

Development of Native Mobile Application Using Android Studio for Car Repairing And Servicing

¹ Satyam Srivastava, ² Himanshu Kumar and ³ Voruganti Naresh Kumar

^{1,2} B.Tech Student, Dept of CSE, CMR Technical Campus, Hyderabad

¹satyamsrivastaviit@gmail.com, ²himanshu747007@gmail.com

³ Assistant Professor of Dept of CSE, CMR Technical Campus, Hyderabad

³nareshkumar99890@gmail.com

Abstract:

This paper describes about the Car Repair application developed using Android Studio new version.. Car(e)Takar, is an android application for every wheelman looking towards an unblemished car. Car(e)Takar at first hand connects a user to a mechanic without the need of the interceder. This system in itself is pristine and uses concomitant technologies. Car(e)Takar also provides the basic aid that is needed by the user in order to introspect the basic fault in the vehicle which may occur. Car(e)Takar also provides the user to pay the bill online after the service is done and also provides with the e copy or the invoice service of the bill. Car(e)Takar also helps in connecting the user to the mechanics who are specialized in their respective domains and hence all the services which are provided are very effective and efficient. In this Application number of services available, so the customer can easily select the services, date of service, time of service etc.

Key Words — Android Studio, Car Repair Application, Services

I. INTRODUCTION

The scope of this paper is an android application prototype for a startup based on solving the above problem, that can be used by the garages and users alike to solve the problem of vehicle breakdown and help ease the process along the way. The application aims at providing a seamless transition between the user and the mechanics without any hassle [1].

This paper is not aiming at owning garages nor hiring any mechanics. It is a medium to connect users with the service providers at the right place at the right time. The actual implementation of the database is not a part of this paper. Issues of application security, other than login id and passwords are not in the scope of this paper. The implementation of Garage side of the application is not currently in the scope of the paper.

Paper includes all about the Car Repairing application development. As future work, an extension to this survey can be made, namely increasing the set of analyzed tools and comparison factors [2].

II.

METHOD

The application which is discussed in this paper is Car Servicing Application whose name is Car-(e)-Takar. For development of the Car-(e)-Takar, Car Servicing Application the platform used is 8.0 and the language used is JAVA and XML. SQLite is used at the backend side.

The first step is to download the new version of android studio that is 4.0 with its android SDK and android Virtual Device. After installing the Android studio and its setting up its environment, API level setting has been done. The procedure of setting API level is as follows New click on file>new>New project , to create a new project. In the Create, click on new project window, and enter the below given values- Application name: "Car Repairing App"

In the last field you will enter the path where you want to save. If you want to create your application according to you, so you need to select the blank activity from the activity module. Like this Activity you will add more things to your application which you want to show. When you open your application in Android Studio you have to See two file, first one is 'activity_main.xml' and the other one is 'content_main.xml'. Basically both do the same Thing, but the activity_main.xml contains the basic layout when you selected Basic Activity. In content_main.xml you can edit the content. There are various types of options in Android various options are called Widgets in which the user can drag and drop the things. Widgets enable users to interact with an Android Studio application page. There are various kinds of widgets, such as Buttons and Text Views. To see all the widgets at your disposal, create a new application project called "Widgets" and select "empty activity". Call your activity "Main Activity". After setting your layout design the user needs to run the application. First the developer needs to test the application on the emulator which we have already set up earlier as mentioned above which is called as Android Virtual Device (AVD). Then after the testing of application on AVD then the user needs to check manually on their mobile application. In order to check on physical device, first connect your device to your development machine with USB cable. Open Developer option and then enable USB debugging option on your device. After that, In Android Studio, you can click the application module in the project window and then click on Run in the toolbar. In the Select Deployment Target window, you will select your device and then click OK button. Android Studio installs the application on your connected device and starts it. Now you will see the application running which you have created on your device. Android Activity Lifecycle comprises of 7 methods. They are as follows.

