

A Project Report

on

JustShop : An E - Commerce Mobile Application

*Submitted in partial fulfilment of the
requirement for the award of the degree of*

Bachelor of Technology in Computer Science and Engineering with specialisation in Cyber Security & Computer Networks



(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under the supervision of

Mr. V. Arul

Assistant Professor

Submitted By

Bhavya Dubey

18SCSE1140009

Pratul Pant

18SCSE1140012

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA**

December, 2021



**SCHOOL OF COMPUTING SCIENCE AND
ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA**

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the project, entitled “**JustShop: An E-Commerce Mobile Application**” in partial fulfillment of the requirements for the award of the Bachelor of Technology in Computer Science and Engineering with specialisation in Cyber Security and Computer Networks submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of July 2021 to December 2021, under the supervision of Mr. V. Arul, Assistant Professor, Department of Computer Science and Engineering, of School of Computing Science and Engineering, Galgotias University, Greater Noida

The matter presented in the project has not been submitted by me/us for the award of any other degree of this or any other places.

Bhavya Dubey, 18SCSE1140009

Pratul Pant, 18SCSE1140012

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Mr. V. Arul

Assistant Professor

CERTIFICATE

The Final Project Viva-Voce examination of Bhavya Dubey, 18SCSE1140009, and Pratul Pant, 18SCSE1140012, has been held on _____ and his/her/their work is recommended for the award of Bachelor of Technology in Computer Science and Engineering with specialisation in Cyber Security and Computer Networks.

Signature of Examiner(s)

Signature of Supervisor(s)

Signature of Project Coordinator

Signature of Dean

Date:

Place: Greater Noida

Abstract

Today, the consumers have got the very taste of shopping everything online, making the E-Commerce industry the fastest growing industry, especially during the Covid-19 Pandemic. Users in India are now familiar with online payments and delivery systems which creates the opportunities for new comers in the industry to make way. However, mobile phone is still the most preferred device to shop online. With more users turning to their smartphone for getting the experience to shop from anywhere, it is vital to bring changes to the de-facto systems, and make development in the mobile application interfaces. So, it is important to bring some new ideas and brands into the E-Commerce Industry in order to leverage the growing opportunities in the field. JustShop is an E-commerce mobile application focussing on targeting audience interested in online shopping with various features such as better and convenient User Interface, faster delivery time, as well as features keep track of products and other entities of the app. It also gives user a benefit to checkout blogs related to trending products and filter their reviews and buy the products at the same place. All in all, it is an E-Commerce App for a business to reach out to its customers while making use of the technology of mobile applications.

List of Figures

Figure No.	Figure Name	Page No.
1	Use Case Diagram	12
2	Sequence Diagram	14
3	Activity diagram	16
4	Layers of Flutter	20
5	Splash Screen	21
6	Login Screen	26
7	Signup Screen	39
8	Cart Screen	53
9	Product Screen	53
10	Home Screen	53
11	Intro Screen	54
12	Wishlist Screens	54
13	Blog Screen	54

Acronyms

B-2-B	Business to Business
E-commerce	Electronic Commerce
ERP	Enterprise Resource Planning
SAP	Systems, Applications & Products in Data Processing
UML	Unified Modelling Language
NPM	Node Package Manager

Title		Page No.
Candidate's declaration		ii
Certificate		iii
Abstract		iv
List of Figures		v
Acronyms		vi
Chapter 1	Introduction	1-2
Chapter 2	Literature Survey	3-7
	2.1 Existing Solution	7
	2.2 Proposed Solution	8
Chapter 3	Feasibility Analysis	9
	3.1 Financial Feasibility	9
	3.2 Technical Feasibility	9
	3.3 Resource Feasibility	10
	3.4 Time Feasibility	10
	3.5 Technology Risks Feasibility	10
	3.6 Social/Legal Feasibility	10
Chapter 4	Project Design	11
	4.1 UML Diagrams	11
	4.1.1 Use Case Diagram	12
	4.1.2 Sequence Diagram	14
	4.1.3 Activity Diagram	15-16
Chapter 5	Requirements & Tools of the Project	17
	5.1 Requirements	17
	5.1.1 Hardware Requirements for Development	17
	5.1.2 Software Requirements for Development	
	5.1.3 Hardware Requirements for Use	17
	5.1.4 Software Requirements for Use	
	5.2 About the tools used	18
	5.2.1 Dart or React	18
	5.2.2 Flutter Framework	18
	5.2.3 Firebase	18-19
		19-20
Chapter 6	Functionality & Implementation	21
Chapter 7	Conclusion and Future Scope	55
	References	56
	Publication(Communicated)	57

CHAPTER-1

Introduction

The term “Electronic commerce” (or e-Commerce) refers to the use of an electronic channel to carry out commercial transactions or to buy or sell any products or services. Most of the time, it refers to the sale of products via the Internet, but the term eCommerce also covers purchasing mechanisms via Internet. A consumer that buys a product/service on the Internet is called a cyber consumer. E-Commerce is not only limited to online sales, but also covers:

- Preparing ratings online
- User consultation
- Electronic catalog supply
- Sales access plan
- Real-time product availability (stock)
- Electronic payment service
- Delivery tracking
- Post sales service

Lastly, in the case of electronic services and products (MP3 files, software programs, e-books, etc.), Electronic marketing makes purchases very short-term, if not immediate.

Most of the online shopping sites are online stores with at least the following office level:

An online electronic catalog that lists all products for sale, its prices and sometimes its availability (product in stock or number of days before delivery);

A search engine that makes it easy to find a product easily through search processes (product type, price list, keyword, etc.);

1. Virtual caddy system (sometimes called virtual cart): This is the heart of the e-commerce system. Visible caddy makes it easy to track client purchases along the way and change the prices of each reference;

2. Secure online payment (accounting) is usually guaranteed by a trusted third party (bank) through secure transactions;

3. Order tracking system, which allows tracking of order processing and sometimes provides details of package delivery by the sender.

4. The back office system allows an online retailer to edit his or her online offerings, change prices, add or remove product references and manage client orders.

Apart from this, the mobile application for Electronic commerce nowadays need to have a robust framework in order to work properly, and provide the optimistic user satisfaction. To fulfil this, the apps need to be developed quickly and on all the possible platforms. Platforms or Operating System that are currently in use in India on mobile phones are Android, and iOS. For Android, one can use Android Studio to leverage Java Technology, XML, PHP, Python for backend, React Native, etc. For iOS, one needs to use Xcode to build application using Swift programming language.

Working on two polar technologies can be extremely time consuming and inefficient, especially when good and fast results are needed. To overcome this, we are using Flutter technology, which was introduced by Google in May 2017. It uses Dart programming language to create cross-platform applications, which means the same code base works for iOS as well as Android.

CHAPTER-2

Literature Survey

In 2010, the United Kingdom had the highest per capita e-commerce spending in the world. As of 2013, the Czech Republic was the European nation where e-commerce delivers the biggest commitment to the enterprises' complete revenue. Right around a quarter (24%) of the nation's complete turnover is generated through the online channel. Among emerging economies, China's e-commerce presence continues to expand every year. With 668 million Internet users, China's online shopping sales reached \$253 billion in the primary portion of 2015, representing 10% of all out Chinese consumer retail sales in that period. The Chinese retailers have been able to help consumers feel more comfortable shopping online. E-commerce exchanges between China and other countries increased 32% to 2.3 trillion yuan (\$375.8 billion) in 2012 and accounted for 9.6% of China's complete international trade. In 2013, Alibaba had an e-commerce market share of 80% in China. In 2014, there were 600 million Internet users in China (twice as numerous as in the US), making it the world's biggest online market. China is additionally the largest e-commerce market on the planet by value of sales, with an estimated US\$899 billion in 2016. In 2013, Brazil's e-commerce was developing rapidly with retail e-commerce sales expected to develop at a double-digit pace through 2014. By 2016, eMarketer expected retail e-commerce sales in Brazil to reach \$17.3 billion. India has an Internet user base of around 243.2 million as of January 2014. Despite being third largest user base in world, the penetration of Internet is low compared to markets like the United States, United Kingdom or France however is developing at a lot faster rate, adding around 6 million new entrants every month. [1]

In India, money down is the most preferred payment method, collecting 75% of the e-retail activities. E-Commerce has become a significant apparatus for little and large businesses worldwide, not exclusively to sell to customers, yet in addition to engage them. In 2012, ecommerce sales topped \$1 trillion without precedent for history.[1] Mobile devices are assuming an increasing part in the blend of eCommerce, this is additionally normally called mobile commerce, or m-commerce. In 2014, one estimate saw purchases made on mobile devices making up 25% of the market by 2017.[1]

Economists have theorized that e-commerce should lead to intensified price competition, as it increases consumers' capacity to gather data about items and prices. Research by four economists at the University of Chicago has tracked down that the development of online shopping has likewise affected industry structure in two areas that have seen huge development in e-commerce, bookshops and travel agencies. Generally, larger firms are able to use economies of scale and offer lower prices. The lone exception to this pattern has been the very smallest category of bookseller, shops with between one and four employees, which appear to have withstood the trend.[2] Depending on the category, e-commerce might move the exchanging costs—procedural, relational, and monetary—experienced by customers.

Individual or business involved in e-commerce whether buyers or sellers rely on Internet-based technology to achieve their exchanges. E-commerce is recognized for its capacity to permit business to communicate and to frame exchange anytime and anyplace. Whether an individual is in the US or overseas, business can be conducted through the internet. The power of e-commerce permits geophysical barriers to disappear, making all consumers and businesses on earth potential customers and suppliers. [2] In this manner, exchanging barriers and exchanging expenses may shift. eBay is a genuine example of e-commerce business people and businesses are able to post their items and sell them around the Globe.

