



MOBILE TRACKING SYSTEM



A PROJECT REPORT

Submitted by

GODFREY T R (2303811710421047)

in partial fulfillment of requirements for the award of the course

CGB1201 - JAVA PROGRAMMING

In

COMPUTER SCIENCE AND ENGINEERING

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

NOVEMBER- 2024

**K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY
(AUTONOMOUS)**

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on “**MOBILE TRACKING SYSTEM**” is the bonafide work of **GODFREY T R (2303811710421047)** who carried out the project work during the academic year 2024 - 2025 under my supervision.



CGB1201-JAVA PROGRAMMING
Dr.A.DELPHIN CAROLINA RANI, M.E., Ph.D.,
HEAD OF THE DEPARTMENT
PROFESSOR

SIGNATURE

Dr.A.Delphin Carolina Rani, M.E., Ph.D.,

HEAD OF THE DEPARTMENT

PROFESSOR

Department of CSE

K.Ramakrishnan College of Technology
(Autonomous)

Samayapuram–621112.



CGB1201-JAVA PROGRAMMING
Mr. M. SARAVANAN, M.E.,
ASSISTANT PROFESSOR

SIGNATURE

Mr. M. Saravanan, M.E.,

SUPERVISOR

ASSISTANT PROFESSOR

Department of CSE

K.Ramakrishnan College of Technology
(Autonomous)

Samayapuram–621112.

Submitted for the viva-voce examination held on 02-12-2024 .



CGB1201-JAVA PROGRAMMING
Mr. M. ARUMUGAN, M.E.,
ASSISTANT PROFESSOR

INTERNAL EXAMINER



CGB1201-JAVA PROGRAMMING
Dr. S. SENTHIL SELVAN, M.E., Ph.D.,
PROFESSOR
8138-SCE, TRICHY

EXTERNAL EXAMINER

DECLARATION

I declare that the project report on “**MOBILE TRACKING SYSTEM**” is the result of original work done by us and best of our knowledge, similar work has not been submitted to “**ANNA UNIVERSITY CHENNAI**” for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **CGB1201 - JAVA PROGRAMMING**.

Signature

A handwritten signature in blue ink, appearing to read 'T.R. Godfrey', is written over a horizontal line.

Godfrey T R

Place: Samayapuram

Date: 02-12-2024

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution “**K.Ramakrishnan College of Technology (Autonomous)**”, for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN, B.E.**, for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.**, for forwarding to our project and offering adequate duration in completing our project.

I would like to thank **Dr. N. VASUDEVAN, M.Tech., Ph.D.**, Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. A. DELPHIN CAROLINA RANI, M.E.,Ph.D.**, Head of the department, **COMPUTER SCIENCE AND ENGINEERING** for providing her encourage pursuing this project.

I express our deep expression and sincere gratitude to our project supervisor **Mr. M. SARAVANAN, M.E.**, Department of **COMPUTER SCIENCE AND ENGINEERING**, for his incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

I render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

VISION OF THE INSTITUTION

To serve the society by offering top-notch technical education on par with global standards

MISSION OF THE INSTITUTION

- Be a center of excellence for technical education in emerging technologies by exceeding the needs of the industry and society.
- Be an institute with world class research facilities
- Be an institute nurturing talent and enhancing the competency of students to transform them as all-round personality respecting moral and ethical values

VISION OF DEPARTMENT

To be a center of eminence in creating competent software professionals with research and innovative skills.

MISSION OF DEPARTMENT

M1: Industry Specific: To nurture students in working with various hardware and software platforms inclined with the best practices of industry.

M2: Research: To prepare students for research-oriented activities.

M3: Society: To empower students with the required skills to solve complex technological problems of society.

PROGRAM EDUCATIONAL OBJECTIVES

1. PEO1: Domain Knowledge

To produce graduates who have strong foundation of knowledge and skills in the field of Computer Science and Engineering.

2. PEO2: Employability Skills and Research

To produce graduates who are employable in industries/public sector/research organizations or work as an entrepreneur.

3. PEO3: Ethics and Values

To develop leadership skills and ethically collaborate with society to tackle real-world challenges.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO 1: Domain Knowledge

To analyze, design and develop computing solutions by applying foundational concepts of Computer Science and Engineering.

