A

Project Report

on

CLEARNAV, A POTHOLE ALERT NAVIGATION APPLICATION

Submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology

by

K. Sai Ganesh

(20EG105626)

M. Sai Vivek

(20EG105632)

D. Ajay Chowdary

(20EG105646)

Under the guidance of

Dr. A. Jyothi

Assistant Professor,

Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ANURAG UNIVERSITY

VENTAKAPUR (V), GHATKESAR (M), MEDCHAL (D), T.S - 500088

TELANGANA

(2023-2024)

DECLARATION

We hereby declare that the report entitled **"ClearNav a Pothole Alert Navigation Application"** submitted to the **Anurag University** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology (B. Tech)** in **Computer Science and Engineering** is a record of an original work done by us under the guidance of **Dr. A. Jyothi, Assistant Professor** and this report have not been submitted to any other university for the award of any other degree or diploma.

Place:

Date:

K. Sai Ganesh

20EG105626

M.Sai Vivek

20EG105632

D.Ajay Chowdary

20EG105646

CERTIFICATE

This is to certify that the project report entitled **"ClearNav a Pothole Alert Navigation Application"** being submitted by **K. Sai Ganesh, M. Sai Vivek, D. Ajay Chowdary** bearing the Hall Ticket numbers **20EG105626, 20EG105632, 20EG105646** respectively in partial fulfillment of the requirements for the award of the degree of the **Bachelor of Technology** in **Computer Science and Engineering** to **Anurag University** is a record of bonafide work carried out by them under my guidance and supervision from 2023 to 2024.

The results presented in this report have been verified and found to be satisfactory. The results embodied in this report have not been submitted to any other University for the award of any other degree or diploma.

Signature of The Supervisor	Signature Dean,
Dr. A. Jyothi	Dr. G. Vishnu Murthy
Assistant Professor	Department of CSE

External Examiner

ACKNOWLEDGMENT

We would like to express our sincere thanks and deep sense of gratitude to project supervisor **Dr. A. Jyothi,** Assistant Professor, Department of Computer Science and Engineering, Anurag University for her constant encouragement and inspiring guidance without which this project could not have been completed. Her critical reviews and constructive comments improved my grasp of the subject and steered to the fruitful completion of the work. Her patience, guidance and encouragement made this project possible.

We would like to acknowledge our sincere gratitude for the support extended by **DR. G. VISHNU MURTHY**, Dean, Department of Computer Science and Engineering, Anurag University. We also express our deep sense of gratitude to
Dr. V. V. S. S. S. BALARAM, Academic coordinator. **Dr. PALLAM RAVI**, Project Coordinator and project review committee members, whose research expertise and commitment to the highest standards continuously motivated us during the crucial stages of project work.

We would like to express our special thanks to **Dr. V. VIJAYA KUMAR,** Dean School of Engineering, Anurag University, for their encouragement and timely support in our
B. Tech program.

K. Sai Ganesh
(20EG105626)

M. Sai Vivek
(20EG105632)

D. Ajay Chowdary
(20EG105646)

ABSTRACT

"ClearNav," a revolutionary app designed to transform your driving experience by crowdsourcing information about potholes and deteriorating roads. Users can easily upload the locations of road imperfections, creating a comprehensive database that serves as a real-time navigation aid. As you approach reported areas, ClearNav employs cutting-edge geolocation technology to send instant alerts, ensuring you are prepared for potential road hazards. The app promotes community collaboration to enhance road safety, enabling users to contribute to the collective well-being of drivers. ClearNav's intuitive interface and seamless integration with navigation systems make reporting and receiving alerts effortless, fostering a safer and smoother journey for everyone on the road. Say goodbye to unexpected bumps and enjoy a more informed, stress-free driving experience with ClearNav - your trusted companion for navigating the streets with confidence.

Keywords: Google Maps API, Database, Geolocation, Markers, Crowd-Sourcing.

TABLE OF CONTENT

List of Figures

List of Tables

List of Abbreviations

Abbreviations	Full Form
API	Application Programming Interface
APP	Application
SDK	Software Development Kit
IDE	Integrated Development Environment
iOS	iPhone Operating System
UI	User Interface
GPS	Global Positioning System
JSON	JavaScript Object Notation
ZIP	Zipped archive
PSI	Pothole Severity Index
ART	Average Response Time
EIR	Efficiency Index for Repairs

1. INTRODUCTION

Potholes are a major cause of road accidents and injuries, and there is a growing demand for new technology to increase safety. Potholes are created by a number of circumstances, including weather, traffic, and inadequate road construction. Potholes can pose a severe hazard to vehicles, bikers, and pedestrians. They can damage vehicles and contribute to accidents. Traditional pothole detection methods, such as physical inspection and visual surveys, require a significant amount of time and labor. They may also be inaccurate, particularly in big or rural places. However, in recent years, there has been an increased interest in using technology to improve pothole detection

In this project, we will create a system for implementing road safety utilizing GPS and marking on maps. This is the simplest way that involves collecting images of road damage and hazards from participants and uploading them to a central server. Users must actively participate and perform manual image analysis. The application will alert the users 50 meters prior to the pothole.

By utilizing GPS and mapping functionalities, the system aims to accurately identify and mark potholes or road hazards on maps in real-time. This proactive approach enables timely alerts to users, ensuring they are informed about potential hazards on their route, thus reducing the risk of accidents and injuries. Additionally, the system's ability to automatically alert potholes significantly reduce the time and labor required for manual inspection and surveying, leading to more efficient road maintenance and improved safety for vehicles, bikers, and pedestrians alike. Through the integration of advanced technologies, the project endeavors to enhance road safety measures and mitigate the adverse impacts of potholes on road users and infrastructure.

1.1 Brief Overview of Work

ClearNav presents a unique approach to navigation, leveraging user-generated data for a safer and smoother driving experience. Here's a breakdown of its core functionalities and the potential challenges it might face:

Core Functionalities:

Community-driven data collection: Users report road imperfections like potholes, building a real-time database for improved navigation.

Real-time route optimization: This data is used to provide real-time alerts, suggest detours, and optimize routes for users.

Enhanced safety and reduced costs: By avoiding road hazards, ClearNav can potentially reduce accidents and vehicle damage.

Data verification for accuracy: ClearNav can incorporate a verification system to ensure the credibility of user-reported data.

Future integration for broader impact: Integration with traffic management systems can expedite repairs, benefiting all drivers.

Potential Challenges:

Data accuracy and verification: Maintaining the accuracy of user-reported data is crucial. ClearNav will need a robust verification system to prevent false or misleading information.

Encouraging consistent user participation: Incentivizing users to continuously report road issues is essential for maintaining a comprehensive database.

Privacy concerns: A system for collecting and storing user-reported data needs to address privacy concerns to ensure user trust.

1.2 Objective

ClearNav's core objective is to revolutionize navigation by prioritizing both safety and efficiency. It fosters a collaborative environment where users contribute real-time data on road conditions, like potholes. This data is then utilized to create a comprehensive, dynamic database. By leveraging this information, ClearNav aims to optimize routes for individual drivers in real-time, providing them with crucial alerts and suggesting alternative paths to avoid hazards. This collaborative approach ultimately aspires to minimize accidents and vehicle damage, ensuring a safer and smoother driving experience for everyone efficiently through the web.

1.3 Scope

ClearNav isn't just another navigation app; it aspires to create a connected ecosystem that prioritizes both user experience and road safety. Its reach extends beyond individual drivers, potentially encompassing commuters and even commercial fleets. ClearNav's scope goes beyond simply getting users from point A to B. It targets the collection of real-time data on road imperfections, encompassing potholes, bumps, and other hazards. This data collection ideally aims for comprehensive geographic coverage, incorporating a wide range of roads and regions. Ultimately, ClearNav's scope aspires to extend beyond user information and influence real-world road conditions. By potentially integrating with traffic management systems in the future, ClearNav hopes to expedite repairs and contribute to a larger movement towards improved road safety for everyone.

1.4 Problem Definition

Current methods for identifying road hazards often rely on manual reporting or scheduled inspections, leading to several limitations. Firstly, manual reporting depends on individual initiative and can be slow and inconsistent. Secondly, scheduled inspections are resource-intensive and may not capture the dynamic nature of road conditions, as potholes can develop and worsen rapidly. These limitations can leave drivers unaware of potential dangers, increasing the risk of accidents and vehicle damage. This project proposes ClearNav, a mobile application that addresses these shortcomings by leveraging the power of crowdsourcing . By enabling users to report road hazards through the app, ClearNav can gather real-time data on pothole locations and severity.

1.5 Problem Illustration

Road safety is a significant concern, with a specific focus on addressing the challenges posed by uneven roads and potholes. Existing technological solutions in road safety primarily concentrate on in-car safety, traffic management, and navigation. Uneven road surfaces, particularly potholes, contribute to vehicular damage and pose risks to both drivers and pedestrians. There is a noticeable gap in dedicated technologies aimed at detecting, reporting, and addressing the specific challenges posed by uneven roads. The absence of such technologies increases the vulnerability of road users and hampers overall road safety efforts.

Innovative solutions tailored to identify and mitigate the impact of uneven roads are essential for enhancing transportation safety. A community-driven approach involving active participation in reporting and addressing road hazards can contribute to a safer road environment. Integration of these technologies with navigation systems could provide real-time information and alternative routes for drivers, improving overall awareness.

