Automated Toll Payment and Vehicle Tracking System

Prof. Kiran Bidua Pandey[#1], Rahul Nanwani[*2], Swapnil Nimbalkar[*3], Tanmay Somkuwar[*4], and Vijay Gupta[*5].

Abstract- **In our daily lives, the cars are in huge numbers, the tollbooth is seemingly the bottleneck through the gates, and the travellers pay the government with a guaranteed amount of tax. Toll gates are located on national highways and bridges, where people pay tolls for using the highways by standing in a long line, wasting time and fuel.**

To address this issue, this proposed system approach for automating toll payments without the use of toll plazas is based solely on the vehicle's distance travelled on a given road using GPS technology. The proposed system will be designed using the Android Development Kit, Open Street Maps, and the Python programming language.

In the proposed work, the user can log into the system, and the software will constantly update the location of the vehicle until it is on the highway.

Keywords- Toll Payment, Vehicle Tracking, Automation.

I. Introduction

Toll booths have traditionally been used to collect revenue on highways, bridges, and tunnels. Toll revenue is typically used to maintain highways and recover construction costs. The revenue generated by bridges and tunnels only covers a small portion of the construction costs because it is used to finance some necessary maintenance and operation first. Currently, the revenue collection system at most toll payment booths requires the driver to stop the vehicle and pay with cash or a credit card to continue their journey. Toll payment booths are widely used, but they have several drawbacks, including high fuel consumption, vehicle congestion, and a time-consuming payment method. This method also necessitates labour, resulting in the waste of manpower. From the standpoint of both the user and the toll agency, automated toll payment systems outperform the current method. Even though this process's technological evolution is slow, it is incremental.

II. Motivation Behind the Project

Drivers of vehicles do not need to stop at a window or waste time waiting in a long line to pay their tolls when using an automated toll payment system. This will reduce fuel consumption, reduce congestion, and increase road safety, and they will only have to pay tax for the length of the road they use.

III. Literature Survey

"Integrated Networking, Caching, and Computing for Connected Vehicles: A Deep Reinforcement Learning Approach," by Ying He, Nan Zhao, and Hongxi Yin, was published in IEEE Transactions on Vehicular Technology, vol. 67, 2018. They proposed in this paper an integrated framework for dynamic orchestration of networking, caching, and computing resources to improve the performance of next-generation vehicular networks. They developed the resource allocation strategy in this framework as a joint optimization problem, considering the benefits of not only networking but also caching and computing in the proposed framework.

"Enhancing GPS with Lane level Navigation to Facilitate Highway Driving," by Tianyi Song, Nicholas Capurso, Xiuzhen Cheng, Jiguo Yu, Biao Chen, and Wei Zhao, challenged IEEE Transactions on Vehicular Technology, 2017 by proposing a GPS aiding system that can sense and track a vehicle's lane position. The system makes use of smart phones' computing power, rear cameras, and inertial motion sensors. The system employs computer vision techniques to achieve lane level positioning with minimal additional computational overhead. They also created a lane switching detection and tracking algorithm based on machine learning.

"Smartphone-based Real Time Vehicle Tracking in Indoor Parking Structures," by Ruipeng Gao, Mingmin Zhao, Tao Ye, Fan Ye, Yizhou Wang, and Guojie Luo, was published in IEEE Transactions on Mobile Computing in 2017. According to the paper, they created algorithms in a Sequential Monte Carlo framework to represent vehicle states probabilistically, and they used constraints provided by the garage map and detected landmarks to robustly infer vehicle location. They also proposed landmark (e.g., speed bumps, turns) recognition methods that are resistant to noise, bumpy ride disturbances, and even handheld movements.

"Design and Implementation of Vehicle Tracking System Using GPS/GSM/GPRS Technology and

Smartphone Application," SeokJu Lee, Girma Tewolde, and Jaerock Kwon, published in IEEE World Forum on Internet of Things (WFIoT), 2014. According to this paper, the system that they developed was able to experimentally demonstrate its efficiency in tracking a vehicle's location at any time and from any location. They claimed that this implementation was low-cost because it was based on readily available off-the-shelf electronic modules.

