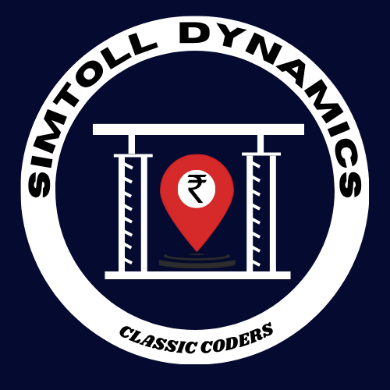


|  |
| --- |
| GPS TOLL-BASED SYSTEM SIMULATION  2024 |
|  |
| July 15      Authored by: CLASSIC CODERS |



|  |  |  |
| --- | --- | --- |
| S.NO. | TITLE | PAGE NO. |
| 01. | Introduction | 03. |
| 02. | Methodology | 06. |
| 03. | Implementation | 11. |
| 04. | Results | 15. |
| 05. | Execution and Timeline | 19. |
| 06. | Conclusion | 21. |

# INDEX

# INTRODUCTION

|  |
| --- |
| Subtitle Text Here *“The road to efficiency is paved with technology”*  In today's rapidly evolving world, technology plays a pivotal role in enhancing the efficiency and effectiveness of various systems. One of the critical areas where technology has significantly impacted is transportation, particularly in the area of toll collection. Traditional toll collection methods, often plagued by congestion and delays, are gradually being replaced by innovative, technology-driven solutions. The current electronic toll collection systems like fast tags still have limitations in terms of scalability, dynamic pricing, and integration with various payment methods. Moreover, these systems cannot often provide real-time data and analytics that could enhance traffic management and decision-making.  This project focuses on developing a GPS-based toll system simulation using Python. The primary aim is to create an efficient and accurate system that can streamline toll collection, reduce congestion, and improve the overall driving experience.  The report provides a comprehensive overview of the project, detailing the objectives, methodologies, implementation, and results. It aims to highlight the importance of integrating modern technology into transportation systems to achieve higher efficiency and reliability. |

# PROBLEM STATEMENT:

Develop a Python-based simulation for a GPS toll collection system, incorporating vehicle movement along predefined routes, toll zone detection, distance calculation within toll zones, and payment processing.

# PROPOSED SOLUTION:

# GPS toll-based calculation uses GPS technology to track a vehicle’s location and calculate tolls in real-time (Mobile phones/Electronic control unit), this gives an advantage over RFID chips and allows us to debit the amount dynamically.

# Live GPS tracking

# Toll route detection and calculation

# Payment processing

# Report generation: User side (Receipt generation) and Admin side (Analytics)

# IMPACT:

# The simulation helps us develop a system of automatic and dynamic methods to collect tolls offering more efficiency and less over-crowding. Some common benefits of GPS toll calculation are:

# Real-time accuracy

# Convenience and reduced congestion

# Increased transparency and improved customer experience

# 

# PROJECT METRICS:

# The following are the metrics that will help assess the performance, accuracy, and effectiveness of the system:

# ACCURACY METRICS: GPS accuracy, simulation time, distance calculation accuracy and Toll calculation accuracy.

# USER EXPERIENCE METRICS: User account balance accuracy, Payment success rate, and Receipt generation accuracy.

# FINANCIAL METRICS: Total toll revenue, revenue per zone, and Average toll per vehicle.

# ANALYTICAL METRICS: Vehicle movement report and Toll collection reports

# SECURITY METRICS: Data breach incidents and login security strength.

# LIMITATIONS:

# The limitations of this project are:

1. ****Simulation Accuracy****: While efforts will be made to simulate realistic vehicle movements and toll zone crossings, the accuracy may be influenced by the quality of input data and the simulation model.

2. ****GPS Signal Variability****: Factors such as weather conditions, terrain, and urban environments may affect the accuracy of GPS signals used in the simulation.

3. ****Simplified Payment Simulation****: The payment simulation will be a simplified model and may not cover all complexities of real-world payment processing systems.

4. ****Integration with Existing Systems****: The project will focus on the simulation itself and may not cover integration with existing fleet management or accounting systems.