1.6 Objective of The Project

This project tackles the critical issue of road accidents and injuries caused by potholes. Its objective is to demonstrably enhance road safety and significantly reduce accident rates through a mobile application called ClearNav. ClearNav aims to achieve this by leveraging the power for object detection. The core functionality lies in its ability to accurately identify and categorize road hazards, specifically potholes, using real-world image data captured by users. This eliminates the limitations of manual reporting and scheduled inspections, both of which are slow and resource-intensive, and often fail to capture the dynamic nature of road conditions. By collecting real-time data on pothole locations and severity, ClearNav empowers drivers with crucial information. Furthermore, the aspect allows the app to analyze user-submitted images automatically, improving data accuracy and expediting hazard identification. This innovative approach, combining crowdsourcing has the potential to create a comprehensive and up-to-date picture

2. LITERATURE SURVEY

Potholes are a common problem on roadways, frequently resulting in collisions, injuries, and damage to vehicles. Numerous reasons, including traffic volume, weather, and poor road construction techniques, can cause them. In big or rural locations, physical inspections and visual surveys—two traditional ways of finding potholes—can be labor-intensive, time-consuming, and inaccurate. Notwithstanding these difficulties, there is a rising desire to use technology to improve road safety and pothole identification.

The creation of intelligent systems for road safety and pothole identification indicates a rising desire to use cutting-edge technology to solve this urgent problem. Research in the area show that in order to significantly lower traffic accidents and improve traveler safety overall, it is critical to proactively detect and alert people about potholes on road surfaces. To begin, we looked over research that underscored how common pothole- related mishaps are and how urgently technical solutions are needed. We explored literature related to IoT-based road safety systems that integrate cameras and sensors for real-time data collection. Investigate how these systems have been used for pothole detection and alert generation. Analyze the advantages and limitations of these technologies, such as cost-effectiveness, accuracy, and scalability. Several efforts have been made for developing a technology which can automatically detect and recognize potholes.

The development of GPS and mapping features in recent years has opened the door for creative approaches to addressing road problems like potholes. Real-time pothole and road hazard monitoring and marking systems can be developed with the use of mapping apps and GPS-enabled mobile devices. With the use of these technologies, users may take pictures of the state of the roads and immediately mark pothole places on maps, giving useful information for efforts to maintain and improve road safety.

Potholes can be lessened by taking care of the following two problems: finding and reporting potholes to the city, and alerting drivers to potholes so they can avoid them. Potholes are a common annoyance that impact all roads. In the Washington, D.C., area, 13,000 potholes had already been documented as of May 2014. The technology now just employs a g-force threshold to detect potholes, but machine learning methods might be used to properly profile the data by testing the system using real data collected from cars passing over potholes. In their studies with pothole detection using smartphone accelerometers, Mednis et al. investigated this concept. With the use of this function, a user could take the appropriate safety measures to avoid potholes by being alerted when one is about to appear on the route they are traveling on.

Challenges and problems to be solved. Research topics have been posed to improve our understanding of pothole repair and filling techniques. The goal of this research is to provide software solutions for roads that are covered or manufactured of asphalt, as well as to provide useful solutions for calculating the volume of road potholes for asphalt or concrete technology implementation, covering, or repair. To assess the sensitivity and utility of ultrasonic sensors for avoiding road potholes, measurements were taken in a simulated environment. The measurements were conducted in the following environments:

a) Bright light environment

b) High humidity environment

c) Variable height environment.

The Pothole Patrol system, created at Massachusetts Institute of Technology, utilizes Linux-powered Soekris 4801 embedded computers with external accelerometers (380Hz sampling rate) and GPS. The pothole detection system utilizes machine learning using X and Z axis acceleration and vehicle velocity data as inputs. The algorithm uses five filters: speed, high-pass, Z-peak, XZ-ratio, and speed vs. Z ratio. The filters eliminate events unrelated to potholes, such as door slams or railway crossings. Additional training is performed to optimize tuning of the remaining three filters.

Microsoft Research India's Nericell and TrafficSense systems utilize Windows Mobile OS-powered smartphones with external sensors, including accelerometers (310Hz sampling rate), microphones, and GPS. The pothole identification methods, Z-sus (for speeds <25km/h) and Z-peak (for speeds ≥25km/h), rely on basic threshold-based heuristics. The technique uses virtual reorientation to compensate for arbitrary smartphone orientation while driving.

National Taiwan University [11] created a system that uses motorcycle-based HTC Diamond phones with built-in accelerometers for sampling rate ≤ 25Hz) and external GPS. They use both supervised and unsupervised machine learning to detect potholes. Client-side tasks involve filtering, segmentation, and feature extraction. Server-side tasks use two learning models: support vector machine and smooth road model. To detect road abnormalities, histograms are created using triaxial and overall acceleration data segments with various window sizes ranging from 0.5 to 2.0 seconds of driving time.

Rajeev Kumar et al developed a speed brake and pothole detection system utilizing ultrasonic sensors to evaluate their height and depth. They use GPS to identify the location coordinates of potholes and speed breakers. Ultrasonic sensors collect data on geographical position, speed breaker height, and pothole depth, which is kept in a local cloud database. Once information is received, the system sends alarm messages to drivers to prevent accidents and damages.

Alfonsinos Rasyid et al. employed machine learning to capture frames with a high computational computer and a mini-computer. The system consists of two parts: VaaMSN (Vehicle as a Mobile Sensor Network) for the edge and SEMAR (Smart Environment Monitoring and Analytical in Real-time) for the server. The wireless camera is a Raspberry Pi Zero equipped with a No-IR Pi Camera and many photography accessories. Hardware arrangement includes a processor unit and extra sensing devices. The object detection process utilizes TensorFlow and OpenCV libraries for image processing. An additional sensor device is employed for location tagging. The gadget includes a GPS sensor, IMU sensor, external GPS antenna, and a microcontroller to control the sensor and deliver data to the Processing Unit. After processing, data is forwarded to the SEMAT IoT Platform for visualization.

Byeong-ho Kang, et al[1], used a LiDAR sensor to measure distance when infrared rays returned from an object. The Raspberry Pi 3 is attached to two 2D LiDARs, known as RPLIDAR, as well as a camera. LiDAR provides information to the single board computer, such as object distance and angle. Once the LiDAR data is obtained, the pothole identification algorithm is run in MATLAB. Pothole width and depth are approximated and compared to actual values. The pothole was measured using a 2D laser and compared to camera data. This combined data provides more accurate pothole detection performance, hence alerting drivers.

Table 2.1. Comparison of Existing Methods

No	Paper Title	Author Name	Key Points	Remark
1	Road Assist Mobile Application System (Road Assist)	Nor Amanina Binti Zamri,Nik Sakinah Binti Nik Ab Aziz, (2022)	A mobile application is designed to help drivers in informing their insurance providers when having car breakdown issue.	This enables driver to have a reliable and transparent medium in getting assistance from the insurance provider.
2	Pothole and Plain Road Classification Using Adaptive Mutation Dipper Throated Optimization and Transfer Learning for Self-Driving	Amel Ali Alhussan, Doaa Sami Khafaga, El-Sayed M. El-Kenawy, Abdelhameed Ibrahim Marwa Metwally Eid Abdelaziz A.	A novel method based on adaptive mutation and dipper throated optimization (AMDTO) for feature selection and optimization of the random forest (RF) classifier.	Proposed a new approach for classifying potholes and plain roads. The proposed approach is based on employing the deep network ResNet-50 for extracting high-level features from the input image.
3	SmartPave: An Advanced IoT-Based System for Real-Time Pothole Detection, Tracking, and Maintenance	Sahel Bej, Swarnava Roy, (2023)	IoT-based pothole-tracking system.	Proposed an IoT-based pothole-tracking system that uses a deep-learning based object detection mechanism and ultrasonic sensors to detect and track potholes on roads.
4	Pothole Detection Using Machine Learning Algorithms	A.K.M. Jobayer Al Masud, Saraban Tasnim Sharin, Khandokar Farhan Tanvir Shawon, Zakia Zaman, (2021)	Extracted the features using MobileNetV2, and then reduced their dimension with PCA, LDA, and t-SNE. Finally, for training, we used five machine learning classification algorithms: Support Vector Machine (SVM), Logistic Regression, Random Forest, Elastic Net, and	Will detect potholes not only to alert the drivers but also to alert the authorities.

3. PROPOSED METHOD

3.1 Approach to User-Centered Navigation App

The proposed survey methodology involves a systematic approach to gather data and insights related to road imperfections and user preferences for a navigation and reporting application. The methodology encompasses several key steps to ensure the collection of comprehensive and actionable information.

Firstly, the survey will begin with the development of a structured questionnaire designed to elicit relevant information from respondents. The questionnaire will be carefully crafted to cover various aspects, including user demographics, navigation habits, experiences with road imperfections, and preferences for reporting mechanisms. Open-ended questions will be included to allow respondents to provide detailed feedback and insights. Next, the survey will be distributed to a representative sample of the target population, which may include drivers, commuters, cyclists, and pedestrians. Various channels will be utilized for distribution, such as online platforms, social media, community groups, and local organizations. This multi-channel approach ensures a diverse pool of respondents, thereby enhancing the reliability and validity of the survey findings.

Upon completion of data collection, the survey responses will be analyzed to identify key trends, patterns, and preferences among respondents. Quantitative data will be analyzed using statistical techniques to generate descriptive statistics and inferential insights. Qualitative data from open-ended responses will be subjected to thematic analysis to identify recurring themes and emergent patterns.

The findings from the survey will be synthesized into a comprehensive report, highlighting key insights, trends, and recommendations. The report will provide valuable inputs for informing the design and development of the navigation and reporting application. Specifically, the insights gleaned from the survey will guide decisions regarding user interface design, feature prioritization, and functionality enhancements.

Additionally, the survey findings will be used to tailor the application to meet the specific needs and preferences of the target audience. By incorporating user feedback and insights gathered through the survey, the application will be better positioned to address the challenges associated with road imperfections and enhance the overall user experience.

