A Main Project report submitted in

CLOUD BASED E-COMMERCE APPLICATION USING FLUTTER AND FIREBASE

Partial fulfilment of the requirement for the award of the Degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

SUBMITTED

By

BALUSU MEHER CHAITANYA	**18671A0572**
NERELLA TARUN REDDY	**18671A0596**
PRODDUTURI VIVEK SAI	**18671A05A2**

Under the esteemed guidance of

DR .G . ARUN SAMPAUL THOMAS

ASSOCIATE PROFESSOR

Department of Computer Science and Engineering

J.B. Institute of Engineering & Technology

(UGC AUTONOMOUS)

(Affiliated to NAAC & NBA, Approved by AICTE & Permanently Affiliated to Jawaharlal Nehru Technological University, Hyderabad)
Bhaskar Nagar Yenkapally, Moinabad mandal, R.R. District, Telangana state 500075.

J.B. INSTITUTE OF ENGINEERING & TECHNOLOGY

(UGC AUTONOMOUS)

(Accredited by NAAC, Permanently Affiliated to JNTUH)

Yenkapally, Moinabad Mandal, R.R. Dist. -500075

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Main Project report entitled **"Cloud Based E-Commerce Application Using Flutter And Firebase"** submitted to the Department of Computer Science and Engineering, J.B Institute of Engineering & Technology, in accordance with Jawaharlal Nehru Technological University regulations as partial fulfillment required for successful completion of Bachelor of Technology is a record of bonafide work carried out during the academic year 2021-22 by,

BALUSU MEHER CHAITANYA	**18671A0572**
NERELLA TARUN REDDY	**18671A0596**
PRODDUTURI VIVEK SAI	**18671A05A2**

Internal Guide
DR .G . ARUN SAMPAUL THOMAS
ASSOCIATE PROFESSOR

Head of the Department
Dr. P. SRINIVASA RAO
PROFESSOR &HOD

External Examiner

J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY

(UGC AUTONOMOUS)

(Accredited by NAAC, Permanently Affiliated to JNTUH)

Yenkapally, Moinabad Mandal, R.R. Dist. -500 075

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We hereby certify that the Main Project report entitled **"Cloud Based E-Commerce Application Using Flutter And Firebase"** carried out under the guidance of**, DR .G . ARUN SAMPAUL THOMAS Associate Professor** is submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering.** This is a record of bonafide work carried out by us and the results embodied in this project report have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

Date:

BALUSU MEHER CHAITANYA	**18671A0572**
NERELLA TARUN REDDY	**18671A0596**
PRODDUTURI VIVEK SAI	**18671A05A2**

ACKNOWLEDGEMENT

At outset we express our gratitude to almighty lord for showering his grace and blessings upon us to complete this Main Project. Although our name appears on the cover of this book, many people had contributed in some form or the other to this project Development. We could not have done this Project without the assistance or support of each of the following.

First of all we are highly indebted to **Dr. P. C. KRISHNAMACHARY**, Principal for giving us the permission to carry out this Main Project.

We would like to thank **Dr. P. SRINIVASA RAO,** Professor & Head of the Department of COMPUTER SCIENCE AND ENGINEERING**,** for being moral support throughout the period of the study in the Department.

We are grateful to, **DR .G . ARUN SAMPAUL THOMAS** Associate Professor COMPUTER SCIENCE AND ENGINEERING, for his valuable suggestions and guidance given by her during the execution of this Project work.

We would like to thank Teaching and Non-Teaching Staff of Department of Computer Science and Engineering for sharing their knowledge with us.

BALUSU MEHER CHAITANYA **18671A0572**
NERELLA TARUN REDDY **18671A0596**
PRODDUTURI VIVEK SAI **18671A05A2**

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 4/Issue 06/40600086484

Date: 15/06/2022

Certificate of Publication

*This is to certify that author "**Balusu Meher Chaitanya**" with paper ID "**IRJMETS40600086484**" has published a paper entitled "CLOUD BASED E-COMMERCE APPLICATION WITH FLUTTER AND FIREBASE" **in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 4, Issue 06, June 2022***

Editor in Chief

IRJMETS Impact Factor 6.752

We Wish For Your Better Future

www.irjmets.com

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 4/Issue 06/40600086484

Date: 15/06/2022

Certificate of Publication

*This is to certify that author "**Nerella Tarun Reddy**" with paper ID "**IRJMETS40600086484**" has published a paper entitled "CLOUD BASED E-COMMERCE APPLICATION WITH FLUTTER AND FIREBASE" **in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 4, Issue 06, June 2022***

Editor in Chief

We Wish For Your Better Future

www.irjmets.com

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 4/Issue 06/40600086484

Date: 15/06/2022

Certificate of Publication

*This is to certify that author "**Prodduturi Vivek Sai**" with paper ID "**IRJMETS40600086484**" has published a paper entitled "CLOUD BASED E-COMMERCE APPLICATION WITH FLUTTER AND FIREBASE" **in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 4, Issue 06, June 2022***

Editor in Chief

IRJMETS Impact Factor 6.752

We Wish For Your Better Future

www.irjmets.com

ABSTRACT

Any online business, requires features such as wishlist, products categorization, various types of filters, payments, customer management and backend. E-commerce app is designed to satisfy all these requirements so that it saves the valuable time of our customers and gives a wide variety of options to choose from at one stop. The interface is designed using flutter, a multi-platform supporting framework. Also, this application involves a lot of data transfer, as a user is expected to scroll through a lot of products before purchasing. Firebase, backed by Google Cloud for backend, makes it a faster and smoother experience for the users. The primary goal of e-commerce is to reach maximum customers at the right time to increase sales and profitability of the business.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVATIONS

OS	Operating system
iOS	iPhone operating system
API	Application Programming Interface
SDK	Software development kit
UML	Unified Modelling Language
HCI	Human Computer Interaction
RDBMS	Relational Database System
OEM	Original equipment manufacturer
B2B	Business to business
C2C	Customer to customer
IDE	Integrated development environment
UI	User interface

CHAPTER 1
INTRODUCTION

1.1 Motivation

- E-commerce is fast gaining ground as an accepted and used business paradigm. More and more business houses are providing functionality for performing commercial transactions over the apps. It is reasonable to say that the process of shopping on the apps is becoming commonplace.
- With launch of these services many new start-ups are easily able sell their products and services online.

1.2 Problem Definition

Most of the small business owners use web interfaces to take their business online. Using an app which works on multiple platforms will help them achieve higher percentage of returning customers. The current small scale online stores do not have a smooth user interface. There are some limitations for the current system to which solutions can be provided as a future development. The thesis focuses on the modern-day problem of cross platform mobile development, that is met with compromises. The developers either need to choose between creating multiple code bases of the same app various operating systems used or they are forced to accept a less optimal solution that handicaps the application by limiting its compile time, accuracy and UI modifications. This is where Flutter steps in, it is a Google's UI toolkit that is used to develop beautiful applications that are natively compiled. Besides the features of development technologies, choosing the correct database is also vital. Firebase is a sturdy BaaS which handles the problem of storing huge unstructured data and is also relatively faster when compared to conventional RDMBS

1.3 Objectives of the Project

- Using flutter to create the interface makes it possible to run on multiple platforms like iOS, android and web.
- The system is designed to provide a smoother experience to the customers using Firebase for backend, which is supported by Google Cloud.
- Customer management is made easier using customer profiles, wishlists, reorder functionality etc.

CHAPTER 2
LITERATURE SURVEY

This literature review examines the factors that makes the use of flutter, favourable in cross-platform app development and the use of Firebase as BaaS for application development. Almost everyone today has a smartphone in their pocket, so it is crucial to build mobile applications which are not only user friendly but also available across multiple platforms. According to a study, mobile phones have contributed to 48% traffic compared to PCs (47%). There are two major operating systems prevalent in the market namely android and iOS, two vastly different systems. There was a need to bind these two by developing a platform that would build apps compatible with both the systems. That's where flutter steps in. Flutter is basically an environment that provides you to build a reliable and efficient application across platforms. Cross-platform frameworks that show resemblance to React Native and flutter, are discussed and implemented by various companies numerous times formerly. Flutter was released in in 2016 by Google. Apart from running on android and iOS , it works on Fuchsia as well. Flutter is exceptional because it is dependent on the device's OEM widgets rather than consuming web views.

A high efficiency rendering engine is used that build the apps as high performance as the native programs. Programming language proficiency is a hurdle for many beginners in the app development. Flutter is based on Dart, a language that was created to replace JavaScript. It is used extensively at Google and apps like Google AdWords have been created by using dart. Approach of flutter is radical for the cross-platform development. Interface objects, render and an engine to build animation, graphics and other libraries is provided. Inspired by React-native, Flutter also follows unidirectional data flow philosophy where widgets (equivalents of React's components) only rely on data provided by the parent widget and optionally their own state[3]A flutter project is written in Dart programming language and AOT (Ahead-of-time) compiled to the native platform architecture, hence achieving uncompromised speed. At the top level, flutter provides widgets that are composed of many widgets to make the most common

interface objects that we're used to on iOS and Android platforms. Because flutter follows an open and layered architecture, developers can make their own widgets compositing other widgets at any level of the layered architecture. In fact, that is how the Flutter team made the existing high-level widgets and there is no barrier from the framework for developers to do the same. This customization flexibility is unmatched by UI toolkits from either iOS or Android with hierarchical implementations and limited access levels. Usually, when the app is developed using native instruments, the resulting application communicates with the OS and requests it to draw the OEM widgets (buttons, text fields etc. provided by the operating system).

With the cross-platform technologies, the situation is different. Because JavaScript is not capable to contact OEM widgets directly, in the frameworks like React Native, developers write JS code which is then interpreted by the framework and the framework requests the OEM widgets from the OS[4]. This 20 creates a JavaScript bridge which is a significant overhead affecting the performance in an extremely negative way. Flutter is different. It does not use the OEM widgets at all. In other words, whenever a developer creates a button in Flutter, the framework does not ask the operating system to draw it. Instead, it uses its own renderer, and draws the button itself, pixel-by-pixel. The high-performance renderer in Flutter utilizes Skia Graphics Engine which is also a product of Google. It is used for rendering in such famous products as Google Chrome, Chrome OS, Mozilla Firefox, Android, LibreOffice and others. In sum, the previously mentioned facts mean that Flutter actually acts more like a game engine rather than a traditional cross-platform development solution. Unity or Unreal Engine, for instance, can be compared to Flutter to some extent[5]. Flutter draws user interfaces instead of sprites and objects in games.

The apps made with Flutter can look exactly the same way as native apps, but in fact they don't have that much in common with native, because instead of requesting the OS to provide an OEM widget, Flutter draws everything itself. This means that if a developer wants, he or she can make the user interface look exactly the same on all iOS and Android versions supported by Flutter. Or, as another example, Flutter can bring iOS

14 look and feel to iOS 10. In this paper, we look to create a application based on previous works and determine how useful Firebase is, in the case of development of messaging applications. Firebase was established by Andrew Lee and James Tamplin back in 2011 yet was launched formally in April 2012. Initially, the framework was designed to be used solely as a real-time database giving its APIs, enabling users to store and synchronize data and information across various users. However, Firebase was taken over by Google in 2014 and today, the service has various functionalities that offer development tools to various enthusiasts and entrepreneurs alike. Firebase is a framework which is useful for building portable and web applications for businesses which require real-time database which implies when one user updates a record in the database, the update should be conveyed to every single user instantly. It gives a basic and unified platform to many applications along with a host of other Google features packed-in with the service.

Firebase handles most of the server-side work when it comes to the development of applications. There are numerous elements that make Firebase such an essential tool in development from a developer's point of view. In this way, it helps maintain a state of harmony between the developer and the client by causing minimal delay of work Many developers are currently developing messaging applications with online solutions similar to Firebase, which provide real-time database integration facilities. Various open-source platforms such as Parse Server or Horizon offer similar services such as Firebase and offer developers to migrate from one vendor to another, but they also come with their own problems. Developers are also trying to develop methods to optimize file transfer through such applications and to integrate more technologically advanced features into their applications.

As we all know, internet and e-commerce are entirely committed towards every developed country. But we think it can be accomplished and can make a remarkable benefit to developing countries also if an ideal business purpose can be made. Ohidujja man et.al [4] clearly discussed that E-commerce is a revolution & turning point in online business practices and can make a huge contribution to the economy and Hasan et.al [5]

also indicated that currently, e-commerce organizations have increasingly become a fundamental component of business strategy and a strong catalyst for economic development. A huge amount of research works has been done on e-Commerce which is basically on online shopping. A large group of researchers has found out and also pointed out the necessity and possibilities of Online Shopping. On the other hand, limitation of ecommerce is found and at the same time, they provided essential suggestion and came to a prediction to make Online Shopping more useful for the consumers. But the contribution of traditional marketing is also inescapable but compare to online shopping it is less effective we think. So on this basis, Mehrdad Salehi et.al [17] found out distinguish between online marketing & traditional marketing. Though most of the people of Bangladesh especially the rural people are not enough capable of operating internet to run the online business. For that reason, they need to be dependent on traditional marketing.