PSO 2: Quality Software

To apply software engineering principles and practices for developing quality software for scientific and business applications.

PSO 3: Innovation Ideas

To adapt to emerging Information and Communication Technologies (ICT) to innovate ideas and solutions to existing/novel problems

PROGRAM OUTCOMES (POs)

Engineering students will be able to:

- 1. Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- 2. Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
- 3. Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
- 4. Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions

- 5. Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations
- 6. The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice
- 7. Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development
- 8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- 9. Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- 10. Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
- 11. Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- 12. Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

ABSTRACT

The Mobile Tracking System (MTS) is designed to provide a real-time solution for tracking mobile devices, with an emphasis on enhancing device security in cases of loss or theft. This project aims to offer users a reliable method for locating their lost or stolen devices, monitoring device movements, and securing sensitive information remotely, ensuring peace of mind for mobile users. The system is built using Java programming with a graphical user interface (GUI), leveraging the AWT (Abstract Window Toolkit) for designing the user interface and ActionListener for event handling.

In its current prototype, the system allows users to input a mobile number, track its location, report the device as lost, and store the data for future reference. The location tracking feature is simulated through random geolocation coordinates (latitude and longitude) for demonstration purposes, as opposed to real-time GPS integration. A mobile lost reporting feature enables users to register a lost device and receive a confirmation message. The application also displays all tracked and lost device data in a structured format, allowing users to monitor the devices in the system.

While the current system offers basic tracking and reporting capabilities, it has several potential enhancements. These include the integration of real GPS tracking through APIs such as Google Maps, remote device management for locking or wiping data on lost devices, and mobile app development for better accessibility.

This project serves as a foundational framework for a mobile security application that can be expanded to meet the growing demands of mobile device users for enhanced security and peace of mind in case of loss or theft.

ABSTRACT WITH POs AND PSOs MAPPING

CO 5 : BUILD JAVA APPLICATIONS FOR SOLVING REAL-TIME PROBLEMS.

ABSTRACT	POs MAPPED	PSOs MAPPED
<p>The Mobile Tracking System (MTS) is designed to provide a real-time solution for tracking mobile devices, with an emphasis on enhancing device security in cases of loss or theft. This project aims to offer users a reliable method for locating their lost or stolen devices, monitoring device movements, and securing sensitive information remotely, ensuring peace of mind for mobile users.</p> <p>The system is built using Java programming with a graphical user interface (GUI), leveraging the AWT (Abstract Window Toolkit) for designing the user interface and ActionListener for event handling.</p> <p>This project serves as a foundational framework for a mobile security application that can be expanded to meet the growing demands of mobile device users for enhanced security and peace of mind in case of loss or theft.</p>	<p>PO1 -3</p> <p>PO2 -3</p> <p>PO3 -3</p> <p>PO4 -3</p> <p>PO5 -3</p> <p>PO6 -3</p> <p>PO7 -3</p> <p>PO8 -3</p> <p>PO9 -3</p> <p>PO10 -3</p> <p>PO11-3</p> <p>PO12 -3</p>	<p>PSO1 -3</p> <p>PSO2 -3</p> <p>PSO3 -3</p>

Note: 1- Low, 2-Medium, 3- High

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	vii
1	INTRODUCTION	1
	1.1 Objective	1
	1.2 Overview	1
	1.3 Java Programming concepts	1
2	PROJECT METHODOLOGY	3
	2.1 Proposed Work	3
	2.2 Block Diagram	3
3	MODULE DESCRIPTION	4
	3.1 User Interface Design	4
	3.2 Location Tracking	4
	3.3 Lost Mobile Reporting	4
	3.4 Data Storage and Retrieval	4
	3.5 Event Handling	4
	3.6 Lost Reporting Modules	4
4	CONCLUSION & FUTURE SCOPE	5
	4.1 Conclusion	5
	4.2 Future Scope	5
	REFERENCES	13
	APPENDIX A (SOURCE CODE)	6
	APPENDIX B (SCREENSHOTS)	10

CHAPTER 1

INTRODUCTION

1.1 OBJECTIVE

The primary objective of the Mobile Tracking System (MTS) is to enhance the security and utility of mobile devices by providing tools for real-time location tracking, lost or stolen device reporting, and remote data security. This system ensures user peace of mind in case of device theft or misplacement by enabling swift action and secure handling of sensitive data.

