograph) or video, websites, SMS messages, QR Codes or [RSS](https://en.wikipedia.org/wiki/RSS_(file_format)) feeds and is a form of [geospatial metadata](https://en.wikipedia.org/wiki/Geospatial_metadata). This data usually consists of [latitude and longitude coordinates](https://en.wikipedia.org/wiki/Geographic_coordinate_system), though they can also include [altitude](https://en.wikipedia.org/wiki/Altitude), [bearing](https://en.wikipedia.org/wiki/Bearing_(navigation)), distance, accuracy data, and place names, and perhaps a [time stamp](https://en.wikipedia.org/wiki/Time_stamp).

Geotagging can help users find a wide variety of location-specific information from a device. For instance, someone can find images taken near a given location by entering latitude and longitude coordinates into a suitable image [search engine](https://en.wikipedia.org/wiki/Search_engine). Geotagging-enabled information services can also potentially be used to find location-based news, websites, or other resources. Geotagging can tell users the location of the content of a given picture or other media or the [point of view](https://en.wikipedia.org/wiki/Perspective_(visual)), and conversely on some media platforms show media relevant to a given location.

The geographical location data used in geotagging can, in almost every case, be derived from the [global positioning system](https://en.wikipedia.org/wiki/Global_positioning_system), and based on a [latitude](https://en.wikipedia.org/wiki/Latitude)/[longitude](https://en.wikipedia.org/wiki/Longitude)-coordinate system that presents each location on the earth from 180° west through 180° east along the [Equator](https://en.wikipedia.org/wiki/Equator) and 90° north through 90° south along the [prime meridian](https://en.wikipedia.org/wiki/Prime_meridian).

**Chapter 3**

***Requirement and Analysis***

1. **REQUIREMENT AND ANALYSIS**
   1. **Problem Definition**

The importance of the road infrastructure for the society could be compared with importance of blood vessels for humans. To ensure road surface quality it should be monitored continuously and repaired as necessary. Potholes and speed ramps are among the most common obstacles on most road networks. These road obstacles and disruptions in the surface condition of roads are among the causative factors of road accidents, carnages and fatalities. Poor road conditions often cause problems and harm to automobiles and passengers that cannot be ignored. Some of the big factors that are causing harm are potholes, speed-breakers, rough patches, etc. The existence of such abnormalities also affects the consistency of the ride and protection of the passengers.

* 1. **Identification of Need**
  + Citizens have to go Municipal Corporation to report the complaint regarding potholes.
  + Citizens are feed up of the long queues of Municipal Corporation for their mere complaints regarding potholes on road and highways.
  + Road Accidents are increasing now days because of bad condition of roads everywhere.
  + The process of complaining is lengthy and time consuming.
  + Less people report a complaint.
  + Road Accidents are increasing now days because of bad condition of roads everywhere.
  + Contractors who take up road and highway contracts and maintenance contract does not do their task as stated in the tenders they file.
  1. **Feasibility Study**
     + - 1. **Technical Feasibility:**

Project Pit Locator app is a complete android based application. The main technologies and tools that are associated with Pit Locator app are

* Java
* Android studio
* XML
* Firebase Database
* Google maps API
* Adobe XD
* Dependencies made for androids
* Geo-Tagging

Each of the technologies are easily available and the technical skills required are manageable. Time implementation of the product development and the ease of implementing using these technologies are synchronized.

The app is free to use, user needs an internet connection to access all the features of app. It’s easy to maintain and upgrade these technologies according to the need.

* + - * 1. **Economic Feasibility**

Being an android app Pit Locator will have an associated hosting cost. The technologies and the resources used have a minimal cost which can be easily covered by the cost that will be saved by automating the pothole complaining procedure, since there will be no need of maintaining paper reports of complain, no need of staffs to listen and maintain the complains, everything will be stored in the database automatically whenever user files a complaint from the app.

This app will be freely available in the android app market, from where user can easily download it to their android smartphones. This app will require an active internet connection in order to use all its services. This would be the only cost incurred by the user.

* + - * 1. **Operational Feasibility**

Since the app will be designed with user friendly UI that will be easy to use and no irrelevant features will be added to make the app simpler and easy to use, no complex features will be added, it will be designed with clean UI so that any type of user can use the app.

Since all these factors will be kept in mind during the designing of the app, so the app will be operationally feasible.

* + - * 1. **Resource Feasibility**

Resources that are required for the development of Pit Locator app includes,

* Programming device(Laptop)
* Programming tools(Android Studio – freely available)
* Programming individuals
* Testing device(Android device)

Since all the resources are easily available, it’s clear that the project has the required resource feasibility.

* 1. **Project Planning**

The planning of the project started when we saw the problems arising due to potholes on the road. Since, Road networks are the means of transporting, sharing and movement of goods and services from place to place in society. Road networks are also channels of communication in some parts of the world.

But the poor nature of design and development of road networks coupled with natural disasters such as heavy rainfall has brought about many unwanted potholes and scratches on the roads which are very dangerous to commuters and other road users as well as vehicles that utilize the roads.

In addition, the lack of a proper road maintenance system has resulted in an ever-increasing number of potholes which endanger efficient transportation and road safety.

Meanwhile road maintenance works has largely depended on manual detection and reporting, Citizens have to go Municipal Corporation to report the complaint regarding potholes. This manual process is Lengthy and time consuming and less people report a complaint.

Thus, there was a need for a system which can help people register complain of pothole with just one click, and also to get assisted while driving their vehicles so that they can choose the best path and avoid the path with large number of potholes for their safer and comfortable journey.

* 1. **Software Requirement Specification ( SRS)**

The requirements specification part of any project is a very important one. This can give us an insight into the perspective of what would be required from the app in order for it to be successful.

1. **Developer’s Hardware Requirements –**

A computer or laptop with following hardware specification is required –

* 64-bit Microsoft® Windows® 8/10.
* x86\_64 CPU architecture; 2nd generation Intel Core or newer, or AMD CPU with support for a [Windows Hypervisor](https://developer.android.com/studio/run/emulator-acceleration#vm-windows).
* 8 GB RAM or more.
* 8 GB of available disk space minimum (Android Studio IDE + Android SDK + Android Emulator).
* 1280 x 800 minimum screen resolution.
* Android mobile device for testing.

1. **Developer’s Software Requirements –**

* Android Studio Software.
* Adobe XD software.

1. **User’s Hardware Requirements –**

This section will describe the user requirements needed so they can use the application efficiently.

* **Android Phone –** User must have an android phone in order to run the application.
* **Internet Connection –** The user will also need an internet connection to upload the picture of pothole and to use google maps for selecting the best path.