"Beginning Google Maps API 3," Gabriel Svennerberg, Appress Publication, 1st Edition, this book provides the reader with the skills and knowledge required to incorporate Google Maps version 3 on web pages in both desktop and mobile applications.

Lan Lake and Reto Meier's "Professional Android", 4th Edition, Wrax Publication, is a comprehensive developer guide to the most recent Android features and capabilities. Professional Android also teaches developers how to use the most recent Android features to create robust and compelling mobile apps that integrate with Google APIs such as Google Maps.

“Android Development with Kotlin,” by Marcin Moskala and Igor Wojda, Packt Publication, 1st Edition, this book provides methodology to make Android development much faster using a variety of Kotlin features, from basic to advanced, to write better quality code. This book uses specific Kotlin features to make Android application development easier.

IV. FUNCTIONAL REQUIREMENTS

A. *User Interface*

- Home Page
- Login Panel.
- Registration Panel
- Admin Panel.
- User Panel.
- Map Interface.
- Billing Interface.
- Enforcement Panel.

B. *Hardware Interfaces*

The entire software requires a fully equipped Android system with GPS and Internet access.

C. *Software Interfaces*

Android can be used as the system's operating system platform. The system also makes use of graphical user interface (GUI) tools. SQLite as a database and Apache Tomcat as a server are required to run this application. The Open Street Maps API is used in the application to implement map-related functions. MySQL database is required to store data on the server.

D. *Communication Interfaces*

- Communication to the server using Kotlin and servlet APIs through Internet.
- User interface - the display, touchscreen, menus of the system.
- User interface - the display, touchscreen, menus of the system.

V. NON-FUNCTIONAL REQUIREMENTS

A. *Performance Requirements*

The system's performance is determined by how it is handled. Every user must be properly instructed on how to use the system. The absence of any of the suggested requirements is another factor that affects performance.

B. *Safety Requirements*

To ensure the system's safety, perform regular system monitoring to ensure the system's proper operation. An authenticated user can only access the system.

C. *Security Requirements*

Any unauthorised user should be barred from gaining access to the system. Password authentication is an option that can be implemented.

D. *Software Quality Attributes*

- Accuracy
  The proposed system will have a higher level of accuracy. All operations would be carried out correctly, and it would ensure that all information coming from the centre is correct. As a result, organic results are obtained.
- Reliability
  Because of the factors mentioned, the proposed system will be highly reliable. The reason for the system's increased reliability is that their would-be proper information storage and tweet analysis model is now in place.

VI. SYSTEM REQUIREMENTS

A. *Database Requirements*

- MySQL
- SQLite

B. *Software Requirements (Platform)*

- Operating System – Android 5.1 and above
- Application Server – Apache Tomcat
- Front End – Kotlin, XML
- Database – MySQL 5.0, SQLite
- IDE – Android Studio
- Services – Open Street Maps

C. *Hardware Requirements*

- Processor – Any with clock Speed - 2.1 GHz
- RAM – 1 GB (min)
- Internal – Storage - 8 GB (min)

- Display – 5 inches and above with touchscreen

VII. METHODOLOGY

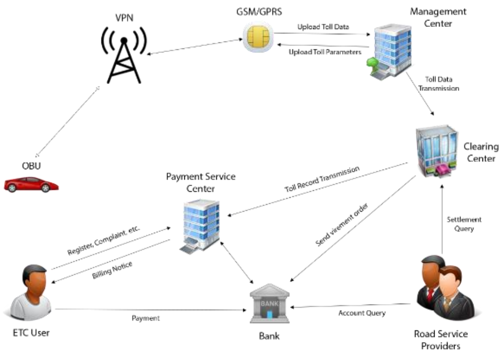

Fig. 1. System Architecture

A. *OBU (On Board Unit)*

A motherboard, GPS module, and GSM module are included. It will be mounted on the user's vehicle to track it.

B. *Connectivity Framework*

It will be used to establish communication between the user vehicle and the management centre. The data collected from the user vehicle OBU will be shared with the Management Centre via the Connectivity Framework.

C. *Management Center*

It will calculate the toll based on the vehicle profile and data received, and it will keep track of the vehicle's mileage and paid receipts.

