



E-Commerce application with Flutter framework

Adnan Naseer

BACHELOR'S THESIS
May 2022

Bachelor of Engineering
Software Engineering

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Bachelor of Engineering
Software Engineering

Adnan Naseer:
E-COMMERCE APPLICATION WITH FLUTTER FRAMEWORK

Bachelor's thesis 29 pages
May 2022

The intention of this thesis project was to develop a mobile e-commerce app. While developing this platform, the present pandemic dilemma was borne in mind. After pondering about all these concerns, it was determined to create a user-friendly e-commerce application that allows users to make their life easier by purchasing online. The app was made utilizing Android Studio and the flutter framework. For design purposes, GitHub was used.

The application architecture was built using the MVVM paradigm, which incorporates all the well-known provider patterns. The databases were built with Google Firebase and were developed to authenticate current users or allow customers to generate an id by signing up. The free source platforms were used to gather all the product information. Further development ideas were also examined.

The project launched with only a rudimentary grasp of the dart language. Many obstacles and disappointments arose along the road, all of which served as excellent learning opportunities. A better appreciation of how to structure a redevelopment, how to better organize and manage the codebase, and what incremental benefits can be gained by utilizing the flutter framework all seem to be vital insights.

Key words: E-commerce, application development, flutter, google firebase

CONTENTS

1	INTRODUCTION	5
2	BACKGROUND AND PURPOSE	6
	2.1 Project Background.....	6
	2.2 Project Objectives	6
	2.3 Comparative Analysis	7
3	ANDROID APPLICATION REQUIREMENTS AND DEVELOPMENT ..	8
	3.1 Ecommerce Application Development	8
	3.2 Android App development.....	9
	3.3 Functional Requirements	9
	3.4 Non-Functional Requirements	10
	3.5 GUI (Graphical User Interface).....	10
4	TECHNOLOGY.....	12
	4.1 Flutter.....	12
	4.2 Android.....	13
	4.3 Dart Language	13
	4.4 Google Firebase.....	14
	4.5 Android Studio.....	15
5	TECHNICAL IMPLEMENTATION.....	16
	5.1 Planning	16
	5.2 MVVM (Model-View-View Model).....	16
	5.3 Configuration and Customization	17
	5.3.1 Flutter Firebase working	18
	5.3.2 Firebase Configuration for Android and IOS:.....	19
6	CODE EXPLANATION	21
	6.1 Assistance from the open sources and friend	21
	6.2 Whole Application background Function calling	21
	6.3 Home Screen	23
	6.4 Login Registration Screen.....	24
	6.5 Add to Card Screen.....	25
7	Discussion and Further Development.....	26
	REFERENCES	27

ABBREVIATIONS AND TERMS

TAMK	Tampere University of Applied Sciences
OS	Operating system
iOS	iPhone operating system
API	Application Programming Interface
SDK	Software development kit
Firebase	Database management service from Google. Here you can store all your online data.
C and C++	Programming languages used for software development
JavaScript	A web-based programming language
Json	JavaScript Object Notation
Array	An object that contains multiple values.
Array index	the order of a data in an array
Init function	Init function runs for the initialization of the software
Stateful widget	A stateful widget describes the part of user interface
Stateless widget	the state which cannot be changed while the program is running
Query	A method for requesting a data from database

1 INTRODUCTION

In this thesis, the Flutter technology is utilized to develop an e-commerce mobile application. E-commerce, or electronic commerce, is a technique of purchasing goods and services through the internet that is growing in popularity and effectiveness for online businesses. Ecommerce offers a number of benefits, the most significant of which is that it is inexpensive and convenient.

As a consequence, we may describe e-commerce as the process of buying and selling goods and services through the internet, whether inside a country or across the globe. These companies have their own mobile apps or websites where customers may explore a range of products and services.

They have a choice of payment alternatives if they purchase anything and need to pay for it, including (credit card, debit card, PayPal). utilizing a website or a smartphone app the order is shipped to the customer's address after confirmation. If required, customers can follow the status of their orders. All of these things happen on the internet. (A Review of Narrative Literature of e-commerce, 2017). This application's main purpose and goal is to empower and assist a person in accessing e-commerce platforms so that they may do their own online shopping.