CHAPTER 3
ANALYSIS

With the development and progress of society, human beings have entered the era of e-commerce, which has changed the traditional business model and improved the level of information management, playing an irreplaceable positive role in economic development. Based on this, it is very necessary to promote the progress of the e-commerce era. In-depth understanding of the actual situation of e-commerce operation found that information management and information systems have a greater impact on e-commerce. Therefore, in order to adapt to the development needs of the e-commerce era, it is very meaningful to strengthen the information management and information system construction.

According to the transaction object of e-commerce, e-commerce can be divided into the following categories: The first category is enterprise-to-business e-commerce (business to business, referred to as “B2B”). It refers to the way in which enterprises conduct transactions through e-commerce, and the sales method relative to B2C is business to customer. For example, a factory that sells disposable tableware, if its main shipment target is a restaurant, this is the B2B model, where the restaurant provides disposable tableware to the user, and the restaurant-to-user model is focused on B2C/B2B [1]: the establishment of an interenterprise network and the stability of the supply chain system. B2C needs to rely on economies of scale to attract purchases and lower prices to increase profits. B2B does not rely on scale, but relies on the establishment of an interenterprise network to stabilize its sales. The second category is business to consumer e-commerce (business to customer, referred to as “B2C”). This form of e-commerce is generally based on the online retail industry, mainly through the network to carry out online sales activities, and is an electronic retail model. The B2C model is the earliest e-commerce model in China. The third category is consumer-to-consumer e-commerce (customer to customer, referred to as “C2C”). It means e-commerce behavior among consumers. For example, a consumer has a computer, trades through the network, and sells it to another consumer. This type of transaction is called C2C e-commerce. The

fourth category is business to government (B2G). It is an e-commerce activity between enterprises and enterprises and government agencies. For example, some local governments in China are currently experimenting with online procurement. The government publishes the purchase list through the Internet and conducts bidding through online bidding. Enterprises and companies bid through electronic means and finally complete the government procurement task electronically.

E-Commerce refers to various business activities carried out by means of servers and browsers in the context of the current network environment, so that merchants and consumers can realize online transactions and directly realize shopping and transactions through electronic payment and the like. Compared to traditional business, e-commerce is a new service that combines financial, transaction, and business activities.

3.1 Existing System

- There are some limitations for the current system to which solutions can be provided as a future development.
- Most of the small business owners use web interfaces to take their business online. Using an app which works on multiple platforms will help them achieve higher percentage of returning customers.
- The current small scale online stores do not have a smooth user interface.
- Assists in the improvement of our standard, carefully crafted products for distribution within the showcase.
- The framework allows a buyer to quickly explore or search for an item.
- It provides an easy way to purchase items directly from sellers without the use of a third-party interface

3.2 Proposed System

- Using flutter to create the interface makes it possible to run on multiple platforms like iOS, android and web.

- The system is designed to provide a smoother experience to the customers using Firebase for backend, which is supported by Google Cloud.
- Customer management is made easier using customer profiles, wishlists, reorder functionality etc.
- More attractive user interface.
- Faster Server-to-client-side response.
- Tried to make the rural individuals fill out the description form which will provide more about their products.
- Tried providing a better and user friendly environment than the existing one.

3.3 Software Requirement Specification

- Operating system : Windows 10
- Technology : Flutter
- IDE : Visual Studio

3.4 Hardware Requirement Specification

- Processor : Dual core
- Ram : 1GB
- Hard disk : 8 GB
- Keyboard : Standard Windows Keyboard
- Mouse : Two or Three Button Mouse

3.5 PURPOSE

In recent years, online purchasing has become increasingly popular. People all around the world prefer to buy things online rather than in stores, which is why Amazon and other similar online retailers are not only successful but also increasing interest in online commerce. (Darrell, 2011) In today's world, though, innovation is assisting us in breaking down some of these barriers by using computer frameworks to aid us with various forms of labour. As we all know, the world of online purchasing is quickly developing. As 7 a result, e-commerce apps have already developed various new tactics and technologies for giving information to individuals. These ideas and technologies, however, provide a number of challenges and concerns for people.

For As a mobile application developer, it's important to remember the distinctions between mobile development and more traditional software development. Consider how varied the settings for utilizing mobile apps vs desktop computers can be, for example, recognizing the constraints and complexities of a touchbased user interface, and considering the unique security vulnerabilities that mobile technologies provide. A mobile developer should also be mindful of the importance of user experience and the fact that most users are acclimated to a certain operating system's design guidelines.

3.5.1 Scope

To deliver information to individuals, several innovative strategies and innovations in e-commerce applications have been established. Globally, online shopping has experienced substantial growth. Online shopping is used by 21.8 percent of the world's population. You're missing out on a huge group of potential purchasers if you're not selling online. 73% of customers browse across several channels, including Google, social media, and email.

3.5.2 Functional Requirements

The functional requirements for this application that are necessary to complete this project are as follows:

- Application must start when client clicks on app icon.

- Application must show the home page of the application first.
- Application must provide the selected category's content list for the client.
- Client must be able to view the products • Client must be able to view the products image.
- Application must provide the selected product's description for the client.
- Client must be able to add the product to the cart.
- Application should add the product to the customer's shopping cart.
- Application must provide the customer's information form for the client.
- Client must send the provided credentials to the application.

3.5.3 Non- Functional Requirements

The non-functional requirements for this application that are necessary to complete this project are as follows:

- Application should have an interface to display products to the customer.
- Application should have an interface to display list of categories.
- Application should be able to display list of categories, so customers easily click on them.
- Application should be able to display a product's description.
- Application should have an interface for the customer to add a product to the shopping cart.

CHAPTER 4
SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

The initial stage in our project's technological execution is to plan what we want to build that will benefit our society. We've attempted to question everything that may be touched by the program up to this point. Finally, we want to create an ecommerce software that allows users to purchase items without having to go to an actual store or shop. This is an excellent concept for easing the difficulties faced by many folks who are unable to go to the market to purchase goods. An entire store is there at their fingertips; all they have to do is scroll and choose the things that best suit their needs.

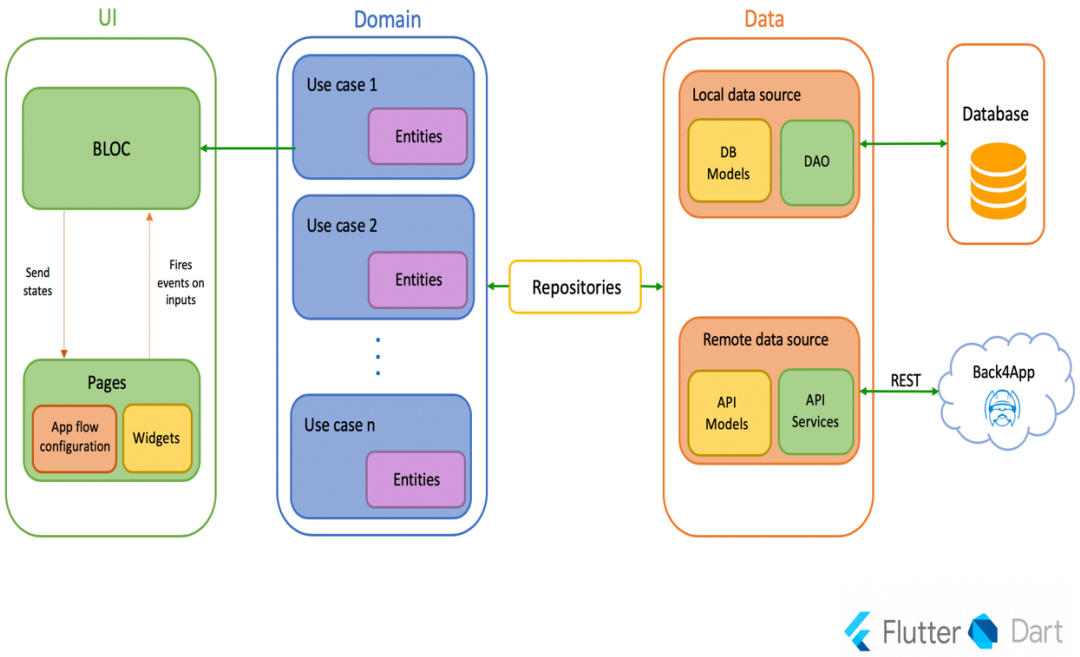


Figure 4.1 : System Architecture

It's now time to install the programmer and begin configuring the functionalities that may be used right away. In fact, this step should always come first before developing any procedures or rushing to customize any module.

- The first step is to install and configure flutter. After that, we'll install it by following the instructions on the official flutter website.
- The second step is to install and configure Android Studio, after which we will install the flutter and dart plugins to use flutter in Android Studio.
- We built up an emulator in Android Studio to run and test our app in this phase.
- The fourth step is to create a home screen. Then we used dart language to create the home screen.
- We developed items and their product detail pages after designing the home screen. Similarly, we have created each design component separately.
- To pick numerous things at once, we have provided add to cart capability to our app. The add to cart feature also allows you to buy numerous items at once.
- We've introduced a login and registration form, and users must first register and login before placing an order. This was accomplished using firebase and flutter.

4.1.1 Flutter Firebase Working

When a client (iOS or Android) requests a firebase API, the client sends its query to the firebase database, which then responds to the firebase API and the client (IOS or Android).

In Flutter, this is a lib folder where we can handle all our app's screens. This folder also contains the functioning section. This folder is responsible for 90% of our app development.

- **Pubspec.yaml** manages all our library files. This is a very private section of our software. It is also space constrained.
- Two responsible libraries for firebase integration are
  1. firebase_core: ^1.16.0
  2. firebase_auth: ^3.3.17
- Flutter asset folder, where we keep track of all the assets for our flutter app. Images, static data, videos, and other assets
- The IOS folder is responsible for all IOS development operations. Flutter is a hybrid technology, which means it can manage several platforms.

- It manages all the tasks required for Android development, same as IOS.

4.1.2 Firebase Configuration for Android and IOS:

To use Firebase with Flutter, we must first create a Firebase console project. We've established a project called Angel on Firebase. After that, we added packages for Android and iOS.. We inserted a google json file for Android and an info plish file for iOS in the root directory of both Android and iOS. This is how we set up our Firebase application for iOS and Android. We've also included packages in the pub spec files for authentication.

We received the project id, which is essentially our project's unique identification. We also issued a project number, which we may use to configure integrations with Firebase or other third-party services. We also gained the public facing name, which will appear on emails sent to users after they create an account inour application.

We ran into the following issues during configuration and customization.

- We placed the google services json file in a different location from the root directory while configuring the application for Android. As a result, when we test an application, we get unexpected problems. However, when we place the google services json file in the root directory of Android, it does not work.
- We had an authentication problem with the application while managing sessions. When a user logs out of an application, they will be sent to the home screen without having to log in again. This is a problem with authentication. We have resolved in our app.
- We have a problem with password letters. When we enter a password with fewer than six letters, it is not checked, and no error is displayed. We've corrected the problem, and now the password must contain more than six letters.
- When we add a product to a cart, it appears in the cart screen. This has been a huge problem for us. This problem was handled by examining indexes in an array. An array does not hold indexes. As a result, the cart displays an empty value.

- Users do not need to log in before placing an order. We were able to resolve this problem by utilizing session and authentication. Users must now be logged in before placing an order or seeing the main home page.

