

ShopSmart



Group Members

Maira Babar (01-131172-011)

Maryam Akram (01-131172-013)

Supervisor: Engr. Aleem Ahmed

A Final Year Project submitted to the Department of Software Engineering
Faculty of Engineering Sciences, Bahria University, Islamabad in the partial
fulfillment for the award of degree in Bachelor of Software Engineering

July 2021

THESIS COMPLETION CERTIFICATE

Student Name: Maira Babar Enrolment No: 01-131172-011

Student Name: Maryam Akram Enrolment No: 01-131172-013

Programme of Study: Bachelor of Software Engineering

Project Title: ShopSmart

It is to certify that the above students' project has been completed to my satisfaction and to my belief, its standard is appropriate for submission for evaluation. I have also conducted plagiarism test of this thesis using HEC prescribed software and found similarity index at 16% that is within the permissible limit set by the HEC. I have also found the thesis in a format recognized by the department.

Supervisor's Signature: _____

Date: 9th June, 2021 Name: Engr Aleem Ahmed

CERTIFICATE OF ORIGINALITY

This is certified that the intellectual contents of the project **ShopSmart** are the product of my/our own work except, as cited properly and accurately in the acknowledgements and references, the material taken from such sources as research journals, books, internet, etc. solely to support, elaborate, compare, extend and/or implement the earlier work. Further, this work has not been submitted by me/us previously for any degree, nor it shall be submitted by me/us in the future for obtaining any degree from this University, or any other university or institution. The incorrectness of this information, if proved at any stage, shall authorities the University to cancel my/our degree.

Name of the Student: Maira Babar

Signature: _____ Date: 9th June, 2021

Name of the Student: Maryam Akram

Signature: _____ Date: 9th June, 2021

Abstract

Online shopping at this point of time is becoming a major trend in Pakistan and the majority of people are adopting online shopping instead of traditional shopping. Due to the recent pandemic, around 75% of the companies went for E-commerce solutions to sell their products and people bought their daily households through these platforms instead of going out respectively. The major challenge faced by online shopping users is to detect the best quality product items for the least amount of money. Thus “ShopSmart” is an application that will resolve this issue.

The main goal of this project is to assist the online shopping users, compare the product available on different online stores, to check which product is the best possible solution for them. The users can also search the product of their choice and the application will provide the list of the best possible solutions for the searched product in this way they can save money and receive the best value for money. The application also assists the users to apply filters to their search. The working of the system is simple, the user searches the product he wishes to buy, and the system will scrap the product information from Daraz, Telemart, and iShopping, then analyze the product based on users’ reviews, ratings, and price, then the user will be provided with the best possible solution. The user can go to the product details or get redirected to the provider’s website to complete the purchase. In this way the users’ time and money will be saved and they can shop with the satisfaction of trust.

Keywords: Scrap

Dedication

This project is a resolute dedication towards our parents, who are our major inspiration. They have always proved their love for us. Their love, support and devotion made us strong morally, emotionally, mentally as well as spiritually.

Acknowledgments

Firstly, all praises for Allah S.W.T for the strength and blessing He gave us through each step of life and the completion of this project. Without His will and support, it would have been impossible to complete this project.

We would like to show our gratitude to our supervisor Engr. Aleem Ahmed for his time, patience, and knowledge that he shared and his support throughout the project from which we were able to accomplish our thesis as partial fulfilment of the requirements for the degree.

Besides our supervisor, we would thank the faculty of Bahria University Islamabad, for providing the infrastructure and environment that allowed us to pursue our degree, and for providing the facilities such as computer Laboratories.

Finally, we would like to thank our parents, family members, and friends who were fully supportive and encouraging during the time we pursued our degree either financially or mentally. We are thankful to them for their valuable help and comments

Table of Contents

Thesis Completion Certificate	i
Certificate of Originality	ii
Abstract.....	iii
Dedication	iv
Acknowledgments	v
Table of Contents	vi
List of Figures.....	x
List of Tables	xii
Chapter 1	1
Introduction.....	1
1.1. Chapter Overview	2
1.2. Motivation	2
1.3. Problem statement or research questions	2
1.4. Objectives.....	3
1.5. Main contributions	4
1.6. Report organisation	4
Chapter 2	6
Background Study/Literature Review	6
2.1. Chapter Overview	7
2.2. Existing Applications	7
2.2.1. MySmartPrice.com	7
2.2.2. GSMArena.com.....	7
2.2.3. Comparebox.com.....	8
2.2.4. Comparison.....	8
2.3. Research work.....	9
2.3.1. Comparison sites.....	9
2.3.2. The Use of Price Comparison Sites	9
2.3.3. Comparison of Products based on Opinion Feature	10
2.4. Conclusion.....	10
Chapter 3	11
System Requirements	11
3.1. Chapter Overview	12
3.2. Use Case Diagram.....	12
3.2.1. Use Case Diagram of proposed system	12
3.3. Functional Requirements.....	14

3.3.1. Use Case Diagram - Sign Up.....	14
3.3.2. Use Case Diagram - Sign In	15
3.3.3. Use Case Diagram – Search and Compare by Keyword	16
3.3.4. Use Case Diagram – Compare Products by Links.....	17
3.3.5. Use Case Diagram – View Categories.....	18
3.3.6. Use Case Diagram – Compare Daraz Products	19
3.3.7. Use Case Diagram - View Product.....	20
3.3.8. Use Case Diagram - Select Product.....	21
3.3.9. Use Case Diagram - Logout	22
3.4. Interface Requirements.....	23
3.4.1. User Interface.....	24
3.4.2. Software Interface.....	26
3.5. Database Requirements	26
3.6. Non-Functional Requirements	26
3.6.1. Performance Requirements.....	26
3.6.2. Reliability	27
3.6.3. Security	27
3.6.4. Consistency.....	27
3.6.5. Usability.....	27
3.6.6. Portability	28
3.6.7. Availability	28
3.7. Project Feasibility.....	28
3.7.1. Technical Feasibility.....	28
3.7.2. Operational Feasibility.....	29
3.7.3. Legal and Ethical Feasibility	29
3.8. Analysis Models.....	29
3.9. Conclusion.....	30
Chapter 4	31
System Design.....	31
4.1. Chapter Overview	32
4.2. Design Approach.....	32
4.3. System Architecture	32
4.4. Logical Design	33
4.4.1. Domain Model.....	33
4.4.2. Class Diagram.....	34
4.5. Dynamic View.....	35

4.5.1. Sequence Diagram	35
4.5.2. State Machine Diagram	43
4.5.3. Activity Diagram	44
4.6. Component Design	45
4.7. Deployment Diagram	46
4.8. Data Models	47
4.8.1. ER Diagram	47
4.9. User Interface Design	48
4.9.1. Low Fidelity Prototypes - Web Wireframes	48
4.10. System Prototype	52
4.10.1. High Fidelity Prototype	52
4.11. Conclusion	56
Chapter 5	57
System Implementation	57
5.1. Chapter Overview	58
5.2. Strategy	58
5.3. Tools Used	59
5.3.1. VS Code	59
5.3.2. Firebase	59
5.3.3. NMP	59
5.3.4. Selenium	60
5.4. Technologies	60
5.4.1. Frontend Technologies	60
5.4.2. Backend Technologies	61
5.5. Methodology	61
5.5.1. Iterative Development Approach	61
5.6. Conclusion	62
Chapter 6	63
System Testing & Evaluation	63
6.1. Chapter Overview	64
6.2. Test Strategy	64
6.3. Component Testing	64
6.4. Unit Testing	65
6.5. Integrated Testing	65
6.6. System Testing	65
6.7. Test Cases	65

6.7.1. Test Case#1 - Sign Up	66
6.7.2. Test Case#2 - Sign In.....	67
6.7.3. Test Case#3 - Search and Compare by Keyword	68
6.7.4. Test Case#4 - Compare Products by Links.....	69
6.7.5. Test Case# 5 - View Categories.....	70
6.7.6. Test Case#6 - Compare Daraz Products	72
6.7.7. Test Case#7 - View Product	74
6.7.8. Test Case#8 - Select Product	74
6.7.9. Test Case#9 - Logout.....	75
6.8. Results & Evaluation.....	75
6.9. Conclusion.....	75
Chapter 7	76
Conclusion	76
7.1. Chapter Overview	77
7.2. Introduction	77
7.3. Contributions	77
7.4. Reflections.....	78
7.4.1. Strengths	78
7.4.2. Weakness	78
7.5. Future work	78
7.6. Conclusion.....	79
References.....	80
Glossary	81

List of Figures

Figure 1.1: Thesis organisation.....	5
Figure 3.1: System Level Use Case Diagram	12
Figure 3.2: Use case diagram - Sign Up	14
Figure 3.3: Use case diagram - Sign In.....	15
Figure 3.4: Use case diagram - Search and Compare by Keyword	16
Figure 3.5: Use case diagram - Compare Products by Link	17
Figure 3.6: Use case diagram - View Categories.....	18
Figure 3.7: Use case diagram - Compare Daraz Products	19
Figure 3.8: Use case diagram - View Products.....	20
Figure 3.9: Use case diagram - Select Product	21
Figure 3.10: Use case diagram - Logout	22
Figure 3.11: System Flowchart	30
Figure 4.1: Client Server Architecture.....	33
Figure 4.2: Domain Model	33
Figure 4.3: Class Diagram	34
Figure 4.4: Sequence Diagram - Sign Up.....	35
Figure 4.5: Sequence Diagram - Sign In.....	36
Figure 4.6: Sequence Diagram - Search and Compare by Keyword	37
Figure 4.7: Sequence Diagram - Compare Products by Links.....	38
Figure 4.8: Sequence Diagram - View Category	39
Figure 4.9: Sequence Diagram - Compare Daraz Products	40
Figure 4.10: Sequence Diagram - View Product.....	41
Figure 4.11: Sequence Diagram - Select Product.....	42
Figure 4.12: Sequence Diagram - Logout	42
Figure 4.13: State Transition Diagram	43
Figure 4.14: Activity Diagram	44
Figure 4.15: Component Diagram.....	45
Figure 4.16: Deployment Diagram.....	46
Figure 4.17: Deployment Diagram.....	47
Figure 4.18: Low fidelity - Sign Up	48
Figure 4.19: Low fidelity - Sign In.....	49
Figure 4.20: Low fidelity – Homepage	49
Figure 4.21: Low fidelity – Categories.....	50
Figure 4.22: Low fidelity – Search and Compare by Keyword.....	50