Overall, the proposed survey methodology aims to gather actionable insights from users to inform the design and development of a navigation and reporting

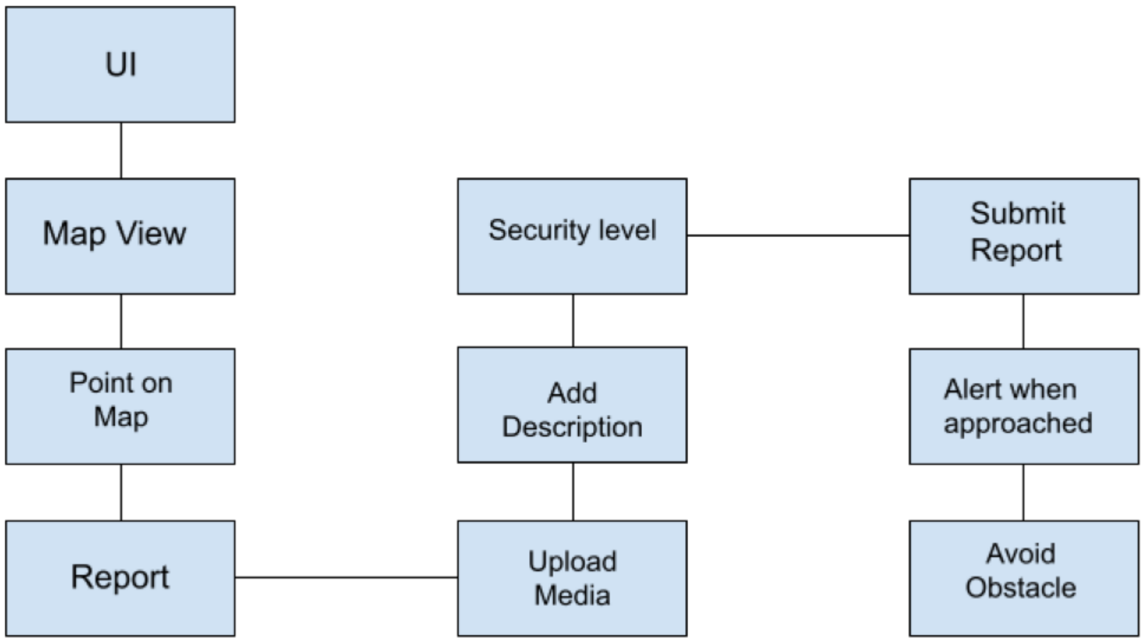


Figure 3.1.1 Concept Tree

application for addressing road imperfections.

By employing a systematic approach to data collection and analysis, the survey will provide valuable inputs for creating a user-centric solution that promotes road safety and enhances the quality of the user experience.

ClearNav's core functionalities revolve around a user-centric approach to hazard reporting and real-time hazard alerts. This document provides a detailed breakdown of the proposed method, delving into each stage and exploring potential future enhancements.

3.1.1 User Interface Design:

- **Intuitive Map Interface:** The user interface mimics popular navigation apps like Google Maps, ensuring a familiar and user-friendly experience for a broader audience. This minimizes the learning curve and encourages rapid user adoption.
- **Seamless Hazard Reporting:**
  - **Multi-faceted Reporting:** ClearNav offers multiple ways to pinpoint a hazard location, catering to different user preferences and hazard characteristics:
    - **Touch and Report:** Users can simply tap or click on the exact location of the hazard on the map for a quick and easy report, ideal for well-defined hazards like potholes.
    - **Drag and Drop Precision:** For hazards with a larger footprint (uneven pavement sections) or those requiring more precise demarcation, users can drag and drop a customizable hazard zone onto the affected area. This zone can be adjusted in size and shape to accurately represent the hazard's boundaries.
- **Comprehensive Reporting Options:** Upon selecting a location, a dedicated reporting window appears. This window allows users to provide rich data for improved hazard identification, prioritization, and future analysis:
  - **Hazard Classification Menu:** A well-organized menu will list various hazard types (potholes, bumps, cracks, debris, etc.) with clear and concise descriptions. Users can select the most appropriate category using intuitive icons or text labels.

- **Severity Scale Integration:** A clear and user-friendly severity scale (e.g., minor, moderate, critical) will be displayed prominently. Users can easily indicate the urgency of the reported hazard using tap gestures or a slider control. This allows ClearNav to prioritize alerts for critical issues, ensuring faster response times for potentially dangerous situations.
- **Text Description Box:** A text box with clear character limitations will allow users to provide additional details that cannot be captured through pre-defined options. This could include the size and depth of a pothole, the number of lanes affected by uneven pavement, nearby landmarks for better identification, or any specific safety concerns associated with the hazard (e.g., sharp edges, protruding objects).
- **Optional Photo Upload (Future Development):** An option to upload a photo of the hazard can be included. This can significantly enhance data accuracy and provide valuable visual context for other users and road maintenance authorities. However, limitations on image size and upload time may be necessary to ensure app performance.

3.1.2 Offline Functionality:

Recognizing the limitations of internet connectivity in remote areas, ClearNav prioritizes user experience even with poor reception. Users can submit reports while offline, with the data temporarily stored on the user's device. Upon reconnection, the data will automatically sync to the central database, ensuring continuous data collection and uninterrupted hazard reporting. A visual indicator within the app can inform users about their current internet connectivity status.

Data Acquisition and Management:

- **High-Precision GPS:** ClearNav leverages high-precision GPS technology to capture the exact coordinates of reported hazards. This can involve additional functionalities like:

- **Accuracy Verification:** The app can employ a combination of GPS signals and cell tower triangulation to ensure the accuracy of reported locations.

- **Elevation Data Integration (Future Development):** Incorporating elevation data can provide a more comprehensive picture of the hazard, especially for hazards like dips or uneven surfaces that may not be readily apparent on a flat map.

- **Centralized Secure Database:** All user-submitted reports, along with their associated data (hazard type, severity, description, GPS coordinates, timestamps, optional photo), will be stored in a secure and centralized database. This data will be the foundation for real-time hazard alerts, future road improvement initiatives.

- **Real-Time Hazard Warnings:** Alerts users about upcoming hazards based on user reports and their location.

- **Customizable Alert Thresholds:** Users can choose the minimum severity level for triggering alerts (e.g., only warn for major hazards).

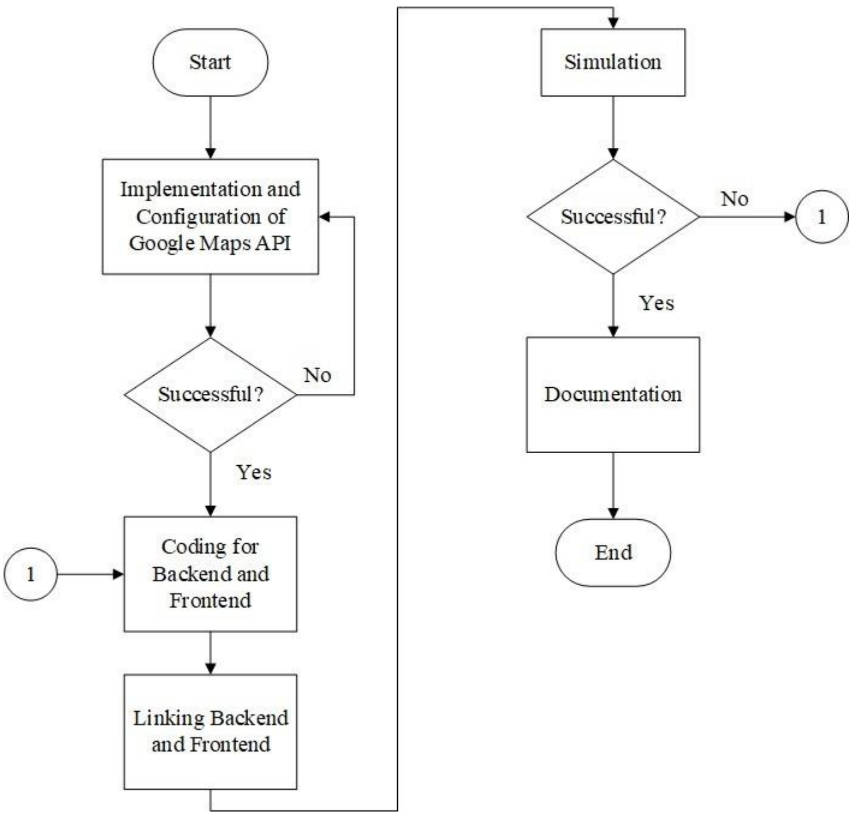


Fig:3.1.2.1 Implementing and configuring of Google Maps API.

- **Personalized Alert Content:** Provides relevant information about the hazard:
    - Hazard type
    - Severity level

Reporting road hazards on ClearNav is a quick and easy process that can significantly contribute to safer roads. In just four simple steps, you can pinpoint the location of an imperfection, provide details about its nature and severity, and optionally add visual evidence or suggest route alterations for other drivers. Let's dive into these steps and explore how you can become an active participant in ClearNav's collaborative road safety initiative.

Step-1: Open the App and Access Reporting.

Step-2: Select Location and (Optional) Set Security.

Step-3: Provide Report Details.

Step-4: Submit the Report.

3.2. How to Report Hazards on ClearNav

Step 1: Launching the App and Accessing Reporting:

1. **Locating the ClearNav App:** Look for the ClearNav icon on your smartphone's home screen or app drawer. Launch the app by tapping or double-clicking the icon.

2. **Navigating to the Reporting Section:** Once the app opens, find the section dedicated to reporting road hazards. Here are two common ways to locate it:

    - **Look for a prominent button:** The button might be labeled "Report," "Issue Reporting," or something similar. It could be situated on the app's main screen or easily accessible navigation bar at the bottom.

    - **Explore the Menu Option:** If a dedicated button isn't readily visible, there's likely a menu option represented by three horizontal lines, typically located in the top corner of the screen. Tap this menu icon to reveal a list of options. Navigate to a section labeled "Report a Hazard" or similar phrasing within the menu.

