

Implementation of Cross-Platform Apps in Android Studio and Native Mobile App Development for Cab

Anjali Parcha¹, Usha shani², Suman Rai³

Professor¹, Student^{2,3}

Department of CSE

Tilak Raj Chadha Institute of Management and Technology

Corresponding Author's E-mail: ushashani22sept@gmail.com

Abstract

This paper discusses the Cab application, which was created with the latest version of Android Studio. It also talks about cross-platform development, which may be used to create apps for both Android and iOS. Platform incompatibility The paper also offers an example of a Cab application that demonstrates how it works and what it may be used for. Cab is an Android application that was created with Android Studio 8.0.1. Android Studio is a Google's Android operating system's official integrated development tool or environment. It is based on the IntelliJ IDEA software from JetBrains. The primary goal of developing Cab Application is to create jobs while also making life easier for drivers, owners, and customers. So, basically, we're aiming to link individuals (clients, drivers, and taxi owners) so that they may profit from one other. This application has a large variety of services accessible, allowing customers to simply pick services, day of service, and hour of service, among other things.

Keywords: *Cross Platform Application, Mobile application, Android Operating System, Cab Application*

INTRODUCTION

Everyone nowadays need a smart phone, and the majority of people use Android phones for day-to-day communication.

Many applications have been created to provide limitless entertainment for people's lives, and the Android operating system has grown popular in the smart

phone industry. The paper discusses Cab application creation as well as the differences between cross platform and native apps. Android is an operating system for smart phones based on Android, and the android studio tool is used to build android applications. It was created by Google.

Cross-platform mobile development, on the other hand, refers to the creation of mobile apps that can run on several platforms using a single set of code. There are several cross-platform solutions accessible online nowadays, making it difficult to determine which one is ideal for achieving a certain user's or company's objectives. Furthermore, cross-platform tools are still under development and have defects and restrictions, but they offer a simple answer to the problem of platform fragmentation. Using MDE ideas more frequently in this area might be a useful way to not only construct cross-platform apps, but also to make development easier and entice a bigger audience through the usage of domain concepts. An expansion

of this study, meaning expanding the collection of studied tools and comparison factors [1,] can be done as future work.

NATIVE AND CROSS PLATFORM APPS

In today's world, 99.6% of phones run on either IOS or Android. Companies are now seeing mobile applications as a medium for promoting their brand and advertising more about the apps for marketing purposes. A native mobile app is one that uses the IDE and SDK of a certain operating system to look for it. Native apps may take use of device-specific hardware and software to deliver high- quality results at high speeds. High performance, ultimate user experience, and greater app store exposure are just a few of the benefits of native mobile apps. A cross platform app is a mobile app that is compatible with many operating systems and so may be used on any smartphone or tablet. The following are some of the benefits of a cross-platform mobile app:

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

- Less time spent developing: It will only require development or coding once, and it will support the complete application platform.
- Economic efficiency.

METHOD

The application covered in this article is RS SHARING CAB, which is a Cab Application. The platform utilised to construct the RS Sharing, Cab Application was 8.0, and the language used was JAVA and XML.

On the backend, SQLite is utilised.

The first step is to download Android Studio 3.0, which includes the Android SDK and Android Virtual Device. The API level has been established when the studio has been installed and its environment has been set up. The technique for adjusting the API level is as follows: To start a new project, go to file>new>New project. In the Create menu, select New Project from the drop-down menu, then fill in the data listed below. – "Cab App" is the name of the application, and the company domain is 'infonex.com'. All of this information will be utilised to construct a package name, such as com.infonex.cabapplication. This will contain all of the data related to the creation of Android, and it is referred to as

an android package file that must be uploaded to the Google Play Store. Finally, provide the location where the application's file should be saved.

If the developer wishes to construct a new application with different needs, the project template should be Empty Activity. Next should be selected. Accept the name of the default activity (Main Activity). Finish by clicking the Finish button. These are all described as android package files that must be uploaded to the Google Play Store.

In the final area, type the location where you wish to save the file. You must pick the blank activity from the activity module if you wish to develop your application according to your specifications. You will add additional stuff to your application that you wish to present if you like this Activity. When you start your app in Android Studio, you'll see two files: one called 'activity main.xml' and the other called 'content main.xml.' Both basically perform the same thing, except when you pick Basic Activity, the activity main.xml file provides the basic layout. You may edit the content in content main.xml.

