PARCEL TRACKING SYSTEM

A PROJECT REPORT

Submitted by

SANTHIYA S (2303811724322096)

in partial fulfillment of requirements for the award of the course

CGB1201 – JAVA PROGRAMMING

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

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

SAMAYAPURAM – 621 112

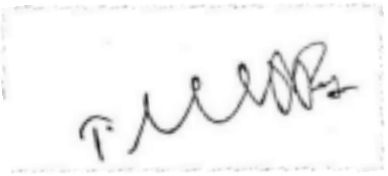
DECEMBER, 2024

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on **"PARCEL TRACKING SYSTEM"** is the bonafide work of **SANTHIYA S (2303811724322096)** who carried out the project work during the academic year 2024 - 2025 under my supervision.



Signature

Dr. T. AVUDAIAPPAN M.E.,Ph.D.,

HEAD OF THE DEPARTMENT,

Department of Artificial Intelligence,

K. Ramakrishnan College of Technology,

Samayapuram, Trichy -621 112.

Signature

Mrs. S. GEETHA M.E.,

SUPERVISOR,

Department of Artificial Intelligence,

K. Ramakrishnan College of Technology,

Samayapuram, Trichy -621 112.

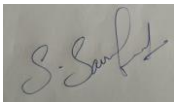
Submitted for the viva-voce examination held on 3.12.24

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

I declare that the project report on "**PARCEL TRACKING SYSTEM** " is the result of original work done by me and best of our knowledge, similar work has not been submitted to "**ANNA UNIVERSITY CHENNAI**" for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This project report is submitted on the partial fulfillment of the requirement of the award of the **CGB1201 – JAVA PROGRAMMING.**



Signature

SANTHIYA S

Place: Samayapuram

Date: 3/12/2024

ACKNOWLEDGEMENT

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

I extend our sincere acknowledgment and appreciation to the esteemed and honourable Chairman, **Dr.K.RAMAKRISHNAN, B.E.,** for having provided 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 our project and offering an adequate duration to complete it.

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

I thank **Dr.T.AVUDAIAPPAN, M.E.,Ph.D**., Head of the Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**, for providing her encouragement in pursuing this project.

I wish to convey our profound and heartfelt gratitude to our esteemed project guide **Mrs.S.GEETHA M.E**., Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**, for her incalculable suggestions, creativity, assistance and patience, which motivated us to carry out this project.

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

VISION AND MISSION OF THE DEPARTMENT

To excel in education, innovation and research in Artificial Intelligence and Data Science to fulfill industrial demands and societal expectations.

Mission 1: To educate future engineers with solid fundamentals, continually improving teaching methods using modern tools.

Mission 2: To collaborate with industry and offer top-notch facilities in a conductive learning environment.

Mission 3: To foster skilled engineers and ethical innovation in AI and Data Science for global recognition and impactful research.

Mission 4: To tackle the societal challenge of producing capable professionals by instilling employability skills and human values.

PROGRAM EDUCATIONAL OBJECTIVES (PEOS)

PEO 1: Compete on a global scale for a professional career in Artificial Intelligence and Data Science.

PEO 2: Provide industry-specific solutions for the society with effective communication and ethics.

PEO 3: Hone their professional skills through research and lifelong learning initiatives.

PROGRAM OUTCOMES

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.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- **PSO 1:** Capable of working on data-related methodologies and providing industry-focussed solutions.
- **PSO2:** Capable of analysing and providing a solution to a given real-world problem by designing an effective program.

ABSTRACT

The Parcel Tracking System is a software application designed to enable customers and logistics companies to monitor the real-time status and location of parcels from dispatch to delivery. By providing accurate updates at each stage of a parcel's journey, this system enhances user experience and ensures timely delivery. The system's functionality is aimed at streamlining operations, improving reliability, and enhancing transparency throughout the logistics process. With the increasing reliance on e-commerce and global trade, customers and businesses alike require a reliable way to monitor the status and location of parcels in real-time, ensuring that deliveries are made accurately and on schedule. This system is developed to empower users, including both customers and logistics companies, by providing up-to-date information at every stage of the parcel. The system's primary goal is to streamline logistics operations by offering reliable tracking information, thereby fostering trust and improving overall operational efficiency.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The Parcel Tracking System is a technological innovation designed to address the challenges of modern logistics and delivery management. With the exponential growth of e-commerce and global shipping, customers have higher expectations regarding the reliability and speed of parcel delivery services. The inability to monitor the location or status of parcels during transit can lead to dissatisfaction, inefficiencies, and operational bottlenecks. This system bridges the gap by providing a robust platform that allows users to access real-time updates on parcel movements, thus ensuring timely deliveries, reducing uncertainties, and enhancing trust between customers and logistics providers. By integrating advanced tracking features, the system facilitates seamless operations, enabling stakeholders to monitor progress and respond to any issues promptly.