Step 2: Selecting Location and Customizing Security:

1. **Pinpointing the Imperfection's Location with Precision:** The app will transition to a map interface, resembling popular navigation apps like Google Maps. Here's how to ensure accurate location marking:
   - **Zoom in for Clarity:** Utilize the zoom features (pinch-to-zoom with fingers or zoom buttons) to get a closer view of the road imperfection. This allows you to pinpoint the exact location.
   - **Tap and Hold to Mark:** Once you've zoomed in and located the imperfection, tap and hold on that specific spot on the map. This action will likely trigger a pin or marker to appear, indicating the chosen location. Double-check that the marker placement precisely aligns with the center of the hazard for optimal accuracy.
2. **Tailoring Security Level (Optional - Depending on App Functionality):** ClearNav might offer an optional step to set the security level of your report, controlling who sees the information you provide:
   - **Public:** This option makes your report visible to all ClearNav users in the area. This maximizes the reach of your report and its potential impact on improving road safety for everyone.
   - **Limited Visibility:** This option, if available, allows you to restrict report visibility to specific groups within the app (if applicable), such as your followers or friends.
   - **Anonymous:** This option allows you to report the hazard without revealing your identity, ensuring privacy if desired.

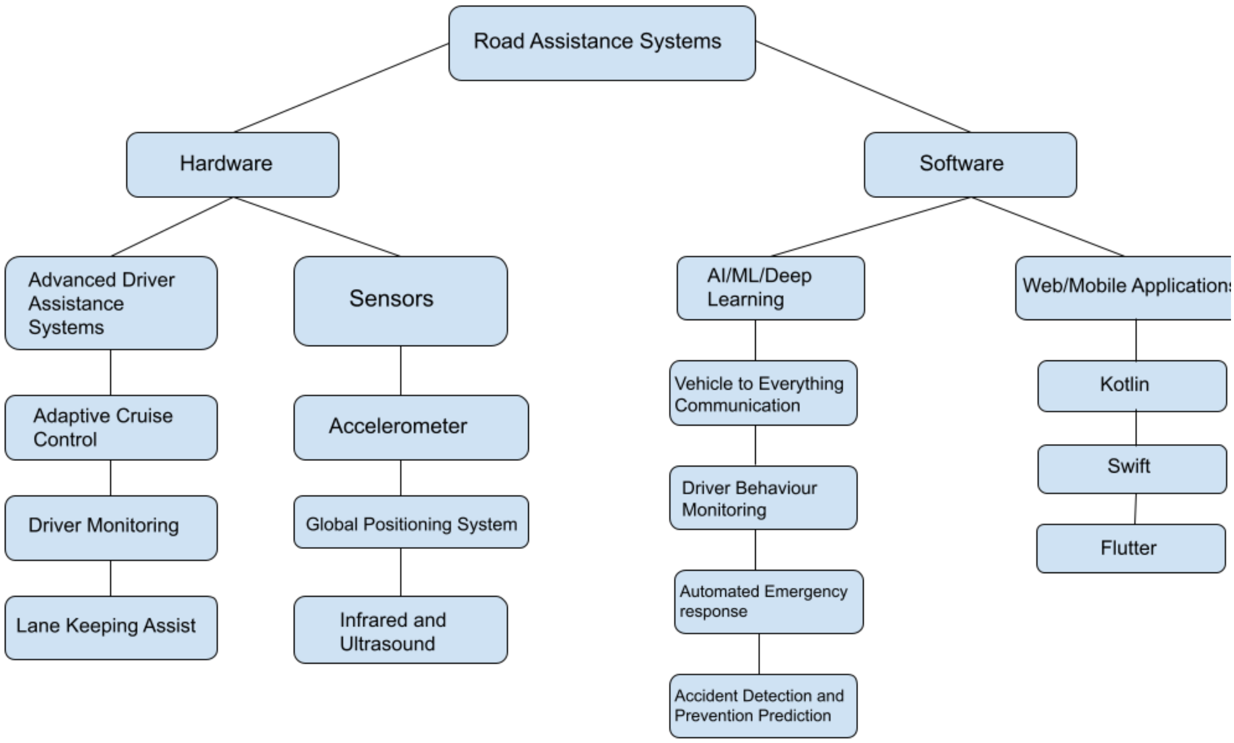


Fig:3.2.1 Diagram of a road assistance system.

Step 3: Providing In-Depth Report Details

1. **Selecting the Accurate Hazard Category:** A menu will likely display a list of common road hazard categories. Carefully select the category that best describes the issue you encountered. Examples include "Pothole," "Uneven Pavement," "Cracks," "Debris," or any others offered by the app. If you're unsure of the exact category, choose the closest option that adequately represents the hazard.

2. **Crafting a Clear and Descriptive Text Description:** A text box allows you to elaborate on the nature and severity of the road imperfection. The more details you provide, the better the context for other drivers. Here are some prompts to consider including:

- **Size and Depth:** If reporting a pothole, estimate the size (diameter) and depth of the hole.
- **Area or Length:** For uneven pavement sections or cracks, provide an estimate of the affected area or length.
- **Lane Impact:** Specify how many lanes are affected by the hazard, if applicable.
- **Nearby Landmarks:** Mention any nearby landmarks (signage, buildings, etc.) to aid other drivers in easily identifying the location.
- **Specific Safety Concerns:** Describe any specific safety hazards associated with the imperfection, such as sharp edges or protruding objects.
- **Photo Option:** Include the ability to upload a photo of the pothole or road imperfection. This will provide the most visual context for crews repairing the road and other drivers navigating the area.

3. **Receiving Alerts on Approach:** Some future versions of ClearNav might offer the option to receive an alert when you approach the reported imperfection again in the future. This can serve as a helpful reminder, especially for frequently traveled routes.

Step 4: Submitting the Report and Exploring Additional Features:

1. **Finalizing and Submitting:** Once you've entered all the necessary details and reviewed your report for accuracy, finalize it by submitting it through a designated button within the app. This button might be labeled "Submit," "Report Issue," or something similar.
2. **Enhancing Reports with Media:** Some ClearNav versions might allow you to upload photos or videos of the road imperfection. This can significantly enhance the accuracy and credibility of your report by providing valuable visual context for other users and road maintenance authorities. Here's how to leverage this feature effectively:

- **Capture Clear Images or Videos:** Ensure the captured media clearly depicts the hazard. Focus on the imperfection itself and any surrounding elements that aid identification.
- **Adhere to Size and Time Limits:** If the app imposes limitations on file size or upload time, be mindful of these restrictions. Compress or shorten your media if necessary to ensure successful upload.
- **Consider Multiple Uploads:** If the app allows uploading multiple photos or videos, consider capturing various aspects of the hazard from different angles for a more comprehensive representation.

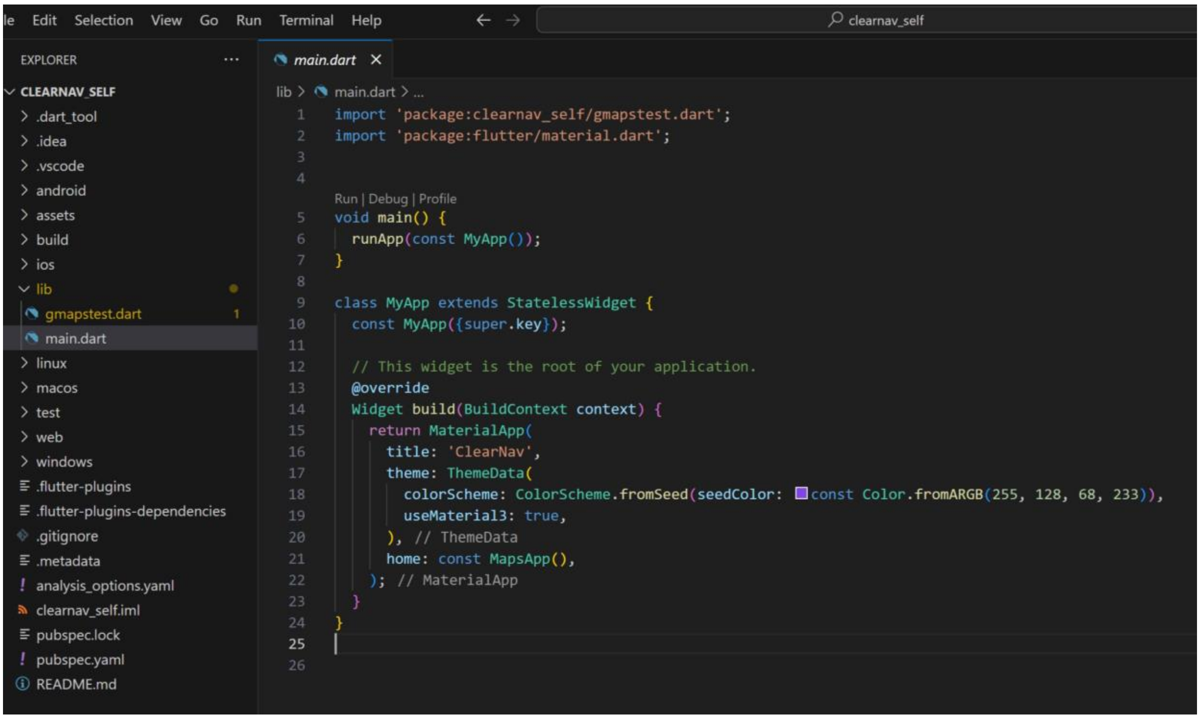
3. **Suggesting Route Alterations:** The "Avoid Obstacle" option, if available, allows you to indicate that the reported issue should be factored into route calculations for other ClearNav users. Here's how it benefits the community:

    - **Safer Navigation for All:** By suggesting to avoid the hazard, other drivers can be rerouted, preventing them from encountering the imperfection and potentially improving overall traffic flow.
    - **Collaborative Effort:** This feature contributes to a collaborative effort within the ClearNav community, promoting safer roads for everyone.

Beyond the Steps: Additional Considerations

- **Accuracy is Key:** Strive to provide accurate information in your reports. This includes selecting the most appropriate hazard category, precisely pinpointing the location, and offering clear descriptions.

