

Application to the Management of Logistics System

A project Report

Submitted by

Rohan N	20171CSE0579
Bruna H Y	20171CSE0140
Ranjan Shetty	20171CSE0569
Sunil Reddy	20171CSE9022
Varshini Sadanand J S	20171CSE9017

Under the guidance of

Ms. Manasa C M

*in partial fulfillment for the award of the
degree of*

Bachelor of Technology

IN

Computer Science and Engineering

At



Department of Computer Science and Engineering

School of Engineering

PRESIDENCY UNIVERSITY

BANGALORE

May 2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF ENGINEERING

PRESIDENCY UNIVERSITY

CERTIFICATE

This is to certified that the Project report “**Application to the Management of Logistics System**” being submitted by “**Rohan N, Brunda H Y, Ranjan Shetty, Sunil Gowda, Varshini Sadanand J S**” bearing roll number(s): **20171CSE0579, 20171CSE0140, 20171CSE0569, 20171CSE9022, 20171CSE9017**, in partial fulfillment of requirement for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** is a bonafide work carried out under my supervision.

Dr. C KALAIARASAN

UP-II

Associate Dean-Admin

Department of CSE

Presidency University

Ms. Manasa C M

Assistant Professor

Department of CSE

Presidency University

**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING**

SCHOOL OF ENGINEERING

PRESIDENCY UNIVERSITY

DECLARATION

I hereby declare that the work, which is being presented in the project report entitled **Application to the Management of Logistics System** in partial fulfillment for the award of Degree of **Bachelor of Technology in Computer Science and Engineering**, is a record of our own investigations carried under the guidance of **Ms. Manasa C M, Assistant Professor, Department of Computer Science and Engineering, School of Engineering, Presidency University, Bangalore.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

Rohan N	20171CSE0579
Brunda H Y	20171CSE0140
Ranjan Shetty	20171CSE0569
Sunil Gowda	20171CSE9022
Varshini Sadanand J S	20171CSE9017

ABSTRACT

In today's competitive business environment, getting the correct products to the right place at the right time via the most-cost effective manner is paramount for any business' success. Whenever entrepreneurs think of widening their footprints across geographies, the sector that lends a helping hand is Logistics. Logistics is one of the most important and complex part of any business. Effectively organized logistics is of paramount importance in any small scale or large-scale business. It includes transportation, shipping, receiving, storage and managing every aspects of supply chain. In order to smoothen the operations and address the ever-changing needs, the logistics domain needs a technology support. There, logistics mobile app development comes into the picture. This application lets one constructively manage the logistics. Our project is focused on developing a logistics android application with the main objective of achieving a successful management system for logistics that is concerned with getting the right product, in the right quality, in the right condition, at the right place, at the right time, to the right customer and at the right price. So the project is primarily concentrated on the three main modules which include the Customer Panel, the Driver Panel and the Admin Panel. The Customer panel consists of registration, booking features and payment options. The Driver Panel consists of login, shipper details, update location feature. Finally the Admin Panel consists driver monitoring dashboard feature, add vehicles and drivers feature and vehicle management option. By never losing touch with one's customer, one is able to collaborate as well as better tracking and responding to cases, allowing them to maintain and build stronger customer relationships.

ACKNOWLEDGEMENT

First of all, we indebted to the GOD ALMIGHTY for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Abdul Sharief**, Dean, School of Engineering, Presidency University for getting us permission to undergo the project.

We record our heartfelt gratitude to our beloved professor **Dr. C. Kalaiarasan**, University Project-II In charge, Associate Dean-Admin, Department of Computer Science and Engineering, Presidency University for rendering timely help for the successful completion of this project.

We are greatly indebted to our guide **Prof. Manasa C M, Assistant Professor**, Department of Computer Science and Engineering, Presidency University for his/her inspirational guidance, valuable suggestions and providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We thank our friends for the strong support and inspiration they have provided us in bringing out this project.

Rohan N
Brunda H Y
Ranjan Shetty
Sunil Gowda
Varshini Sadanand J S

TABLE OF CONTENT

1. Introduction.....	08
2. Previous Work.....	10
3. Literature Review.....	11
4. Requirement Analysis.....	23
4.1 Advantages of Android Development.....	23
4.2 Configure build.....	25
4.3 Connect o Firebase.....	27
4.4 Custom build configurations.....	28
4.5 7 Key Android Concepts.....	30
5. Proposed System Design and Working.....	35
5.1 Architecture Diagram.....	35
5.2Design of Proposed System.....	36
6. Implementation.....	45
7. Verification and Testing.....	53
8. Future Scope.....	62
9. Conclusion.....	63
10. References.....	64

11. Appendix 1.....	64
12. Appendix 2.....	66

1. Introduction

The cool and alluring features that come with apps are what draw the interests of users. Apps make phone “smart” and through their benefits, apps have drastically transformed how we function today. It is user-friendly, has huge community support, provides a greater extent of customization, and a large number of companies build Android-compatible smart phones. As a result, the market observes a sharp increase in the demand for developing Android mobile applications.

The Logistics drive business globally. Logistics is one of the most important and complex part of any business. Whenever entrepreneurs think of widening their footprints across geographies, the transportation and logistics sector lends a helping hand. This is the reason why this sector is thriving by leaps and bounds in the current scenario. Logistics can manage the transportation of the goods and materials and involve various processes including warehousing, material handling, supply chain management and the like. To smoothen the operations and address the ever-changing needs, the logistics domain needs a technology support. There, logistics mobile app development comes into the picture. This application lets one constructively manage the logistics. here are certain concepts in the transport and logistics domain: logistics, transport, transportation etc. There are also a number of different study programs and disciplines such as: transport economics, transport logistics, logistics management, etc. All of these conceptual topics are often related to the content of the plan, planning, organization, management and control of the movement of human, material, informational and financial flows. However, when it comes to actual practice, those conceptual titles are often misused and correlated. Hence there is an important task to perfectly define these concepts. The Logistics hold a major contributor to any business globally. Logistics bear the management of goods and materials transportation and involve various processes including, warehousing, material handling, supply chain management and the like.

Any Logistics app primarily has the following features:

1. Create a warehouse and hub database.
2. Add vehicles and drivers.
3. Create a shipment booking form.
4. Enable notifications for tracking shipments.

5. Build a logistics dashboard.
6. Build a separate dashboard for individual or small scale shipments.
7. Create a scheduler to book meetings.

Our project is primarily concentrated on the three main modules which include the Customer Panel, the Driver Panel and the Admin Panel. The Customer panel consists of registration, booking features and payment options. The Driver Panel consists of login, shipper details, and update location feature. Finally the Admin Panel consists of driver monitoring dashboard feature; add vehicles and drivers feature and vehicle management option.

This application helps logistics professionals achieve their business process from tracking inventory and to shipments. It also helps any individual user by providing transport facilities for their goods from one place to another place. For any business officials or an individual user, this app provides a platform to collaborate with better tracking and shipment facilities. Logistics and supply chain management have always been at the forefront of industrial innovations, and numerous technologies have been tested and applied with success to increase efficiency of processes, reduce costs, or improve the relationships among different actors, such as buyers, sellers, retailers and carriers. In recent times, there is a great debate in both practitioners and academic' worlds about the impacts of these new technologies on the traditional logistic operations. The perspective of extant literature on technology adoption in logistics is quite fragmented: in fact, a stream of literature studies how a technology might provide improvements in logistics processes (e.g. warehouse management, goods distribution) and/or logistics management objectives (e.g. warehouse efficiency), while other research works focus on the impact of technology adoption in terms of overall business performance. Moreover, despite the amount of articles, reports and news regarding innovations in logistics, one of the last literature review about existing technologies applied in logistics has been published almost ten years ago and it is limited only to ICT technologies. Therefore, we reckon the need for an updated view on the literature in this field. Through a systematic literature review, we try to give an answer to the following research questions: i) how the interest in this topic changed in the last ten years from the point of view of scientific literature; ii) what are the main approaches and methodologies used in addressing this issue; iii) what are the most applied innovative technologies in the logistics field.

2. Previous Work

Logistics is an important part in many service industries. But as the industries expand and the network grows complex distribution and transport is the main problem to be dealt with most efficiently. The inventory management is another aspect that needs proper planning. Also synchronization between the retailer/vendor/supplier and the delivery management system and the customer is another difficulty. With the proposed logistics management system those problems were promisingly dealt with. It is an application to manage an optimized solution for local logistics and transportation needs for both businesses and consumers in the urban areas. Logistics is one of the important aspects of the service and the manufacturing industry. Timely delivery of goods and products to appropriate customer is necessary. The prevalent logistics systems are efficient but not effective enough. There are multiple drawbacks which eventually lead to the loss of the customer. The proposed logistics management system aimed to achieve an integrated logistics system which entirely controls the logistics and also controls the delivery of the product to the said destination with care on time. The system also comes with a comprehensive route tracker which helps to find the optimal route for the delivery of products and also keeps a track of the delivery status. Thus, the proposed system helps the vendor and the customer to deal with the goods reliably. The proposed system had a comprehensive logistics management app where the vendor or the supplier can register and provide the details of the customer and the shipping details. Once registered the system searches and hires the nearest transport service in the vicinity. It forwards the customer details and also calculates the optimal path for the delivery. The system tracks the delivery and also collects and analyses the customer feedback. The Logistics Management System app helps the vendor and the customer by providing the vendor an easy to use facility to ensure proper and timely delivery of its product and the customer the assurance of delivery. Thus it is a win-win situation for everyone. Also the path finder and receipt generation feature makes this app more effective and efficient. Using previous work on the concept as a source of inspiration. In their research paper, the system had a comprehensive logistics management app where the vendor or the supplier can register and provide the details of the customer and the shipping details. Once registered the system searches and hires the nearest transport service in the vicinity. It forwards the customer details and also calculates the optimal path for the delivery. The system tracks the delivery and also collects and analyses the customer feedback.

3. Literature Review

Logistics and supply chain management have always been at the forefront of industrial innovations, and numerous technologies have been tested and applied with success to increase efficiency of processes, reduce costs, or improve the relationships among different actors, such as buyers, sellers, retailers and carriers. In recent times, there is a great debate in both practitioners and academic' worlds about the impacts of these new technologies on the traditional logistic operations. [**Innovative Technologies adopted in Logistics Management**].

Sl. No.	Article Title	Author Name	Publisher and Year	Article Summery
1.	A social network analysis of business Logistics and transportation	Philips, D.M. and Philips, J.K.	International Journal of Physical Distribution and Logistics Management. (1998)	Introduces social network analysis techniques to business logistics and transportation. The case study has two specific goals. First, it introduces social network analysis techniques to the business logistics and transportation community as a useful tool with which to study the dynamic flows of communication between members of a social network. Second, it describes a wide variety of techniques and then utilizes them to examine artifacts of scholarly communication - journal citations. In doing so, it tracks the changing communication patterns across two separate time periods to describe the evolution and maturation of the

				fields of business logistics and transportation. Concludes that over a period of ten years the flow of information between the journals in the area of business logistics and transportation has become more efficient and that journals directly communicate with one another. Also, there is no longer a distinct break between logistics and transportation.
2.	Developing an E-Logistics System: A Case Study	Prof. Angappa Gunasekaran	International Journal of Logistics Systems and Management	Third-party logistics (3PL), a relatively new industry, has gained momentum since the emergence of global market and the Internet, in particular electronic commerce (e-commerce). Global competitive-ness places more pressure on companies to improve their delivery performance of products and services to customers. In an effort to improve the quality of delivery service, companies have outsourced their logistics services, including packing, warehousing (inventory management) and shipping of goods to customers. Communication plays an

				<p>important role in integrating the activities along the logistics value chain. Information technologies such as electronic data interchange (EDI), the Internet, World Wide Web (WWW) and e-commerce have contributed greatly to improving communication with partners in the logistics chain. In particular, real-time information systems such as web-based logistics information systems help to improve 3PL services. In this paper, a case study of e-logistics is used to illustrate the implications of information technology, in particular the Internet, WWW and EDI, on the performance of the logistics value chain. A framework based on the literature survey and case study is proposed to help companies develop an e-logistics system to improve their competitiveness</p>
3.	The integrated logistics management system: a framework and case study	Huan Neng Chiu	National Taiwan Institute of Technology, Taipei, Taiwan, Republic of China	<p>Presents a framework for distribution companies to establish and improve their logistics systems continuously. Recently, much attention has been given to automation in</p>

				<p>services, the use of new information technology and the integration of the supply chain. Discusses these areas, which have great potential to increase logistics productivity and provide customers with high level service. The exploration of each area is enriched with Taiwanese logistics management practices and experiences. Includes a case study of one prominent food processor and retailer in Taiwan in order to demonstrate the pragmatic operations of the integrated logistics management system. Also, a survey of 45 Taiwanese retailers was conducted to investigate the extent of logistics management in Taiwan. Concludes by suggesting how distribution companies can overcome noticeable logistics management barriers, build store automation systems, and follow the key steps to logistics success.</p>
4.	The impact of the logistics management in	Amine GHOUMRASSI and Gabriela	Proceedings of the 11th International	Logistics management is one of nowadays tools to face economic challenges; it's a mix

	customer satisfaction	<p>ȚIGU</p> <p>The Bucharest University of Economics studies, Bucharest, Romania</p>	<p>Conference on Business Excellence 2017</p>	<p>of business and core activities of the organization. The supply and distribution activities integrated together form what's known as logistics activities. The logistics activities within a business organization attempt to satisfy customers through achieving the time and location related market challenges and also through the cost of the service provided as well as the quality, taking into consideration customers needs and purchase power. Customer satisfaction is important because it provides marketers and business owners with a metric that they can use to manage and improve their businesses. Customer satisfaction is also a way to determinate the continuity of the business or of a product life by measuring the loyalty of the customers. If the customers are happy and satisfied, it will ensure the continuity of sales which means the continuity of the business. In the past customer satisfaction was more focused on requirements such</p>
--	-----------------------	--	---	--

				<p>as quality and reliability reducing costs of poor quality. In mid 50's the production costs were continuously increasing, The way to maintain the company's position within a changing market and increase profit starts by focusing on the service provided to the customer and on decreasing the cost, logistics activities became the backbone of these organizations that target the customer satisfaction while achieving competitive advantage. This study aims to show the impact of the logistics management on customer satisfaction in small and mid-sized Algerian industrial companies, by interviewing the companies managers and everybody in charge of the logistic process, the interview questions will be based on some literature review issues.</p>
5.	Analysis of the Logistics Research in India – White Paper	C. Thallera , N. Moraitakisb , H. Rogersc , D. Sigged , U. Clausena , H.-C. Pfohlb , E.	the German Federal Ministry of Education and Research (BMBF)	The project's main goal was to identify, analyze and describe the scientific and cooperative research in the field of logistics and supply chain management in India. The study provides a