1.2 OVERVIEW

Mobile Tracking System (MTS) integrates modern tracking mechanisms, device data monitoring, and security protocols into a single application. It provides functionalities such as:

- Real-time tracking of mobile device locations using simulated geolocation APIs.
- Reporting and storing lost device information for tracking purposes.
- Secure storage and display of tracked and lost mobile data. MTS utilizes Java's robust GUI framework (AWT) to create an interactive, user-friendly application that offers convenience and security in managing mobile devices.

1.3 JAVA PROGRAMMING CONCEPTS

- **Basic OOP Concepts in Java:**
 - **Class and Object:** The Godfrey class represents the application framework, where objects are instantiated for each GUI component.
 - **Inheritance:** The Godfrey class extends Frame, inheriting GUI capabilities.
 - **Polymorphism:** Overriding methods like actionPerformed allows handling multiple events uniquely.
 - **Encapsulation:** The application restricts access to certain attributes (e.g., API_KEY) and exposes only required methods.
 - **Abstraction:** The application uses high-level methods like validateApiKey without exposing implementation details.

- **Project-Related Concepts:**
 - **AWT Framework:** Utilized for GUI design, incorporating components such as buttons, text fields, and labels.
 - **Event Handling:** Implemented through ActionListener for handling button clicks and WindowAdapter for frame-close behavior.
 - **Random Class:** Used to simulate geolocation data generation.
 - **Lists:** ArrayLists are used to store and manage tracked and lost mobile numbers dynamically.

CHAPTER 2

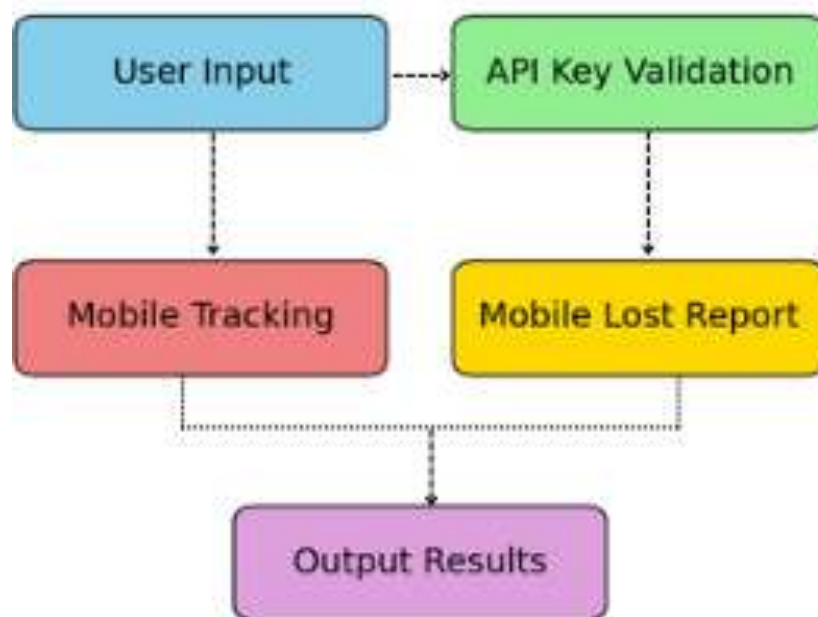
PROJECT METHODOLOGY

2.1 PROPOSED WORK

The proposed Mobile Tracking System (MTS) aims to achieve the following:

- Provide a robust mechanism for real-time device location tracking.
- Allow users to report lost devices and maintain records for future retrieval.
- Deliver secure and easy-to-use GUI for user interactions. The methodology follows a structured approach:
- Requirement Gathering: Identify core functionalities such as tracking, reporting, and data security.
- System Design: Create block diagrams and UI layouts to ensure clarity in workflow.
- Implementation: Develop the application using Java, incorporating features like API simulation, data storage, and event handling.
- Testing and Debugging: Ensure functionality correctness and resolve potential issues.
- Deployment: Package the system for user access.