1. **User’s Software Requirements –**

* **API Level –** The minimum sdk level for the app is 26 and the target sdk is 31.
* **Android Version –** Android Noughat 7.0 or later.

1. **Functional Requirement**

* **Complaint Box Fragment -**  Here users can lodge the complaint of a pothole on the road by clicking the picture of the pothole from their smartphone.
* **Locator Fragment -** Here user can see the potholes location marked in the map with the help of location that we fetched during the image upload. And by entering the starting and ending destination the map will display all the potholes in entered path.
* **Report Fragment –** Here user can see the area wise reports that will ensure transparency to the user.

1. **Non – Functional Requirements**

**Permissions**

* **Location Permission –** location permission is required in order to access the map.
* **Internet Permission –** Internet permission is required to access all the features of the app like uploading image and using maps to see the path and also to see the reports of the pothole in report fragment.
* **Camera Permission –** Camera permission is required so that user can click the image of pothole through their smartphone’s camera.

**Performance**

The system must be interactive and the updates must be fast. So, in every action-response of the system, there are no immediate delays. In case of uploading the image it should be smooth and OTP for verification should also be received as soon as the get OTP button is clicked. Also loading the map should be fast. And the reports shown in the reports section should be accurate.

**Scalability**

The app should be able to adopt itself to increased usage or able to handle more data as time progress. When the user data increases app should be capable of handling them without delay by optimizing the way storage is done and accessed.

**Responsiveness**

The application should be responsive to the user input or to any external interrupt which is of highest priority and return back to same state.

**Availability**

The user can access the application to install and look for regular updates and give feedback from Google Play store.

* 1. **Software Engineering Paradigm applied**

For developing this Pit Locator Application we used the Iterative Model.

**Iterative Model:**

In the Iterative model, iterative process starts with a simple implementation of a small set of the software requirements and iteratively enhances the evolving versions until the complete system is implemented and ready to be deployed.

An iterative life cycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which is then reviewed to identify further requirements. This process is then repeated, producing a new version of the software at the end of each iteration of the model.

Iterative process starts with a simple implementation of a subset of the software requirements and iteratively enhances the evolving versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities are added. The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental).



**Chapter 3**

***System Design***

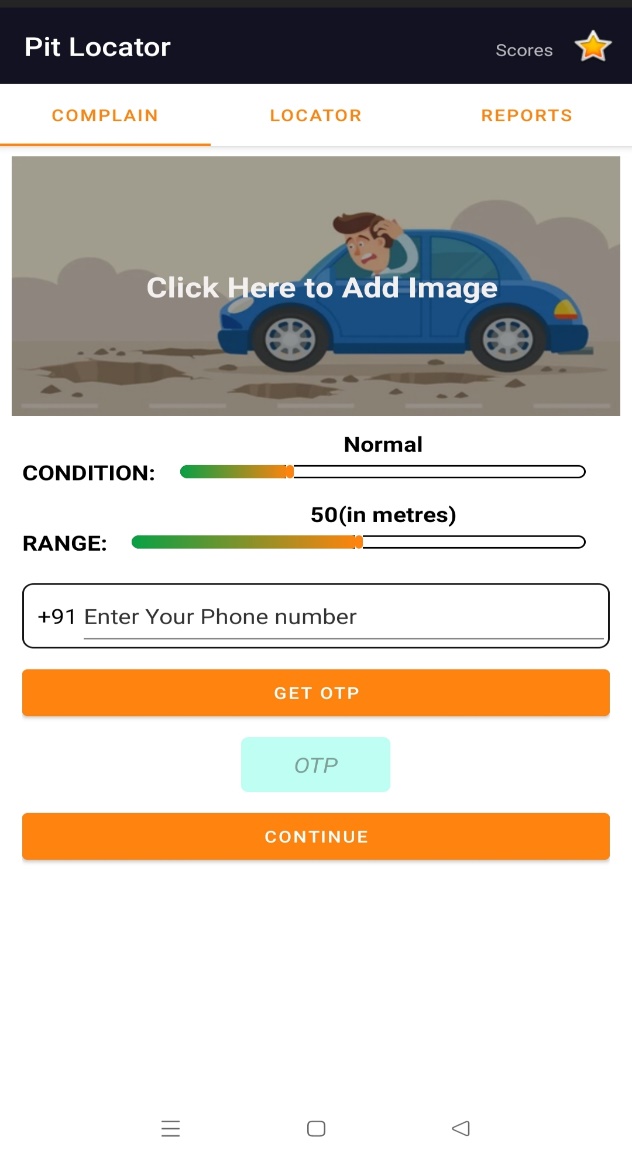
1. **SYSTEM DESIGN**
   1. **Basic Modules**

Our Project Idea is divided into the following two parts :-

* User Panel
* Admin Panel

**User Panel**

**Module 1 – Complaint Box Fragment:**

****

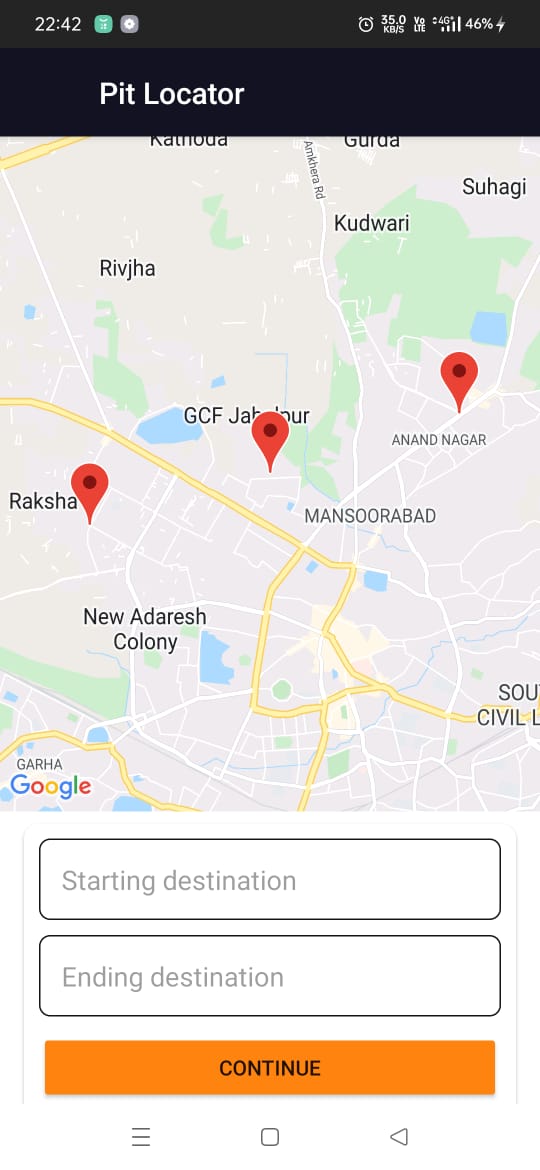
In the Complain Box fragment user can add complain about the potholes in their path while driving their vehicle.