4.2 Introduction

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

4.3 UML Diagrams

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

4.3.1 Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

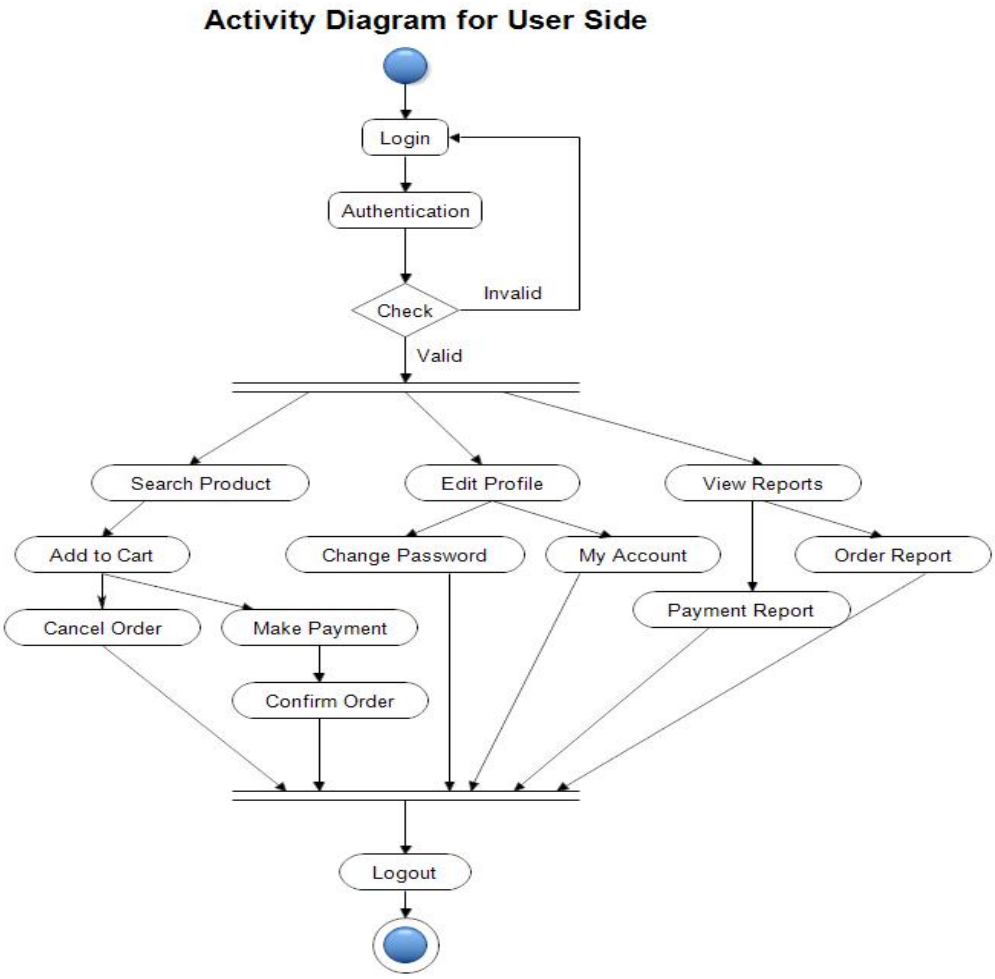


Figure 4.3.1 : Activity Diagram

4.3.2 Object Diagram

Object diagrams are derived from class diagrams so object diagrams are dependent upon class diagrams. Object diagrams represent an instance of a class diagram. The basic concepts are similar for class diagrams and object diagrams. Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment. These are used to render a set of objects and their relationships as an instance.

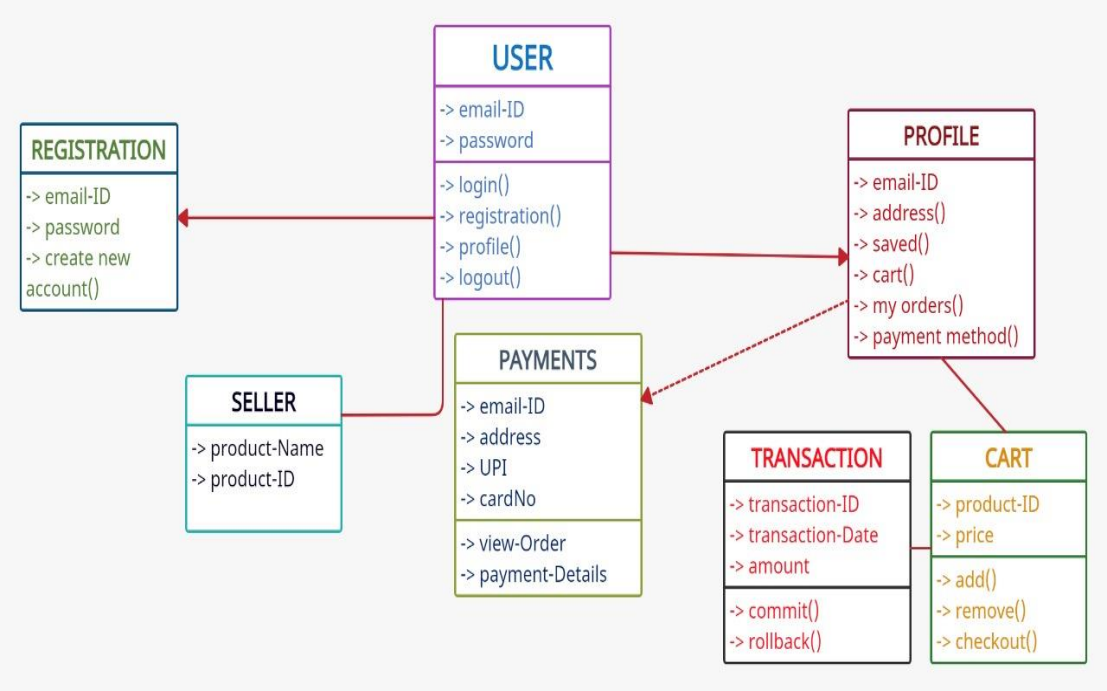


Figure 4.3.2 : Object Diagram

4.3.3 State-Chart Diagram

State-Chart Diagram describes different states of a component in a system. The states are specific to a component/object of a system. A State-chart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events.

Figure 4.3.3 : State-Chart Diagram

4.3.4 Component Diagram

Component diagrams are different in terms of nature and behavior. Component diagrams are used to model the physical aspects of a system. Now the question is, what are these physical aspects? Physical aspects are the elements such as executables, libraries, files, documents, etc. which reside in a node. Component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems

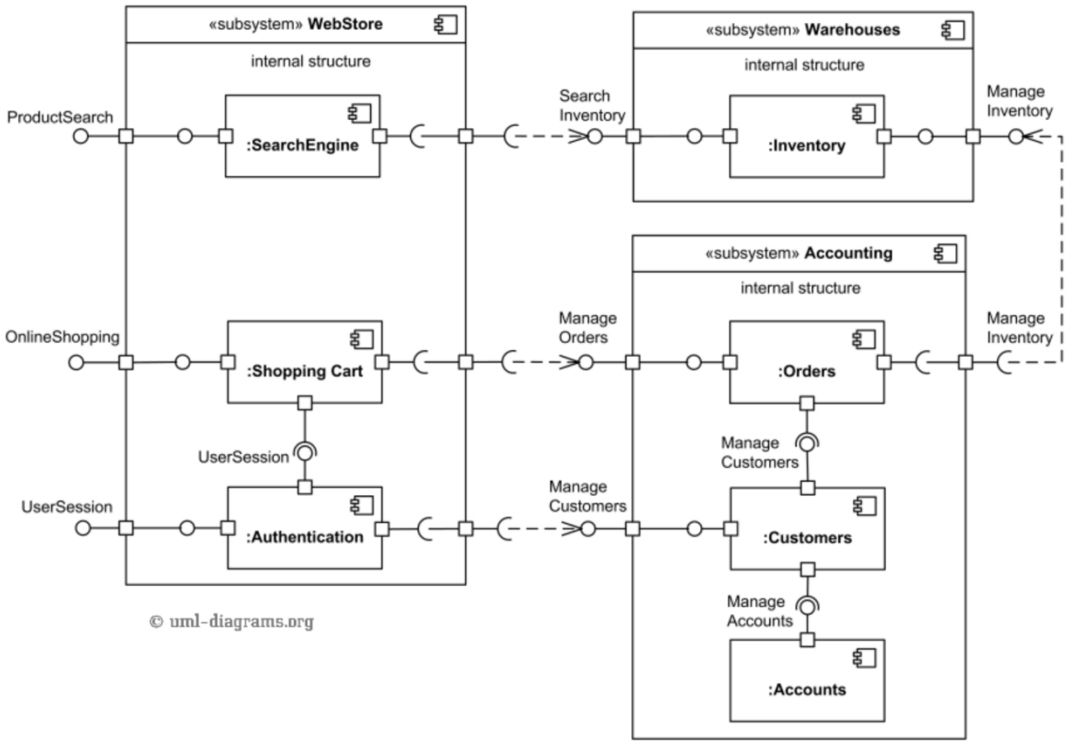


Figure 4.3.4 : Component Diagram

4.3.5 Deployment Diagram

Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed. Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

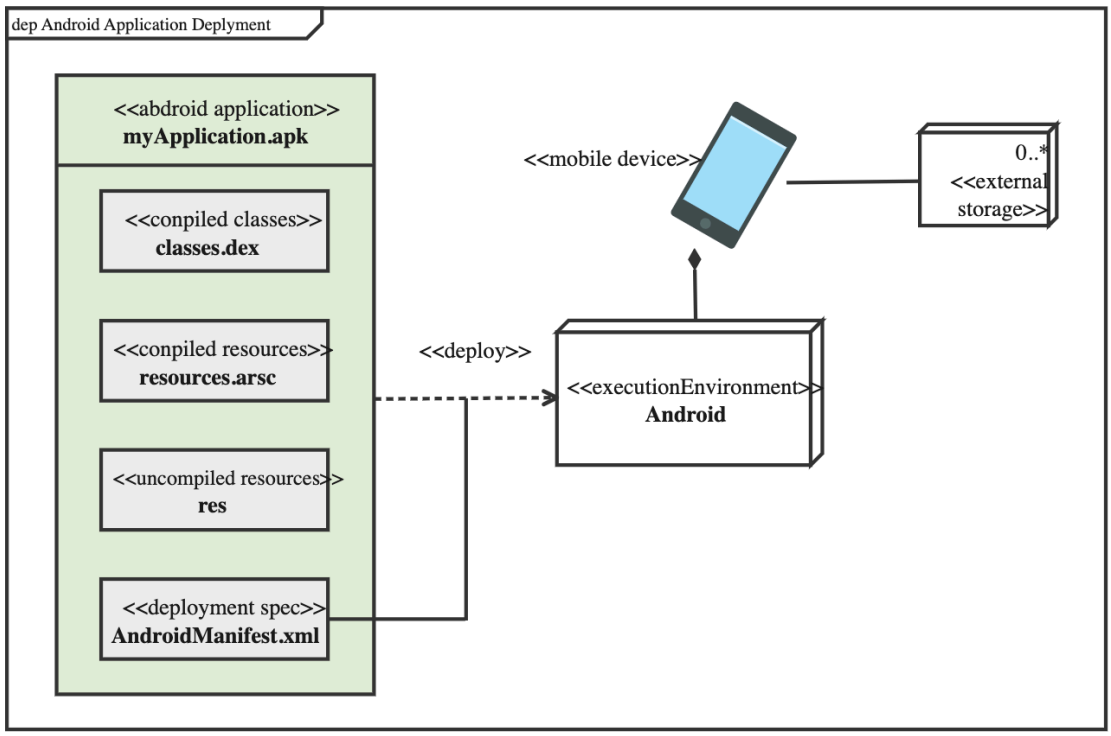


Figure 4.3.5 : Deployment Diagram

4.3.6 USECASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML is a kind of behavioral outline positive by and produced using a Use-contextual investigation. Its determination is to surviving a graphical sign of the usefulness giving by a framework regarding performing artists and their points (spoke to as use cases), and any conditions between those utilization cases. The key reason for an utilization case outline is to show what framework capacities are performed for which on-screen character. Parts of the on-screen characters in the framework can be delineated.