Widgets are a collection of options in Android that allow you to drag and drop

items. It is not XML code that is displayed in Android Studio, but rather a depiction of how the layout will appear on the screen. You should open MainActivity.java if you wish to specify your application's behaviour. Those tabs may be accessed under Application>Java. You must execute your programme after you have established your code or layout design. To begin, you must first test your application on an Android Virtual Device known as an Emulator. After you've completed your testing, you'll need to put your app through its paces on a real device. As a result, you must first connect your device to your development machine via a USB cable. Open the Developer menu, then select USB debugging from the drop-down menu. Following that, Click the application module in the project window in Android Studio, then click Run in the toolbar. You will choose your device in the Select Deployment Target box and then click the

OK button. Android Studio launches and installs the app on your connected device. You will now see the application you generated on your smartphone running.

Android Activity Lifecycle is controlled by 7 methods of Android App Activity class.

- On Create: Called when activity is first created.
- On Start: Called when activity is becoming visible to the user.
- On Resume: Called when activity will start interacting with the user.
- On Pause: Called when activity is not visible to the user.
- On Stop: Called when activity is no longer visible to the user.
- On Restart: Called after your activity is stopped, prior to start.
- On Destroy: Called before the activity is destroyed.

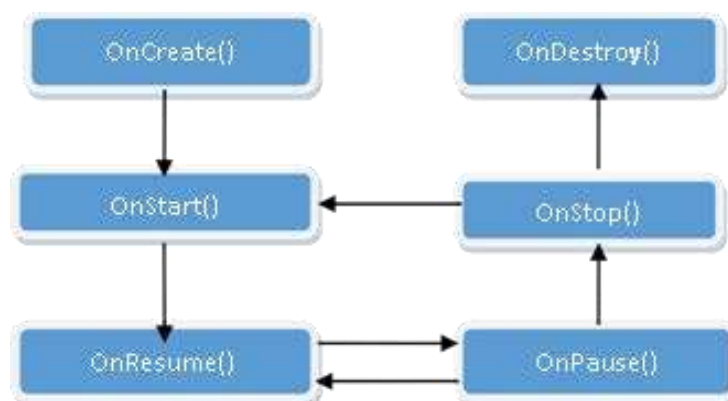


Figure1. Activity lifecycle

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

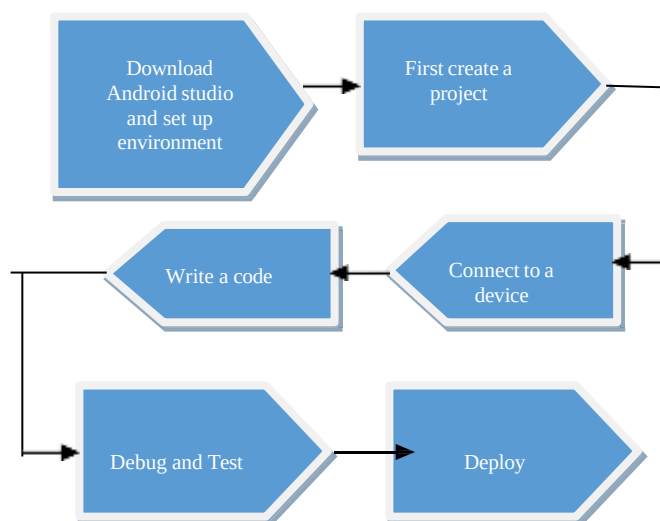


Figure 3. Process to build an Android app

DESIGN MODULES OF CAB APPLICATION

The application is separated into two phases, one for the user and another for the 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

Splash screen, registration page, home screen with navigation bar, and payment gateways are all on the user side. The categories' specifics are shown below.

When an Android programme is initially launched, it displays a splash screen that will flicker for 30 seconds. The splash screen displays the application's name as well as its logo. Basically, the splash screen is an inherent component of the Android Studio tool, and when you pick the splash screen module, it immediately opens the inbuilt design of the screen, which may be adjusted according to your needs.

SCREEN FOR REGISTRATION

The registration screen will appear after the splash screen. The user must first

register in order to proceed with the procedure. The registration page requires the user's complete name, email address, mobile phone number, and full address. Then, after hitting the Proceed button, the following screen, a search screen, will appear. If a user forgets his password, he will receive a forget password link to his registered email address, which he may use to change his password.

MENU SCREEN/HOME SCREEN

The application's home screen will appear on the left side, with a variety of alternatives for users to choose from. The user's contact number and registered logo are displayed on the home screen, which the user can alter as needed. My bookings, Booking Rides, Support, About, Update, and Share are also included.

PAYMENT

The user may book a taxi from Ludhiana to Delhi, and there is an option for online fare payment. Payment gateways like as BHIM, PAYTM, and PAYUMONEY have been implemented in this application. For online payments, the user can utilise any of the payment gateways.