2 BACKGROUND AND PURPOSE

2.1 Project Background

Electronic commerce, generally known as online business and trade, alludes to the acquiring and selling of products and services well over internet. In the 1960s, the term "ecommerce" was coined. With the growing usage of mobile devices, social media has emerged as an effective reinforcement of the web page's strength. (salesforce)

Ecommerce (electronic commerce) is the term used to describe all online activity along with the purchase and sale of goods and services. To put it differently, ecommerce is the process of doing transactions over the internet. Ecommerce has existed since 1994, when Phil Branden Berger bought the first product online, Sting's "Ten Summoner's Tales." By 2021, an estimated 27.2 percent of the world's population will be shopping online. (Oberlo, no date)

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. (salesforce)

2.2 Project Objectives

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

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. (sgrlaw)

2.3 Comparative Analysis

Existing Solutions	Proposed Solution
Assists in the improvement of our standard, carefully crafted products for distribution within the showcase.	More attractive user interface.
The framework allows a buyer to quickly explore or search for an item.	Faster Server-to-client-side response.
It provides an easy way to purchase items directly from sellers without the use of a third-party interface	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 ANDROID APPLICATION REQUIREMENTS AND DEVELOPMENT

3.1 Ecommerce Application Development

The history of e-commerce dates back more than 50 years, when early inventions such as EDI (Electronic Data Interchange) and teleshopping were shown in the 1970s, paving the way for today's e-commerce companies. The history of e-commerce is inextricably linked to the history of the internet. When the internet was invented in the 1990s, online shopping became possible. Since then, Amazon has become one of the most popular e-commerce sites in the United States, with hundreds of e-commerce enterprises springing up as a result. (Everydayonlinemarketing, no date)

Michael Aldrich demonstrated e-shopping by connecting a phone connection to an adjustable TV and a transaction-processing computer. (Aldricharchive, no date)

While desktop spending has remained stable throughout the year, the worldwide market for online business transactions has grown at an annual rate of 33.7 percent since 2016. This happens as a result of customers' continuous preparation with each screen tap. The creation of mobile applications for e-commerce enterprises appears to be a long-term trend, according to simple reasoning. Nothing suggests that the worldwide public will soon reject smartphones. Diverse internet shopping, on the other hand, is predicted to reach \$3.5 trillion by 2021. (Shopify, no date)

The e-Commerce business is experiencing the most notable growth of adaptable arrangements being prompted by the need to meet client needs in the context of rising online sales. Customers want the ability to purchase items and services at any time and from any location, without having to travel. Customers choose various e-commerce applications or websites to purchase things because they advance faster, give more comfort, and are more flexible.

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 touch-based 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. (creativebloq, no date)

3.2 Android App development

Android app development is the process of creating mobile apps for devices using the Android operating system. Languages like C, object-oriented languages (like Java), and the SDK (Android Software Development Kit) are used to create Android applications. Android was initially introduced in 2009, and it is based on the Java programming language. In today's world, Android dominates the whole portable app sector, with the majority of the market share and a global clientele.

3.3 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.

3.4 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.