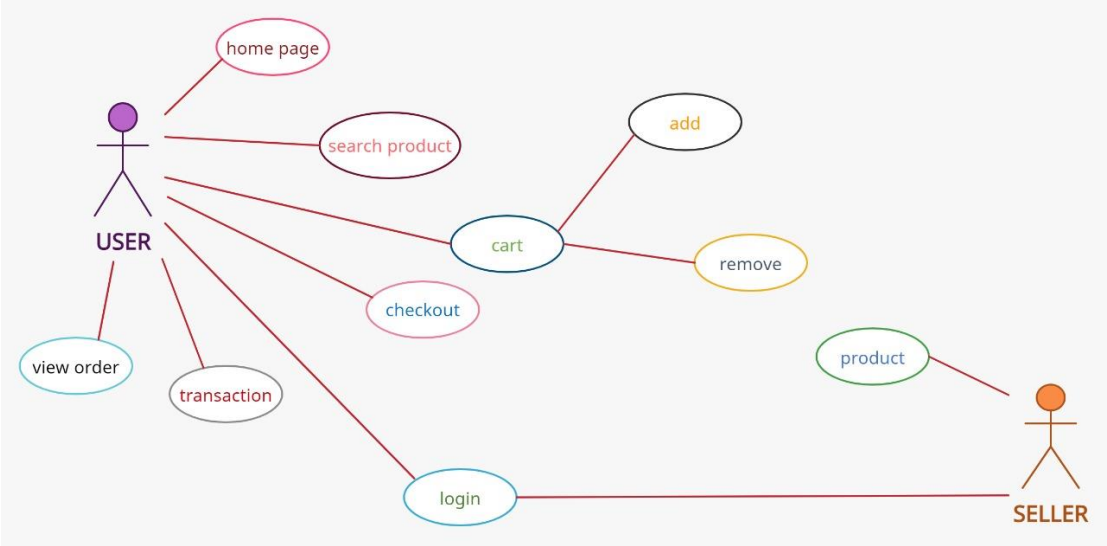


Figure 4.3.6 : Usecase Diagram

CHAPTER 5
IMPLEMENTATION

5.1 SYSTEM MODULES

5.1.1 Flutter

This application is built on Flutter technology. Flutter offers several advantages because it is a Google product. The following is a quick overview of flutter. The app development process is revolutionized by Flutter. You can design, test, and publish stunning mobile, web, desktop, and embedded apps by writing onetime code. Google built Flutter as an open-source project. It's used to make hybrid apps for Android, iOS, Linux, macOS, Windows, Google Fuchsia, and the web from a single codebase. (Flutter, no date) You can develop and iterate quickly using Hot Reload. You'll observe changes very immediately after updating the code, with no loss of state. Maintain every pixel to create unique, responsive designs that look and feel great on every device.

Every component in flutter is referred to as a widget. It is entirely widget based. Stateless or stateful widgets are available. There are several states in a stateful widget, not just one. They have the ability to modify their state in response to user interest. A Stateless widget, on the other hand, is a static widget that is utilized for static data or functionality. It is devoid of any state. We need to utilize it in some circumstances because we work with static data. Flutter is an open-source project, anybody may use and contribute to it at any level. It's better to use it and offer your passion for Google technology by becoming a Google contributor. Flutter has a massive user base since it is a Google product, and Google is the most widely used search engine on the planet. Flutter outperforms other hybrid solutions in terms of cost. It's one of the many advantages of flutter. Flutter developers may encrypt app data with both asymmetric and symmetric keys using the iOS SecKey API and the Common Crypto library. Flutter code is written in Dart, which has many cryptos and encrypts libraries that employ various cryptographic hashing and encryption techniques.

5.1.2 Android

Android is a multitasking mobile operating system that runs on smartphones, tablets, readers, televisions, and even domestic robots. Google purchased and marketed this operating system, which was developed by 'Android Inc' and based on Linux. Android's introduction as an operating system (OS) in 2008 was a spark, and it quickly became popular among smart gadgets. Modern smartphones and tablets might be considered stash minicomputers thanks to this operating system. By 2020, the Android platform will have about 35 billion users worldwide, making it the most popular mobile operating system on the planet. A large number of smart gadgets run on this operating system (like phones, tablets, and smartwatches). Every company organization or brand in the world nowadays need an e-commerce Android app in order to grow their business, money, and popularity. (Android, no date) Android is a very adaptive and engaging system, and a basic acquaintance takes less than an hour. Because there are so many essential programs accessible, any customer may easily configure OS settings. You can modify everything past recognition: if you don't like the look, symbols, or ringtone, simply go to the Google Play Store, download a significant program, and quickly customize everything to your desire

5.1.3 Dart Language

Dart is a client-oriented programming language that enables rapid app development across all platforms. Its primary goal is to create one of the most productive languages on a variety of platforms. Both the server and the user will benefit from it. The Dart SDK includes the Dart VM compiler as well as the dart2js tool, which generates the JavaScript version of a Dart Script so that it may be executed on sites that don't support Dart. It is a very similar object-oriented programming language to C++. Dart is a popular programming language for creating single-page websites and online apps. (Dart, no date). Dart is designed to provide logic and, as a result, a beautiful user interface. For mobile, desktop, and backend apps, compile specialized machine code. Alternatively, for web use, compile to JavaScript.

Benefits

The first advantage is that it is a very easy language to pick up. If any of us are familiar with C or C++, we can quickly learn this language because the grammar is comparable. The vast community, which provides extensive documentation, is the second item to note. And because it's a Google product, they make it as simple as possible. The third advantage is its excellent performance. Dart-based programs are faster to execute than JavaScript-based programs. (Code Carbon, 2020)

Drawbacks

Of course, we must consider the drawbacks in addition to the rewards. The first disadvantage is that it has limited resources, making it harder to find solutions to problems. This is due to a lack of larger and more cohesive development communities to assist you. Dart is currently in development. That's fantastic, and there's nothing "wrong" with Dart, but if you start writing in Dart today, the API may change, or things may not be fully or accurately documented, and the quantity of knowledge accessible on the web may be less than, example, the amount of information available on jQuery. (Code Carbon, 2020)

5.1.4 Google Firebase

Firebase is a Google program that allows developers to create apps for a variety of platforms, including iOS, Android, and web-based apps. Firebase is a quick tool for developing things that would take a long time in a traditional database system, such as bespoke APIs. Firebase has built-in APIs that allow us to construct applications quickly. One of the nicest features of Firebase is that it gives us tools for tracking business reports and experimenting with different goods.

The Firebase database is a real-time database that allows clients to immediately access data. Even if you are not connected to the internet, you can still use your program with all of your prior data sets.

Firebase allows users to use data in a secure and secured manner. The Firebase database is a NOSQL database, similar to the MYSQL database used in PHP applications. The real-time database API was created exclusively for rapid operations and capabilities.

5.1.5 Android Studio

It's a Google-created integrated development environment (IDE). The goal of Android is to speed up progress and make it easier for users to create high-quality apps for Android devices. Commonly used operating frameworks, such as Mac and Windows, allow variations of this IDE. It also provides a development kit and plugins for developers. cross-platform application support (IDE).

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA . On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Apply Changes to push code and resource changes to your running app without restarting your app
- Code templates and GitHub integration to help you build common app features and import sample code
- Extensive testing tools and frameworks
- Lint tools to catch performance, usability, version compatibility, and other problems
- C++ and NDK support

- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine.

5.2 Module Description

5.2.1 Home Screen

As our Home screen has following categories Shirts, shoes, and purse. Every product is associated with a specific category. We have made a dart file name as Angel_Home_Screen.Dart. In this file our main home categories are managed. Then we make a file name as prepare_data. In this file we make a class Data_View_Model.

In this class we are managing our all products, categories as well as cart. We are managing all the stuff in a single file in order to reduce no of files and optimization of code. In this class we make a method for each functionality. In this class we make get purse function in order to get all products in a purse. We make get shoe ad get shirts as well in a same logic.

We added loops for every category where are the products in a category are managed. If a category has 10 numbers of products, then loop move 10 time in order to show all the products for a specific category. As we use provider patterns for this application, so the main class extends change notifier. By using this provider pattern listen about the list that are prepared in this class. When any change occur in this class then it will notify to provider about change. Basically, this is the best thing in providers for managing data in application.

These packages are used in application to perform login and registration. Firebase core provide us an access to the flutter database. Firebase core also allow us to perform initialization actions. On the other hand, Firebase auth allow 25 us to authenticate user when they perform registration or Login. By using these packages, we can use firebase APIs for login or registration.

- firebase_core: ^1.16.0
- firebase_auth: ^3.3.17

5.2.2 Add to Cart Screen

In Add to cart screen we are using stateful widget. In Stateful widget we call its state. In Its state we are using scaffold widget. Scaffold is a layout of our whole screen which gives us many things like App Bar, Body and many more. We are using column widget to align in vertically direction. In column we have a children. In Children we are using list view. In list view widget all the selected items are arranged. Session check also included in add to cart functionality. If a user login to their account, then its session started then they will purchase product by using add to cart functionality. It is very good practice to use session in purchasing points to authenticate user.

We are fetching a data in our list view which index marked as true. In every item in list view, we are making edit and delete button. Below list view we are using elevation button to proceed to checkout. Elevation button has a on tap functionality. We set checkout page path in on tap functionality

5.3 SOURCE CODE

all_product_screen.dart

```
import 'package:e_commerce_flutter/core/app_color.dart';
import 'package:flutter/material.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:get/get.dart';
import '../../../core/app_data.dart';
import '../../controller/product_controller.dart';
import '../widget/product_grid_view.dart';

final ProductController controller = Get.put(ProductController());
```

```
class AllProductScreen extends StatelessWidget {
  const AllProductScreen({Key? key}) : super(key: key);

  PreferredSize _appBar(){
    return PreferredSize(
      preferredSize: const Size.fromHeight(100),
      child: SafeArea(
        child: Padding(
          padding: const EdgeInsets.symmetric(horizontal: 15, vertical: 10),
          child: Row(
            mainAxisAlignment: MainAxisAlignment.spaceBetween,
            children: [
              Container(
                margin: const EdgeInsets.all(8),
                decoration: BoxDecoration(
                    borderRadius: BorderRadius.circular(10),
                    color: AppColor.lightGrey),
                child: IconButton(
                  padding: const EdgeInsets.all(8),
                  constraints: const BoxConstraints(),
                  onPressed: () {},
                  icon:
                  const Icon(Icons.ac_unit_outlined, color: Colors.black),
                ),
              ),
              Container(
                margin: const EdgeInsets.all(8),
                decoration: BoxDecoration(
                    borderRadius: BorderRadius.circular(10),
                    color: AppColor.lightGrey),
                child: IconButton(
```

```
            padding: const EdgeInsets.all(8),
            constraints: const BoxConstraints(),
            onPressed: () {},
            icon: const Icon(Icons.search, color: Colors.black),
          ),
        )
      ],
    ),
  ),
 );
}

Widget _recommendedProductListView(BuildContext context){
 return  SizedBox(
  height: 170,
  child: ListView.builder(
    padding: const EdgeInsets.symmetric(vertical: 10),
    shrinkWrap: true,
    scrollDirection: Axis.horizontal,
    itemCount: AppData.recommendedProducts.length,
    itemBuilder: (_, index) {
     return Padding(
      padding: const EdgeInsets.only(right: 20),
      child: Container(
       width: 300,
       decoration: BoxDecoration(
        color: AppData.recommendedProducts[index]
          .cardBackgroundColor,
        borderRadius: BorderRadius.circular(15),
       ),
```

```
child: Row(
  children: [
    Padding(
      padding: const EdgeInsets.only(left: 20),
      child: Column(
        crossAxisAlignment:
        CrossAxisAlignment.start,
        mainAxisAlignment:
        MainAxisAlignment.center,
        children: [
          Text(
            '30% OFF DURING \nCOVID 19',
            style: Theme.of(context)
                .textTheme
                .headline3
                ?.copyWith(color: Colors.white),
          ),
          const SizedBox(height: 8),
          ElevatedButton(
            onPressed: () {},
            style: ElevatedButton.styleFrom(
              primary: AppData
                  .recommendedProducts[index]
                  .buttonBackgroundColor,
              elevation: 0,
              padding: const EdgeInsets.symmetric(
                  horizontal: 18),
              shape: RoundedRectangleBorder(
                borderRadius:
                BorderRadius.circular(18),
              ),
```

```
                    ),
                    child: Text(
                      "Get Now",
                      style: TextStyle(
                          color: AppData
                              .recommendedProducts[index]
                              .buttonTextColor!),
                    ),
                  )
                ],
              ),
            ),
            const Spacer(),
            Image.asset(
              'assets/images/shopping.png',
              height: 125,
              fit: BoxFit.cover,
            )
          ],
        ),
      ),
    );
  }),
);
}

Widget _topCategoriesHeader(BuildContext context){
 return Padding(
   padding: const EdgeInsets.only(top: 10),
   child: Row(
     mainAxisAlignment: MainAxisAlignment.spaceBetween,
```

```
      children: [
        Text(
          "Top categories",
          style: Theme.of(context).textTheme.headline4,
        ),
        TextButton(
          onPressed: () {},
          style: TextButton.styleFrom(primary: AppColor.darkOrange),
          child: Text(
            "SEE ALL",
            style: Theme.of(context)
                .textTheme
                .headline6
                ?.copyWith(color: Colors.deepOrange.withOpacity(0.7)),
          ),
        )
      ],
    ),
  );
}

Widget _topCategoriesListView(){
  return SizedBox(
    height: 50,
    child: ListView.builder(
      scrollDirection: Axis.horizontal,
      itemCount: controller.length,
      itemBuilder: (_, index) {
        return Padding(
          padding: const EdgeInsets.only(left: 5),
          child: GetBuilder<ProductController>(
```

```
          builder: (ProductController controller) {
            return Container(
              width: 50,
              height: 100,
              decoration: BoxDecoration(
                color:
                controller.categories[index].isSelected ==
                    false
                    ? const Color(0xFFE5E6E8)
                    : const Color(0xFFf16b26),
                borderRadius: BorderRadius.circular(10),
              ),
              child: IconButton(
                icon: FaIcon(controller.categories[index].icon,
                    color: controller
                        .categories[index].isSelected ==
                        false
                        ? const Color(0xFFA6A3A0)
                        : Colors.white),
                //  onPressed: controller.filterItemsByCategory,
                onPressed: () =>
                    controller.filterItemsByCategory(index),
              ),
            );
          },
        ),
      );
    },
  ),
);
}
```