1.2 OBJECTIVE

The objective of the Parcel Tracking System is to design and develop an efficient tool that meets the needs of both customers and logistics companies. The system is intended to provide real-time updates on the status and location of parcels, enabling users to track their journey from the point of dispatch to the final delivery. By ensuring that accurate information is available at each stage, the system aims to reduce delays, enhance customer satisfaction, and increase transparency in the delivery process. Additionally, the system seeks to improve operational efficiency by streamlining communication between customers and logistics providers, allowing for better planning and execution. Ultimately, the goal is to foster trust and reliability in parcel delivery services by delivering a

robust, user-friendly, and technologically advanced tracking system.

The primary objective of this project is to design and develop a Parcel Tracking System that empowers customers and logistics companies to:

- Monitor the real-time status and location of parcels.
- Receive accurate updates at each stage of the parcel's journey.
- Ensure timely delivery by allowing users to track progress.
- Improve operational efficiency and transparency for logistics companies**.**

CHAPTER 2
PROJECT METHODOLOGY

2.1 PROPOSED WORK

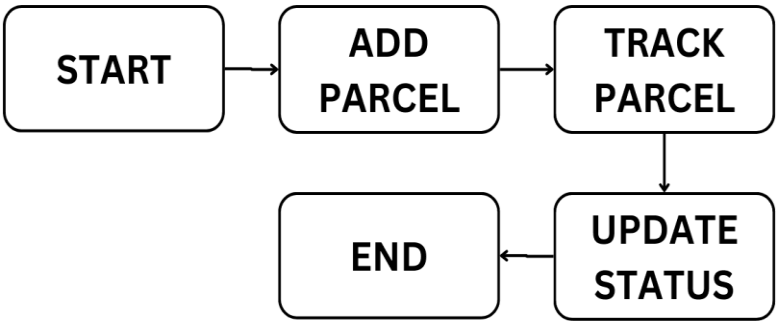
The Parcel Tracking System is a GUI-based application designed to simulate the process of parcel tracking within predefined locations in Tamil Nadu. This system aims to enhance the understanding of Java programming concepts, especially GUI development, threading, and real-time simulations.

Key Objectives:

1. User-Friendly Parcel Management:
    - Allow users to add parcels with a unique ID, sender, and receiver information.
    - Dynamically assign current and destination locations from predefined Tamil Nadu cities.
2. Real-Time Status Updates:
    - Simulate real-time tracking of parcels by transitioning through predefined statuses:
        - Dispatched
        - In Transit
        - Out for Delivery
        - Delivered
    - Display current status and location updates on the GUI.
3. Dynamic GUI Display:
    - Provide clear and interactive output to users via the AWT-based interface.
    - Include essential buttons, text fields, and a non-editable status area for a clean user experience.

4. Thread-Based Simulation:
    - Implement multithreading to handle background operations for status updates without freezing the user interface.
5. Randomized Parcel Routes:
    - Assign random, unique start and destination locations from a list of Tamil Nadu cities to emulate realistic scenarios.

2.2 BLOCK DIAGRAM



CHAPTER 3
JAVA PROGRAMMING CONCEPTS

3.1 OBJECT-ORIENTED PROGRAMMING (OOP):

- **Classes and Objects**: The system defines classes like Parcel and ParcelTrackingAWT to represent entities in the system. Objects are created for individual parcels, and their properties and behaviors are encapsulated in these classes.
- **Encapsulation**: The Parcel class encapsulates the properties of a parcel such as sender, receiver, status, and location. Methods like moveParcel() and is Delivered() encapsulate the logic of moving and checking the parcel status.
- **Inheritance**: The ParcelTrackingAWT class extends Frame, a built-in Java class for creating graphical user interfaces. This allows the program to inherit the behavior and appearance of a window-based application.

3.2. EVENT HANDLING:

- **Event Listeners**: The program listens for user actions (such as button clicks) using event listeners. The ActionListener interface is used for responding to events triggered by the "Add Parcel" and "Track Parcel" buttons.
- **GUI Interaction**: Components like buttons, text fields, and text areas are added to the GUI, and the program reacts to user input dynamically (adding parcels and tracking their status).

