

Development of Native Mobile Application Using Android Studio for Cabs and Some Glimpse of Cross Platform Apps

Priyanka Sisodia
Associate Professor
Geetanjali Institute of Technical Studies

Abstract

This paper tells about cab application developed using new version of android studio. It also includes about the cross platform on which both Android and iOS platform applications can be developed. The cross platform paper also includes an example of a cab application that will show its working and its uses. Cab application is an Android application created in Android Studio 8.0. Android Studio is an official integrated development tool or environment for Google's Android operating system. It is built on IntelliJ IDEA software from Jet Brains. The main objective of creating a cab application is to provide employment and make life easier for drivers, owners and customers. So basically we are trying to connect people (customers, drivers and cab owners) so that they can be mutually beneficial. Number of services available in this application, so that customer can easily select services, date of service, time of service etc.

Keywords: Android Studio, Cab Application, Cross Platform Application.

INTRODUCTION

CROSS PLATFORM APPS AND NATIVE APPS

Now days 99.6% of phones runs on either IOS or Android. Companies now treat mobile apps as a channel for raising their brand and advertise more for the marketing purpose about the apps. A native mobile app is an application which only looks for particular operating system by using their IDE and SDK. Native apps have the ability to use device-specific hardware and software and can provide quality with huge performance rates. The advantages of

Table No. 1

	Native	Cross-Platform
Multiple OS Support	No	Yes
User Interface Quality	High	Medium to High
Cost Of Ownership	High	Medium
Application Update	Native Market	Native Market

Nowadays everyone needs smart phone, mainly people use android phone for day to day communication. Many applications have been developed for unlimited entertainment of people's life and Android system has become popular in the smart phone market. The paper covers all about cab application development and difference between cross platform apps and native apps. Android is an operating system used for smart phones based on Android and Android Studio tools are used for developing Android applications. It is developed by Google. On other hand Cross-platform mobile development refers to the development of mobile apps that can be used on multiple mobile platforms with a single coding. There are lots of cross-platform tools which are available online nowadays being the major challenge understand which one is the best to achieve the goals of a certain user or company. Moreover, cross-platform tools are still evolving tools have flaws and limitations, but represent a straight forward solution to address the platform fragmentation problem. More frequent use of MDE concepts in this area may represent a good solution not only to develop cross-platform apps, but also to simplify development and attract a larger number of users through the use of domain concepts. As future work, this survey may be expanded, i.e. increasing the set of tools and comparison factors analyzed [1].

native mobile apps include – High performance, Ultimate user experience and Greater app store visibility [3]. A cross platform application is a mobile app which is compatible with multiple operating systems and can therefore run on any smart phones or tablets. The advantages of cross platform mobile app include:

Application	High	Medium
Maintenance		
Development Language	Java, C, C++, Objective C, Objective C++	Java, HTML, CSS, JavaScript

- Shorter Development time: It will need development or coding for one time and it will support the entire applications platform.
- Cost- effectiveness.

METHOD

The application which is discussed in this paper is Cab Application whose name is RS SHARING CAB. For development of the RS Sharing, Cab 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 3.0 with its android SDK and android Virtual Device. After installing the studio and its setting up its environment, API level setting has been done. The procedure of setting API level is 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-

The application which is discussed in this paper is a cab application named RS SHaring CAB. RS Sharing, the platform used to develop the CAB application is 8.0 and the language used is Java and XML.

SQLite is used on the backend side..