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    extendBodyBehindAppBar: true,
    appBar: _appBar(),
    body: SafeArea(
      child: SingleChildScrollView(
        child: Padding(
          padding: const EdgeInsets.all(20),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Text(
                "Hello Sina",
                style: Theme.of(context).textTheme.headline1,
              ),
              Text(
                "Lets gets somethings?",
                style: Theme.of(context).textTheme.headline5,
              ),
              _recommendedProductListView(context),
              _topCategoriesHeader(context),
              _topCategoriesListView(),
              const SizedBox(
                height: 400,
                  child: ProductGridView())
            ],
          ),
        ),
```

```
        ),
      ),
    );
  }
}
```

cart_screen.dart

```
import 'package:e_commerce_flutter/core/extensions.dart';
import 'package:e_commerce_flutter/src/view/widget/animated_switcher_wrapper.dart';
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import '../../controller/product_controller.dart';
import '../../model/product.dart';
import '../widget/empty_cart.dart';

final ProductController controller = Get.put(ProductController());

class CartScreen extends StatelessWidget {
  const CartScreen({Key? key}) : super(key: key);

  PreferredSizeWidget _appBar(BuildContext context){
    return AppBar(
      title: Text(
        "My cart",
        style: Theme.of(context).textTheme.headline1,
      ),
    );
  }

  Widget cartListView(){
    return ListView.builder(
```

```
shrinkWrap: true,
padding: const EdgeInsets.all(20),
itemCount: controller.cartProducts.length,
itemBuilder: (_, index) {
  Product product = controller.cartProducts[index];
  return Container(
    margin: const EdgeInsets.only(bottom: 20),
    padding: const EdgeInsets.all(15),
    height: 120,
    decoration: BoxDecoration(
        color: Colors.grey[200]?.withOpacity(0.6),
        borderRadius: BorderRadius.circular(10)),
    child: Row(
      crossAxisAlignment: CrossAxisAlignment.end,
      children: [
        Container(
          padding: const EdgeInsets.all(5),
          decoration: BoxDecoration(
              borderRadius: BorderRadius.circular(10),
              color: ColorExtension.randomColor),
          child: ClipRRect(
            borderRadius:
            const BorderRadius.all(Radius.circular(20)),
            child: ClipRRect(
              borderRadius: BorderRadius.circular(10),
              child: Image.asset(
                product.images[0],
                width: 70,
                height: 120,
                fit: BoxFit.contain,
              ),
```

```
    ),
  ),
),
Padding(
  padding: const EdgeInsets.only(left: 10),
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    mainAxisAlignment:
    MainAxisAlignment.spaceEvenly,
    children: [
      Text(
        product.name.nextLine,
        maxLines: 2,
        overflow: TextOverflow.ellipsis,
        style: const TextStyle(
            fontWeight: FontWeight.w600,
            fontSize: 15),
      ),
      Text(
        controller.getCurrentSize(product),
        style: TextStyle(
            color: Colors.black.withOpacity(0.5),
            fontWeight: FontWeight.w400),
      ),
      Text(
        controller.isPriceOff(product)
            ? "\$${product.off}"
            : "\$${product.price}",
        style: const TextStyle(
            fontWeight: FontWeight.w900,
            fontSize: 23),
```

```
      ),
    ],
  ),
),
const Spacer(),
Container(
  decoration: BoxDecoration(
      color: Colors.white,
      borderRadius: BorderRadius.circular(10)),
  child: Row(
    children: [
      IconButton(
        splashRadius: 10.0,
        onPressed: () =>
            controller.decreaseItem(index),
        icon: const Icon(
          Icons.remove,
          color: Color(0xFFEC6813),
        ),
      ),
      GetBuilder<ProductController>(
        builder: (ProductController controller) {
          return AnimatedSwitcherWrapper(
            child: Text(
              '${controller.cartProducts[index].quantity}',
              key: ValueKey<int>(controller
                  .cartProducts[index].quantity),
              style: const TextStyle(
                  fontSize: 18,
                  fontWeight: FontWeight.w700),
            ),
```

```
                );
              },
            ),
            IconButton(
              splashRadius: 10.0,
              onPressed: () =>
                  controller.increaseItem(index),
              icon: const Icon(
                Icons.add,
                color: Color(0xFFEC6813),
              ),
            ),
          ],
        ),
      )
    ],
  ),
 );
 },
 );
}

Widget bottomBarTitle(){
 return Expanded(
  flex: 1,
  child: Container(
   padding: const EdgeInsets.symmetric(horizontal: 30),
   child: Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [
     const Text("Total",
```

```
            style:
            TextStyle(fontSize: 22, fontWeight: FontWeight.w400)),
          Obx(() {
            return AnimatedSwitcherWrapper(
              child: Text(
                "\$${controller.totalPrice.value}",
                key: ValueKey<int>(controller.totalPrice.value),
                style: const TextStyle(
                  fontSize: 25,
                  fontWeight: FontWeight.w900,
                  color: Color(0xFFEC6813),
                ),
              ),
            );
          })
        ],
      ),
    ),
  );
}

Widget bottomBarButton(){
  return Expanded(
    child: SizedBox(
      width: double.infinity,
      child: Padding(
        padding: const EdgeInsets.only(left: 30, right: 30, bottom: 20),
        child: ElevatedButton(
          child: const Text("Buy Now"),
          onPressed: controller.isEmptyCart ? null : () {},
        ),
```

```
      ),
    ),
  );
 }

 @override
 Widget build(BuildContext context) {
  return Scaffold(
    appBar: _appBar(context),
    body: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Expanded(
          flex: 10,
          child: !controller.isEmptyCart
              ? cartListView()
              : const EmptyCart(),
        ),
        bottomBarTitle(),
        bottomBarButton()
      ],
    ),
  );
 }
}
```

home_screen.dart

```
import 'package:animations/animations.dart';
import 'package:bottom_navy_bar/bottom_navy_bar.dart';
import 'package:e_commerce_flutter/src/view/screen/profile_screen.dart';
import 'package:flutter/material.dart';
```

```
import 'package:get/get.dart';
import '../../../core/app_data.dart';
import '../../controller/product_controller.dart';
import 'cart_screen.dart';
import 'favorite_screen.dart';
import 'all_product_screen.dart';

final ProductController controller = Get.put(ProductController());

class HomeScreen extends StatelessWidget {
  const HomeScreen({Key? key}) : super(key: key);

  static const List<Widget> screens = [
    AllProductScreen(),
    FavoriteScreen(),
    CartScreen(),
    ProfileScreen()
  ];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      bottomNavigationBar: Obx(
        () {
          return BottomNavyBar(
            itemCornerRadius: 10,
            selectedIndex: controller.currentBottomNavItemIndex.value,
            items: AppData.bottomNavyBarItems
                .map((item) => BottomNavyBarItem(
                    icon: item.icon,
                    title: Text(item.title),
```

```
                activeColor: item.activeColor,
                inactiveColor: item.inActiveColor))
            .toList(),
        onItemSelected: controller.switchBetweenBottomNavigationItems,
      );
    },
  ),
  body: Obx((){
    return PageTransitionSwitcher(
      duration: const Duration(seconds: 1),
      transitionBuilder: (
          Widget child,
          Animation<double> animation,
          Animation<double> secondaryAnimation,
          ) {
        return FadeThroughTransition(
          animation: animation,
          secondaryAnimation: secondaryAnimation,
          child: child,
        );
      },
      child: screens[controller.currentBottomNavItemIndex.value],
    );
  }),
 );
 }
}
```

product_detail_screen.dart

```
import 'package:e_commerce_flutter/core/app_color.dart';
import 'package:flutter/material.dart';
```

```
import 'package:flutter_rating_bar/flutter_rating_bar.dart';
import 'package:get/get.dart';
import 'package:smooth_page_indicator/smooth_page_indicator.dart';
import '../../controller/product_controller.dart';
import '../../model/product.dart';

final ProductController controller = Get.put(ProductController());

class ProductDetailScreen extends StatelessWidget {
  final PageController _pageController = PageController(initialPage: 0);

  final Product product;

  ProductDetailScreen(this.product, {Key? key}) : super(key: key);

  PreferredSizeWidget _appBar(BuildContext context){
    return AppBar(
      backgroundColor: Colors.transparent,
      elevation: 0,
      leading: IconButton(
        onPressed: () {
          controller.productImageDefaultIndex.value = 0;
          Navigator.pop(context);
        },
        icon: const Icon(
          Icons.arrow_back,
          color: Colors.black,
        ),
      ),
    );
  }
```

```
Widget productPageView(double width,double height){
  return  Container(
    height: height * 0.42,
    width: width,
    decoration: const BoxDecoration(
      color: Color(0xFFE5E6E8),
      borderRadius: BorderRadius.only(
        bottomRight: Radius.circular(200),
        bottomLeft: Radius.circular(200),
      ),
    ),
    child: Column(
      children: [
        SizedBox(
          height: height * 0.32,
          child: PageView.builder(
            itemCount: product.images.length,
            controller: _pageController,
            onPageChanged: controller.switchBetweenProductImages,
            itemBuilder: (_, index) {
              return FittedBox(
                fit: BoxFit.none,
                child: Image.asset(
                  product.images[index],
                  scale: 3,
                ),
              );
            },
          ),
        ),
```

```
      Obx(
          () => SmoothIndicator(
          effect: const WormEffect(
              dotColor: Colors.white,
              activeDotColor: AppColor.darkOrange),
          // ),
          offset: controller.productImageDefaultIndex.value
              .toDouble(),
          count: product.images.length),
      )
    ],
   ),
 );
}

Widget _ratingBar(BuildContext context){
 return    Row(
   children: [
    RatingBar.builder(
        initialRating: product.rating,
        direction: Axis.horizontal,
        itemBuilder: (_, index) {
          return const Icon(Icons.star,
              color: Colors.amber);
        },
        onRatingUpdate: (rating) {}),
    const SizedBox(width: 30),
    Text(
      "(4500 Reviews)",
      style: Theme.of(context)
          .textTheme
```

```
            .headline3
            ?.copyWith(fontWeight: FontWeight.w300),
      )
    ],
  );
}

Widget productSizesListView(){
  return ListView.builder(
    scrollDirection: Axis.horizontal,
    itemCount: controller.sizeType(product).length,
    itemBuilder: (_, index) {
      return InkWell(
        onTap: () {
          controller.switchBetweenProductSizes(
              product, index);
        },
        child: Container(
          margin:
          const EdgeInsets.only(right: 5, left: 5),
          alignment: Alignment.center,
          width:
          controller.isNominal(product) ? 40 : 70,
          decoration: BoxDecoration(
              color: controller
                  .sizeType(product)[index]
                  .isSelected ==
                  false
                  ? Colors.white
                  : AppColor.lightOrange,
              borderRadius: BorderRadius.circular(10),
```

```
            border: Border.all(color: Colors.grey,width: 0.4)),
          child: FittedBox(
            child: Text(
              //Map<String,bool>
              controller
                  .sizeType(product)[index]
                  .numerical,
              style: const TextStyle(
                  fontWeight: FontWeight.w500,
                  fontSize: 15),
            ),
          ),
        ),
      );
    },
  );
}

@override
Widget build(BuildContext context) {
  double height = MediaQuery.of(context).size.height;
  double width = MediaQuery.of(context).size.width;
  return SafeArea(
    child: Scaffold(
      extendBodyBehindAppBar: true,
      appBar: _appBar(context),
      body: SingleChildScrollView(
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            productPageView(width,height),
```

```
const SizedBox(height: 20),
Padding(
  padding: const EdgeInsets.all(20),
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Text(
        product.name,
        style: Theme.of(context).textTheme.headline2,
      ),
      const SizedBox(height: 10),
      _ratingBar(context),
      const SizedBox(height: 10),
      Row(
        children: [
          Text(
            product.off != null
                ? "\$${product.off}"
                : "\$${product.price}",
            //   style: const TextStyle(
            style: Theme.of(context).textTheme.headline1,