6. ****Security Measures****: While basic security measures will be implemented, the project may not cover advanced security features required for deployment in a real-world scenario.

# METHODOLOGY

# The methodology section provides a detailed account of the processes and tools used to achieve the project objectives. By following a systematic approach, we ensured the accuracy and efficiency of the GPS Toll-based System Simulation. The methodology is divided into several key components, each addressing a specific aspect of the project. From setting up the simulation environment and initializing vehicles to detecting toll zone crossings and processing payments, every step is meticulously documented to provide a clear understanding of the project's workflow.

# COMPONENTS:

# The following are the main components of this project:

# *VEHICLE MOVEMENT SIMULATION:* The main simulation is run by using Simpy, a process-based discrete-event simulation framework based on Python, which was used to model the movement of vehicles along pre-defined routes with coordinate

# Process:

* *Initialization:* Each vehicle is initialized with attributes such as vehicle type, starting location, destination, and pre-defined route.
* *Simulation Environment:* A simulation environment is created using SimPy, where the movement of each vehicle is defined as a process.
* *Movement Logic:* Vehicles move along their routes by updating their coordinates at each time step.
* *Time Management:* SimPy’s environment.run() function manages the simulation time, allowing for the scheduling of events and updating vehicle positions at each time step.
* *Logging:* The movement data of each vehicle, including coordinates, distance traveled and tolls crossed, is logged for analysis and visualization.

# *TOLL ZONE DEFINITION:* Defining toll zones accurately is essential for detecting when a vehicle crosses into a toll zone. Geospatial libraries were used to define toll zones using coordinates.

Process:

* *Importing Shapefiles:* The road network and toll zones are imported as shapefiles using GeoPandas.
* *Defining Toll Zones:* Toll zones are defined as polygons with specific GPS coordinates. Each toll zone is stored as a geometric object in Shapely.
* *Spatial Reference:* All geospatial data is converted to a common spatial reference system to ensure accuracy.
* *Storage:* The toll zone data is stored in a structured format, allowing for easy access and reference during the simulation.

# *DISTANCE CALCULATION:* Calculating the distance traveled by vehicles within toll zones is necessary for computing toll charges. The GeoPy library was used for accurate distance calculations between GPS coordinates.

**Process:**

* *Entry and Exit Points:* The entry and exit points of each vehicle within a toll zone are recorded using their GPS coordinates (either via GPS App or manual entry).
* *Haversine Formula:* The Haversine formula, implemented in GeoPy, is used to calculate the great-circle distance between two points on the Earth’s surface.
* *Distance Summation:* The total distance traveled by each vehicle within a toll zone is calculated by summing the distances between consecutive GPS coordinates logged during the vehicle’s movement

# 

# *TOLL CALCULATION:* Toll charges are computed based on the distance traveled by vehicles within toll zones or the number of zones passed.

*Process:*

* *Predefined Rates*: Toll rates are predefined based on distance traveled within a zone or as fixed fees for zone crossings.
* *Distance-Based Calculation:* For distance-based tolls, the total distance traveled within each toll zone is multiplied by the rate per unit distance.
* *Zone-Based Calculation:* For zone-based tolls, a fixed fee is charged for each zone crossed by the vehicle.
* *Congestion-based calculation:* Depending on a certain vehicle threshold, the rates of the toll changes dynamically offering vehicles a 5% discount if more than the threshold the vehicles are present in the toll zone.
* *Computation:* The toll charges for each vehicle are computed in real-time as they cross toll zones, and the results are logged for payment processing.

# *PAYMENT SIMULATION:* Simulating toll payments involves deducting toll charges from user accounts and managing transaction records. The Pandas library is used to handle data related to toll transactions and user accounts.