Figure 4.23: Low fidelity – Compare by Links	51
Figure 4.24: Low fidelity – Daraz Product Selector.....	51
Figure 4.25: Low fidelity - Searching and Comparing.....	52
Figure 4.26: High fidelity - Sign up	52
Figure 4.27: High fidelity - Sign In	53
Figure 4.28: High fidelity - Homepage	53
Figure 4.29: High fidelity – Compare Products	54
Figure 4.30: High fidelity - Categories.....	54
Figure 4.31: High fidelity – Search and Compare by Keyword.....	55
Figure 4.32: High fidelity – Scraping and Comparing Information	55
Figure 4.33: High fidelity - Best to Buy.....	56
Figure 5.1: Strategy Flow	58
Figure 5.2: Iterative Development Approach	62

List of Tables

Table 2.1: Differences and similarities between the existing applications	8
Table 3.1: Use Case Description - Sign Up	15
Table 3.2: Use Case Description - Sign In.....	16
Table 3.1: Use Case Description - Search and Compare by Keyword	17
Table 3.4: Use Case Description - Compare Products by Links.....	18
Table 3.5: Use Case Description - View Categories.....	19
Table 3.6: Use Case Description - Compare Daraz Products	20
Table 3.7: Use Case Description - View Product	21
Table 3.8: Use Case Description - Select Product	22
Table 3.9: Use Case Description - Logout.....	23
Table 6.1: Test Case # 1 - Sign Up	66
Table 6.2: Test Case # 2 - Sign In.....	67
Table 6.3: Test Case # 3 - Search and Compare by Keyword	68
Table 6.4: Test Case # 4 - Compare Products by Links.....	70
Table 6.5: Test Case # # 5 - View Categories.....	71
Table 6.6: Test Case # 6 - Compare Daraz Products	73
Table 6.7: Test Case # 7 - View Product	74
Table 6.8: Test Case # 8 - Select Product	74
Table 6.9: Test Case # 9 - Logout.....	75

Chapter 1

Introduction

1.1. Chapter Overview

The first chapter of the thesis report builds a brief idea of what the proposed system “ShopSmart” is and what is the need behind having it. It also mentions the motivation for the production of this application and its usefulness along with background information. It also provides goals and objectives we are going to achieve.

1.2. Motivation

Online shopping at this point of time is becoming a major trend in Pakistan and the majority of people belonging to socio-economic class “B” are adopting online shopping instead of traditional shopping. Due to the recent pandemic, around 75% of the companies went for E-commerce solutions to sell their products and people bought their daily households through these platforms instead of going out respectively.

As statistics assure us, more than 80% of users take the advantage of special websites to find the cheapest prices but the problem is that the special websites only compare the product based on price. There is a need for a system that compares the products not only based on price but also quality thus detecting the best quality product items for the least amount of money is the main challenge.

The thesis aims to propose a solution to the above problem. The system will generate the best possible solution for the desired user product, by comparing the user reviews, ratings, and, prices of three different online shopping stores i.e., Daraz, Telemart, and iShopping. The user will search the product that he desires, the system will look up for the product on the above online store, then the system will the product information such as, the product picture, specification, reviews, and price, the system will compare the users’ reviews, ratings and price and analyze that which store offers the best quality item for the least amount of money. The user will also be redirected to the provider’s website to complete the purchase.

1.3. Problem statement or research questions

Inflation has brought great impact to everyone, and people like to spend their money wisely, they like to compare the quality of product before purchasing so that they can

choose the better one. Besides, considering the customers' busy lifestyle, most of the customers prefer to buy their needs through online shopping stores to save their time.

The customer's lookup for comparison websites so that they don't have to travel from shop to shop only to survey the price offered by different shops for the same product. There are price comparison websites like CompareBox.pk or PriceBlaze.pk, but the feature they lack is the quality comparison. Thus, there was a need for a website that compares the item based on reviews, ratings, and price therefore "ShopSmart" is the perfect solution.

1.4. Objectives

The Following objectives are to be accomplished when the system is developed:

1. A mobile and web application would be created using the Visual Studio Code tool.
2. Allow users to create their profile, so that their search history is maintained.
3. Allow users to search for the product by entering the keyword. The results of the search should be presented quickly keeping in mind the customers' busy lifestyle.
4. The user can also compare the product of two stores, by providing their links, and the system will guide the user, which one of them is the right choice to purchase.
5. The user can look up for the product in the desired categories section i.e., phones, air conditioner, LED televisions, refrigerators.
6. Through web scraping, the searched product information is scrapped from the particular online store for instance product reviews, ratings, price, and specifications.
7. By using the text mining technique, the best possible solution would be presented to the user.
8. The user will be provided with the link to the provider's online store where the product is available so that the user can make the purchase.

1.5. Main contributions

There are many online price comparison applications available but at the time of inflation, the customers not only want to save the money but also want to receive the best value for the money. Thus the “ShopSmart” application will compare the products of various online stores based on user reviews, ratings, and price so that the product quality could be analyzed along with the price. The user can also compare the products of online stores, and get the best possible solution. The user will also be provided with an opportunity to compare two daraz products. The application also provides the user with the electronics category to ease their search process.

ShopSmart application is an online platform to help customers spend their money wisely and make an informed decision before purchasing the product by providing them with the best possible solution. Users can use this application as a reference to check which store offers the better product and is more reasonable to approach.

Instead of taking hours and energy to go to each store just to check the price, ratings, and reviews about a particular product, ShopSmart offered a better solution by providing the user with the best possible solution to their search and the user just needs to go to the online store to make the purchase.

The document now reveals the clear-cut idea of what is going to be developed and the reader is well aware of the project, why is it important, how would it help the online shopping users make the informed decisions.

1.6. Report organisation

Chapter 1 of this thesis report builds a brief idea on what the proposed system “ShopSmart” is and what is the need behind having it. It also mentions the motivation for the production of this application. It also provides our contributions, goals, and objectives that we are going to achieve.

Chapter 2 covers the literature review of the thesis report. This section includes the past work done and the contributions made earlier in this field and it also portrays our contribution and work that is new and significant.

Chapter 3 focuses on all the requirements of our project, the interface requirement, functional requirements as well as non-functional requirements, and a detailed explanation of Use cases of the entire system.

Chapter 4 includes all the pictorial representations of our project, the design strategies, and design models.

Chapter 5 discusses all the implementation details, strategy, the tools, techniques, and methodology used to implement the project.

Chapter 6 consists of all the testing strategies and test cases that we made to test every module of our system.

Chapter 7 concludes the entire thesis report with some limitations and future work that can be carried out in the future.

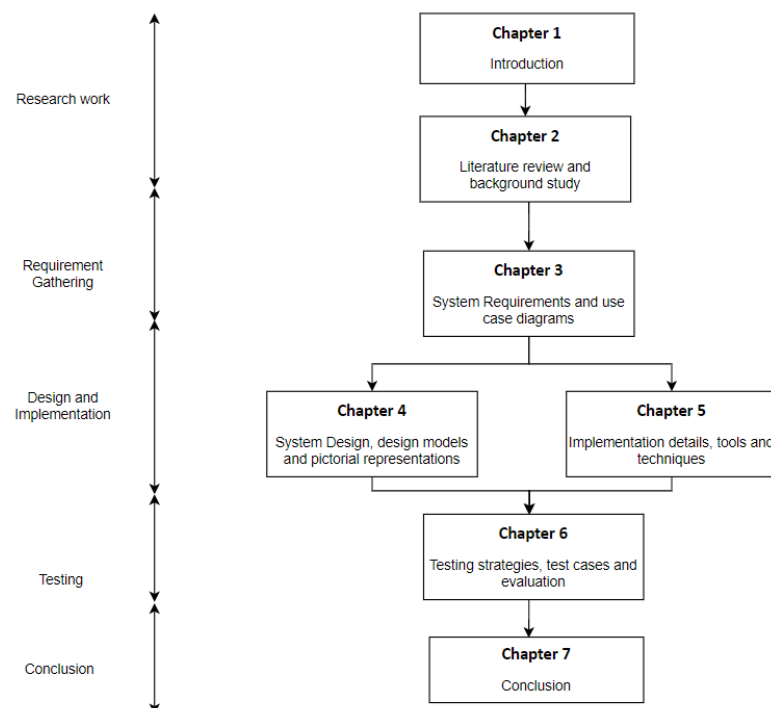


Figure 1.1: Thesis organisation

Chapter 2
Background Study/Literature
Review

2.1. Chapter Overview

The general objectives defined in this section are firstly to highlight the strengths and the weaknesses of an existing application, to set up the essentialness of the general research made by previous researchers, and then identify a portion where another enhancement could be made. This chapter will also portray our contribution and work that is new and significant.

2.2. Existing Applications

The three existing or similar applications are reviewed and analyzed to compare their features to innovate the ShopSmart application. The applications are MySmartPrice.com, GSMArena.com, and CompareBox.com.

2.2.1. MySmartPrice.com

It is a price comparison application.

Strengths

- The prices are displayed under the product with a buy now button.
- The history of prices is shown through the price graph. This feature helps the user in buying decisions as they know about the price flow.

Weaknesses

- The price graph is the strength of this application but not every product has a price graph.
- If a user sets an alert for a certain item, they cannot trace back for what product the alert was set for.
- It does not guide the user that which product is more suitable to buy.

2.2.2. GSMArena.com

GSMArena.com is a price and specification comparison website.

Strengths

- It provides detailed and accurate information about mobile phones and their features.

- It also provides links to the online store that sell phones and electronics but those are external entities.

Weaknesses

- It allows the user to compare the specifications of two or three devices, but it does not suggest to the user, which device is better to purchase.

2.2.3. Comparebox.com

It is a product comparison platform.

Strengths

- Allows the user to compare the prices of products from a wide range of online stores like Daraz.pk, Dreams.pk, Yayvo.com, etc.
- The website offers the user to compare from six different categories such as Cars, Mobile Phones, Laptops, and Motor Bikes, etc.
- Six different products can be compared side by side.