3.5 GUI (Graphical User Interface)

Main App
splash
screen and
user login
screen

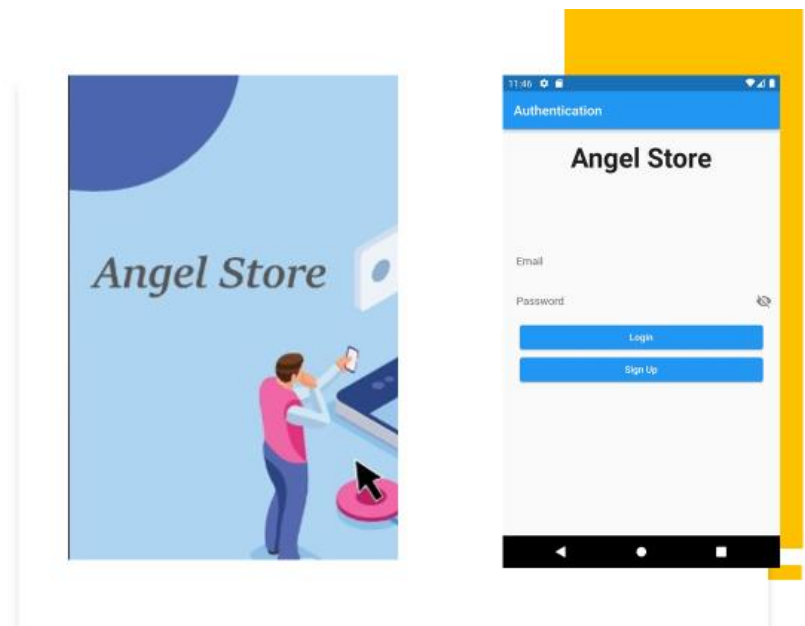


Figure 1: Customer login and splash screen

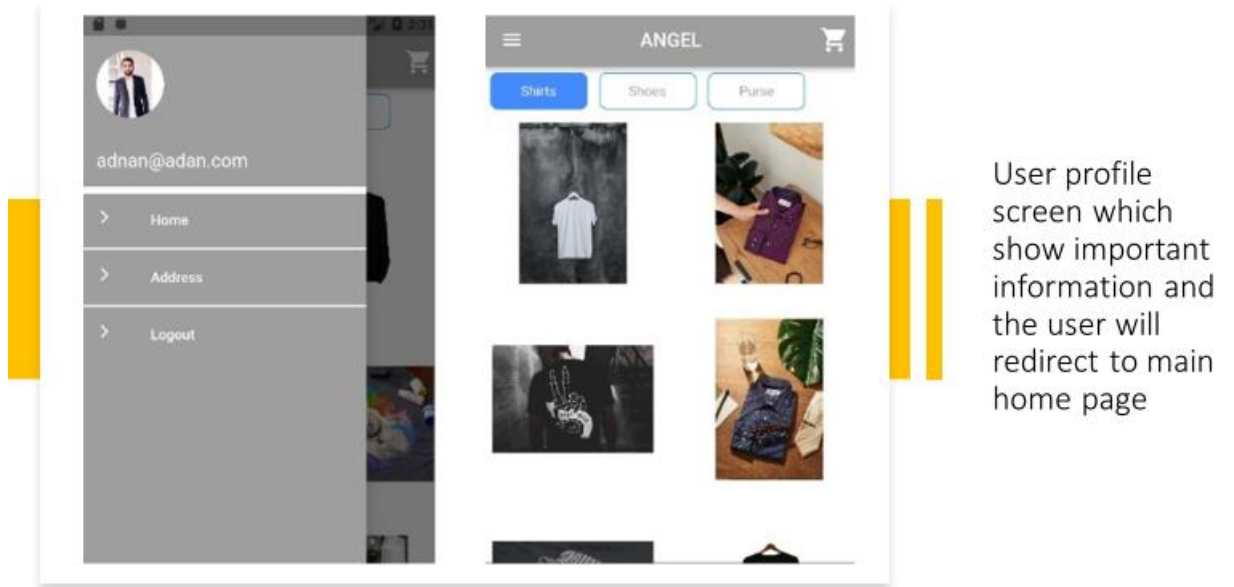


Figure 2: Profile and main screen

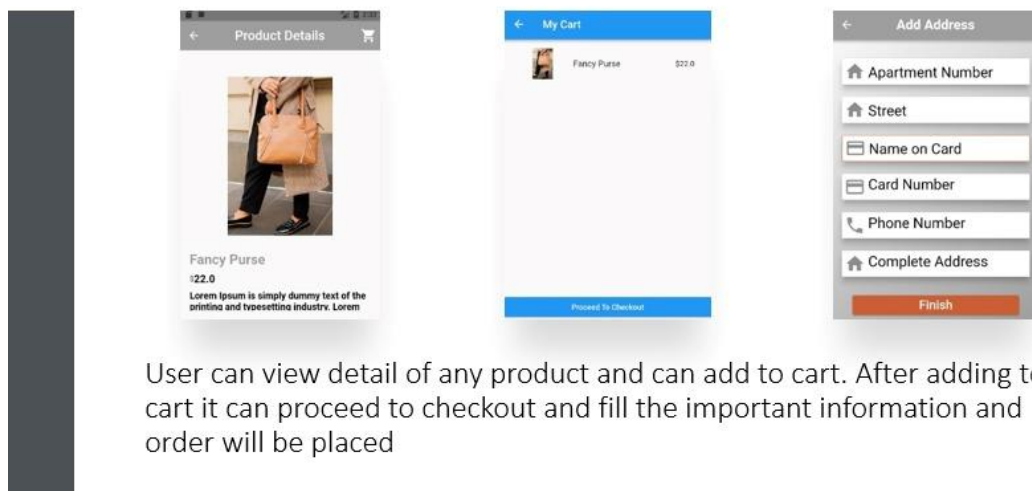


Figure 3: Order processing

4 TECHNOLOGY

4.1 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 one-time 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. (AppDividend, no date)

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.