          ),
          const SizedBox(width: 3),
          Visibility(
            visible: product.off != null ? true : false,
            child: Text(
              "\$${product.price}",
              style: const TextStyle(
                decoration: TextDecoration.lineThrough,
                color: Colors.grey,
```

```
          fontWeight: FontWeight.w500,
        ),
      ),
    ),
    const Spacer(),
    Text(
      product.isAvailable
          ? "Available in stock"
          : "Not available",
      style: const TextStyle(fontWeight: FontWeight.w500),
    )
  ],
 ),
 const SizedBox(height: 30),
  Text(
   "About",
   style: Theme.of(context).textTheme.headline4,

 ),
const SizedBox(height: 10),
 Text(product.about),
const SizedBox(height: 20),
 SizedBox(
   height: 40,
   child: GetBuilder<ProductController>(
     builder: (ProductController controller) {
       return productSizesListView();
     },
   ),
 ),
const SizedBox(height: 20),
```

```
                SizedBox(
                  width: double.infinity,
                  child: ElevatedButton(
                    child: const Text("Add to cart"),
                    onPressed: product.isAvailable
                        ? () => controller.addToCart(product)
                        : null,
                  ),
                )
              ],
            ),
          )
        ],
      ),
    ),
  );
 }
}
```

5.5 Implementation

Flutter

This app is created with Flutter. Flutter is the latest framework in the world of mobile app development. Digging deeper here What is the Flutter framework, its strengths and weaknesses, and different ways to test Flutter application. The Flutter framework consists of both a software development kit (SDK) and its widget-based kit. UI library. This library consists of various reusable UI elements such as sliders, buttons, and text input. Developers who create mobile applications using the Flutter framework create using programming languages It is called darts. Using JavaScript-like syntax, Dart is a typed object programming language focused on front-end development. Flutter's high performance and productivity are achieved using several techniques. Unlike many other

popular mobile platforms, Flutter doesn't use JavaScript at all. Darts Programming language. Compiles into binary code, so it runs at native performance ObjectiveC, Swift, Java, or Kotlin. Flutter does not use native UI components. It may sound unpleasant at first. However, since the components are implemented in Flutter itself, there is no communication layer in between.

Firebase

The Firebase Framework that's used on this software is useful for constructing one of these database in which each admin with get admission to to the Google account in which firebase it set up, to replace the records and those adjustments are contemplated right away to each consumer. Firebase carrier handles maximum of the server-aspect paintings and there are numerous different factors that makes firebase a becoming preference as a unfastened BaaS for applications. These offerings encompass a real time database that's hosted on cloud in which the records is saved as JSON files, it additionally permits the builders to authenticate the consumer the use of emails and passwords, additionally firebase affords garage for all of the records this is generated through the consumer, which in our software is the approximately segment and all of the records entered throughout the registration of events.

We create the interface using flutter and it is then connected to the database using firebase. Using flutter to create the interface makes it possible to run on multiple platforms like iOS, android and web. The system is designed to provide a smoother experience to the customers using Firebase for backend, which is supported by Google Cloud. Customer management is made easier using customer profiles, wishlists, reorder functionality etc.

Deployment diagrams depicts about the whole process. We've used firebase as our database and for the development of application we've used flutter (dart).

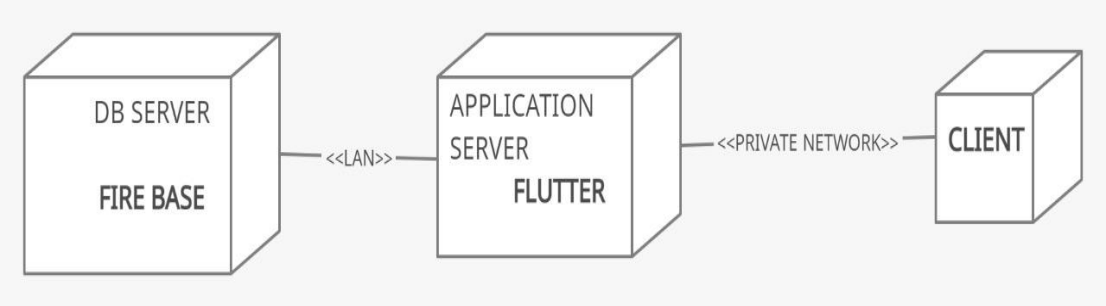


Fig 5.1 : Deployment diagram of E-Commerce Store

The main objective of the Online Ecommerce Store is to manage the details of Products, Customer, Payment, Delivery, Bills. The project is totally built at administrative end and thus only the administrator is guaranteed the access. *Cost savings*- The average online ecommerce store offers the consumer huge savings over traditional brick and mortar stores. *Convenience*- As with all Internet shopping, online stores offer great convenience to the consumer. *Greater variety*- Physical stores are typically limited on what they're able to stock in terms of space and budget; on the contrary, an online store is restricted by neither. The user can start shopping on the e-commerce website once he or she finishes the registration process. Features such as favorites, categories, add to cart etc. help develop a comfortable space for the customer.

5.6 Working

Code is written in Dart using firebase as database and to implement the code we've used android studio. Below figure is a snap of code running in android studio.

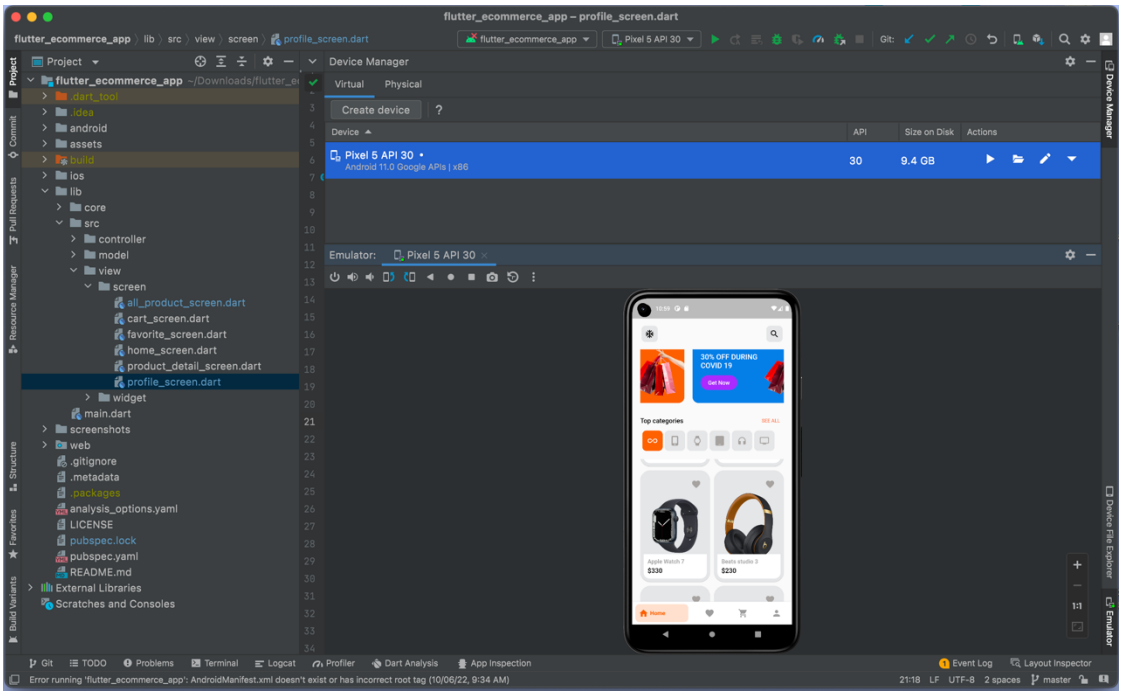


Figure 5.2 : Android Studio

