Visvesvaraya Technological University, Belagavi – 590018

PROJECT REPORT

on

DESIGN AND IMPLEMENTATION OF SECURED LOCATION TRACKING SYSTEM

Submitted in partial fulfillment for the award of degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by

PREKSHA	**4CB21CS075**
PRITHWIN B U	**4CB21CS076**
RAKSHAN SHETTY K	**4CB21CS084**
SUHANI	**4CB21CS113**

Under the Guidance of

Prof. Alok Ranjan

Assistant Professor, Department of Computer Science and Engineering

DEPT. OF COMPUTER SCIENCE AND ENGINEERING

CANARA ENGINEERING COLLEGE

(Affiliated to VTU Belagavi, Recognized by AICTE, Accredited by NBA)

Sudhindra Nagar, Benjanapadavu, Mangaluru - 574219,

Karnataka.

2024-25

CANARA ENGINEERING COLLEGE

(Affiliated to VTU Belagavi, Recognized by AICTE, Accredited by NBA)

Sudhindra Nagar, Benjanapadavu, Mangaluru - 574219, Karnataka.

DEPT. OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work entitled **"DESIGN AND IMPLEMENTATION OF SECURED LOCATION TRACKING SYSTEM"** carried out by

Preksha	**4CB21CS075**
Prithwin B U	**4CB21CS076**
Rakshan Shetty K	**4CB21CS084**
Suhani	**4CB21CS113**

the bonafide students of VII semester Computer Science and Engineering in partial fulfillment for the award of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belagavi during the year 2024-2025. It is certified that all corrections/suggestions indicated for Internal Assessment as indicated during internal assessment. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Prof. Alok Ranjan	**Dr. Karthik Pai B H**	**Dr. Nagesh H R**
Project Guide	HOD-CSE	Principal

External Viva:

Examiner's Name	Signature with Date
1.	.....
2.	.....

CANARA ENGINEERING COLLEGE

(Affiliated to VTU Belagavi, Recognized by AICTE, Accredited by NBA)

Sudhindra Nagar, Benjanapadavu, Mangaluru - 574219, Karnataka.

DEPT. OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We hereby declare that the entire work embodied in this Project Report titled **"DESIGN AND IMPLEMENTATION OF SECURED LOCATION TRACKING SYSTEM"** has been carried out by us at CANARA ENGINEERING COLLEGE, Mangaluru under the supervision of **Prof. Alok Ranjan,** for the award of **Bachelor of Engineering** in **Computer Science and Engineering**. This report has not been submitted to this or any other University for the award of any other degree.

Preksha	**4CB21CS075**
Prithwin B U	**4CB21CS076**
Rakshan Shetty K	**4CB21CS084**
Suhani	**4CB21CS113**

Acknowledgement

We dedicate this page to acknowledge and thank those responsible for the shaping of the project. Without their guidance and help, the experience while constructing the dissertation would not have been so smooth and efficient.

We sincerely thank our Project guide **Prof. Alok Ranjan**, Assistant Professor, Computer Science and Engineering for his guidance and valuable suggestions which helped us to complete this project. We also thank our Project coordinator **Prof. Janardhana Bhat K**, Dept of Computer Science and Engineering, for his consistent encouragement.

We owe a profound gratitude to **Dr. Karthik Pai B H**, Head of the Department, Computer Science and Engineering, whose kind support and guidance helped us to complete this project work successfully. We also take this opportunity to thank our Dean Academics and Vice-Principal **Dr. Demian Antony D'Mello** and we are extremely thankful to our Principal, **Dr. Nagesh H R**, for their support and encouragement.

We would like to thank all faculty and staff of the Department of Computer Science and Engineering who have always been with us extending their support, precious suggestions, guidance, and encouragement through the project. We also express our gratitude to our beloved friends and parents for their constant encouragement and support.

Preksha
Prithwin B U
Rakshan Shetty K
Suhani

Abstract

The Design and Implementation of NextGen Secured Location Tracking System project aims to develop an advanced location tracking system that provides accurate real-time monitoring while ensuring robust data security and user privacy. The aim is to create a versatile and scalable solution applicable in various domains, such as personal safety, asset management, and smart city initiatives.The project centers on the increasing demand for secure and reliable location tracking solutions in various fields, such as personal safety and smart city applications, where existing systems often suffer from security vulnerabilities and privacy concerns. Testing scenarios assess the system's accuracy, security, and performance before deployment. It utilizes modular design methodologies to build and integrate the system's components, including GPS-based tracking and real-time data communication between the mobile apps. The system is tested under different scenarios, including varying network conditions, mobile device movement, and simulated data interruptions, to evaluate tracking accuracy and system performance. Additionally, stress testing is conducted to assess the system's response under high user loads and real-time data updates.Its versatile architecture and robust performance highlight its applicability in modern location-based services across various sectors.

Keywords :Real Time Location tracking, Geofencing, Android app.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Location tracking technologies are proving to be increasingly necessary in different sectors: personal safety applications, logistics, and smart city applications. The best possible solutions in these technologies are being challenged by their key limitations, primarily issues of data security, user privacy, and accuracy. Besides these, the vulnerabilities associated with these traditional tracking systems make them prone to unauthorized access to sensitive information. In addition, the Internet of Things and massive development of mobile devices increase the demand for secure location tracking systems that offer trustworthy location tracking in heterogenous environments. Existent systems fail to provide real-time location updates while avoiding user privacy. The goal of this project is to bridge the functionality-security gap by the application of advancements in GPS, GSM, and encryption technologies. In developing this solution, the project would help to build a trustworthy and privacy-oriented tracking system in order to answer the current challenges facing the industry. In this way, the answer to the demands from modern users will help it to instill confidence and guarantee the safe use of location-based services that indeed are becoming increasingly critical to daily life. The approach should not only improve the overall effectiveness of tracking technologies but also set the standard for security and privacy in the implementation of future location tracking solutions.

1.2 Motivation and Problem Statement

Motivation: The rapidly increasing demand for secure and efficient location tracking solutions across various sectors, including personal safety, transportation, and smart city applications. As mobile technology and the Internet of Things continue to evolve, there is a pressing need for systems that provide accurate real-time location data while ensuring the security and privacy of users.

Problem Statement: Existing tracking systems often expose users to risks of data breaches and unauthorized access, which can undermine trust and hinder adoption. By addressing these challenges, this project aims to enhance user confidence in location-based services and improve their overall effectiveness. This project seeks to tackle these issues by creating a secure and reliable location tracking system that ensures accurate monitoring while prioritizing user privacy and data protection. By leveraging advancements in location tracking, the project aims to develop a comprehensive solution that meets contemporary demands for secure tracking system.

1.3 Objectives

- Develop a user-friendly interface that includes an intuitive mobile application and web dashboard for ease of access and control.
- Develop an email scheduling feature for automated, scheduled email notifications based on user-defined triggers, ensuring timely updates and user engagement.
- Develop a geofencing feature that allows users to set virtual boundaries and receive alerts for entry and exit, providing enhanced location control and monitoring.

1.4 Scope and Limitations

Scope: The project will develop a comprehensive, secure real-time location tracking system with GPS and GSM technology and features such as geofencing and email scheduling. It will support easy access via a mobile app and web dashboard, thereby ensuring the security of the data, scalability, and effective location monitoring.

Limitations: The system may suffer from internet connectivity issues, GPS accuracy problems in certain environments, and potentially shorter battery life. Additionally, user adoption could be affected if the system is not user-friendly.

1.5 Organization of the Report

The report begins with a Title Page and an Abstract summarizing the learning objectives and significance. The remaining part of the report is organized as following chapters:

Chapter-1: It includes sections such as the Introduction, which outlines the scope and motivation, followed by sections on Objectives, Scope, and Limitations.

Chapter-2: This Literature Survey includes a review of various articles and the design, methodology, and techniques adopted from each. It also presents the results achieved in those articles.

Chapter-3: This chapter covers the Software Requirements Specification, which includes detailed descriptions of functional and non-functional requirements, hardware and software requirements, performance requirements, and other specific requirements. The chapter concludes with a summary.

Chapter-4: These were the system design specification which includes Architectural Diagram, Proposed System and it includes Modular Design Diagram, Sequence Diagram, Use Case Diagram, Activity Diagram for

use case, Zero-Level Data Flow Diagram, First-Level Data Flow Diagram, Second-Level Data Flow Diagram and this chapter ends with the summary.

Chapter-5: This chapter includes Software used with justification which has Front-end Development and Back-end development and Framework used and coding languages used for development and the Operating system which has been used and Hardware used with justification and the algorithms which were made use of and coding and the chapter ends with summary.

Chapter-6: This chapter consists of Results and Discussions and testing which were done and also the types software testing and the unit testing performed. And testing criteria which were supposed to be followed. And the chapter ends with the summary.

Chapter-7: This chapter consists of the project conclusion and the future work which can be done on the project.

The report concludes with references, drill-bit plagiarism report and the project expo details.

Chapter 2

Literature Survey

The mobile tracking systems highlight impressive developments with regard to GPS and GSM technologies, improving on-time location accuracy for as diverse an application as personal safety or logistics. Present-day research fosters the growing importance attached to data security: many systems now employ encryption and other secure communication protocols that obviate fears about privacy. The user interfaces play a crucial role in enhancing adoption rates and ultimate user satisfaction-a critical aspect toward making these systems more accessible. Continued expansion into mobile and web functionalities fuels increased applicability of tracking solutions in several sectors. In general, the broader work here asserts that tracking technology as applied by mobiles continues evolving to meet the user demands and security challenges.

2.1 Article 1: Radar: An In-Building RF-Based User Location and Tracking System

The Radar system tracks users indoors using the strength of the RF signal from multiple base stations, including modeling signal propagation and real-time location inference. Its capability is to perform such tasks with high accuracy at a median error of less than 2.5 meters. It enhances indoor navigation and location-based services in complex environments[1].

2.1.1 Brief Findings of Article 1

The Radar system relies on the strength of RF signals from multiple base stations to compute user locations within indoor environments. This work will demonstrate how a hybrid offline modeling of propagation, combined with real-time location inference, can power Radar to accurately track users, even in the most complex layouts and configurations. The results demonstrate that the system greatly enhances location-aware applications by providing an absolute solution to indoor navigation and tracking.

2.1.2 Design and Methodology Adopted in Article-1

• Base stations were set up in a building to collect Signal Strength and Signal-to-Noise Ratio data, providing a realistic environment for signal propagation analysis.

• A k-nearest neighbors algorithm, using multiple signal measurements, improved location accuracy by refining real-time positioning based on multiple base station inputs.

2.1.3 Results Achieved in Article 1

Taking a wide array of indoor scenarios as its test cases, the system displayed a fairly high level of accuracy with median error not more than 2.5 meters in tracking user location.The project indicate the system to successfully manage the complexities of indoors environments with all the hurdles and spotty signal conditions involved. Results do hint at the possibility that Radar could indeed emerge as a new solution for enhancing location-based services in real-time applications.

2.2 Article 2: Live Location Tracker

The Live Location Tracker project uses GPS for real-time positioning, created using Android Studio. It utilizes the Map SDK for map-based visualization and Firebase's Real-Time Database for seamless data storage

and retrieval. This makes it possible to enhance user experience with timely updates on location as well as proper handling of data, making the system applicable in environments that focus on logistics, personal safety, and even emergency response[2].

2.2.1 Brief Findings of Article 2

This uses GPS for location positioning and was designed in Android Studio, and made use of Map SDK to easily display real-time location data. Moreover, it utilized Firebase Database and Real-Time Database to efficiently store and retrieve the location. The improved user experience was realized through real-time updates, better accessibility, and secure handling of data, manifesting its practical application in various sectors such as logistics and personal safety applications.

2.2.2 Design and Methodology Adopted in Article 2

- The system combines GPS and GSM technologies for real-time tracking, accessible via a mobile app and web interface. It uses a client-server architecture with encryption for secure data transmission.
- The system follows best practices in mobile tracking, focusing on usability and privacy, based on insights from studies that enhance user experience and address privacy issues.

2.2.3 Results Achieved in Article 2

The system could boast of getting actual, real-time location tracking by integrating GPS and GSM, having a large boost in user safety and logistical efficiency. The users elaborated on the following improvement in ease of use of web and mobile applications while data safety was further enhanced by encryption protocols that met best practice standards for offering a safe mode in which to maintain user anonymity with mobile tracking systems.

2.3 Article 3: Mobile Tracking System using Web Application and Android Apps

A GPS and GSM-based real-time mobile location tracking system wherein web-based applications integrate Android apps has proved to be efficient in monitoring real-time locations. Such a dual-platform approach increases user involvement due to intuitive interfaces and supports strong security offered through encryption protocols. Such systems are, therefore, valuable in personal safety, logistics, and in emergency response, bringing forth reliable tracking with effective obviation of privacy concerns[3].

