

APPLICATION DEVELOPMENT USING FLUTTER

Aakanksha Tashildar*¹, Nisha Shah*², Rushabh Gala*³,
Trishul Giri*⁴, Pranali Chavhan*⁵

*^{1,2,3,4}SPPU, Department of Computer Engineering, Vishwakarma Institute of Information Technology,
Pune, Maharashtra, India.

*⁵Assistant Prof, Department of Computer Engineering, Vishwakarma Institute of Information Technology,
Pune, Maharashtra, India.

ABSTRACT

Cross-platform mobile application development is the pressing priority in today's world and generation. Developers are enforced to either construct the same application numerous times for various OS (operating systems) or accept a low-quality similar solution that trades native speed and accuracy for portability. Flutter is an open-source SDK for developing high-performance and more reliable mobile applications for operating systems like iOS and Android. Significant features of the Flutter are Just-in-time compilation which executes the computer code that encompasses compiling during program execution at run time rather than preceding execution. More frequently, this comprises of bytecode translation lesser-known as source code to machine code, which is unswervingly executed. AOT compilation (Ahead-of-time compilation) compiles a high-level programming language such as C or C++, or an intermediary representation such as Java bytecode or NET Framework Common Intermediate Language (CIL) code, into native system-dependent machine code so that the subsequent binary file can execute natively. Flutter has a feature called as hot reload which helps you easily experiment, build UIs, add features, and fix bugs. Hot reload works by inserting updated source code files into the running Dart Virtual Machine (VM). After the VM updates classes with the new versions of fields and functions, the Flutter framework automatically reconstructs the widget tree, permitting you to rapidly view the special effects of your changes. Flutter targets the top mobile operating systems like Android and iOS, it gives you a solution for GPU rendering and UI, powered by native ARM code.

KEYWORDS: Cross-Platform Mobile application development, IDE, Android development, iOS development, Flutter, Dart.

I. INTRODUCTION

Mobile applications are having a progressively more significant role in our day to day lives. Ever since November 2016, there is more network traffic made by mobile devices (48.19%) compared to desktops or laptops (47%). To dispense it to most of the users, a mobile application needs to familiarize itself with two independent platforms which are Android and iOS. These two platforms share immense dissimilarities which often necessitate different skill sets for developing. For example Java or Kotlin for Android and Object-C or Swift for iOS. Hence, developers and companies usually struggle to deal with the complex nature involved in developing cross-platform applications. On March 15, Facebook introduced an open-source cross-platform JavaScript framework called as React Native which strives to solve the aforesaid problem. In addition to its effective developing process, developers use React framework extensively because of its non-complex and easy nature. Further, Google announced another mobile SDK named Flutter in the latter half of 2016. Inspired by React Native, Flutter application can also run equally on both platforms, consequently decreasing the cost and complexity of application creation across iOS and Android. Flutter is fully built from scratch and at the time of scripting this study (Aug 2017), only Google uses it for commercial project.

Cross-platform frameworks that show resemblance to React Native and Flutter, are discussed and implemented by various companies numerous times formerly. Still, neither of them suffices to satisfy the requirement of industrial development. In spite of the ineffective precursors, React Native and Flutter which are backed up by Facebook and Google, draw attention and people are optimistic about its prospect.

II. FLUTTER

Flutter is a cross-platform framework that targets developing high-performance mobile applications. Flutter was publicly released in 2016 by Google. Besides running on Android and iOS flutter applications also run on Fuschia. Flutter is chosen as Google's application-level framework for its next-generation operating system. Flutter is exceptional because it is dependent on the device's OEM widgets rather than consuming web views. Flutter uses a high-performance rendering engine to render each view component using its own. This provides a chance to build applications that are as high-performance as native applications can be. In view of architecture, the engine's C or C++ code involves compilation with Android's NDK and LLVM for iOS respectively, and during the compilation process, the Dart code is compiled into native code. Hot reload feature in Flutter is called as Stateful hot reload and it is a major factor for boosting the development cycle. Flutter supports it during development. Stateful hot reload is implemented by sending the updated source code into the running Dart Virtual Machine (Dart VM) without changing the inner structure of the application, therefore the transitions and actions of the application will be well-preserved after hot reloading.