In e-commerce activities, store network and coordinations are two most significant components need to be considered. Commonly, cross-border coordinations need about few weeks time round. Based on this low efficiency of the production network service, customer fulfillment will be greatly reduced. Some researcher stated that consolidating e-commerce competence and IT setup could well enhance organization's overall business worth.[1] Other researcher stated that e-commerce need to consider the establishment of warehouse centers in foreign countries, to create high efficiency of the coordinations system, improve customers' fulfillment, yet additionally can improve customers' steadfastness.

E-commerce has the ability to integrate all inter-organization and intra-organization capacities, meaning that the three flows (physical flow, financial flow and data flow) of the store network could be likewise affected by e-commerce. The affections on actual streams improved the method of item and inventory movement level for companies. For the data streams, e-commerce optimized the limit of data processing than companies used to have, and for the monetary streams, e-commerce permits companies to have more efficient payment and settlement solutions.

Moreover, e-commerce affects supply chains: Firstly, the performance hole will be eliminated since companies can identify holes between different levels of supply chains by electronic means of arrangements; Secondly, as a result of e-commerce emergence, new capabilities such implementing ERP systems, like SAP ERP, Xero, or Megaventory, have helped companies to manage operations with customers and suppliers. [3] Yet these new capabilities are as yet not completely exploited. Thirdly, technology companies would keep investing on new e-commerce software arrangements as they are expecting investment return. [3] Fourthly, e-commerce would help to solve numerous aspects of issues that companies might feel hard to cope with, for example, political barriers or crosscountry changes. At long last, e-commerce provides companies a more efficient and effective way of teaming up with each other inside the production network.

Alongside the e-commerce and its unique appeal that has appeared step by step, virtual enterprise, virtual bank, network marketing, online shopping, payment and advertising, such this new jargon which is unheard-of and presently has become as recognizable to people. This reflects that the e-commerce gigantically affects the economy and society from the other side. For instance, B2B is a quickly developing business on the planet that leads to lower cost and afterward improves the economic efficiency and furthermore bring along the development of employment.

E-commerce is definitely not a new industry, technically speaking, yet it is creating a new economic model. A great many people agree that e-commerce will positively affect economic society in the future, yet in its early stages its effects are hard to gauge. Some have noted that e-commerce is a kind of incorporeal revolution.[1] E-commerce has numerous social benefits: one, the expense of maintaining an e-commerce business is very low when compared with running an actual store; two, there is no rent to pay on expensive premises; and three, business processes are simplified and less worker hours are required to maintain a normal business easily. In the area of law, education, culture and furthermore strategy, e-commerce will continue to rise in sway. [2]E-commerce will really take people into the data society.

Management Electronic commerce efforts need to be managed properly, and because of the interdisciplinary nature of E-Commerce, its management might require new approaches and theories. Business Law and Ethics Legal and ethical issues are extremely significant in E-Commerce, especially in a worldwide market. [3]A large number of legislative bills are pending, and numerous ethical issues are interrelated with legal ones, like security and intellectual property. Others Several other disciplines are involved in different aspects of E-Commerce to a lesser extent-for example, phonetics (interpretation in international trades), mechanical technology and sensory systems, operations research/management science, insights, and public approach and organization. Additionally, E-Commerce is of Interest to engineering, health care,

E-business is the process of directing business on the Internet. Its scope includes purchasing and selling as well as services, satisfying the needs of customers and teaming up with business partners. Business to business e-commerce is savvy business. The chance for business to business e-commerce is even greater. A wholesaler might sell items to the retailer. There are advanced e-commerce software which support multi-tier valuing. [3]This helps to set up online stores to offer preferred estimating to some vendors and shared price to others. This includes internet-enabled initiatives of an enterprise to frame commercial linkages with another enterprise, dealer, warehouse or manufacturer. In this type of e-commerce, e paperwork and

time-to-market get immeasurably reduced.[3] All through the world, this ecommerce mode is the biggest.

In a B2B exchange, the interaction is between businesses. For example, a website that is getting for the steel business may have office for buyers and sellers to list their requirements and post their items. It helps them in rapidly shutting the exchanges and the buyer can get quality, material and can choose from different suppliers. [4]B2B commerce is a developing business in the e-commerce arena-with the increasing use of the internet, more and more business are realizing the commercial advantage of giving business clients a streamlined and easy manner to order items or service online. It facilitates access to the ordering process to just those with whom a concern has a commercial relationship. Business to Business e-commerce provides small and medium enterprises (SMEs) with an excellent chance to access new markets, improve customer service and reduce costs.[4] And keeping in mind that hurdles exist, they ought to be viewed more as speed breakers rather than street barriers.

B2C is the most well known type of e-commerce, wherein the people are directly involved in B2C e-commerce, and businesses use the internet for offering their items or services 24 hours per day through worldwide access. The sites Amazon.com and Rediff are among these. These websites spell products directly to consumers over the Internet. The two way accessibility feature of the internet enables operating companies to ascertain consumer preferences and purchasing trends directly

2.1 Existing Solution:

- **This system helps in enhancing products to distribute in the market:** The existing applications reach to a worldwide audience in order to offer good products to the global customers.
- **The system helps buyers to search the item on their finger tips:** The existing applications do not require the buyer to move to offline stores, and buy products from the shop itself. But rather it allows them to shop from anywhere and anytime which earlier was restricted as shops

were sometimes far away, and they offered a specific time to visit. Whereas now, 24x7 orders can be placed.

- **It provides the easy way of purchasing the item directly from the merchants without the interface of any outsider, especially in Covid-19 pandemic:** In the view of the covid-19 pandemic, the whole world could not go to shops to even get the essentials such as fruits vegetables and medicines, however E-commerce apps like Amazon, Myntra, PharmEasy, etc allowed customers to shop from the safety of their homes.
- **It additionally helps the country people to sell their items to the market and to earn money to improve their status:** While customers could sell their items to people in their own countries, they were also able to sell outside the country, improving the condition of their economy and increasing the export.
- **It did have a review system, but not a way to give influencers and blogs a way to buy products:** The existing apps offered a review system where customers could put a review on the same page where the purchase is happening, but it was more like comment section for the product. People who are putting the review had no as such profit.

2.2 Proposed Solution:

- **In the proposed system, influencers and reviewers get proper credit:** JustShop app allows influencers and reviewers to put reviews in the form of blogs so that people or potential customers can see those blogs and buy the products from the same place, so that the reviewers also get credit for putting time and efforts into the process, and be a part of the community.
- **Attempt to make the country people to feel free to fill the description about their items:** All people will be free to post a review about a product or service offered by the JustShop app. This will make the platform free and independent of any discrimination, and allow people from everywhere to share their opinions and ideas and present it in any way they like.

CHAPTER-3

Feasibility Analysis

A feasibility study is an analysis that takes the entire project's applicable elements into consideration—including economic, technical, legal, and scheduling considerations—to learn the probability of finishing the project successfully.

Project managers use feasibility studies to recognise the upsides and downsides of undertaking a project before they contribute a great deal of time and money into it. They can likewise furnish a company's management with critical or confidential data that could prevent the company from entering aimlessly into any risky businesses.

Feasibility studies are essential to business advancement. They can permit a business to address where and how it will work. They can likewise distinguish potential impediments that may block its activities and perceive the measure of financing it should get the business ready for action.

Plausibility reads target promoting techniques that could help persuade financial specialists or banks that putting resources into a specific venture or business is a savvy decision. Ultimately, feasibility means how practical the solution to the problem is.

Having a look at the feasibility analysis from different perspectives of consideration for the project JustShop.

3.1 Financial Feasibility

Being a mobile application, most of the software requirements are open source and hence very low. The only burden would be of the hardware. At the initial stage it would be using a small dataset which could be easily stored on the cloud database called the fire base.

When the database grows ,if a larger institution uses it then the cloud cost will be added. But it is not much and can be easily taken from the user/client by the name of premium membership.

From these it is clear that the e-commerce app for exams is feasible financially.

3.2 Technical Feasibility

This mobile application requires Android Studio, Flutter SDK , XCode, and all these software are easily available.

The languages required are Dart, jQuery and jsPDF whose tutorials are easily available. Such skills can be generated easily.

Initially, a free database server could work, but later paid version will be required which wouldn't be such big of a deal.

From these it is clear that the e-commerceapp for exams is feasible technically.

3.3 Resource Feasibility

The resources required for this mobile application or programming tools programming hardware and programming individuals which are available readily.

Also the database resource which is firebase is also available and well-known for providing highly secure and reliable service.

Hence, the resource feasibility of this project is also met.

3.4 Time Feasibility

This mobile application would take about 3-4 months of time which is sufficient. Hence time feasibility is also met.

3.5 Technology Risks Feasibility

The risks associated here are related to security, but due to the use of firebase secure server, this won't be causing any issue.

Hence risk feasibility is also met.

3.6 Social/Legal Feasibility

This project only uses open source software tools that are secure and free to use hence the customers will only be charged for maintenance and premium services (if they wish to take so) and this will give an opportunity to this app for being used by more institutions/schools especially during the time of online education.

Hence, the JustShop App is feasible in all manners and has a great future.

CHAPTER-4

Project Design

A software architecture includes the elevated level structure of programming framework deliberation, by utilising decay and arrangement, with design style and quality credits. [5] The architecture configuration must adjust to the significant usefulness and execution prerequisites of the framework, just as fulfil the non-functional necessities or requirements. For example, the dependability, adaptability, compactness, and accessibility.

Software architecture must portray its gathering of segments, their associations, cooperations among them and organisation design, all things considered. An application architecture is a guide of how an association's product applications are gathered as a component of its overall endeavour architecture and how those applications interface with one another to meet business or client necessities.

4.1. UML Diagrams for JustShop

UML which stands for Unified Modelling Language, is one of object-oriented solutions used in software modelling and design. It is a pictorial language used to make programming outlines. UML was made by Object Management Group (OMG). [6] The UML 1.0 determination draft was proposed to the OMG in January 1997. It fills in as a norm for programming prerequisite investigation and configuration records which are the reason for building up a product.

