Parking Getaway: An Android Application

Yukta Mehta[1], Neha Singh[2]
[1, 2] Department of Information Technology
[1, 2] Maharaja Agrasen Institute of Technology, New Delhi, India

Abstract- *In India, with the increasing number of vehicles, the problem of parking is also increasing. It is a common scenario that people park their cars on the roads behind other vehicles. This, in turn, creates chaos among the people, leads to fights and most importantly, creates traffic problems on the roads. To overcome these problems, very few numbers of people stick their contact details on their vehicles to avoid any problem caused due to their vehicle. However, this lets strangers invade their privacy and thus is not an effective solution. The "Parking Getaway" is an android application which helps the owners of the vehicles to find an effective solution of the above problem without compromising with their privacy. A QR Code is generated for every user who registers on the Parking Getaway app. The owner can stick this code on the vehicle and thereafter, if any person is affected because of the owner's parked vehicle, he/she can easily contact the owner by scanning the QR Code.*

Keywords- Android Application, Firebase, JAVA, QR Code, Simple Mail Transfer Protocol

I. INTRODUCTION

With nearly 1,200 vehicles being added to the city roads daily, parking problems in Delhi are compounding by the day. Colony roads, parking slots, basements and open parking areas in buildings and even open spaces now remain clogged with vehicles.

With families getting smaller and the total number of motor vehicles exceeding the total number of heads per family, the parking scenario is woefully falling short of the current requirements in the country. The situation is such that on any given working day approximately 40% of the roads in urban India are taken up for just parking the cars. The problem has been further exacerbated by the fact that nowadays even people from low income group are able to own cars. The number of families with cars has become much more than what the country is able to manage.

As it is, the cities in India are highly congested and on top of that the parked cars claim a lot of space that could otherwise be used in a better way. Thanks to poor, and at times zero, navigability, Indian cities are regarded as some of the worst options for living. One can also add the issue of pollution to this mix and understand the enormity of the crisis. In this context it needs to be understood that the Indian cities, with the possible exception of Chandigarh, were never planned in such a way so as to accommodate a deluge of cars as is the situation now. The apathy of present day urban planners has only made the situation worse.

It is mostly seen that people park their vehicles behind other vehicles and inappropriately anywhere on the roads. Frequent arguments, fights and even murders over the issue are becoming commonplace. The authorities have fallen well short of providing a wholesome solution to the problem. And with five lakh people coming to the city every year, even the suggested solutions come with a huge question mark.

The Parking Getaway application is implemented in Java based Android operating system and provides an efficacious solution to the parking problem by a simple concept of QR Code generation. A QR Code is generated for every user who registers on the Parking Getaway app. This QR Code is generated by the combination of Date of Birth and Phone Number, and is therefore unique for each user. It is generated through QRGen API which is an open-source, multi-format 1D/2D barcode image processing library implemented in Java, with ports to other languages. The owner can stick this code on the vehicle and thereafter, if any person is affected because of the owner's parked vehicle, he/she can easily contact the owner by scanning the QR Code. To check the genuineness of the affected person, a feature of sending real-time image through Email is provided so that no person takes undue advantage of the application by notifying the owner of the vehicle without any fault.

II. PROPOSED SYSTEM

A. Modules

1. Register User

This module helps the user to register him on this application. The user signs in through his/her Email ID. The registration process is then done by entering the following details:

- Phone Number
- Date of Birth

- Vehicle Number
- Token ID

All these fields are necessary to complete the registration process. The data is saved on the Firebase Database which is a real-time cloud-hosted database. It is a NoSQL database and has different optimizations and functionality as compared to a relational database. Data is stored as JSON and each connected client is synchronized in real time to the stored data.