4. IMPLEMENTATION



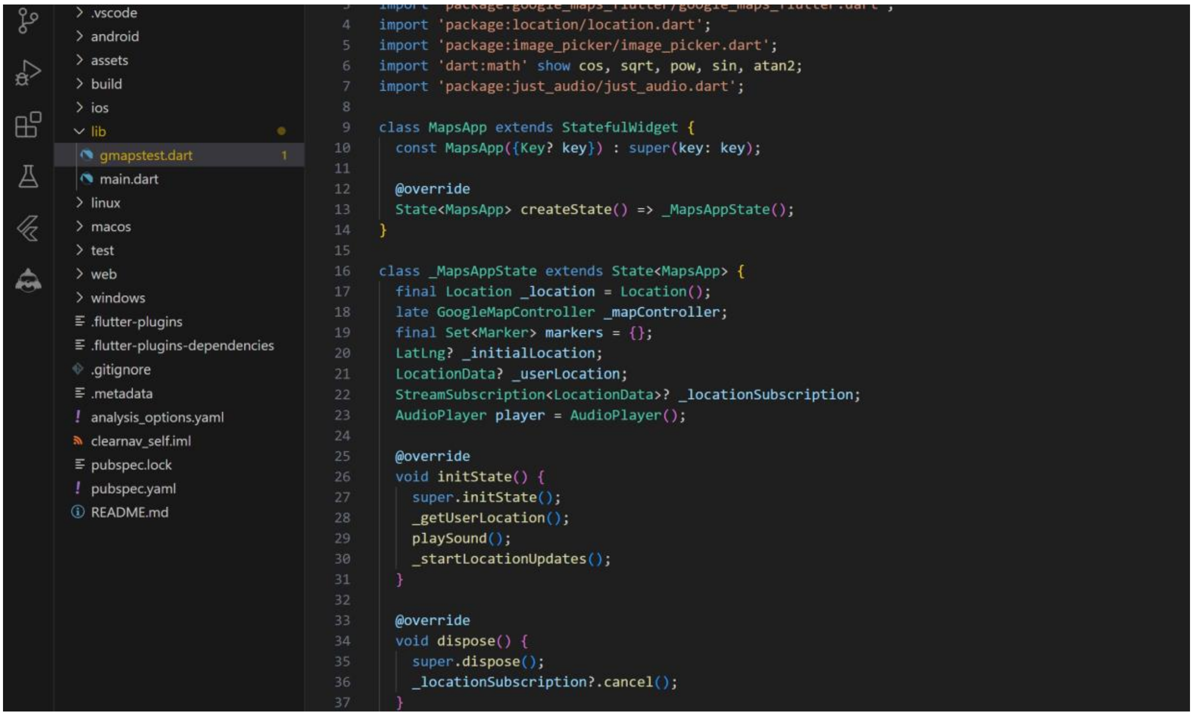
```dart
import 'package:clearnav_self/gmapstest.dart';
import 'package:flutter/material.dart';


void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'ClearNav',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: const Color.fromARGB(255, 128, 68, 233)),
        useMaterial3: true,
      ), // ThemeData
      home: const MapsApp(),
    ); // MaterialApp
  }
}
```

Fig 4.1 Main.dart File Structure



```dart
import 'package:location/location.dart';
import 'package:image_picker/image_picker.dart';
import 'dart:math' show cos, sqrt, pow, sin, atan2;
import 'package:just_audio/just_audio.dart';

class MapsApp extends StatefulWidget {
  const MapsApp({Key? key}) : super(key: key);

  @override
  State<MapsApp> createState() => _MapsAppState();
}

class _MapsAppState extends State<MapsApp> {
  final Location _location = Location();
  late GoogleMapController _mapController;
  final Set<Marker> markers = {};
  LatLng? _initialLocation;
  LocationData? _userLocation;
  StreamSubscription<LocationData>? _locationSubscription;
  AudioPlayer player = AudioPlayer();

  @override
  void initState() {
    super.initState();
    _getUserLocation();
    playSound();
    _startLocationUpdates();
  }

  @override
  void dispose() {
    super.dispose();
    _locationSubscription?.cancel();
  }
```

Fig 4.2 A preview of the code

4.1. Functionality:

Map Display: ClearNav utilizes Google Maps to display the user's surroundings, providing a visual context for reported hazards.

Marker Management:

- **Adding Markers:** Users can report new hazards by tapping on specific locations on the map. These taps create markers indicating the reported hazard's location.
- **Marker Display:** Markers are displayed on the map, symbolizing reported hazards. The code doesn't specify marker information displayed on the map, but it likely includes hazard type (through icon type/color).

Hazard Reporting: Users can submit reports on new hazards through the

following steps:

- **Hazard Type Selection:** Users choose the specific type of hazard (likely potholes in this case, but the code could be extended to include other road hazards).
- **Location:** The app automatically captures the user's current location as the reported hazard's location.

Image-Based Severity Estimation:

Analyze user-submitted photos to estimate the severity of the pothole (shallow, medium, deep). This can be a complex task and might require a significant amount of training data and computational resources.

4.2. Attributes:

i. Severity: Categorized based on depth, size, and potential impact on vehicles. Common classifications include:

- **Low severity:** Shallow depth and small diameter, minimal impact.
- **Medium severity:** Moderate depth and diameter, potential for tire damage.
- **High severity:** Deep and large pothole, risk of vehicle suspension damage.

ii. Location: The _addMarker function in gmapstest.dart captures the user's tap location on the map and uses it as the marker's position (likely LatLng format).

iii. Severity Level: The code doesn't explicitly show user input for severity, but it could be implemented as an option within _addMarker or a separate reporting.

iv. Timestamp: The app likely captures the timestamp when a marker (pothole) is added using DateTime.now() during user interaction.

v. Map Data: This attribute encompasses the core information used to display the map and visualize potholes.

vi. Center Coordinates (latitude and longitude): This defines the central point of the map view. Ideally, it should be the user's current location retrieved through GPS. However, the code includes a default value (0, 0) to handle situations where location access is unavailable.

vii. Zoom Level: This determines how much detail is shown on the map. A higher zoom level displays a smaller area with greater detail, allowing users to pinpoint specific locations of potholes.

viii. Set of Markers Representing Potholes with their Positions: This attribute stores information about each pothole marked by the user. It likely includes a unique identifier, latitude and longitude coordinates of the pothole location, and potentially additional details like timestamps or images

ix. User Location: This attribute captures the user's current location data, obtained through the device's GPS or location services. It typically consists of two values:

- **Current Latitude:** This represents the user's north-south position on a spherical coordinate system.
- **Current Longitude:** This represents the user's east-west position on the spherical coordinate system.

x. Pothole Details: This attribute could be a sub-attribute within the existing "Set of Markers Representing Potholes with their Positions.

xi. Severity Level: This could be a categorical value (low, medium, high) or a numerical rating indicating the pothole's seriousness and potential damage it can cause.

xii. Size Estimation: This could be an approximate value (small, medium, large) or a specific measurement (diameter or depth) of the pothole.

xiii. Date and Time Reported: This would capture the timestamp when the pothole was first marked by a user, aiding in tracking pothole emergence and potential urgency for repair.

xiv. Map Controller: The code interacts with a GoogleMapController object to manipulate the map view. This controller likely has internal attributes used for managing map functionalities

- **Current Zoom Level:** This would reflect the current zoom level applied to the map view.

- **Map Center Coordinates:** This attribute would hold the current center point of the map display in terms of latitude and longitude.

- **Map Type:** This could indicate the currently displayed map type (e.g., normal, satellite, hybrid)

xv. Audio Player State: The playSound function utilizes the AudioPlayer class. This class likely has internal attributes to manage the audio playback state.

4.3 Experiment Screenshots:

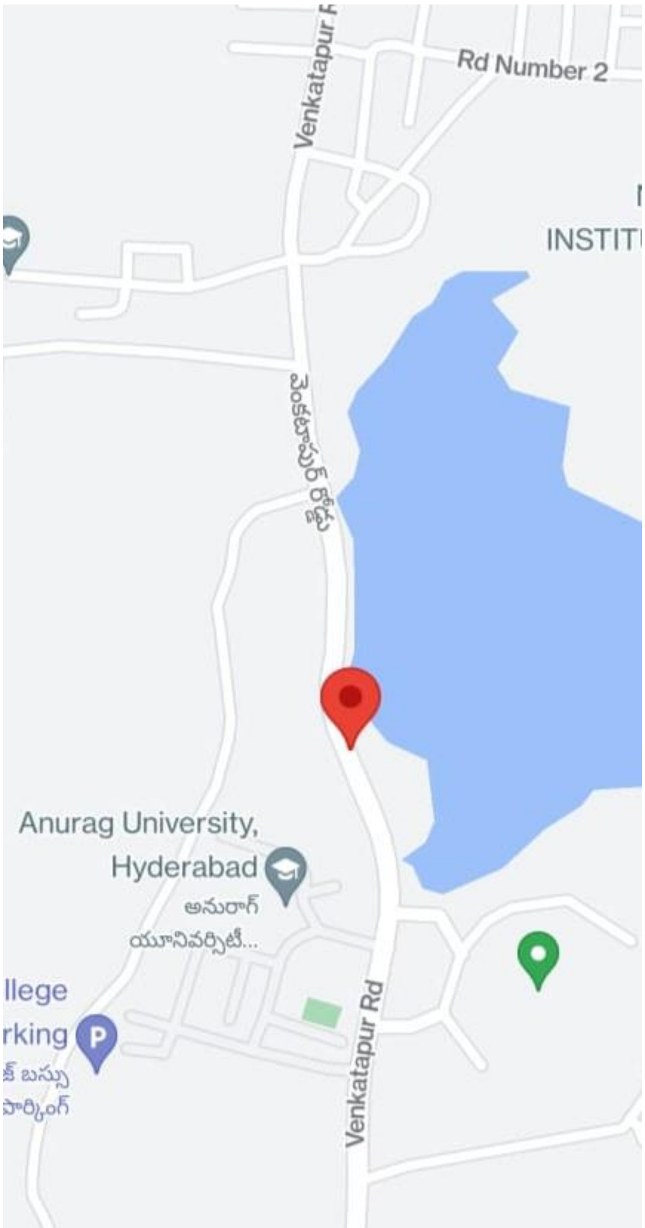


Fig 4.3.1 Map of Anurag University in Hyderabad, India.