Weaknesses

- It allows the user to compare the specifications of more than two devices, but it does not suggest to the user, which device is better to purchase.

2.2.4. Comparison

Table 2.1: Differences and similarities between the existing applications

Characteristics	MySmartPrice	GSMarena	CompareBox
Registration	✓	✗	✗
Search feature	✓	✓	✓
GUI Design	✓	✓	✓
Adequate Categories	✓	✗	✓
Adequate Information	✓	✓	✓
Add product to favourite	✗	✗	✗
Online purchasing	✓	✗	✗

2.3. Research work

To increase the credibility of the project, research work and articles are analyzed related to the price comparison websites. For the literature review of this project, three research articles are being used as the main reference.

2.3.1. Comparison sites

The “Comparison sites” research paper was composed by Moraga-Gonzalez J.L and Wildenbeest M.R and it was distributed in July 2011. [1]

The research paper focuses on the comparison sites and how they contribute towards the market proficiency and price rivalry. These days buyers first search and examine the product of their need on the internet before making any decision for purchasing which is very much convenient than those days when there was no internet. Initially, the price comparison applications were used to act as online classified but these days they can do a great deal of work, for example, they can draw the product price, search sites, and give a larger control to the user with the help of the latest technologies. The price comparison sites collect relevant information about a product from a third party and present it to the consumers.

The comparison sites are very valuable for the further analysis of data for instance how precisely customers look for items and also these signs give signs about the most ideal approach to improve the current framework.

2.3.2. The Use of Price Comparison Sites

The research paper named “The Use of Price Comparison Sites in the UK General Insurance Market” is composed by Emily Knight. [2] She was a policymaker for buyer Intelligence has reported the present performance, media coverage, usage, and marketing activity of price comparison sites within the UK General Insurance sector.

The number of users that use the price comparison sites has expanded over time. According to research, in the future 8 out of 10 people are likely to use the price comparison sites for the product of their need before purchasing them. This indicates that the price comparison sites will remain in business in the future.

2.3.3. Comparison of Products based on Opinion Feature

The research paper named “Automatic Comparison of Products based on Opinion Features using Synonym and Jaccard Similarity” is composed by Agus Zainal Arifin, Dinial Utami Nurul Qomariah, Muhammad Riduwan, Addien Haniefardy, Yufis Azhar, Rizka Wakhidatus Sholikah, and Dini Adni Navastara, in October 2018. [3]

The research paper emphasis that the quality of the product recommendations can be improved by the opinion feature. The comparison of the product based on the opinion feature has been discussed by some previous researches but the approach discussed in the previous researches are less effective because they weigh the product on the similar features and does not consider the other relevant features. Whereas, this research paper introduces new strategies that compare the important features using synonym and Jaccard similarity for automatic product comparison. The result of this research paper shows that the new strategy gets up to 90% accuracy.

This research introduced strategies like preprocessing, feature extraction, feature weighing, checking synonym features, calculating similarity, and product scoring. All these new strategies were very helpful for us in the implementation of ShopSmart application.

2.4. Conclusion

This chapter covered the literature review of the thesis report. It included the past work done and the contributions made earlier in this field and it also portrays the special techniques and platforms we are going to use to implement the ShopSmart application.

Chapter 3

System Requirements

3.1. Chapter Overview

This chapter briefly describes the requirements required to complete this project. Some of the requirements that would be discussed in this chapter are interface requirements, database requirements, functional and non-functional requirements. This chapter also includes complete use case description.

3.2. Use Case Diagram

This section represents the use case diagrams of the ShopSmart application.

3.2.1. Use Case Diagram of proposed system

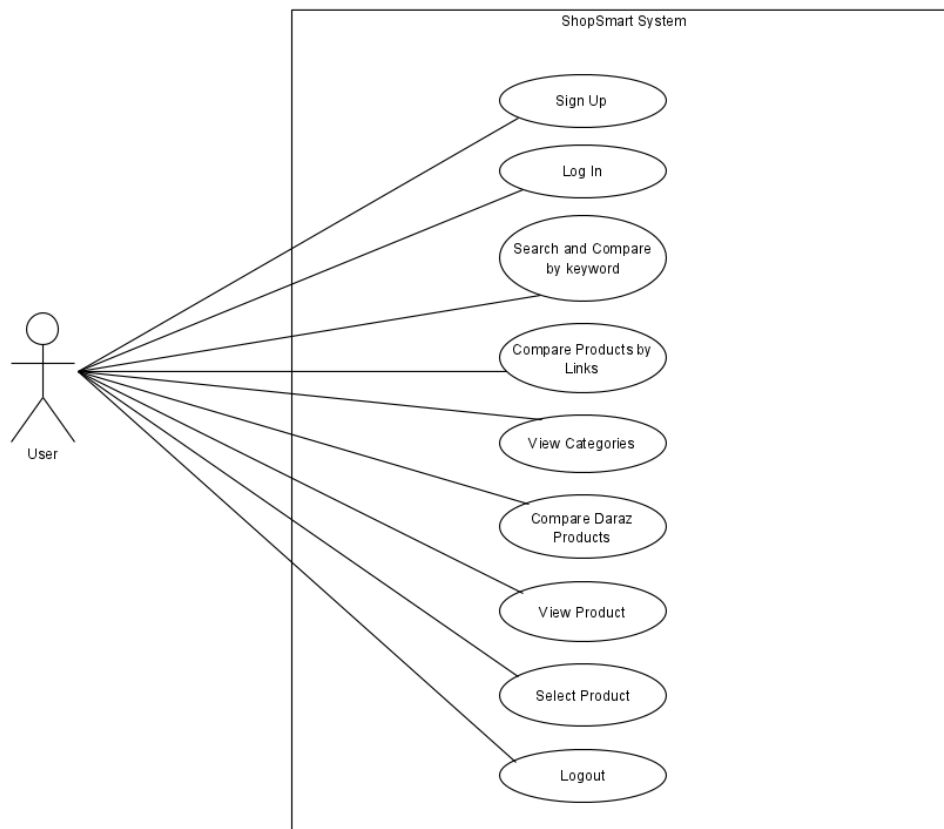


Figure 3.1: System Level Use Case Diagram

Explanation:

The use-diagram illustrated the function to be performed by the actors to achieve their desired goals. In the ShopSmart system there is only one actor i.e., the user. User is any person who wants to make a comparison of products before purchasing them,

User: User is a person who wants to purchase the best quality product at a reasonable rate. The user does not have to travel from shop to shop only to survey the price offered by different shops for the same product

Sign up and Login: To the system, user must register himself and then he can login to the system after registration. So, Sign Up and Log In use-cases will be executed, respectively. After the successful login, user will be provided by four options, among the options are Search and Compare by Keyword, Compare by Links, Categories, and Daraz Product Selector.

Search and Compare by Keyword: The user will be provided by a search bar along with three checkboxes, named after each store he wishes to search the product on. After entering the keyword and pressing the search button, the user will be presented by the store name and the desired product which is suitable to purchase.

Compare by Links: The user can place the link of the product available on Daraz against the link of the product available on iShopping.pk. The system will generate the result of which store offers the better product.

View Categories: There are four categories provided to the user, including Mobile phones, air conditioners, LED Televisions, and Refrigerators. All these categories consist of compared products from Daraz and iShopping stores indicating best one to buy.

Compare Daraz Products: The user can also compare two products available on daraz website by entering their links to the respective fields.

View Product: When the user searches the desired product, the user will be provided by its image, specifications, ratings, and price. He can also visit the provider website.

Select Product: The user can also select the product by visiting the providers website. The link will be provided when the user searches for an item, this will provide an ease for purchasing the product.

The user can logout of the system after using it which will execute Logout use-case

3.3. Functional Requirements

3.3.1. Use Case Diagram - Sign Up

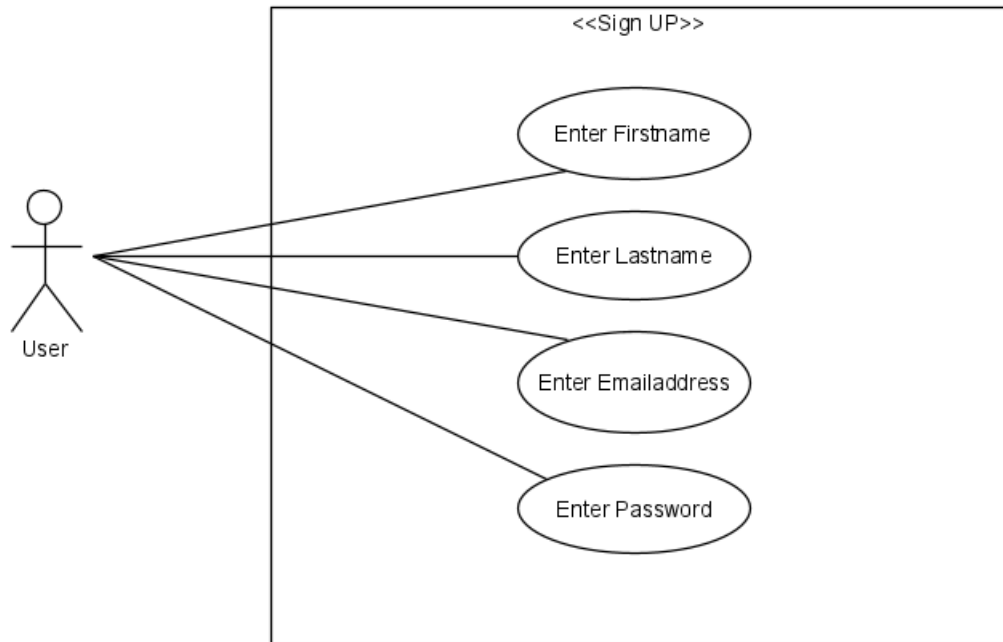


Figure 3.2: Use case diagram - Sign Up

Use Case ID:	UC -01	
Use Case Name:	Sign Up	
Actor(s):	User	
Pre-Conditions:	The user's device should be connected to the internet and the user should be on ShopSmart website.	
Priority:	High	
Basic Flow:	The user fills the sign-up form, clicks on 'Sign Up' button and the user get registered.	
Actor Actions		System Response
1.	a. User fills the Sign-up form. The form contains the following fields: <ul style="list-style-type: none"> ○ First Name (Required) ○ Last Name (Required) ○ Email address (Required) ○ Password (Required) 	2. The system validates the credentials and creates the user account.

	b. User clicks on ‘Sign Up’ button under the form.	
Alternative Course of Action (if any)		
Actor Action		System Response
1b. i	User has left required fields blank.	1b.i. 1 System displays error message: “Field is required!”
1b. ii	User has entered the email that is already registered.	1b. ii.1 System displays error message: “Email already exists!”