2.2 BLOCK DIAGRAM



CHAPTER 3

MODULE DESCRIPTION

3.1 MODULE 1: USER INTERFACE DESIGN

This module focuses on creating a graphical user interface using Java AWT. It includes components like text fields, labels, buttons, and text areas, which allow users to interact with the system.

3.2 MODULE 2: LOCATION TRACKING

Simulates real-time location tracking to generate latitude and longitude values. The tracked mobile numbers are stored for future reference.

3.3 MODULE 3: LOST MOBILE REPORTING

Allows users to report lost mobile devices. The application stores these numbers for retrieval and future reference.

3.4 MODULE 4: DATA STORAGE AND RETRIEVAL

Uses ArrayList for dynamic storage and retrieval of tracked and lost mobile numbers. The data is displayed in a structured format in the text area.

3.5 MODULE 5: EVENT HANDLING

Handles user interactions and ensures secure operation through simulated API key validation. Implements ActionListener and WindowAdapter for event management.

3.6 MODULE 6: LOST REPORTING MODULES

Core functionalities for tracking mobile devices and reporting them as lost.

If a user enters a valid mobile number, the system simulates tracking the device and displaying the location (latitude, longitude).

If the user reports a lost mobile, the number is stored in the lost mobile list.

CHAPTER 4

CONCLUSION & FUTURE SCOPE

4.1 CONCLUSION

The Mobile Tracking System (MTS) successfully addresses the growing need for mobile device security and location tracking. Through its simple and interactive graphical user interface, users can track the real-time location of their mobile devices, report lost mobiles, and view stored data.

The system leverages basic concepts of Java programming, such as AWT for GUI, ActionListener for event handling, and ArrayLists for managing dynamic data storage. By simulating location tracking using random values, the system provides a prototype for real-time tracking applications. The application can serve as a foundation for further enhancements, including real GPS integration and advanced security features.

4.2 FUTURE SCOPE

The Mobile Tracking System can be further developed and improved in the following ways:

- **Real-Time GPS Integration:** The system can be enhanced by integrating real GPS tracking through third-party APIs, such as Google Maps API, to provide actual location data.
- **Remote Data Security:** Implement features to remotely lock the device or wipe sensitive data in case of theft.
- **Mobile App Development:** The system can be transformed into a mobile application for easier access and usability.
- **Advanced Notifications:** Push notifications or SMS alerts can be added to inform users of device movements or if the device enters unauthorized zones.
- **AI-Based Detection:** Incorporating machine learning for detecting unusual movements or patterns to provide proactive security alerts.

These enhancements would make the Mobile Tracking System more robust, secure, and user- friendly, extending its use for both personal and enterprise-level device security.

APPENDIX A

(SOURCE CODE)

```
import java.awt.*;
import java.awt.event.*;
import java.util.ArrayList;
import java.util.List;
import javax.swing.JOptionPane;

public class Godfrey extends Frame implements ActionListener {
    private static final String API_KEY = "0bef6614413443725df6f1a8c65b8825"; // API key

    private List<String> trackedMobiles = new ArrayList<>();
    private List<String> lostMobiles = new ArrayList<>();
    private Random random = new Random();

    private Label label;
    private TextField inputField;
    private TextArea outputArea;
    private Button trackButton, reportLostButton, displayButton, exitButton;

    public Godfrey() {
        // Frame setup
        setTitle("Mobile Tracking System");
        setSize(500, 400);
        setLayout(new FlowLayout());
        setResizable(false);

        // UI Components
        label = new Label("Enter Mobile Number:");
        inputField = new TextField(20);
        outputArea = new TextArea(15, 40);
        trackButton = new Button("Track Mobile Location");
        reportLostButton = new Button("Report Lost Mobile");
        displayButton = new Button("Display Stored Data");
```



```

exitButton = new Button("Exit");

// Adding components to the frame
add(label);
add(inputField);
add(trackButton);
add(reportLostButton);
add(displayButton);
add(exitButton);
add(outputArea);

// Event Listeners
trackButton.addActionListener(this);
reportLostButton.addActionListener(this);
displayButton.addActionListener(this);
exitButton.addActionListener(this);

// Frame close behavior with confirmation
addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent we) {
        int response = JOptionPane.showConfirmDialog(null, "Are you sure you want to exit?",
"Confirm Exit",
        JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE);
        if (response == JOptionPane.YES_OPTION) {
            dispose();
        }
    }
});

setVisible(true);
}

@Override
public void actionPerformed(ActionEvent e) {
    String action = e.getActionCommand();
    String mobileNumber = inputField.getText().trim();