		Hartmann , B. Hellingrathd		comprehensive overview of the current state of the art in the logistics research sector and helps identify opportunities for academic and industrial research cooperation. An overarching purpose of this study was the identification of relevant logistics clusters in India and the exploration and initiation of new cooperations, as only these can provide the prerequisite for our countries to promote and build on bilateral exchange of knowledge – at both an international and intercultural level. A common platform for exchange of experience can thus facilitate an effective mutual transfer of knowledge between partners in India and Germany. In terms of specific goals we identify, analyze and describe the scientific and cooperative research in the field of logistics and supply chain management in India. Further, we identified current trends in Indian logistics research.
6.	Logistics Management	Yenchun Jim Wu, National	The International	The purpose of this paper is to investigate the state of logistics

	Research Collaboration in Asia	Taiwan Normal University. Chih-Hung Yuan, University of Electronic Science and Technology of China, Zhongshan Institute	Journal of Logistics Management January 2017	management research in Asia. The study focuses on the research agenda, the topics of interest, and the extent of research collaboration in logistics theory building and knowledge specific to Asia. Design/methodology/approach This study uses a mixed methods approach namely, content analysis drawn from the articles found in six well- recognized peer-reviewed logistics management related journals from 2003 to 2013, followed by social network analysis which is applied on the selected articles to provide a structure of the collaboration relationship. Findings Initial findings suggest that there are some scholars in Asia who are instrumental in research collaboration and in building a body of knowledge on logistics management focused on Asia. More co-production of knowledge from deeper and tightly knit industry-academic collaboration is needed to progress this domain. Most of the published works use an
--	--------------------------------------	--	---	--

				<p>empirical instrument drawn from the resource-based view to explore firm level supply chain collaboration and strategy. This suggests a positivist research tradition within logistics. There is a shortage of studies conducted on the supply chain as a network of enterprises.</p> <p>Research limitations/implications The review of the articles is limited to six logistics specific journals and the authors only concentrate on logistics management research focused on Asia. The contributions from the other journals may have been missed. More collaboration at the institutional, national, and international levels is called for especially on cross-collaboration between practice and theory. Practical implications Though the analysis is restricted to 260 articles found in six journals, this paper can shed light on the research needs from different perspectives and facilitate the progress of logistics</p>
--	--	--	--	--

				<p>management research in Asia. Originality/value This is the first paper to discuss the state of logistics management research collaboration in Asia, and provides an overview of the research issues, topics, and approaches undertaken thus far. Through this work, this study hopes that it will encourage greater research collaboration between industry and academia, and academics themselves.</p>
7	Logistics management and supply chain management: A critical evaluation	<p>Shlomo Globerson , Gal Wolbrum², School of Business, Tel Aviv University, Tel Aviv, Israel, Azrieli College of Engineering, Jerusalem, Israel</p> <p>²Maccabi Health Care, Tel Aviv, Israel</p>	International Journal of Business and Economics Research	<p>The purpose of this research is to identify the content of Logistics Management and Supply Chain Management, as perceived by course instructors and textbook content, and identify the core subjects of the discipline. The study is based on review and content analysis of two samples; 30 syllabi used for teaching a basic course, and 10 textbooks that are dedicated to the discipline. A list of belonging subjects was identified, importance for each subject was estimated, and the level of agreement between the syllabi and the textbook samples was evaluated. Results</p>

				<p>identified 27 relevant subjects, of which eight were identified as "core subjects". The core subjects are: Information and technology, Inventory, Overview and concepts, Transportation and conveying, Logistics flow, Facility location, Customer service, and Performance measurement. There is a high level of agreement within each sample and between the two samples, with regard to the relative importance of the identified subjects. The above results support an agreed framework for a Logistics Management course, which becomes a common one in different management programs.</p>
--	--	--	--	---

Logistics encompasses a complex set of activities which require a collection of metrics to adequately measure performance. Ideally, the performance metrics used should be selected and maintained as a system, so they complement and support each other and provide the decision makers with a well-balanced picture of the logistics process. Often in practices, however, measurement systems are not formally managed or evaluated. The result is a performance measurement “system” where the interrelations between the metrics are not known, duplication is frequent, and omission is undetectable. This paper addresses this shortcoming by developing a set of evaluation criteria for logistics performance measurement systems and applying it in two cases studies. **[A Review and Evaluation of Logistics Performance Measurement System].**

Today, logistics management requires a comprehensive set of performance indicators that measure both tangible assets and intellectual capital (IC) of organizations. Nevertheless, most of the measures used in the past mainly related to the financial aspect, although some specific components of IC, such as process efficiency and effectiveness, have been considered. Logistics literature lacks a comprehensive consideration of the diverse IC measures, and it is unclear which area of IC requires more focus and development. **[Performance Measures of Logistics Management: an Intellectual Capital Perspective]**.

The scope of the field has broadened bringing new challenges for researchers and managers, but the broader scope was envisioned from its formation. Improvements in information technology and the just-in-time philosophy are the principal drivers for realizing the potential of boundary-spanning channel management. **[The evolution and future of logistics and supply chain management]**.

The large outsourcing and refocusing movement, regarding the key skills, initiated by many companies, has made a new profession emerge: the one of the logistics service providers. The logistics service providers, along the multi-actor Supply Chains, are considered as real pilots of the interfaces and represent a radical innovation on the managerial, strategically and operational plan. **[The logistics function to the logistics service: A literature review]**.

4. Requirement Analysis

The chosen technologies for the development of this application are Android Application Development, Firebase for real-time database. The programming language used is JAVA, XML and Firebase.

Logistics management System needed the following logistics and transportation app solutions:

(I) Features to manage fleet that can record all the data related to fleet management. (II) Features for warehouse for warehousing management. (III) Tracking Features for tracking the location of driver. One can get real-time updates about delivery and route of fleet.

4.1 Advantages of Android Development: It is an open-source Operating System that possesses a vast community for support. It is enhanced with a feature called fragmentation that means the application can run two activities on a single screen. Releasing the Android application in the Google play store is easier compared to other platforms.

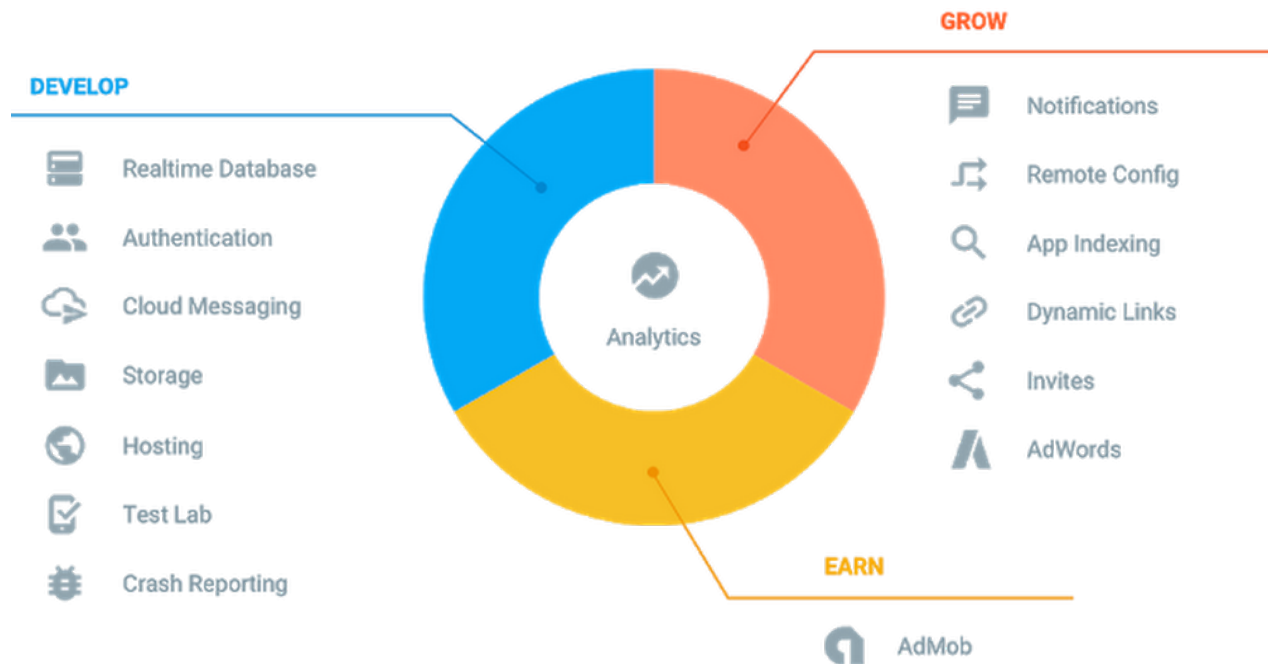
- **Android Application Development or Mobile App Development** is the process of creating software applications that run on a mobile device, and a typical mobile application utilizes a network connection to work with remote computing resources.

Fundamentals of Android Development:

1. **Activities:** An application that has a visible user interface is implemented via an activity. When one select an application from the Home screen or application launcher, an activity is started.
2. **Services:** One can use a service for any application that needs to persist for a long time such as network monitor or update checking application.
3. **Content providers:** The easiest way to think about content provider is to view them as a database server. A content provider's job is to manage access to persisted data, such as the contacts on a phone
4. **Broadcast receivers:** One can launch an Android Application to process a specific element of data or respond to an event, such as receiving a text message.

- **JAVA Environment**

- **Firestore Server:** Firestore is the Real-time Backend for an application. It provides –Real-time database, Authentication, Cloud Messaging, Storage, Hosting, Remote Configuration, Test Lab, Crash Reporting.



- **Adobe Xd:** Adobe XD was used to design logos, buttons and background. It provides features to develop drawables that are necessary to develop backgrounds that could fit in to the screen of any size. ldpi, mdpi, udpi, xhdpi, xxhdpi and xxxhdpi are the set of sizes that is used to achieve flexibility and portability in implementing user interface background.
- **Languages:** JAVA, XML
- **Hardware Requirement:** 8 GB RAM

4.2 Configure build

The Android build system compiles app resources and source code, and packages them into APKs that you can test, deploy, sign, and distribute. Android Studio uses Gradle, an advanced build toolkit, to automate and manage the build process, while allowing you to define flexible custom build configurations. Each build configuration can define its own set of code and resources, while reusing the parts common to all versions of your app. The Android plugin for Gradle works with the build toolkit to provide processes and configurable settings that are specific to building and testing Android applications.

Gradle and the Android plugin run independent of Android Studio. This means that you can build your Android apps from within Android Studio, the command line on your machine, or on machines where Android Studio is not installed (such as continuous integration servers). If you are not using Android Studio, you can learn how to build and run your app from the command line. The output of the build is the same whether you are building a project from the command line, on a remote machine, or using Android Studio.

The flexibility of the Android build system enables you to perform custom build configurations without modifying your app's core source files. This section helps you understand how the Android build system works, and how it can help you customize and automate multiple build configurations. If you simply want to learn more about deploying your app, see [Building and Running from Android Studio](#). To start creating custom build configurations right away using Android Studio, see [Configuring Build Variants](#).

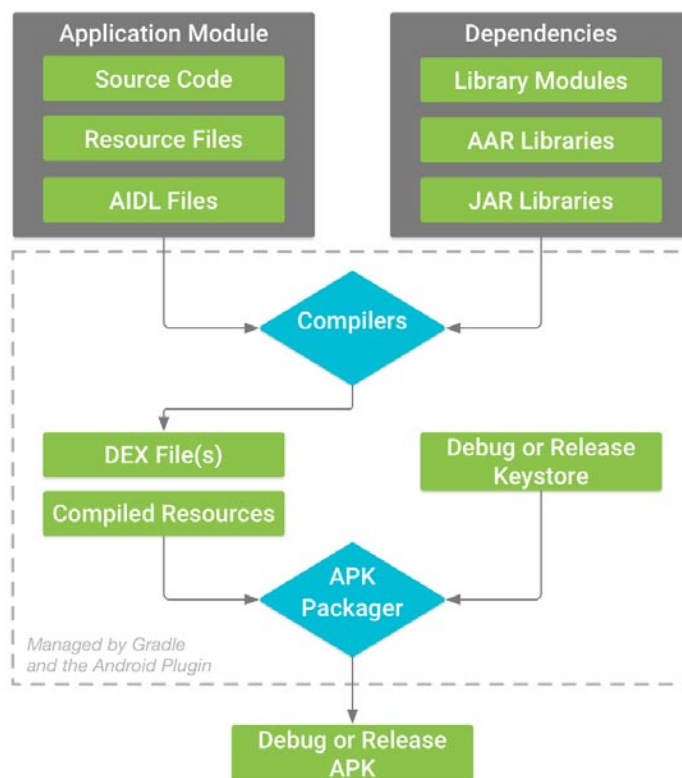
The build process involves many tools and processes that convert your project into an Android Application Package (APK). The build process is very flexible, so it's useful to understand some of what is happening under the hood.

The build process for a typical Android app module, as shown in figure 1, follows these general steps:

1. The compilers convert your source code into DEX (Dalvik Executable) files, which include the bytecode that runs on Android devices, and everything else into compiled resources.

2. The APK Packager combines the DEX files and compiled resources into a single APK. Before your app can be installed and deployed onto an Android device, however, the APK must be signed.
3. The APK Packager signs your APK using either the debug or release keystore:
 - a. If you are building a debug version of your app, that is, an app you intend only for testing and profiling, the packager signs your app with the debug keystore. Android Studio automatically configures new projects with a debug keystore.
 - b. If you are building a release version of your app that you intend to release externally, the packager signs your app with the release keystore. To create a release keystore, read about signing your app in Android Studio.
4. Before generating your final APK, the packager uses the zipalign tool to optimize your app to use less memory when running on a device.

At the end of the build process, you have either a debug APK or release APK of your app that you can use to deploy, test, or release to external users.



4.3 Connect to Firebase

Firebase is a mobile platform that helps you quickly develop high-quality apps, grow your user base, and earn more money. Firebase is made up of complementary features that you can mix-and-match to fit your needs, with Google Analytics for Firebase at the core. You can explore and integrate Firebase services in your app directly from Android Studio using the **Assistant** window shown in figure.