Table 3.1: Use Case Description - Sign Up

3.3.2. Use Case Diagram - Sign In

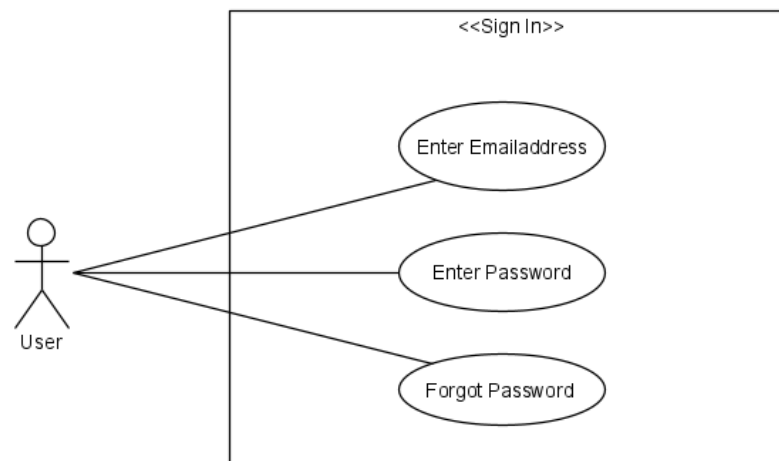


Figure 3.3: Use case diagram - Sign In

Use Case ID:	UC -02
Use Case Name:	Sign In
Actor(s):	User
Pre-Conditions:	User already has an account in the system.
Priority:	High
Basic Flow:	The user can log into their account by using their respective email and password.
Actor Actions	System Response

1.	The user enters their email and password into the respective fields then the user clicks on a sign-in button.	2.	The system validates the credentials entered by the user and displays the home page.
3.	User clicks on forgot password button.	4.	An email is sent to the users' account to reset the password.
Alternative Course of Action (if any)			
Actor Action		System Response	
1.	User has entered wrong Email address and/or password	1.a.	An error message is displayed: "Email/ password" is Incorrect!

Table 3.2: Use Case Description - Sign In

3.3.3. Use Case Diagram – Search and Compare by Keyword

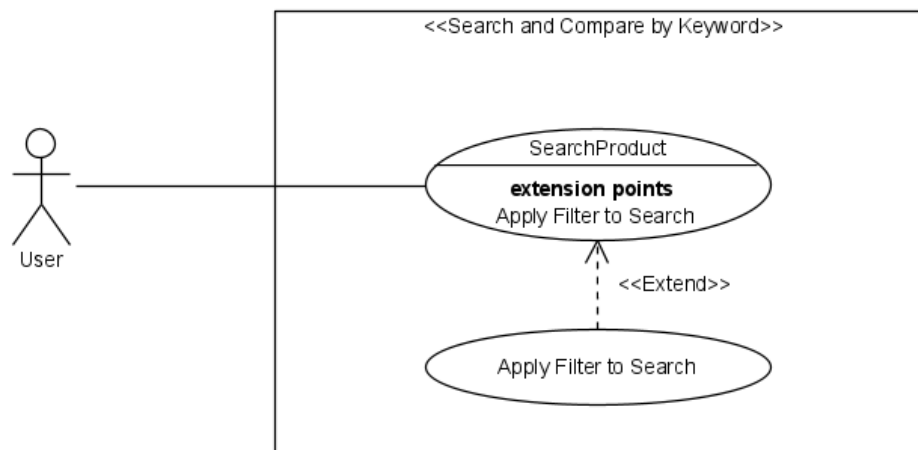


Figure 3.4: Use case diagram - Search and Compare by Keyword

Use Case ID:	UC -03
Use Case Name:	Search and Compare by Keyword
Actor(s):	User
Pre-Conditions:	The user should be logged in to their account.
Priority:	Medium
Basic Flow:	The user will search for the item of their need and the system will display the best possible solution available for them from various online stores.

Actor Actions		System Response
1.	a. The user enters the product name in the search bar and check the checkboxes of store from where he looking for the desired product. b. User clicks on the “search” button.	2. System will search on Daraz, iShopping, and Telemart whether the desired product is present. 2.1. If the product is available in the above-mentioned stores, the system will compare their reviews, rating, and prices and display the user with the best possible solution.
Alternative Course of Action (if any)		
Actor Action		System Response
1.a	The user searched for any item that is not available.	2.a. The system will display the “not found” message.

Table 3.1: Use Case Description - Search and Compare by Keyword

3.3.4. Use Case Diagram – Compare Products by Links

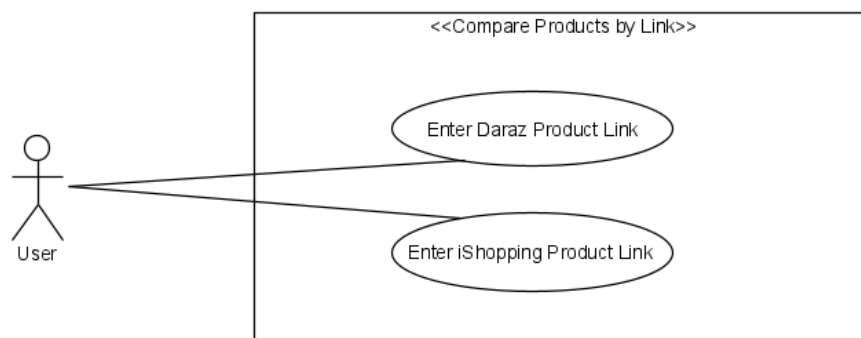


Figure 3.5: Use case diagram - Compare Products by Link

Use Case ID:	UC -04
Use Case Name:	Compare Products by Link
Actor(s):	Medium
Pre-Conditions:	The user should logged in to their account.
Priority:	High

Basic Flow:		The user will insert the link of the product they wanted to compare from Daraz and iShopping online stores, and click on the Scrape and compare button. The system will inform the user which store offers the best solution.
Actor Actions		System Response
1.	a. User enters the Daraz Product Link b. User enters the iShopping product Link c. User presses the button “Scrape and Compare”.	2. The system will scrap the information from the desired stores. Compare the product reviews, ratings, and prices available on stores. 2.1. The system will inform the user, which store’s product is better to buy.
Alternative Course of Action (if any)		
Actor Action		System Response
1.	The user enters an incorrect link	2.a. The system will display a “not found” message.

Table 3.4: Use Case Description - Compare Products by Links

3.3.5. Use Case Diagram – View Categories

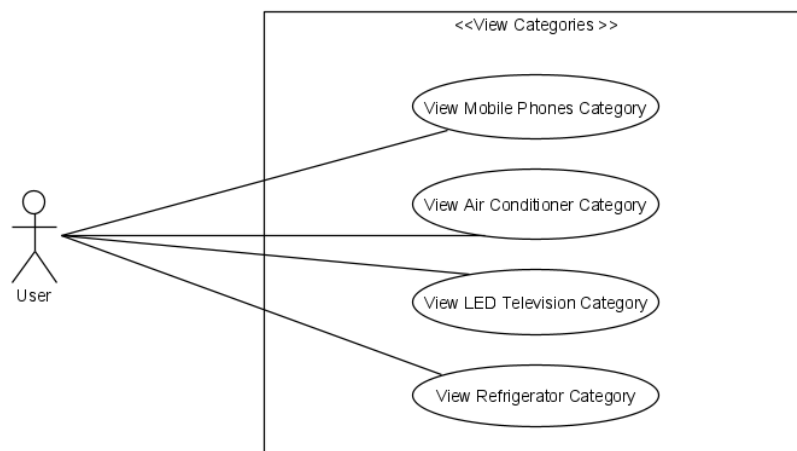


Figure 3.6: Use case diagram - View Categories

Use Case ID:	UC -05	
Use Case Name:	View Categories	
Actor(s):	User	
Pre-Conditions:	The user should be logged in to their account	
Priority:	Medium	
Basic Flow:	The user can view different categories and get to know the best product to buy compared from Daraz and iShopping.	
Actor Actions		System Response
1.	The user views the mobile phone category.	2. The system will present the best to buy compared products from Daraz and iShopping.
3.	The user views the air conditioner category.	4. The system will present the best to buy compared products from Daraz and iShopping.
5.	The user views the refrigerator category.	6. The system will present the best to buy compared products from Daraz and iShopping.
7.	The user views the LED Television category.	8. The system will present the best to buy compared products from Daraz and iShopping.
Alternative Course of Action (if any)		
Actor Action		System Response
	None	None

Table 3.5: Use Case Description - View Categories

3.3.6. Use Case Diagram – Compare Daraz Products

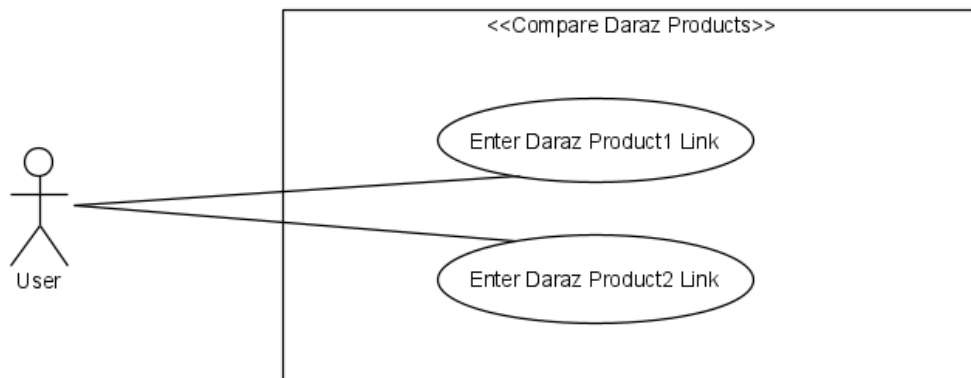


Figure 3.7: Use case diagram - Compare Daraz Products

Use Case ID:	UC -06	
Use Case Name:	Compare Daraz Products	
Actor(s):	Medium	
Pre-Conditions:	The user should logged in to their account.	
Priority:	High	
Basic Flow:	The user will insert the link of two products available on Daraz and click on the Scrape and Compare button. The system will inform the user which one is the best solution.	
Actor Actions		System Response
1.	a. User enters the Daraz Product 1 Link b. User enters the Daraz Product 2 Link c. User presses the button “Scrape and Compare”.	2. The system will scrap the information of both products and compare the product reviews, ratings, and prices available on stores. 2.1. The system will inform the user, which product is better to buy.
Alternative Course of Action (if any)		
Actor Action		System Response
1.	The user enters an incorrect link	2.a. The system will display a “not found” message.