* When user will open the app, it will ask for permissions to access their location and camera permission to take pictures of the potholes.
* When the user will click on the image view (click here to add image) it will be redirected to the phone's camera app from where user can click the image of a pothole.
* In the **“Condition Seek bar”** user can specify the condition of the pothole, like whether the condition is normal, moderate or extreme.
* In the **“Range Seek bar”** user can specify the range of pothole.
* Below range seek bar there is an **“Edit Text view”** where user will have to enter their mobile number from which they want to upload the image of pothole, it will help for the OTP verification of the mobile in order to ensure that the image is being uploaded by a genuine person.
* After entering the mobile number user will have to click on the **“GET OTP”** button in order to receive an OTP for verification in the above mentioned phone number.
* Below the **“GET OTP”** button is the **“OTP Edit Text view”** in this **“OTP Edit Text view”** user will enter the received OTP.
* By clicking on the continue button the entered OTP will be verified and if the verification is successful the image will be sent to Firebase ML where it will be verified that whether the particular image is really an image of a pothole or not and if it is verified successfully then the image will be uploaded in the database.
* In case the entered OTP is wrong then user will be asked to inter the correct OTP.
* If the OTP is not received by the user, then user can again click on the **“GET OTP”** button after 30 seconds.

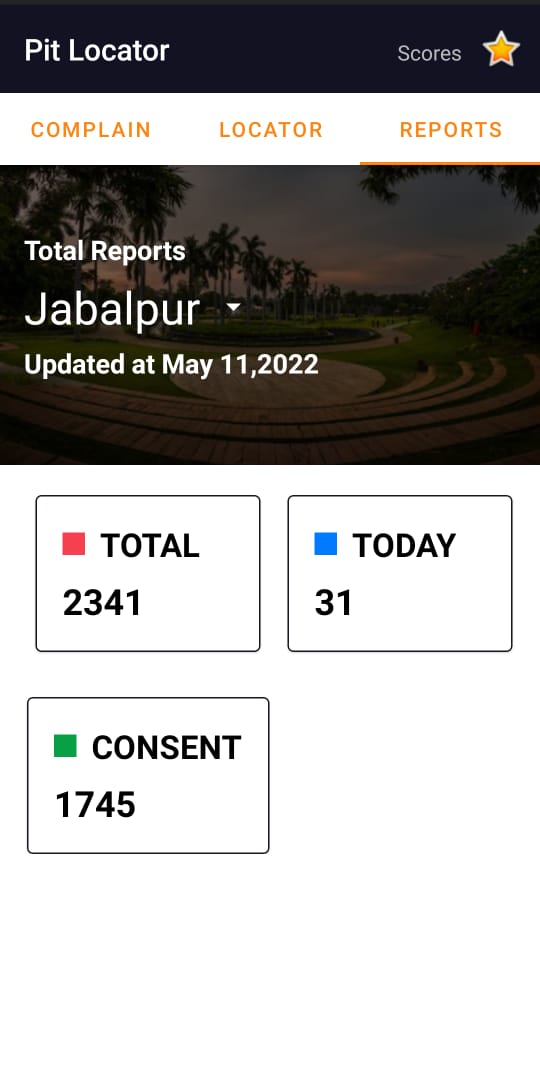
**Module 2 – Locator Fragment:**

When user will open the app and will toggle to Locator fragment then initially the map will show all those potholes locations through marker. The potholes location that is visible on the map is the location which was captured when the user uploaded the image of the pothole.

And if the user wants to travel by road for their journey then the users can enter their starting and ending destination and the map will display all the potholes between the starting and ending destination and it will also suggests the best possible route. In order to make the journey safer and comfortable.