D. *Clearing Center*

Data collected from the management centre will be cleaned and analysed to gain a better understanding of track patterns and the need to implement good track control strategies.

E. *Road Service Providers*

Highway authorities and RTO authorities will share information about toll charges and vehicle registration documents to provide road service.

VIII. MODULES

A. *Login*

This module will authenticate the user and grant him access to and monitoring of all vehicle profiles registered in his name. The server-side database will be used to validate the user's username and password.

B. *Register Vehicle*

The module will validate and store vehicle information in the database under the username. If a user attempts to register the same vehicle twice, the module will return the previously registered vehicle ID. This module can also deregister and block the vehicle.

C. *Get Location*

The module will validate the vehicle ID and collect GPS data from the GPS module. It will send the GPS data to the server after formatting it in longitude and latitude format.

D. *Highway Check*

The module will collect GPS data and send it to the Roads API. The Roads API will return a universal highway ID to the Highway Check module, who will then return it to the server.

E. *Trace Route*

This Module will track GPS data and trace the route of the vehicle on the highway. It will calculate the distance travelled on the highway and send the total kilometres to the server.

F. *Payment*

The module will validate the vehicle ID, the number of kilometres travelled, and the payment processing. The payment status and Bill ID will be sent to the user and updated in the database.

IX. RESULTS

Vehicle tracking will begin as soon as the vehicle begins to move. GPS coordinates will be collected and stored in a central database using the GPS in the vehicle. As soon as coordinates are received, they will be saved to the database along with their current timestamp. Coordinates saved in the database will then be processed by a service running on the server using an open street routing machine. The coordinates will be divided into sets. Each set was then mapped to the map's nearest route. Following the mapping of sets of coordinates that belong to the same highway/road, they will be combined to create a route travelled by vehicle, and the distance travelled will be calculated and stored in a database.

X. CONCLUSION

The automated toll payment and vehicle tracking system provides real-time data that allows the user to track the vehicle, allows road users to pay the toll without stopping or slowing down, and allows for early retrieval if the car is stolen. Implementing a GPS tracker in a vehicle can undoubtedly bring about revolutionary change in a developing country like India, where there is a high rate of urban and rural vehicular transition daily. As a result, traffic congestion will be reduced, and fuel efficiency will improve. The use of this system ensures a fair toll collection policy.

REFERENCES

[I] Ying He, Nan Zhao, and Hongxi Yin," *Integrated Networking, Caching, and Computing for Connected*

Vehicles: A Deep Reinforcement Learning Approach", IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 67, 2018.

[II] Tianyi Song, Nicholas Capurso, Xiuzhen Cheng, Jiguo Yu, Biao Chen, and Wei Zhao, *Enhancing GPS with Lane-level Navigation to Facilitate Highway Driving*, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, 2017.

[III] Ruipeng Gao, Mingmin Zhao, Tao Ye, Fan Ye, Yizhou Wang, and Guojie Luo, *Smartphone-based Real Time Vehicle Tracking in Indoor Parking Structures*, IEEE Transactions on Mobile Computing,2017.

[IV] SeokJu Lee, Girma Tewolde, Jaerock Kwon, *Design and Implementation of Vehicle Tracking System Using GPS/GSM/GPRS Technology and Smartphone Application*, IEEE World Forum on Internet of Things (WF-IoT), 2014 Gabriel Svennerberg," Beginning Google Maps API 3", Appress.

[V] Lan Lake and Reto Meier, *Professional Android*, John Wiley and Sons Publication, 4th Edition, 2018.

[VI] Marcin Moskala and Igor Wojda, *Android Development with Kotlin*, Packet Publication, 1st Edition, 2017.

[VII] Marcin Moskala and IgorWojda," Android Development with Kotlin" Packet Publication, 1st Edition, 2017.

[VIII] Samra J. Comparing Performance of Plain PHP and Four of Its Popular Frameworks. Thesis Project. Linnaeus University, Sweden.

[IX] Nylén. H. PHP Framework Performance for Web Development Between Codeigniter and CakePHP. Bachelor Thesis.