Table 3.6: Use Case Description - Compare Daraz Products

3.3.7. Use Case Diagram - View Product

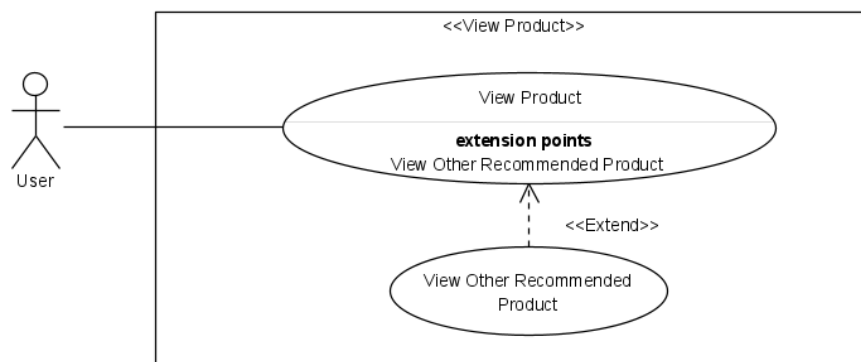


Figure 3.8: Use case diagram - View Products

Use Case ID:	UC -07	
Use Case Name:	View Product	
Actor(s):	User	
Pre-Conditions:	The user should have searched an item.	
Priority:	Medium	
Basic Flow:	The system will display the specifications of the searched product.	
Actor Actions		System Response
1.	a. The user enters the product he wants to search b. The user clicks on the search button.	2. System will display the search results along with the products image, specification, price and rating.
Alternative Course of Action (if any)		
Actor Action		System Response
1.	The user searched for any item that is not available.	2.a. The system will display the “not found” message.

Table 3.7: Use Case Description - View Product

3.3.8. Use Case Diagram - Select Product

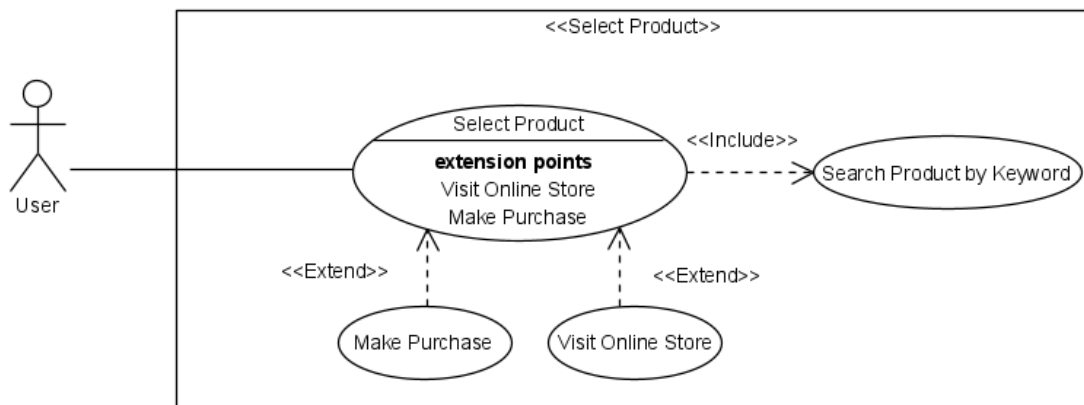


Figure 3.9: Use case diagram - Select Product

Use Case ID:	UC -08	
Use Case Name:	Select Product	
Actor(s):	User	
Pre-Conditions:	The user should have searched an item.	
Priority:	Medium	
Basic Flow:	The user will be provided by the link of the searched product store so that the user can visit the respective stores where they can make their purchase.	
Actor Actions		System Response
1.	a. The user wants to purchase a product from the given list and clicks on the link available below the product specifications.	2. The system will take the user to the desired online store website where the product is available, and then the user can make their purchase if they want.
Alternative Course of Action (if any)		
Actor Action		System Response
1.	The user searched for any item that is not available.	2.a. The system will display the “not found” message.

Table 3.8: Use Case Description - Select Product

3.3.9. Use Case Diagram - Logout

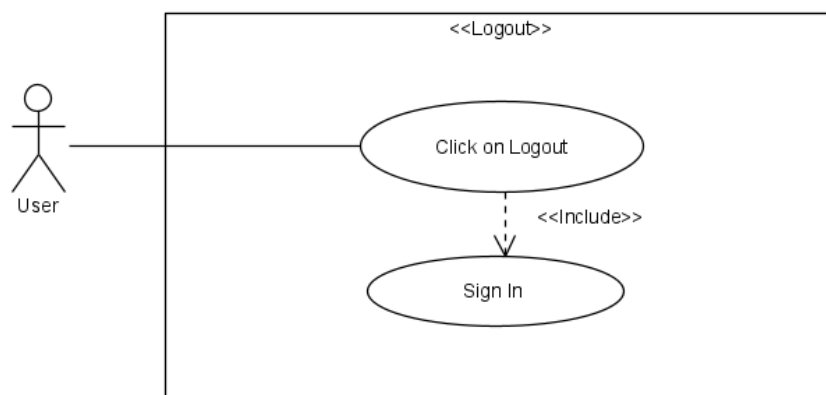


Figure 3.10: Use case diagram - Logout

Use Case ID:	UC -09	
Use Case Name:	Logout	
Actor(s):	User	
Pre-Conditions:	The user is logged in to their account.	
Priority:	Medium	
Basic Flow:	The User clicks on the ‘Logout’ button and is logged out.	
Actor Actions		System Response
1.	User clicks on the ‘Logout’ button	2. System logs out the user and displays respective Login Page
Alternative Course of Action (if any)		
Actor Action		System Response
	None	None

Table 3.9: Use Case Description - Logout

3.4. Interface Requirements

An interface is such a bridge through which components of a system or two or more mediums interact. Identifying this interface can allow you to set your system boundaries and goals. It can also assist you in identifying dependencies your system has with other systems. Failing in successfully identifying interface requirements can result in system failure or may exceed the expected budget. To write interface requirements there are three steps:

1. Identifying the interfaces.
2. Defining the interface requirements.
3. Writing the interface requirements.

For a system to work properly, there are mainly three types of interface requirements that are needed to be fulfilled.

1. User Interface

2. Software Interface
3. Hardware Interface

As our project does not contain any hardware, we are going to explain only our user and software interfaces.

3.4.1. User Interface

3.4.1.1. Sign In page

The user interface begins with a sign-in page through which only registered users can log into their respective accounts. The user is required to have a registered email, and password to use the services of the application. This page contains a sign-in button that leads to the home page of the application if the user credentials are correct.

3.4.1.2. Sign Up page

If the user is new to the application, they can click on the sign-up button on the sign-in page, leading to the sign-up page. On this page, the users are required to enter their first name, last name, email address and a password. After this the user clicks on the “Sign up” butt

3.4.1.3. Home page

The home page of the application consists of four buttons, all leading to multiple pages for different activities.

1. The first button is “Search and Compare by Keyword”, this page allows the user to search any product using keyword.
2. The second button is “Compare by Links”, this page allows the user to compare the products of two stores using the link page of the product.
3. The third button is “Categories”, this page allows the customer to view the four categories. In which the best to buy product is suggested to the customer from Daraz and iShopping store.
4. The fourth button is “Daraz Product Selector”, this page allows the user to compare daraz products.

3.4.1.4. Search and Compare by Keyword page

On the home page, the user clicks on the “Search and Compare by Keyword” button. The user is displayed with a new page. On the top of this page, there is a bar from which the user can log out or go back to the home page. The page consists of a search bar along with three checkboxes named Daraz.pk, iShopping.pk and Telemart.pk. The check boxes are named after the stores from where the system is capable of searching the product. The user can select the stores as per choice and the system will search the product only on those selected stores. Beneath the checkboxes, there is a search button.

3.4.1.5. Compare by Links

On the home page of this application, there is a button ‘Compare by Links’, when the user clicks on the button, he is displayed with a new page that allows them to compare the product of two different stores i.e., Daraz and iShopping.

On the top of this page, there is a bar from which the user can log out or go back to the home page. The page provides the user with two text fields, with the placeholder of the respective stores i.e., Daraz and iShopping, along with a button name ‘Search and Scrape’. Once the user places the correct link in the text field and presses the button, the progress bar is shown and then the results are displayed.

3.4.1.6. Categories

On the home page of this application, there is a button ‘Categories’, when the user clicks on the button, he is displayed with a new page that consists of four categories which are Mobile Phones, Air Conditioner, LED Televisions, Refrigerators. Each of the category consists of few products that are compared on the based of user reviews, ratings and price. This page suggests user which product is best to buy between the products from Daraz and iShopping.

3.4.1.7. Daraz Product Selector

On the home page of this application, there is a button ‘Daraz Product Selector’, when the user clicks on the button, he is displayed with a new page that allows them to compare two Daraz products.

On the top of this page, there is a bar from which the user can log out or go back to the home page. The page provides the user with two text fields, with the placeholder Daraz Product 1 link and Daraz Product 2 link, along with a button name ‘Search and Scrape’. Once the user places the correct link in the text field and presses the button, the progress bar is shown and then the results are displayed.

3.4.1.8. Log out

On every page of the application, on the top right, user is provided with the button “Logout” which signs off the user from their account.