**Module 3 – Reports Fragment:**

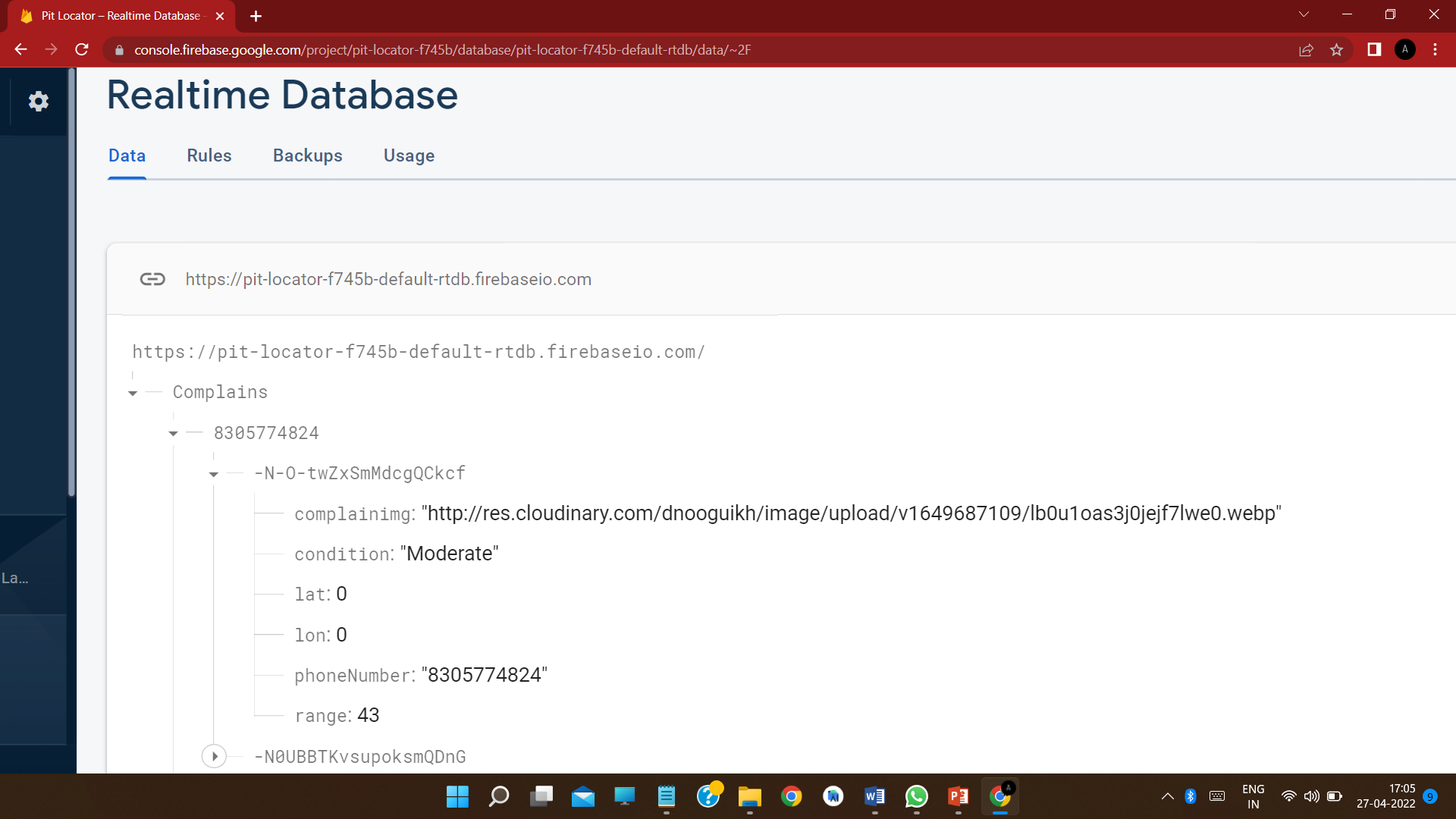


On the user's end, the number of daily new complaints and the number of problems resolved will be displayed through which they can easily monitor the status of the complaints to keep a regulatory check.

Here user will find area-wise reports that will ensure transparency to the user.

**Admin Panel**

* The admin panel will show record of complaints ward/area wise in a sorted manner to prioritize and work on.
* It will also regularly keep updating the status of the submitted report.
* If a contractor marks an unresolved complaint as resolved and removes it from the database, the next time when the identical complaint is made, it will be highlighted.
* For repairs, the administrator can reach out to the contractor.
  1. **Data Design**
     1. **Database Design**



To Store the data uploaded by the user we have used **“Firebase Realtime Database”**

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client.

The Firebase Realtime Database is a NoSQL database from which we can store and sync the data between our users in real-time. It is a big JSON object which the developers can manage in real-time. By using a single API, the Firebase database provides the application with the current value of the data and updates to that data

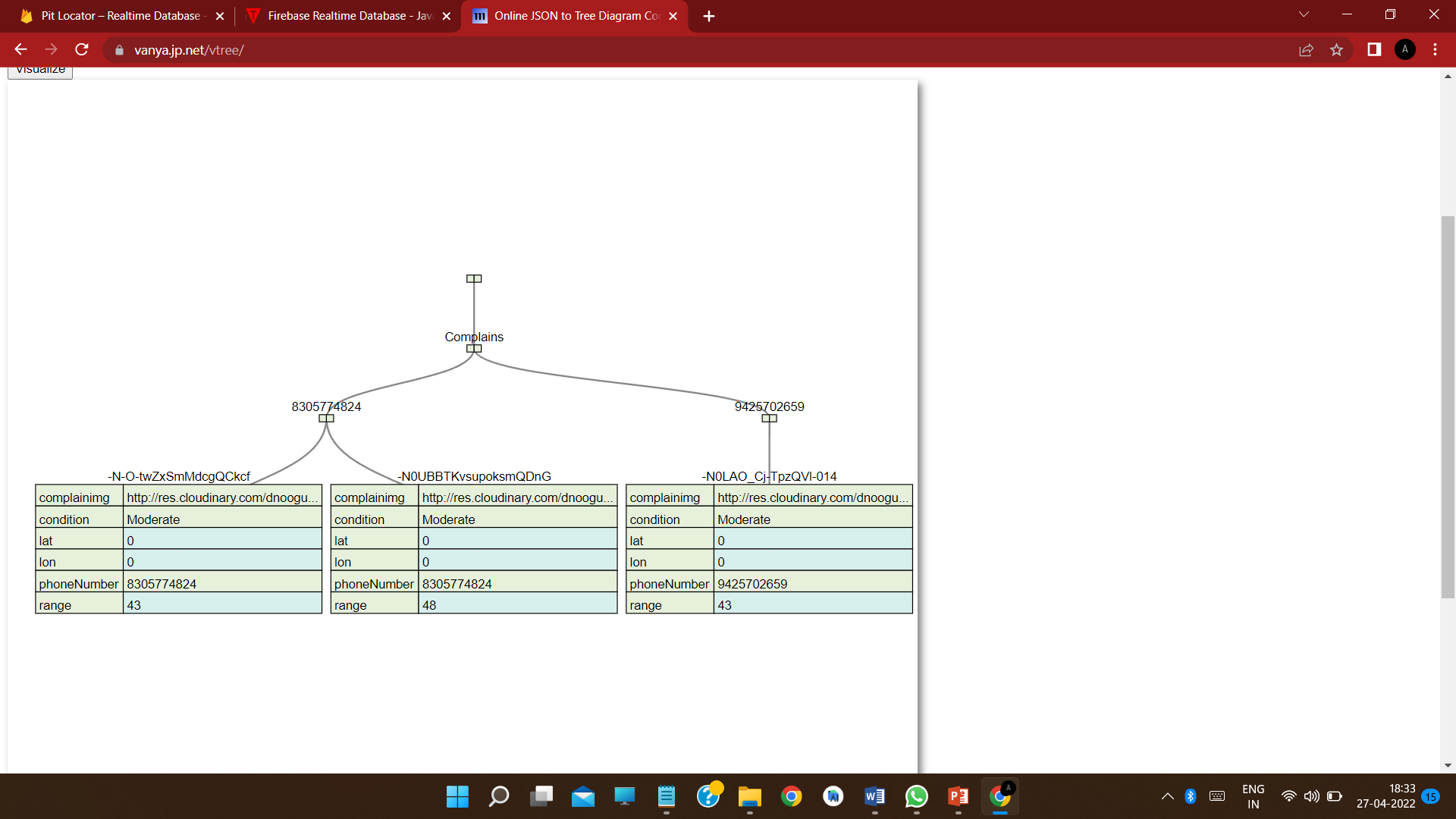
Here the name of the database table is “Complains”

All the complains that the user make are identified uniquely with the help of mobile number

Since from one mobile number multiple complains can be made therefore firebase realtime database automatically generates unique id for each complain from same mobile number.

Each unique complain id contains the **“Complain Image URL”**, **“Pothole condition”**, **“Pothole Range”**, **“Latitude and Longitude”** of the location from where the image was clicked, **“Phone Number”** of the user who made the complaint.