3.3 COLLECTIONS FRAMEWORKS:

- **HashMap**: The program uses a HashMap to store parcels using their unique Parcel ID as the key. This allows for efficient retrieval of parcel data based on the Parcel ID.
- **Random**: The Random class is used to simulate the selection of random locations for the parcel's current and destination addresses from a predefined list of Tamil Nadu cities.

3.4 GUI DEVELOPMENT (AWT):

- **AWT Components**: The program uses various **AWT components** such as Frame, Label, TextField, Button, and TextArea to create the user interface. These components are used to collect user inputs and display tracking information.
- **Layout Management**: The FlowLayout is used to arrange components in a simple, flowing design where components are displayed sequentially on the screen.

CHAPTER 4
MODULE DESCRIPTION

4.1 ADD PARCEL

- The Add Parcel module enables users to input a unique Parcel ID along with sender and receiver details. Once the details are provided, the system randomly assigns a starting location and a destination from predefined options representing Tamil Nadu cities. The module validates the uniqueness of the Parcel ID, ensuring no duplicate entries exist in the system. If a duplicate ID is entered, an appropriate error message is displayed; otherwise, the system confirms the successful addition of the parcel and displays the assigned locations.

4.2 TRACK PARCEL

- The Track Parcel module facilitates parcel tracking by accepting a Parcel ID and retrieving its details from the HashMap. Once the parcel is located, a separate thread is initiated to simulate real-time tracking. The module periodically updates the parcel's status as it transitions through different stages: "In Transit," "Out for Delivery," and finally, "Delivered." These updates occur at fixed intervals, mimicking the real-life process of a parcel's journey.

4.3 UPDATE STATUS

- The Update Status module dynamically refreshes the status area within the GUI to reflect the current status and location of the parcel. As the parcel progresses, the module ensures smooth and clear transitions between states, using fixed delays to simulate the passage of time. This interactive feature keeps users informed in real-time, enhancing the overall user experience and providing clear visibility into the parcel's delivery process.

CHAPTER 5
CONCLUSION

In conclusion, the Parcel Tracking System demonstrates an efficient and interactive approach to managing parcel deliveries using Java and AWT. The system enables users to add parcels, assign unique IDs, and track their status in real time through a user-friendly graphical interface. By leveraging multithreading, dynamic status updates, and predefined location data, the application simulates the real-world parcel delivery process effectively.

This project highlights key Java programming concepts such as object-oriented programming, event-driven programming, and concurrent execution through threads. It provides a practical understanding of how these concepts can be applied to solve real-life problems, ensuring smooth and accurate parcel tracking.

The system's modular structure ensures scalability and clarity, making it easy to maintain and enhance for future needs. By automating parcel status updates and providing real-time insights into the delivery process, this project adds value to logistics management and enhances user satisfaction.

REFERENCES:

- **Java AWT (Abstract Window Toolkit)** https://www.tutorialspoint.com/awt/index.htm
  - A comprehensive guide to the Abstract Window Toolkit (AWT), which forms the basis for GUI applications in Java.
  - Includes examples for creating windows, handling events, and adding components like buttons, text fields, and text areas.
- **Multithreading in Java** https://www.javatpoint.com/event-handling
  - Explains the concept of multithreading in Java for concurrent execution of tasks.
  - Demonstrates how threads can be used to simulate real-time processes like updating parcel statuses in this project.
- **Java HashMap** https://www.baeldung.com/java-hashmap
  - Offers detailed insights into the HashMap class, which is used for storing and retrieving key-value pairs.
  - Discusses operations such as adding, retrieving, updating, and deleting elements efficiently.
  - Provides examples of how HashMap is used to manage parcel data in the system.

APPENDICES

APPENDIX A – SOURCE CODE

```
import java.awt.*;
import java.awt.event.*;
import java.util.HashMap;
import java.util.Map;
import java.util.Random;
public class ParcelTrackingAWT extends Frame {
    static Map<String, Parcel> parcels = new HashMap<>();
    static String[] locations = {
        "Chennai", "Coimbatore", "Madurai", "Trichy", "Salem", "Erode",
        "Tirunelveli", "Vellore", "Thoothukudi", "Dindigul"
    };
    TextField parcelIdField, senderField, receiverField;
    TextArea statusArea;
    Button addParcelButton, trackParcelButton;
    static class Parcel {
        String sender;
        String receiver;
        String status;
        String currentLocation;
        String receiverLocation;
        Parcel(String sender, String receiver, String currentLocation,
         String   receiverLocation) {
            this.sender = sender;
            this.receiver = receiver;
            this.status = "Dispatched";
            this.currentLocation = currentLocation;
```