2.3.1 Brief Findings of Article 3

A mobile tracking system based on web applications and Android apps will be bringing bigger advantages in real-time monitoring of locations for personal safety and overall logistical efficiency. The general idea is to have the ability to access such information through both mobile and web interfaces, by using GPS and GSM technologies. The project have integrating mobile and web functionalities increases the potential applications of the tracking solutions in fields like transportation and emergency responses.

2.3.2 Design and Methodology Adopted in Article 3

- The system uses a client-server architecture to integrate GPS and GSM technologies for real-time location tracking. It features a user-friendly Android app for location data management and a complementary web application with a dashboard for tracking and analysis.
- The system's design ensures efficient location tracking and data management, leveraging both mobile and web platforms for user interaction and analysis.

2.3.3 Results Achieved in Article 3

The web application and Android app-based mobile tracking system has greater accuracy in the retrieving real-time location, better design for the user interface, and strong data security. The project experienced higher satisfaction due to ubiquitous access from the devices and, therefore, increased chances of successful applications in terms of safety and logistics. Such developments have proven that this system is effective and easy to use.

2.4 Article 4: Bus Location Tracking System

This project is aimed at improving public transportation. It is possible for GPS and RFID to be used to track buses in real time, accompanied by a user-friendly Android app that can easily be accessed. This therefore reduces waiting times for commuters and enhances the efficiency of operations. Testing confirmed the reliability of the system, and further improvements through the app are expected to enhance the user experience further[4].

2.4.1 Brief Findings of Article 4

This is a Bus Location Tracking System that enhances the efficiency in the provision of public transport service through tracking buses on real time; it aims to eliminate some irritation when consumers are using traditional web-based applications through provision of an android application for easier access. The system mainly depends on GPS technology and RFID in a bid to drastically cut down on passengers waiting times. The findings indicated that through this system, user experience in bus management can be optimized along with operational efficiency.

2.4.2 Design and Methodology Adopted in Article 4

- Buses are tracked using GPS, and the data is recorded on a server. Users can access real-time information via a web application, with

an Android app utilizing features like Web View, GPS tracking, and push notifications for easy access.

- The Unit testing of front-end components was done with Jest and Enzyme, back-end integration testing used Supertest, and user acceptance testing ensured functionality and usability.

2.4.3 Results Achieved in Article 4

Buses locations and schedules were offered in real time. The computer convenience and efficiency would be enhanced. The presented application's stability and dependability were given during the testing period. With the proposed Android app version, user experience will be further maximized through smartphone potential, which reduces the waiting time for bus riders.

2.5 Article 5: Location Based Services using Android Mobile Operating System

The location-based services are offered through the use of GPS and mapping technologies to enrich the experience of the users. Such services make possible various applications, including ride-sharing, food delivery, and location-specific advertising. The availability of location data provides businesses with the ability to align the offerings in light of customer needs, thus enhancing satisfaction and effectiveness in operations. In location-based services are part of innovation and convenience in modern applications involving mobile devices[5].

2.5.1 Brief Findings of Article 5

The LBS findings from the Android mobile OS indicate a highly advanced engagement of users and functionality. The implementation of GPS technology assists in applications that track location in real time, navigate, and provide personal services. This research highlights the significance of

LBS for modern user experiences in transport and tourism industries. In addition to this, location data becomes easily accessible with easy access to platforms like Google Maps, allowing for innovation in developing and delivering services via mobile applications.

2.5.2 Design and Methodology Adopted in Article 5

- The system integrates GPS, network-based location techniques, and Google Maps API for real-time location tracking. It uses Android applications with user-friendly interfaces, incorporating geofencing and location caching for improved performance.
- Background services and location permissions are utilized to ensure efficient tracking and responsiveness, enhancing functionality and user experience.

2.5.3 Results Achieved in Article 5

The application using the Android mobile operating system, location-based services have experienced notable boosts regarding user engagement and service delivery. With GPS and Google Maps, there is increased accuracy in the tracking and navigation to utilize real-time data while at the same time increasing users experience across applications. The setup has been shown to increase efficiency in several operations and satisfy customers in fields ranging from logistics to personal safety.

2.6 Article 6: Survey on Live Tracking Phone No. Via Android/IOS Based App

The study on live tracking of phone numbers through Android/iOS apps strongly insists on the combination of GPS and LBS for real-time monitoring. Results in higher user safety, better navigation, and better communication efficiency. Important applications are emergency services and personal safety, proving more importance of its role in daily life[6].

2.6.1 Brief Findings of Article 6

The research suggests the tracking of live numbers by way of both Android and iOS applications, indicating massive progress made in the location-based services, mainly involving GPS and Google Maps. It shows the project improved on keeping people safe and real-time tracking capabilities, thus improving navigation and communicating.

2.6.2 Design and Methodology Adopted in Article 6

- The system uses GPS for location tracking and Google Maps for real-time visualization. It is developed using the native frameworks of both Android and iOS to ensure cross-platform compatibility.
- Data security is ensured through encryption, and the user interface is optimized for a seamless experience across various devices.

2.6.3 Results Achieved in Article 6

The live tracking of phone numbers through Android-based or iOS-based apps provided major outcomes in terms of enhancing real-time monitoring and user safety. The system made use of GPS, LBS, and Google Maps to acquire precise location tracking together with smooth switching between platforms. The results presented that reliability in navigation and emergency response applications improved when accessed by users within a personal environment and commercial environment.

2.7 Article 7: Farm Animal Location Tracking System Using Arduino and GPS Module

This system will use GPS and IoT technology to track farm animals in real-time. The accuracy of the updated information obtained from difficult environments helps avoid livestock theft while increasing the security of the animals. It is cost-effective, reliable, accessible through cloud computing, and therefore an efficient solution for farming management[7].

2.7.1 Brief Findings of Article 7

The article explains a real-time animal traceability system based on GPS and IoT. It states how precise identification is significant in the avoidance of stealing and safety of animals. It should be low-cost and reliable. Arduino and a GPS module are used to send location data to the cloud. This can monitor farm animals effectively even in challenging environments with better improvements in animal welfare and farm management.

2.7.2 Design and Methodology Adopted in Article 7

- The system consists of a collar, GPS module, Arduino, and Wi-Fi module designed to be lightweight for animals. The GPS data is sent to the cloud, and a web application visualizes the real-time location.
- Cloud computing enables easy access to location data, enhancing tracking capabilities by allowing remote monitoring and real-time visualization of the animal's location.

2.7.3 Results Achieved in Article 7

The proposed tracking system demonstrated in real-time the location of the farm animals and even displayed accurate geographical data under different adverse weather conditions. Monitoring was continuous with GPS and IoT technologies, which was highly beneficial in preventing theft cases and ensuring safety in cattle. Low-cost design and reliable performance make the system practical to use for farmers.

2.8 Article 8: Real Time Vehicle Tracking Scheme By Analysing Spatial Position Through GPS And GPS

These technologies are now applied to the real-time vehicle tracking scheme to analyze the spatial position, which implies effective vehicle monitoring

and location-based services. Such a system permits users to trace moving vehicles in real-time and implement geo-fencing for enhanced safety. Combining these technologies, the scheme elevates fleet management and emergency response in greatly accessible tracking data through mobile and web applications[8].

2.8.1 Brief Findings of Article 8

These technologies are now applied to the real-time vehicle tracking scheme to analyze the spatial position, which implies effective vehicle monitoring and location-based services. Such a system permits users to trace moving vehicles in real-time and implement geo-fencing for enhanced safety. Combining these technologies, the scheme elevates fleet management and emergency response in greatly accessible tracking data through mobile and web applications.

2.8.2 Design and Methodology Adopted in Article 8

- The system uses GPS for location acquisition, GSM for data transmission, and a microcontroller for signal processing. It includes geo-fencing and data analytics techniques for enhanced monitoring and user alerts. The system is accessible via mobile and web applications for efficient fleet management and real-time tracking.
- The system ensures data security while utilizing GPS and GSM technologies for accurate tracking. Geo-fencing and analytics techniques improve monitoring capabilities and alert users for better fleet management.

2.8.3 Results Achieved in Article 8

The real-time vehicle tracking system has improved accuracy and responsiveness to observe the vehicle. As the system has integrated GPS and GSM technologies, it furnished real-time location updates. The scheme facilitates fleet management and also ensures user safety to higher extents.

Geo-fencing helped the system warn and track a vehicle in advance so that greater operational efficiency and reduced time is maintained during an emergency response.

2.9 Article 9: The Android App For Women's Security With Sms Alert

The live tracking of phone numbers through Android-based or iOS-based apps provided major outcomes in terms of enhancing real-time monitoring and user safety. The system made use of GPS, LBS, and Google Maps to acquire precise location tracking together with smooth switching between platforms. The results presented that reliability in navigation and emergency response applications improved when accessed by users within a personal environment and commercial environment[9].

2.9.1 Brief Findings of Article 9

The findings reveal that location-based services (LBS) on Android effectively deliver real-time, location-sensitive information, enhancing user convenience and decision-making. Applications range from traffic monitoring to security and surveillance, enabling personalized data access on mobile devices. The results show a shift from desktop-restricted applications to mobile solutions, providing relevant information.

2.9.2 Design and Methodology Adopted in Article 9

- The system uses GPS and Google Maps to capture current location data, with location-based contexts to personalize information. The user interface is designed for easy navigation, and data accuracy is ensured through Location Services (LCS).
- The system is user-centric, designed to provide effective access to personalized location data for applications like traffic monitoring and

security. The focus is on ensuring accurate and user-friendly experiences.

2.9.3 Results Achieved in Article 9

The adoption of LBS through Android mobile operating systems improved the delivery of real-time information and also the convenience of the user. Utilization of GPS and Google Maps ensured accurate location tracking as well as access to personalized data. The results indicated that users were highly engaged with applications like security, traffic monitoring, and travel assistance with uninterrupted access of information to the mobile devices.

2.10 Article 10: EyeKids: Real-Time Tracking and Monitoring System for Child Safety

The EyeKids system provides real-time child tracking using a GPS and GPRS-equipped device that sends location updates every 30 seconds. Paired with a parent application, it enhances child safety by enabling location monitoring and distress alerts. A NoSQL database efficiently handles dynamic data, while the Rational Unified Process ensures iterative development and refinement[10].

2.10.1 Brief Findings of Article 10

The EyeKids system was very effective in integrating the tracking unit with a parent application to monitor real-time locations. Utilizing the power of GPS and GPRS, the device captures data on a child's location every 30 seconds and starts transmitting this information. The system is aimed at increasing child safety as parents can track their children's movements and distress alerts. The project demonstrates the usage of NoSQL database for dynamic data handling, which shows the efficient way of storing and retrieving tracking information.

2.10.2 Design and Methodology Adopted in Article 10

- The system uses a NoSQL database to manage dynamic location data, storing details like latitude, longitude, and timestamps in separate directories for parents and children. Hardware includes an Arduino with GPS and GPRS for cloud data transmission.
- The system follows the Rational Unified Process, utilizing iterative refinement through all project phases for continuous improvement and development.

2.10.3 Results Achieved in Article 10

The EyeKids tracking and monitoring system was successfully developed, meeting its objectives and project scope. The child tracking device effectively captures and transmits location data every 30 seconds, while the distress alert feature notifies parents when triggered. The integration of all system components was successful, demonstrating the system's functionality and reliability in real-world applications.

2.11 Comparision Table

Table 2.1: Comparison of Existing Work and Gap Identification

Project Title and Author	Problem Addressed	Implementation and Results	Limitations
Mobile Tracking System using Web Application and Android by Mohammad Jahangir Alam, 2017	The paper addresses issues with real-time location tracking accuracy, data synchronization, accessibility across devices, and security for location data.	This involves a mobile tracking system using GPS and web technologies, providing real-time location updates through an integrated web application and Android platform.	Limitations:Include The paper's limitations include network issues.

Live Location Tracker Article by Sarika Chaudhary, 2019	The paper addresses some issues about the accuracy of location tracking in real-time, privacy of information, and seamless availability on mobile.	It has incorporated GPS tracking with a real-time update mechanism so that the location is tracked accurately, and the interface is user-friendly.	Limitations:Include Data accuracy and connectivity.
Real Time Vehicle Tracking Scheme By Analysing Spatial Position Through GPS And GSM by T.Sindhu and L SaiRamesh, 2020	The need for an efficient vehicle tracking system that ensures safety and security in real-time.	The system utilizes GPS and GSM technology, implemented through an Android application. It allows users to track vehicles continuously, share locations, and receive alerts. Results showed high user satisfaction and effective tracking capabilities.	Limitations:Include GPS dependency and privacy concerns.
Location Based Services using Android Mobile Operating System by Amit Kushwaha and Vineet Kushwaha, 2021	The article explains how GPS, using satellites and ground stations, provides precise location data for fields like emergency response..	The article explains GPS technology's use of satellites for precise location tracking, benefiting fields like emergency response, wildlife monitoring, and mobile commerce.	Limitations:Include accuracy in obstructed areas.
Survey on Live Tracking Phone No. Via Android/IOS Based App by Aman Mishra and Advin Manhar, 2020	The need for effective and accurate tracking of phone numbers using mobile applications.	The survey discusses various existing technologies used for live tracking. It evaluates their methodologies and highlights the disparities in accuracy and effectiveness.	Limitations:Include accuracy, range, and storage issues .