UML can be depicted as a universally useful visual displaying language to envision, indicate, build, and report a product framework. Despite the fact that UML is commonly used to display programming framework, it isn't restricted inside this limit. It is likewise used to demonstrate non programming frameworks, for example, measure streams in an assembling unit.

Structural diagrams represents the static parts of a framework. These static parts represents those pieces of a diagram which shapes the principle structure and is subsequently steady. Example of structural diagram is Class Diagram.

Behavioural diagrams fundamentally catch the dynamic part of a framework. Dynamic parts are fundamentally the changing/moving pieces of a framework. Example of behavioural diagram is Sequence Diagram.

UML Diagrams of JustShop help us analyse how the working of each feature will take place so that we can be well planned of the decisions that we will take throughout the development of this application. JustShop offers a range of features that might create a complex structure at later stage and so diagrams will be created.

4.1.1. Use Case Diagram

Use case diagrams model behavior within a system and helps the developers understand of what the user require. The stick man represents what's called an actor. Use case diagram can be useful for getting an overall view of the system and clarifying who can do and more importantly what they can't do. Use case diagram consists of use cases and actors and shows the interaction between the use case and actors. [6]The purpose is to show the interactions between the use case

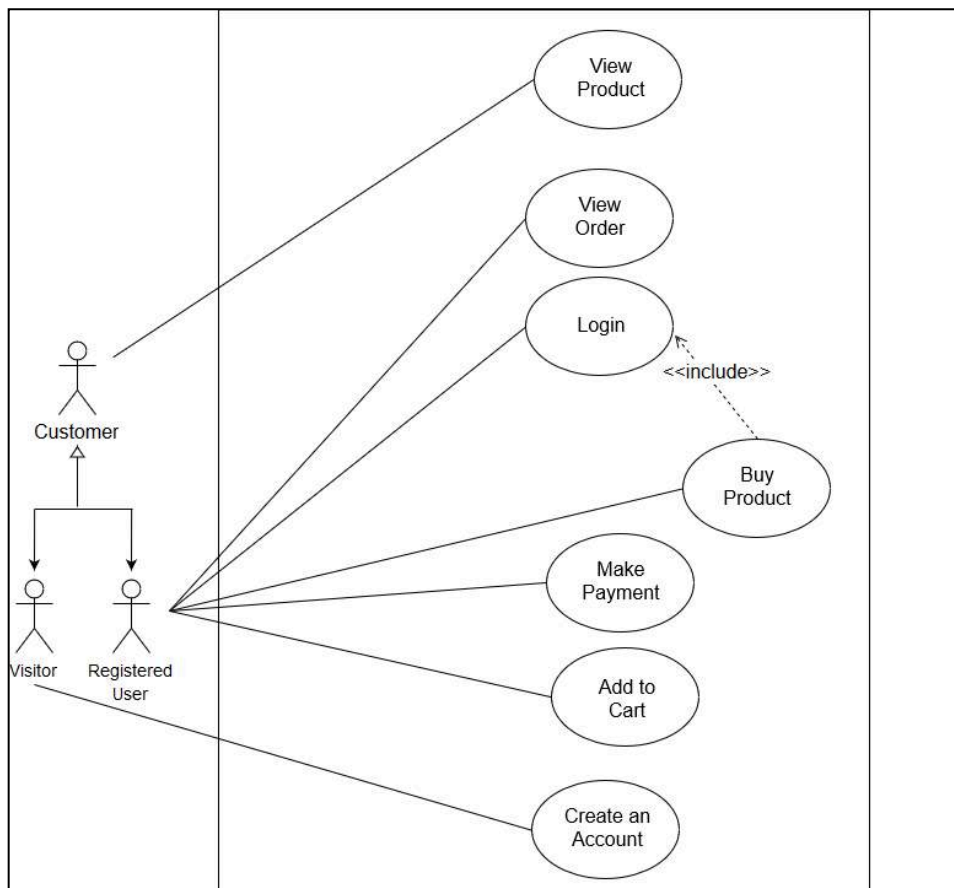


Figure 1 - Use Case Diagram

and actor. To represent the system requirements from user's perspective. An actor could be the end-user of the system or an external system. To show a framework, the main perspective is to catch the dynamic behaviour. Dynamic behaviour implies the behaviour of the framework when it is running/working.

Just static behaviour isn't adequate to demonstrate a framework preferably dynamic behaviour is more significant over static behaviour.

In UML, there are five diagrams accessible to show the dynamic nature and use case diagram is one of them. Presently as we need to talk about that the use case diagram is dynamic in nature, there ought to be some interior or outside elements for making the association.

These interior and outside operators are known as entertainers. Use case diagrams comprises of entertainers, use cases and their connections. The diagram is used to demonstrate the framework/subsystem of an application. A solitary use case diagram catches a specific usefulness of a framework.

Thus to demonstrate the whole framework, various use case diagrams are used.

The main reason behind the use case diagram is to catch the dynamic part of a framework. In any case, this definition is too nonexclusive to even think about describing the reason, as other four diagrams (action, grouping, joint effort, and State-chart) additionally have a similar reason. We will investigate some particular reason, which will recognise it from other four diagrams. Figure 1 shows the use-case diagram for JustShop.

This shows that there is a customer which can be of two types, a registered customer and a non-registered(visiting) customer. Both can view the products on the application. However to make the purchase to buy a product the user has to login to the mobile application.

When the user will login to the application, the data will be fetched and checked for authentication through the Firebase service, as Firebase by Google is being used for storing the database.

A visitor can get to create an account and add products in cart only after creating an account.

4.1.2 Sequence Diagram

Sequence Diagram is the most broadly utilised interaction diagram. A sequence diagram basically portrays interaction between objects in a successive request for example the request where these interactions occur. [6]

We can likewise utilise the terms function diagrams or function situations to allude to a sequence diagram. Sequence diagrams portray how and in what request the articles in a framework work.

These diagrams are generally utilised by financial specialists and programming engineers to archive and comprehend prerequisites for new and existing frameworks.

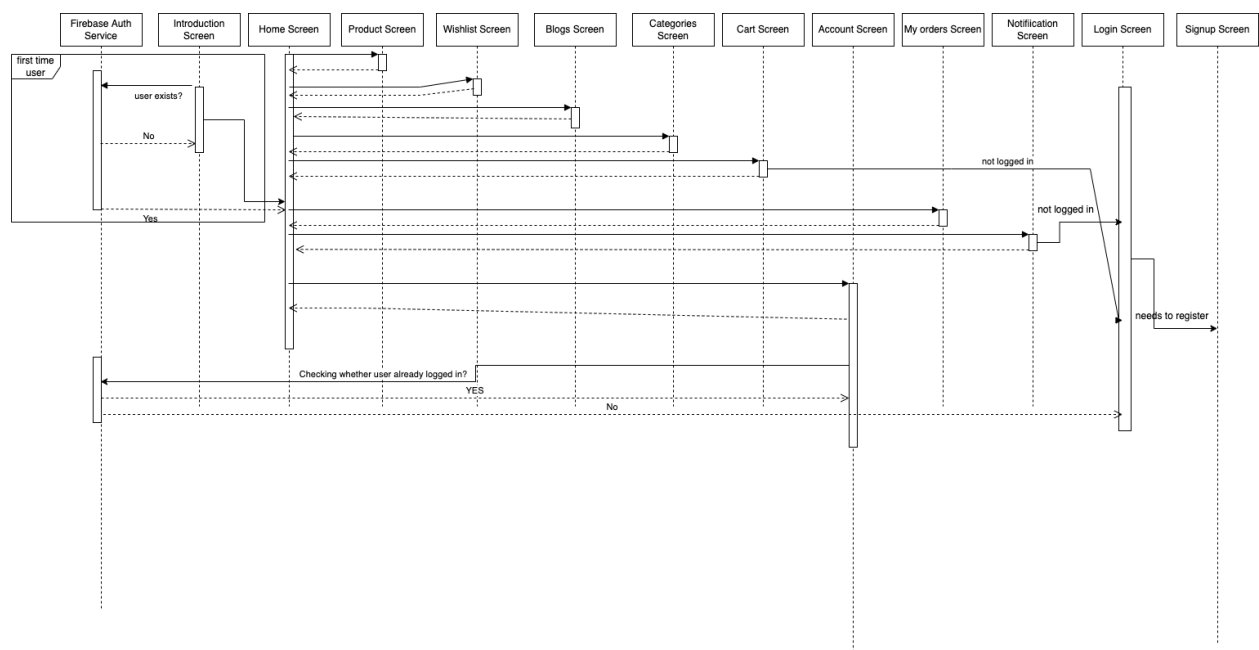


Figure 2 - Sequence Diagram

Here, whenever a user enters the application, they are directed to introduction page, but a check is done whether the user is using for the first time or it has been used earlier. If the app has been used earlier then the user is directed to the home screen else the user is directed to introduction screen. Whenever the user lands on the introduction page they have three carousal screens where the features of the app are explained.

From the home screen the user is given a range options where they can move from the product screen back then back to the home screen then to the Wish List screen and back to the home

screen. Similarly there is blogs screen category screen and more. When user wants to use the cart screen, if the user is already logged in then the cart is showing to them otherwise they are asked to log in and if they click on the login then they move to the login screen. Similarly if user is going to the account screen, then a check is done by the fireplace authentication service that whether or not this user is logged in or not. If the user is logged in then various settings show up on the account screen, else they are redirected to the login screen. At the login screen they can also choose an option to forget password and move to its screen and later they can also sign up to move to the signup screen.