**JSON Object Tree**



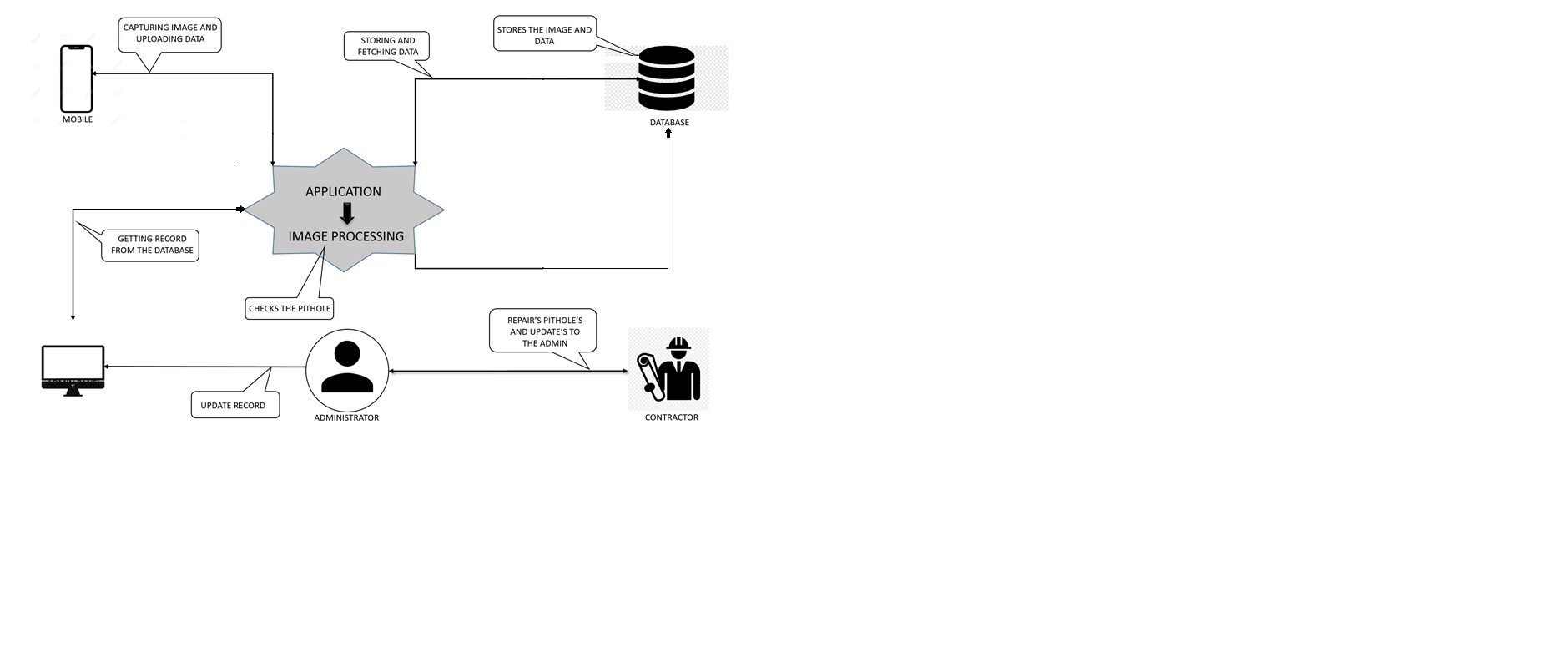
Here the name of the database table is “Complains”

All the complains that the user make are identified uniquely with the help of mobile number

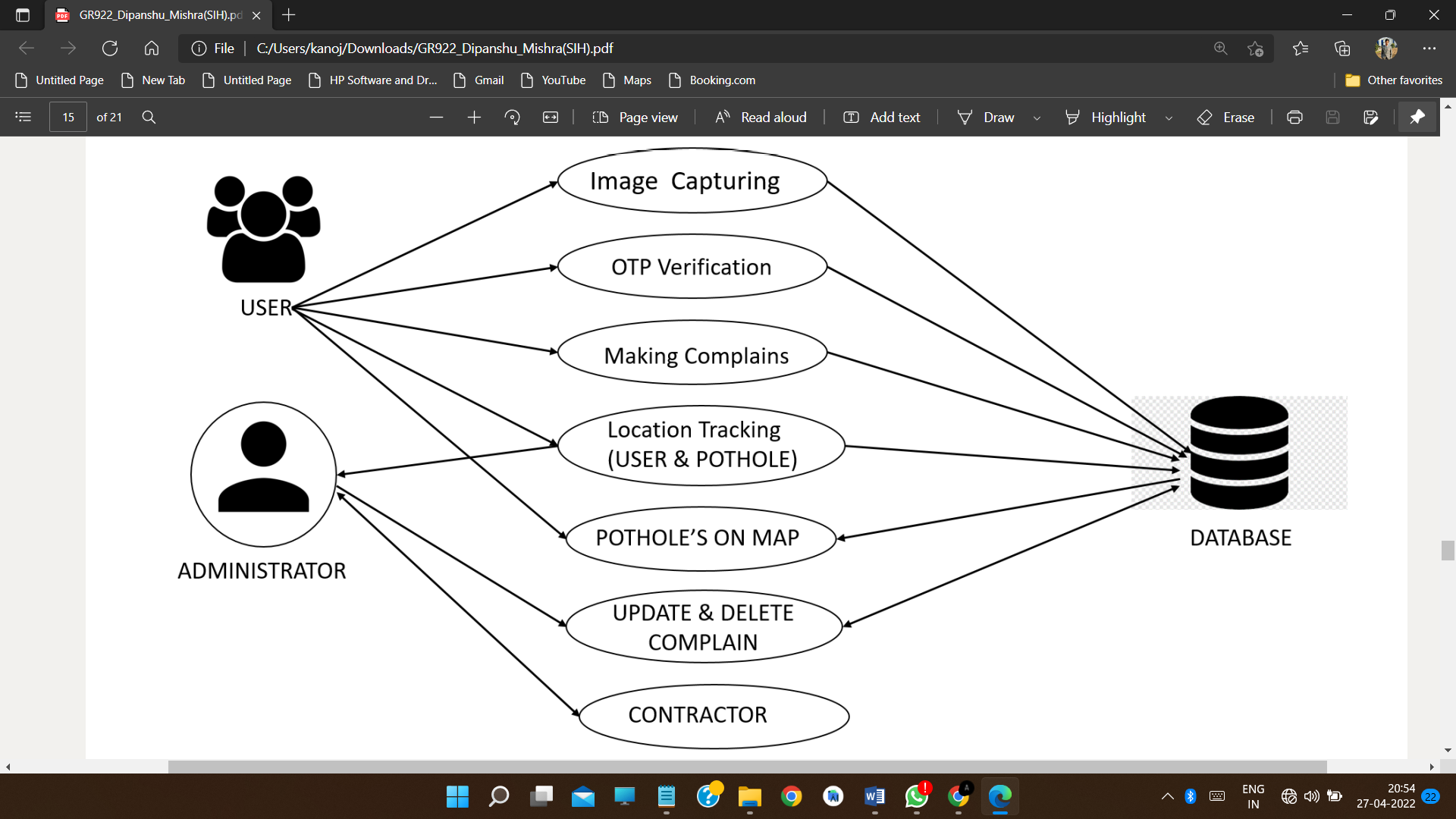
Since from one mobile number multiple complains can be made therefore firebase realtime database automatically generates unique id for each complain from same mobile number.