The Android App For Women's Security With Sms Alert by Dr. K Srinivas and Dr. Suwarna, 2021	The app addresses the rising incidents of violence and harassment against women, providing a mechanism for immediate assistance in distress situations.	The application was developed on the Android Studio, utilizing GPS technology to send real-time location alerts to registered contacts.	Limitations:Include The app could expand by supporting iOS , integrating with law enforcement, and increasing user awareness.
Real-Time Bus Location Tracking System by Ishwar Bagad and Samruddhi Pimpale, 2024	Inconsistent public transportation schedules and lack of real-time tracking for buses.	The system utilizes GPS technology and an Android app for real-time tracking, resulting in improved commuter efficiency and positive user feedback.	Limitations:Include reliance on GPS accuracy and potential server downtime.
Farm Animal Location Tracking System Using Arduino and GPS Module by G. Ramesh , 2021	The project provides a reliable real-time tracking solution to enhance farm animal safety and prevent loss or injury in large areas.	The system uses Arduino, GPS, and Wi-Fi modules for real-time animal tracking, enabling farmers to monitor livestock with accurate data sent to the cloud, improving farm management.	Limitations:Include GPS signal loss in dense areas and battery dependency.
Radar: An In-Building RF-Based User Location and Tracking System by Paramvir Bahl and Venkata N Padmanabhan, 2020	The challenge of accurately locating and tracking mobile users in indoor environments, where traditional GPS fails.	Implemented an RF-based system using signal strength data from multiple base stations. Achieved a median location accuracy of 2 to 3 meters, demonstrating the potential for various location-aware services.	Limitations:Include interference from obstacles and extensive calibration needs.

EyeKids: Real-Time Tracking and Monitoring System for Child Safety by siti Zahidah Zaharan and Raja Zahilah, 2022	The project addresses the growing concern of child safety and the need for parents to monitor their children's movements in real-time to prevent potential dangers.	The EyeKids system uses GPS and GPRS in a child tracking device, sending real-time location data to a cloud database, with a parent app for live tracking and monthly reports, validated through user testing.	Limitations: Include system currently supports only single-child tracking per account, future upgrades could allow multi-child tracking.

2.12 Gaps Identified and Addressed

Gaps Identified: The study of existing literature survey on loaction based tracking system reveals following gaps. systems reveals following gaps

- Current tracking systems have a precision problem in signal-dense environments. Their approach above improves the processing of GPS data for reliability.
- Most systems support single-child tracking, which really limits the usability; this project includes multi-child tracking under one parent account.
- Continuous tracking consumes much battery life, and in this respect, this project enhances the battery efficiency and contains the real-time status display of the battery by the parents.
- Standard systems often lack real-time alerts, or even geofencing; this project is going to integrate geofencing notifications. It will be safer for you since you get alerts every time a child leaves the safe zone.

Gaps Addressed: Proposed location tracking system learn comprehensively addresses the identified gaps in the literature as follows.

- Improved accuracy in signal-dense environments to enhance location tracking reliability.
- Multi-child tracking feature under a single parent account to increase usability for families.

- Enhanced data security with encrypted storage and restricted access for privacy protection.
- Battery efficiency optimizations and real-time battery status updates to prevent unexpected shutdowns.
- Geofencing alerts to notify parents if a child exits designated safe areas, enhancing safety measures.

2.13 Summary

The literature survey explores a range of real-time tracking systems, covering applications in child safety, farm animal monitoring, and indoor navigation. Systems like EyeKids emphasize the critical need for accurate and reliable child tracking, utilizing GPS and GPRS to offer real-time monitoring and emergency alerts. However, limitations such as restricted compatibility, single-user tracking, and reliance on signal strength in obstructed environments are common. Farm animal monitoring systems use Arduino-based collars with GPS and Wi-Fi modules, which allow farmers to track livestock efficiently, although challenges include battery life and potential GPS signal interference in dense areas. Indoor systems, such as RADAR, employ RF signal strength and k-nearest neighbors for location inference, addressing indoor tracking but facing calibration demands. Future research suggests expanding functionality to include multi-user tracking, integrating geo-fencing, improving power efficiency, and adding health monitoring sensors, paving the way for more versatile and user-friendly tracking applications.

Chapter 3

Software Requirements Specification

3.1 Functional Requirements

The functional requirements of this project include GPS and GSM technology to accurately track the location of a user or object in real time. It should enable a mobile application and a web interface that can fetch location data of a user, with geofencing that creates area-based alerts where alerts are sent when an entity leaves or enters a defined zone. More than that, it includes notifications by email or SMS when the tracked entity enters or exits defined zones. In addition, the system should offer secure data transfer with data encryption techniques and ensure information about a user. The system should be scalable to accommodate multiple users and provide efficient performance in different conditions.

a) Real-Time Location Tracking

The system must continuously track the location of a user or object in real-time using GPS and GSM technologies. This will allow for precise and up-to-date tracking at all times.

b) Data Security and Privacy

Ensuring the security of transmitted and stored data is crucial. The system must implement encryption protocols to protect location data from unauthorized access and maintain user privacy.

c) Geofencing

The system should include the ability to set up virtual boundaries, or geofences, around specific areas. Users will receive notifications whenever the tracked object enters or exits these predefined zones.

3.2 Non-Functional Requirements

The system needs to be reliable, therefore, provide continuous operation with minimal downtime. It has to be scalable enough to be able to keep up with growth into the future and still handle large volumes of data. Performance must optimize response times while keeping the interface user-friendly. Data security is an imperative requirement that contains an encryption aspect and mechanisms for controlling secure access.

3.2.1 Safety Requirements

The system should have to ensure data privacy through strong encryption along all communication channels. Mechanisms for access control should be ensured to avoid unauthorized access. Additionally, physical safety of the hardware components should ensure that they are not damaged or malfunction during the operation and thus promote safe usage by users.

3.2.2 Performance Requirements

The system should have a minimum latency so as to give real-time location tracking, providing location updates in time, within seconds. It has the capability of supporting several users or devices at a time without its performance being degraded in the process of tracking. It is reliable and supports varying environmental conditions; it further ensures that GPS data accuracy prevails both in urban and rural settings.

3.3 User Interface Design

The user interface should be intuitive and easy to navigate, featuring a mobile app and web dashboard for seamless access to real-time location data. The map view should be interactive, displaying the tracked entity's current location and geofencing alerts. It should include simple controls for setting alerts and monitoring locations.

3.4 Hardware and Software Requirements

Hardware Requirements: The requirements for your phone tracking project include smartphones (iOS/Android) with GPS and internet connectivity for real-time tracking. Cloud servers are needed for data storage and processing. Additionally, the system requires reliable battery performance, optional Bluetooth for proximity detection, and secure authentication devices for user privacy.

Software Requirements: The requirements for the phone tracking application include a mobile app (iOS/Android) for real-time location tracking, a web dashboard for data visualization, and cloud-based storage for data processing. It also requires secure authentication, geofencing, and alert functionalities, with data encryption for privacy and scalability, using cloud platforms like AWS or Google Cloud.

3.5 Performance Requirements

The system should ensure real-time location updates with minimal latency, delivering accurate data within seconds. It must support multiple simultaneous users without performance degradation, maintaining high reliability. The system should function effectively in diverse environments, ensuring stable data connectivity and data accuracy across urban, rural, and challenging locations.

3.6 Summary

The mobile location tracker requires real-time tracking with secure data storage and transmission. It should include a user-friendly mobile app and web dashboard for easy access, supporting geofencing and alert features. The system must be scalable, battery-efficient, and compatible with both iOS and Android. It should offer secure authentication, offline functionality, and utilize cloud infrastructure for reliable data processing, ensuring optimal performance and user privacy.

Chapter 4

System Design

4.1 Abstract Design

4.1.1 Architectural Diagram

The system consisting of three main components: the parent device, child device, and the backend System. The parent device runs the parent app, allowing parents to sign in, request data from the child's device, and receive the fetched information from the backend. The child device runs the child app, where the child logs in, and the app collects relevant data and sends it to the backend system. The backend system comprises a database for storing and retrieving data, a data sync service for managing the flow of information between devices, and an authentication service for validating sign-in requests. The flow begins with the child app sending data to the backend, where it is stored and synchronized. When the parent requests information, the backend fetches the relevant data and sends it back to the parent app, ensuring secure and real-time communication between all components.

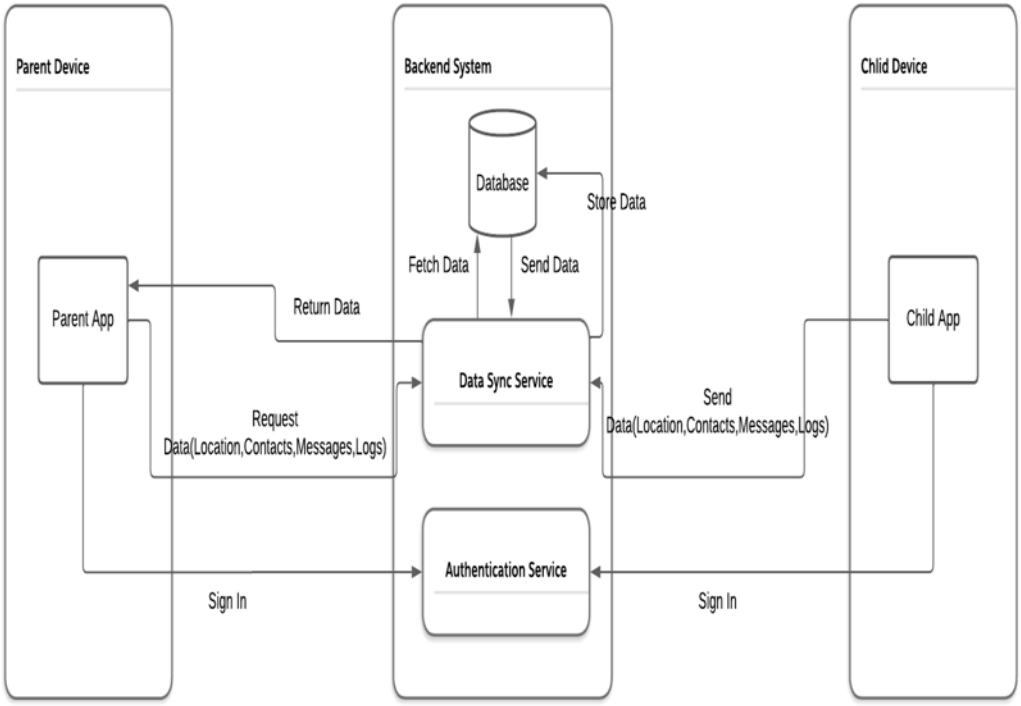


Figure 4.1: Architecture Diagram

Figure 4.1 represents the collection of datas from modules and sends it to the Backend System, where it is stored in a database. The parent app requests this data through the backend's Data Sync Service and retrieves it. An authentication service manages sign-ins for both apps, ensuring secure communication and real-time data synchronization.

4.2 Proposed System

The System describes the system initialization and monitoring process. It begins with system initialization, where the system checks if it is ready. If the system is ready, operations commence, and the system is continuously monitored. If any issues are detected during monitoring, they are logged, and alerts are sent for necessary actions, after which the system continues to monitor. If no issues arise, the system proceeds to stop operations and ends successfully. If the system is not ready during initialization, an error is logged, and the process retries until the system becomes ready. This process ensures smooth initialization, proactive issue management.