4.1.3. Activity Diagram

Activity diagram is fundamentally a flowchart to speak to the stream starting with one activity then onto the next activity. The activity can be portrayed as an activity of the framework. [6]

The control stream is attracted starting with one activity then onto the next. This stream can be consecutive, extended, or simultaneous. Activity diagrams manage all sort of stream control by utilizing various components, for example, fork, join, and so on

The fundamental reasons for activity diagrams is like other four diagrams. It catches the dynamic conduct of the framework. Other four diagrams are utilized to show the message stream starting with one article then onto the next yet activity diagram is utilized to show message stream starting with one activity then onto the next.

Whenever a user enters the application, they are directed to introduction page, but a check is done whether the user is using for the first time or it has been used earlier. If the app has been used earlier then, the user is directed to the home screen else the user is directed to introduction screen. Whenever the user lands on the introduction page they have three carousal screens where the features of the app are explained.

From the home screen the user is given a range options where they can move from the product screen back then back to the home screen then to the Wish List screen and back to the home screen. Similarly there is blogs screen category screen and more. When user wants to use the cart screen, if the user is already logged in then the cart is showing to them otherwise they are asked to log in and if they click on the login then they move to the login screen. Similarly if user is

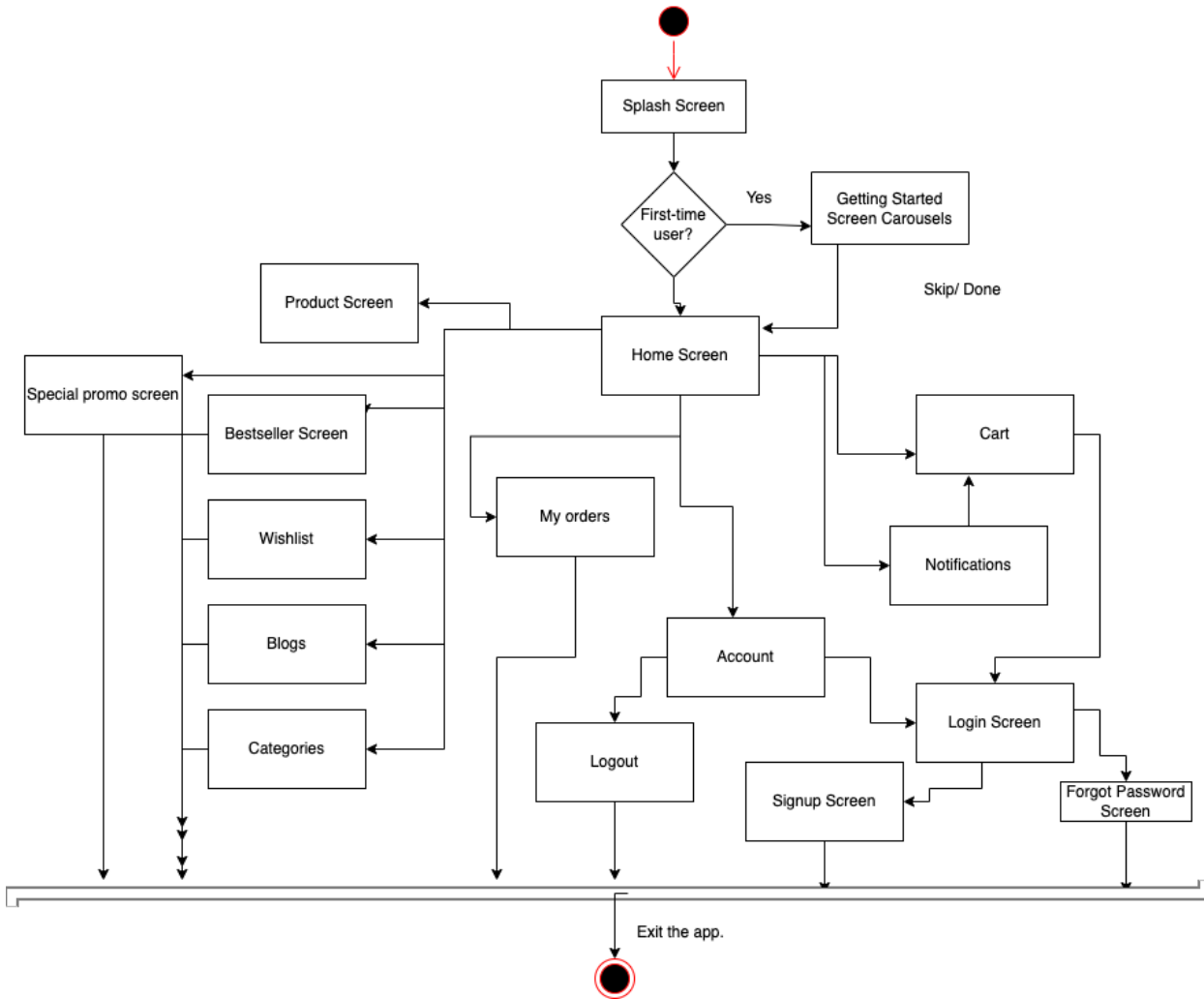


Figure 3 - Activity Diagram

going to the account screen, then a check is done by the fireplace authentication service that whether or not this user is logged in or not. If the user is logged in then various settings show up on the account screen, else they are redirected to the login screen. At the login screen they can also choose an option to forget password and move to its screen and later they can also sign up to move to the signup screen. The last thing is, on any page when user clicks back, or exits, then the activity finishes.

CHAPTER-5

Requirements and Tools of the project

5.1. Requirements

5.1.1 Hardware Requirements for Development :

Once the planning is completed, the process of creating the software starts. For developing the software or application we also require either some kind of a testing tool hardware or actual hardware, or an emulator on the laptop as well as a working system, to develop the app on.

- Operating System must be Microsoft® Windows® 7/8/10 (64-bit) / MacOS High Sierra ++
- The size of ram ought to be a minimum of 4 GB RAM , however 8 GB RAM is also preferred.
- It is essential to have at least 2 GB of disk space , 4 GB Recommended (IDE requires 500 MB + 1.5 GB in Android SDK and emulator system images)
- It is also necessary to possess 1280 x 800 minimum screen resolution.
- A 64-bit environment is required for Android 2.3.x (Gingerbread) and higher versions, including the master branch. You can compile older versions on 32-bit systems.
- A minimum of 250GB of empty disk space is required to check the code along with an additional 150 GB for building it. If you perform a lot of builds, then you will need more space.

5.1.2 Software Requirements for Development :

The softwares required to build this application are :

- Android Studio
- Android SDK
- Android Emulator
- Xcode for iOS
- iOS emulator
- Flutter SDK
- Android version
- Languages - Dart

- Firebase Support - Google Cloud Server[3]
- Flutter Plugins - firebase auth, dcdg:

5.1.3. Hardware Requirements for Use :

Again just like the software, once the development process is completed, application has to be given to normal public users. These users can have any type of hardware. To make the working of the application developed smooth as well as successful it is very critical to possess the right hardware.

So this is the minimum hardware specification to run the application on a device:

- Mobile hardware can be iOS devices (iPhone 4S or a later model) and ARM Android machines.

5.1.4. Software Requirements for Use :

Once the development process is completed, means the code has been written, and has been tested, it has to be given to normal public users. These users can have any type of software. To make the working of the application developed smooth as well as successful it is very critical to possess the right software.

So this is the minimum software specification to run the application on a device:

- Mobile OS supported are : Android Jelly Bean, v16, 4.1.x or later, and iOS 8 or later.

5.2. About the tools used

To make a mobile application that can easily run on both Android and iOS, there are two options easily available, one is Dart language for Flutter Software Development and the other is React Native.

5.2.1 Dart or React

Dart is a client - advanced programming language for applications on various stages. It is created by Google and is utilized to construct versatile, work area, server, and web applications. Dart is an object-oriented, class-based, trash gathered language with C-style punctuation. In Flutter all

applications are made with Dart.. It is by and large used inside Google. It has been exhibited to can make tremendous web applications.[8]

Another technology that works on similar thing is React Native. [9] In React Native after each OS UI update, the application parts will be in a split second redesigned, also. That said – this can break the application's UI, likewise applications written in React Native should have the option to run JavaScript code. Android doesn't have this functionality as a matter of course. Though Flutter is cross stage and can construct applications for a considerable length of time utilizing a solitary codebase. Not each part of the stages is talked about exhaustively. For example, render process, which both React Native and Flutter have put incredible exertion into streamlining, has not been pondered. Besides, correlations among native and cross-platform applications are not referenced thoroughly. React Native Can be introduced utilizing Node Package Manager (NPM) while utilizing macOS you really want to have the homebrew bundle chief also. So for engineers having NPM introduced the cycle is truly simple while other developers would have to gain proficiency with the intricate details of NPM. [10] While, Flutter Supports twofold download directly from GitHub for a particular stage. Likewise, on account of macOS in the wake of downloading the record should be added as a PATH variable. React Native Relies intensely on outsider libraries to get to a large portion of the native modules.[10] Flutter has Rich arrangement of parts involving every native module required which avoids the need of a third-party library. Likewise, Flutter is loaded with WIDGETS. [8]. All in all, flutter has a greater advantage over React Native and so this can be easily used for the development of the E-commerce mobile application such that it can be used on both Android and iOS seamlessly.

5.2.1 Flutter Framework

An application architecture guarantees that applications are versatile and solid, and helps undertakings distinguish holes in usefulness.

An application architecture chart involves a high level review of the parts and major communications inside the framework, for example micro-services, information bases, and so forth. The application architecture graph essentially addresses the "What" according to the framework.

All in all, application architecture characterises how applications communicate with middleware, information bases and different applications. Application architectures typically follow programming plan rules that are commonly acknowledged among its disciples however may need formal industry guidelines.