Process:

* *Account Management:* User accounts are maintained with initial balances. Each account corresponds to a unique vehicle ID.
* *Transaction Logging:* As vehicles cross toll zones, the computed toll charges are deducted from the respective user accounts.
* *Balance Updates:* The account balances are updated in real-time, and transaction details are logged, including vehicle ID, toll amount, and remaining balance.
* *Error Handling:* Mechanisms are implemented to handle scenarios such as insufficient balance and transaction failures.
* *Reporting:* Detailed records of all transactions are maintained, which are used to generate analytical reports on toll collections and system performance. These reports are sent to the user in the mail using google sheets , and we have used GScript for mail sending process. The report will contain distance travelled by the vehicle , all the vehicle information , toll amount , remaining balance , and a folium map which has the path travelled by that user

# *LIVE GPS UPDATES APP (ADDITIONAL FEATURE):*

# *Tracking the live GPS coordinates of your phone using an app , and plotting your coordinates in a map and deducting toll charges based on your travelled path*

Process:

* *Fetching GPS Coordinates:* We have built an custom app using android studio which connects to the server and send your live precise GPS coordinates to the server every 1.5 seconds. This GPS coordinates will be continuously sent to the server as a foreground service , hence even if you close the app the GPS coordinates sending process will continue
* *Server’s work:* As the GPS coordinates are received by the server , the server will plot it in a map for visualisation of the vehicle , then using the GPS coordinates it will automatically detect the toll zones and deduct toll amounts accordingly.
* *Balance Updates:* The account balances are updated in real-time, and transaction details are logged, including vehicle ID, toll amount, and remaining balance.
* *Error Handling:* Mechanisms are implemented to handle scenarios such as insufficient balance and transaction failures.
* *Reporting:* Detailed records of all transactions are maintained, which are used to generate analytical reports on toll collections and system performance. These reports are sent to the user in the mail using google sheets , and GScript for mail sending process.

# LIBRARIES AND FRAMEWORKS:

# The success of the GPS Toll-based System Simulation project heavily relied on a variety of libraries and frameworks that provided essential functionalities for different aspects of the simulation. Each library was chosen based on its strengths and suitability for specific tasks, ensuring efficient and accurate implementation of the project components.

# Simpy: Used in Event-driven simulation, Time management, and Process management.

# GeoPandas and Shapely: Used to handle Geospatial data, spatial analysis, and the creation of toll zones.

# GeoPy: Used in the implementation of distance calculation and geocoding

# Pandas: Used in data management, analysis and real-time updates

# Folium and matplotlib: Used in mapping, plotting, and reporting aspects

# Plotly: Used to create a live visualization of the simulation while it is executing.

# PROCESS FLOW:

# 

# IMPLEMENTATION

# The implementation phase of the GPS Toll-based System Simulation project involves translating the design into functional code, developing algorithms to handle various simulation tasks, overcoming challenges, and adding additional features to enhance the simulation.

# This section provides a comprehensive overview of the code structure, key functions and algorithms, and the features implemented.

# CODE STRUCTURE:

# The code for this project is organized into several modules, each responsible for different aspects of the simulation. This modular approach ensures clarity, maintainability, and scalability of the codebase.

# MAIN MODULES:

# *****Simulation Module******:* Handles the initialization and execution of the simulation environment.

*****Vehicle Module******:* Manages vehicle attributes, movement logic, and GPS updates.

*****Geospatial Module******:* Defines toll zones and road network.

*****Distance Calculation Module******:* Computes distances traveled by vehicles within toll zones.

*****Toll Calculation Module******:* Calculates toll charges based on predefined rates and distance traveled along with dynamic pricing.

*****Payment Module******:* Simulates toll payments and updates user account balances.

*UI Module:* The interface from the admin side that can be used to decide how many vehicles can be used for simulation and retrieve reports for each.

*GPS App (Additional feature):* A real-life implementation of the toll zone calculation that can be used to track live GPS coordinates through the road network.

# *****Visualization Module******:* Generates maps and plots for visualizing the simulation

# KEY FUNCTIONS AND ALGORITHMS:

# 

# 

# 

# EXTRA FEATURES ADDED:

# To enhance the simulation, several additional features were implemented:

# *GPS App:* Implemented an app that can send live GPS coordinates directly to SimToll Dynamics to execute live simulation.

# *Dynamic pricing:* Implemented congestion-based, toll-specific and vehicle-specific pricing models.

# *Interactive UI*: Our simulation runs in an interactive website/Tkinter that allows the admin to enter the vehicle data and conditions and observe the respective reports.