III. DART

In Flutter, every application is written with the help of Dart. Google has developed and maintained a programming language called Dart. It is extensively used inside Google and it has been verified to have the proficiency to develop enormous web applications, such as AdWords. Originally Dart was developed to replace and succeed JavaScript. Thus, it implements most of the important characteristics of JavaScript's next standard (ES7), such as the keywords "async" and "await". Nonetheless, to attract developers that are not acquainted with JavaScript, Dart has a Java-like syntax. Flutter application renews the view tree on every new frame even when few other systems use reactive views. This behavior leads to a drawback that many objects, which might survive for a singular frame, will be created. As Dart is a modern programming language, it is optimized to handle this scenario in memory level with the help of "Generational Garbage Collection".

IV. PROPOSED METHODOLOGY

a) System Overview-

We have developed a flutter based mobile application for- Billing and Reward system. The redeem point or point system emphasizes on the principle that the more money you spend, the more points you get in return so that you could redeem those points in the future purchase. Each time a customer purchases something, they get a definite sum of points depending on how much they've purchased i.e. money spent.

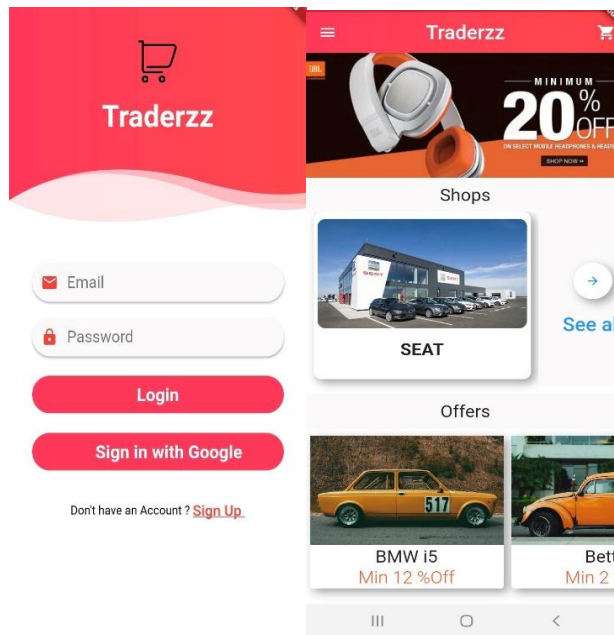
For example, customers earn 10 points for every Rs.1000 they spend online in the app. Further, customers can direct these points towards future purchases.

To further enhance customer engagement, we have developed a mobile application where users are facilitated to manage their accounts, buy new products, check their point status, redeem points, and so forth. By rewarding the customers with redeemable points, you increase your customer's average order value and encourage them to invest in your brand, and their likeliness to switch to a competitor decreases.

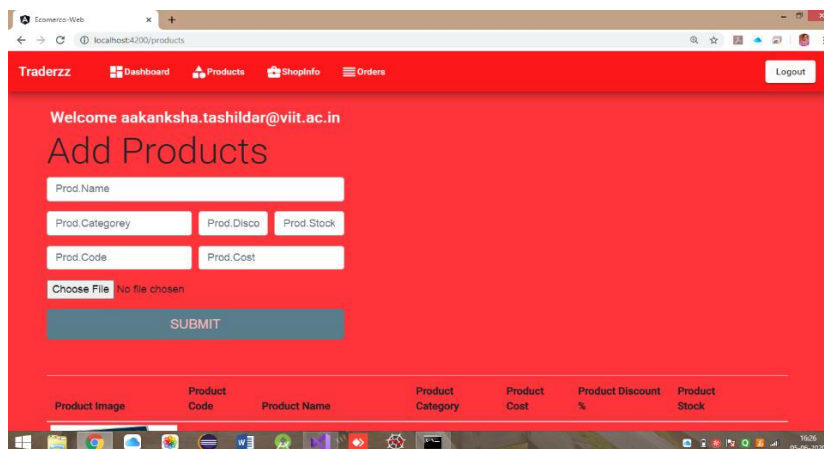
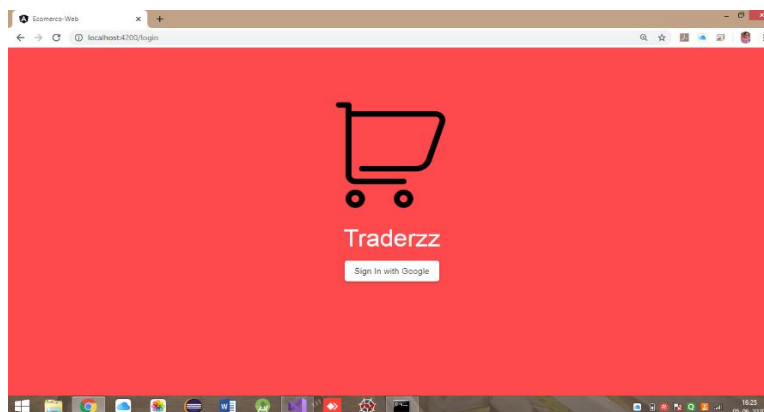
b) Proposed System-