Flutter is planned as an extensible, layered

framework. It exists as a progression of free libraries that each rely upon the hidden layer. No layer has restricted admittance to the layer beneath, and all aspects of the system level is intended to be discretionary and replaceable. Below is the pictorial representation of the same in Figure 1.[7]

5.2.2. Firebase

Firebase is a platform developed by Google for creating mobile and web applications. It was originally an independent company founded in 2011. In 2014, Google acquired the platform and it is now their flagship offering for app development. Firebase is open-source for low-level softwares that would help them create database on the cloud without having to pay them for anything until the app becomes large-scaled.

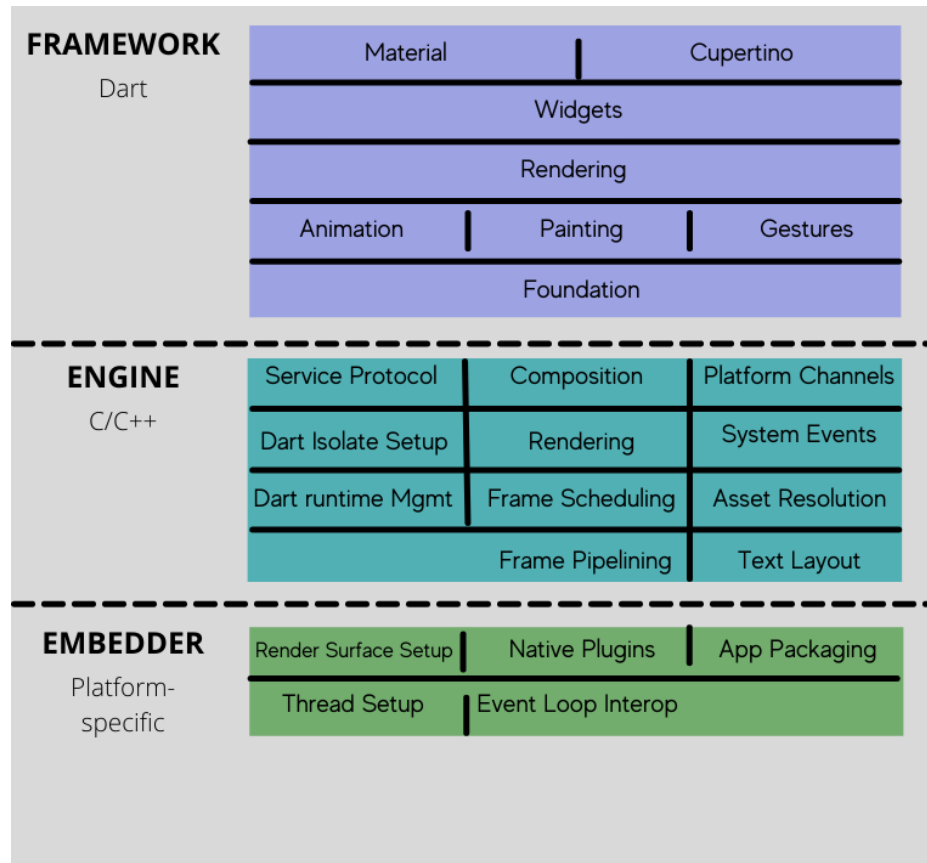


Figure 4 - Architectural Layers of Flutter

CHAPTER-4

Functionality & Implementation

The functionality of the JustShop app is to provide the basic need of shoppers to provide a range of products and the basic need of sellers to provide good customers. Along these lines JustShop also engages influencers in this site by adding a blogs page to the site in order to review trendy products and watch/read reviews of the same.

Following is the implementation of the application with respect to various modules and screens with the specified name of the files and their sample code :

1. **Splash Screen Module** : This is responsible to show the company logo at the starting of the app, to show to user in the time that app starts and configures itself for a successful start.

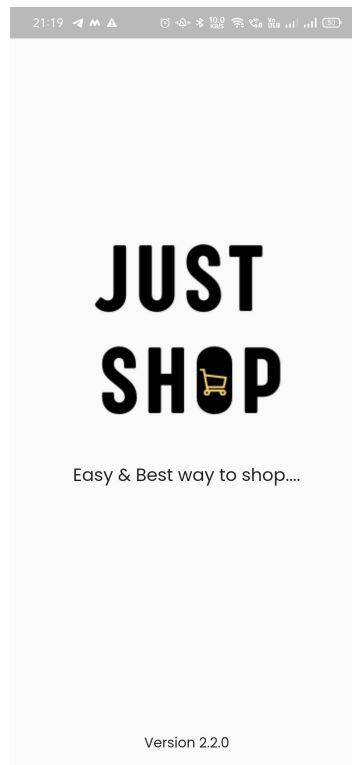


Figure 5 - Splash Screen

Code :

```
Splashscreen.dart
```

```
import 'dart:async';
```

```

import 'package:cached_network_image/cached_network_image.dart';
import 'package:flutter/material.dart';
import 'package:nyoba/constant/constants.dart';
import 'package:nyoba/pages/intro/IntroScreen.dart';
import 'package:nyoba/pages/home/HomeScreen.dart';
import 'package:nyoba/provider/HomeProvider.dart';
import 'package:nyoba/services/Session.dart';
import 'package:nyoba/utills/utility.dart';
import 'package:provider/provider.dart';

```

```

class SplashScreen extends StatefulWidget {
  final Future Function() onLinkClicked;
  SplashScreen({Key key, this.onLinkClicked}) : super(key: key);

```

```

  @override
  _SplashScreenState createState() => _SplashScreenState();
}

```

```

class _SplashScreenState extends State<SplashScreen> {
  startSplashScreen() async {
    final home = Provider.of<HomeProvider>(context, listen: false);
    var duration = const Duration(milliseconds: 2500);

    return Timer(duration, () {
      Navigator.of(context).pushReplacement(MaterialPageRoute(builder: () {
        return Session.data.getBool('isIntro')
          ? HomeScreen()
          : IntroScreen(
              intro: home.intro,

```

```
        );  
    });  
    if (widget.onLinkClicked != null) {  
        print("URL Available");  
        widget.onLinkClicked();  
    }  
});  
}
```

```
@override  
void initState() {  
    super.initState();  
    if (!Session.data.containsKey('isIntro')) {  
        Session.data.setBool('isLogin', false);  
        Session.data.setBool('isIntro', false);  
    }  
    printLog(widget.onLinkClicked.toString());  
    loadHome();  
}
```

```
@override  
void setState(fn) {  
    if (mounted) {  
        super.setState(fn);  
    }  
}
```

```
@override  
void dispose() {
```

```

    super.dispose();
  }

  loadHome() async {
    await Provider.of<HomeProvider>(context, listen: false)
      .fetchHome()
      .then((value) async {
        this.setState(() {});
        await startSplashScreen();
      });
  }

  @override
  Widget build(BuildContext context) {
    final home = Provider.of<HomeProvider>(context, listen: false);

    return Scaffold(
      body: home.loading
        ? Container()
        : Container(
            width: MediaQuery.of(context).size.width,
            height: MediaQuery.of(context).size.height,
            child: Column(
              children: [
                Expanded(
                  child: Column(
                    crossAxisAlignment: CrossAxisAlignment.center,
                    mainAxisAlignment: MainAxisAlignment.center,
                    children: [

```

```

Container(
  margin: EdgeInsets.symmetric(horizontal: 25),
  child: CachedNetworkImage(
    imageUrl: home.splashscreen.image,
    placeholder: (context, url) => Container(),
    errorWidget: (context, url, error) =>
      Icon(Icons.error),
  ),
),
Text(
  home.splashscreen.title,
  style: TextStyle(fontSize: 22),
),
Text(
  home.splashscreen.description,
  style: TextStyle(fontSize: 18),
),
],
),
),
Container(
  margin: EdgeInsets.symmetric(vertical: 20),
  alignment: Alignment.bottomCenter,
  child: Text(
    "Version $version",
    style: TextStyle(fontSize: 14),
  ),
),
],

```



```

    ));
  }
}

```

2. Login Screen Module : This module is responsible for signing in user to the application platform.

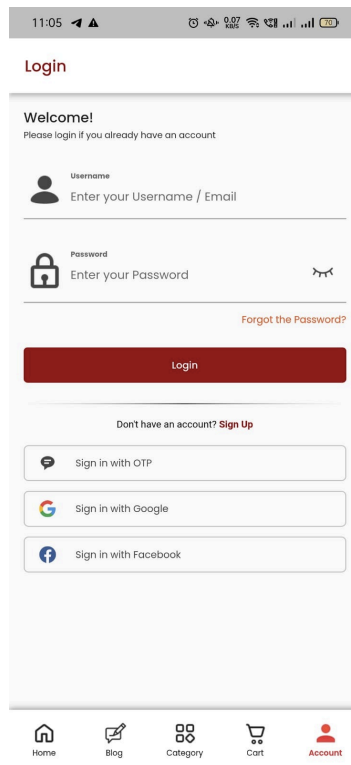


Figure 6 - Login Screen

Code :

```

import 'dart:io';

import 'package:flutter/material.dart';
import 'package:flutter_screenutil/flutter_screenutil.dart';
import 'package:flutter/gestures.dart';
import 'package:hexcolor/hexcolor.dart';
import 'package:nyoba/pages/auth/SignInOTPScreen.dart';

```

```

import 'package:nyoba/pages/home/HomeScreen.dart';
import 'package:nyoba/provider/LoginProvider.dart';
import 'package:nyoba/services/Session.dart';
import 'package:provider/provider.dart';
import '../AppLocalizations.dart';
import 'ForgotPasswordScreen.dart';
import 'SignUpScreen.dart';
import '../utils/utility.dart';
import 'package:auto_size_text/auto_size_text.dart';

class Login extends StatefulWidget {
  final bool isFromNavBar;

  Login({Key key, this.isFromNavBar}) : super(key: key);

  @override
  _LoginState createState() => _LoginState();
}

class _LoginState extends State<Login> {
  bool isVisible = false;

  TextEditingController username = new TextEditingController();
  TextEditingController password = new TextEditingController();

  bool isFromNavBar = true;

  @override
  void initState() {
    super.initState();
  }
}