```

```

if (action.equals("Track Mobile Location")) {
    if (mobileNumber.isEmpty() || !mobileNumber.matches("\\d{10}")) {
        outputArea.setText("Please enter a valid 10-digit mobile number to track.");
        return;
    }

    // Simulate API key validation
    if (validateApiKey(API_KEY)) {
        double latitude = 10 + (random.nextDouble() * (20 - 10)); // Random latitude
        double longitude = 70 + (random.nextDouble() * (80 - 70)); // Random longitude
        trackedMobiles.add(mobileNumber);
        outputArea.setText("Tracking mobile number: " + mobileNumber + "\n");
        outputArea.append(String.format("Location: Latitude %.4f, Longitude %.4f\n", latitude,
longitude));
    } else {
        outputArea.setText("Failed to authenticate with the tracking service. Please check the
API key.");
    }
} else if (action.equals("Report Lost Mobile")) {
    if (mobileNumber.isEmpty() || !mobileNumber.matches("\\d{10}")) {
        outputArea.setText("Please enter a valid 10-digit mobile number to report as lost.");
        return;
    }
    lostMobiles.add(mobileNumber);
    outputArea.setText("Lost mobile report for number " + mobileNumber + " has been
registered.\n");
    outputArea.append("Please contact your service provider for further support.");
} else if (action.equals("Display Stored Data")) {
    outputArea.setText("==== Stored Data ==== \n");
    outputArea.append("Tracked Mobile Numbers: \n");
    if (trackedMobiles.isEmpty()) {
        outputArea.append("No mobile numbers have been tracked yet.\n");
    } else {
        trackedMobiles.forEach(num -> outputArea.append(num + "\n"));
    }
}