4.2 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

4.3 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)

Drawback

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)

4.4 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. (Firebase, no date).

4.5 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).

5 TECHNICAL IMPLEMENTATION

5.1 Planning

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:

- Will the development take place in stages or all at once?
- How much time will it take?
- Will there be new security challenges?

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.

5.2 MVVM (Model-View-View Model)

The app's current design is built on the MVVM paradigm, which employs a well-known provider pattern. Views in this design closely monitor the view Model and react to any changes that are detected. We have a home-view that is watching Data View Model, and the Data View Model has a reference to the model in our example.

In the Data View Model, we have a method for filling categories that is responsible for adding categories to the list of categories and informing the listener, much like populating categories. The Home-view has a Horizontal List View widget that monitors the category list and updates the content of the list View whenever the list is modified.

The same goes for categorizing items and filling them. The Data View Model has three methods: `getShoes()`, `getPurse()`, and `getShirts()`. When we choose a

category, it queries the items in that category and alerts the Grid view widget, which oversees displaying products. The Selected Product reference variable in the DataViewModel is responsible for displaying product details on the product details page.

When we pick a product from the Home Page, we update the reference variable for that product, which is visible in the product details page, and we alter the content of the detail page appropriately. In DataViewModel, we additionally keep track of the listOfCartProducts. We update the listOfCartProducts in DataViewModel when we add a product from the details page to the cart. This variable is tracked in the cart Screen and updates the listView widget when the list changes.

Furthermore, we manage users using Firebase authentication. Orders may only be placed by authenticated users. We built a project for Angel for firebase configuration, and under Project, we developed an app for Android and iOS based on the package name of Android and the bundle id of iOS, respectively. For Android and iOS, Firebase generates a google-services.json and info.plist file, accordingly. We placed the files in the appropriate directories and then implemented the authentication system using the Firebase APIs.

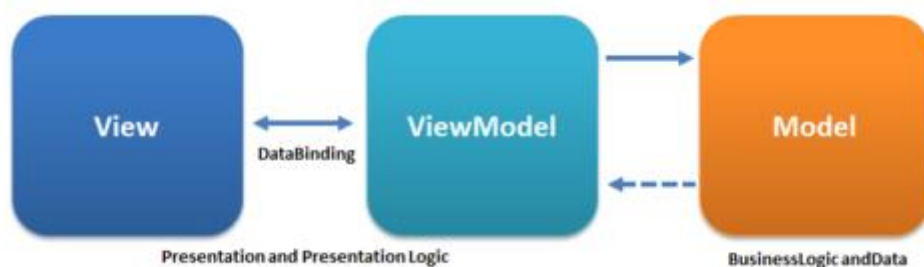


Figure 4: Model–view–viewmodel

5.3 Configuration and Customization

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.

5.3.1 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.

5.3.2 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 in our application.

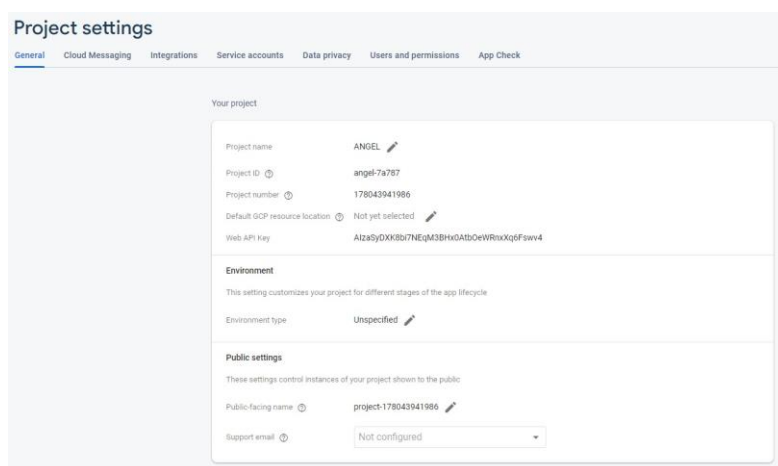


Figure 5: Firebase project settings

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.