The first step is to download the new version of Android Studio which is 3.0 with Android SDK and Android Virtual Device. After installing Studio and setting up its environment, the API level setting is done. The process to set the API level is to create a new project by clicking File > New > New Project. Click Create in the New Project window, and enter the values below - in the last field you'll enter the path where you want to save. If you want to build your application accordingly, you need to select a blank activity from the Activity module. Like in this activity you will add more things to your application that you want to show. When you open your application in android studio you have to see two files, first is 'activity_main.xml' and second is 'content_main.xml'. Basically both do the same thing, but when you select the parent activity the activity_main.xml contains the original layout. In content_main.xml you can edit the content. In Android there is a bunch of option called 'Widgets' in which you can drag and drop things. In Android Studio, it is not showing XML code, but rather a rendering how the layout will appear on the screen. If you want to define the behavior of your application then you need to open MainActivity.java. You have found those tabs under Application>Java. After setting up your code or layout design you need to run your application. First, you test your application which you can run on Android Virtual Device called Emulator. After successful testing you need to test your application on real device. So first you connect your device with USB cable to your development machine. Open Developer Options and then enable the USB Debugging option on your device. After that, in Android Studio, you can click Application Module in the Project window and then click Run in the toolbar. In the Choose Deployment Target window, you will select your device and then click the OK button. Android Studio installs the application on your connected device and starts it. Now you will see that the application is running on your device. Android Activity Lifecycle is controlled by 7 methods of Android App Activity class.

- **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.
- **OnStop:** Called when activity is no longer visible to the user.

- **OnRestart:** Called after your activity is stopped, prior to start.
- **OnDestroy:** Called before the activity is destroyed.

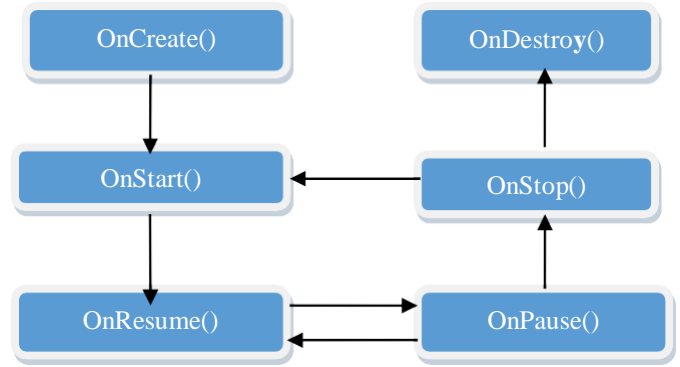


Figure1. Activity lifecycle

Intent is used as a message to move from one Activity to Another in a proper way.



Figure 2. Android Intent to navigate from one activity to another

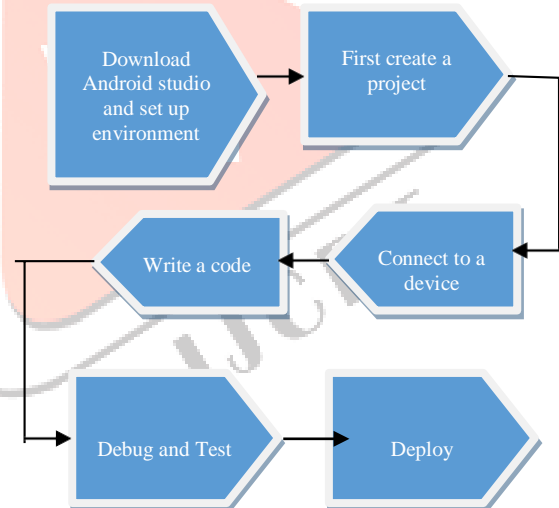


Figure 2. Process to build an Android app

DESIGN MODULES OF CAB APPLICATION

Application divided into two phases one is USER and another is ADMIN:

A. USER SIDE:

User side includes splash screen, registration screen, home screen with menu bar, payment gateways. Below are the details of the categories.

SPLASH SCREEN

Initially, when android application is started, at a very first splash screen has been opened, which will blink for 30 Sec. Splash screen includes application names with its logo. Basically android studio tool has inbuilt feature of the splash screen and after selecting the splash screen module it directly opens the inbuilt design of the screen which can be changed according to the requirements.