```dart
class ProfileScreen extends StatelessWidget {
  const ProfileScreen({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Image.asset('assets/images/profile_pic.png'),
          const Text(
            "Hello Sina!",
            style: TextStyle(fontWeight: FontWeight.bold, fontSize: 25),
          ), // Text
          Row(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Image.asset(
                'assets/images/code.png',
                width: 60,
              ), // Image.asset
              const SizedBox(width: 10),
              style: TextStyle(
                fontSize: 20
              ),)
            ],
            ) // Row
          ],
        ), // Column
      ); // Scaffold
    }
}
```

Figure 5.3 : Flutter Dart Code

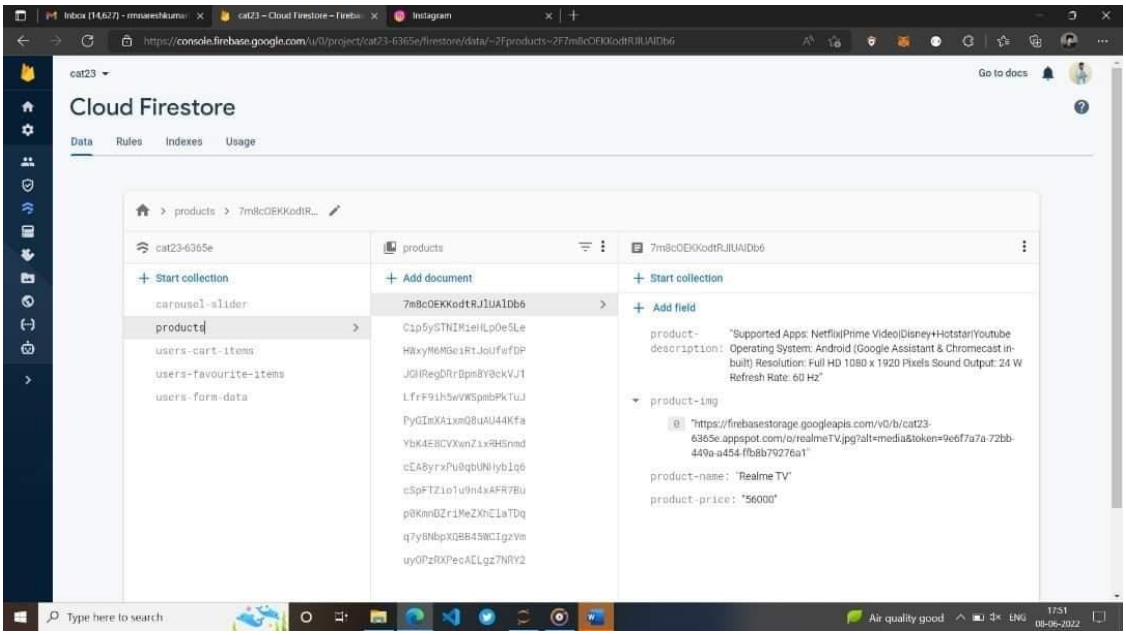


Figure 5.4 : Firebase Cloud

In cloud firestore of firebase, for each book we have created separate collection and each collection consists of all the features of the products listed on the e-commerce store.

CHAPTER 6
TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software manner. There are various types of test. Each test type addresses a specific testing requirement.

6.1 TESTING METHODOLOGIES:

The following are the Testing Methodologies:

- Unit Testing.
- Integration Testing.
- User Acceptance Testing.
- Output Testing.
- Validation Testing.

Unit Testing:

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module's control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit.

Integration Testing:

Integration testing addresses the issues associated with the dual problems of verification and program construction.

User Acceptance Testing (UAT):

It is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment. UAT is done in the final phase of testing after functional, integration and system testing is done.

Output Testing:

Output testing is a type of software testing whereby the system is tested against the functional requirements/specifications. Functions (or features) are tested by feeding them input and examining the output. ... Determine the output based on the function's specifications. Execute the test case.

Validation Testing:

Validation testing is the process of ensuring if the tested and developed software satisfies the client /user needs. The business requirement logic or scenarios have to be tested in detail. All the critical functionalities of an application must be tested here.

6.2 INTEGRATION TESTING:

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design. The following are the types ofIntegration Testing:

Top-Down Integration:

This method is an incremental approach to the construction of program structure. Modulesare integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner. In this

method, the softwareis tested from main module and individual stubs are replaced when the test proceeds downward.

Bottom-up Integration:

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubsis eliminated. The bottom-up integration strategy may be implemented with the followingsteps:

- The low-level modules are combined into clusters into clusters that perform a specific Software sub-function.
- A driver (i.e.) the control program for testing is written to coordinate test case input and output.

6.3 ACCEPTANCE TESTING:

Acceptance testing, a testing technique performed to determine whether or not the softwaresystem has met the requirement specifications. The main purpose of this test is to evaluatethe system's compliance with the business requirements and verify if it is met the requiredcriteria for delivery to end users. There are various forms of acceptance testing:

- User acceptance Testing
- Business acceptance Testing
- Alpha Testing
- Beta Testing

This is a type of testing done by users, customers, or other authorized entities to determineapplication/software needs and business processes. Description: Acceptance testing is the most important phase of testing as this decides whether the client approves the application/software

CHAPTER 7

OUTPUT SCREENS

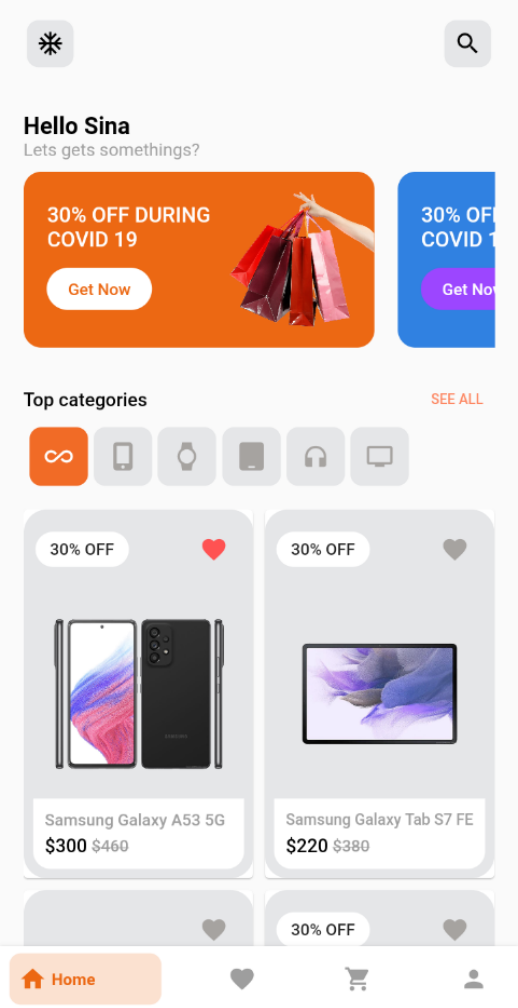


Figure 7.1 : Home Screen

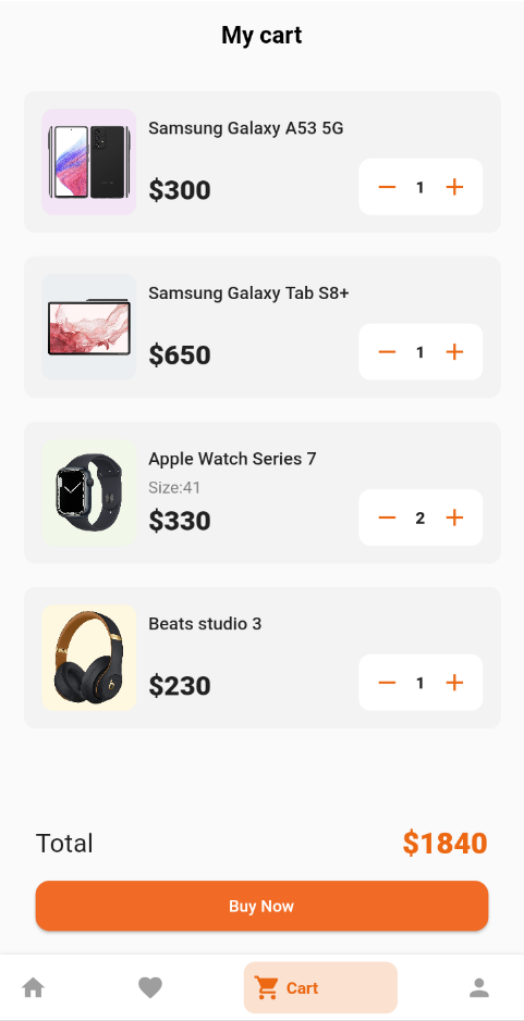


Figure 7.2 : Cart Screen

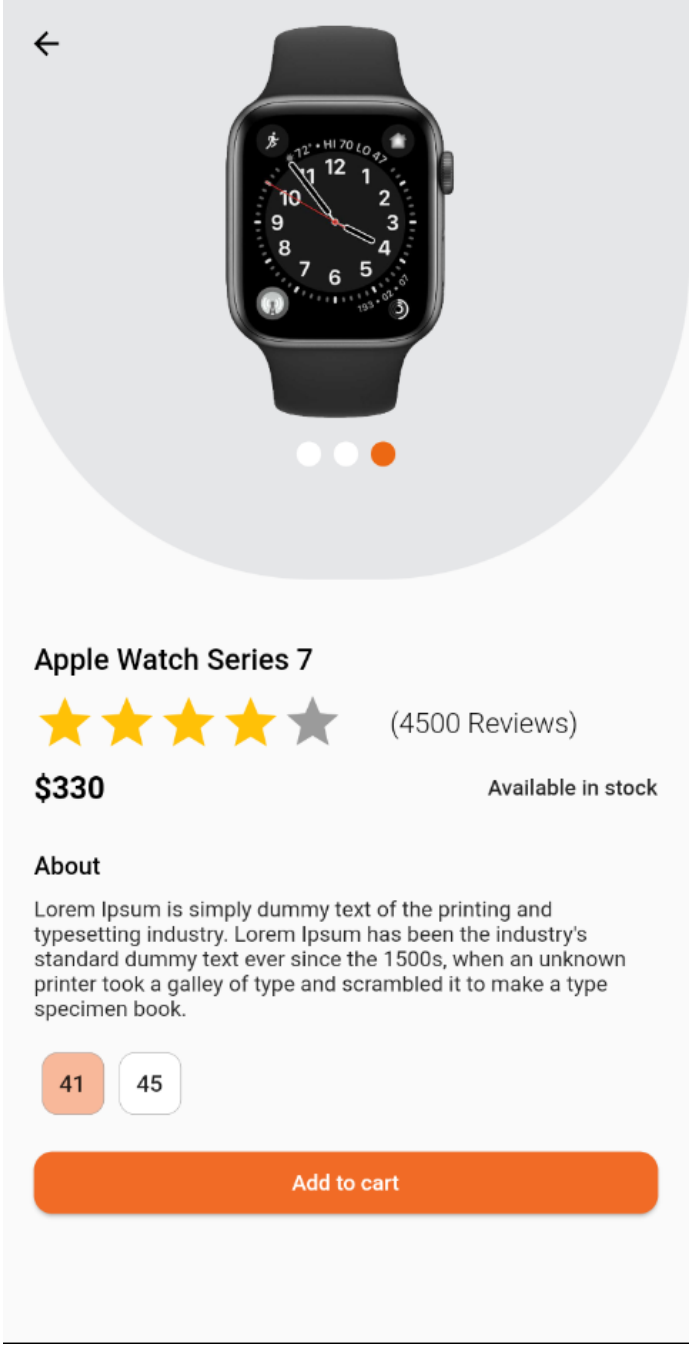


Figure 7.3 : Product Screen

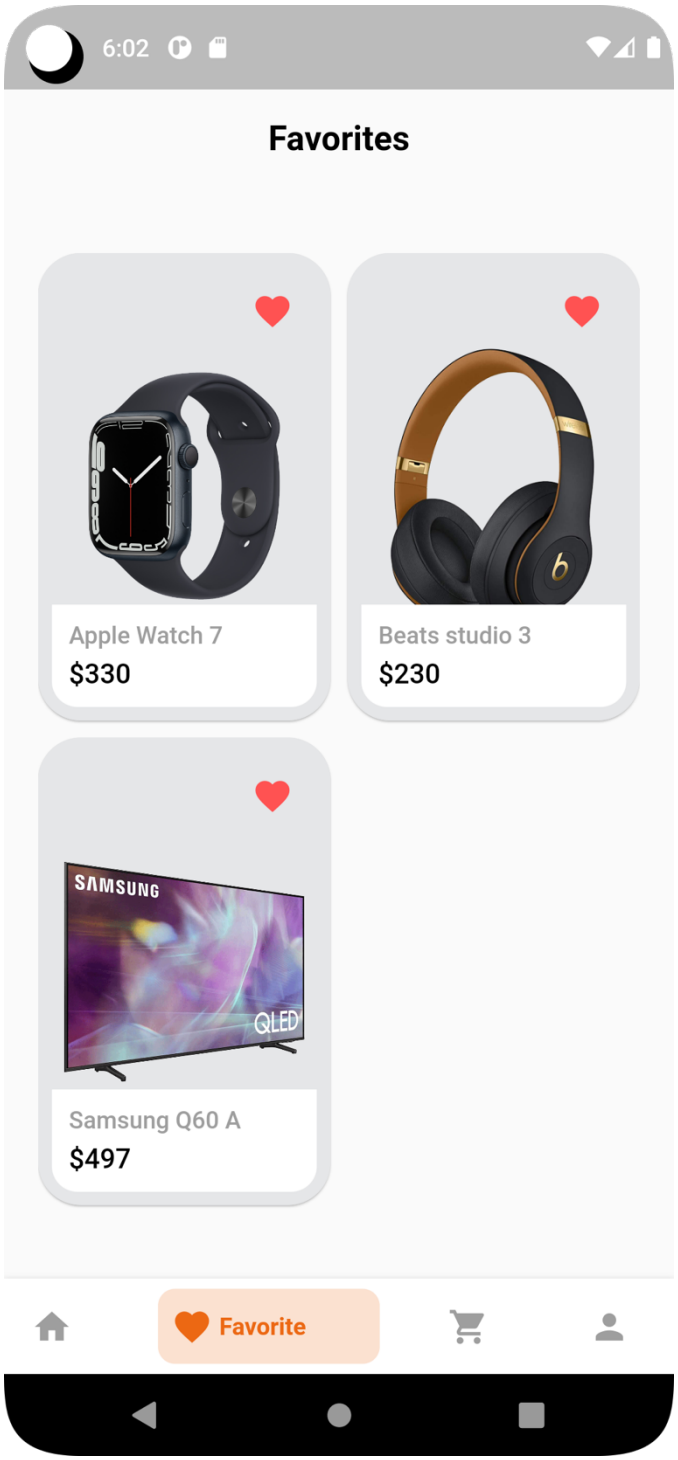


Figure 7.4 : Favourites Screen

CHAPTER 8
CONCLUSION

UI is simple and minimalistic, understandable for everyone. It greatly saves the time. We can give more advance software for E-Commerce Store including more facilities. Wishlists, product classification, numerous sorts of filters, payments, customer administration, and backend are all aspects that each online business needs. The e-commerce app is designed to meet all of these objectives, saving our consumers important time and providing a wide range of alternatives in one place. Flutter, a multi-platform supporting framework, was used to create the interface. Also, because a user is required to scroll through a large number of goods before making a purchase, this application requires a lot of data transfer. Users will enjoy a quicker and smoother experience thanks to Firebase's backend, which is powered by Google Cloud. The basic purpose of e-commerce is to reach as many people as possible at the correct moment in order to boost sales and profitability. Finally, the system is implemented and tested according to test cases. We will attempt to advance this application as it could contact an ever-increasing number of individuals.

CHAPTER 9

FUTURE ENHANCEMENT

Several future enhancements can be made to the system. Online purchasing has grown increasingly popular in recent years. People all over the world prefer to purchase online rather than go to the store, which is why Amazon and other similar online stores are not only profitable but also generating interest in online business. Many new strategies and technologies have been created in e-commerce apps to offer users with information. Online commerce is experiencing considerable development all around the world. We've provided answers, but they'll need some time and resources. This App will provide users with modules to help them with their task. The player's major goal in building the app was to prioritize quality above quantity. As previously indicated, more modules and functionality will be introduced soon to make the program more solid and adaptable. The primary goal and aim of this program are to enable and assist people in using an e-commerce-based platform, which is widely utilized throughout the world for online shopping and purchase of commodities and products nowadays. As a result, individuals may enjoy, conveniently access, and purchase all things through internet shopping and feel fully integrated into society. Our effort will not end here; we will continue to improve the software and provide new features for those who are unable to perceive the world from any angle (completely blind people). We'll aim to include a feature that makes this app totally accessible to blind people; they'll be able to control it with their voice and dress themselves according to their preferences. We'll be adding additional features as time goes on.

CHAPTER 10
REFERENCES

[1] B. J. Crha and R. V. Rusnak, “Comparison of Technologies for Multiplatform Mobile Applications Development,” 2020.

[2] S. Dmitrii, “STATE MANAGEMENT APPROACHES IN FLUTTER,” 2020.

[3] J. M. C. da and S. Penim, Online grocery shopping: An exploratory study of consumer decision making processes, 2013.

[4] N. Katuk, T. Jayasangar, and Y. Yusof. Design and Development of Smart List: A Mobile App for Creating and Managing Grocery Lists, Baghdad Science Journal, vol. 16, pp. 462-476, 2019

[5] Nazmun Nessa Moon , Shaheena Sultana, Fernaz Narin Nur & Mohd Saifuzzaman. 2017. A Literature Review of the Trend of Electronic Commerce in Bangladesh Perspective.

[6] J. Polaski. We Know You Want It: Perspectives on Predictive Shopping, Honors Thesis in Management Bridgewater State University, 2015

[7] The Flutter way. 2020. Online Shop App - Flutter UI - Speed Code. Published on 03.07.2020.

[8] W. A. Harsha Jayawilal and S. Premeratne, "The smart shopping list: An effective mobile solution for grocery list-creation process," *2017 IEEE 13th Malaysia International Conference on Communications (MICC)*, 2017, pp. 124-129, doi: 10.1109/MICC.2017.8311745.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:06/June-2022 Impact Factor- 6.752 www.irjmets.com

CLOUD BASED E-COMMERCE APPLICATION WITH FLUTTER AND FIREBASE

Nerella Tarun Reddy*[1], Prodduturi Vivek Sai*[2], Balusu Meher Chaitanya*[3], Dr. G. Arun Sampaul Thomas*[4], Dr. P. Srinivasa Rao*[5]

*[1,2,3]Student, Department of Computer Science and Engineering, JB Institute of Engineering and Technology, India

*[4]Associate Professor, Department of Computer Science and Engineering, JB Institute of Engineering and Technology, India

*[5]HOD, Department of Computer Science and Engineering, JB Institute of Engineering and Technology, India.

ABSTRACT

Any online business, requires features such as wishlist, products categorization, various types of filters, payments, customer management and backend. E-commerce app is designed to satisfy all these requirements so that it saves the valuable time of our customers and gives a wide variety of options to choose from at one stop. The interface is designed using flutter, a multi-platform supporting framework. Also, this application involves a lot of data transfer, as a user is expected to scroll through a lot of products before purchasing. Firebase, backed by Google Cloud for backend, makes it a faster and smoother experience for the users. The primary goal of e-commerce is to reach maximum customers at the right time to increase sales and profitability of the business.

I. INTRODUCTION

E-commerce is quickly becoming a recognised and widely utilised business model. More and more companies are giving capabilities for conducting commercial transactions using applications. It is fair to state that the procedure of purchasing on apps is getting more frequent. With the introduction of these services, many new start-ups may now readily offer their products and services online. The majority of small company owners utilise web interfaces to take their operations online. Using a cross-platform app will help them reach a larger percentage of returning clients. The present small-scale internet retailers lack a user-friendly interface. There are several limits to the existing system, however solutions can be supplied in the future.

II. LITERATURE SURVEY

Flutter

This app is created with Flutter. Flutter is the latest framework in the world of mobile app development. Digging deeper here What is the Flutter framework, its strengths and weaknesses, and different ways to test Flutter application. The Flutter framework consists of both a software development kit (SDK) and its widget-based kit. UI library. This library consists of various reusable UI elements such as sliders, buttons, and text input. Developers who create mobile applications using the Flutter framework create using programming languages It is called darts. Using JavaScript-like syntax, Dart is a typed object programming language focused on front-end development. Flutter's high performance and productivity are achieved using several techniques. Unlike many other popular mobile platforms, Flutter doesn't use JavaScript at all. Darts Programming language. Compiles into binary code, so it runs at native performance ObjectiveC, Swift, Java, or Kotlin. Flutter does not use native UI components. It may sound unpleasant at first. However, since the components are implemented in Flutter itself, there is no communication layer in between.

Firebase

The Firebase Framework that's used on this software is useful for constructing one of these database in which each admin with get admission to to the Google account in which firebase it set up, to replace the records and those adjustments are contemplated right away to each consumer. Firebase carrier handles maximum of the server-aspect paintings and there are numerous different factors that makes firebase a becoming preference as a unfastened BaaS for applications. These offerings encompass a realtime database that's hosted on cloud in

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:06/June-2022 Impact Factor- 6.752 www.irjmets.com

which the records is saved as JSON files, it additionally permits the builders to authenticate the consumer the use of emails and passwords, additionally firebase affords garage for all of the records this is generated through the consumer, which in our software is the approximately segment and all of the records entered throughout the registration of events.

III. PROPOSED SYSTEM

We create the interface using flutter and it is then connected to the database using firebase. Using flutter to create the interface makes it possible to run on multiple platforms like iOS, android and web. The system is designed to provide a smoother experience to the customers using Firebase for backend, which is supported by Google Cloud. Customer management is made easier using customer profiles, wishlists, reorder functionality etc.

Deployment diagrams depicts about the whole process. We've used firebase as our database and for the development of application we've used flutter (dart).

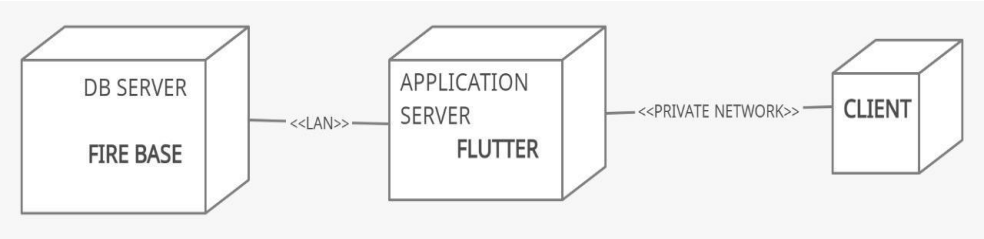


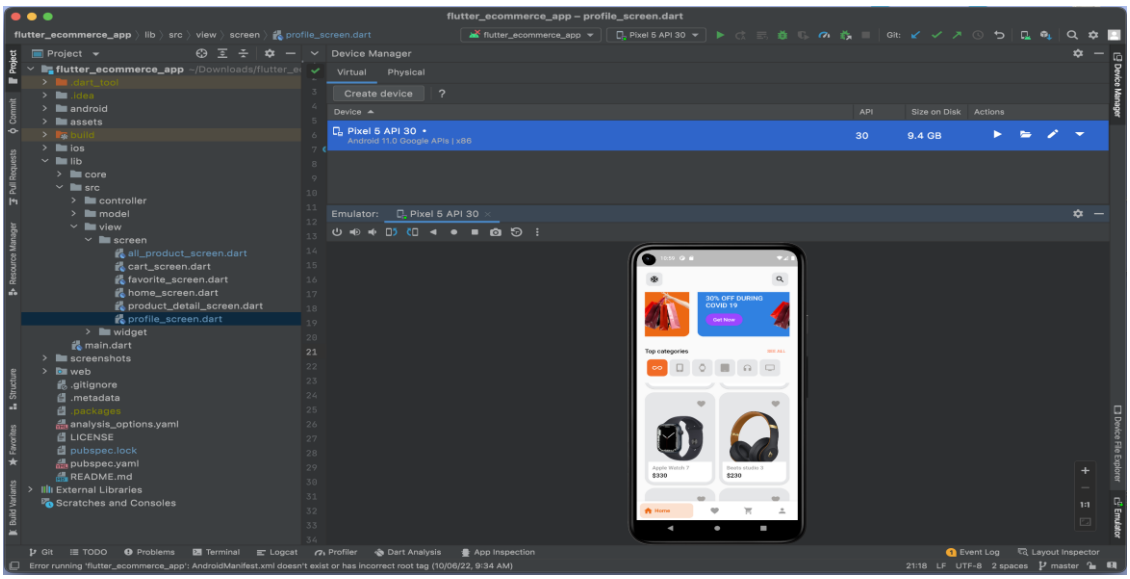
Fig 3.1 Deployment diagram of E-Commerce Store

IV. PROJECT DESCRIPTION

The main objective of the Online Ecommerce Store is to manage the details of Products, Customer, Payment, Delivery, Bills. The project is totally built at administrative end and thus only the administrator is guaranteed the access. Cost savings- The average online ecommerce store offers the consumer huge savings over traditional brick and mortar stores. Convenience- As with all Internet shopping, online stores offer great convenience to the consumer. Greater variety- Physical stores are typically limited on what they're able to stock in terms of space and budget; on the contrary, an online store is restricted by neither. The user can start shopping on the e-commerce website once he or she finishes the registration process. Features such as favorites, categories, add to cart etc. help develop a comfortable space for the customer.

V. PROJECT IMPLEMENTATION

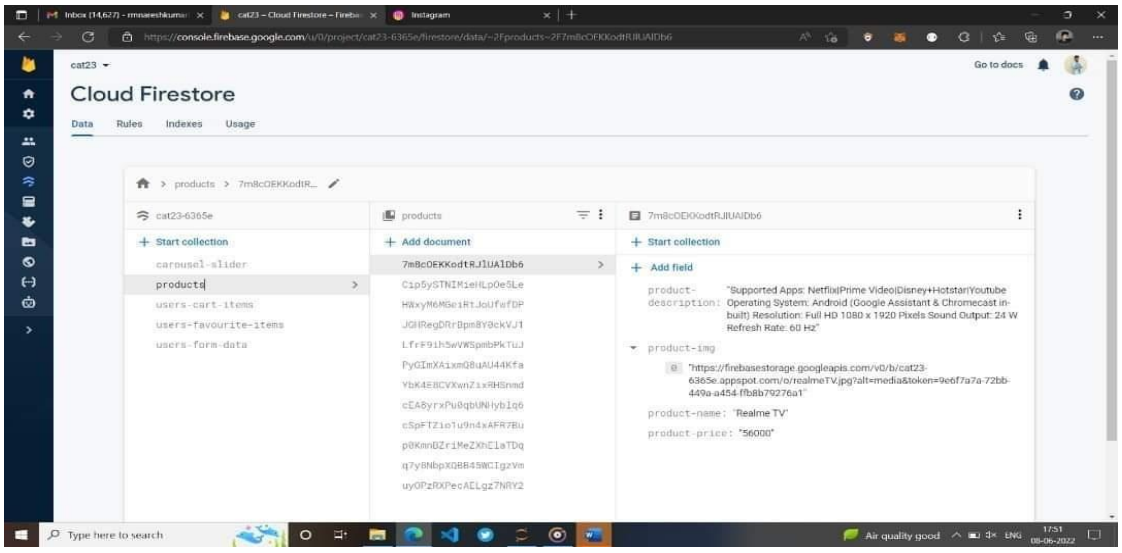
Code is written in Dart using firebase as database and to implement the code we've used android studio. Below figure is a snap of code running in android studio.



e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:06/June-2022 Impact Factor- 6.752 www.irjmets.com