```

```

if (widget.isFromNavBar != null) {
    isFromNavBar = widget.isFromNavBar;
}
}

```

```

loginGoogle() async {
    loadingPop(context);
    await Provider.of<LoginProvider>(context, listen: false)
        .signInWithGoogle(context)
        .then((value) {
            if (Session.data.getString('cookie') != null) {
                Navigator.pushAndRemoveUntil(
                    context,
                    MaterialPageRoute(builder: (BuildContext context) => HomeScreen()),
                    (Route<dynamic> route) => false);
            } else {
                snackBar(context,
                    message:
                        "Invalid error when trying sign in using google, please contact admin or developer",
                    color: Colors.red);
                Navigator.pop(context);
            }
        });
}

```

```

loginFacebook() async {
    loadingPop(context);
    await Provider.of<LoginProvider>(context, listen: false)
        .signInWithFacebook(context)

```

```

        .then((value) {
    if (Session.data.getString('cookie') != null) {
        Navigator.pushAndRemoveUntil(
            context,
            MaterialPageRoute(builder: (BuildContext context) => HomeScreen()),
            (Route<dynamic> route) => false);
    } else {
        snackBar(context,
            message:
                "Invalid error when trying sign in using facebook, please contact admin or developer",
            color: Colors.red);
        Navigator.pop(context);
    }
    });
}

```

@override

```

Widget build(BuildContext context) {
    final auth = Provider.of<LoginProvider>(context, listen: false);

    var loginByDefault = () async {
        if (username.text.isNotEmpty && password.text.isNotEmpty) {
            FocusScopeNode currentFocus = FocusScope.of(context);

            if (!currentFocus.hasPrimaryFocus) {
                currentFocus.unfocus();
            }
            await Provider.of<LoginProvider>(context, listen: false)
                .login(context, password: password.text, username: username.text)

```

```

        .then((value) => this.setState({}));
    } else {
        snackBar(context, message: 'Username & password should not empty');
    }
};

```

```

return Scaffold(
  resizeToAvoidBottomInset: false,
  appBar: AppBar(
    leading: !isFromNavBar
      ? IconButton(
        onPressed: () => Navigator.pop(context),
        icon: Icon(
          Icons.arrow_back,
          color: Colors.black,
        ))
      : null,
    title: AutoSizeText(
      AppLocalizations.of(context).translate('login'),
      style:
        TextStyle(fontSize: responsiveFont(16), color: secondaryColor),
    ),
    backgroundColor: Colors.white),
  body: Container(
    margin: EdgeInsets.all(15),
    child: SingleChildScrollView(
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [

```

```

Text(
  "${AppLocalizations.of(context).translate('welcome')}!",
  style: TextStyle(
    fontWeight: FontWeight.w500,
    fontSize: responsiveFont(14),
  ),
),
Text(
  AppLocalizations.of(context).translate('subtitle_login'),
  style: TextStyle(
    fontSize: responsiveFont(9),
  ),
),
Container(
  height: 20,
),
Container(
  height: MediaQuery.of(context).size.height / 12,
  child: TextField(
    controller: username,
    decoration: InputDecoration(
      prefixIcon: Container(
        width: 24.w,
        height: 24.h,
        padding: EdgeInsets.only(right: 5),
        child: Image.asset("images/account/akun.png")),
      labelStyle: TextStyle(
        fontWeight: FontWeight.w700,
        fontSize: responsiveFont(10),

```

```

    ),
    hintStyle: TextStyle(
      fontWeight: FontWeight.w400,
      fontSize: responsiveFont(12),
    ),
    floatingLabelBehavior: FloatingLabelBehavior.always,
    labelText: "Username",
    hintText: AppLocalizations.of(context)
      .translate('hint_username')),
  ),
),
Container(
  height: 15,
),
Container(
  height: MediaQuery.of(context).size.height / 12,
  child: TextField(
    controller: password,
    obscureText: isVisible ? false : true,
    decoration: InputDecoration(
      suffixIcon: GestureDetector(
        onTap: () {
          setState() {
            isVisible = !isVisible;
          });
        },
      ),
    child: Container(
      padding: EdgeInsets.only(top: 20, bottom: 20),
      child: Image.asset(isVisible

```

```

        ? "images/account/melek.png"
        : "images/account/merem.png")),
    ),
    prefixIcon: Container(
      width: 24.w,
      height: 24.h,
      padding: EdgeInsets.only(right: 5),
      child: Image.asset("images/account/lock.png")),
    labelStyle: TextStyle(
      fontWeight: FontWeight.w700,
      fontSize: responsiveFont(10),
    ),
    hintStyle: TextStyle(
      fontWeight: FontWeight.w400,
      fontSize: responsiveFont(12),
    ),
    floatingLabelBehavior: FloatingLabelBehavior.always,
    labelText: "Password",
    hintText: AppLocalizations.of(context)
      .translate('hint_password')),
  ),
),
Container(
  height: 10,
),
InkWell(
  onTap: () => Navigator.push(
    context,
    MaterialPageRoute(

```



```

        builder: (context) => ForgotPasswordScreen()))
    .then((value) => this.setState() {})),
child: Container(
  alignment: Alignment.centerRight,
  width: double.infinity,
  child: Text(
    AppLocalizations.of(context).translate('forgot_password'),
    style: TextStyle(
      color: HexColor("FD490C"),
      fontSize: responsiveFont(10),
    ),
  ),
),
),
),
),
Container(
  height: 15,
),
Container(
  width: double.infinity,
  child: TextButton(
    style: TextButton.styleFrom(
      padding: EdgeInsets.symmetric(vertical: 10),
      backgroundColor:
        auth.loading ? Colors.grey : secondaryColor),
    onPressed: loginByDefault,
    child: auth.loading
      ? customLoading()
      : Text(
        AppLocalizations.of(context).translate('login'),

```

```

        style: TextStyle(
          color: Colors.white,
          fontSize: responsiveFont(10),
        ),
      ),
    ),
  ),
  Container(
    height: 15,
  ),
  Image.asset("images/account/baris.png"),
  Container(
    height: 15,
  ),
  Center(
    child: RichText(
      text: TextSpan(
        style: TextStyle(
          color: Colors.black,
          fontSize: responsiveFont(10),
        ),
        children: <TextSpan>[
          TextSpan(
            text:
              "${AppLocalizations.of(context).translate("don't_have_account")}" ,
          ),
          TextSpan(
            recognizer: new TapGestureRecognizer()
              ..onTap = () => Navigator.push(

```

```

        context,
        MaterialPageRoute(
            builder: (context) => SignUp()),
        text:
            AppLocalizations.of(context).translate('sign_up'),
        style: TextStyle(
            fontWeight: FontWeight.bold,
            color: secondaryColor)),
    ],
),
),
),
Container(
    height: 15,
),
signInButton(
    "${AppLocalizations.of(context).translate('sign_in')} OTP",
    "message"),
Container(
    height: 10,
),
signInButton(
    "${AppLocalizations.of(context).translate('sign_in')} Google",
    "google"),
Container(
    height: 10,
),
signInButton(
    "${AppLocalizations.of(context).translate('sign_in')} Facebook",

```

```

        "facebook"),
    Container(
      height: 10,
    ),
    if (Platform.isIOS)
      signInButton(
        "${AppLocalizations.of(context).translate('sign_in')} Apple",
        "apple"),
  ],
),
),
);
}

```

```

Widget signInButton(String title, String image) {
  return InkWell(
    onTap: () {
      if (image == 'message') {
        Navigator.push(context,
          MaterialPageRoute(builder: (context) => SignInOTPScreen()))
          .then((value) => this.setState(() {}));
      } else if (image == 'google') {
        loginGoogle();
      } else if (image == 'facebook') {
        loginFacebook();
      }
    },
    child: Container(

```

```

padding: EdgeInsets.symmetric(horizontal: 15, vertical: 7),
decoration: BoxDecoration(
  borderRadius: BorderRadius.circular(5),
  border: Border.all(color: HexColor("c4c4c4")),
width: double.infinity,
child: Row(
  children: [
    Container(
      width: 17.w,
      height: 17.h,
      child: Image.asset("images/account/$image.png")),
    SizedBox(
      width: 20,
    ),
    Text(
      title,
      style: TextStyle(
        fontSize: responsiveFont(10), color: HexColor("464646")),
    )
  ],
)),
);
}
}

```

3. **Signup Screen Module** : This module is responsible to sign up any user to the application .