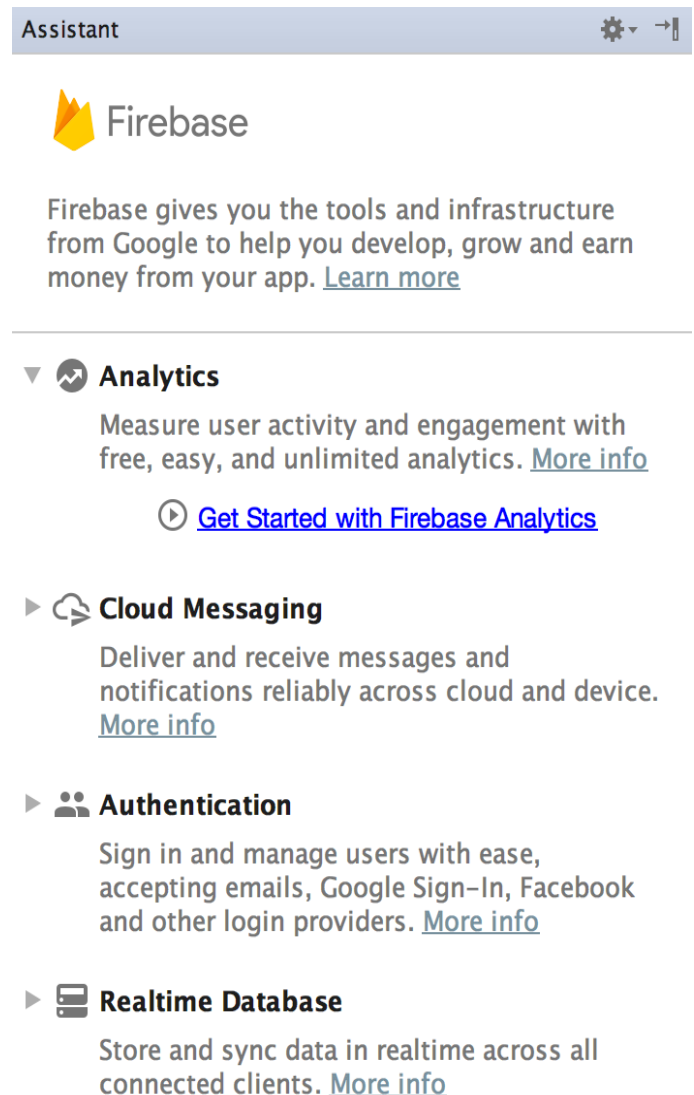
First make sure you have installed Google Repository version 26 or higher, using the following steps:

1. Click **Tools > SDK Manager**.
2. Click the **SDK Tools** tab.
3. Check the **Google Repository** checkbox, and click **OK**.
4. Click **OK** to install.
5. Click **Background** to complete the installation in the background, or wait for the installation to complete and click **Finish**.

One can now open and use the **Assistant** window in Android Studio by following these steps:

1. Click **Tools > Firebase** to open the **Assistant** window.
2. Click to expand one of the listed features (for example, Analytics), then click the **Get Started** tutorial to connect to

Firebase and add the necessary code to your app.



4.4 Custom build configurations

Gradle and the Android plugin help you configure the following aspects of your build:

Build types: Build types define certain properties that Gradle uses when building and packaging your app, and are typically configured for different stages of your development lifecycle. For example, the debug build type enables debug options and signs the APK with the debug key, while the release build type may shrink, obfuscate, and sign your APK with a release key for distribution. You must define at least one build type in order to build your app—Android Studio creates the debug and release build types by default. To start customizing packaging settings for your app, learn how to [Configure build types](#).

Product flavors: Product flavors represent different versions of your app that you may release to users, such as free and paid versions of your app. You can customize product flavors to use different code and resources, while sharing and reusing the parts that are common to all versions of your app. Product flavors are optional and you must create them manually. To start creating different versions of your app, learn how to [Configure product flavors](#).

Build variants: A build variant is a cross product of a build type and product flavor, and is the configuration Gradle uses to build your app. Using build variants, you can build the debug version of your product flavors during development, or signed release versions of your product flavors for distribution. Although you do not configure build variants directly, you do configure the build types and product flavors that form them. Creating additional build types or product flavors also creates additional build variants. To learn how to create and manage build variants, read the [Configure build variants overview](#).

Manifest entries: You can specify values for some properties of the manifest file in the build variant configuration. These build values override the existing values in the manifest file. This is useful if you want to generate multiple APKs for your modules where each of the apk files has a different application name, minimum SDK version, or target SDK version. When multiple manifests are present, Gradle merges manifest settings.

Dependencies: The build system manages project dependencies from your local filesystem and from remote repositories. This prevents you from having to manually search, download, and copy binary packages of your dependencies into your project directory. To find out more, see [Add Build Dependencies](#).

Signing: The build system enables you to specify signing settings in the build configuration, and it can automatically sign your APKs during the build process. The build system signs the debug version with a default key and certificate using known credentials to avoid a password prompt at build time. The build system does not sign the release version unless you explicitly define a signing configuration for this build. If you do not have a release key, you can generate one as described in [Sign your app](#).

Code and resource shrinking: The build system enables you to specify a different ProGuard rules file for each build variant. When building your app, the build system applies the appropriate set of rules to shrink your code and resources using its built-in shrinking tools, such as R8.

Multiple APK support: The build system enables you to automatically build different APKs that each contain only the code and resources needed for a specific screen density or Application Binary Interface (ABI). For more information see [Build multiple APKs](#).

Build configuration files: Creating custom build configurations requires you to make changes to one or more build configuration files, or `build.gradle` files. These plain text files use Domain Specific Language (DSL) to describe and manipulate the build logic using Groovy, which is a dynamic language for the Java Virtual Machine (JVM). You don't need to know Groovy to start configuring your build because the Android plugin for Gradle introduces most of the DSL elements you need. To learn more about the Android plugin DSL, read the [DSL reference documentation](#).

When starting a new project, Android Studio automatically creates some of these files for you, as shown in [figure 2](#), and populates them based on sensible defaults.

4.5 7 Key Android Concepts

Android UI Controls

Android provides a number of standard UI controls that enable a rich user experience. Designers and developers should thoroughly understand all of these controls for the following reasons:

- They are faster to implement. It can take up to ten times longer to develop a custom control than to implement a user interface with standard Android controls.
- They ensure good performance. Custom controls rarely function as expected in their first implementation. By implementing standard controls, you can eliminate the need to test, revise and improve custom controls. Moreover, while designers will spend a great deal of time thinking about how a control should look, they may not always consider the many ways in which a custom control will behave in the user's hands. Items on a mobile device often need to grow and shrink in size as they are pinched, or scroll if they are part of a list. As a result, creating a “clean” custom control from scratch can take a significant amount of design and development time. Google, however, has already thought about these interactions and developed standard controls to properly address them.
- Android users expect standard controls. Through their interactions with other Android apps, users become accustomed to Android's standard controls. Deviating from the standard Android user experience can confuse and frustrate users, making them less likely to want to use your app and incorporate it into their daily activities.

With a solid awareness of Android's standard controls, designers and developers can speed app development while offering users an intuitive experience that feels instantly familiar.

Activities

Android applications are composed of “activities” which are unique, focused actions a user can take. Because it can be difficult or time-consuming to scroll, zoom in, or click links on a small screen, it is recommended that an app display only one activity per screen. This practice presents the user with only the most relevant information and allows them to launch a new screen for additional information, or click the “back” button to view the previous activity. While a screen can expose multiple tasks, it should help the user complete just one activity at a time.

In Gmail for example, a user can only read the body of an e-mail (right) once he has clicked the relevant message (left). This layout reduces the amount of information displayed on each screen and allows the user to easily navigate between the Inbox and the message text.

User Interactions

When a user first downloads your application, he will make snap judgments on the usability and intuitiveness of the application within the first few minutes of use. It is, therefore, crucial to balance the creativity of your app with the standard user interactions Android users have come to expect. These include:

- **Hard buttons:** including Back, Menu, Home and Search buttons. Soft buttons that duplicate these features will only confuse or frustrate Android users. Moreover, back button behavior can be tricky and needs to be defined up-front for every screen, as it is not always as simple as returning to the previous activities. Most mobile phones, for example, offer both an “incoming call” activity and an “active call” activity. Once a user has answered and completed the call, the user would not expect to return to the “incoming call” activity upon pressing the “back” button, but rather to the activity that occurred before the incoming call. If the app offers only one activity, the back button should return the user to the device’s home page.
- **Long press elements:** Items of a list can be long pressed to open a context menu that provides secondary information. “ToDo” list apps, for example, often use a touch interaction to mark a task as completed and a long press interaction to display a menu with “edit” or “delete” functionality.

Layouts

Android UI screens are frequently resized, both on the fly via pinch and zoom as well as at startup when Android adjusts the size of the UI to fit the screen size of the mobile device on which it’s running. In order to make the most of the screen size and handle this resizing gracefully, Android provides a number of screen layout options.

First, Android developers must specify whether each screen should follow a linear layout which manages controls in a horizontal or vertical fashion or a relative layout which manages controls in relation to one another. Linear layouts are the most common, as in the example below. At left,

the controls only stretch to accommodate the text and are positioned in a horizontal line. In the middle image, the same rules apply but in a vertical layout. At right, the vertical layout is maintained but the middle button stretches to accommodate the screen rather than the text.

A relative layout defines the position of controls by their relationship to other components on the same screen. In the example below from the droidcake.com blog, the “OK” button was specified to be set below the radio button group. The “Cancel” button was specified to be set to the right of the OK button with its right edge extended to the edge of the screen. This relative layout positioning ensures the position of the buttons remains constant across a variety of screen sizes.

Android also offers specific layout properties to control the way in which screen elements are displayed across Android devices and during use:

- **Weight:** The weight property allows the developer to determine how free space is divided on the screen.
- **Gravity:** Gravity is the term used for control alignment (right, bottom, top, or left) on an Android device.
- **Density independence:** Your application achieves “density independence” when it preserves the physical size (from the user’s point of view) of user interface elements displayed on screens with different densities. Without density independence, a UI element (such as a button) will appear larger on a low-density screen and smaller on a high-density screen.

So who specifies all of these properties?

If an Android application is designed in a vacuum and then “thrown over the wall” to the development team, you must rely on the developers’ interpretation of the design which may vary significantly from the original intent. On the other hand, the development team shouldn’t be expecting the designer to specify the weight, gravity and other layout properties of each screen and control.

In our experience, the best practice is to have the designer document the layout and resize behavior of each screen to the development team via a series of wireframes, if not a full style guide. The designer should then stay in close communication with the development team as the

developers work to determine the right combination of Android layout properties to realize the design.

Screen Size

A common misconception is that an Android app should be designed to support only a specific set of Android devices. Many teams assume their app will only look right on a screen of a particular screen size and limit their design to suit only a handful of devices supporting that size. In reality, Android offers you tools needed to develop a visually impressive interface that supports the full range of devices and screen sizes on the market.

To help you accommodate the range of Android screen sizes, Android recommends designing four versions of the application UI:

- A small version for screens under 3”.
- A normal version to accommodate 3” to 4.5” screens.
- A large version for viewing on 4.5” to 10” screens.
- An extra large version for devices with screens larger than 10” (tablet).

It is not strictly necessary to create a design for all four versions – in some cases; one “normal” and one “extra large” version may suffice. If, however, you need to display a large number of controls on your screen, or your organization wishes to ensure perfect consistency across screen sizes, you may decide to accommodate all four size categories listed above.

Fragments

A smart phone should only display one activity per screen due to its small screen size. Tablet devices, however, offer additional screen real estate and are often used in a similar setting as a desktop or notebook, meaning the application could show more information at once on the screen. Using an Android construct called fragments, designers and developers can merge portions of the UI onto one large screen or split them into individual screens for use on small screens. This can help to reduce the number of interactions a user must perform on a device with a large screen and eliminate wasted space.

The example below shows a Gmail interface on a tablet display. This design uses fragments to display both the navigation list at left and the Inbox content at right. The design reduces the number of screens that must load before the user reaches the desired message.

If you anticipate your app will someday be used on a tablet device, we strongly recommend you incorporate fragments into your design. Designers need to be aware of the concept of fragments in order to design by fragment, and developers also need to be aware of this concept and its implementation details.

By designing custom, reusable fragments for each screen activity at the beginning of the project, you can eliminate the need to create an entirely new layout for a tablet device.

Intents

Android applications typically borrow from other applications already on the device. Using intents you can simplify both the programming requirements for your app and offer simpler, less cluttered screens.

If your app needs to perform a function beyond its core abilities such as opening a photo, looking up a contact, or playing a video, the team should investigate whether a tool that can perform that function already exists in the OS or in a popular third-party app. If so, you can leverage that functionality using intents.

For example, if your app accesses user contacts, you can use intent objects to launch the device's existing Contacts application. This will eliminate programming duplication and speed up the user's interaction with the device since the user will not need to re-learn how to add a contact to your particular app.

Android offers specific UI controls, activities, interactions, layout and resize options, as well as special constructs like fragments and intents. While on the surface these appear to be things that the design team needs to work with, we contend that the entire team must be immersed in Android to coordinate design, workflow, and execution into a single, intuitive application — one that grabs users' attention and draws them into the real value of your product.

6. Proposed System Design and Working

The three panels are: Admin, Customer and Driver Panels.