6 CODE EXPLANATION

6.1 Assistance from the open sources and friend

I received design assistance from one of my mates as well as other sources. The Shipping details page, intro screens, and Splash screen were created by a friend, and the design was pulled from site on the internet and integrated into one.

We got a reasonably decent open-source code from GitHub for designing part of the code and used parts of the code from that open source and made adjustments to it by watching YouTube tutorials to add some other components as well. (MonsieurZbanowanYY, no date)

We used the cart component with the help of YouTube instead of the search bar on the top right side of the application. We've also created a shipping detail section where we ask for buyer information such as address, phone number, and so on. (The Flutter Way, 2020)

6.2 Whole Application background Function calling

When a user opens an app for the first time, we set a background function to call once when the app is opened by a new user. It has 3 to 4 guides to keep users entertained. When the home screen is opened, it calls the home screen widget, which then calls the state widget to receive product data and display it in a grid on the home screen.

Navigator is a tool that allows us to navigate data by category. When a user taps on a product, we acquire its array index and retrieve the desired object detail. When a user taps on a product (which are taken from an open source), the product's details screen opens. On the detailed screen, we match a product's id with the array and display data when the id matches the array index. (Unsplash)

On the detail screen, we have fields for description, price, add to cart, and buy now. It will be added to the cart when the user clicks on add to cart. When a user clicks the buy now button, they will be taken to the checkout page. If the user is

logged in, he or she can place an order; if not, the user must first log in to their account, then register for an app, and then log in.

```

void login() {
  AuthenticationHelper()
    .signIn(email: usernameController.text, password:
passwordController.text)
    .then((result) {
      if (result == null) {
        // Navigator.pop(context);
        Navigator.pushReplacement(context,
          MaterialPageRoute(builder: (context) => CardScreen()));

        ScaffoldMessenger.of(context).showSnackBar(SnackBar(
          content: Text(
            "Login Successfully",
            style: TextStyle(fontSize: 16),
          ),
        ));
      } else {
        ScaffoldMessenger.of(context).showSnackBar(SnackBar(
          content: Text(
            result,
            style: TextStyle(fontSize: 16),
          ),
        ));
      }
    });
}
}

```

Figure 6: Login Code

When the user adds products to the cart, he or she will check the cart to see which items have been added. He/she will inspect the items and proceed to checkout from here. When he/she taps to check out again, the login session is

checked, and if he/she is not logged in, they must first log in before placing an order.

We also display products underneath the details screen of each product, allowing the user to select products from the same category. User interface processes are managed by libraries. We just display a Thank You screen after the order is placed. Because our program has numerous screens, we will discuss each screen's workflow as well as the packages that are included in order to explain the code.

6.3 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 and 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.

In our application we are using `init` function. `init` function we are calling our `Data_View_Model` class. Everything is prepared before running. Let's suppose

you open an application. You can see only 1 category products by default. When we click on other category notifier note the change and send to the listener that is call in init() function. Then home screen data index is updated and across index specific category products are call and display in a home screen.

6.4 Login Registration Screen

In login and registration page 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 text fields. One for email and other for password.

We have used `signInWithEmailAndPassword` and for the registration we have used `createUserWithEmailAndPassword` API's.

When a user enters email and password. A query is generated which checks whether registered on firebase database or not. If credential are match it return a true and we redirect it into the screen where they enter detail.

When a user login to their account, we save its email and give it to unique key. When they press logout, its key will be removed and when they try to login again we will give them a new key again. In this way we authenticate our user.

We are managing session as well when a login to their account session start and when user logout an application it will be over. We are managing session by using shared preferences in flutter. Shared preference allow us to manage user session with a specific interval of time. We have added a package in pub spec in order to use in flutter.

