
QR based Automatic Penalty Charging for Violation of Traffic Rules

S. Hariswetha
Dept. of Information
Technology
Vivekanandha College
of Technology for
Women

S. Indira
Dept. of Information
Technology
Vivekanandha College of
Technology for Women

S. Latha
Dept. of Information
Technology
Vivekanandha College of
Technology for Women

T.Sivabharathi
Dept.of Information
Technology
Vivekanandha College of
Technology for Women

Abstract: The project cares with managing traffic violation penalty information through mobile app. The project contains licensedetails, vehicle details, penalty details, and accident information science through this app. The project also will specialize in mobile application which can help to understand information about the traffic violation penalty tracked by QR code on road. The fast Response (QR code) system became popular outside the automotive industry thanks to its fast readability and greater storage capacity compared to plain barcodes. Applications include product tracking, item identification, time tracking, document management, and general marketing. A QR code consists of black squares arranged during a square grid on a white background, which may be read by an imaging device like a camera. The specified data is then extracted from patterns that are present in both horizontal and vertical components of the image. This application will receive the fine automatically from the owner's checking account consistent with the rule broken by vehicle driver and its actual fine decided by government. This application will also send the message to vehicle owner about the sort of rule broken by them, fine consistent with that rule; proof of breaking the rule out the shape of image capture by mobile camera, date and time of breaking the rule and accurate amount receiving by their checking account a fine purchased breaking the rule.

Keywords : QR code, Penalty charging, Information security, Authentication.

I. INTRODUCTION

This project is being developed as an android application and aimed to assist traffic police to document all road traffic offences incurred by the road users. QR code payments allow merchants to receive payments from customers just by scanning generated QR codes employing a Smartphone-camera.

QR codes aka Quick Response code are two-dimensional bar codes. They contain a pattern of black squares arranged during a square grid on a white background. QR codes are often read with an imaging device like a smartphone camera. Merchants can now initiate a payment, QR code is generated and customer pays by scanning it together with his phone to transfer funds. The QR code payments carry the acquisition transaction information to the mobile device of the buyer/customer. A payment code is going to be generated for a customer to scan using his or her Easy pay app to transfer funds into the merchant account. So generate the QR code using the automated penalty charging system.

II. LITERATURE SURVEY

Driver distraction continues to receive considerable research interest but the drivers' perspective is a smaller amount well documented. The current research focused on identifying features that are salient to drivers in their risk perception judgments for 19 in-vehicle distractions [1]. It describes the overall framework and components of an experimental platform for driver behavior monitoring based on driver's traffic violation records [2].

A system designed for driver assistance vehicles operating during a wide selection of environments. [3]. Along these same lines, the majority of traffic violations such as speeding and ignoring stop signs are unintentional, and they occur due to a lack of concentration, rather than because drivers deliberately break the law [4]. Android is a mobile operating system (OS) currently developed by Google, supported the Linux kernel and designed primarily for touch screen mobile devices like smartphones and tablets. Android's interface is predicated on direct manipulation, using touch gestures that loosely correspond to real-world actions, like swiping, tapping and pinching, to control on-screen objects, alongside a virtual keyboard for text input. In addition to the touch screen devices, Google has further developed Android

TV for televisions, Android Auto for cars, and Android Wear for wrist watches, each with a specialized interface. Systems for automated image analysis are useful for a spread of tasks and their importance remains growing thanks to technological advances and arise of social acceptance [5]. Variants of Android also are used on notebooks, game consoles, digital cameras, and other electronics. As of 2015, Android has the most important installed base of all operating systems. It is the second most ordinarily used mobile OS within the us, while iOS is that the first. Initially developed by Android, Inc., which Google bought in 2005, Android was unveiled in 2007, alongside the founding of the Open Handset Alliance – a consortium of hardware, software, and telecommunication companies dedicated to advancing open standards for mobile devices. As of July 2013, the Google Play store has had over a million Android applications ("apps") published and over 50 billion applications downloaded.

An April–May 2013 survey of mobile application developers found that 71% of developers create applications for Android, and a 2015 survey found that 40% of full-time professional developers see Android as their priority target platform, which is like Apple's iOS on 37% with both platforms far above others. At Google I/O 2014, the corporate revealed that there have been over one billion active monthly Android users, up from 538 million in June 2013.