# *Realistic road network:* We have implemented a realistic road network of the state and national highways of Chennai to allow our simulation to run in an organic scenario*.*

# *User-side report emails:* Our user-side reports are sent directly to their emails with an interactive folium map of their vehicle traversal.

# *****Emergency Contingencies******:* Added logic to handle emergency situations, such as when the account balance is low during payment transactions.

# RESULTS

# The GPS Toll-based System Simulation project successfully achieved its objectives, demonstrating the feasibility and efficiency of using GPS technology for toll calculation and payment processing. This section presents the key outcomes, visual outputs, and potential future advancements.

# KEY OUTCOMES:

1. *****Accurate Vehicle Movement Simulation******:* The simulation accurately modeled vehicle movements along predefined routes, updating coordinates in real time. It also ran the simulation using Live GPS accurate coordinates using the App.
2. *****Efficient Toll Zone Detection******:* The system reliably detected toll zone crossings and calculated the distance traveled within these zones.
3. *****Precise Toll Calculation******:* Toll charges were computed accurately based on distance traveled or zones crossed, with dynamic pricing models effectively implemented.
4. *****Seamless Payment Processing******:* The simulation successfully processed toll payments, deducting charges from user accounts and updating transaction records.
5. *****Informative Visualizations******:* The project provided clear and intuitive visualizations of vehicle movements, toll zone locations, and toll charge distributions.

# VISUAL OUTPUTS:

# 

# Login Page Admin Reports

# 

# Main Window- Adding vehicle data

# 

# User Side Payment Email

# 

# User Side Folium visualization

# 



# GPS Toll Application: Detailed Functionality and Operational Procedure

The GPS\_Toll application is designed to automate the toll collection process by using live GPS coordinates transmitted from your mobile phone. The application communicates with a local server to send and receive necessary data, ensuring that the toll amount is dynamically calculated and debited from the user's account based on the vehicle's movement. This document provides a comprehensive overview of the application's functionality, from initial setup to continuous operation.

Application Setup and Initial Operation

1. Launching the Application

- Step 1: Open the GPS\_Toll application on your device.

- Step 2: Enter your vehicle number in the designated input field.

2. Finding the Local Server IP Address

- Step 3: Press the "Get IP Address" button.

- Mechanism: The app scans the local network by sending requests to all possible 254 subnet IP addresses. This scan helps identify the local server's IP address.

- \*\*Technical Details\*\*: The app sends HTTP requests to each IP address in the subnet (typically 192.168.0.1 to 192.168.0.254). The server responds to the correct request, revealing its IP address.

Continuous GPS Coordinate Transmission

3. Starting the Foreground Service

- Step 4: After identifying the server IP address, the app begins transmitting live GPS coordinates.

- Mechanism: A foreground service is initiated to send GPS coordinates every 1.5 seconds.

- Technical Details:

- The app leverages Android's `LocationManager` to obtain precise GPS coordinates.

- A `Service` runs in the foreground, ensuring continuous operation even when the app is closed.

- The coordinates are sent as HTTP POST requests to the server's IP address at regular intervals (1.5 seconds).

4. Stopping the Service

- Step 5: To stop the coordinate transmission, open the app and press the "Stop" button.

- Mechanism: This action stops the foreground service, halting the transmission of GPS data.

Server-Side Operations

5. Setting Up the Local Server

- Step 6: Start the server developed using Python on the local network before initiating the app scan.

- Mechanism:

- The server listens for incoming HTTP POST requests containing GPS coordinates.

- The server processes these coordinates and uses them for real-time tracking and toll calculation.

6. Real-Time Tracking and Toll Calculation

- Step 7: The server plots the vehicle's coordinates on a map using Matplotlib.

- Mechanism:

- As the vehicle moves, the marker on the map updates dynamically based on the received coordinates.

- The server calculates toll amounts dynamically based on predefined toll zones and vehicle movement patterns.

- The calculated toll amount is automatically debited from the user's account.