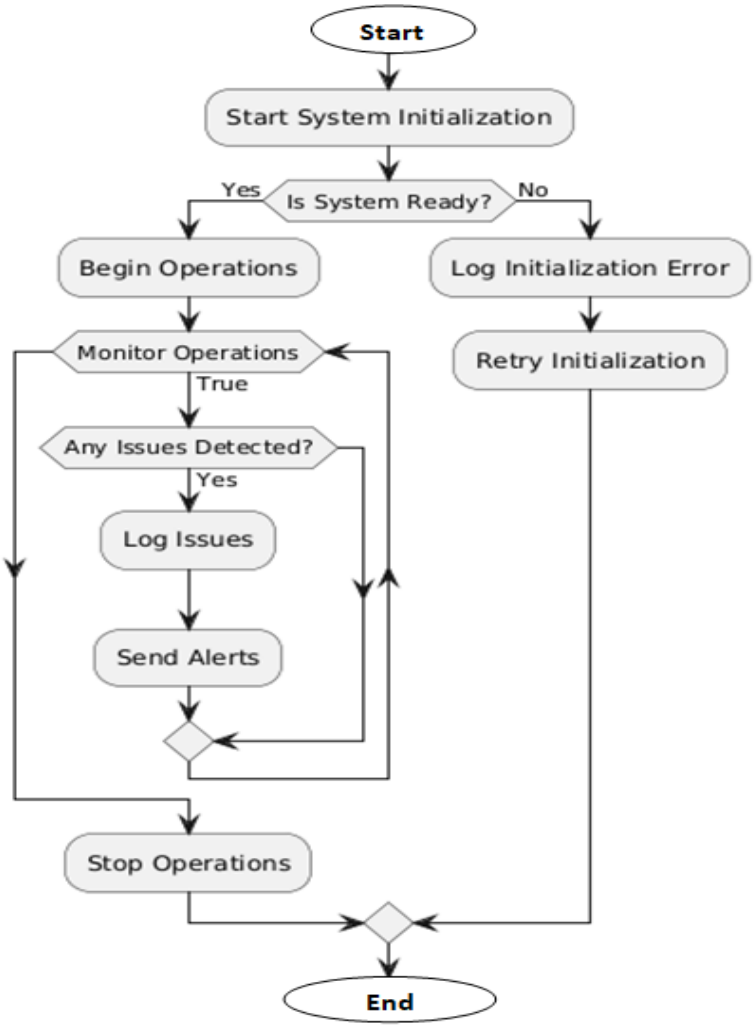


Figure 4.2: Proposed System

Figure 4.2 represents the a system initialization and monitoring process. If the system is ready, operations start and are monitored for issues. detected issues are logged, and alerts are sent. If no issues arise, operations stop successfully. If the system isn't ready, errors are logged, and initialization retries. This ensures smooth operation and issue management.

4.3 Functional Design

4.3.1 Modular Design Diagram

The modular diagram illustrates the architecture of a child location tracking application, consisting of three key modules: the parent module, child module, and backend module. The parent module allows parents to sign in, manage their child's profile, and view data such as location, contacts, messages, and call logs. The child module operates on the child's device and sends location, contacts, messages, and call log data to the backend. The backend module handles core functionalities, including authentication services to validate users, data synchronization to manage the flow of information, and database management for storing and fetching data. The parent app retrieves data from the backend after authentication, while the child app sends data, ensuring secure and seamless communication across the system.

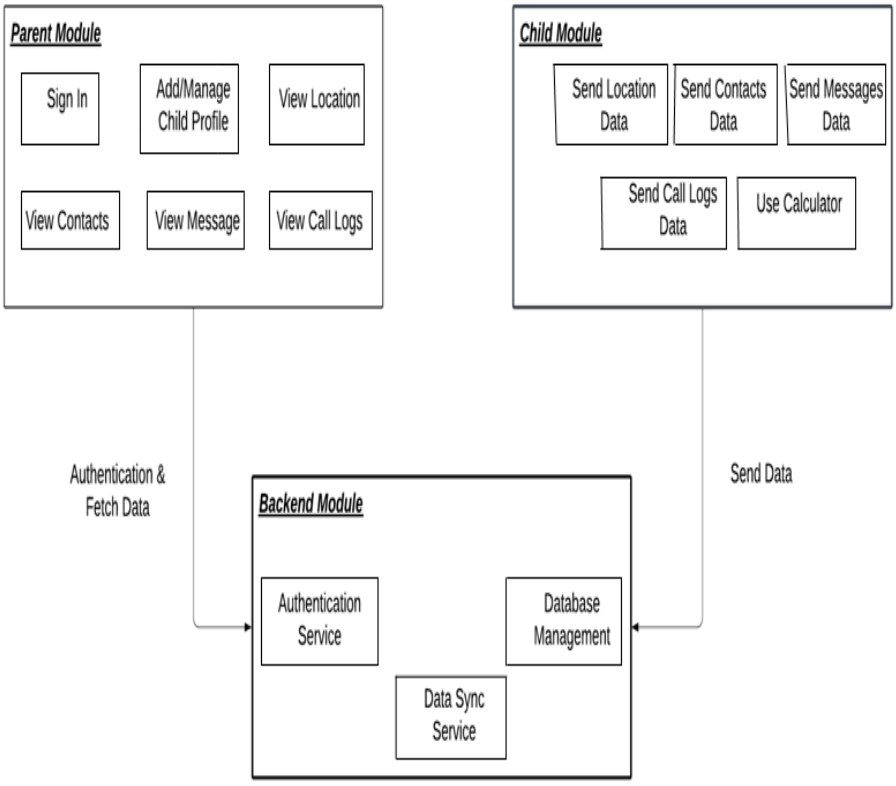


Figure 4.3: Modular Design Diagram

Figure 4.3 represents the a child location tracking initializes with three modules: the parent module for managing and viewing the child's data, the child module for sending the child's data and the backend module for handling authentication, data sync, and database management.

4.3.2 Sequence Diagram

The system interaction between a Parent, Child, Parent App, Child App, Backend System, and Database, focusing on authentication, data storage, and retrieval processes. It begins with the Parent signing into the Parent App. The credentials are passed to the Backend System, where they are validated against the Database. If successful, a login response is sent to the Parent App. The parent can then add a child profile, which is transmitted to the Backend System for storage in the Database, followed by a confirmation. Similarly, the Child signs into the Child App, and once authenticated, the app transmits data such as location, contacts, messages, and call logs to the Backend System, where it is securely stored in the Database.

For monitoring purposes, the parent app allows parents to request specific data, such as the child's location or other stored activities. The parent app sends the request to the backend system, which retrieves the necessary information from the database. The retrieved data, like location or logs, is then sent back to the Parent App, where it is displayed to the parent. This process ensures smooth and secure communication between the different components, efficient data storage in the database, and quick retrieval of data when needed. Overall, the diagram demonstrates a seamless system for secure authentication, effective data management, and reliable monitoring to help parents keep track of their child's activities through connected applications.

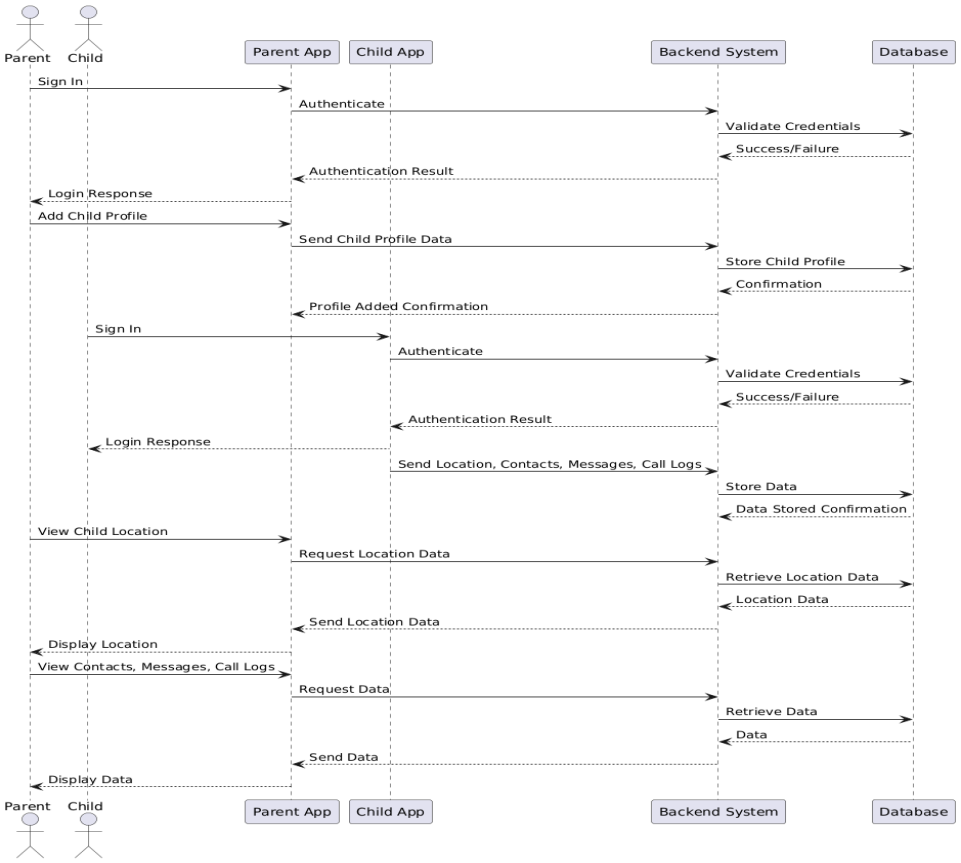


Figure 4.4: Sequence Diagram

Figure 4.4 represent the parent app and child app interacting with the backend system and database for authentication, data storage, and retrieval, enabling secure monitoring of a child's activities like location and logs.

4.3.3 Use Case Diagram

This diagram represents the use case model for a parent-centric application, which depicts the different functionalities that can be accessed by a parent user. The major functionalities include account management through sign up/sign in which includes authentication, and the ability to manage child details allowing the parent to add or update child information. Parents can also view child activities by seeing call logs, messages, and where the child is through the view child activities use case. geofencing capabilities are also provided, which enable parents to create or edit geofences and receive notifications if breaches of geofence happen. This

comprehensive use case model outlines the interaction flow between the parent user and the system to ensure effective monitoring and enhanced child safety.

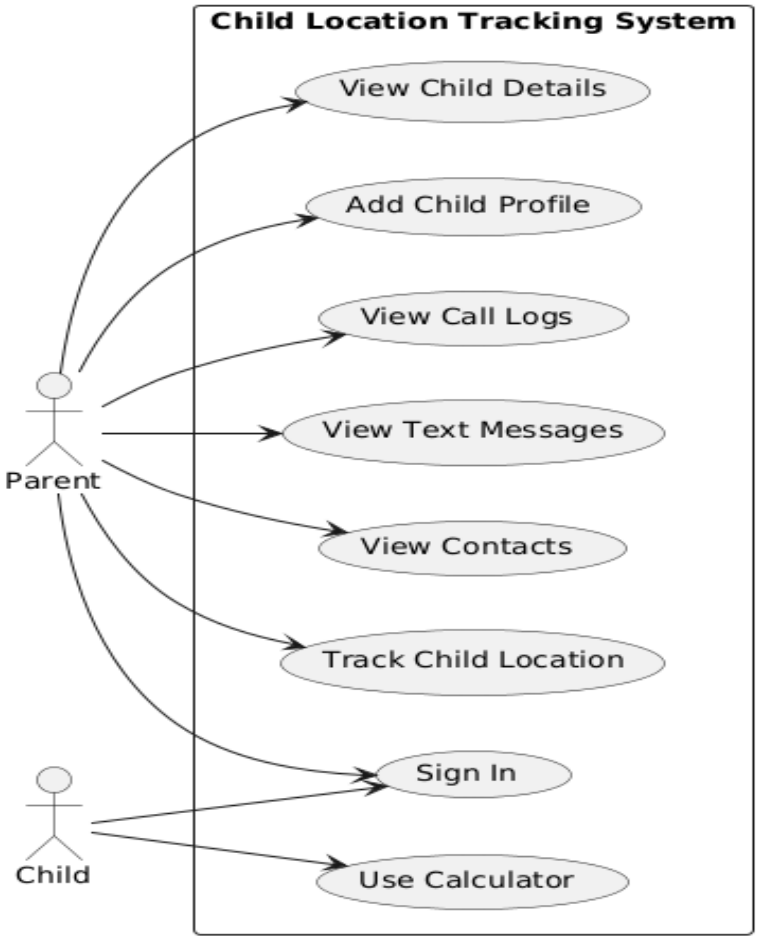


Figure 4.5: Use Case Diagram

Figure 4.5 represent the use case diagram depicts a child location tracking system where the parent can view details, call logs, text messages, contacts, and track the child's location. The Parent can also add a child profile. The child can sign in and use a calculator but has limited access to other features. The system balances parental monitoring with child privacy.

4.4 Control Flow Design

4.4.1 Activity diagram For Use Cases

This flowchart represents the process flow of a parental control system with geofencing and monitoring capabilities. The application begins with the parent logging in. When valid credentials have been entered, the application takes the parent to the main menu, there is an error message prompting them to retry. The parent can start a geofence with a defined location and its radius, which is stored. Notifications are sent to the parent whenever the child enters or leaves the geofence. The child's phone gathers information, and through this, the parent is able to track calls made, messages received, and even current location. It also offers the ability to edit child information and view gathered information.