```dart
class ProfileScreen extends StatelessWidget {
  const ProfileScreen({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Image.asset('assets/images/profile_pic.png'),
          const Text(
            "Hello Sina!",
            style: TextStyle(fontWeight: FontWeight.bold, fontSize: 25),
          ), // Text
          Row(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Image.asset(
                'assets/images/code.png',
                width: 60,
              ), // Image.asset
              const SizedBox(width: 10),
              style: TextStyle(
                fontSize: 20
              ),)
            ],
          ) // Row
        ],
      ), // Column
    ); // Scaffold
  }
}
```



In cloud firestore of firebase, for each book we have created separate collection and each collection consists of all the features of the products listed on the e-commerce store.

VI. CONCULSION

UI is simple and minimalistic, understandable for everyone. It greatly saves the time. We can give more advance software for E-Commerce Store including more facilities. Finally, the system is implemented and tested according to test cases. We will attempt to advance this application as it could contact an ever-increasing number of individuals.

VII. REFERENCES

[1] B. J. Crha and R. V. Rusnak, “Comparison of Technologies for Multiplatform Mobile Applications Development,” 2020.
[2] S. Dmitrii, “STATE MANAGEMENT APPROACHES IN FLUTTER,” 2020.
[3] J. M. C. da and S. Penim, Online grocery shopping: An exploratory study of consumer decision making processes, 2013.
[4] N. Katuk, T. Jayasangar, and Y. Yusof. Design and Development of Smart List: A Mobile App for Creating and Managing Grocery Lists, Baghdad Science Journal, vol. 16, pp. 462-476, 2019
[5] J. Polaski. We Know You Want It: Perspectives on Predictive Shopping, Honors Thesis in Management Bridgewater State University, 2015
[6] W. A. Harsha Jayawilal and S. Premeratne, "The smart shopping list: An effective mobile solution for grocery list-creation process," 2017 IEEE 13th Malaysia International Conference on Communications (MICC), 2017, pp. 124-129, doi: 10.1109/MICC.2017.8311745.