```

```

    }
    outputArea.append("\nLost Mobile Reports:\n");
    if (lostMobiles.isEmpty()) {
        outputArea.append("No lost mobile reports have been filed yet.\n");
    } else {
        lostMobiles.forEach(num -> outputArea.append(num + "\n"));
    }
} else if (action.equals("Exit")) {
    int response = JOptionPane.showConfirmDialog(null, "Are you sure you want to exit?",
"Confirm Exit",
        JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE);
    if (response == JOptionPane.YES_OPTION) {
        dispose();
    }
}
}

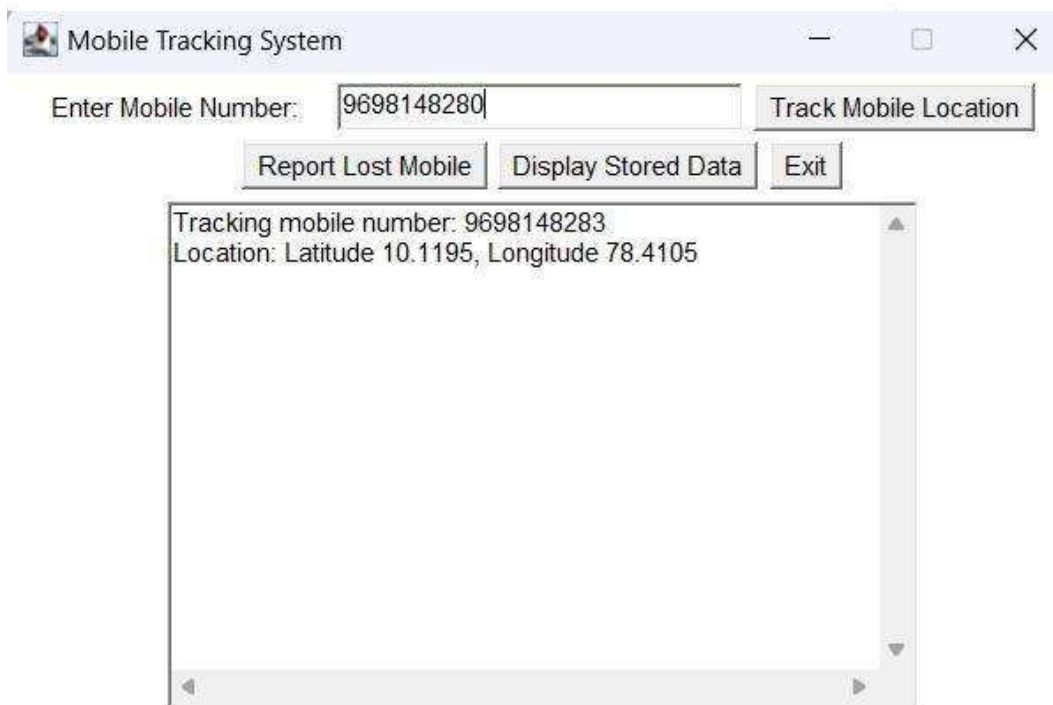
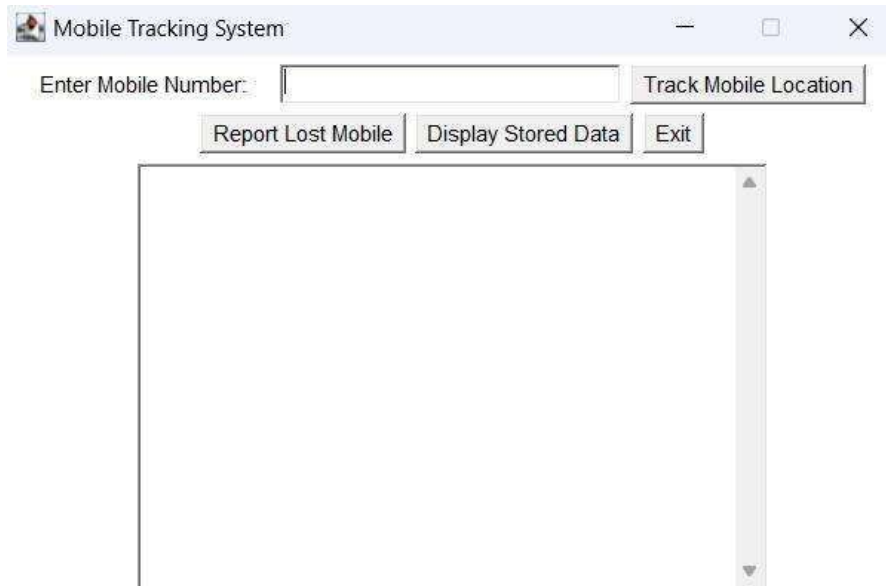
private boolean validateApiKey(String apiKey) {
    // Simulating an API key validation process
    return apiKey.equals("0bef6614413443725df6f1a8c65b8825");
}

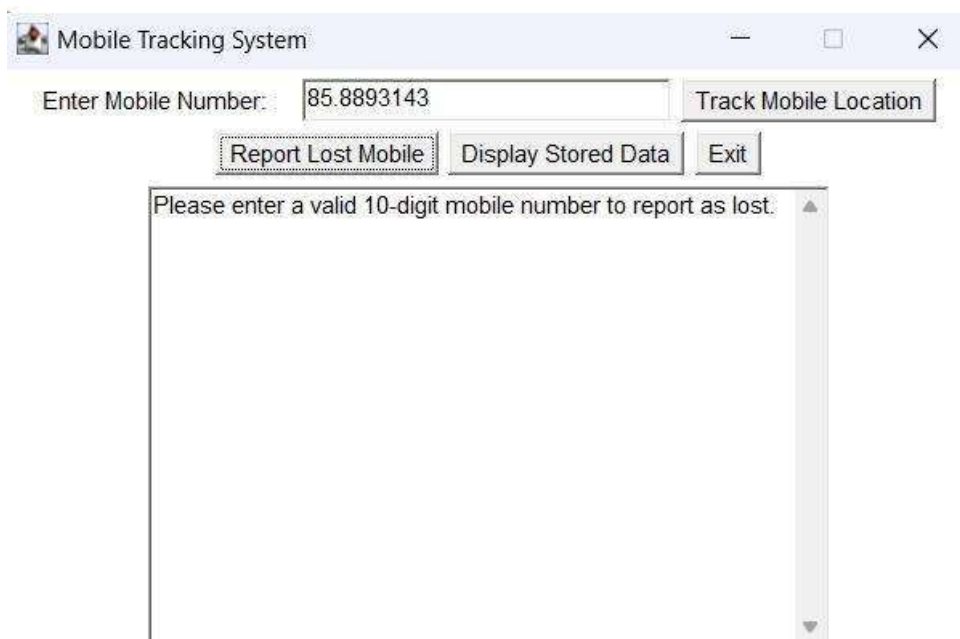
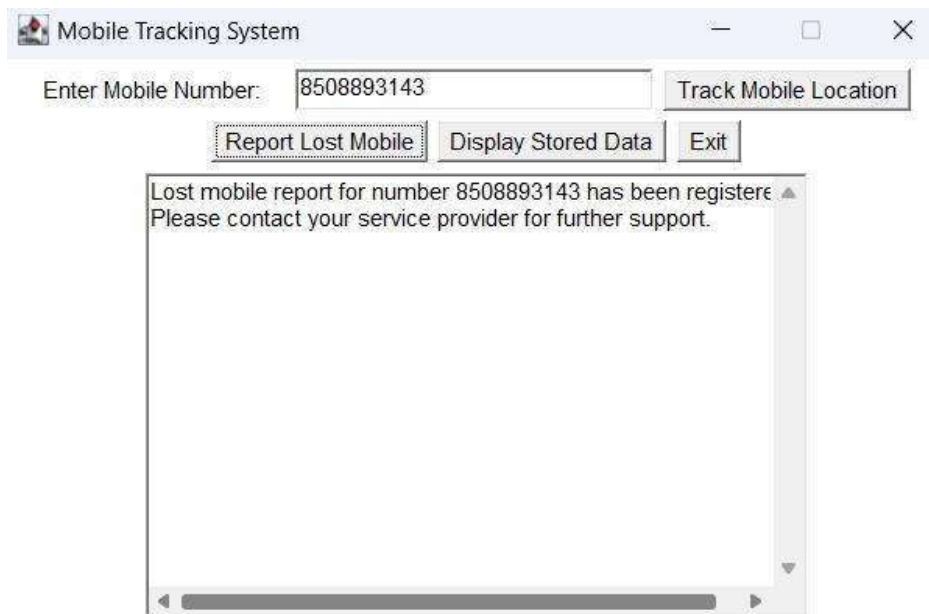
public static void main(String[] args) {
    new Godfrey();
}
}