B. ADMIN SIDE: Admin may control passenger and driver actions, as well as

USER SIDE SCREEN

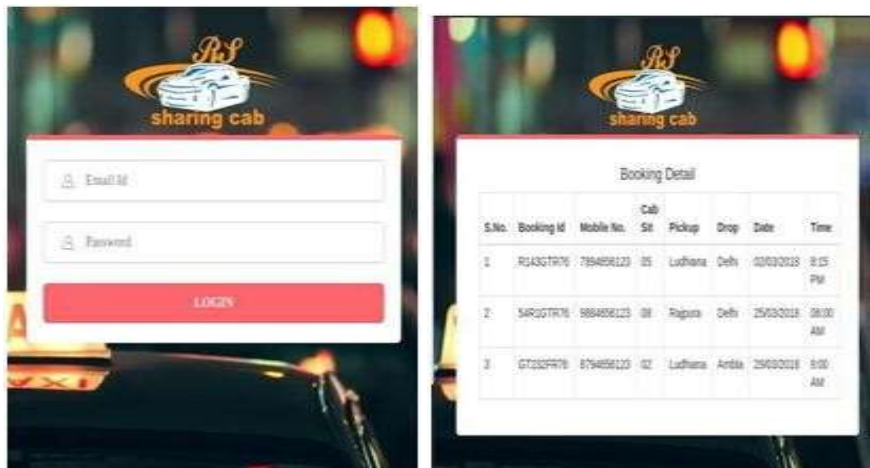
add/delete/modify user, cab, and location information, as well as co-partners and vendors.

SCREEN FOR LOGIN

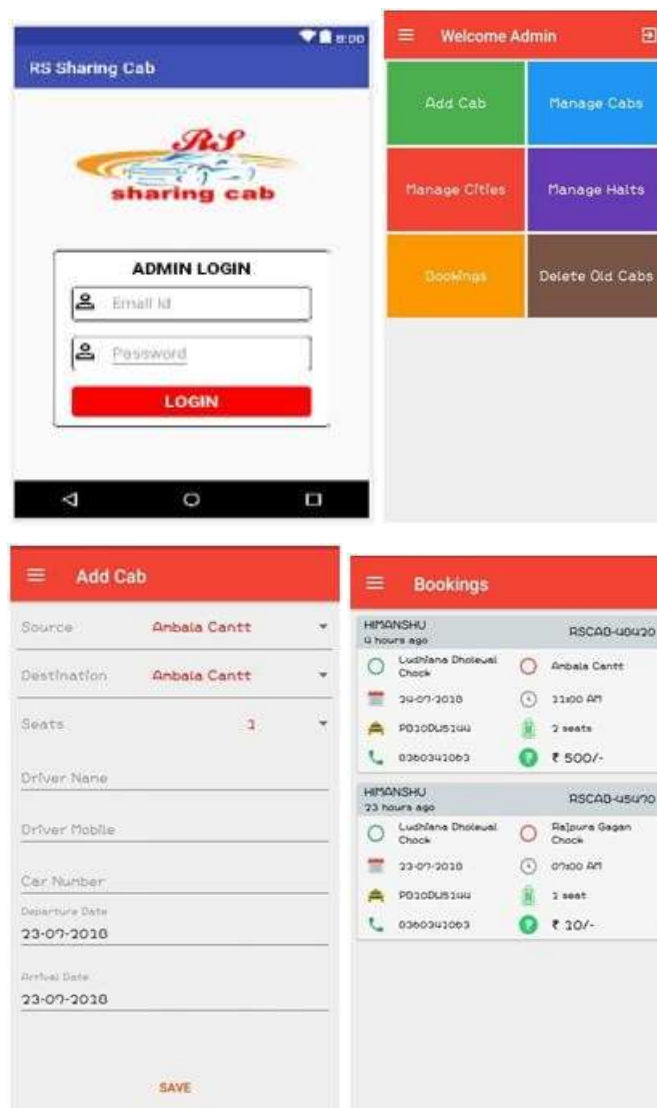
A login screen will appear on the first screen. Login as an administrator using your email address and password.

CAB MANAGEMENT is a company that specialises in the management of taxicab. The cabs that the Drivers have uploaded may be managed by the Admin. Admin has the ability to evaluate the Vehicle paperwork and approve or reject it.

Any Cab can be added, modified, or deleted by the administrator. For each Vehicle Type, Admin would establish the Price per Kilometer, Price per Minute, Minimum Fare, Base Fare, Commission (percentage), Cancellation Charges, and Peak Time Charges.



ADMIN SIDE SCREEN



CONCLUSION

The cab application makes life a lot easier. Some applications can even connect you with a driver within five minutes of placing an order. Passengers pay for these cabs with the expectation of a safe and enjoyable travel, and these app-based services should honour that expectation. Proper investment should be made to make the travel safer, since these applications' social responsibilities will pay off in the long run.

REFERENCES

1. 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.
2. 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.
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. Pohares, 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.
5. Ribeiro 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.
6. 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.
7. 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.