Android's ASCII text file has been used because the basis of various ecosystems most notably that of Google which is related to a set of proprietary software called Google Mobile Services (GMS), that regularly comes pre-installed on said devices. This includes core apps like Gmail, the digital distribution platform Google Play and associated Google Play Services development platform, and typically apps like the Google Chrome browser. These apps are licensed by manufacturers of Android devices certified under standards imposed by Google. Other competing Android ecosystems include Amazon.com's Fire OS, or Lineage OS. Software distribution is usually offered through proprietary application stores like Google Play Store or Samsung Galaxy Store, or open source platforms like Aptoide or F-Droid, which utilize software packages in the APK format.

Android Studio is that the official integrated development environment (IDE) for Google's Android OS, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) because the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage ranging from version 0.1 in May 2013, then entered beta stage ranging from version 0.8 which was released in June 2014. The first stable build was released in December 2014, ranging from version 1.0.

III. EXISTING SYSTEM

The existing system is in offline. Traffic violation penalty information is writing in printed charge sheet. Hard copies are maintained in traffic police department. Users need to pay to traffic police on the spot. This increases the time and corruption. The existing system has several drawbacks, such as traffic problems, complexity, cost, etc. The maintenance of the traffic offence Management system is difficult by using the existing spot billing machine (SBM), which increases the paper work. Therefore the matter stated above is often overcome using proposed application.

IV. PROPOSED SYSTEM

The proposed system is android based. The proposed system will reduce all the issues related to traffic rule violation on road. In this system there are automatic traffic rules violation fine collection through QR code, message sending and automatic fine receiving from owner's bank account.

In this system, QR reader will read the QR tag which is given to the individual vehicle while purchasing or passing the vehicle driving license received by RTO which will be mandatory.

V. ARCHITECTURE

Eclipse uses plug-ins to supply all the functionality within and on top of the run-time system. Its run-time system is predicated on Equinox, an implementation of the OSGI core framework specification.

In addition to allowing the Eclipse Platform to be extended using other programming languages, like C and Python, the plug-in framework allows the Eclipse Platform to figure with typesetting languages like LaTeX and networking applications like telnet and management systems. The plug-in architecture supports writing any desired extension to the environment, like for configuration management. Java and CVS support is provided within the Eclipse SDK, with support for other version control systems provided by third-party plug-ins.

With the exception of a little run-time kernel, everything in Eclipse may be a plug-in. Thus, every plug-in developed integrates with Eclipse within the same way as other plug-ins; during this respect, all features are "created equal". Eclipse provides plug-ins for a good sort of features, a number of which are from third parties using both free and commercial models.

Examples of plug-ins include for Unified Modeling Language (UML), for Sequence and other UML diagrams, a plug-in for DB Explorer, and many more.

The Eclipse SDK includes the Eclipse Java development tools (JDT), offering an IDE with a built-in Java incremental compiler and a full model of the Java source files. This allows for advanced refactoring techniques and code analysis. The IDE also makes use of a workspace, in this case a set of metadata over a flat file space allowing external file modifications as long as the corresponding workspace resource is refreshed afterward.

Eclipse implements the graphical control elements of the Java toolkit called Standard Widget Toolkit (SWT), whereas most Java applications use the Java standard Abstract Window Toolkit (AWT) or Swing. Eclipse's interface's also uses an intermediate graphical user interface layer called JFace, which simplifies the development of applications supported SWT. Eclipse was made to run on Wayland during a Google Summer of Code (GSoC) Project in 2014

VI. MODELLING PLATFORM

The Modeling project contains all the official projects of the Eclipse Foundation that specialize in model-based development technologies. All are compatible with the Eclipse Modeling Framework created by IBM.

Those projects are separated in several categories: Model Transformation, Model Development Tools, Concrete Syntax Development, Abstract Syntax Development, Technology and Research, and Amalgam. Model Transformation projects uses Eclipse Modeling Framework (EMF) based models as an input and produces either a model or text as an output. Model to model transformation projects includes ATLAS Transformation Language (ATL), an open source transformation language and toolkit won't to transform a given model or to get a replacement model from a given EMF model. Model to text transformation projects contains

Acceleo is an Implementation of MOFM2T, atypical model to text language from the thing Management Group (OMG). The Acceleo code generator can generate any textual language (Java, PHP, Python, etc.) from EMF based models defined with any meta-model (Unified Modeling Language (UML), Systems Modeling Language (SysML), etc.). It is open- source.