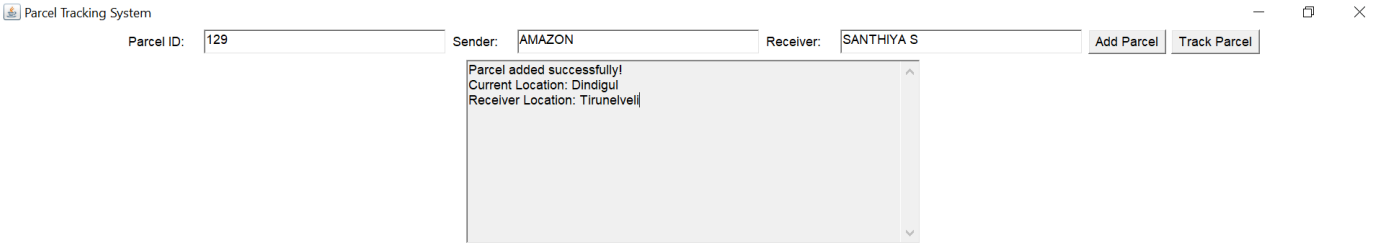
```
        this.receiverLocation = receiverLocation;
    }
}
public ParcelTrackingAWT() {
    setTitle("Parcel Tracking System");
    setSize(500, 400);
    setLayout(new FlowLayout());
    add(new Label("Parcel ID:"));
    parcelIdField = new TextField(20);
    add(parcelIdField);
    add(new Label("Sender:"));
    senderField = new TextField(20);
    add(senderField);
    add(new Label("Receiver:"));
    receiverField = new TextField(20);
    add(receiverField);
    addParcelButton = new Button("Add Parcel");
    trackParcelButton = new Button("Track Parcel");
    add(addParcelButton);
    add(trackParcelButton);
    statusArea = new TextArea(10, 40);
    statusArea.setEditable(false);
    add(statusArea);
    addParcelButton.addActionListener(e -> addParcel());
    trackParcelButton.addActionListener(e -> trackParcel());
    addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent e) {
            System.exit(0);
        }
```

```
        });
    }
    void addParcel() {
        String parcelId = parcelIdField.getText();
        if (parcels.containsKey(parcelId)) {
            statusArea.setText("Parcel ID already exists!");
            return;
        }
        String sender = senderField.getText();
        String receiver = receiverField.getText();
        Random random = new Random();
        String currentLocation = locations[random.nextInt(locations.length)];
        String receiverLocation = locations[random.nextInt(locations.length)];
        while (currentLocation.equals(receiverLocation)) {
            receiverLocation = locations[random.nextInt(locations.length)];
        }
        Parcel parcel = new Parcel(sender, receiver, currentLocation,
receiverLocation);
        parcels.put(parcelId, parcel);

        statusArea.setText("Parcel added successfully!\n" +
                           "Current Location: " + currentLocation + "\n" +
                           "Receiver Location: " + receiverLocation);
    }
    void trackParcel() {
        String parcelId = parcelIdField.getText();
        if (!parcels.containsKey(parcelId)) {
            statusArea.setText("Parcel ID not found!");
            return;
```

```
        }
        Parcel parcel = parcels.get(parcelId);
        statusArea.setText("Tracking Parcel: " + parcelId);
        new Thread(() -> {
            try {
                parcel.status = "In Transit";
                updateStatus(parcelId, parcel);

                Thread.sleep(60000);
                parcel.status = "Out for Delivery";
                updateStatus(parcelId, parcel);
                Thread.sleep(60000);
                parcel.status = "Delivered";
                parcel.currentLocation = parcel.receiverLocation;
                updateStatus(parcelId, parcel);
            } catch (InterruptedException ex) {
                ex.printStackTrace();
            }
        }).start();
    }
    void updateStatus(String parcelId, Parcel parcel) {
        statusArea.setText("Tracking Parcel: " + parcelId + "\n" +
                           "Status: " + parcel.status + "\n" +
                           "Current Location: " + parcel.currentLocation + "\n" +
                           "Receiver Location: " + parcel.receiverLocation);}
    public static void main(String[] args) {
        new ParcelTrackingAWT().setVisible(true);
    }
}
```

APPENDIX B – SCREENSHOTS

1)ADD PARCEL



2)TRACK PARCEL