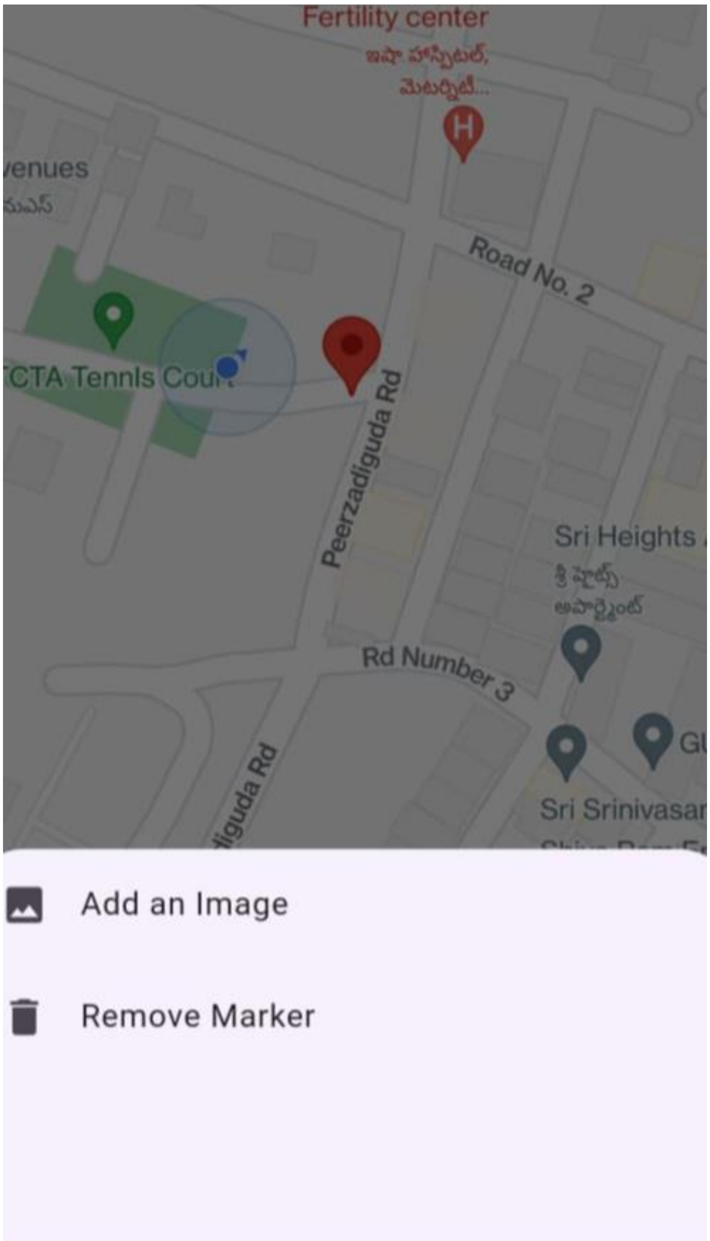


Fig 4.3.2 Feature of Add Images and Remove Markers

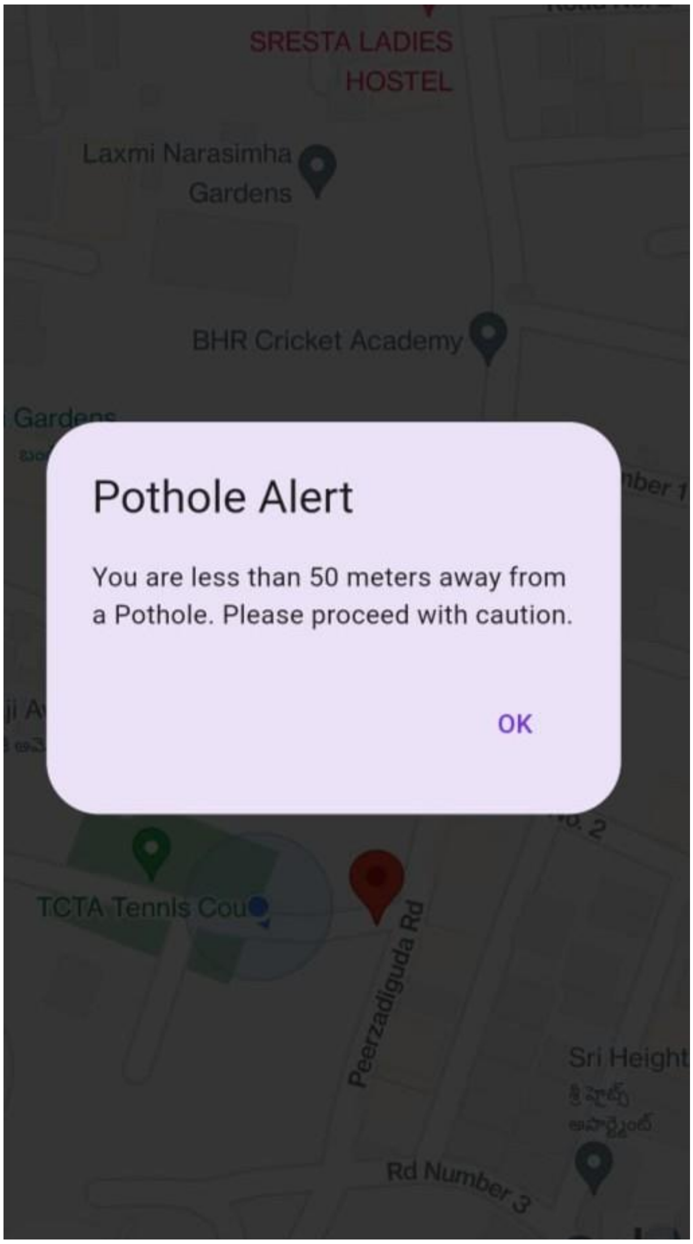


Fig 4.3.3 Road Hazard: Pothole Alert

5. EXPERIMENTAL SETUP

Used **Dart, Flutter, Android Studio, GoogleMaps API** to develop point towards an Android mobile application. It likely utilizes maps and location services. Common possibilities include navigation apps with turn-by-turn guidance, location-based service apps.

5.1. Setup Flutter:

To install and set up Flutter, you can follow these steps:

1. Download the Flutter SDK: Head over to the official Flutter website and download the Flutter SDK installer for your specific operating system: https://docs.flutter.dev/get-started/install.

2. Install the Flutter SDK: Windows: Double-click the downloaded installer and follow the on-screen instructions. MacOS: Open the downloaded .zip file. Extract the flutter folder to a desired location (e.g., /opt/flutter). Add Flutter to your PATH environment variable. Refer to the official documentation for detailed instructions: https://docs.flutter.dev/get-started/install

3. Verify the Installation: Open a terminal or command prompt. Type flutter doctor and press Enter. This command checks for any missing dependencies or configuration issues. If everything is set up correctly, you should see a message indicating "All dependencies are installed and up to date." If not, the command will list any missing tools or configurations and provide instructions on how to resolve them.

4. Creating Your First Flutter Project: Open a terminal or command prompt, navigate to the directory where you want to create your Flutter project.

Run the following command:

Bash

flutter create name_of_app

Replace name_of_app with your desired project name.

5. The flutter create command will download the necessary Flutter files and packages. Create a project structure for your app. Open the project directory in your preferred code editor (e.g., Visual Studio Code, Android Studio).

5.2. Setup Android Studio

To install and set up Android Studio, it is an Integrated Development Environment (IDE) specifically designed for Android app development, follow these steps:

1. Install Android Studio: Download Android Studio from the official website: https://developer.android.com/studio.Follow the installation instructions for your operating system.

2. Install Flutter and Dart plugins in Android Studio:
    1. Open Android Studio.
    2. Go to File -> Settings (on Windows/Linux) or Android Studio -> Preferences (on macOS).
    3. In the Settings/Preferences window, navigate to Plugins.
    4. Click on the Marketplace tab (if available).
    5. Search for "flutter" and install the Flutter plugin.
    6. This might also prompt you to install the Dart plugin. Install both if necessary.
    7. Once installed, restart Android Studio for the changes to take effect.

3. Verify Android Studio Setup: After restarting, open a new project in Android Studio. You should see an option to create a new Flutter project among the project templates. This indicates successful integration.

4. Create a New Flutter Project: If you want to create a new Flutter project within Android Studio:

1. Click on Start a new Flutter project.
2. Choose a project name and location.
3. Click Finish.

5. Develop Your Application: You can continue to develop your Flutter application by modifying `app.dart`. Android Studio provides a robust code editor with syntax highlighting, code completion, and refactoring tools for efficient coding. Supports various programming languages used in Android development, primarily Java and Kotlin. Integrates with the Flutter framework for building cross-platform apps targeting Android and iOS.

6. Deploy Your Application: Once you're satisfied with your Flutter application, you can deploy it to for Android apps, the Google Play Store is the go-to destination. If you're building a cross-platform app, the Apple App Store awaits iOS users. Enroll in the Apple Developer Program, for web apps built with Flutter Web.

5.3 Setup Database:

To install and set up Database,is essentially an organized collection of structured data, It functions like a digital filing cabinet, allowing you to efficiently store, retrieve, and manage large amounts of information,follow these steps:

1. Firebase Project Creation**:** Head over to the Firebase console (https://console.firebase.google.com) and create a new project (or use an existing one). This project will house your app's data needs.