VII. DEVELOPMENT

Android is developed privately by Google until the newest changes and updates are able to be released, at which point the ASCII text file is formed available publicly. This ASCII text file will only run without modification on select devices, usually the Nexus series of devices. The source code is, in turn, adapted by OEMs to run on their hardware. Android's ASCII text file doesn't contain the usually proprietary device drivers that are needed surely hardware components.

In 2007, the green Android logo was designed for Google by a graphic designer Irina Blok. The design team was tasked with a project to create a universally identifiable icon with the specific inclusion of a robot in the final design. After numerous design developments supported science-fiction and space movies, the team eventually sought inspiration from the human symbol on restroom doors and modified the figure into a robot shape. As Android is open-sourced, it

Was agreed that the brand should be likewise, and since its launch the green logo has been reinterpreted into countless variations on the first design.

VIII. DESIGN

Unlike client-server management systems, the SQLite engine has no standalone processes with which the appliance program communicates. Instead, the SQLite library is linked in and thus becomes an integral a part of the appliance program. The library can also be called dynamically. Due to the server-less design, SQLite applications require fewer configurations than client-server databases. SQLite is named zero-conf because it doesn't require service management (such as startup scripts) or access control supported GRANT and passwords.

Several computer processes or threads may access an equivalent database concurrently. Several read accesses can be satisfied in parallel. A write access can only be satisfied if no other accesses are currently being serviced. Otherwise, the write access fails with a mistake code (or can automatically be retried until a configurable timeout expires). This concurrent access situation would change when handling temporary tables. This restriction is relaxed in version 3.7 when write-ahead logging (WAL) is turned on enabling concurrent reads and writes.

SQLite version 3.7.4 first saw the addition of the FTS4 (full text search) module, which features enhancements over the older FTS3 module. FTS4 allows users to perform full text searches on documents almost like how search engines search webpages. Version 3.8.2 added support for creating tables without rowed, which can provide space and performance improvements. Common table expressions support was added to SQLite in version 3.8.3. SQLite with full Unicode function is optional.

IX. CREATE AUTOMATIC PENALTY SYSTEMS

The existing system is in offline. Traffic violation penalty information is writing in printed charge sheet. Hard copies are maintained in traffic police department. Users

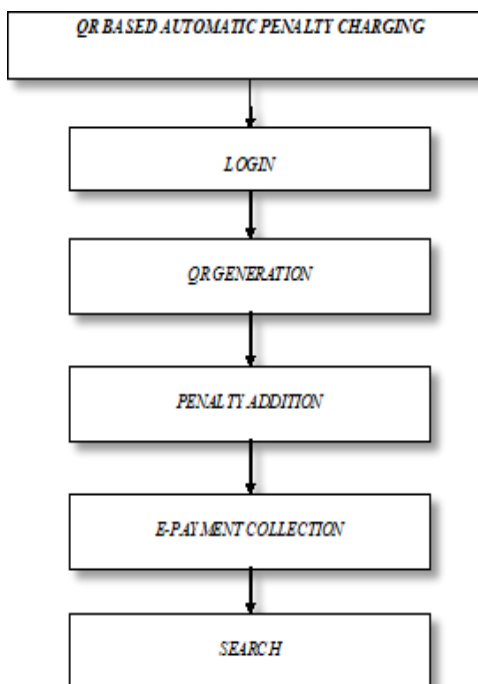


Figure 1 System Flow Diagram

need to pay to traffic police on the spot. This increases the time and corruption. The existing system has several drawbacks, such as traffic problems, complexity, cost, etc. The maintenance of the traffic offence management system is difficult by using the existing spot billing machine (SBM), which increases the paper work. Therefore the matter stated above is often overcome using proposed application.

User Login: In this module, only authorized users can login into our application using their username and password. A number of username and password are saved in the database and of which one of the username, password need to be given. A username may be a name that uniquely identifies someone on a computing system. For example, a computer could also be setup with multiple accounts, with different usernames for every account. A password may be a string of characters won't to verify the identity of a user during the authentication process.

QR Generation: during this module, the administrator generates QR code alongside vehicle information and owner details supported the knowledge registered.

(Code reader application). It's a machine-scannable image which will instantly be read employing a Smartphone camera. Every QR code consists of variety of black squares and dots which represent certain pieces of data.