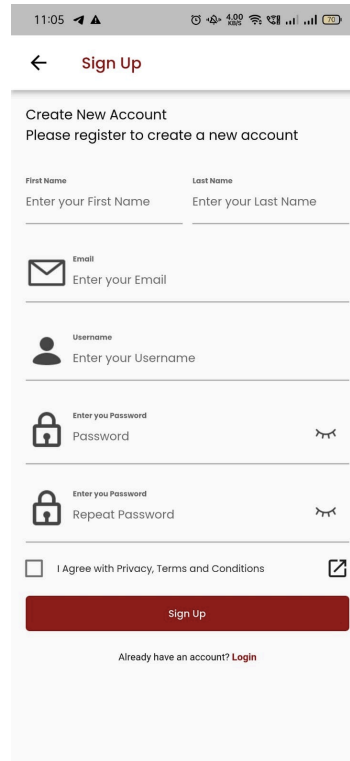


Figure 7 - Signup Screen

Code :

```
import 'package:flutter/material.dart';
import 'package:hexcolor/hexcolor.dart';
import 'package:flutter_screenutil/flutter_screenutil.dart';
import 'package:email_validator/email_validator.dart';
import 'package:nyoba/provider/HomeProvider.dart';
import 'package:nyoba/provider/RegisterProvider.dart';
import 'package:nyoba/widgets/webview/WebView.dart';
import 'package:provider/provider.dart';
import 'package:flutter/gestures.dart';
import '../AppLocalizations.dart';
import '../utils/utility.dart';
```

```

class SignUp extends StatefulWidget {
  SignUp({Key key}) : super(key: key);

  @override
  _SignUpState createState() => _SignUpState();
}

class _SignUpState extends State<SignUp> {
  bool isVisible = false;
  bool checkedValue = false;

  bool isValidEmail = false;

  TextEditingController controllerFirstname = new TextEditingController();
  TextEditingController controllerLastname = new TextEditingController();
  TextEditingController controllerUsername = new TextEditingController();
  TextEditingController controllerEmail = new TextEditingController();
  TextEditingController controllerPassword = new TextEditingController();
  TextEditingController controllerPasswordConfirm = new TextEditingController();

  @override
  void initState() {
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    final register = Provider.of<RegisterProvider>(context, listen: false);
    final generalSettings =

```

```
Provider.of<HomeProvider>(context, listen: false);
```

```
var signUp = () async {  
  if (controllerFirstname.text.isNotEmpty &&  
      controllerLastname.text.isNotEmpty &&  
      controllerEmail.text.isNotEmpty &&  
      controllerUsername.text.isNotEmpty &&  
      controllerPassword.text.isNotEmpty &&  
      controllerPasswordConfirm.text.isNotEmpty) {  
    FocusScopeNode currentFocus = FocusScope.of(context);  
  
    if (!currentFocus.hasPrimaryFocus) {  
      currentFocus.unfocus();  
    }  
  
    isValidEmail = EmailValidator.validate(controllerEmail.text);  
  
    if (!isValidEmail) {  
      return snackBar(context, message: 'Your email is not valid.');    }  
  
    if (controllerPassword.text != controllerPasswordConfirm.text) {  
      snackBar(context,  
        message: 'Your password and confirmation password do not match.');    } else {  
      if (controllerPassword.text.length < 8) {  
        return snackBar(context,  
          message: 'Your password cannot less than 8 character.');      }  
    }  
  }  
}
```



```

if (checkedValue) {
  setState(() {
    register.loading = true;
  });
  final Future<Map<String, dynamic>> authResponse = register.signUp(
    username: controllerUsername.text,
    password: controllerPassword.text,
    email: controllerEmail.text,
    firstname: controllerFirstname.text,
    lastname: controllerLastname.text);

  authResponse.then((value) {
    if (value['cookie'] != null) {
      Navigator.pop(context);
      snackBar(context,
        message: "Success create new account", color: Colors.green);
    } else {
      snackBar(context, message: value['message'], color: Colors.red);
    }
    setState(() {
      register.loading = false;
    });
  });
} else {
  snackBar(context,
    message:
      'You must agree to our privacy policy and terms of use');
}
}

```

```

    } else {
      snackBar(context, message: 'Form field should not be empty');
    }
  };

return Scaffold(
  appBar: AppBar(
    leading: IconButton(
      onPressed: () {
        Navigator.pop(context);
      },
      icon: Icon(
        Icons.arrow_back,
        color: Colors.black,
      ),
    ),
    backgroundColor: Colors.white,
    title: Text(
      AppLocalizations.of(context).translate('sign_up'),
      style:
        TextStyle(fontSize: responsiveFont(16), color: secondaryColor),
    ),
  ),
  body: Container(
    margin: EdgeInsets.all(15),
    child: SingleChildScrollView(
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [

```

```

Text(
  AppLocalizations.of(context).translate('create_account'),
  style: TextStyle(
    fontSize: responsiveFont(14), color: Colors.black),
),
Text(
  AppLocalizations.of(context)
    .translate('subtitle_create_account'),
  style: TextStyle(color: Colors.black),
),
Container(
  height: 20,
),
Row(
  children: [
    Expanded(
      child: form(
        AppLocalizations.of(context)
          .translate('enter_firstname'),
        AppLocalizations.of(context)
          .translate('first_name'),
        false,
        controllerFirstname)),
    SizedBox(
      width: 10,
    ),
    Expanded(
      child: form(
        AppLocalizations.of(context)

```

```

        .translate('enter_lastname'),
        AppLocalizations.of(context).translate('last_name'),
        false,
        controllerLastname))
    ],
  ),
  Container(
    height: 15,
  ),
  form(AppLocalizations.of(context).translate('enter_email'),
    "Email", true, controllerEmail,
    icon: "email"),
  Container(
    height: 15,
  ),
  form(AppLocalizations.of(context).translate('enter_username'),
    "Username", true, controllerUsername,
    icon: "akun"),
  Container(
    height: 15,
  ),
  passwordForm(
    AppLocalizations.of(context).translate('enter_password'),
    "Password",
    controllerPassword),
  Container(
    height: 15,
  ),
  passwordForm(

```

```

AppLocalizations.of(context).translate('enter_password'),
AppLocalizations.of(context).translate('repeat_password'),
controllerPasswordConfirm),
Container(
  height: 15,
),
Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: [
    Row(
      children: [
        Container(
          width: 17.w,
          height: 17.h,
          child: Checkbox(
            checkColor: Colors.white,
            activeColor: HexColor("ED625E"),
            splashRadius: 0,
            value: checkedValue,
            onChanged: (value) {
              setState(() {
                checkedValue = value;
              });
            },
          ),
          SizedBox(
            width: 15,
          ),
          Text(

```

```

AppLocalizations.of(context)
  .translate('term_condition_sign_up'),
  style: TextStyle(fontSize: responsiveFont(10)),
),
],
),
InkWell(
  onTap: () {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => WebViewScreen(
          url: generalSettings.privacy.description,
          title: generalSettings.privacy.slug
            .toUpperCase(),
        )),
      ),
    child: Icon(Icons.open_in_new),
  )
],
),
Container(
  height: 10,
),
/*Container(
  width: double.infinity,
  height: MediaQuery.of(context).size.height / 6,
  child: RecaptchaV2(
    apiKey: '6Lei7XsaAAAAAMv_EkpWMtGbuDXrdWp0N1x9vtCZ',

```

```

apiSecret: '6Lei7XsaAAAAAMGbqOvMQb8Y4kS-eqTS8xFPTV2P',
controller: recaptchaV2Controller,
onVerifiedError: (err) {
  print(err);
},
onVerifiedSuccessfully: (success) {
  setState() {
    if (success) {
      // You've been verified successfully.
    } else {
      // "Failed to verify.
    }
  });
},
),
),*/
Container(
width: double.infinity,
child: TextButton(
style: TextButton.styleFrom(
padding: EdgeInsets.symmetric(vertical: 10),
backgroundColor:
register.loading ? Colors.grey : secondaryColor),
onPressed: register.loading ? null : signUp,
child: register.loading
? customLoading()
: Text(
AppLocalizations.of(context).translate('sign_up'),
style: TextStyle(

```

```

        color: Colors.white,
        fontSize: responsiveFont(10),
      ),
    ),
  ),
  Container(
    height: 15,
  ),
  Center(
    child: RichText(
      text: TextSpan(
        style: TextStyle(
          color: Colors.black,
          fontSize: responsiveFont(10),
        ),
        children: <TextSpan>[
          TextSpan(
            text:
              "${AppLocalizations.of(context).translate('have_account')} ",
          ),
          TextSpan(
            recognizer: new TapGestureRecognizer()
              ..onTap = () => Navigator.pop(context),
            text:
              AppLocalizations.of(context).translate('login'),
            style: TextStyle(
              fontWeight: FontWeight.bold,
              color: secondaryColor)),

```



```

        ],
      ),
    ),
  ],
),
));
}

```

```

Widget passwordForm(
  String hints, String label, TextEditingController controller) {
  return Container(
    height: MediaQuery.of(context).size.height / 12,
    child: TextField(
      controller: controller,
      obscureText: isVisible ? false : true,
      decoration: InputDecoration(
        suffixIcon: GestureDetector(
          onTap: () {
            setState(() {
              isVisible = !isVisible;
            });
          },
        ),
        child: Container(
          padding: EdgeInsets.only(top: 20, bottom: 20),
          child: Image.asset(isVisible
            ? "images/account/melek.png"
            : "images/account/merem.png")),

```

```

    ),
    prefixIcon: Container(
      alignment: Alignment.topCenter,
      width: 24.w,
      height: 24.h,
      padding: EdgeInsets.only(right: 5),
      child: Image.asset("images/account/lock.png")),
    labelStyle: TextStyle(
      fontWeight: FontWeight.w700,
      fontSize: responsiveFont(10),
    ),
    hintStyle: TextStyle(
      fontWeight: FontWeight.w400,
      fontSize: responsiveFont(12),
    ),
    floatingLabelBehavior: FloatingLabelBehavior.always,
    labelText: hints,
    hintText: label),
  ),
);
}

```

Widget form(

String hints, String label, bool prefix, TextEditingController controller,

{String icon = "email"}) {

return Container(

height: MediaQuery.of(context).size.height / 12,

child: TextField(

controller: controller,

```

keyboardType:
  icon == "email" ? TextInputType.emailAddress : TextInputType.text,
decoration: InputDecoration(
  prefixIcon: prefix
    ? Container(
      padding: EdgeInsets.only(right: 5),
      width: 24.w,
      height: 24.h,
      child: Image.asset("images/account/$icon.png"))
    : null,
  labelStyle: TextStyle(
    fontWeight: FontWeight.w700,
    fontSize: responsiveFont(10),
  ),
  hintStyle: TextStyle(
    fontWeight: FontWeight.w400,
    fontSize: responsiveFont(12),
  ),
  floatingLabelBehavior: FloatingLabelBehavior.always,
  labelText: label,
  hintText: hints),
),
);
}
}

```

Other modules are as follows :

Intro Module : Gives an introduction of the app and its features when any user opens the app for the very first time.