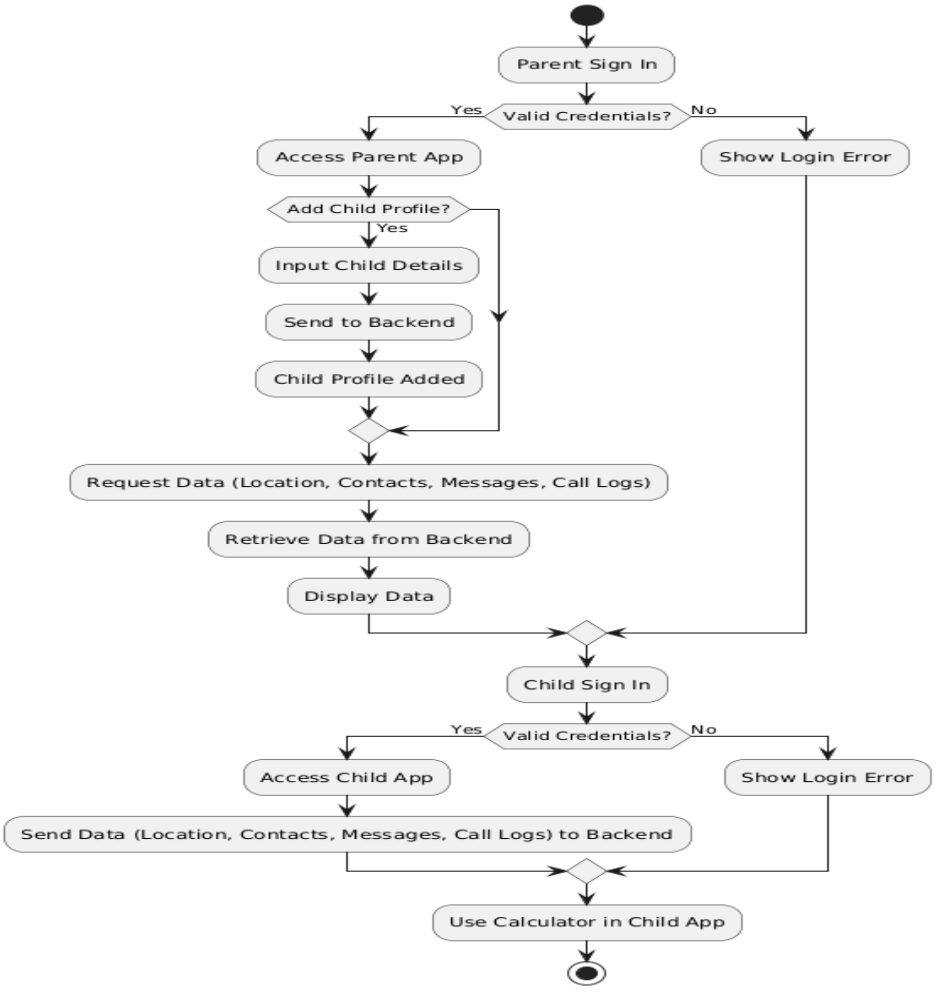


Figure 4.6: Activity Diagram

Figure 4.6 represent the illustrates the process of a parental control app. Parents can sign in and access the app, where they can add their child's profile. Once the profile is added, parents can request data such as location, contacts, messages, and call logs from the child's device. The app retrieves this data from the backend and displays it to the parent. The child can also sign in to the app and use the calculator function.

4.5 Data Flow Diagram

4.5.1 Zero-Level Data Flow Diagram

The Level 0 Data Flow Diagram visually captures the elements and flow of information within the Child Monitoring System. There are three major entities; the User, Child Monitoring System Database (DB), and the Database. This process is initiated by a User sending a query into a central process represented by Child Monitoring System DB. This process communicates with the Database to retrieve the desired information and then sends back the result to the user. The diagram clearly visualizes at a high level the data flow between these components and the role of the central process in facilitating communication.

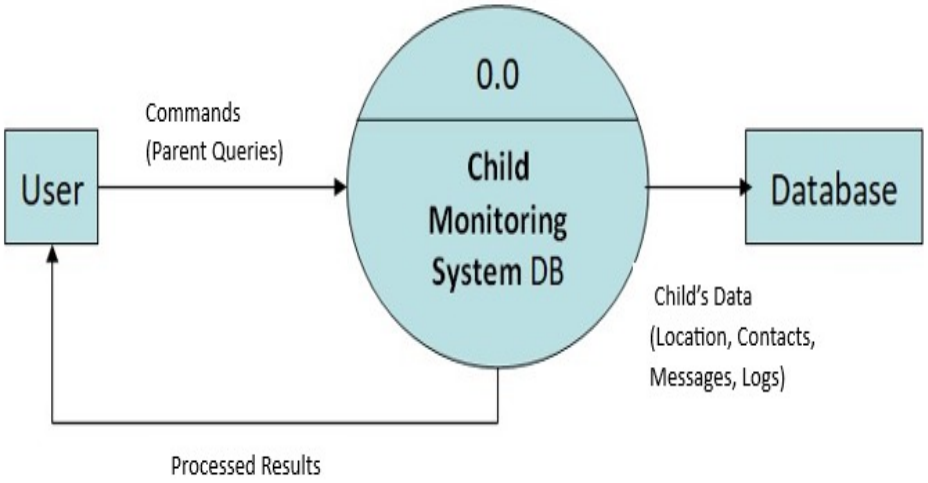


Figure 4.7: Zero-Level Data Flow Diagram

Figure 4.7 represents a simplified version of the Child Monitoring System's functionality. It outlines how the system responds to queries from users by fetching and processing data from a database. This structure ensures that queries are handled in an efficient manner, wherein the user input is processed, relevant information is fetched, and a response is generated. The system is built with the intention of making it easy for users to get their required information by streamlining data retrieval and communication. The diagram represents the interaction between the user, system, and database in a way that demonstrates intuitively the necessary operations of a child monitoring system.

4.5.2 First-Level Data Flow Diagram

In this process flow, a User submits a request to the Process Request stage (1.0). The process checks the requirements from the user. Depending on the necessity, it accesses the information required in the Database by engaging with the Relevant stage (1.1). After getting all the relevant data, the process sends that data to the user as feedback and is considered complete.

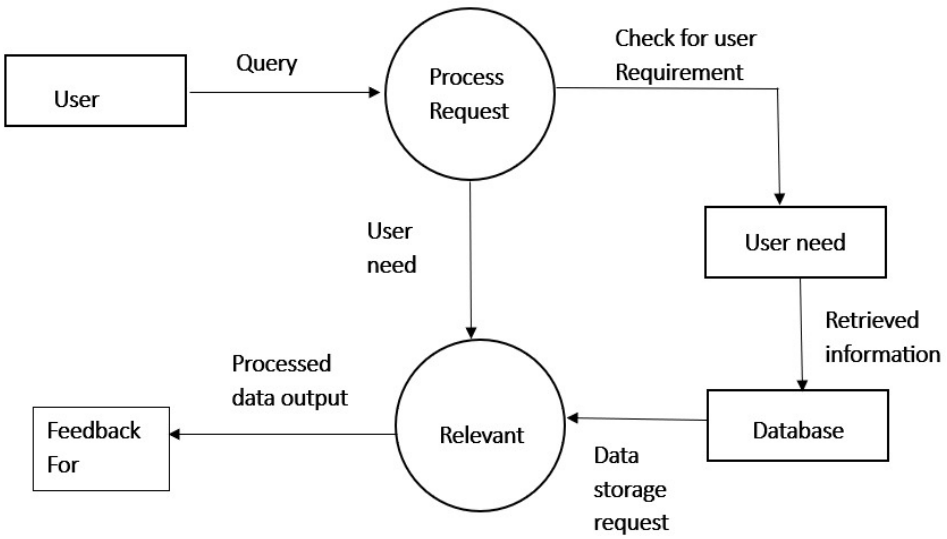


Figure 4.8: First-Level Data Flow Diagram

Figure 4.8 represents a structured query-response process that ensures user requirements are addressed efficiently. It highlights how a system processes

user input, retrieves relevant data from a database, and provides feedback to fulfill the user's needs. This flow emphasizes the importance of systematic communication between the user, system, and database to achieve accurate and relevant results.

4.5.3 Second-Level Data Flow Diagram

This is a simplified query processing system that follows the following steps:

User Input: The process starts with the user sending a query or request to the system.

Step 2.0 - Accept Query: The system accepts the query from the user. This is the first step where the system acknowledges receiving the request.

Step 2.1 - Check Availability: The system checks the availability of required information or resources in the database. The database is queried to determine if the information needed to process the user's query exists or is accessible.

Step 2.2 - Process Query: If the required data or resources are available, the system processes the query. This might comprise access to the required data from the database or relevant calculations.

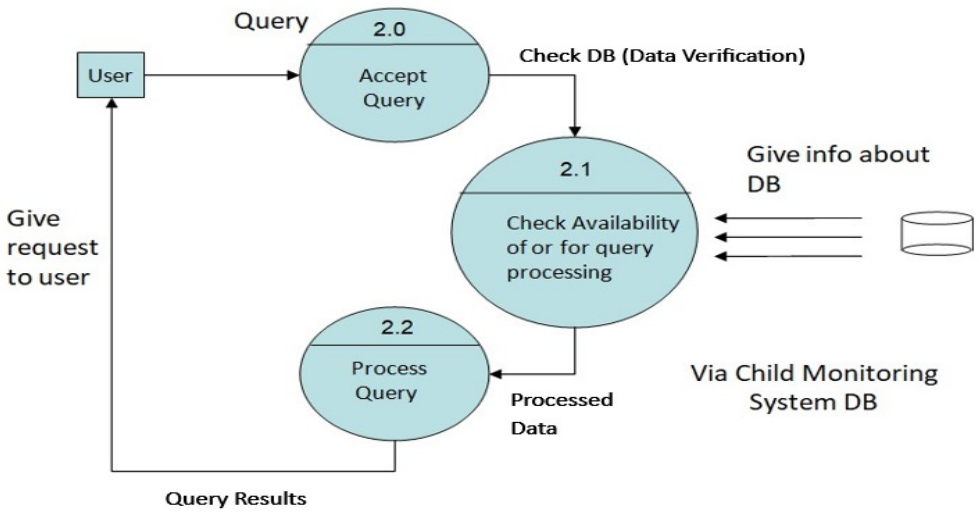


Figure 4.9: Second-Level Data Flow Diagram

Figure 4.9 represents a systematic querying process in a child-monitoring system. It highlights how the query put in by a user is received, validated through the database, and then processed to respond appropriately. The flow further emphasizes checking the availability of the database and ensuring that user requests are handled with accuracy and promptness through a step-by-step mechanism.

4.6 Summary

This chapter designs a parental control system that encompasses geofencing and monitoring capabilities, with special considerations to efficient communication, secure data management, and real-time processing. It consists of the architectural layout, database schema, and modular design to ensure seamless interaction among the Parent and Child Apps, backend services, and databases. Functional diagrams at sequence, activity, and use case illustrate the workflows, while data flow diagrams are at multiple levels to elucidate information handling. With these designs, a child monitoring and safety management scalable, user-friendly, and secure foundation can be established.

Chapter 5

Implementation

5.1 Software Used With Justification

5.1.1 Frontend Development

The frontend of the system is built using XML and also java, which serves as the design layout language for creating intuitive and user-friendly interfaces. It is integrated with Android Studio, a powerful IDE specifically designed for Android development, ensuring seamless implementation of design components. The app interface provides functionality for parents to manage child details, view geofences, and interact with maps, ensuring ease of use.

- XML: It is used as the designing layout language for creating intuitive user interfaces that are visually appealing to users in the system. This allows developers to determine how the app's structure and appearance should be developed clearly and in an orderly manner. Using XML makes it easy to arrange different elements such as buttons, text views, and container layouts to create user-friendly interfaces. Its flexibility and compatibility with Android Studio make it an essential tool for frontend development, ensuring that design components are effectively structured for optimal usability.

- Java: It is the primary programming language used for Android development, which is used to implement the core functionality and logic of the application. It enables the integration of XML-defined layouts

with backend processes, ensuring seamless interaction between the user interface and the application's features. Java's object-oriented nature allows for efficient code organization and reuse, making it suitable for developing complex functionalities like managing child details, handling geofence-related operations, and interacting with mapping APIs.

5.1.2 Backend Development

The backend uses Microsoft SQL Server (MS SQL) as the database for safe storage and management of details, including child information and call logs, messages, and geofence information. It is a robust, feature-rich relational database management system highly used for handling large volumes of data with high security and reliability, supporting complex queries, stored procedures, and scalability - making it suitable for applications where data handling is efficient.

- MS SQL :It is integrated perfectly with the application backend and thus provides the capability to store, retrieve, and manage structured data effectively. Unlike local databases, MS SQL runs on a server; hence, it offers the central management of data. Thus, the performance will be improved, especially for applications requiring synchronized and remote access to data. This would allow the application to scale with the improved ability to handle secure real-time data.