Penalty Addition: In this module, penalty details about the vehicle are added. It includes information such as person license number, name, address, penalty amount and kind of rule breaking. After the penalty addition is updated, SMS about the details of the penalty is sent to vehicle owner. Penalty may be a punishment imposed for breaking a law, rule, or contract. It is usually imposed on persons who violate the law. Commonly penalty used in reference to financial punishment.

E-payment Collection: In this module collect penalty amount from vehicle owner bank account. It includes information such as person license number, name, address, penalty amount and kind of rule breaking. After the penalty collected from vehicle owner, details are sent by SMS to vehicle owner. An electronic payment is any quite of non-cash payment that does not involve a paper check.

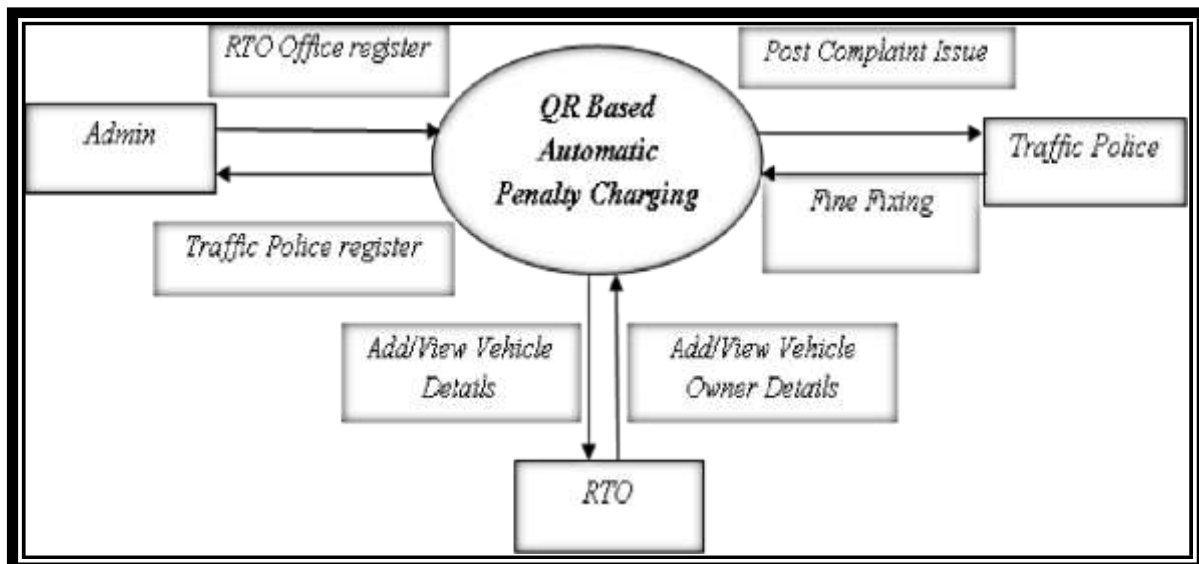


Figure 2 Data Flow Diagram of Automatic Penalty Charging System

S. No	FILED NAME	TYPE	SIZE	DESCRIPTION
1	Vehicle No	Varchar	15	Vehicle number details
2	Entry Date	Date Time	15	Details about the entry date
3	Fine Amount	Varchar	20	Amount details about the fine
4	Fine Details	Varchar	50	Details about the fine
5	Mobile No	Varchar	15	Mobile number Details

X. SYSTEM TESTING AND IMPLEMENTATION

TESTING

Testing is significant to the success of the system. System testing makes a logical assumption that if all parts of the system are correct, the goals are going to be successfully achieved. Within the testing process we test the particular system in a corporation and gather errors from the new system operates fully efficiency as stated. System testing is that the stage of implementation, which is aimed to making sure that the system, works accurately and efficiently.



Figure 3 QR Code on Number Plate

The purpose of the system testing is to think about all the likely variations to which it'll be suggested and push the system to its limits. The testing process focuses on logical intervals of the software ensuring that each one the statements are tested and on the function intervals (i.e.,) conducting tests to uncover errors and make sure that defined inputs will produce actual results that accept as true with the specified results.

SYSTEM IMPLEMENTATION

In the System development life cycle, the system implementation and maintenance will be occurring after the completion of analysis and system design.