1. Admin Panel contains the following features:

1. Login
2. Add Trucks
3. Add Drivers
4. View Truck Location Updates

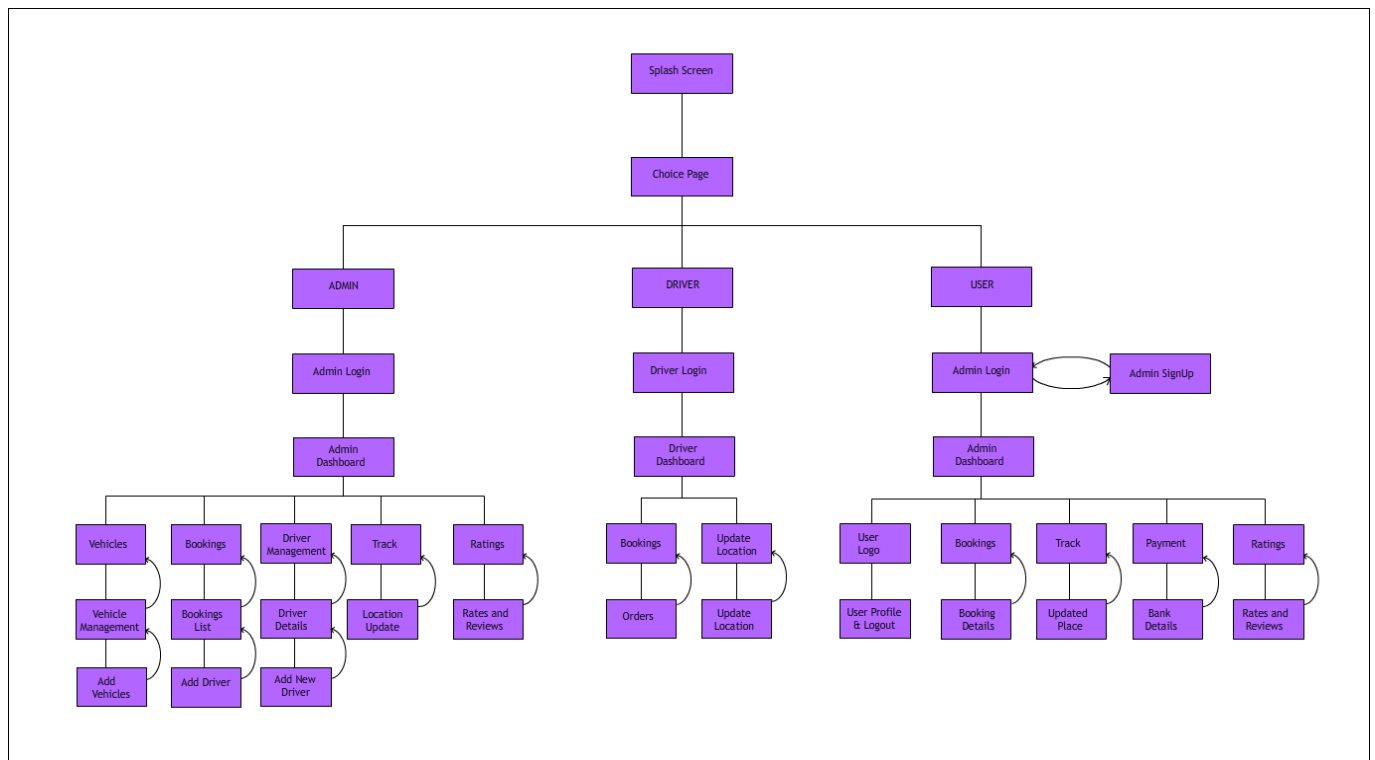
2. Driver Panel contains the following features:

1. Login
2. View Booking Details
3. Update Location

3. Customer Panel contains the following features:

1. Registration
2. Login
3. View Trucks
4. Book Trucks
5. Location Update

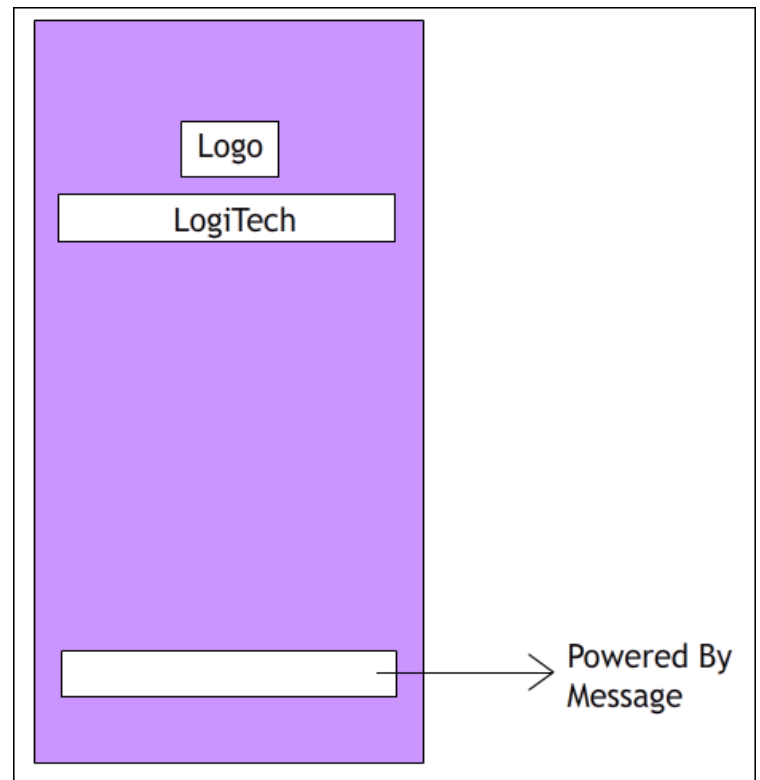
5.1 Architecture Diagram



5.2 Design of Proposed System

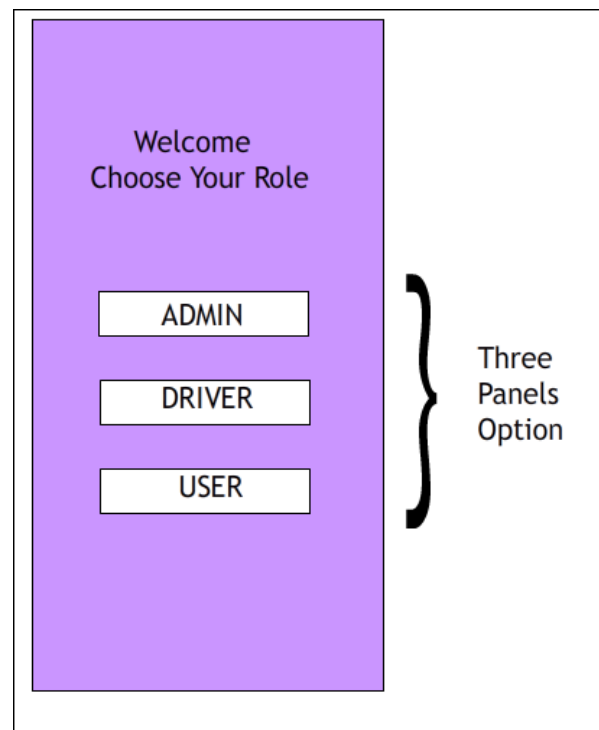
1. Splash Screen: Splash Screen runs for 5 seconds or 5000ms. It contains the logo, app name and powered by message along with year at the bottom [Powered by UP2-G69].

1.1 Logo:



2 Choice Pages:

The choice page consists of three buttons namely, ADMIN, DRIVER, USER/CUSTOMER. The user end actor should select their role accordingly.



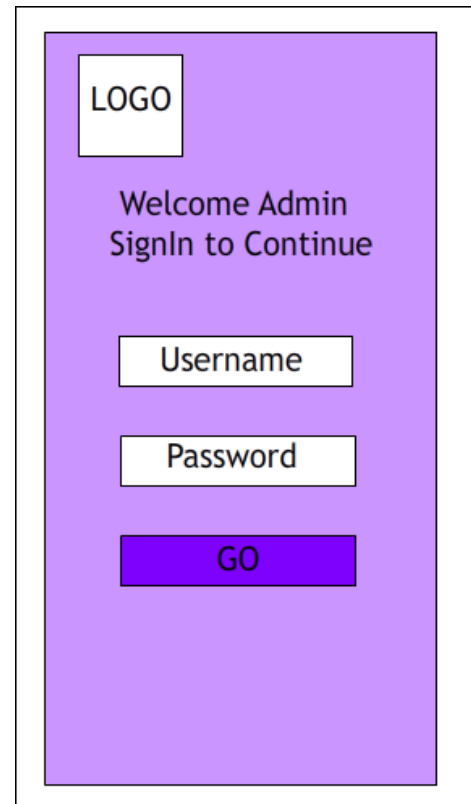
3 Admin

3.1 Admin Login:

Admin Login page has a logo and welcome message saying, “Welcome Admin SignIn to Continue”. It has two input fields, username and password. Admin has direct login feature. The username and password is inbuilt and had access only to the admin.

Username: UP2G69

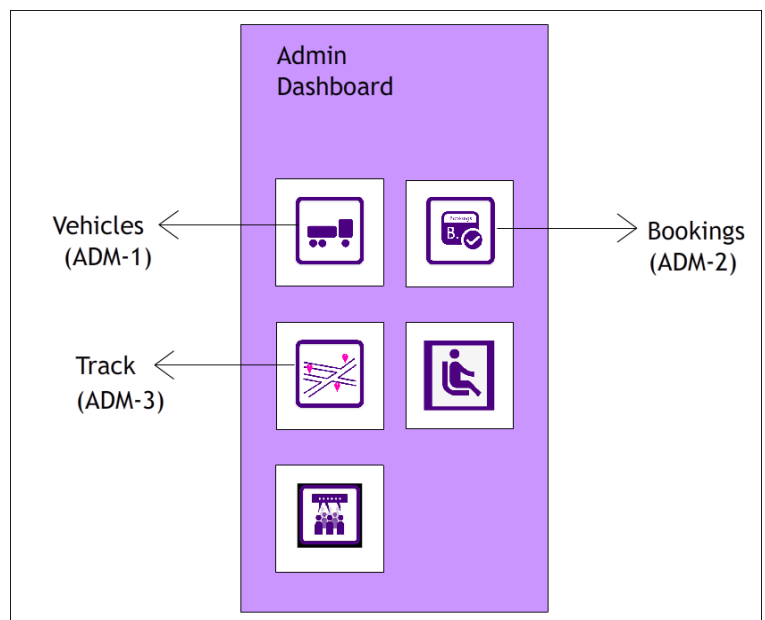
Password: admin123



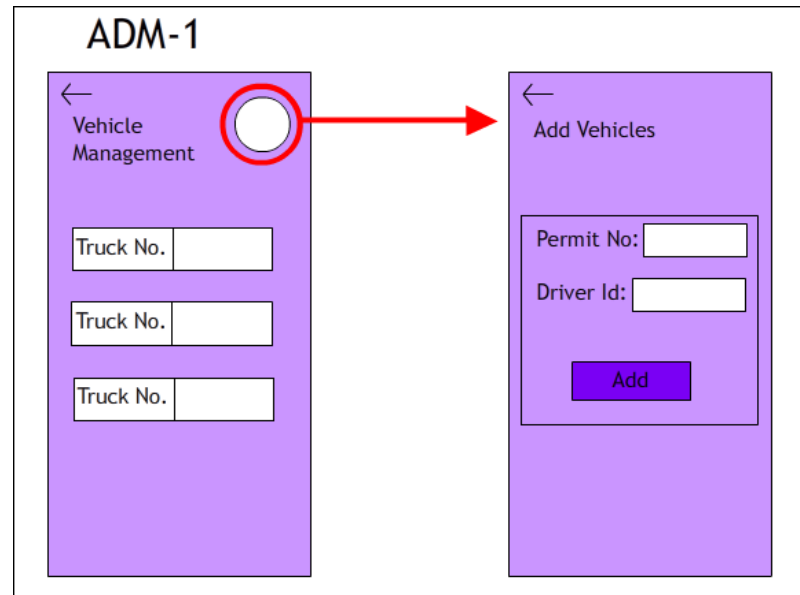
The Admin Login form is a vertical rectangle with a light blue background. At the top left is a white box labeled "LOGO". Below it, centered, is the text "Welcome Admin SignIn to Continue". Further down are two white input fields: "Username" and "Password". At the bottom is a red button labeled "GO".

3.2 Admin Dashboard: This page contains four button options namely: Vehicles, Bookings, Track and Rate and Review.

3.2.1 Vehicles: On clicking the Vehicles button, it navigates to Vehicle Management page.



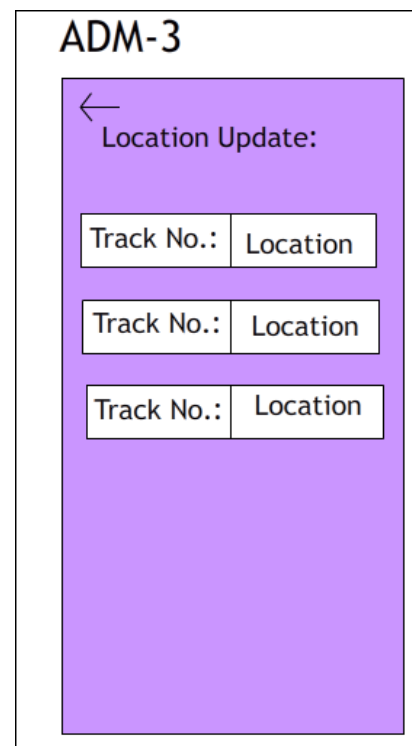
3.2.1.1 Vehicle Management: It contains the list of all trucks added previously by the admin. Every truck details include vehicle number, permit numbers and assigned driver id. On the top right of the page, there is an option to add new truck. On clicking that button it navigates to Add Vehicle page.



3.2.1.1.1 Add Vehicle: It provides two input fields namely, Permit No. and Driver Id. Which the driver have to manually enter and hit on the add button below. On clicking the add button a new truck will be added.

3.2.1.1.2 Track Updates: Location updated by the driver live is listed.

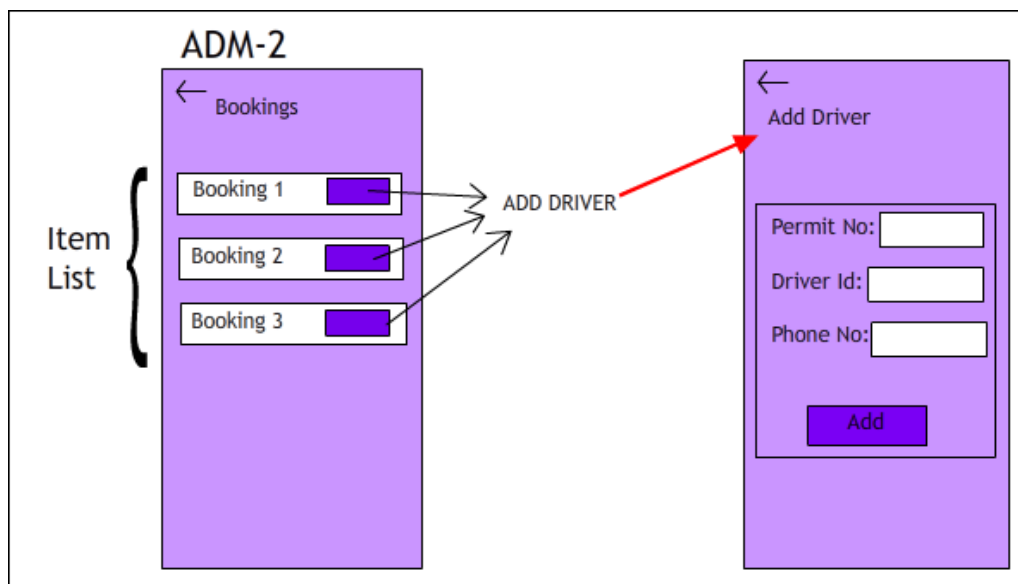
3.2.1.1.3 [Navigation can be achieved between Admin Dashboard and Add Vehicle pages]



3.2.2 Bookings: On clicking Booking Button, it navigates to Bookings page.

3.2.2.1 Bookings: It contains the list of booking from the customer. Admin will manually allocate the driver to the orders. On clicking the Add Driver button it navigates to Add Driver page.

3.2.2.2 Add Driver Page: It provides three input fields namely, Name, Driver Id, phone number. On clicking add, if driver added successfully the toast message pops up.