3.4.2. Software Interface

Front-end: HTML, CSS, Bootstrap, JavaScript, React.js

Back-end: Python, Node.js, firebase

3.5. Database Requirements

Following are the database requirements:

1. All data must be stored in Firebase database.
2. User information taken at the time of Sign Up like name, email address and password should be stored in the firebase database (Authentication).
3. User information must be accessible only by them at any time.
4. The results of search and scrapping must also be stored in the firebase database (Realtime).
5. The product title, rating, price, link and image must also be stores in the database.

3.6. Non-Functional Requirements

3.6.1. Performance Requirements

1. The product load time depends on the internet connection strength and the media from which it is using this application.
2. The authorization process should not take than 10 second.

3. The end user's hardware components might also affect the performance of the application.
4. The data access speed is important and the application must be scalable to handle the concurrent users.
5. The overall efficiency of the system should be maintained.

3.6.2. Reliability

The reliability of components of the system determines the reliability of the entire system. All components must show enough reliability so that not even a single action leaves the user ambiguous. All commands must lead to accurate actions in return.

1. The product should be searched instantly without any hassle to the user.
2. The scraper should accurately get the information and present it to the user
3. The system must be reliable enough that maintenance is permissible.

3.6.3. Security

1. Only authorized users can access the system with a valid Email and Password.
2. The reset password link should only be sent to user's email to keep user's account secure.
3. The data stored in database must be kept private and it must only be accessible by the user
4. For web scraping, we need high-quality proxy, so that the IP does not gets blocked from the system.

3.6.4. Consistency

For better performance, the system should be consistent.

1. Accurate results should be given with each command.
2. The system shall work under excessive pressure.
3. Load should be easily managed by the system.

3.6.5. Usability

1. The web and mobile applications should be easy to use and easy to understand.
2. The design of both applications should not be complicated.

3. The navigations should be simple, and icons and headings should be self-explanatory.
4. All relevant information should be easily available to the respective user.

3.6.6. Portability

1. The ShopSmart application is HTML-based so, the end-user part is fully portable and any system using any web browser should be able to use the features of the system, including any available hardware platform.
2. The end-users will use this system on any operating system either it is Linux or windows.
3. The system shall run on personal computers, laptops, and mobile phones.

3.6.7. Availability

1. The ShopSmart application will be available 24/7.
2. The user can access it using a web browser or on his smartphones anytime he wants to have a compared product result from the three most popular online shopping stores.

3.7. Project Feasibility

This study is to essentially analyze the system to determine whether the system being developed is worthy of investing money and time. Our project can be proven to be a useful product because it eliminates most of the difficulty faced by online shopping users of finding the right product at the right place. This application can be used to compare the products of three different stores based on reviews, ratings, and price. Such types of applications are needed since most of the applications only provide the comparison of products based on price. Thus, ShopSmart will help the users shop in a better way.

3.7.1. Technical Feasibility

This software application is very easy to use. Very little cost was required in building the project. This application is technically feasible at a minimum cost and gives higher benefits in return. Little to no training is required for this application to be used by any of the users, due to a simple and user-friendly interface. ShopSmart uses express.js framework, firebase database, HTML, CSS, Bootstrap, JavaScript, Node.js,

React.js, and Python as the programming language. All the requirements of this project are easily available and easy to implement. The ease of implementation of these technologies makes this project technically feasible.

3.7.2. Operational Feasibility

Operational feasibility can be defined as a process through which a proposed system would be able to solve problems that are faced by users. This application would help the online shopping users know from where they can purchase the right product for themselves. It will eliminate the hazard of going to stores to stores and searching for the right product. This application would allow the users to compare any product available on three respective stores (Daraz.pk, Telemart.pk, iShopping. pk) by providing the link to the desired field, and the system will generate the result. The ShopSmart application will also provide the user with the link to the particular store so that the user can purchase his product and with the help of a proxy their IP address will not be blocked. This application would be able to operate on every mobile, laptop, and computer.

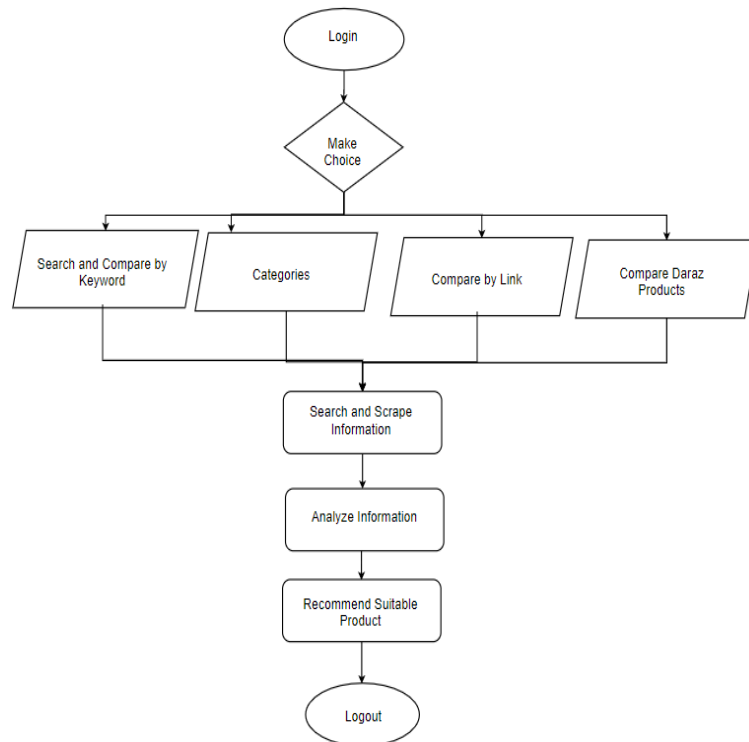
3.7.3. Legal and Ethical Feasibility

The following application is ethically and legally feasible

1. Wrong and unethical means were not used in developing this application.
2. “ShopSmart” is a useful application for online shopping users and it can not cause any possible harm.
3. Our application respects the privacy of every user and will not reveal any confidential data.
4. Shopping online and making comparisons would become easier with this application.

3.8. Analysis Models

The user is first required to log in to the system. The user is allowed to make a choice among the four choices given. If the user chooses the search and compare by keyword, he has to provide the products’ keyword. He can view the categories. If the user wants to compare the products of daraz and iShopping, he has to opt for compare



by links. The user can also compare daraz products. After making the choice, the system searches and scraps the data. Analyze the products’ reviews, rating and price. The system recommends a suitable product to the user. The user can either purchase the product by visiting the website.

Figure 3.11: System Flowchart

3.9. Conclusion

After analyzing this chapter, a reader would get an overall general idea of the interface requirements, the functional requirements as well as non-functional requirements. This chapter also includes use case diagrams along with the use case descriptions. We have briefly discussed the user interface and software interface of our project. All functional requirements are also discussed briefly so that the reader can get an insight into how the application will operate, how its different modules will interact, and what functionalities are to be performed by the system. Non-functional requirements are stated in this chapter to inform the user of the features of the system.

Chapter 4

System Design

4.1. Chapter Overview

This chapter contains the approaches and design constraints that are used to make this project. This chapter also includes the low fidelity and high-fidelity design of our application along with sequence diagram, activity diagram and class diagram.

4.2. Design Approach

In this project we have used an Object-Oriented Design (OOD) approach that is a part of Object-Oriented Programming (OOP). With this methodology, the system is defined in terms of the classes and the associations between them. Each class consists of class name, class attributes and the operations to be performed by the class.

4.3. System Architecture

The system uses the client server architecture, as the user requests for a product (client) and system generates the response (server).

Client Server Architecture:

Client-server architecture is a computing model in which the server hosts, delivers and manages most of the resources and services to be consumed by the client. This type of architecture has one or more client computers connected to a central server over a network or internet connection

- **Client:** a piece of software or application that takes the input and sends request to the servers.
- **Server:** a piece of software that receives and processes requests from clients.

A typical topology data flow goes as follows:

- Client requests data from server
- Server processes the request client
- Server queries appropriate database for some data
- Database returns the queried data back to the server
- The server processes the data and sends the data back to the client
- This process repeats.

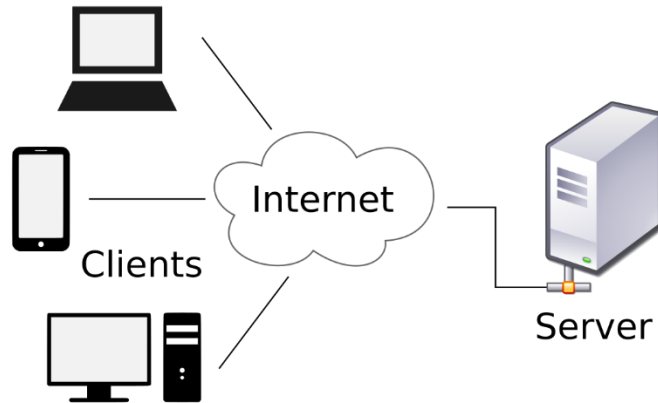


Figure 4.1: Client Server Architecture

4.4. Logical Design

4.4.1. Domain Model

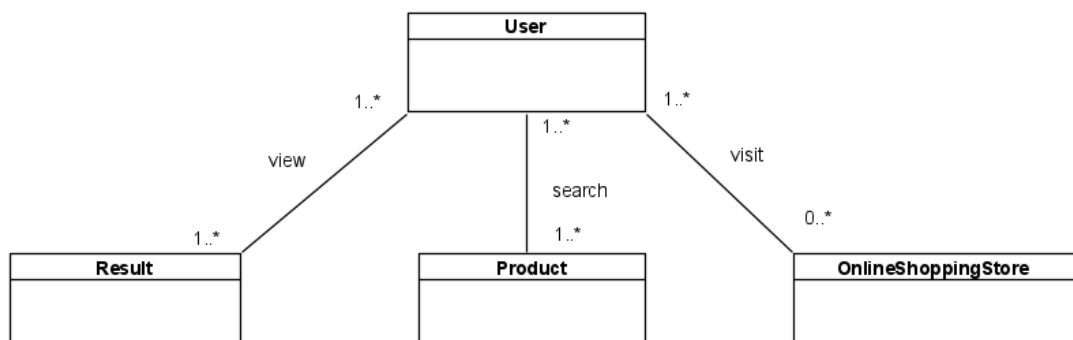


Figure 4.2: Domain Model

Explanation

Domain model represents the real-world conceptual classes not software components. it represents the classes and relationship between those classes. The domain model of ShopSmart is given in the figure above.