The term implementation is ranging from the conversion of a basic application to a complete replacement of a computer system. In other term, implementation is used to process of converting a new or a revised system design into an operational one.

XI. FUTURE SCOPE

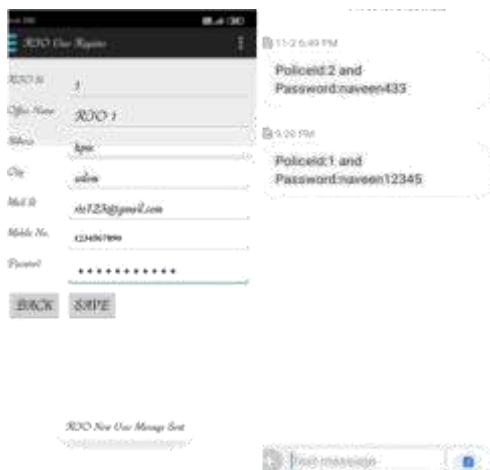
In case of a road network, navigation systems can attempt to bypass the critical zone. Furthermore, any traffic control systems can inform the drivers about the traffic jam risk in order to guide them around the critical zone. In order to detect the traffic different sensors are being used and different techniques are used to determine the traffic and thus solve

the problem related to traffic. If the application is designed as web service, it can be integrated in many web sites. The feedback details can be sent as mail to admin.

XII. OUTPUT DESIGN



Admin Menu Description



Set Details of RTO User

Message Notify as Username and Password to Traffic Police User



After Save the Details, generate QR of the Vehicle

Generate the QR as per the User Details



View All the Traffic Police User



Description of Complaint Details

CONCLUSION

Through this project, the issues related to traffic rule violation on road get reduced. This interface helps not only to control the traffic rule violation but also to control the corruptions. This project includes the feature to collect fine easily by using automatic fine reduction from owner's bank account. Since the application is designed as android application, any user can have the application by downloading or sharing this application. The application is tested well and end users satisfaction is found to be more.

The web application will reduce the paper work and therefore the time and energy spend for searching the vehicle. The fine payment is easier and even the person would know the number of violations he/she has committed, which will eventually let the person stop from doing more violations. using the RTO database would make this framework more efficient.

REFERENCES

- [1] Horrey, W.J. and M.F. Lesch, "Driver - initiated distractions: Examining strategic adaptation for in-vehicle task initiation", *Accident Analysis & Prevention*, Vol.41, no. 1, pp.115-122, 2009.
 - [2] Manubhai M., T. G. Trivedi, J. C. McCall, "Looking-In & Looking Out of a Vehicle: Computer-Vision-Based Enhanced Vehicle Safety", *IEEE Intelligent Transportation Systems*, Vol.8, no.1, pp.108-120, 2007.
 - [3] McCall, J.C and M. M. Trivedi, "Video- based lane estimation and tracking for driver assistance: survey, system, and evaluation", *IEEE Intelligent Transportation Systems*, Vol.7, no.1, pp. 20-37, 2006.
 - [4] Handmann U., T. Kalinke, C. Tzomakas, M. Wernerand, and W. V. Seelen, "An image processing system for driver assistance", *Image and Vision Computing*, Vol. 18, no.5, pp.367-376, 2000.
 - [5] Yulan L., M. L. Reyes, and J. D. Lee, "Real-Time Detection of Driver Cognitive Distraction Using Support Vector Machines", *IEEE Intelligent Transportation Systems*, Vol. 8, no.2, pp.340-350, 2007.
 - [6] Escalera, A, J. M. Armingol and M. Mata, "Traffic Sign Recognition and Analysis for Intelligent Vehicles". *Image and Vision Computing*, vol.12, no.3, pp. 247-258, 2003.
 - [7] Amudha, K, Nelson Kennedy Babu, C & Balu, S 2015, 'Effectual reversible watermarking method for hide the patient details in Brain tumor image', *International Journal of Computer, Electrical, Automation, Control and Information Engineering of World Academy of Science, Engineering and Technology*, vol. 9, no. 7, pp. 1713-1717.
 - [8] Horrey, W.J., M. F. Lesch, and A. Garabet, "Assessing the awareness of performance decrements in distracted drivers". *Accident Analysis & Prevention* Vol. 41, no.1 pp. 675- 682, 2008.
-