

ANDROID SUBURBAN BUS TICKET SYSTEM

Kushal Pal Singh¹, Saurabh Kulkarni², Harshdev Singh Randhawa³, Deepesh Jain⁴

Prof L.J. Sankpal⁵

^{1,2,3,4}Student, ⁵Guide Department of Computer Engineering, Sinhgad Academy of Engineering, Pune, Maharashtra, India

Abstract- In cities like Pune and Mumbai the busses are the nerves of the city. But they are behaving as an open invitation for evil minds to do mishaps as there is no maintenance of data of passenger. The tickets cost being of odd amounts and many other different problems while buying tickets. Also in this advance world we are dependent on the paper tickets and we even cannot book the ticket in advance doesn't seem fare so here is a solution. In our proposed system ticket can be bought with just a smart phone application and, where you can carry your suburban tickets in your smart phone as a QR (Quick Response) code. It uses the smart phones "GPS" facility to validate and delete your ticket automatically after a specific interval of time and once the user reaches the destination. User's ticket information is stored in a CLOUD database for security purpose which is missing in the present suburban system. Also the ticket checker is provided with a checker application to search for the user's ticket with the ticket number in the cloud database for checking purposes

KeyWords: Android; SQLite; Cloud Database; QR code

Introduction

In the past few years there were more advancement in the field of technology. Considering bus department, e-ticket facility was introduced where users browse through a governmental website and book their long journey bus tickets which can be printed out after confirmation to show it to the checker when needed. Also in foreign countries the use of Oyster cards & Octopus card has become mandatory during travel. Even the PMPML and BEST buses system have the facility of monthly passes. But we suffer if we forget our travel cards and we have to stand in the Queue for our local suburban tickets, which is a place where e-ticketing and m-ticketing were able to lay their foot prints. Android Suburban bus (ASB) ticketing is mainly to buy the suburban tickets which are the most challenging. Our ASB ticket can be bought with just a smart phone application, where you can carry your suburban bus tickets in your smart phone as a QR (Quick Response) code. It uses the smartphones "GPS" facility to validate and delete your ticket automatically after a specific interval of time once the user reaches the destination. User's ticket information is stored in a phones "GPS" facility to validate and delete your ticket automatically after a specific interval of time once the user reaches the destination. User's ticket information is stored in a cloud database for security purpose which is missing in the present suburban system. Also the ticket checker is provided with a checker application to search for the user's ticket with the ticket number in the cloud database for checking purposes.

SQLite implements most of the SQL standard, that uses a dynamically and weak typed SQL syntax that does not guarantee the domain integrity. SQLite operations can be multitasked, though writes can only be performed sequentially. The source code for SQLite is in the public domain. SQLite has many bindings to programming languages. It is the most widely used database, the most widely deployed database engine.

3. Present System

Android Cloud to Device Messaging (C2DM) is a service that helps developers to send data from servers to their applications on Android devices. This service provides a simple and lightweight mechanism that servers can use to tell mobile applications to contact the server directly, to fetch updated application or user data. The C2DM service handles all aspects of queuing messages and delivery to the target application running on target device. A QR code is a type of matrix-barcode (or 2-D code). The code consists of black modules arranged in a square pattern on a white background. The information encoded can be made up of four standardized modes of data (numeric, alphanumeric, Kanji, byte/binary).

Encryption:

An android app manages encryption and decryption of QR codes using DES Algorithm (56 bits), Japanese immigration use encrypted QR codes when placing visas in passports. Error correction:

Code word's are 8-bits long and use the Reed-Solomon error correction algorithm consisting of four error correction levels. The higher is the error correction level, the less is it's storage capacity. The approximate error correction capabilities at each of the four levels are as follows:

- Level L 7% of code word's can be restored.
- Level M 15% of code word's can be restored.
- Level Q 25% of code word's can be restored.
- Level H 30% of code word's can be restored.