Each unique complain id contains the **“Complain Image URL”**, **“Pothole condition”**, **“Pothole Range”**, **“Latitude and Longitude”** of the location from where the image was clicked, **“Phone Number”** of the user who made the complaint.

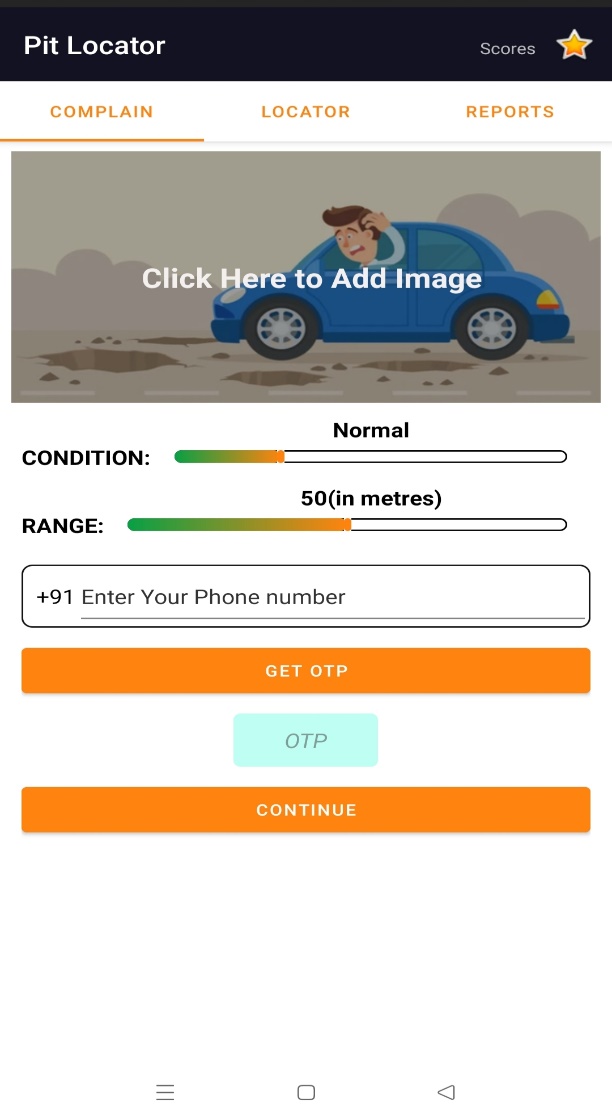
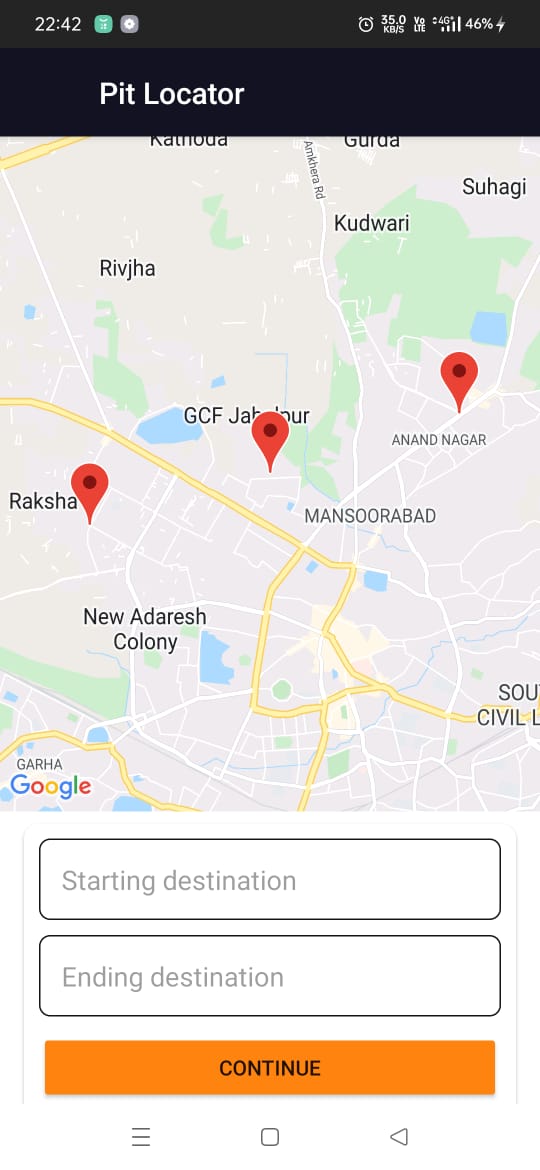
* 1. **Diagrams**
     1. **Data Flow Diagram**