3.2.2.3 [Navigation can be achieved between Admin Dashboard and Add Driver pages]

3.3 Track: On clicking Track Button, it navigates to Track page.

3.3.1 Track Page: Location updated by the driver is listed with truck number in this page.

3.3.1.1 [Navigation can be achieved between Admin Dashboard and Location Update]

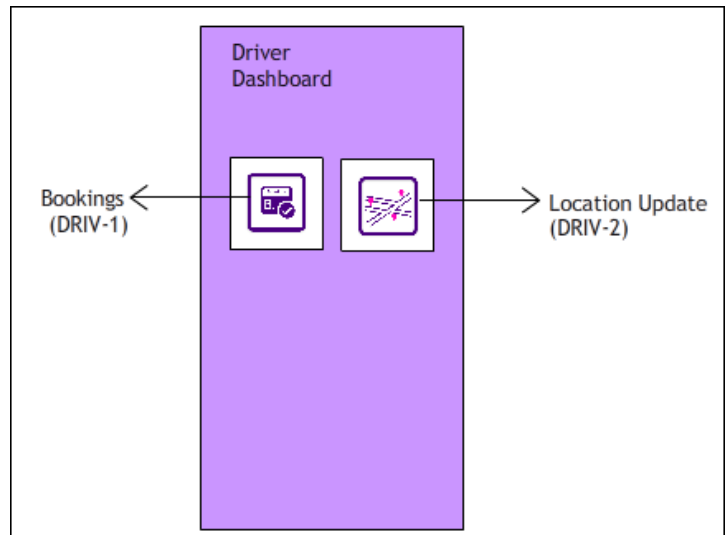
3.4 Rate and Reviews: On clicking Rates Button, it navigates to Rates and Reviews page.

3.4.1 Rates and Reviews: It contains the reviews given by customers.

4 Driver

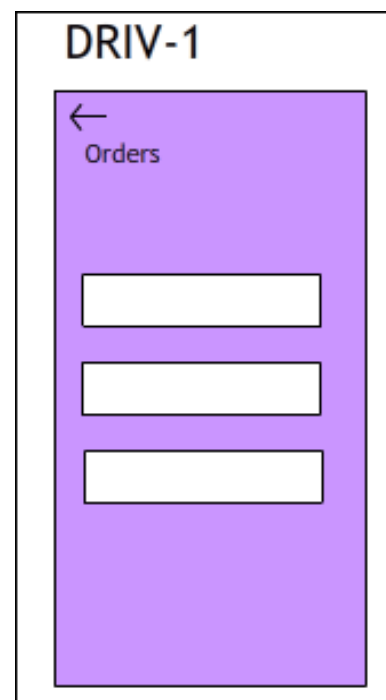
4.1 Driver Login: Driver Login page has a logo and welcome message saying, “Welcome Driver SignIn to Continue”. It has two input fields, username and password. Driver Login credentials are assigned by admin.

4.2 Driver Dashboard: This page contains two button options namely: Bookings and Update Location.



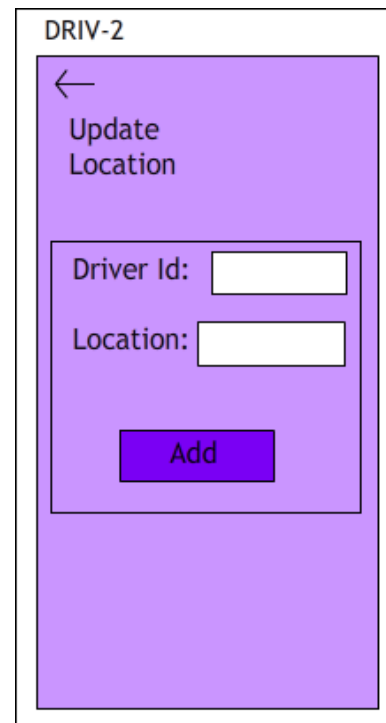
4.2.1 Bookings: On clicking Booking Button, it navigates to Orders page[DRIV-1].

4.2.1.1 Orders: Driver can view Booked Items in this page.



4.2.2. Location: On clicking Track Button, it navigates to Update Location page.

4.2.2.1 Update Location: Driver has to manually enter the current location here.



DRIVER-2

← Update Location

Driver Id:

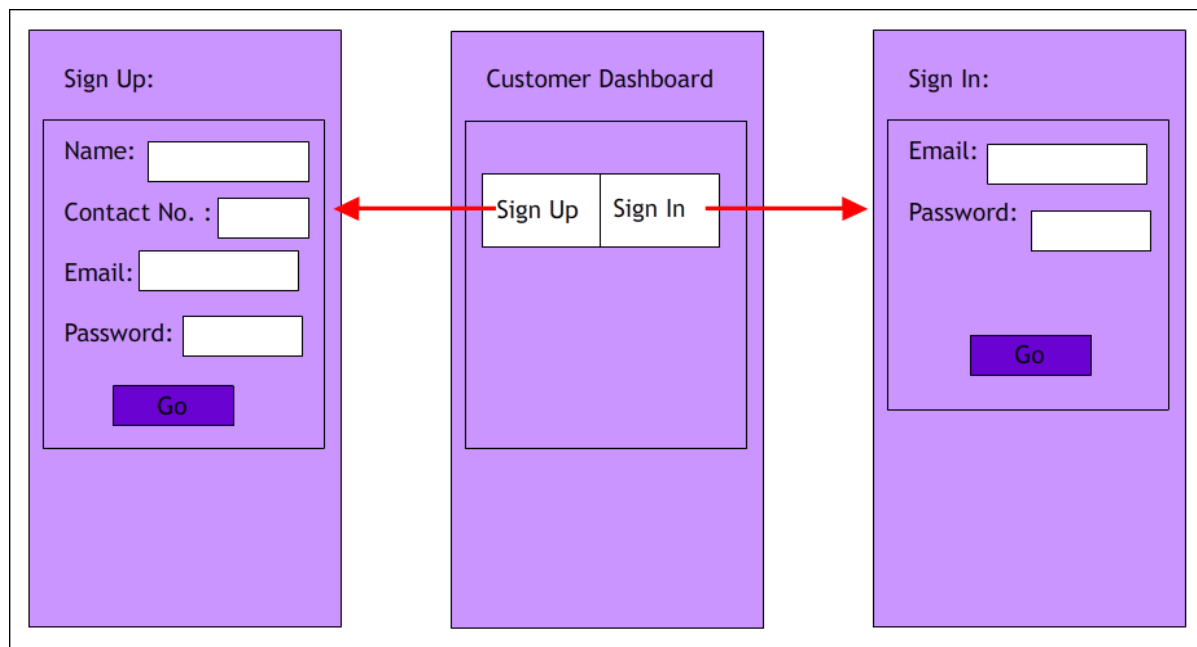
Location:

Add

5 Customers

5.1 Customer Login: Driver Login page has a logo and welcome message saying, “Welcome SignIn to Continue”. It has two input fields, username and password. If customer is a new user, there is an option to register ie., Sign-Up option.

5.2 Customer Sign-Up: It provides four input fields namely, username, contact number, email and password.



Sign Up:

Name:

Contact No. :

Email:

Password:

Go

Customer Dashboard

Sign Up Sign In

Sign In:

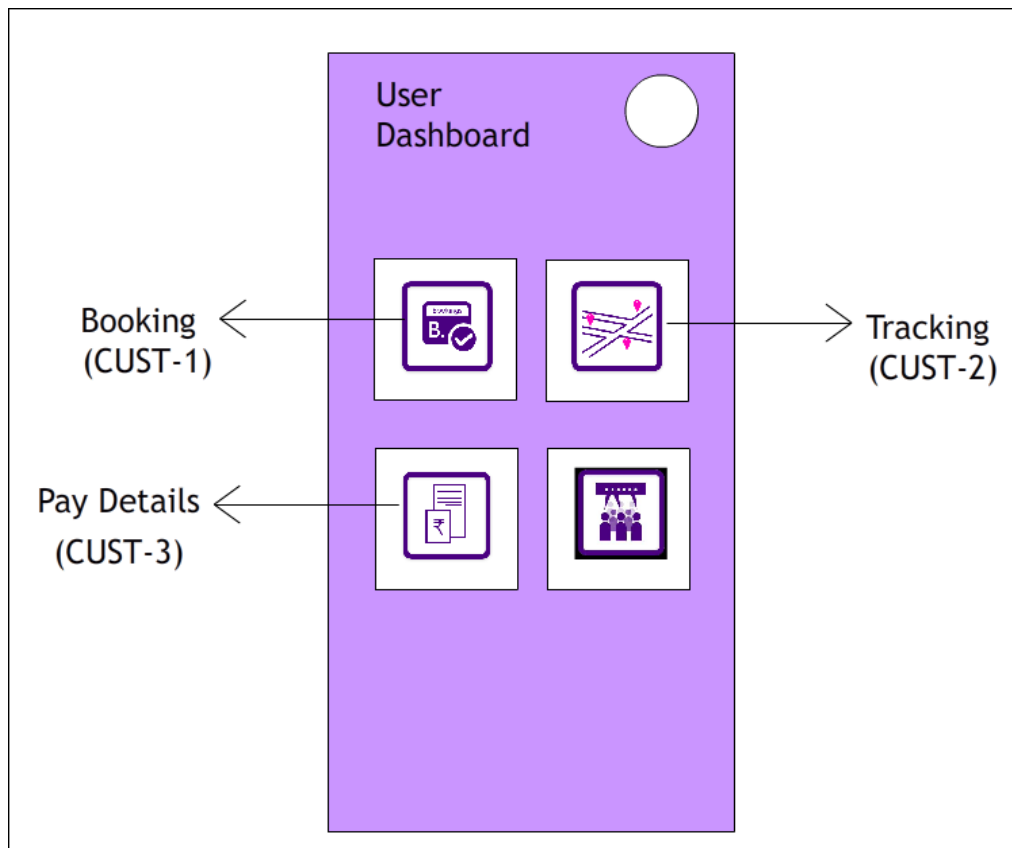
Email:

Password:

Go

Diagram illustrating the flow between three screens: Sign Up, Customer Dashboard, and Sign In. Red arrows indicate navigation from the Sign Up and Sign In screens to the Customer Dashboard.

5.3 Customer Dashboard: This page contains four button options namely: Bookings, Track, Pay and Rate and Review. On the right top it contains log-out button.



5.3.1 Bookings: On clicking Booking Button[C], it navigates to Bookings Details page.

5.3.1.1 Booking Details Page: It contains 3 input fields namely, To, From and Weight in Kgs/Tons.

5.3.1.1.1 To and From: It is a drop down menu which contains a list places where the boarding and shipping services are provided.

5.3.1.1.2 Weight: Customer have to enter estimated approximate weight, in kilograms or tons, of the good to be shipped.

5.3.1.2 [Navigation can be achieved between Customer Dashboard and Booking

The screenshot shows a mobile app interface for a customer (CUST-1) on the 'Booking Details' page. At the top, there is a back arrow and the title 'Booking Details'. Below this, there are three input fields: 'From:', 'To:', and 'Weightl:'. Each field has a corresponding text input box. To the right of the 'Weightl:' field, there is a label 'in Kg or Ton' with an arrow pointing to the input box. At the bottom of the form, there is a blue button labeled 'Book'.

5.3.1.2.1 Book Button: On clicking this button[C] toast message pops up if the booking is successful.

5.3.2 Track: On clicking Track Button[E], it navigates to Location Update page.

5.3.2.1 Update Location: It contains the updated location by the driver.

5.3.2.2 [Navigation can be achieved between Customer Dashboard and Update Location pages]

The screenshot shows a mobile app interface for a customer (CUST-2) on the 'Track' page. At the top, there is a back arrow and the title 'Track'. Below this, there is a large rectangular box containing the text 'Updated Place'.

5.3.3 Pay Button: On clicking this button[B] it takes to Bank Details page.

5.3.3.1 Bank Details: It contains certain banking information like, Name, Bank Name, Account Number, IFSE code and Phone Number.

CUST-3

←

Bank Details

Name:

Bank Name:

A/C No. :

IFSCE Code:

Phone No. :

5.3.3.2 [Navigation can be achieved between Customer Dashboard and Bank Details pages]

5.3.4 Rate and Reviews: On clicking Rates Button, it navigates to Rates and Reviews page.

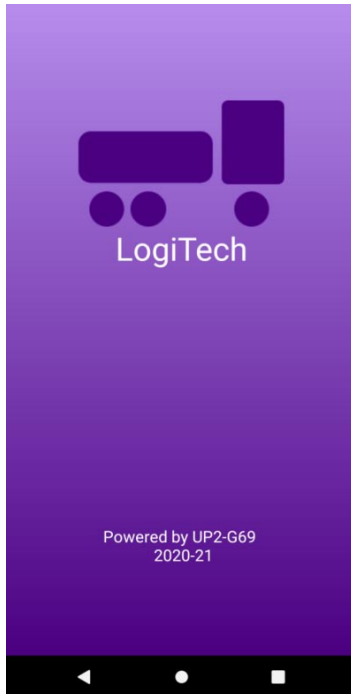
5.3.4.1 Rate and Reviews: Customer has to enter the rate in number out of 5.