5.1.3 Framework Used

The app development relies on the integrated Android SDK (Software Development Kit) with Android Studio, the official IDE of Android. The Android SDK provides an all-inclusive set of tools, libraries, and API's that aid developers in building robust mobile applications, such as feature-rich applications with geofencing, location tracking, and secure data storage capabilities. The frameworks used

- Android Jetpack Components: This set of libraries simplifies applica-

tion development by providing pre-built functionality for tasks such as lifecycle management, navigation, and UI design. This gives an application a modular and scalable architecture.

- Google Play Services: This is used for incorporation of advanced mapping features, geofencing, and location-based services to enhance interactive capabilities of the application.
- Material Design Framework: It is used to implement modern UI/UX standards such that the interface is intuitive, consistent, and beautiful.

5.1.4 Coding Languages Used for Development

- Java: It is the primary programming language used for Android application development. It is employed to write the core functionality of the system, including the business logic, data processing, and user interaction. Java ensures the app operates smoothly, provides reliability, and supports the extensive features required for the application, such as geofencing, call log tracking, and background data collection.
- XML: It is used for structuring the layouts and visual components of the app in Android Studio. It defines the design elements, including buttons, text views, and other UI components, enabling a clear and intuitive user interface. XML ensures that the app's frontend is organized, responsive, and visually appealing.
- SQL: It is utilized in the backend for database management. It allows for the secure storage and efficient querying of data, such as user credentials, child details, geofence information, and logs. SQL plays a critical role in ensuring that data is well-organized and readily accessible, supporting the seamless operation of the application.

5.1.5 Operating System

Android v 6.0 was known as Lollipop, an important milestone in the journey of Android as it ushered in the Material Design language, which gave a revolution to the user interface as it was cleaner, more intuitive, and visually appealing. Launched in November 2014, the operating system was packed with enhanced functionality and performance features, including support for ART (Android Runtime), where the Dalvik runtime has been replaced to give way to faster application performance along with better memory management.

Lollipop introduced the notifications on the lock screen, priority modes, and heads-up notifications that improved user engagement. Security enhancements included Smart Lock, SELinux enforcing for all applications, and device encryption by default, enhancing user data protection. In addition, it included native support for 64-bit processors to allow apps to make the most of modern hardware. API level 21 gave developers features such as the JobScheduler API for optimized battery use, new camera APIs, and increased multimedia support like HEVC video playback.

More modern versions, like Marshmallow v6.0 and more up-to-date, added things to Lollipop such as App Permission Management, Doze - mode for long battery usage-etc, and greater hardware. It supports applications targeted toward Android v5.0 and above, thereby promising the latest, safe experience with KitKat and any versions above it.

5.2 Hardware Used With Justification

- Laptop or PC
- Windows 7 or more
- I3 processor system or moresssss
- 8 GB RAM or more
- 100 GB ROM or more

- Android Phone (6.0 and above)

- Hardware Requirements: The development and operation of the system both require a laptop/PC and an Android phone. The laptop or PC should run on Windows 7 or higher, at least with an Intel i3 processor, 8 GB of RAM, and 100 GB of storage, so it can execute development tasks smoothly and manage resources efficiently. For mobile use, the application needs an Android phone with version 6.0 or above to support all the features of the application, such as geofencing, GPS tracking, and so on. This combination ensures optimal performance during both development and deployment.

5.3 Procedures Used in the Project in Different Modules

- User Sign Up Module:

  The User initiates the sign up by filling out personal information details. The system authenticates these details for their completeness and correctness. The system stores them in a database if they are correct. A confirmation email will be sent to the user, otherwise, an error message will be returned to the user.

- User Sign In Module:

  The user selects the option of Google sign-in; then the browser redirects them to the page of authentication with Google. After getting authenticated, it fetches information of that user from the database and checks for any existing account. It grants access if the account already exists, and otherwise gives an error message.

- Module Child Details Management:

  The Parent shall be able to manage their child's information by choosing any action (add, update, delete, or view). For adding or updating the system first checks the validity of the child information. If valid,

it inserts or updates the information in the database. For deletion, it shall check if the child is present in the database then proceed. If the child exists, it deletes the record and if not, returns with an error message.

- Manage Geofence Module:

  The ability to create, update, delete, or view geofences. The system verifies the geofence information as valid. If the geofence information is valid, it is then saved or updated in the database. For deletion, the system checks if there exists a geofence, then deletes it if there is; otherwise, returns an error message.

- Notification on Geofence Entry/Exit Module:

  The system is always tracking the location of the child against the geofences defined. On every update of the child's location, the system checks whether the child is within or outside the geofence. If the child enters a geofence, a notification is sent to the parents. In case the child leaves a geofence, another notification is sent, ensuring that parents are aware of the movement of their child.

5.4 Coding

5.4.1 Introduction

The goal of coding or programming phase is to translate the design of the system produced during the design phase into code in a given programming language, which can be executed by a computer and that performs the computation specified by the design.

There are many different criteria for judging a program, including readability, size of the program, execution time and required memory. Having readability and understanding as a clear objective of the coding activity can itself help in producing software that is more maintainability.

5.4.2 Pseudocode

```
    <?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/
android"
    package="com.child.calculator">

    <uses-permission android:name="android.permission.
    INTERNET" />
    <uses-permission android:name="android.permission.
    ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.
    READ_CONTACTS"/>
    <uses-permission android:name="android.permission.
    READ_SMS"/>
    <uses-permission android:name="android.permission.
    READ_CALL_LOG"/>
    <uses-permission android:name="android.permission.
    WRITE_CALL_LOG"/>
    <uses-permission android:name="android.permission.
    ACCESS_COARSE_LOCATION"/>
    <uses-permission android:name="android.permission.
    RECEIVE_BOOT_COMPLETED" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/calculator"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">

        <activity android:name=".MainActivity"
            android:label="Calculator">
```

```
</activity>

<activity android:name=".LoginActivity">
    <intent-filter>
        <action android:name=
        "android.intent.action.MAIN"/>
        <category android:name=
        "android.intent.category.
        LAUNCHER"/>
    </intent-filter>
</activity>

<activity android:name=".testActivity"
    android:label="Test">

    <!--<intent-filter>-->
    <!--<action android:name=
    "android.intent.action.MAIN"/>
    <!--<category android:name=
    "android.intent.category.
    LAUNCHER"/>
    <!--</intent-filter>-->
</activity>

<service android:name=".BackgroundService"/>

<receiver
    android:name=".MyReceiver"
    android:enabled="true"
    android:exported="true">
    <intent-filter>
        <action android:name=
```

```
                "android.intent.action.
                BOOT_COMPLETED"/>
            </intent-filter>
        </receiver>
    </application>
</manifest>
```

The given XML pseudocode is an Android app manifest file that defines the configuration of the application along with its package name com.child.calculator and required permissions such as internet access, location services, contact reading, SMS reading, call log access, and boot event handling. It also defines all the components of the application, including activities like LoginActivity, MainActivity, and testActivity. A background service and a broadcast receiver are also declared, supporting background tasks and device boot events. App-wide settings such as icon, name, and theme are defined within the application tag.

5.5 Summary

It highlights the software and hardware resources, methodologies applied, coding standards followed, and guidelines toward the development of the system. The frontend uses XML along with Java in Android Studio to provide a user interface with the application. The use of SQL for efficient database management is done at the backend. The functionality added by frameworks such as Android SDK, Jetpack Components, and Google Play Services has ensured robust location tracking and geofencing. Core functionality is developed through coding languages Java, XML and SQL. Interface design and database management are done using different types of coding languages. The hardware specifications required are a minimum version PC and an Android running version 6.0 and above. Coding practices are maintained according to readability, modularity and secure data handling by proper naming convention, user-defined types, information hiding, and efficient module interfaces.

Chapter 6

Results and Discussion

6.1 Results

This section includes the presentation of results using suitable figures.

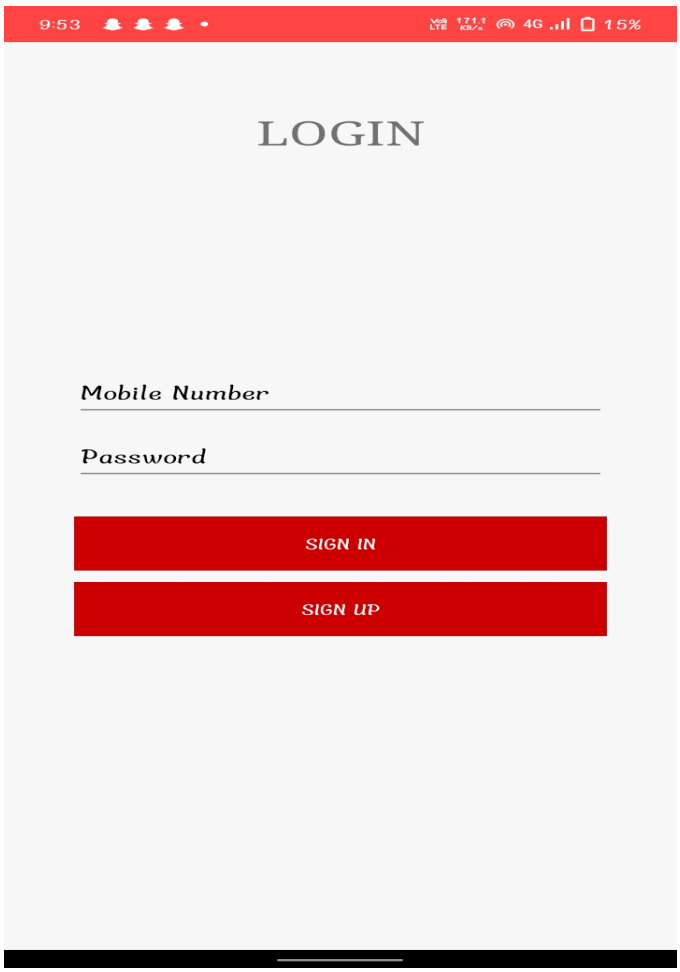


Figure 6.1: Login Page

Figure 6.1 represents the Login Page design for a mobile application. The

layout features a clean and minimalistic interface with a header displaying login at the top in bold and large text, ensuring clear identification of the page's purpose. The red buttons contrast with the light background, drawing the user's attention to the primary actions. This design balances simplicity and functionality while ensuring clarity for users.

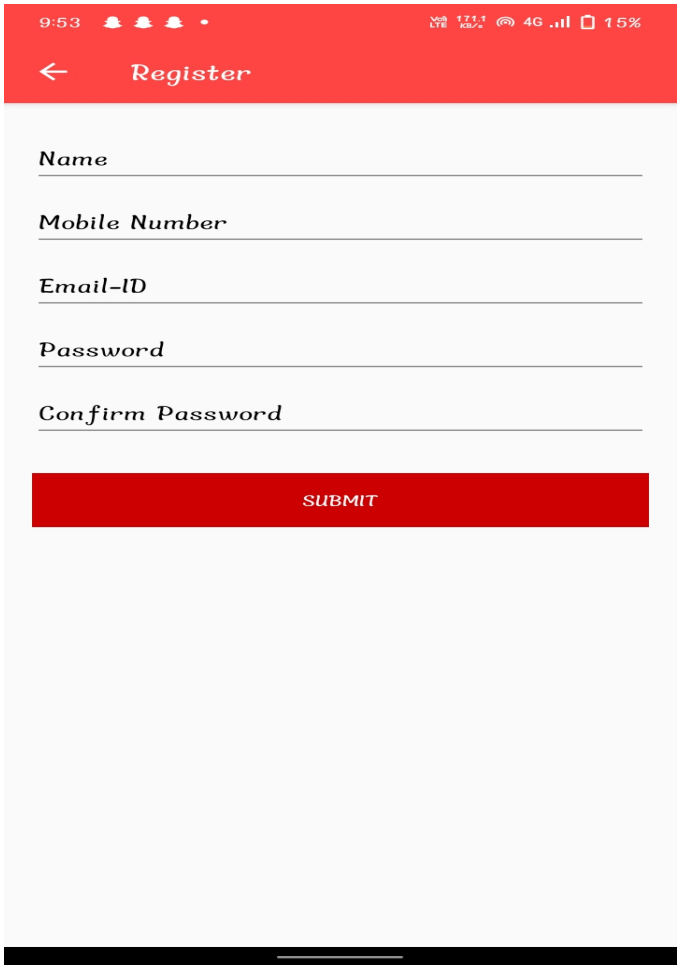


Figure 6.2: Register Page

Figure 6.2 represents the user registration page for a mobile application. The Dteatils to br filled in your personal details, such as Name, Mobile Number, Email-ID, Password, and Confirm Password. Each input field is separated by a thin black line for clarity. At the end of the page, there's a big red submit button to allow completing the registration. There's a focus on usability along with easy navigation. The entire color scheme of red and white appears highly striking and readable.