* + 1. **Use Case Diagram**

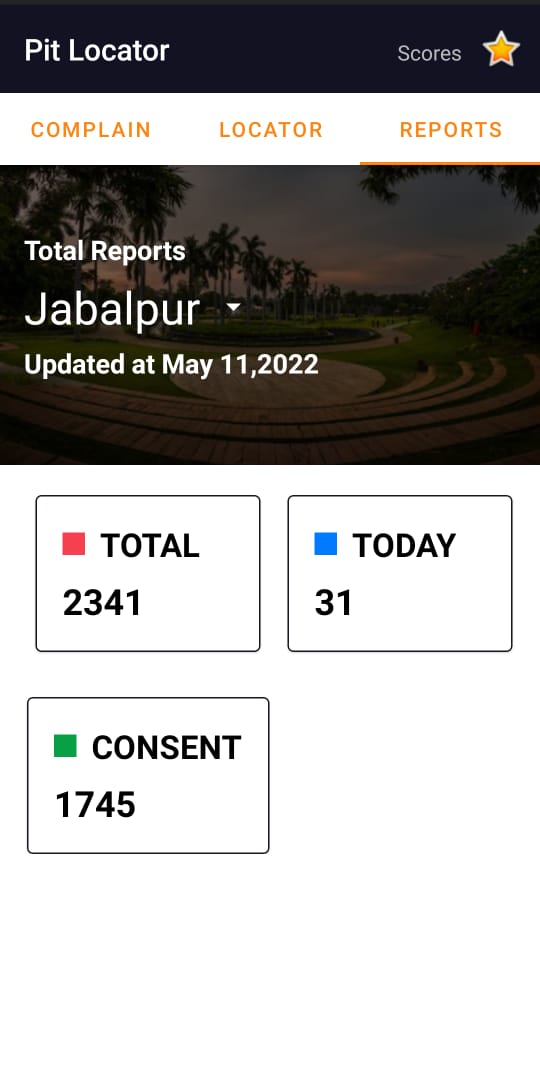


* 1. **Use Interface Design**



Complain Box Fragment

Locator Fragment



Report Fragment

**Chapter 5**

***Implementation***

1. **Implementation**
   1. **Coding Details**

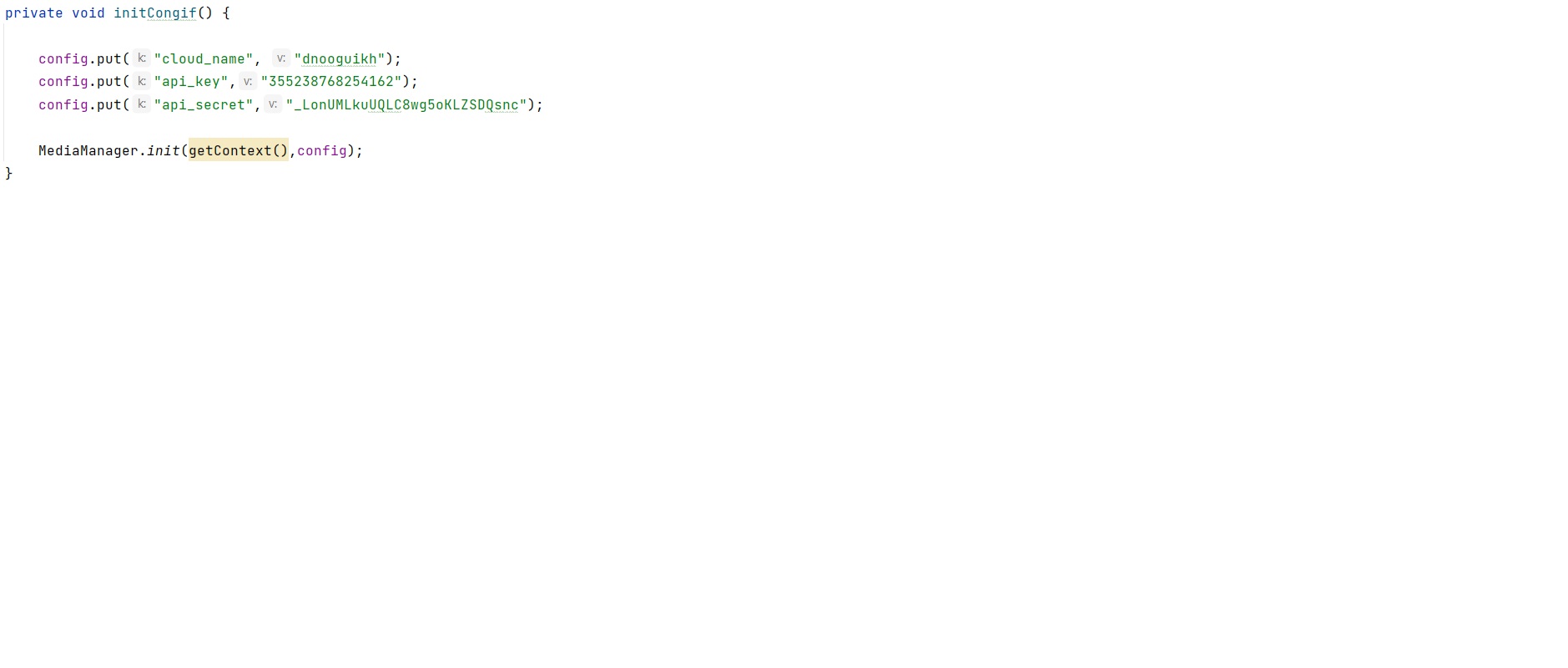


1. **Code for requesting storage permission**

**B.** **Code for checking and requesting location permission**



**C.** **Initializing Cloudinary database**



**D.** **Code to select and load the clicked image of pothole to the app**



**E.** **Code for OTP verification**



**F.** **Code for uploading image to the database**



**G.** **Complain Modal Class**

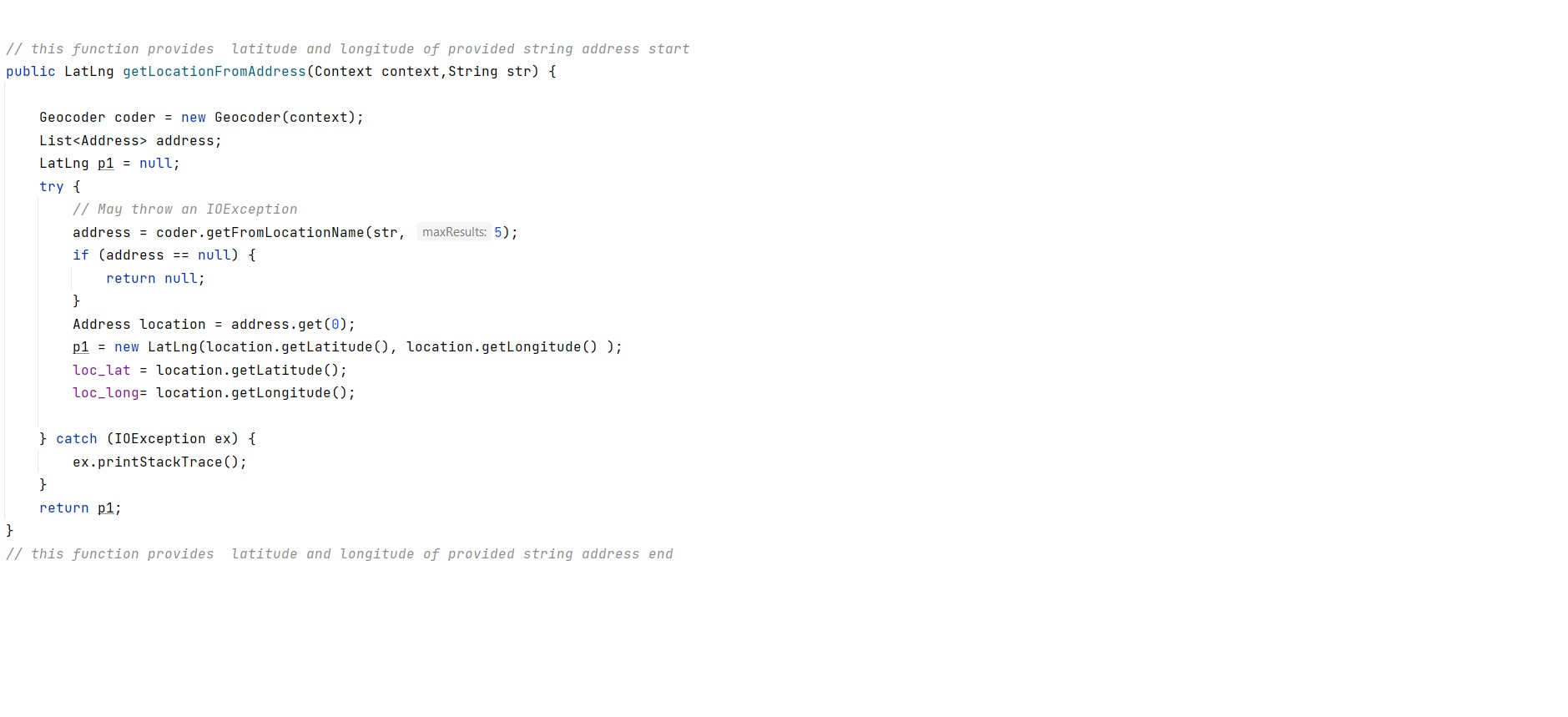
**H.** **Requesting location from a new user**