- **OnCreate:** Called when activity is first created.
- **OnStart:** Called when activity is becoming visible to the user.
- **OnResume:** Called when activity will start interacting with the user.
- **OnPause:** Called when activity is not visible to the user.
- **OnRestart:** Called after your activity is stopped, prior to start.
- **OnDestroy:** Called before the activity is destroyed.

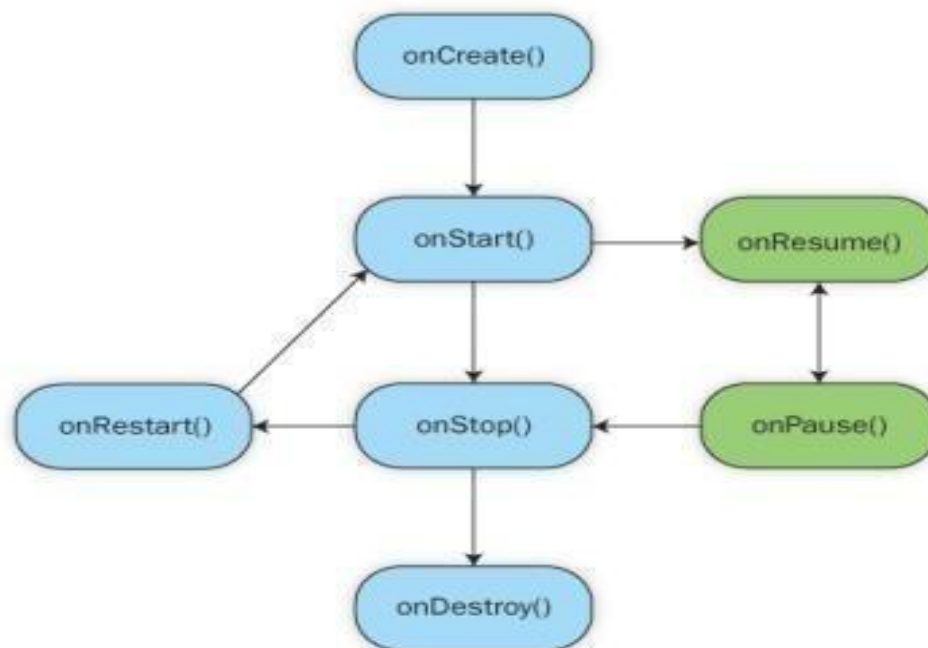


Fig 2.1: Life Cycle Of Android Application

III. Design Module of Car Repairing Application

The design module includes signup screen, car model screen, type of service screen, issue screen, payment gateways. Below are the details of the categories.

Signup Screen:

The user needs to register himself for the further process, the signup screen requires the user to enter the details like the User Name, phone number of user, user name, email id of the user

And the password field which is required to be filled by the user which will again get verified by re-entering the password.

Car Model Screen:

This Screen will allow the user to select its car model, and then further requirements can be fulfilled in the types of service screen.

Types Of Service Screen:

The user needs to select, whether the car is broken or it requires servicing, and then can choose whatever his requirements are.

Issue Screen:

This screen will redirect the user to select what is the issue, and will then redirect them to the basic fix for the vehicles. If the issue is not resolved then it will connect the user with the near by mechanic.

Payment:

The user need to pay the amount in final to the mechanic after the job is done. Payment gateway has been used in this application which include BHIM, PAYTM, PAYUMONEY etc. user can use this all payment gateways for the online payment.

IV.