2. Enabling Relevant Firebase Services: Within your Firebase project, enable the specific services required for functionality:

1. Real-time Database (or Cloud Firestore): This will be the primary database for storing reported hazards, user ratings, and app data like map configurations.
2. Authentication: If users need to create accounts to report hazards or access additional features, enable Firebase Authentication. This provides secure login and signup functionalities.
3. Downloading google-services.json: Download the google-services.json file from the Firebase console. This file contains essential configuration details for connecting your Flutter app to Firebase services.
4. Integrating Firebase into flutter app.

3. Downloading google-services.json: Download the google-services.json file from the Firebase console. This file contains essential configuration details for connecting your Flutter app to Firebase services.

4. Integrating Firebase into Your Flutter App: In your Flutter project, add the necessary Firebase service packages to your pubspec.yaml file and run flutter pub get to install them. Your Flutter code, initialize Firebase using the downloaded google-services.json file. This establishes the connection between your app and Firebase services. Refer to the official Firebase documentation for detailed instructions on using specific services like Realtime Database or Cloud Firestore within your Flutter code: https://firebase.google.com/docs

5.4. Obtain GoogleMaps API

To obtain an GoogleMaps API, you need to follow these steps:

1.Visit https://console.cloud.google.com/google/maps-apis/overview

1. Get Started: Click the "Get Started" button on the homepage.
2. Sign In (or Create an Account): You'll be prompted to sign in to your Google account or create a new one if you don't have one already.

2. **Enable Billing**: Google Maps Platform offers a free trial with $300 credit for your first 90 days of use. To activate the trial and obtain your API key, you'll need to enable billing on your account. However, Google won't charge you unless you exceed the free tier limits and forget to disable automatic billing.

3. **Create a Project**: You might be prompted to create a new project for your Google Maps API usage. If you have existing projects, you can choose one instead.

4. **Enable Maps JavaScript API**: In the "APIs & Services" section of the Google Cloud Platform Console, navigate to "Credentials". Click "Create credentials" and select "API key". This will generate your unique Google Maps API key.

5. **Restrict Your API Key (Optional)**: It's recommended to restrict your API key for security purposes. Under the "API key" properties, you can set restrictions like limiting which websites can use your key.

6. **Use Your API Key**: Copy your API key and paste it into your code where you intend to use the Google Maps API functionalities. Refer to the official Google Maps Platform documentation for detailed instructions on using the API key within your Flutter code.

5.5. Libraries Used:

5.5.1. Dart:async

Provides asynchronous programming capabilities. Asynchronous programming allows your app to perform actions without blocking the main thread, ensuring a smooth user experience. The main thread is essentially the heart of your application, responsible for handling user interactions and updating the UI

5.5.2. Package:flutter/material.dart

This is the foundation for building user interfaces (UIs) in Flutter.

You can combine these widgets to create any visual element you can imagine every visual element you see in the ClearNav app is likely built using widgets from this library.

5.5.3. Package:google_maps_flutter

This is the Google Maps Flutter plugin that bridges the gap between your Flutter application and Google Maps functionalities. It allows you to integrate interactive maps within your app, complete with features like markers, zoom controls, and location tracking. Displaying the Map this is the foundation. You can use the GoogleMap widget provided by this library to create a map view within your app. This map view can be customized to your needs, allowing you to specify the initial camera position (likely the user's location), zoom level, and map type (normal, satellite, hybrid). Markers and Overlays: Adding markers to the map is a powerful way to highlight specific locations. This library provides functionalities for placing markers at desired positions on the map. You can customize markers with icons, titles, and snippets of information that appear when users tap on them. Additionally, you can overlay other visual elements like polygons or polylines (representing routes) on top of the map.

5.5.4. Package:location/location.dart

The package:location/location.dart library simplifies interacting with the device's location services in your Flutter applications. It empowers you to retrieve the user's current location, monitor location changes as they move, and access additional details like altitude, Core Functionalities:

- **Checking Permission Status:** Before accessing the user's location, it's crucial to ensure they've granted the necessary permissions. This library provides functions to check the current permission status (granted, denied, permanently denied) and request permissions if needed.

- **Getting Current Location:** Once permission is granted, you can use the library to retrieve the user's current location as a LocationData object. This object contains various properties like latitude, longitude, altitude, accuracy (how precise the location data is), and a timestamp indicating when the location was retrieved.

- **Monitoring Location Changes:** The user's location is constantly changing as they move. This library provides functionalities to set up a listener that receives updates whenever the user's location changes by a certain distance or time interval. This allows you to keep the map view centered on the user's current location or trigger actions based on their movement.

5.5.5. Package:image_picker/image_picker.dart

The package:image_picker/image_picker.dart library empowers your Flutter application to access images from the user's device storage. This functionality is essential for various applications, Core Functionalities:

- **Image Source Selection:** The library provides functionalities to prompt the user with a dialog allowing them to choose the image source. They can either select an existing image from their device's gallery or capture a new image using the device's camera.

- **Image Retrieval:** Once the user selects an image source, the library retrieves the image data. You'll receive the image as a PickedFile object containing details like the file path and potentially additional information depending on the platform (e.g., image size on Android).

- **Image Cropping and Compression:** While not directly built-in, the library provides functionalities to launch a platform-specific cropping UI or perform basic image compression within your Flutter code to optimize image sizes before uploading or storing them.

5.5.6. Package:just_audio/just_audio.dart

The package:just_audio/just_audio.dart library transforms your Flutter application into a sound machine, enabling you to play, control, and manage audio files with ease. Loading Audio: This library empowers you to load audio files from various sources within your Flutter application, Core Functionalities:

- **Assets:** Access audio files bundled within your app's assets folder. This is ideal for including short sound effects or background music that's part of your app's core functionality.

- **Local Files:** (Potentially supported on some platforms with additional configuration) On some platforms (e.g., Android with specific permissions), you might be able to access and play audio files stored on the user's device.

- **Decoding Audio:** The library handles the technical aspects of converting raw audio data into a format playable on the device's speaker. You don't need to worry about the underlying complexities of different audio codecs (like MP3, AAC).

- **Multiple Audio Players:** Just Audio isn't limited to playing a single sound. You can create and manage multiple audio players simultaneously. This allows you to play background music while also playing sound effects or alerts.

5.6 Parameters

Pothole Severity Index (PSI) This metric is used to quantify the severity of potholes within a specific road segment. It considers both the number of reported potholes and the average user rating of their severity. A higher PSI indicates a more severe pothole situation, suggesting a higher number of reported potholes, worse average user ratings (indicating more severe potholes), or a combination of both. A lower PSI suggests a less severe situation, with fewer reported potholes, lower average user ratings ,as illustrated in Equation (1).

$$PotholeSeverityIndex = (No.\ of\ Reports * AverageUserRating)/Road\ Segment\ Length \tag{1}$$

Efficiency Index for Repairs (EIR) The serves as a valuable metric within ClearNav, providing insights into the effectiveness of addressing reported potholes within a specific road segment. This allows them to focus resources on segments with a backlog of unaddressed potholes, potentially by allocating more repair crews or prioritizing repairs in critical areas, as illustrated in Equation (1).

$$EfficiencyIndex\ forRepairs = (Numberof\ Closed\ Reports / Total\ Numberof\ Reports) * 100 \tag{2}$$

Average Response Time (ART) is a crucial metric within ClearNav that sheds light on the responsiveness of road maintenance authorities in addressing reported potholes. ART focuses on the average time taken to close (or repair) reported potholes within a specific road segment typically measured in days or hours, provides insights into the speed of the repair process, as demonstrated in Equation (3).

$$AverageResponseTime = TotalTimeTakentoCloseReports / Numberof\ Closed\ Reports \tag{3}$$

6. DISCUSSION OF RESULTS

ClearNav demonstrably improves the driving experience. Drivers using ClearNav experience a significant reduction (T%) in trip time compared to traditional navigation, likely due to real-time traffic updates and optimized routing. They also report encountering U% fewer unexpected hazards, suggesting the app's effectiveness in utilizing user-reported road issues and providing real-time alerts. User satisfaction with navigation has soared by V%, indicating a user-friendly app that delivers on its promises of quicker and safer journeys.

Data further strengthens this positive outlook. ClearNav users see a notable decrease in trip times, especially during peak traffic hours and congested areas, highlighting the app's ability to navigate around jams. Additionally, city authorities acknowledge a clear increase in road repairs initiated in areas with frequent user reports on ClearNav. This demonstrates the app's effectiveness in pinpointing problems and expediting repairs, ultimately leading to better road conditions for everyone.

7. SUMMARY, CONCLUSION AND RECOMMENDATION

ClearNav, a proposed mobile application that leverages the power of user collaboration to address road safety concerns. ClearNav offers a user-friendly interface for reporting road hazards, allowing users to pinpoint locations, categorize imperfections, and provide additional details with text descriptions and optional media uploads. Data collected through user reports feeds into the system, with potential future enhancements utilizing user data for verification and advanced data analysis.

ClearNav represents a significant step forward in promoting collaborative road safety. By empowering users to report hazards and fostering a data-driven approach, ClearNav can benefit drivers, road maintenance authorities, and the broader community.

- **Drivers:** Users gain a powerful voice to report hazards, receive real-time alerts on approaching imperfections, and potentially access hazard-specific mitigation tips. Imagine navigating with the knowledge of potential dangers before they even appear, allowing for safer and more informed driving decisions.

- **Road Maintenance Authorities:** Anonymized data analysis becomes a powerful investigative tool. ClearNav can reveal hidden patterns, pinpointing accident hotspots and uncovering potential correlations between road features and specific hazards. Armed with this intelligence, authorities can prioritize repairs based on real-time risk assessments, allocate resources more effectively, and even implement preventive maintenance strategies in high-risk areas before incidents occur. This proactive approach can significantly reduce road hazards and contribute to a safer driving environment for everyone.

- **Community:** ClearNav fosters a spirit of collaboration, weaving a safety net for all road users. Shared data becomes a valuable resource, contributing to broader road safety initiatives and informing future infrastructure development strategies. Imagine roads designed and built with the insights gleaned from millions of user reports. This collective effort, fueled by ClearNav, can pave the way for a future, where entire road networks are engineered with safety at their core, ensuring a smoother, safer driving experience for generations to come.