Due to the design of the codes and the use of 8-bit code word's, an individual code block cannot exceed 255 code word's in length. Therefore, the larger QR symbols contain much more data and therefore it is necessary to break the message up into multiple blocks. The QR specification defines the block sizes so that no more than 15 errors can be corrected within each block. This limits the complexity of certain steps in the decoding algorithm. The Code blocks are then interleaved together, making it less.

4. PROPOSED SYSTEM

In this proposed system we actually propose that a system is possible for ticket booking of suburban where we use the android mobile application for ticket booking. The checker is provided with a checker application which can check the ticket validation either by referring the code generated in the ticket or by checking the Quick response (QR) code. Every ticket is given a unique code and the code is transferred to QR code by using google API. The Google API is the tool which can make a QR code of it which is secure code. The interface of can be created using the Eclipse. The simple interface can be developed using the development tool for android 2.2 and 8 API the phone of the user should have the minimum required android system and the phone should be GPS Enabled. The Checkers application is a simple decoder application of the QR code which will decode the QR Code and verify the ticket with the cloud database. In case by some defect the QR code is not able to be decoded then the checker can put the unique ticket code and do the verification. The Database of the system can be created on the cloud and the reference can be maintained. We even propose the GPS ticket verification system. Here the ticket is actually stored in the cloud and during the user travelling through the bus when enters the bus has actually activate the ticket and the ticket application will track the user from the source to destination. And as the user reaches the destination the ticket will be deleted from the cloud. OR in the technical words the status flag will be changed to the status used. This is the proposed system which is capable of using the ticket and work for it.

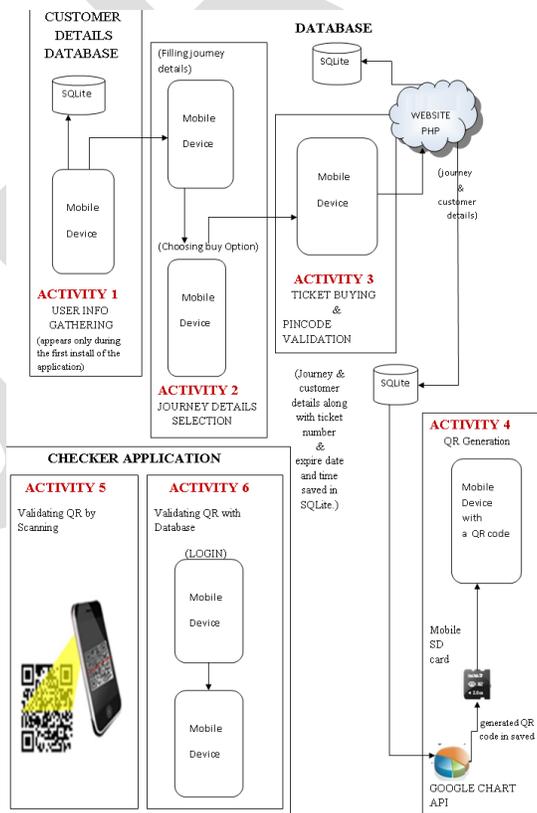


Figure 4.1: Block Diagram for ASB ticket booking

4.1 Individual Details

The installation of application starts from personal information. It gathers the customer information such as first name, last name, date of birth, city, state etc., and all this information will be stored into user mobile's, SQLite database. So whenever user buys the ticket this information is also sent to database. This process is basically used for security purpose and QR generation.

4.2 Ticket Buying

First the user selects source point, destination point, class, no. of child and adult tickets, ticket type is also choose by user like return or single etc. Then the user surf through the list of options to prefer either credit buy option or coupon buy it simplifies the buy process by remembering the credit card details. Once the user prefers any of these options the application moves on to the pin code validation module.

4.3 Pin Code Justification:

When the customer press the buy button a PHP code in the bus server validates the pin number and passwords, if it is triumphant it saves both journey details and customer info in the server's MySQL database. After this ticket number and time of buying is generated by PHP code and balance credit value is displayed.

4.4 QR Code Generation:

Once the PHP code generate the ticket number and time of buy details saved in the MySQL database are sent to GoogleChart API engine in order to generate the QR code. All the individual particulars and ticket information are transformed into QR codes and sent back to the user mobile as HTTP response and saved in the application recollection.