REGISTRATION SCREEN

After splash screen, the next screen will be a registration screen. The user has to register himself for the further process, registration screen includes full name, email Id, Mobile number and full address. Then after clicking on Proceed button, the next screen will open which is a search screen. If the user forgets the password, then he will get forget password link on the registered email id by which user can update the password.

HOME SCREEN/MENU SCREEN

Home screen will open from the left side of the application, which includes different different options for the users to open. Home screen contains registered logo and user name with his contact number which user can edit as per the need. It also includes My bookings, Booking Rides, Support, About, Update, Share.

PAYMENT

User can book the cab from Ludhiana to Delhi and it has an option for online payment of fares, 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.

B. ADMIN SIDE:

Admin can manage passengers and driver activities, add/delete/modify user, cab, location, Driver information like maintaining records, co-partners and Vendors etc.

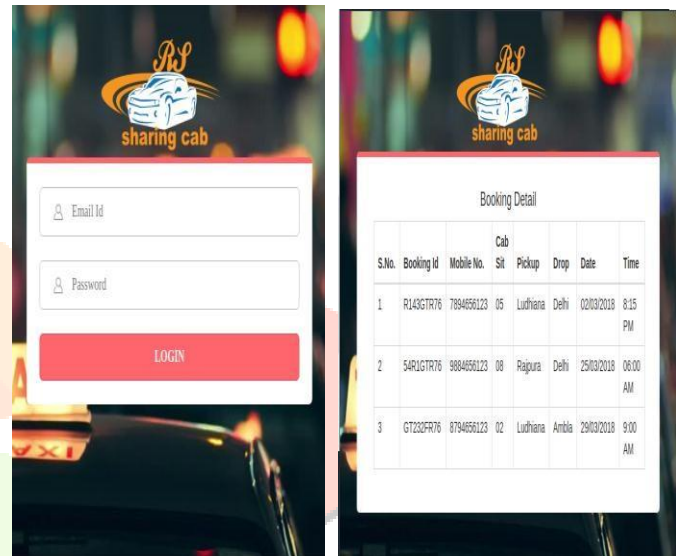
LOGIN SCREEN

The first screen will be a login screen. Admin login with Email ID and Password.