5.3.4.2 [Navigation can be achieved between Customer Dashboard and Rates and Review pages]

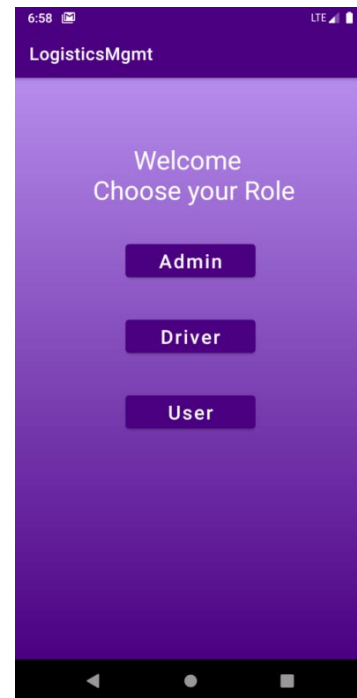
6. Implementation

1. Home

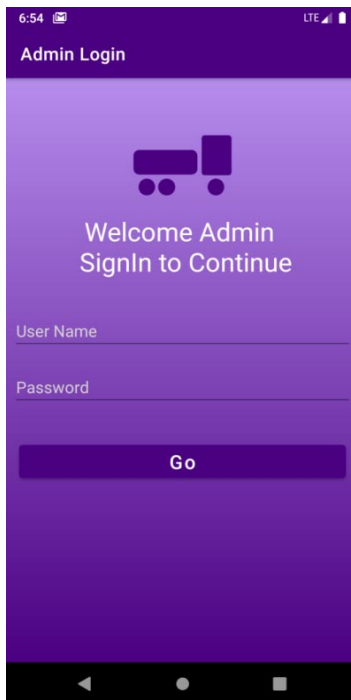
1.1 Splash Screen



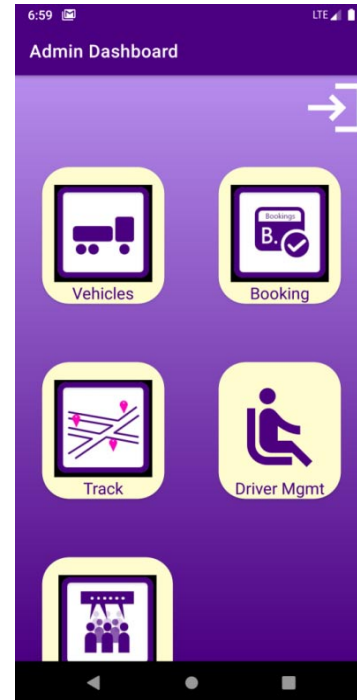
1.2 Choice Page



2.1 Admin Login

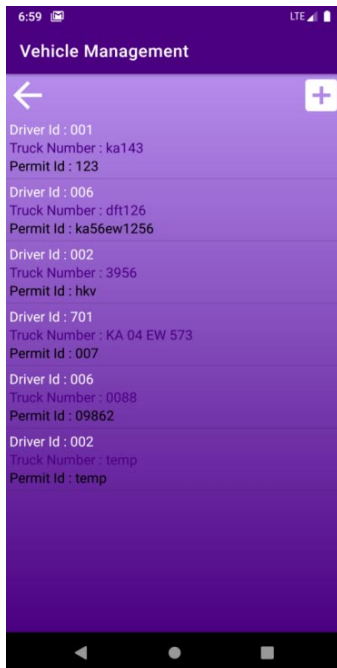


2.2 Admin Dashboard

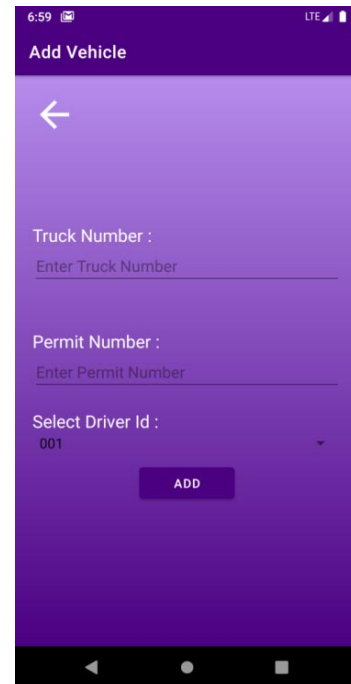


2.2.1 Vehicles

2.2.1.1 Vehicle Management

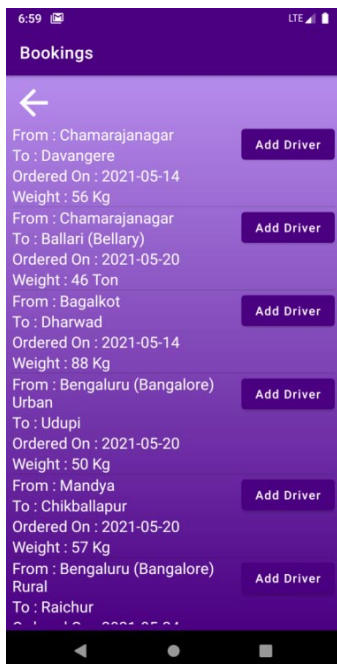


2.2.1.2 Add Vehicle

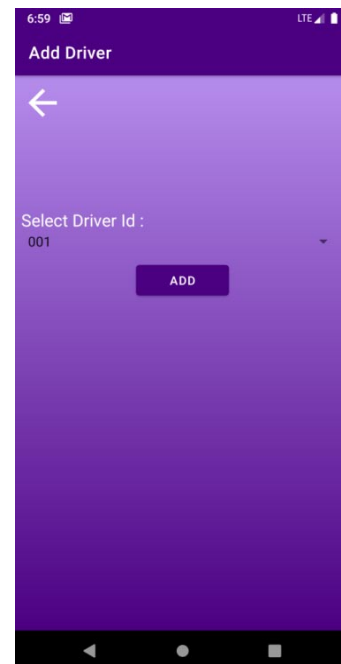


2.2.2 Bookings

2.2.2.1 Bookings

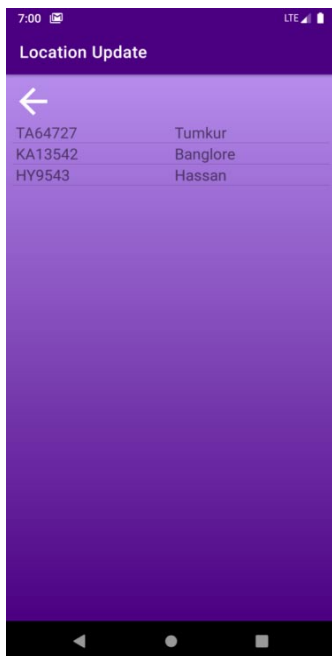


2.2.2.2 Add Driver Page



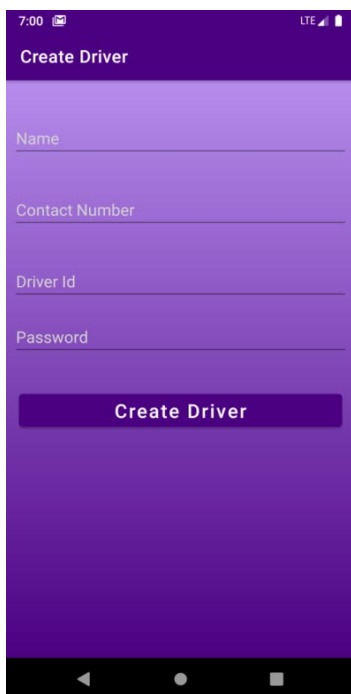
2.2.3 Track

2.2.3.1Track Page

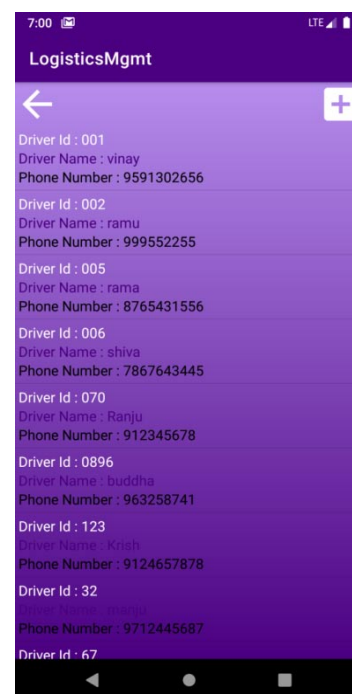


2.2.4 Driver Management

2.2.4.1 Driver Management

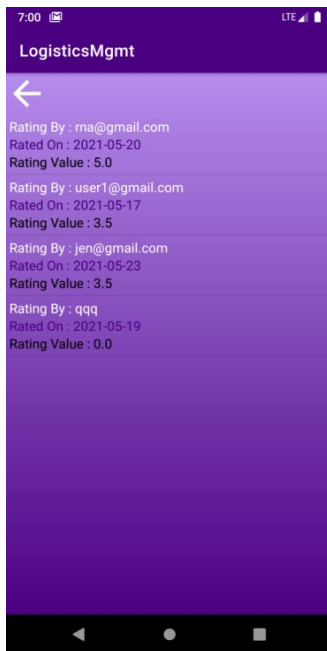


2.2.4.2 Create Driver



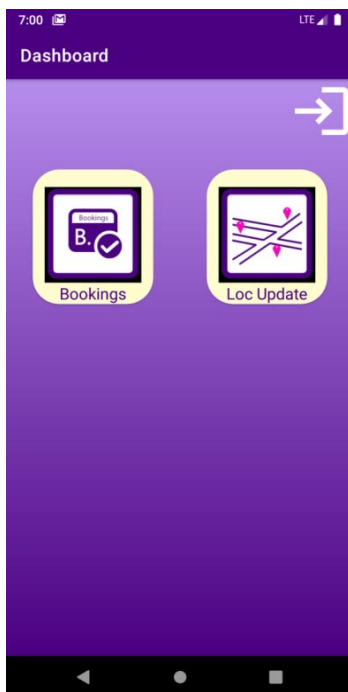
2.2.5 Ratings

2.2.5.1 Rates and Reviews

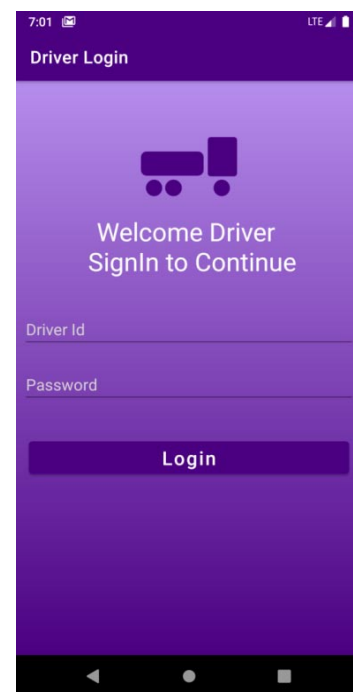


3.1 Driver

3.1 Driver Login

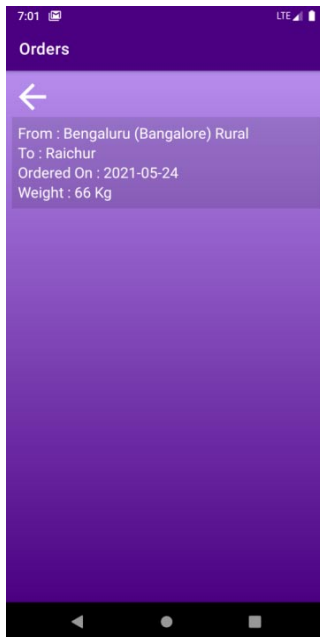


3.2 Driver Dashboard



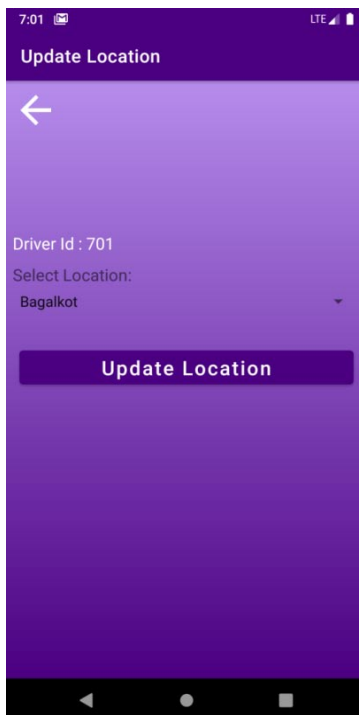
3.2.1 Bookings

3.3.1.1 Orders



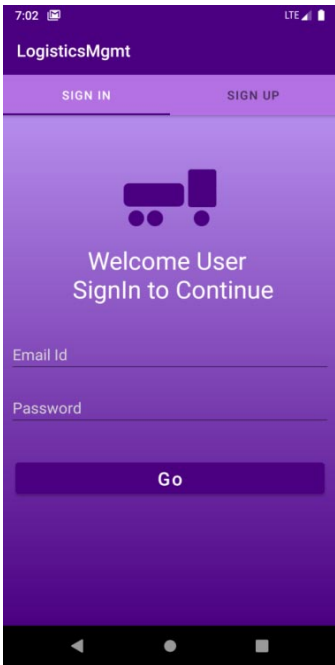
3.2.2 Location

3.2.2.1 Update Location

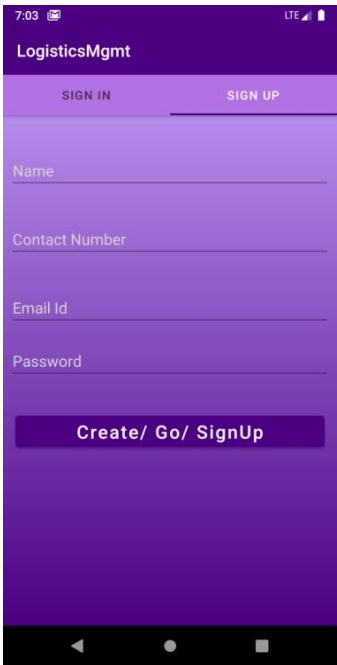


4. Customer

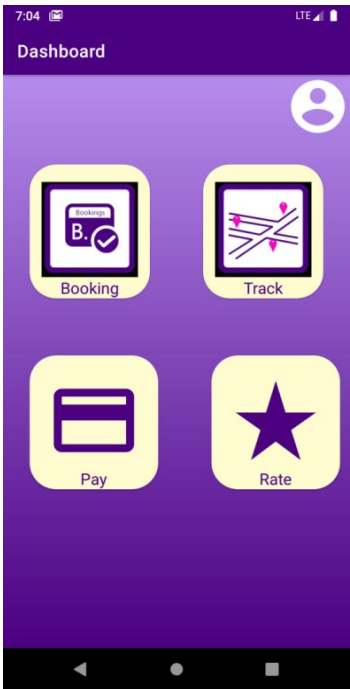
4.1 Customer Login



4.2 Customer Sign-Up

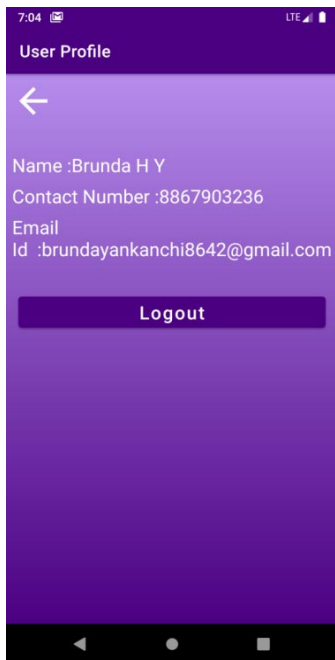


4.3 Customer Dashboard



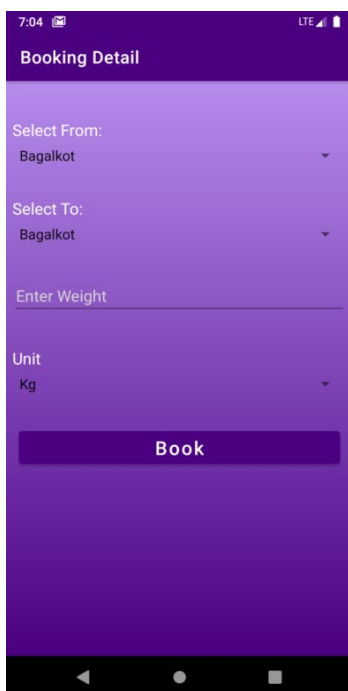
4.3.1 User Profile

4.3.1.1 User Profile Page



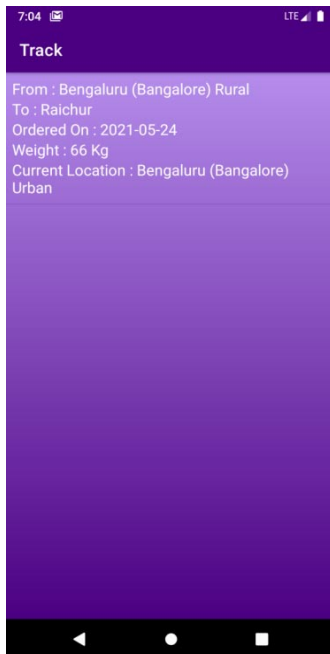
4.3.2Bookings

4.3.2.1 Booking Details Page



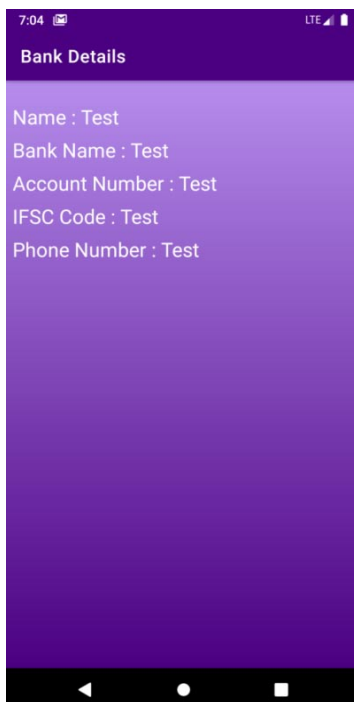
4.3.3 Track

4.3.3.1 Update Location



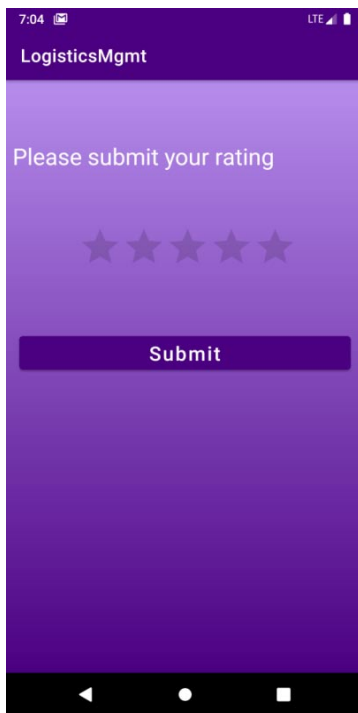
4.3.4 Payment

4.3.4.1 Bank Details



4.3.5 Ratings

4.3.5.1 Rate and Reviews



7. Verification and Testing

Test Case ID	HB_001	Test Case Description	Test the Home Page of the Logistic Management app		
Created By	Brunda H Y	Reviewed By	Rohan N, Brunda H Y	Version	
Tester's Name	Brunda H Y	Date Tested	May 3, 2021	Test Case (Pass/Fail/Not Executed)	Pass

<u>Test Scenario</u>	Verify on the Home Page of the app
-----------------------------	------------------------------------

S#	Step Details		Expected Result	Actual Result	Pass/Fail/Not executed /Suspended
1	Choose your role	Clicking Admin button	Opens the admin login page	Opens the admin login page	Pass
2		Clicking Driver button	Opens the driver login page	Opens the driver login page	Pass
3		Clicking User button	Opens the user signup and signin page	Opens the user signup and signin page	Pass

Test Case ID	HB_002	Test Case Description	Test the Admin Page of the Logistic Management app		
Created By	Brunda H Y	Reviewed By	Rohan N, Brunda H Y	Version	
Tester's Name	Brunda H Y	Date Tested	\May 3, 2021	Test Case (Pass/Fail/Not Executed)	Pass

<u>Test Scenario</u>	Verify on the Admin Page of the app
-----------------------------	-------------------------------------

S#	Step Details		Expected Result	Actual Result	Pass/Fail/Not executed /Suspended
1	login for the admin	valid username and valid password username:admin password: admin123	Login successful	Login successful	Pass
2		invalid username and valid password username:Admin in Password:admin123	username or password is wrong	username or password is wrong	Pass

3		invalid username and invalid password username:Adm in Password:Adm in123	username or password is wrong	username or password is wrong	Pass

S#	Step Details		Expected Result	Actual Result	Pass/Fail/Not executed /Suspended
4	login for the admin	valid username and invalid password username:admin Password:Adm in123	username or password is wrong	username or password is wrong	Pass

S#	Step Details		Expected Result	Actual Result	Pass/Fail/Not executed /Suspended
5	Create Driver	Name: shiva Contact Number:786764344 5 Driver Id:006 Password: shiva@123	Document created/updated	Document created/updated	Pass

6	Driver Management after adding	in the driver manegement list the above data is updated Driver Id:006 Driver Name:shiva Phone Number: 7867643445	Added to the list of driver management	Added to the list of driver management	Pass
7	Adding Vehicle management	Truck number: dft126 Permit number: ka56ew1256 Select Driver Id: 006	Vehicle Created Successfully	Vehicle Created Successfully	Pass
8	Booking from the customer / user and to add driver for it	From: Bagalkot To:Dharwad Ordered On: 2021-05-16 Weight:88kg Add Driver clicking Select Driver Id: 006	Booked Successfully with id 65eN3unRnexIZJTikqQP	Booked Successfully with id 65eN3unRnexIZJTikqQP	Pass

Test Case ID	HB_003	Test Case Description	Test the User Page of the Logistic Management app		
Created By	Brunda H Y	Reviewed By	Rohan N, Brunda H Y	Version	
Tester's Name	Brunda H Y	Date Tested	\May 3, 2021	Test Case (Pass/Fail/Not Executed)	Pass

<u>Test Scenario</u>	Verify on the Admin Page of the app
-----------------------------	-------------------------------------

S#	Step Details		Expected Result	Actual Result	Pass/Fail/Not executed /Suspended
1	Sign Up	Name:chethu Contact Number: 6756453423 Email Id: chethu@gmail.com Password:Chethu@ 123	Document created / updated	Document created / updated	1
2	Sign In	valid email and valid password Email: chethu@gmail.com Password:Chethu@ 123	Login Successfully	Login Successfully	2
3		valid email and invalid password Email: chethu@gmail.com Password:Chethu12 3	Email or password wrong	Email or password wrong	3

4		invalid email and valid password Email: chhu@gmail.com Password:Chethu123	Email or password wrong	Email or password wrong	4
5		invalid email and invalid password Email: chthu@gmail.com Password:chethu123	Email or password wrong	Email or password wrong	5
6	Booking	Select From: Bagalkot Select To: Dharwad Enter weight:88 unit:kg	Booked Successfully with id 65eN3unRnexIZJTikqQP	Booked Successfully with id 65eN3unRnexIZJTIkqQP	6
7	Tracking before assigning the driver from admin	From: Bagalkot To:Dharwad Ordered On: 2021-05-16 Weight:88kg Current Location:No Driver Assigned yet	No driver Assigned yet	No driver Assigned yet	7
8	Tracking after location updated from the driver	From: Bagalkot To:Dharwad Ordered On: 2021-05-16 Weight:88kg Current	Location updated is visible	Location Updated will be visible	8

		Location:Chitradurga			
9	clicking on userprofile> Logout	logout from the user	logout from the user	Pass	
Test Case ID	HB_004	Test Case Description	Test the Driver Page of the Logistic Management app		
Created By	Brunda H Y	Reviewed By	Rohan N, Brunda H Y	Version	
Tester's Name	Brunda H Y	Date Tested	\May 3, 2021	Test Case (Pass/Fail/Not Executed)	

<u>Test Scenario</u>	Verify on the Driver page of the app
-----------------------------	--------------------------------------

S#	Step Details		Expected Result	Actual Result	Pass/Fail/Not executed /Suspended