The GPS\_Toll application streamlines toll collection by leveraging real-time GPS data and local network communication. By ensuring continuous GPS data transmission through a foreground service, the application remains functional even when closed. The server-side operations, including real-time tracking and dynamic toll calculation, provide a seamless user experience.

# FURTHER ADVANCEMENTS:

To further enhance the GPS Toll-based System Simulation project, the following advancements can be considered:

1. *Enhanced User Interface*: Developing a more user-friendly interface for better interaction with the simulation and visualization tools.
2. *Scalability Improvements:* Optimizing the system to handle larger volumes of vehicles and transactions more efficiently.
3. *Advanced Analytics:* Implementing advanced analytics to provide deeper insights into toll system performance and user behavior.
4. *Mobile App Development:* Creating a mobile app to allow users to query toll information, make payments, and receive real-time updates on their journeys.
5. *Security Enhancements:* Strengthening security measures to protect user data and ensure the privacy and integrity of transactions.
6. *Integration with Existing Toll Systems:* Exploring ways to integrate the simulation with existing toll collection systems for real-world application and testing.

# EXECUTION AND TIMELINE

# The GPS Toll-based System Simulation project was executed in several phases, with clearly defined tasks and roles assigned to each team member. This section outlines the timeline of the project and details the contributions of each team member.

# PROJECT TIMELINE:

Phase 1: *Planning and Research*

* Objective: Define the project scope, objectives, and methodologies.
* Tasks:
  + Conduct a literature review and research on GPS-based toll systems.
  + Define project goals and objectives.
  + Outline the project methodology and identify required tools and technologies.

Phase 2: *Design and Setup*

* Objective: Design the system architecture and set up the development environment.
* Tasks:
  + Design the overall system architecture and data flow.
  + Set up the development environment and install necessary libraries and frameworks.
  + Define toll zones and routes using geospatial data.

Phase 3: *Development*

* Objective: Develop the core components of the simulation
* Tasks:
  + Implement vehicle movement simulation using SimPy.
  + Develop toll zone detection and distance calculation algorithms.
  + Create toll calculation and payment simulation modules.
  + Integrate all components and ensure seamless operation.

Phase 4: *Visualization and Reporting*

* Objective: Develop visualization tools and prepare the final project report.
* Tasks:
  + Implement visualization tools using Matplotlib and Folium.
  + Generate visual outputs such as maps and plots.
  + Compile the final project report, including methodology, results, and future work.

# TEAM ROLES AND RESPONSIBILITIES:

# The project was completed by a team of five members, each contributing their expertise to different aspects of the project:

1. ASWIN (*Project manager and app developer*): Coordinated the project and developed the GPS app which allows live location tracking.
2. ATHIKA *(Data analyst):* Compiled the reports from each simulation and allowed for both user and admin side report generation.
3. DHIKSHA *(Geospatial analyst):* Collected road networks and toll zones required and handled the payment calculation of the project
4. HARSHA VARDHAN *(UI and visualization specialist):* Created the website interface to run the simulation and allowed for live visualization of the simulation.
5. TEJASWINI *(Simulation engineer):* Implemented the simulation of the project and developed the toll calculation logic

# CONCLUSION

# The GPS Toll-based System Simulation project successfully demonstrated the potential of integrating GPS technology with toll collection systems to create a more efficient and user-friendly tolling process. By leveraging advanced simulation frameworks, geospatial analysis tools, and real-time data processing, the project achieved its objectives and provided valuable insights into the feasibility and benefits of GPS-based tolling.

# FINAL REMARKS:

The GPS Toll-based System Simulation project has laid a strong foundation for exploring the use of GPS technology in toll collection systems. It has showcased the potential benefits of such systems, including real-time accuracy, convenience, reduced congestion, increased transparency, and improved customer experience. By addressing the identified challenges and exploring future advancements, this project can contribute to the development of more efficient, secure, and user-friendly tolling systems.

In conclusion, the project not only met its initial objectives but also opened avenues for future research and development in the field of intelligent transportation systems. The insights gained from this project will be instrumental in guiding future efforts to create smarter, more efficient toll-collection solutions.

# THANK YOU