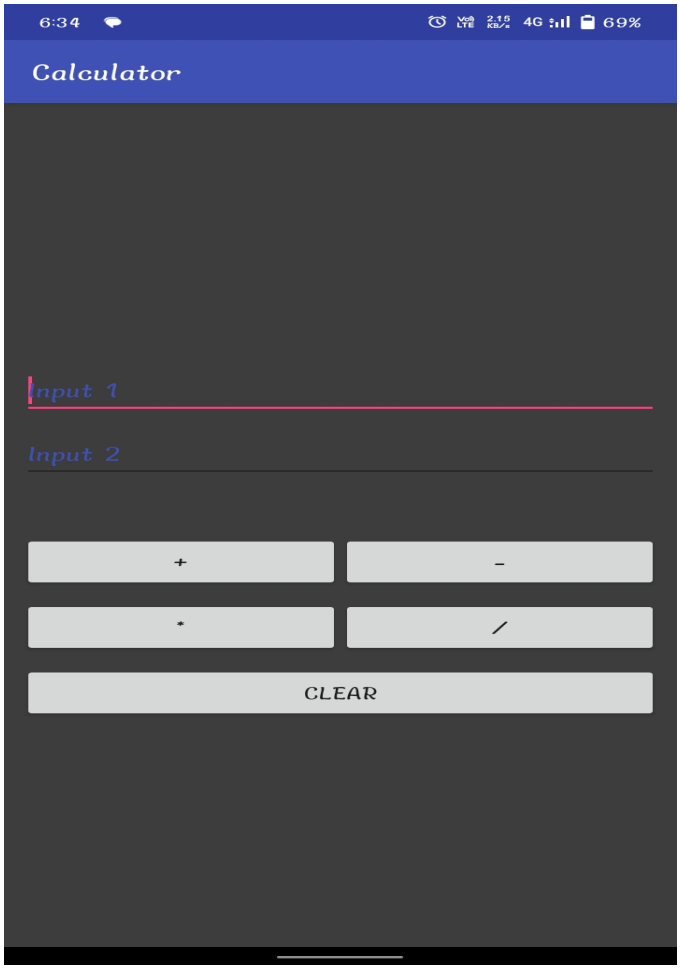


Figure 6.3: Calculator Page

Figure 6.3 represents the calculator page of the application, designed simply and functionally. Where users can enter numerical values. Below the input fields, there are buttons for basic arithmetic operations: addition, subtraction, multiplication, and division. The layout is intuitive and user-friendly, with a dark background and contrasting elements to make it readable and usable.

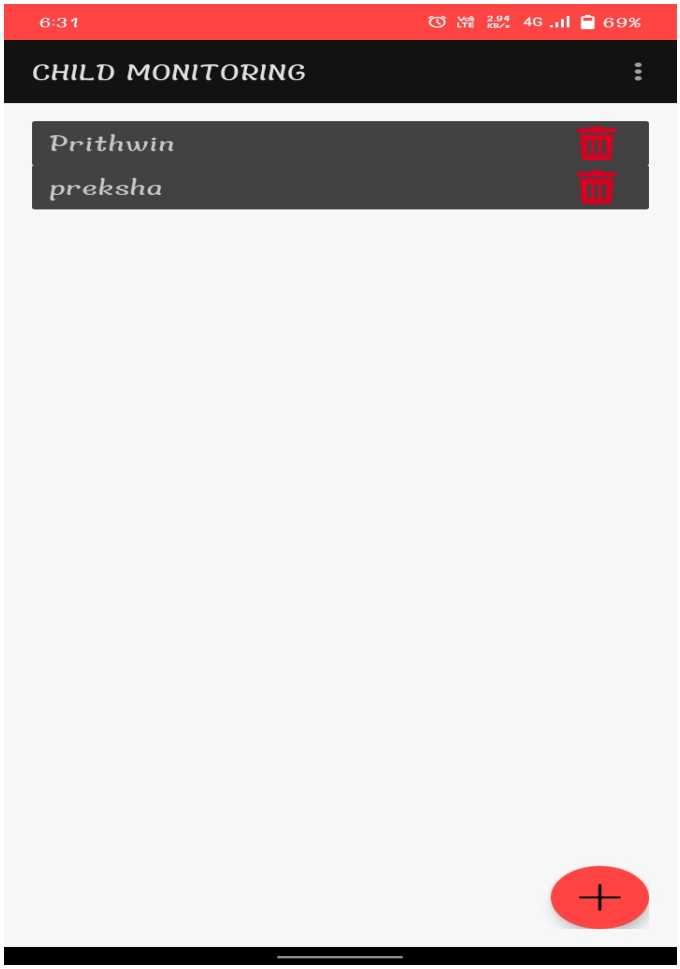


Figure 6.4: Child Monitoring Page

Figure 6.4 represents the child monitoring feature of the application. It has a clean and well-structured layout, with a red header holding the title on it. Main content includes a list of monitored entries, each being accompanied by a red trash bin icon to delete the respective entry. The floating red button at the bottom right enables users to add new entries. Contrasting colors are used to ensure that the whole interface is clear and user-friendly, with a functional and navigational ease.

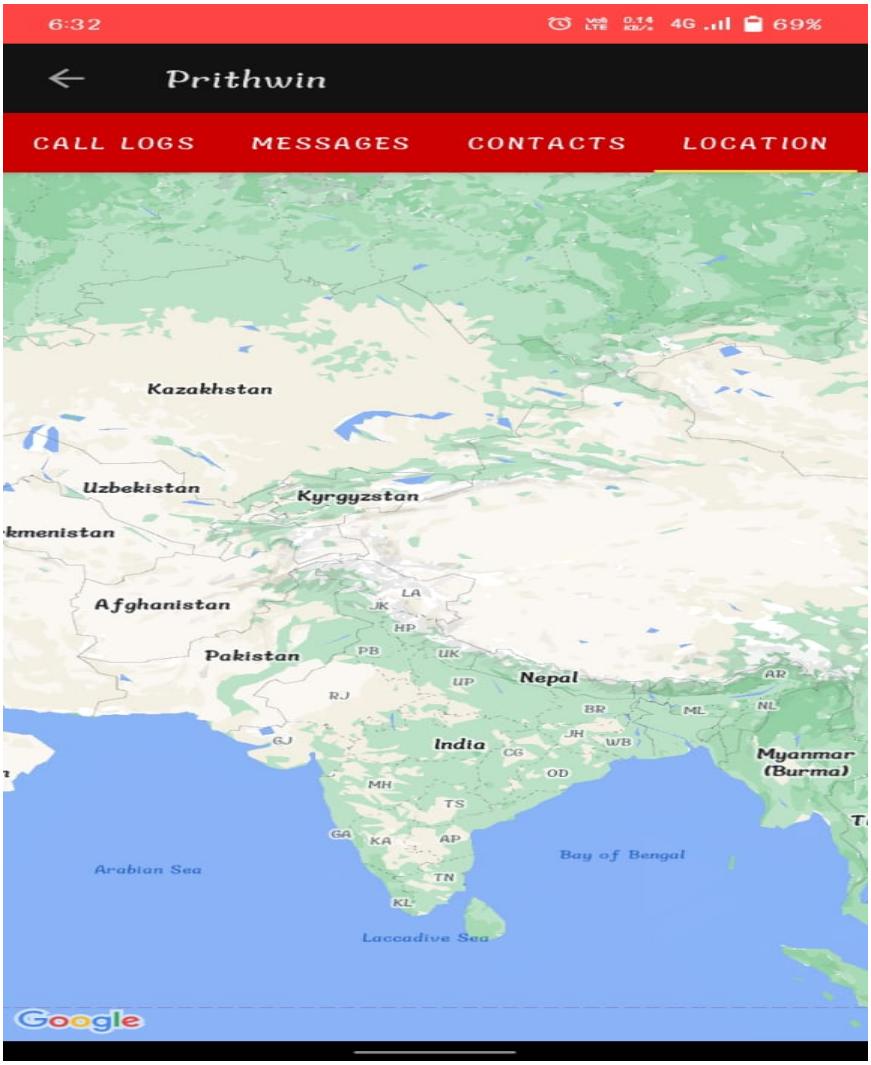


Figure 6.5: Location Tracking Page

Figure 6.5 represents the child monitoring app displayed provides parents with a comprehensive tool to track their child's activities and location. With features such as call logs, messages, contacts, and live location tracking, the app ensures safety and effective parental supervision. Its user-friendly interface integrates GPS mapping, allowing parents to keep an eye on their child's whereabouts in real time. This tool can be invaluable for fostering peace of mind while ensuring children's security in today's fast-paced world.

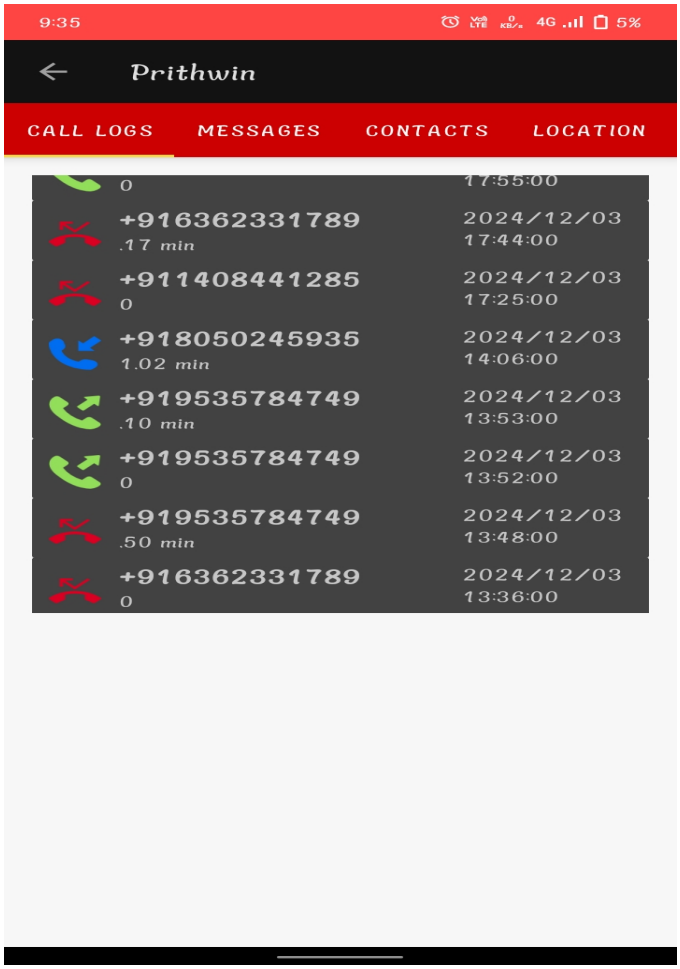


Figure 6.6: Call Log Page

Figure 6.6 represents the parents or guardians with organized access to their child's digital activities. The interface includes tabs for viewing Call Logs, Messages, Contacts, and Location, allowing a comprehensive overview of a child's interactions and whereabouts. In the contacts section shown in the screenshot, the app lists saved contacts along with their respective phone numbers. This feature ensures that guardians can track who the child communicates with, promoting safety and providing a way to intervene when necessary. The app's design prioritizes clarity, with bold titles, segmented tabs, and a simple layout to enhance usability. It serves as a tool for ensuring children's safety in a connected digital environment.

6.2 Testing

As the project is on a bit large scale, we always need testing to make it successful. If each component works properly in all respect and gives desired output for all kinds of inputs then the project is said to be successful. So the conclusion is to make the project successful, it needs to be tested.

The testing done here was System Testing checking whether the user requirements were satisfied. The code for the new system has been written completely using JAVA as the coding language and Android Studio as the interface for front-end designing. using Python as the coding language and Django Framework as the interface for front-end designing. The new system has been tested well with the help of the users and all the applications have been verified from every nook and corner of the user.

Although some applications were found to be erroneous these applications have been corrected before being implemented. The flow of the forms is very much by the actual flow of data.

6.3 Types of Software Testing

- Black Box Testing: The testing for the secured location tracking system should focus on ensuring functionality with security in place to minimize the risks of data breaches and unauthorized access. Tests are conducted to ensure proper monitoring of locations in various scenarios, including the geofencing and emergency alerts. Authentication and access controls are tested to prevent unauthorized usage, whereas data encryption is validated to ensure secure storage and transmission. This involves testing for compliance with the changes in permission that the system offers and tests usability and performance so that the system ensures that the tracking is seamless, reliable, and privacy-preserving in keeping with the needs of modern secure location-based services.

- White box testing:The testing for the secured location tracking system

is a focus on the internal code analysis to ensure robust functionality and security. This includes authentication module verification, preventing vulnerabilities such as hardcoded credentials or session misuse, and encryption algorithms testing for secure data transmission and storage. The location tracking algorithms are checked for accuracy and robustness against edge cases such as weak GPS signals or network failures. Geofencing and alert logic are tested to ensure precise detection and proper event handling, and data validation mechanisms are checked for injection attacks or corruption. Permission and access control logic is reviewed to enforce user preferences and ensure compliance with privacy requirements.