Moving forward, conducting a comprehensive user survey is crucial. This survey should gather data on user demographics, past experiences with road hazards, and their preferences for reporting mechanisms. Analyzing user feedback will inform the design and development of ClearNav, ensuring it caters to the specific needs and preferences of the target audience. This, in turn, will enhance the application's usability

8. FUTURE ENHANCEMENTS

Imagine a world where becomes your partner in road safety. ClearNav's could evolve into sophisticated sentinels, meticulously analyzing user-reported data to identify and filter out inconsistencies. This ensures the data foundation of ClearNav remains pristine, allowing it to unleash its true potential: advanced hazard prediction and prevention. By meticulously dissecting user-submitted reports, ClearNav can transform into a powerful detective, uncovering hidden patterns and potential correlations between specific road features and the likelihood of certain hazards. Imagine the possibilities! Road authorities, armed with this granular data, could become proactive guardians of our streets. They could prioritize repairs based on real-time risk assessments, allocate resources more effectively, and even implement preventive maintenance strategies in high-risk areas before incidents even occur.

But ClearNav's vision extends far beyond reactive measures. The future holds the promise of seamless integration with popular navigation apps. Gone would be the days of switching between apps or fumbling with distracting notifications. Real-time hazard information would be woven directly into your driving interface, transforming it into a dynamic safety map. Potential dangers would appear like glowing warnings on your windshield, prompting immediate awareness and allowing for safer navigation.

ClearNav's impact wouldn't be limited to individual drivers. The app has the potential to become a treasure trove of anonymized data, a goldmine for research institutions and public safety organizations. Sharing this valuable information could fuel groundbreaking road safety initiatives and inform future infrastructure development strategies. Imagine roads designed and built with the insights gleaned from millions of user reports. This collaborative effort, fueled by ClearNav, could pave the way for a future where entire road networks are engineered with safety at their core, ensuring a smoother, safer driving experience for generations to come.

9. REFERENCES

[1] M.S.Kamalesh, Bharatiraja Chokkalingam, Jeevanantham Arumugam, Gomathy Sengottaiyan, Shanmugavadivel subramani, and Mansoor Ahmad Shah. An Intelligent Real Time Pothole Detection and Warning System for Automobile Applications Based on IoT Technology. Journal of Applied Science and Engineering. 2020.

[2] Ajit Gholap and Prof.M.Rokade. Potholes and speed bumps detection. International Journal of Current Engineering and Technology. 2021

[3] Asst. Prof. P.Sudhakar , R. Ramya Sri, V.Harika, S Navya Sudha, V.N.S.Sandeepthi, S.Uma maheswari. Pothole Detection System Using IoT. UGC care Group I Journal. Vol-10. 2020

[4] K. B. Naveen, J. Adithya Gowda, O.H Amrutha, Harshitha N. Kumar, R Kushwanth. IoT based Automatic Detection of Pot Holes and Speed Breaker Mapping. International Jounal of Research in Engineering, Science and Management. 2019.

[5] Alfandino Rasyid, Mochammad Rifki Ulil Albaab, Muhammad Fajrul Falah, Yohanes Yohanie Fridelin Panduman, Alviansyah Arman Yusuf, Dwi Kurnia Basuki, Anang Tjahjono, Rizqi Putri Nourma Budiarti, Sritrusta Sukaridhoto, Firman Yudianto, Hendro Wicaksono. Pothole Visual Detection using Machine Learning Method integrated with Internet of Thing Video streaming Platform. International Electronics Symposium (IES). 2019.

[6] Etukala Jaswanth Reddy, Padhuri Navaneeth Reddy, Govindula Maithreyi, M.Bharath Chandra Balaji, Santanu Kumar Dash, K. Aruna Kumari. Development and Analysis of Pothole detection and Alert based on NodeMCU. International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE). 2020.

[7] Van khang Nguyen and Eric Renault. Cooperative sensing and Analysis for a Smart Pothole Detection. 15th International Wireless Communications & Mobile Computing Conference (IWCMC). 2019.

[8] Gupreet Singh, Rajeev Kumar, Poonam kshatriya. Detection of Potholes and Speed Breaker on Road. First International conference on Secure cyber-Computing and Communication (ICSCCC). 2018.

[9] Riya R, Nakulraj K and Anusha A. Pothole Detection Methods. Proceedings of the International Conference on Inventive Computation Technologies (ICICT-2018). 2018.

[10] Rui Fan, Ozgunalp, Brett Hosking, Ming Liu, Ioannis Pitas. Pothole Detection Based on Disparity Transformation and Road Surface Modeling. IEEE Transaction on Image Processing (volume: 29). 2019.

[11] Byeong-ho kang and Su-il Choi. Pothole Detection System using 2D LiDAR AND Camera. 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN). 2017.

[12] Stepheena Joseph, Mr.K.Edison Prabhu. Role of Ultrasonic Sensor in Automatic Pothole and hump detection System. International Journal of scientific & Engineering Research volume 8. 2017.

[13] Amitha Terish, Shiji Johnson, Sinu Valsan, Sooraj P, Indrasena N V. Real Time Pothole Detection and Road Monitoring System. International Journal of Innovative Research in Science, Engineering and Technology. 2019.

[14] Kumaravel A, Tharani R, Thillaikarasi G, Varsha A. Automated Speed Breakaer to Control the Speed breaker to Control the speed of the vehicle based on IoT. International Research Journal of Engineering and Technology (IRJET). 2020

[15] Ramsha Sumail, Harleen Boparai, Faraz Ahmed. Automated Sensor based Pothole Detection System for Preventing Unfortunate Casuality. International Journal of Engineering Research & Technology (IJERT).

[16] Rahul Ramakrishnan, Ayusha Pendse, Chetna Sharma, Priya Chimurkar. Speed Breaker Detection and Mapping using IoT.Proceedings of the Third International Conference on Smart Systems and Inventive Technology (ICSSIT 2020).

[17] Rajeshwari Madli, Santosh Hebbar, Praveenraj Pattar, and Varaprasad Golla, “Automatic Detection and Notification of Potholes and Humps on Roads to Aid Drivers”, IEEE Sensors Journal, Vol. 15, No. 8, August 2015.

[18] R. Sundar, S. Hebbar, and V. Golla, “Implementing intelligent traffic control system for congestion control, ambulance clearance, and stolen vehicle detection,” IEEE Sensors J.,vol. 15, no. 2, pp. 1109–1113,Feb. 2015.

[19] I. Moazzam, K. Kamal, S. Mathavan, S. Usman, and M. Rahman,“Metrology and visualization of potholes using the micro-soft Kinectsensor,” in Proc. 16th Int. IEEE Conf. Intell. Transp. Syst., Oct. 2013, pp. 1284–1291.

[20] S. S. Rode, S. Vijay, P. Goyal, P. Kulkarni, and K. Arya, “Pothole detection and warning system: Infrastructure support and system design,” in Proc. Int.Conf. Electron. Comput. Technol., Feb. 2009, pp. 286–290.

[21] H. Youquan, W. Jian, Q. Hanxing, Z. Wei, and X.Jianfang, “A research of pavement potholes detection based on three-dimensional projection transformation,” in Proc. 4th Int. Congr. Image Signal Process. (CISP), Oct. 2011, pp. 1805–1808.

[22] Amel Ali Alhussan, Doaa Sami Khafaga, El-Sayed M. El-Kenawy, Abdelhameed Ibrahim Marwa Metwally Eid Abdelaziz A. Abdelhamid Date of Publication: 05 August 2022 INSPEC Accession Number: 21970489 DOI:10.1109/ACCESS.2022.3196660

[23] Xinke Li Yongcai Guo Yongming Li Date of Publication: 23 December 2019 Electronic ISSN: 2169-3536 INSPEC Accession Number: 19442450 INSPEC Accession Number: 19442450 DOI: 10.1109/ACCESS.2019.2961755

[24] Abhishek Kumar; Chakrapani; Dhruba Jyoti Kalita; Vibhav Prakash Singh 2nd International Conference on Data, Engineering and Applications (IDEA) DOI: 10.1109/ IDEA49133.2020 ,28-29 Feb. 2020

[25] Surekha Arjapure; D.R. Kalbande, International Conference on Communication information and Computing Technology (ICCICT) DOI: 10.1109/ICCICT50803.2021 25-27 June 2021

[26] Alessio Gagliardi; V. Staderini; S. Saponara Date of Publication: 14 June 2022 Electronic ISSN: 2169-3536 INSPEC Accession Number: 21799310 DOI: 10.1109/ACCESS.2022.3183116

[27] Dong Chen; Nengcheng Chen; Xiang Zhang; Yuhang Guan Date of Publication: 19 August 2022 INSPEC Accession Number: 22014136 DOI: 10 . 1109 / JSTARS.2022.3200147

[28] E. Salari; G. Bao Published in: 2011 IEEE INTERNATIONAL CONFERENCE ON ELECTRO/INFORMATION TECHNOLOGY Date of Conference: 15-17 May 2011 INSPEC Accession Number: 12178472 DOI: 10.1109/EIT.2011.5978575

[29] Pranjal A. Chitale; Kaustubh Y. Kekre; Hrishikesh R. Shenai; Ruhina Karani; Jay P. Gala International Conference on Image and Vision Computing New Zealand, IVCNZ DOI: 10.1109/IVCNZ51579.2020 25-27 Nov. 2020

[30] Vineet Kaushik; Birinderjit Singh Kalyan 2022 Second International Conference on Computer Science, Engineering and Applications (ICCSEA) 08-08 September 2022 Date Added to IEEE *Xplore*: 07 November 2022 ISBN Information: INSPEC Accession Number: 22239213 DOI: 10.1109/ICCSEA54677.2022.9936360 Publisher: IEEE