Cart Module : Responsible to show the products in the cart as a scroll view widget in the vertical form.

Home Page Module : Responsible to give the user a home page for the app, where they can then navigate to various other features.

Product Module : This shows the screen where products can be seen by user and their details can be displayed.

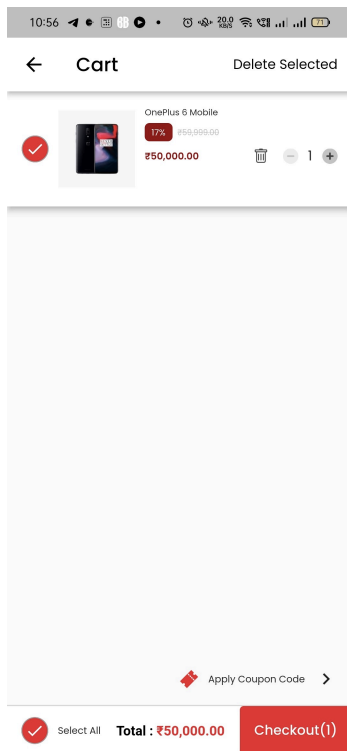


Figure 8 - Cart Screen



Figure 9 - Product Screen

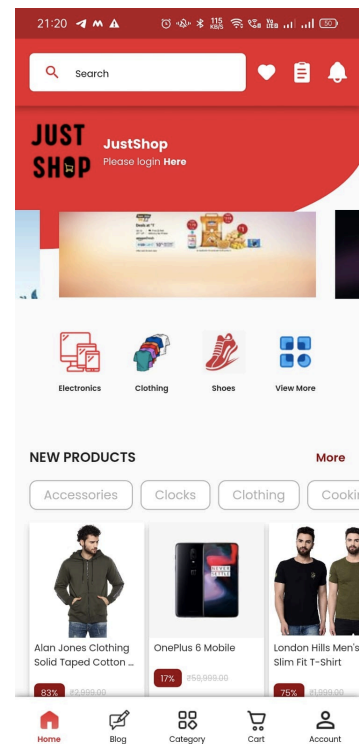


Figure 10 - Home Screen

Wishlist Screen module : Responsible to show the Wishlist of the user, in which selected products are successfully shown.

Blogs Screen module : Responsible for giving out the opportunity to users to write and promote products through blogs.

Figure 11 - Intro Screens

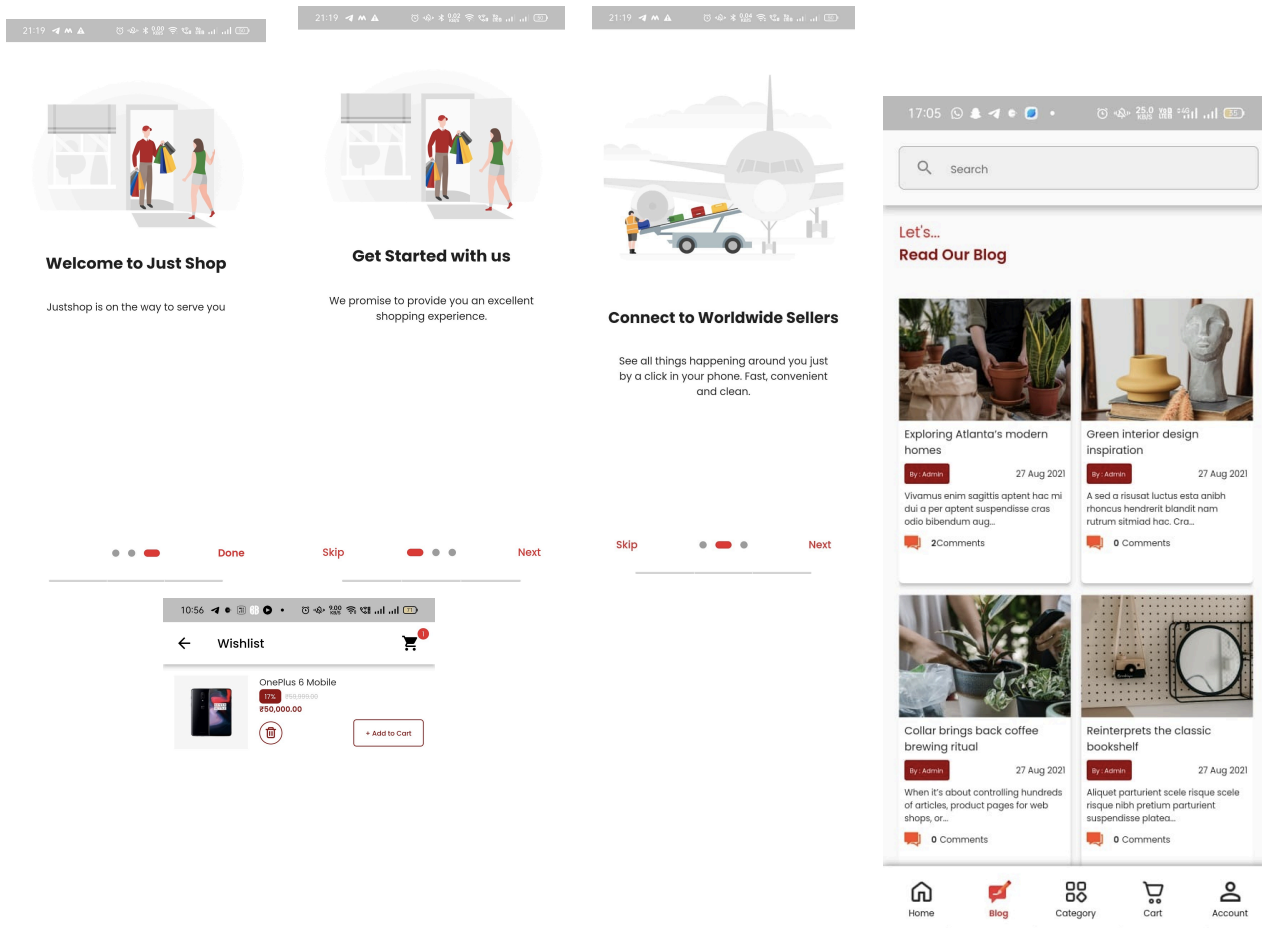


Figure 13 - Blog Screen

Figure 12 - Wishlist Screen

CHAPTER-5

CONCLUSION & FUTURE SCOPE

Conclusion

The results show that using Flutter and firebase, the software application for cross-platform usage has proved to be beneficially during the development phase as well as usage, since it is operable on android as well as iOS. JustShop is the only mobile application offering a range of products and also giving a chance to celebrities and influencers to earn through referring products and get benefit. Blogs allow users to check the products or items that are trending in the market and get a real review in detailed form on the same space where they can buy the product too.

Future Scope

JustShop app offers great variety of products, and has some future scope of scalability and expansion which can be easily handled using firebase, since it allows various features to implement the same. In future, the payment transaction can be made easier and more convenient as well as there can be some security mechanisms. The app can also offer some on-call assistance on the page itself in order to seek FAQs regarding the products.

REFERENCES

1. European E-commerce Report, at https://www.ecommerce-europe.eu/wp-content/uploads/2019/07/European_Ecommerce_report_2019_freeFinal-version.pdf
2. Economic and Social Impact of E-Commerce, at <https://www.oecd-ilibrary.org/docserver/236588526334.pdf?expires=1638511979&id=id&accname=guest&checksum=0A84CF06FA833610454CBE2949C2EC0E>
3. Zhihan Lv . *Optimization of E-Commerce Supply Chain Management Process Based on Internet of Things Technology* , Hindawi, Open article.
4. Anne Engström Esmail Salehi-Sangari, Assessment of Business-to-Business (B2B) e-Marketplaces' Performance, Doctoral Thesis.
5. Software Architecture, Tutorials Point, at https://www.tutorialspoint.com/software_architecture_design/introduction.htm
6. Unified Modelling Language, Geeks for Geeks, at <https://www.geeksforgeeks.org/unified-modeling-language-uml-introduction/>
7. Flutter Development Document, at <https://flutter.dev>
8. Dart Development, Available at : <https://dart.dev/>
9. React Native Development, Available at : <https://reactnative.dev/>
10. Yeshwin Anil, Archana Sahoo, Powel Shoby, *Pet-friendly application using Flutter*.
International Journal for Research in Applied Science & Engineering Technology
(IJRASET) ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429 Volume 8 Issue XII
Dec 2020

PUBLICATION

Status : Communicated

[TiS] Submission Acknowledgement Inbox x



Editor of Trends in Sciences <journal.wu@gmail.com>
to me ▾

Thu, 2 Dec, 16:07 (19 hours ago) ☆ ↶ ⋮

Dear Dr. Bhavya Dubey:

Thank you for submitting the manuscript, "E- Commerce Cross Platform Mobile Application Development using Flutter" to Trends in Sciences. With the online journal management system that we are using, you will be able to track its progress through the editorial process by logging in to the journal web site:

Submission URL: <https://tis.wu.ac.th/index.php/tis/authorDashboard/submission/1009>
Username: bhavyadubey

If you have any questions, please contact me. Thank you for considering this journal as a venue for your work and hope you will enjoy publishing in this journal, and look forward to receiving high-quality articles from you in future.

- PS. - The submission has not been previously published, nor is it before another journal for consideration.**
- This article has been reviewed by native English speakers.
 - The work are discussed in reference to recent journal publications. Mostly citations should not be older than 5 years.
 - Publication Ethics, Copyright Transfer, Authorship, Highlights, Graphical Abstract, Journal's Template, and References format have been appropriately considered.

Editor of Trends in Sciences

[Trends in Sciences](#)

Formerly know as [Walailak Journal of Science and Technology](#)

2020 SJR (SCOPUS): 0.146