`shared_preferences: ^2.0.13`

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

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
```

6.5 Add to Card 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.

7 Discussion and Further Development

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.

REFERENCES

Abilitynet. 2013. How to use TalkBack in Android Jelly Bean. Read on 25.05.2022.

<https://mcmw.abilitynet.org.uk/talkback-android-jelly-bean-4-2-2>

Android Accessibility Help. 2022. Get started on Android with TalkBack. Read on 11.05.2022.

<https://support.google.com/accessibility/android/answer/6283677?hl=en>

AppDividend. no date. Flutter Mobile Development Guide For Beginners. Read on 08.05.2022.

<https://appdividend.com/category/flutter/>

Code Carbon. 2020. Major Advantages And Disadvantages Of Dart Language. Read on 15.05.2022.

<https://codecarbon.com/pros-cons-dart-language/>

Collin, C. 2021. 5 Critical Steps for Your Software Implementation Plan. Read on 11.05.2022.

<https://www.softwareadvice.com/resources/software-implementation-plan/>

Dart. No date. Dart programming language | Dart. Read on 05.05.2022.

<https://dart.dev/>

Darrell, R. 2011. The Future of Shopping. Harvard Business Review. Read on 08.05.2022.

<https://hbr.org/2011/12/the-future-of-shopping>

Everyday. No date. Essential reads: Where marketing meets strategy - Everyday Online Marketing. Read on 05.05.2022.

<https://everydayonlinemarketing.com/>

Firebase. No date. Firebase documentation: Learn the fundamentals. Read on 01.05.2022.

<https://firebase.google.com/docs>

Flutter. no date. Flutter: Build apps for any screen. Read on 10.05.2022.

<https://flutter.dev/>

GeeksforGeeks. No date. A computer science portal for geeks. Read on 06.05.2022.

<https://www.geeksforgeeks.org>

Google. 2022. Everything you need to build on Android. Read on 24.05.2022.

<https://developer.android.com/studio/features>

James, H. 2022. What is Java? Definition, Meaning & Features of Java Platforms. Read on 12.05.2022.

<https://www.guru99.com/java-platform.html>

Lockwood, B. no date. 10 Technology Challenges. Read on 03.05.2022.

<https://sgrlaw.com/ttl-articles/10-technology-challenges/>

Michael Aldrich Archive. No date. Welcome to the Michael Aldrich Archive. Read on 09.05.2022.

<https://www.aldricharchive.co.uk/>

MonsieurZbanowanYY. 2022. E-Commerce-App-UI-Flutter. Read on 27.05.2022.

<https://github.com/MonsieurZbanowanYY/E-Commerce-App-UI-Flutter>

Nazmun Nessa Moon , Shaheena Sultana, Fernaz Narin Nur & Mohd Saifuzzaman. 2017. A Literature Review of the Trend of Electronic Commerce in Bangladesh Perspective. Read on 23.05.2022.

https://globaljournals.org/GJMBR_Volume17/3-A-Literature-Review-of-the-Trend.pdf

Oberlo. 2022. FULFILL YOUR DROPSHIPPING DREAMS. Read on 21.05.2022.

<https://www.oberlo.com/>

Pravine, V. no date. A 5-STEP PLAN FOR SOFTWARE IMPLEMENTATION SUCCESS. Read on 18.05.2022.

<https://www.chetu.com/blogs/technical-perspectives/software-implementation-plan.php>

Richard F. Schmidt. 2013. Chapter 19 - Software Implementation. Read on 25.05.2022.

<https://www.sciencedirect.com/science/article/pii/B9780124077683000197>

Salesforce, Inc. 2022. CRM Software; Cloud Computing Solutions - Salesforce EMEA. Read on 05.05.2022.

<https://www.salesforce.com/eu/>

Santex. No Date. 7 Steps for Software Implementation Success. Read on 20.05.2022.

<https://santexgroup.com/blog/7-steps-for-software-implementation-success/>

Shopify. No Date. Shopify: Start and grow your e-commerce business. Read on 17.05.2022.

<https://www.shopify.com>

The Flutter way. 2020. Online Shop App - Flutter UI - Speed Code. Published on 03.07.2020. Referred on 23.05.2022.

<https://www.youtube.com/watch?v=XBKzpTz65lo>

Unsplash . no date. Beautiful Free Images & Pictures | Unsplash. Read on 27.05.2022.

<https://unsplash.com/>

Wikipedia contributors. 2022. Model–view–viewmodel. In Wikipedia, The Free Encyclopedia. Read on 05.05.2022.

<https://en.wikipedia.org/w/index.php?title=Model%E2%80%93view%E2%80%93viewmodel&oldid=1088169504>