**I.** **Getting the device location** **(Latitude and Longitude)**



**J.** **Code for getting the latitude and longitude of starting and ending address**



**Chapter 6**

***Testing Approach***

1. **TESTING APPROACH**
   1. **Testing techniques and testing strategies**
      1. **Unit Testing**

For basic testing needs, **Android Studio** includes features that help you create, run, and view results of tests all from the IDE. Using Studio, we can point and click in the app source code to create and run tests for specific classes or methods, use drop-down menus to configure multiple test devices, and interact with the Text Matrix tool window to visualize test results.

While developing this application, after a new feature was added or any changes were made we performed unit testing. Whenever we updated something we ran the app in emulator as well as in 2 android mobile devices. We also took the help of logcat to identify the errors and removed it.

* + 1. **Integrated Testing**

Integration tests verify the way different parts of your app work together. They are slower than unit tests, and should therefore only be used when you need to test how things interact. When interacting with the Android framework we can rely on an Android device or emulator.

While developing the application after performing the Unit testing we integrated all the units of programs and did integrated testing to check that the code works correctly when all the features are put together.

* + 1. **Mobile Device Compatibility Testing**

Mobile Compatibility testing refers to specific testing of software applications and their suitability with mobile devices.  These mobile devices can have different operating systems, hardware and software requirements and the application must meet these requirements to pass the testing phase.

Mobile compatibility testing is essential for ensuring the optimal performance of mobile apps. The goal of the testing procedure is to examine hardware dimensions in a mobile device. These hardware dimensions may include components like the motion sensors, GPS and navigation capabilities, camera (resolution and type), processing speed, screen (size, resolution, or rotation), main memory, etc.

To check whether the app is compatible on majority of the android devices we did compatibility testing by running the app in various android mobiles and also in android tablets with min sdk 26 and higher.

**Chapter 7**

***SWOT Analysis***

1. **SWOT ANALYSIS**
   1. **Strength**

* Getting a pothole on your street should be as easy as sending a photo on your phone so a city roads for man can look it up on his android phone and get his crew working on it. Citizens are set aloof of the long queues.
* The mobile app is aimed at providing government with detailed statistics of areas with high numbers of potholes. No long procedures of complaints.
* It’s easy for the citizens to complaint the potholes and doesn’t need to go for long process of complaining. Corporations are having evidence to all the work done or doing.
* It’s also easy for the admin and the contractor to the potholes.
* Map will help drivers by showing the potholes on the map, and will suggest best route.
  1. **Weakness**
* **Human Error** - First of all, we admit, that a human giving the range and condition of pothole is subject to multiple errors due to human factors.
* **Network Connection –** Since this app requires internet connection in order to perform its tasks, therefore in places with no network coverage this app will not be able to perform all the functionalities.

**Chapter 8**

***Conclusions***

1. **CONCLUSIONS**

We have developed a mobile application “Pit Locator” through which the user can click pictures of the potholes on the road and, upload the pothole locations to a remote database that stores pothole location data, the data will be sent to the Administrative panel, from where the concerned authorities can fetch the data and send it to municipal corporation so that the information can be shared to the contractor to fix the road. Also, the locations of the potholes will be visible to the user through geo-locaters for convenience in route selection which will result in safer journey.

With the help of “Pit Locator” app it will be easy to track down the pothole on the road and can easily get rectified. There is lot of other research about finding potholes but it can be more expensive and not feasible to implement but with the help of android application we can use it at very less cost and much reliable. By this system the Municipal can also keep an eye on the quality of work of the contractor & we can get world class roads in our city. With further work in this field, it is possible for this project to play a proactive part in improving road conditions in developing countries.

This application will also help in avoiding accidents that are taking place due to the potholes and will save the precious lives of the citizens.

**8.1 Limitations of the System**

* User will be able to upload photos and apprise the civic officials on a real time basis.
* User need to have an android phone with active internet connection.
* Users need to choose and upload the geo-tagged image of the pothole or capture the image of the pothole with GPS enabled on their mobile phones.
* User’s mobile number will be required for verification purpose.
* First of all, we admit, that a human giving the range and condition of pothole is subject to multiple errors due to human factors.
* Since this app requires internet connection in order to perform its tasks, therefore in places with no network coverage this app will not be able to perform all the functionalities

**8.2 Future Scope and Further Enhancement of the Project**

* Can be implemented in Cities under Municipal Corporation.
* Can also be implemented for Sewage Lines and Water Pipelines.
* Efficient and Easy Work for Citizens i.e. Human Interaction.
* This System Can Also Be Implemented Using GPS Sensors On Vehicles.

**REFERENCES**

1. <https://firebase.google.com/docs/build> - Firebase documentation.
2. <https://developer.android.com/docs> - Android documentation.
3. <https://stackoverflow.com/> - Stack Overflow.
4. <https://cloudinary.com/documentation/android_integration> - Cloudinary Android integration documentation.
5. <https://www.wikipedia.org/> - WikiPedia
6. Jorg Roth, “The Resource Framework for Mobile Applications”, 58084 Hagen, Germany. Jan 7, 2003