6.4 Testing Methodology

The different types of testing are as follows:

6.4.1 Unit Testing

Unit testing focuses verification efforts on the smallest unit of the software design, the module. This is also known as "Module Testing". The modules are tested separately. This testing is carried out during the programming stage itself. In this testing, each module is found to be working satisfactorily as regards the expected output from the module.

6.4.2 Integration Testing

Data can be grossed across an interface; one module can have adverse efforts on another. Integration testing is systematic testing for the construction of the program structure while at the same time conducting tests to uncover errors associated with the interface. The objective is to take unit-tested modules and build a program structure. All the modules are combined and tested as a whole. Here correction is difficult because the isolation of cause is complicated by the vast expense of the entire program.

Thus, in the integration testing stop, all the errors uncovered are corrected for the text testing steps.

6.4.3 System Testing

System testing is the stage of implementation that is aimed at ensuring that the system works accurately and efficiently for live operation commences. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, then the goal will be successfully achieved.

6.4.4 Validation Testing

At the conclusion of integration testing software is completely assembled as a package, interfacing errors have been uncovered and corrected and a final series of software tests begins, validation test begins. Validation tests can be defined in many ways. But the simple definition is that validation succeeds when the software function in a manner that can reasonably be expected by the customer. After the validation test has been conducted one of two possible conditions exists.

One is the function or performance characteristics conform to specifications and are accepted and the other is a deviation from the specification is uncovered and a deficiency list is created. A proposed system under consideration has been tested by using validation testing and found to be working satisfactorily.

6.4.5 Acceptance Testing

After performing validation testing, the next step is output testing of the proposed system since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated by the system under consideration. Here the output format is considered in two ways, one is on the screen and the other is the printed format. The output format on

the screen is found to be correct as the format was designed in the system design phase according to the user's needs. For the hard copy also, the output comes as the specified requirements by the users. Hence output testing does not result in any corrections in the system. User acceptance of a system is the key factor to the success of any system. The system under study is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required.

6.5 Testing Criteria

6.5.1 Testing for Valid User Name

Table 6.1: Testing Criteria for User Name

Test Case	Input	Test Description	Expected Output	Actual output	Pass/Fail
1	User name starts with number	User name cannot start with number	Throws an error	Does not accept the input	Pass
2	User name is left blank	User name cannot be left blank	Throws an error	Does not accept the input	Pass

Table 6.1 will make sure the input provided by the user is in a format consistent with predefined criteria. Scenarios to check invalid user name input should be tested against the system to test the strength in handling such inputs. Here are two test cases taken:. In the first test case, the user name began with a number, which is against the validation rule that the user name should start with an alphabetic character. The system was able to identify the error and reject the input; thus, the test was passed. The second test case checked the behavior of the system when the field for the user name was left blank. As expected, the system produced an error message and did not accept the input, thereby passing this test as well. These results confirm that the mechanism for validation of user name in the system is functioning correctly and invalid inputs are effectively handled to prevent the occurrence of potential errors or inconsistencies in operation.

6.5.2 Testing for Data Insertion

Table 6.2: Testing Criteria for Data Insertion

Test Case	Input	Test Description	Expected Output	Actual output	Pass/Fail
1	Mandatory fields left empty	Mandatory fields cannot be left empty	Must enter data	Does not redirect to next page	Pass
2	Input for all fields	Valid input	Record inserted successfully	Record inserted succesfully	Pass

Table 6.2 determines whether the system correctly handles data input. Two test cases were run: the first involved leaving mandatory fields blank,which was expected to result in a warning or error, and it successfully blocked redirection to the next page. The second test scenario entailed entering valid data in all fields, and the system successfully inserted the record as planned. Both tests passed, demonstrating that the system prop- erly enforces mandatory fields and processes valid inputs, assuring data integrity and functioning.

6.5.3 Testing for Login

Table 6.3: Testing Criteria for Login

Test Case	Input	Test Description	Expected Output	Actual output	Pass/Fail
1	Enter the correct email and password and hit the enter button	Correct mail and password must be entered	Logged in	Logged in and redirect to next page	Pass
2	If email is provided but the the password is blank	Both email and password pages must be filled	Please enter the password	Display alert to enter a password	Pass

Table 6.3 shows the login functionality of the application was tested against different scenarios to validate it is reliable and accurate. First, the application was tested by using valid credentials: It logged in successfully and redisplayed the next page by passing the test. The second test case was of an email provided, but a blank password field. As expected, the application displayed an alert asking the user to enter a password, so the test was passed also. These results show how the system adheres to defined requirements and its ability to handle user input effectively. Further testing will involve checking more edge cases to test the robustness of the login feature.

6.6 Summary

This chapter details the project's outcome, emphasizing major functionalities and testing procedures. It is presented as figures along with explanations about the outcomes of the system under discussion; the login interface and all the results derived. Detailed testing, consisting of system, unit, integration, validation, and acceptance, was done. These testings ensured the application served the needs and purpose for which it was devised to run. Testing methodologies using the black-box and white-box methodologies validate functionality and codes logic, respectively. Detailed test cases were developed to verify user input validation, data insertion, deletion, and duration handling. Each of the test cases ensures robust functionality. The tests identify and rectify all errors that may have appeared. This testing phase validated that the system's output format ensured both on-screen and printed results are as expected by the user. Overall, the system performed well in its run and met requirements.

Chapter 7

Conclusion and Future work

In the demonstration of the next generation of the secured location tracking system, major improvement in the services can be seen as it overcomes major concerns regarding data security, user privacy, and real-time accuracy which are critical issues of any location-based service with the help of modular design principles and well-defined testing scenarios and resilience to changes in the network's operational parameters and environment are provided as well. A few of the various applications for this adaptable structure are personal security, asset protection, and smart cities, proving its potential for expansion in the future. It also enhances its performance by having a scalable system that does not break easily under heavy user loads, allowing it to withstand realism as well.

The Location Tracking System focuses on several enhancements can be explored in the future to further optimize its functionality and expand its applicability. Utilize IoT devices for enhanced asset tracking and integration with smart infrastructure. Develop offline tracking capabilities by caching location data and synchronizing when connectivity is restored. Enable support for areas with limited or no network coverage through technologies Bluetooth mesh. Expand compatibility with wearable devices for personal safety and health monitoring. Integrate with Automotive Systems for Better Vehicle Monitoring and Telematics.

References

[1] Paramvir Bahl and Venkata N Padmanabhan. "RADAR: An in-building RF-based user location and tracking system". In: *Proceedings IEEE INFOCOM 2000. Conference on computer communications. Nineteenth annual joint conference of the IEEE computer and communications societies (Cat. No. 00CH37064)*. Vol. 2. Ieee. 2000, pp. 775–784.

[2] Sarika Chaudhary and Pooja Batra Nagpal. "Live location tracker". In: *Global Research and Development Journal for Engineering* 4.10 (2019).

[3] Mia Md Karimul Hoq, Mohammad Jahangir Alam, and Md Nurul Mustafa. "Mobile tracking system using web application and android apps". In: *Int. J. Eng. Res. Technol.(IJERT)* 6.2 (2017), pp. 2278–0181.

[4] Pooja Patil Ishwar Bagad Samruddhi Pimpale and Jayashree Rane. "Bus Location Tracking System". In: *IJFMR* (2024). DOI: 10.36948/ijfmr.2024.v06i02.18196.

[5] Amit Kushwaha and Vineet Kushwaha. "Location based services using android mobile operating system". In: *International Journal of Advances in Engineering & Technology* 1.1 (2011), p. 14.

[6] Aman Mishra and Advin Manhar. "Survey on Live Tracking Phone No. Via Android/IOS Based App". In: *International Journal of Scientific Research in Computer Science, Engineering and Information Technology* (Dec. 2020), pp. 323–327. DOI: 10.32628/CSEIT206651.

[7] G Ramesh et al. "Farm animal location tracking system using arduino and gps module". In: *2021 International Conference on Computer Communication and Informatics (ICCCI)*. IEEE. 2021, pp. 1–4.

[8] T Sindhu and L SaiRamesh. "REAL TIME VEHICLE TRACKING SCHEME BY ANALYSING SPATIAL POSITION THROUGH GPS AND GSM". In: ().

[9] Dr Srinivas et al. "Android App for Women Safety". In: *International Journal of Scientific Research in Computer Science, Engineering and Information Technology* (May 2021), pp. 378–386. DOI: 10.32628/CSEIT1217368.

[10] Raja Zahilah and Siti Zahidah Zaharan. "EyeKids: Real-time tracking and monitoring system for child safety". In: *International Journal of Innovative Computing* 12.2 (2022), pp. 1–8.

Appendix A

Drill-bit Plagiarism Report

DrillBit

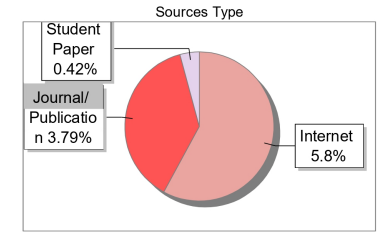
The Report is Generated by DrillBit Plagiarism Detection Software

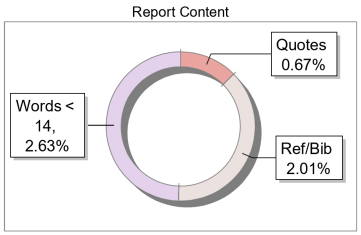
Submission Information

Author Name	Rakshan Shetty K
Title	Design And Implementation Of Secured Location Tracking
Paper/Submission ID	2814570
Submitted by	alok.ranjan@canaraengineering.in
Submission Date	2024-12-19 19:19:53
Total Pages, Total Words	74, 15845
Document type	Project Work

Result Information

Similarity **10 %**





Exclude Information

Quotes	Not Excluded
References/Bibliography	Not Excluded
Source: Excluded < 14 Words	Not Excluded
Excluded Source	**0 %**
Excluded Phrases	Not Excluded

Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File

Appendix B

Expo Details

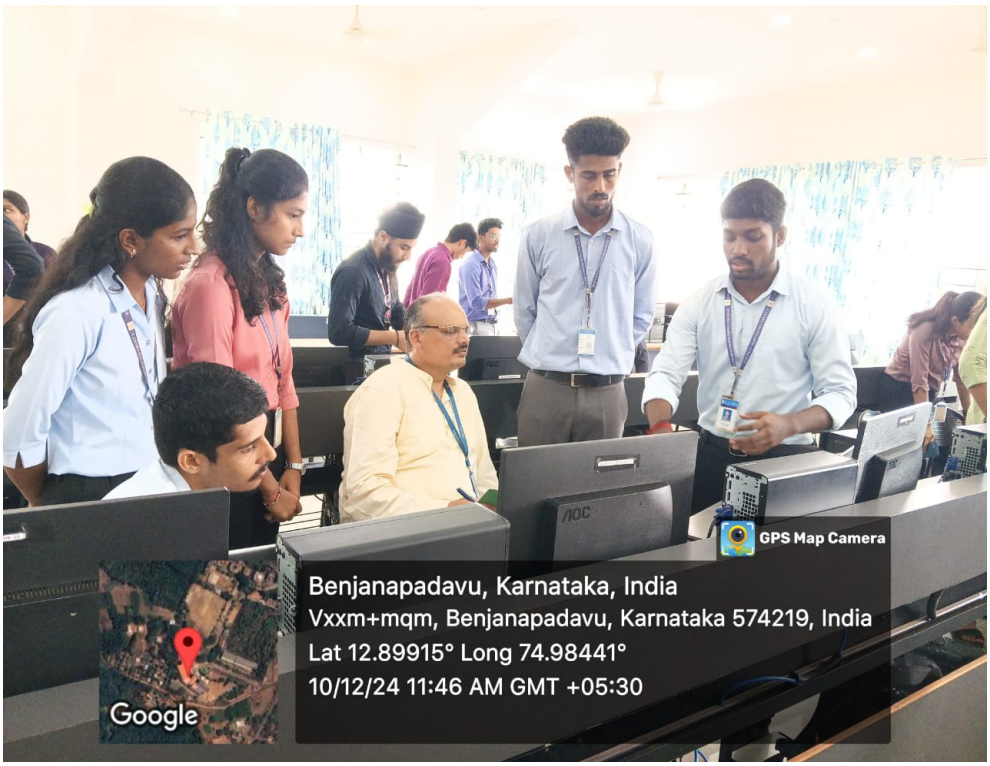


Figure B.1: Innovation Showcase

The Project team has successfully attended Project-Expo Innovation Showcase Project Exhibition of Prototypes 2024-25 conducted by Canara Engineering College on 10/12/2024.