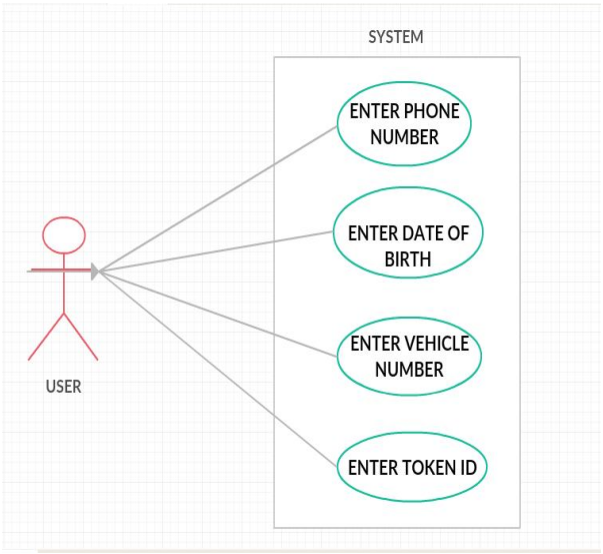

Figure 1. Use Case Diagram: Register User

2. QR Code Generation

Once the user successfully enters the above fields and clicks on "SAVE DETAILS", he gets registered on the application. The QR Code is then automatically generated for the user using the QRGen API. The QR Code is generated by the combination of Name, Email ID and Token ID so that it is unique for each user.

3. Send Notification

This module helps the user to send notification to the driver of the vehicle whose vehicle has created problem for the user.This module is further divided into two parts:

- **Scan the QR Code**

This lets the user to scan the QR Code which is pasted on the vehicle which has created problem for him/her. By scanning the QR Code, the name of the driver is displayed on the screen.

- **Click Image**

To ensure the authenticity of the problem, a feature of real-time image capturing is given. This assures that the user does not unnecessarily disturb the driver of the vehicle. After capturing the QR Code and the image, the notification is sent to the driver by clicking on the "Send Notification" button. The user can not resend the notification to the same driver before a specified time limit.

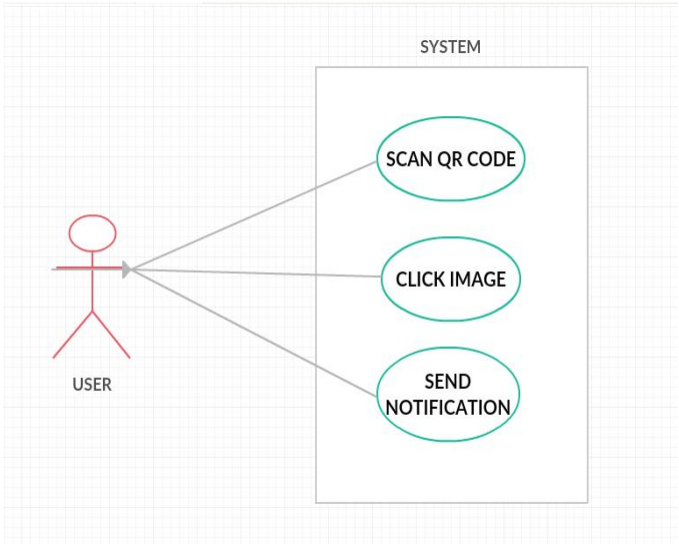

Figure 2. Use Case Diagram: Send Notification

4. Receive Notification

After the user sends a notification to the driver, the notification is received in two parts:

- **Application Notification**

This notification is received by the Firebase Cloud Messaging Service. The driver receives the notification and has two options:

A. To accept the message
B. To delete the message

Correspondingly, the notification is sent back to the user whether his request as accepted or deleted.

- **Image Notification**

The real-time image of the vehicle is sent to the driver at his registered Email ID through Simple Mail Transfer Protocol (SMTP). SMTP is a TCP/IP protocol used in sending and receiving e-mail and is usually implemented to operate over Internet port 25. By viewing the image, the driver can decide to take the desired action.

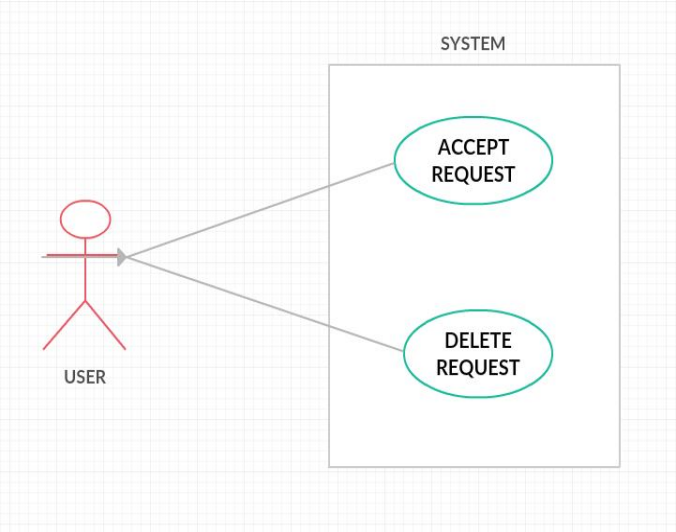

Figure 3. Use Case Diagram: Receive Notification

III. WORKING OF APPLICATION

The application works in the following manner:

- The user should have this application installed in their mobile phones.
- The user is then required to register on the app by entering personal details like Name, Date of Birth, Phone Number, Email ID, etc.
- The QR Code is generated with the help of the combination of Name, Email ID and Token ID.
- The user can then paste the generated QR code on his/her vehicle.
- If, at any time, a chaos is created by the parked vehicle, then, by scanning the QR Code, "Send Notification" option is shown.
- By clicking on this option, an automated message is sent to the owner of that vehicle, and he can then take appropriate action.

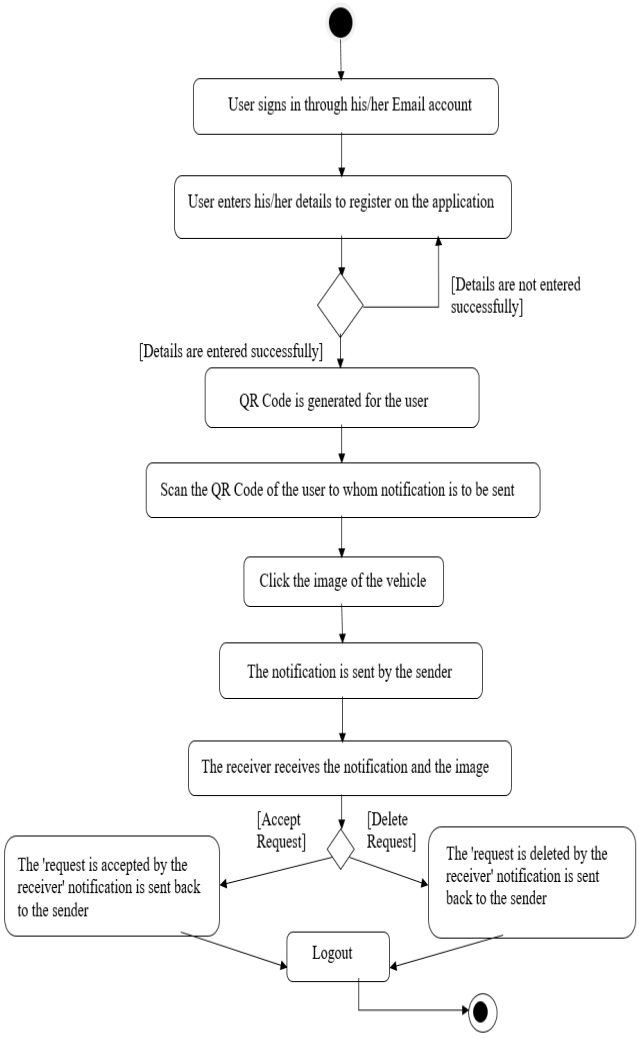

Figure 4. Activity Diagram

IV. CONCLUSION

In this paper, we proposed a method to easily get away from the unorganized parking problems. It uses the technologies like Android, Firebase Database and Simple Mail Transfer Protocol (SMTP) which makes it more efficient. In future, to make it more advanced, features like GPS, Chatbot can also be implemented in the application. In whole, this application will bring a huge difference and will also contribute in making the roads less chaotic.

REFERENCES

[1] Android (Operating System), http://en.wikipedia.org/wiki/Android_(operating_system).

[2] ADT Plugin, available at http://developer.android.com/sdk/installing/installing-adt.html

[3] The Android Developer's Cookbook, J. Steele, N. To, Addison-Wesley (2011).

[4] Android Application Development in 24 Hours , L. Darcey and S. Conder, Sams (2010).

[5] Android Developer site, http://developer.android.com/index.html

[6] Beginning Android 2, M. Murphy, Apress (2010).