1	Order	From: Bagalkot To:Dharwad Ordered On: 2021-05-16 Weight:88kg	Updated Assigned list will be visible	Updated Assigned list will be visible	Pass
2	Update Location	Driver Id:006 Select Location: Chitradurga	Location Updated Successfully	Location Updated Successfully	Pass

8. Future Scope

Logistics & Supply Chain Management professionals are high in demand in various organizations in national as well as international level. Logistics & Supply Chain Management are the necessary terms of company success and customer satisfaction.

Logistics management is successfully implemented with the following features:

1. Admin can add trucks, add and assign drivers, view booking, view rating and view updated location.
2. Driver can view the assigned bookings and update the location.
3. User can make bookings, view the updated locations and give ratings accordingly.

One of the most common entry-level jobs in this field is customer service management. Customer service is an important part of logistics and supply chain management on an individual level. A career in supply chain provides travel opportunities with better remuneration. The career fields in logistics management are business sustainability and green operations, general management, operations management, project management and small business management/entrepreneurship.

Following functionalities can be implemented within the application: Real Time Location Tracking, Authentication, Map Integration, Receipt Generation, and Final Acknowledgement. A full flexible logistics management system can be developed considering this project as basis with the inclusion of following features: Registration, List of Vehicles, Vehicle Booking, Real-time alerts, Payment Gateway, Track Consignment, Book Meetings, Rate and Review, Request management, Shipper details, Route tracking, Driver Monitoring, Billing, Vehicle Management, Schedule Meetings

9. Conclusion

In this project, we presented a basic version of a logistics management system and a low cost development of application. We went through all the features that were included in our application. Through the design of this system we acquired much knowledge of logistics business.

Through this application we have touched all the necessary features that is required in any application to be called as a logistics management app. This is a miniature version of a flexible logistics management system. It provides platform for all three actors to perform their role efficiently and consistently. This application can be used in shipment business which has a small scale number of trucks and employees. In today's competitive business environment, getting the correct products to the right place at the right time via the most cost-effective manner is paramount for one's business' success. To stay ahead, logistics professionals are using logistics and supply chain mobile applications as must-have tools for operational excellence. Aligning their business with the application will help them keep track of various business activities and increase productivity. This application helps logistics professionals achieve every business process from tracking inventory and shipments, books meetings with clients, to tracking tasks and projects. By never losing touch with one's customer, one is able to collaborate as well as better tracking and responding to cases, allowing you to maintain and build stronger customer relationships.

10. REFERENCES

- [1] Alexandra Lagorio, Giovanni Zenezini, Giulio Mangano, Roberto Pinto 2020. “A systematic literature review of innovative technologies adopted in logistics management”.
- [2] Chris Caplice and Yossi Sheffi Massachusetts Institute of Technology. The International Journal of Logistics Management, January 1995. “A Review and Evaluation of Logistics Performance Measurement Systems”.
- [3] Ratapol Wudhikarn Chiang Mai University, Nopasit Chakpitak Chiang Mai University, Gilles Neubert Emlyon business school. International Journal of Production Research, January 2018. “Performance Measures of Logistics Management: An Intellectual Capital Perspective”.
- [4] Ronald H. Ballou, Case Western Reserve University, July 2007. “The Evolution and Future of Logistics and Supply Chain Management”.
- [5] Darja Topolšek , University of Maribor;; Kristina Čižiūnienė; Tina Cvahte Ojsteršek, University of Maribor, December 2018. “Defining transport logistics: A literature review and practitioner opinion based approach”.
- [6] Mamdouh Tlaty & Mohamed Moutmihi Hassan II University, FSJES- Mohammedia, Morocco; Global Journal of Management and Business Research: A Administration and Management, Year 2015; “From the logistics function to the logistics service: A literature review”.
- [7] Huan Neng Chiu, National Taiwan Institute of Technology, Taipei, Taiwan, and Republic of China; “The integrated logistics management system: a framework and case study”.
- [8] Prof. Angappa Gunasekaran, International Journal of Logistics Systems and Management; “Developing an E-Logistics System: A Case Study”.
- [9] C. Thallera , N. Moraitakisb , H. Rogersc , D. Sigged , U. Clausena , H.-C. Pfohlb , E. Hartmannc , B. Hellingrathd , The German Federal Ministry of Education and Research (BMBF) ; “Analysis of the Logistics Research in India – White Paper”.
- [10] Shlomo Globerson , Gal Wolbrum², School of Business, Tel Aviv University, Tel Aviv, Israel, Azrieli College of Engineering, Jerusalem, Israel ²Maccabi Health Care, Tel Aviv, Israel International Journal of Business and Economics Research ; “Logistics management and supply chain management: A critical evaluation”.

Appendix – 1

Paper Accepted Submit Payment Proof

IJSREM Journal <ijsremjournal@gmail.com>

Mon 5/24/2021 12:51 PM

To: ROHAN N <201710100751@presidencyuniversity.in>

Dear Author,

We are pleased to inform you that your paper titled Application for the Management of Logistics System was accepted for publication in “**International Journal of Scientific Research in Engineering and Management (IJSREM)**”, Volume 05 Issue 05, May 2021.

Paper ID : [IJSREM8252](#)

Regards,
Editor-in-chief

International Journal of Scientific Research in Engineering and Management (IJSREM) [h
ttps://www.ijsrem.com/](https://www.ijsrem.com/) ijsremjournal@gmail.com

--

Editor

IJSRE
M

International Journal of Scientific Research in Engineering and Management

Bangalore,

Karnataka, 560037

ijsremjournal@gmail.com

Paper Accepted Submit Payment Proof

IJSREM Journal <ijjsremjournal@gmail.com>

Mon 5/24/2021 12:51 PM

To: ROHAN N <201710100751@presidencyuniversity.in>

Dear Author,

We are pleased to inform you that your paper titled Application for the Management of Logistics System was accepted for publication in "International Journal of Scientific Research in Engineering and Management (IJSREM)", Volume 05 Issue 05, May 2021.

Paper ID : [IJSREM8252](#)

Kindly send the payment RS: 800/- towards the publication charges of your article.