4.5. GPS Ticket Justification:

GPS plays the job of the checker, where when the user buys the ticket, the source geopoints, destination geopoints, ticket-type, termination time and date are stored in a mobile SQLite database. This facility checks the user's current location in accordance with the destination geopoints, after which the ticket type is checked and therefore the ticket is deleted if it is single or updated if it is return.

4.6. Examining QR code with QR Reader

In this part, checker will have QR code reader and inspect the QR code with the application with the purpose of authenticating the code and validate the journey tickets, particularly the time and date of the ticket.

4.7. Read-through with Database:

If suppose the user's display is being damaged and not able to examine the QR code because of other reasons like battery collapse, the user can use another infallible option to check the ticket by probing the ticket database with user's ticket number for justification purposes.

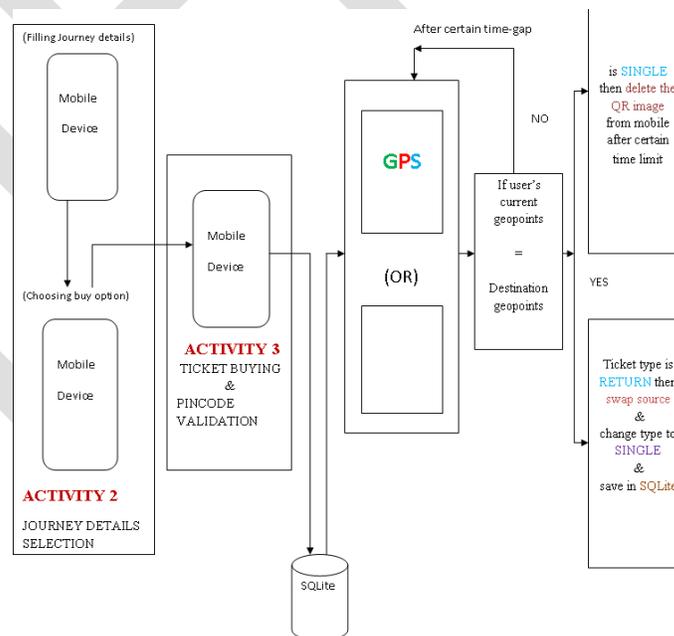


Figure 4.2 Block Diagram for ticket checking

CONCLUSION

The system will add a more secure and systematic way to travel in suburban buses. The system will be very adaptive in nature and can be implemented in different cities with ease. Also the system will give an alternative to the conventional ticket booking and will change the style of ticket booking.

ACKNOWLEDGEMENTS

We would like to sincerely thank Prof. L.J. Sankpal, our guide from Sinhgad Academy of Engineering for her support and encouragement.

REFERENCES

- [1] Damon Oehlman and Sebastien Blanc (2011) "ProAndroid WebApps develop for Android using HTML5, CSS3 & JavaScript" - Apress Publications.
- [2] Dave Smith and Jeff Friesen's (2011) "Android Recipes Problem Solution Approach" - Apress Publications.
- [3] Jeff "JavaJeff" Friesen's (2010) "LearnJava for Android Development" - Apress Publications.
- [4] Lauren Darcey and Shane Conder (2010) "Sams Teach yourself Android Application Development" - Sams Publications.
- [5] Mark Murphy's (2011) "Beginning Android 3" - Apress Publications.
- [6] Reto Meier (2009) "Professional Android Application Development" - Wiley Publishing Inc.
- [7] Satya Komatineni (2009) "Pro Android" - Apress Publications.
- [8] Shawn Van Every's (2009) "Pro Android Media developing Graphics, Music, Video and Rich Media Apps for Smartphones and Tablets" - Apress Publications.
- [9] Wallace Jackson's (2011) "Android Apps for Absolute Beginners" - Apress Publications.
- [10] Wei - Meng Lee (2011) "Beginning Android Application Development" - Wiley Publishing Inc.