In the given model, one or more users can search for a product, view the results and the user can visit the online shopping stores.

4.4.2. Class Diagram

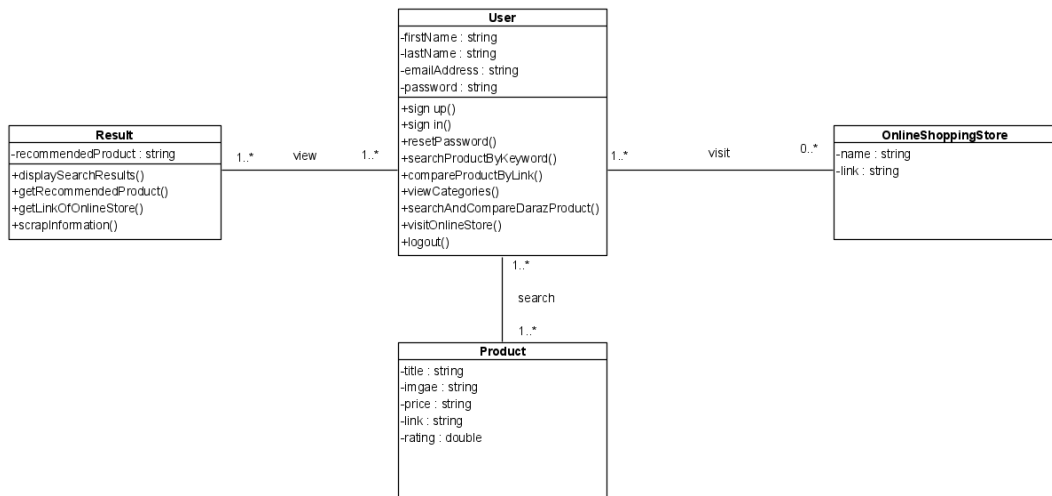


Figure 4.3: Class Diagram

Explanation

User class contains some properties that are required for completion of their profile. User can perform multiple operations that are defined in User class. User can create account which will execute `signUp()` operation, user can login to the portal which will execute `login()` operation and User can also `logout()`, user can reset his password if he forgets his existing password which will execute `resetPassword()`.

User can search the product of their need by typing the keyword this will execute the `searchProductByKeyword()` operation. The user will also be provided with an option to compare the product of two stores (Daraz and iShopping) by providing the links to the products, this will execute the `compareProductByLink()` operation. The user would be able to view the assigned categories (mobile phones, LED Televisions, air conditioners, Refrigerator) by executing `viewCategory()` operation. If the user wants to compare two of Daraz products then `searchAndComapreDarazProduct()` operation would execute. If the user will also be provided with the products provider website link, thus if he wants to pay a visit then the `visitOnlineStore()` operation would execute. Lastly the user can logout the system by executing the `logout()` operation.

The user can search multiple products, the product class consists of attribute title, image, price, link and rating. This all information would be saved on the database.

The user can view the result. The search result will be shown to the user using `displaySearchResult()` operation, the class would also display the recommended product and this will execute the `getRecommendedProduct()`. When the user is provided by his search results, the links of the particular store would also be provided to the user by executing the `getLinkOfOnlineStore()`. The information about the searched product would be scrapped from the website by executing the `scrapInformation()` operation.

4.5. Dynamic View

4.5.1. Sequence Diagram

4.5.1.1. UC -01: Sign Up

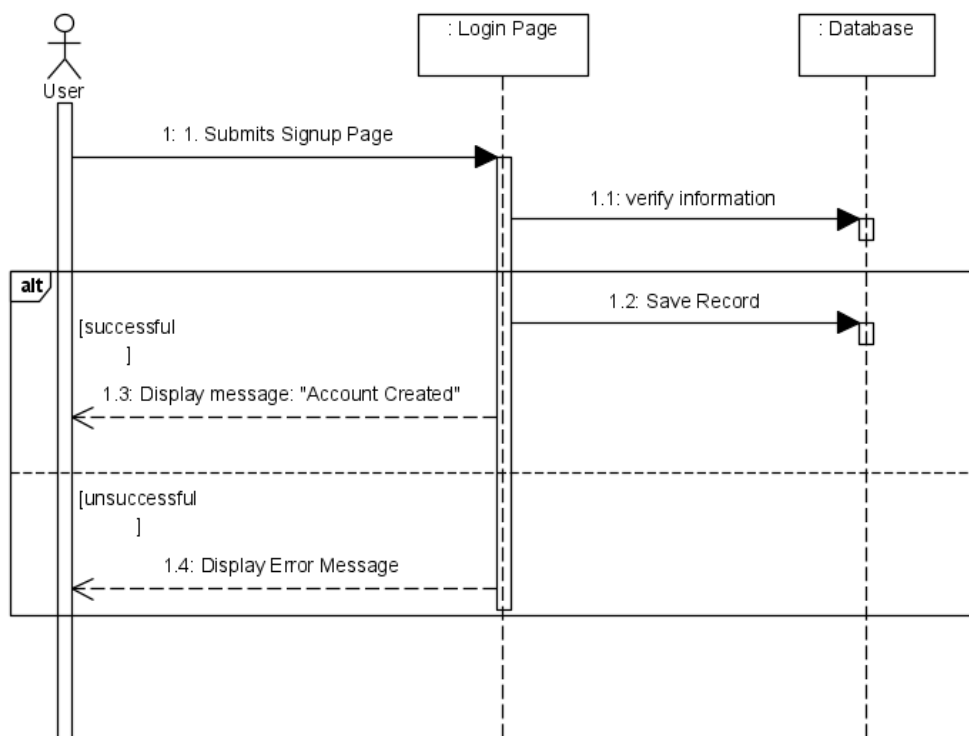


Figure 4.4: Sequence Diagram - Sign Up

Explanation

The user submits the signup form after filling it. The application verifies the information from the database.

If all required fields are filled, email is unique and passwords match then successful sequence will be executed, user record will be saved in the database and “Account is Created” message will be displayed to the user.

If user has left required fields empty, then “Fill required fields!” error prompt will be displayed. If user has entered the email address that is already registered, then “Email already exists!” error prompt will be displayed.

4.5.1.2. UC -02: Sign In

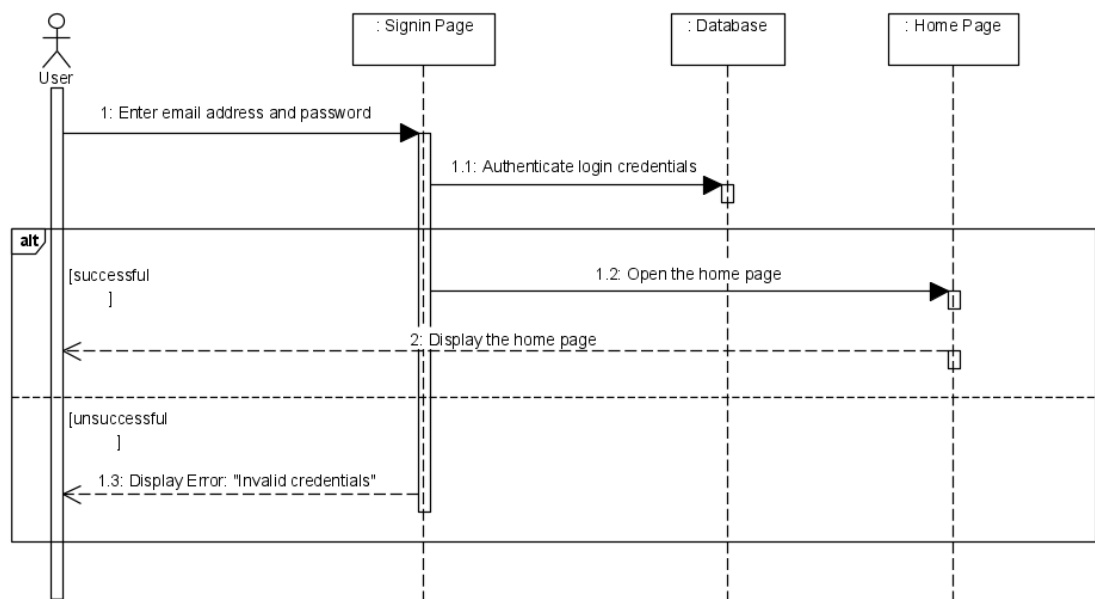


Figure 4.5: Sequence Diagram - Sign In

Explanation

The user submits the login form to logged in to the home page. The email address and password entered by the user will be authenticated from database.

- If user entered valid credentials, then Valid User sequence will be executed, and home page will be displayed.
- If user entered invalid login credentials, then error will be prompted that is “Invalid login credentials. Please try again.”

4.5.1.3. UC -03: Search and compare by Keyword

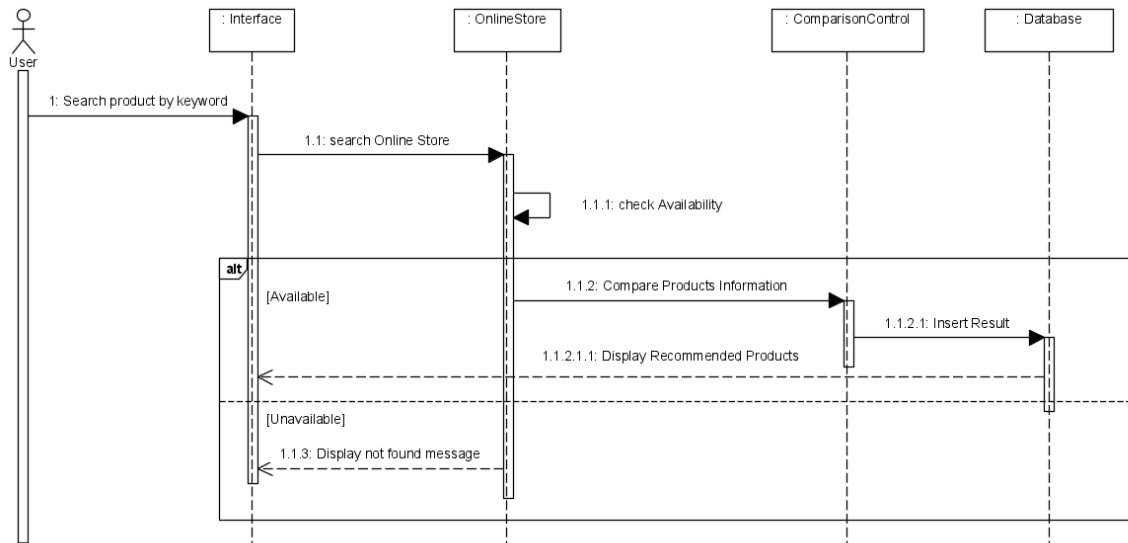


Figure 4.6: Sequence Diagram - Search and Compare by Keyword

Explanation

The user will enter the products keyword, the system will search on the online stores for the product, check its availability.