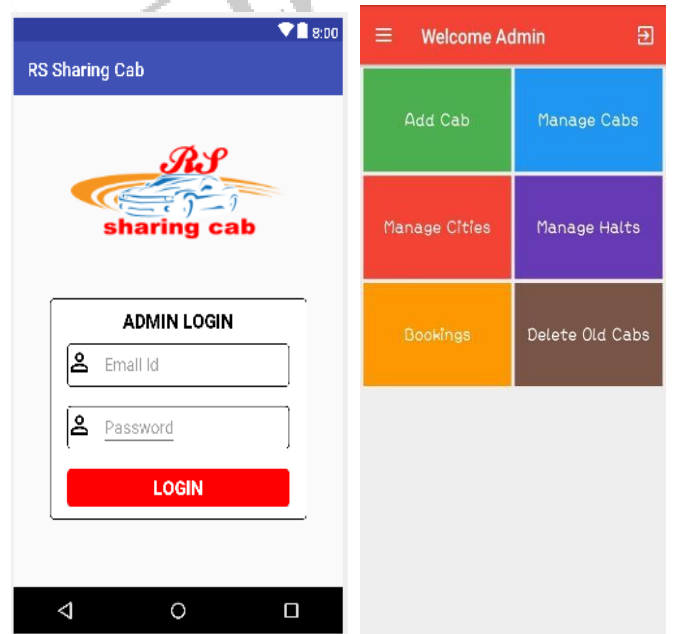
CAB MANAGEMENT

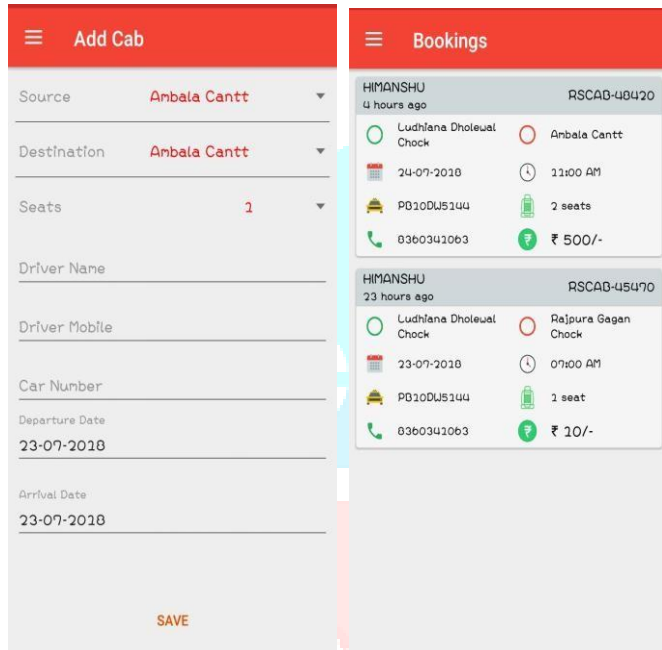
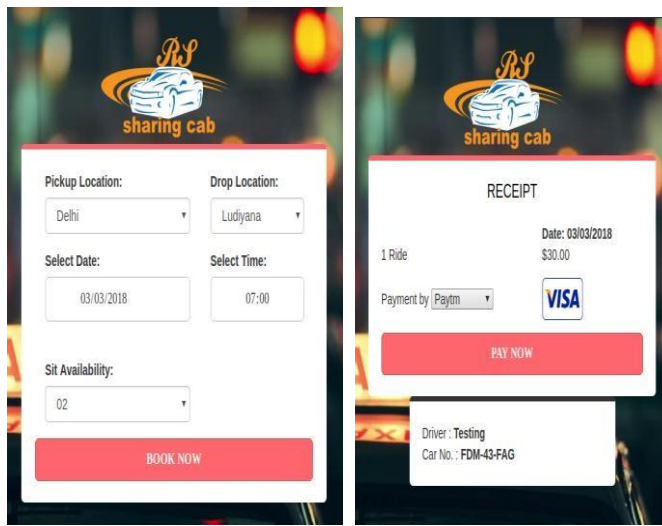
Admin can manage the cab added by the Drivers. Admin can review the Vehicle document and can approve/reject the Vehicle. Admin can add/modify/delete any Cab. Admin would define the Price per Kms, Price Per minute, Minimum Fare, Base Fare, Commission (%), Cancellation Charges and Peak time charges for each Vehicle Type.

USER SIDE SCREEN



ADMIN SIDE SCREEN





CONCLUSION

Cab application made the life quite easier. Some apps even bring drivers within five minutes of ordering cab. Passengers pay for these cabs assuming safe and comfortable journey and these app based services should respect customers for that. Proper investment should be made to make the journey safer as fulfilling the social responsibility by these apps will pay back in future.

ACKNOWLEDGEMENT

I would like to thank Mr. Huned Malviya, Mr. Mayanka Pandey for their genuine support in Infonex Solutions, Indore to complete this project.

REFERENCES

- [1]. Rineshi and A. R. D. Silva, "Survey on Cross-Platforms and Languages for Mobile Apps," *Eighth International Conference on the Quality of Information and Communications Technology*, 2012.
- [2]. Robin, V. C. Kulloli, T. Bhattacharyya, and S. Bhure, "Cross Platform Mobile Application Development," *International Journal of Computer Trends and Technology*, vol. 4, no. 5, pp. 1095–1100, 2013.
- [3]. 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.
- [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.
- [5]. N. Litayem, B. Dhupia, and S. Rubab, "Review of Cross-Platforms for Mobile Learning Application Development," *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 1, pp. 31–39, 2015.
- [6]. 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.
- [7]. 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.