```

APPENDIX B

(SCREENSHOTS)





Mobile Tracking System

Enter Mobile Number:

Track Mobile Location

Report Lost Mobile Display Stored Data Exit

==== Stored Data ====

Tracked Mobile Numbers:

9344462238

9698148283

Lost Mobile Reports:

8508893143

9698148283

REFERENCES

BOOKS

- *Effective Java* by Joshua Bloch, Addison-Wesley.
- *Head First Java* by Kathy Sierra and Bert Bates, O'Reilly Media.
- *Core Java Volume I - Fundamentals* by Cay S. Horstmann, Pearson.
- *Java Programming and Problem Solving* by James Cohoon and Jack Davidson, McGraw-Hill Education.
- *Object-Oriented Programming with Java* by David J. Barnes and Michael Kölling, Pearson.

WEBSITES

- Java AWT and Swing Tutorials: <https://www.tutorialspoint.com/awt/index.htm>
- Geolocation APIs Overview https://developer.mozilla.org/en-US/docs/Web/API/Geolocation_API
- Oracle Java Tutorials: <https://docs.oracle.com/javase/tutorial/>
- GeeksforGeeks (Java Programming): <https://www.geeksforgeeks.org/java/>

DOCUMENTATION AND API REFERENCES

- Java SE 8 API Documentation: <https://docs.oracle.com/javase/8/docs/api/>
- Geolocation API Documentation: https://developer.mozilla.org/en-US/docs/Web/API/Geolocation_API
- Stack Overflow discussions for Java GUI issues: <https://stackoverflow.com/questions/tagged/java+gui>

SOFTWARE TOOLS AND LIBRARIES

- IntelliJ IDEA for Java Development: <https://www.jetbrains.com/idea/>