Our system consists of two parts: "the client" and "the server". The client side is deployed on an Android or iOS based mobile phone. The server side is deployed on a Windows OS.

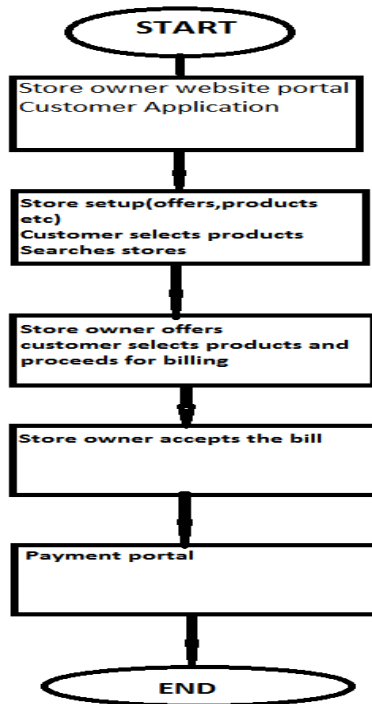
The client side on Samsung Android 10 mobile phone-



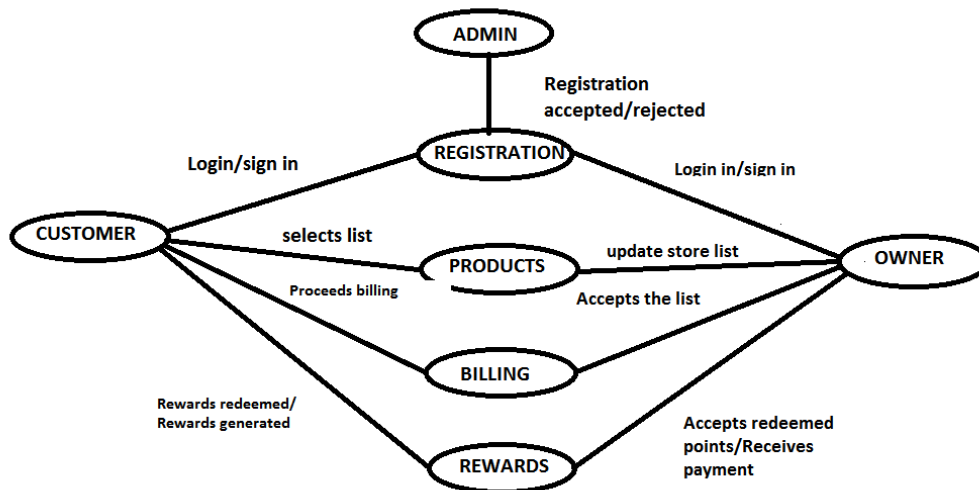
The server side on Windows 8.1 Laptop-



c) Workflow Diagram-



d) State Diagram-



V. CONCLUSION

Points to the success of a mobile-driven reward system are helping the retailers and small to medium shop owners to attract new customers, retain existing ones, and motivate increased purchase among current consumers. Offering loyalty programs attract the customers to invest in your brand and also leaves its imprints on their mind and brings your brand in the limelight.

VI. REFERENCES

- [1] React Native vs Flutter, Cross-Platform Mobile Application Framework, Thesis March 2018- Wenhau Wu.
- [2] A clean approach to Flutter Development through the Flutter Clean architecture package, IEEE 2019, Shady Boukhary, Eduardo Colemenares.
- [3] Exploring end user's perception of Flutter mobile apps, Malmo University Nov 2019- Dahl, Ola.
- [4] Flutter for Cross-Platform App and SDK Development, Metropolia University Thesis May 2019- Lucas Dagne.
- [5] Cross-Platform Framework comparison- Flutter vs React Native.
- [6] Flutter Native Performance and Expressive UX/UI, paper 2019- Tran Thanh.