RESULTS

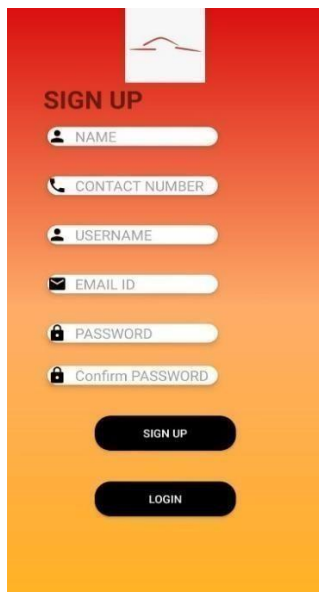


Fig 4.1: Signup Screen

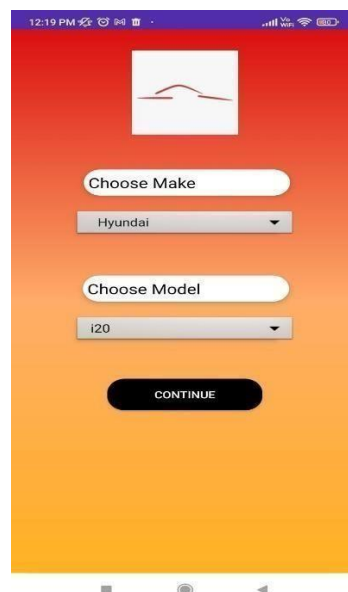


Fig 4.2: Car Model Screen

R



Fig 4.3: Types of service Screen

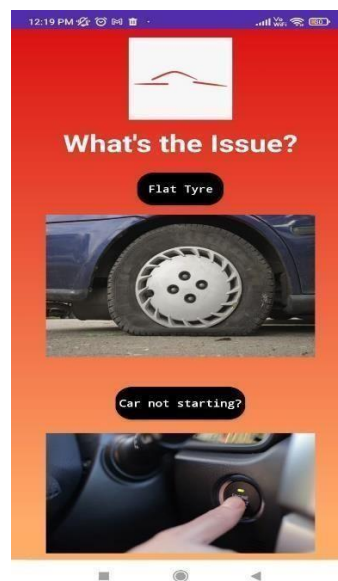


Fig 4.4: Issue Screen



Fig 4.5: Payment Screen

IV. FUTURE ENHANCEMENT

The project can be further developed into the real time application for which can be applied in the organizations on the networks. In future, the application can be made for more available models and can be made lighter in size. New Technologies can be used in order to improve the UI design of the application.

V. ACKNOWLEDGEMENT

We thank CMR Technical Campus for supporting this paper titled with “Development of Native Mobile Application Using Android Studio for Car Repairing and Servicing”, which provided good facilities and support to accomplish our work. Sincerely thank to our Chairman, Director, Deans, Head Of the Department, Department Of Computer Science and Engineering, Guide and Teaching and Non-Teaching faculty members for giving valuable suggestions and guidance in every aspect of our work.

VI. CONCLUSION

Car(e)Takt is the android application which helps the users to connect with their nearest mechanics depending upon their vehicle issues and also allows them to choose the payment method to the mechanic. It can be either online payment or the payment upon servicing. Users can select the mechanics depending upon their ratings. Car(e)Takt also helps the users in identifying their vehicle problems if they are unfamiliar with it.

Consider a situation where a user's car somehow ends up breaking down in the middle of nowhere! The user might try fixing the issue himself or even call the service centre but still, this would take up a lot of time!. Car (e) Takt aids people in such a situation by looking for mechanics in local garages and assigning a mechanic to the user. The mechanic would know the issue before hand and that will increase the chance of the issue getting resolved on spot.

VII. REFERENCE S

- [1] S. S. Jagtap and D. B. Hanchate, "Development of Android Based Mobile App for PrestaShop eCommerce Shopping Cart (ALC)," *International Research Journal of Engineering and Technology (IRJET)*, vol. 4, no. 7, pp. 2248–2254 , Jul. 2017.
- [2] A. Kathuria and A. Gupta, "Challenges in Android Application Development: A Case Study" *International Journal of Computer Science and Mobile Computing*, vol. 4, no. 5, pp. 294–299, May 2015.
- [3] L. Ma, L. Gu, and J. Wang, "Research and Development of Mobile Application for Android Platform," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 9, pp. 187– 198, 2014.
- [4] M. Q. Huynh and P. Ghimire, "Browser App Approach: Can It Be an Answer to the Challenges in Cross-Platform App Development?" *Journal of Information Technology Education: Innovations in Practice*, vol. 16, pp. 047–068, 2017.