Pay Through Google Pay : 9550803309 (Doogole-Apps) [For Fast Publication]

*Submit Payment Proof through Online : <http://ijjsrem.com/submit-payment-proof/>

Payment options:

1) Pay online by using debit card, credit card, Net banking, Cash card, Mobile payments etc,

To pay online click: <http://ijjsrem.com/pay-online/>

2). Pay Through Google Pay (Tez) : 9550803309 (Doogole-Apps) [For Fast Publication]

Regards,

Editor-in-chief

International Journal of Scientific Research in Engineering and Management (IJSREM)

<https://www.ijjsrem.com/> ijjsremjournal@gmail.com

--

Editor

IJSREM

International Journal of Scientific Research in Engineering and Management

Bangalore,

Appendix – 2

Splash Sheet

XML Code:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
android:background="@drawable/bg"
tools:context=".Splash">
<LinearLayout
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
android:layout_weight="1">

<ImageView
android:id="@+id/background_image"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="300px"
android:src="@drawable/app_logo" />
<TextView
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:gravity="center"
android:textColor="@color/white"
android:textSize="100px"
android:text="LogiTech">
</TextView>
</LinearLayout>
<LinearLayout
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_weight="4">
<TextView
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:gravity="center"
android:textColor="@color/white"
android:textSize="50px"
android:text="Powered by UP2-G69 \n 2020-21">
</TextView>

</LinearLayout>
</LinearLayout>
```

Java Code:

```
package com.upg.logisticsmgmt;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.util.Log;

import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.FirebaseFirestore;

import java.util.HashMap;
import java.util.Map;

public class Splash extends Activity {

    private static int SPLASH_TIMER = 3000;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash);

        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                Intent intent= new Intent(getApplicationContext(),MainActivity.class);
                startActivity(intent);
                finish();
            }
        }, SPLASH_TIMER);
    }

    @Override
    protected void onPause() {
        super.onPause();
        finish();
    }
}
```

Choice Page

XML Code:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
android:background="@drawable/bg"
tools:context=".MainActivity">

<TextView
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:gravity="center"
android:layout_marginTop="200px"
android:textSize="80px"
android:textColor="@color/white"
android:text="Welcome \n Choose your Role" />
<Button
android:id="@+id/button1"
android:layout_width="400px"
android:layout_height="wrap_content"
android:layout_gravity="center"
android:textSize="60px"
android:textAllCaps="false"
android:layout_marginTop="100px"
android:text="Admin"/>

<Button
android:id="@+id/button2"
android:layout_width="400px"
android:layout_height="wrap_content"
android:layout_gravity="center"
android:textSize="60px"
android:textAllCaps="false"
android:layout_marginTop="100px"
android:text="Driver"/>
<Button
android:id="@+id/button3"
android:layout_width="400px"
android:layout_height="wrap_content"
android:layout_gravity="center"
android:textSize="60px"
android:textAllCaps="false"
android:layout_marginTop="100px"
android:text="User"/>

</LinearLayout>
```

Java Code:

```
package com.upg.logisticsmgmt;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

import com.upg.logisticsmgmt.pojo.Booking;
import com.upg.logisticsmgmt.pojo.Driver;
import com.upg.logisticsmgmt.pojo.User;

public class MainActivity extends AppCompatActivity implements View.OnClickListener{

    Button adminBtn;
    Button driverBtn;
    Button userBtn;

    public static User user;
    public static Booking booking;
    public static Driver driver;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        adminBtn= findViewById(R.id.button1);
        adminBtn.setOnClickListener(this);

        driverBtn= findViewById(R.id.button2);
        driverBtn.setOnClickListener(this);

        userBtn= findViewById(R.id.button3);
        userBtn.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        if (v== adminBtn) {
            Intent intent= new Intent(getApplicationContext(),AdminLogin.class);
            startActivity(intent);
            finish();
        }

        if (v == driverBtn) {
            Intent intent= new Intent(getApplicationContext(),DriverLoginActivity.class);
            startActivity(intent);
            finish();
        }

        if (v == userBtn) {
            Intent intent= new Intent(getApplicationContext(),CustomerRegistration.class);
```

```

startActivity(intent);
    finish();

    }
}
}

```

Driver

Login:

XML:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bg"
    android:orientation="vertical"
    android:padding="20px"
    tools:context=".AdminLogin">

```

```

<ImageView
    android:id="@+id/background_image"
    android:layout_gravity="center"
    android:layout_marginTop="100px"
    android:layout_width="300px"
    android:layout_height="300px"
    android:src="@drawable/app_logo" />

```

```

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:textSize="80px"
    android:textColor="@color/white"
    android:text="Welcome Driver \n SignIn to Continue" />

```

```

<EditText
    android:id="@+id/driverLoginId"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textColorHint="@color/hint_color"
    android:layout_marginTop="100px"
    android:hint="Driver Id"

```

```

/>

```

```

<EditText
    android:id="@+id/driverPassword"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="50px"
    android:inputType="textPassword"

```

```

android:textColorHint="@color/hint_color"
android:hint="Password"
/>
<Button
android:id="@+id/driverLoginBtn"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_gravity="center"
android:textSize="60px"
android:background="@drawable/rounded_corner"
android:layout_marginLeft="20px"
android:layout_marginRight="20px"
android:textAllCaps="false"
android:layout_marginTop="100px"
android:text="Login"/>
</LinearLayout>

```

Java Code:

```

package com.upg.logisticsmgmt;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.firestore.CollectionReference;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;
import com.upg.logisticsmgmt.pojo.Driver;
import com.upg.logisticsmgmt.pojo.User;

public class DriverLoginActivity extends AppCompatActivity implements View.OnClickListener{

    Button driverLoginBtn;
    EditText driverName, driverPassword;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_driver_login);

        driverLoginBtn= findViewById(R.id.driverLoginBtn);
        driverLoginBtn.setOnClickListener(this);

        driverName= findViewById(R.id.driverLoginId);
        driverPassword= findViewById(R.id.driverPassword);
    }
}

```

```

    }

    @Override
    public void onClick(View v) {
        if (v == driverLoginBtn) {
            if (validate()) {
                checkDriverLogin();

            } else {
                Toast.makeText(this, "Please check credentials", Toast.LENGTH_SHORT).show();
            }
        }
    }

    private boolean validate() {
        if (driverName.getText().toString().equals("")) {
            driverName.setError("Please enter valid Name");
            return false;
        }
        if (driverPassword.getText().toString().equals("")) {
            driverPassword.setError("Please enter valid Password");
            return false;
        }

        return true;
    }

    private void checkDriverLogin() {
        FirebaseFirestore firebaseFirestore = FirebaseFirestore.getInstance();
        CollectionReference collectionReference = firebaseFirestore.collection("Drivers");
        collectionReference.document(driverName.getText().toString()).get().addOnCompleteListener(new
        OnCompleteListener<DocumentSnapshot>() {
            @Override
            public void onComplete(@NonNull Task<DocumentSnapshot> task) {
                if (task.isSuccessful()) {
                    DocumentSnapshot document = task.getResult();
                    if (document != null && document.getString("driverPassword") != null
                        && document.getString("driverPassword").equals(driverPassword.getText().toString())) {
                        MainActivity.driver = document.toObject(Driver.class);
                        MainActivity.driver.setId(document.getId());
                        Intent intent = new Intent(getApplicationContext(), DriverDashboard.class);
                        startActivity(intent);
                        finish();
                    } else {
                        //No record
                        Toast.makeText(getApplicationContext(), "No Record Found", Toast.LENGTH_SHORT).show();
                    }
                } else {
                    // Failed
                    Toast.makeText(getApplicationContext(), "Something went wrong please try later", Toast.LENGTH_SHORT).show();
                }
            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                Toast.makeText(getApplicationContext(), "Something went wrong please try later", Toast.LENGTH_SHORT).show();
            }
        });
    }

```



```
}  
}
```

Dashboard:

XML:

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
  xmlns:app="http://schemas.android.com/apk/res-auto"  
  xmlns:tools="http://schemas.android.com/tools"  
  android:layout_width="match_parent"  
  android:layout_height="match_parent"  
  android:orientation="vertical"  
  android:background="@drawable/bg"  
  tools:context=".DriverDashboard">  
  <ImageView  
    android:id="@+id/driverLogout"  
    android:layout_width="200px"  
    android:layout_height="200px"  
    android:layout_gravity="right"  
    android:src="@drawable/ic_baseline_login_24" />  
  
  <LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:orientation="horizontal">  
  
    <androidx.cardview.widget.CardView  
      android:id="@+id/driverBookingBtn"  
      android:layout_width="wrap_content"  
      android:layout_height="wrap_content"  
      android:layout_margin="30dp"  
      android:layout_weight="1"  
      app:cardBackgroundColor="@color/cream"  
      app:cardCornerRadius="30dp">  
  
        <LinearLayout  
          android:layout_width="wrap_content"  
          android:layout_height="wrap_content"  
          android:layout_gravity="center"  
          android:orientation="vertical"  
          android:paddingTop="20dp">  
  
            <ImageView  
              android:layout_width="300px"  
              android:layout_height="300px"  
              android:src="@drawable/booking_icon" />  
  
            <TextView  
              android:layout_width="wrap_content"  
              android:layout_height="wrap_content"  
              android:layout_gravity="center"  
              android:text="Bookings"  
              android:textColor="@color/colorPrimary"
```

```

        android:textSize="50px"></TextView>
    </LinearLayout>

</androidx.cardview.widget.CardView>
<androidx.cardview.widget.CardView
    android:id="@+id/locationUpdateBtn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="30dp"
    android:layout_weight="1"
    app:cardBackgroundColor="@color/cream"
    app:cardCornerRadius="30dp">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:orientation="vertical"
        android:paddingTop="20dp">

        <ImageView
            android:layout_width="300px"
            android:layout_height="300px"
            android:src="@drawable/track_icon" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text="Loc Update"
            android:textColor="@color/colorPrimary"
            android:textSize="50px"></TextView>
    </LinearLayout>

</androidx.cardview.widget.CardView>

</LinearLayout>

</LinearLayout>

```

Java Code:

```

package com.upg.logisticsmgmt;

import androidx.appcompat.app.AppCompatActivity;
import androidx.cardview.widget.CardView;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;

public class DriverDashboard extends AppCompatActivity implements View.OnClickListener{

```

```

CardViewbookingBtn, trackBtn;
ImageViewlogoutBtn;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_driver_dashboard);

    bookingBtn= findViewById(R.id.driverBookingBtn);
    bookingBtn.setOnClickListener(this);

    trackBtn= findViewById(R.id.locationUpdateBtn);
    trackBtn.setOnClickListener(this);

    logoutBtn= findViewById(R.id.driverLogout);
    logoutBtn.setOnClickListener(this);
}

@Override
public void onClick(View v) {
    if (v == bookingBtn) {
        Intent intent= new Intent(getApplicationContext(),OrderListActivity.class);
        startActivity(intent);
        finish();
    }

    if (v == logoutBtn) {
        Intent intent= new Intent(getApplicationContext(),DriverLoginActivity.class);
        startActivity(intent);
        finish();
    }

    if (v == trackBtn) {
        Intent intent= new Intent(getApplicationContext(),UpdateLocationActivity.class);
        startActivity(intent);
        finish();
    }
}
}

```

Booking:

XML:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="20px"
    android:background="@drawable/bg"
    tools:context=".BookingActivity">

```

</LinearLayout>

```

bookingList= findViewById(R.id.bookingList);

backBtn= findViewById(R.id.bookingBackBtn);
backBtn.setOnClickListener(this);
progressDialog= new ProgressDialog(BookingActivity.this);
progressDialog.setCancelable(false); // set cancelable to false
progressDialog.setMessage("Please Wait"); // set message
progressDialog.show(); // show progress dialog
getAllBooking();

}

@Override
public void onClick(View v) {
if (v == backBtn) {
Intent intent= new Intent(getApplicationContext(),AdminDashboard.class);
startActivity(intent);
finish();
}

}

private void getAllBooking() {
FirebaseFirestore firebaseFirestore= FirebaseFirestore.getInstance();
CollectionReference collectionReference= firebaseFirestore.collection("Booking");
collectionReference.get().addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {
@Override
public void onComplete(@NonNull Task<QuerySnapshot> task) {
if (task.isSuccessful()) {

for (QueryDocumentSnapshot document : task.getResult()) {
Booking b = document.toObject(Booking.class);
b.setId(document.getId());
arrayList.add(b);
}

adapter = new BookingAdapter(BookingActivity.this, arrayList);
bookingList.setAdapter(adapter);

} else {
Toast.makeText(getApplicationContext(), "Failed to load", Toast.LENGTH_SHORT).show();
}

progressDialog.dismiss();
}

}).addOnFailureListener(new OnFailureListener() {
@Override
public void onFailure(@NonNull Exception e) {
Toast.makeText(getApplicationContext(), "Failed to load", Toast.LENGTH_SHORT).show();
progressDialog.dismiss();
}

});
}
}
}

```

Update Location:

XML:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@drawable/bg"
    android:padding="20px"
    tools:context=".UpdateLocationActivity">
```

```
<ImageView
    android:id="@+id/driverBackBtn"
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:src="@drawable/ic_baseline_arrow_back_24" />
<TextView
    android:id="@+id/driverIdTxtField"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="300px"
    android:textSize="50px"
    android:textColor="@color/white"
    android:text="Driver Id : 23423423"/>
```

```
<!-- Text Label -->
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dip"
    android:textSize="50px"
    android:text="Select Location:"
    android:layout_marginBottom="5dp"
/>
```

```
<!-- Spinner Element -->
<Spinner
    android:id="@+id/currentLoc"
    android:textSize="50px"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:entries="@array/locationList"
/>
```

```
<Button
    android:id="@+id/updateLocation"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginLeft="20px"
    android:layout_marginTop="100px"
    android:layout_marginRight="20px"
    android:background="@drawable/rounded_corner"
```

```

android:text="Update Location"
android:textAllCaps="false"
android:textSize="60px" />
</LinearLayout>

```

Java Code:

```

package com.upg.logisticsmgmt;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.firestore.CollectionReference;
import com.google.firebase.firestore.FirebaseFirestore;
import com.upg.logisticsmgmt.pojo.Booking;
import com.upg.logisticsmgmt.pojo.Driver;

public class UpdateLocationActivity extends AppCompatActivity implements View.OnClickListener{

    ImageView driverBackBtn;
    TextView driverIdTxt;
    Button updateLocation;
    Spinner currentLoc;
    ProgressDialog progressDialog;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_update_location);

        driverBackBtn= findViewById(R.id.driverBackBtn);
        driverBackBtn.setOnClickListener(this);

        updateLocation= findViewById(R.id.updateLocation);
        updateLocation.setOnClickListener(this);

        driverIdTxt= findViewById(R.id.driverIdTxtField);
        driverIdTxt.setText("Driver Id : " + MainActivity.driver.getId());

        currentLoc= findViewById(R.id.currentLoc);
    }

    @Override

```

```

public void onClick(View v) {
    if (v == driverBackBtn) {
        Intent intent= new Intent(getApplicationContext(),DriverDashboard.class);
        startActivity(intent);
        finish();
    }

    if (v == updateLocation) {

        progressDialog= new ProgressDialog(UpdateLocationActivity.this);
        progressDialog.setCancelable(false); // set cancelable to false
        progressDialog.setMessage("Please Wait"); // set message
        progressDialog.show(); // show progress dialog
        updateCurrentLocation();
    }
}

private void updateCurrentLocation() {
    FirebaseFirestore firebaseFirestore= FirebaseFirestore.getInstance();
    CollectionReference collectionReference= firebaseFirestore.collection("Drivers");
    String id = MainActivity.driver.getId();
    Driver d = MainActivity.driver;
    d.setCurrentLocation(currentLoc.getSelectedItem().toString());
    collectionReference.document(id).set(d).addOnCompleteListener(new OnCompleteListener<Void>() {
        @Override
        public void onComplete(@NonNull Task<Void> task) {
            if (task.isSuccessful()) {
                Toast.makeText(getApplicationContext(), "Location Updated Successfully", Toast.LENGTH_SHORT).show();
            }
            progressDialog.dismiss();
        }
    }).addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {
            Toast.makeText(getApplicationContext(), "Failed to update", Toast.LENGTH_SHORT).show();
            progressDialog.dismiss();
        }
    });
}
}

```