- If the product is available on the website, the products information is compared with the product information of another store. The results are stored in the database. The recommended product is displayed on the interface.
- If the product is unavailable, not found message is displayed.

4.5.1.4. UC -04: Compare Products by Links

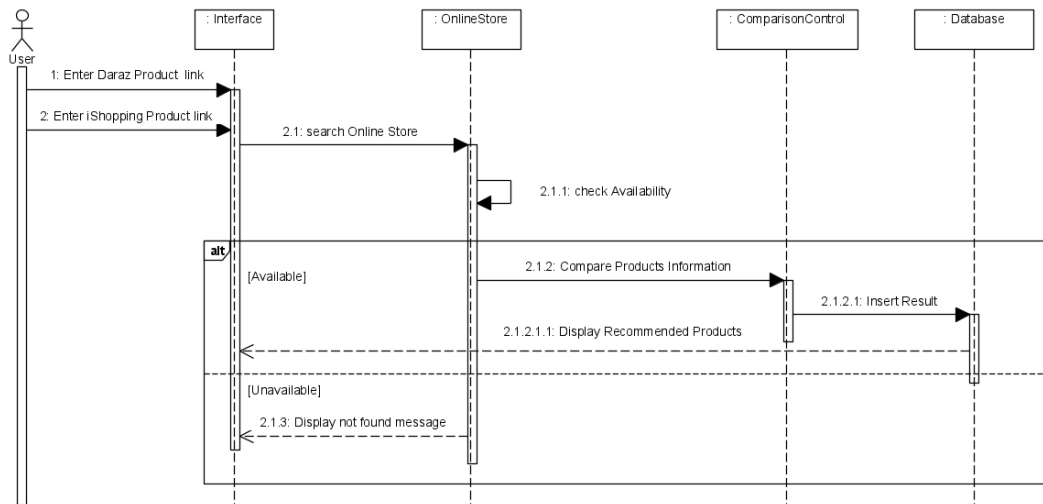


Figure 4.7: Sequence Diagram - Compare Products by Links

Explanation

The user enters the products link one of Daraz and other of iShopping, the system checks their availability on the stores.

- If the product is available on the website, the Daraz products information is compared with the iShopping product information. The results are stored in the database. The recommended product is displayed on the interface.
- If the product is unavailable, not found message is displayed.

4.5.1.5. UC -05: View Category

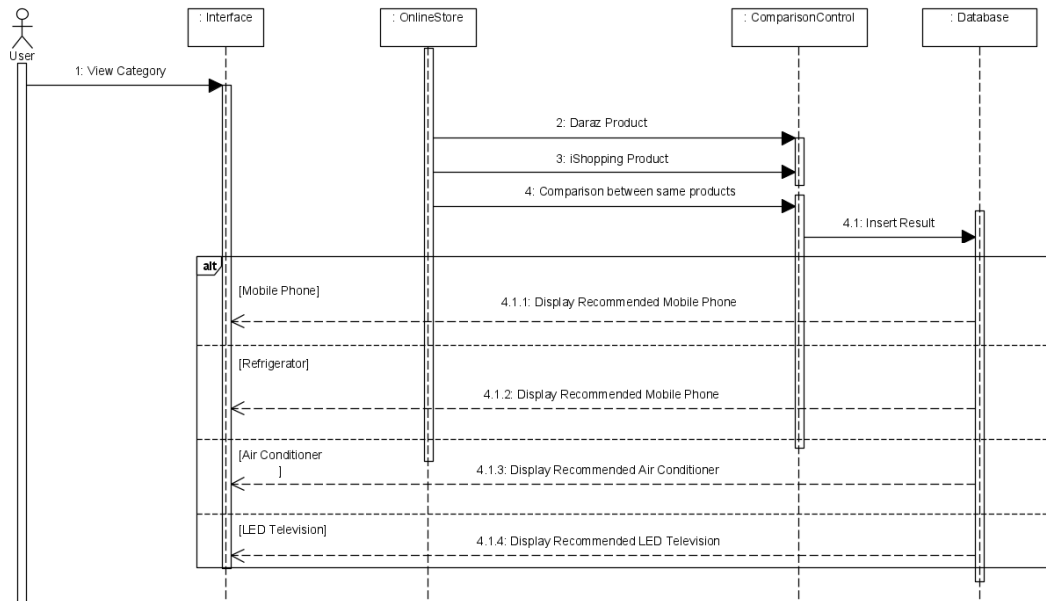


Figure 4.8: Sequence Diagram - View Category

Explanation

The user clicks on the view category option on the interface provided, the system will scrape the information of two same products available on Daraz and iShopping. Then compare their results and insert them in the database.

- If the user clicks on the mobile phone category, then the recommended mobile phone will be displayed.
- If the user clicks on the refrigerator category, then the recommended refrigerator will be displayed.
- If the user clicks on the air conditioner category, then the recommended air conditioner will be displayed.
- If the user clicks on the LED Television category, then the recommended LED Television will be displayed.

4.5.1.6. UC -06: Compare Daraz Products

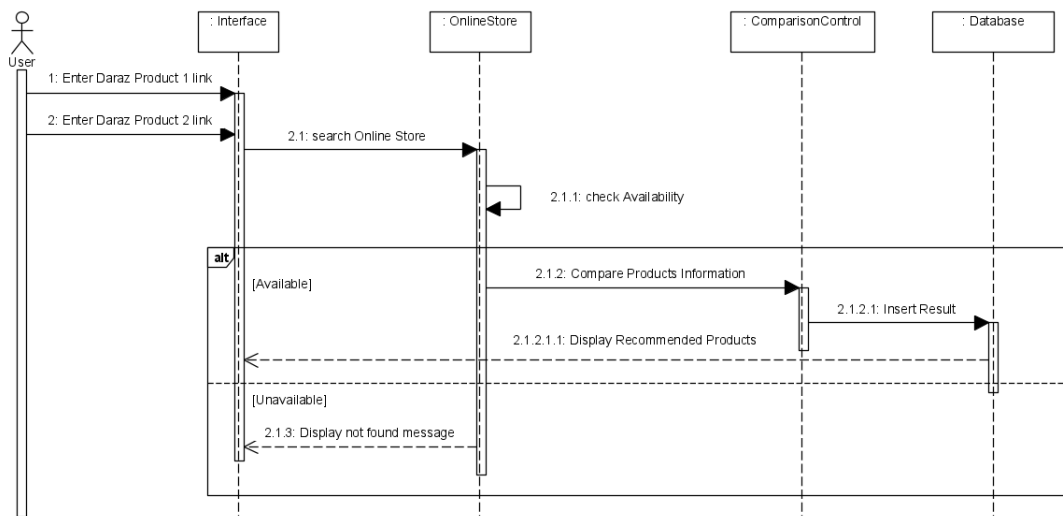


Figure 4.9: Sequence Diagram - Compare Daraz Products

Explanation

The user enters the products link two of Daraz products and, the system checks their availability on the stores.

- If the product is available on the website, then both Daraz products information is compared. The results are stored in the database. The recommended product is displayed on the interface.
- If the product is unavailable, not found message is displayed.

4.5.1.7. UC -07: View Product

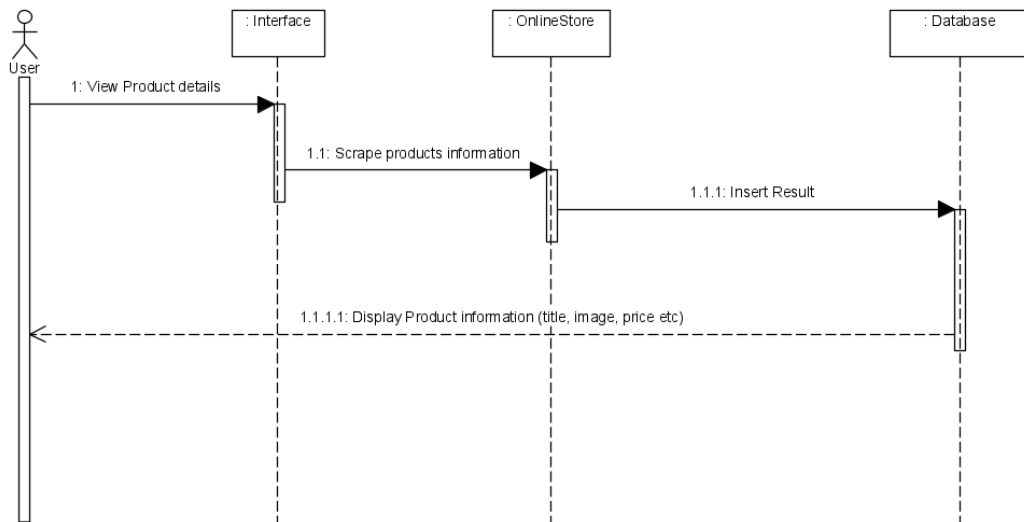


Figure 4.10: Sequence Diagram - View Product

Explanation

When the user wants to view the products details, the system will scrape the products information from the respective website. The information included would be title, image link, respective page link, price and ratings, this will be stored in the database at the time of the search and then displayed to the user.

4.5.1.8. UC -08: Select Product

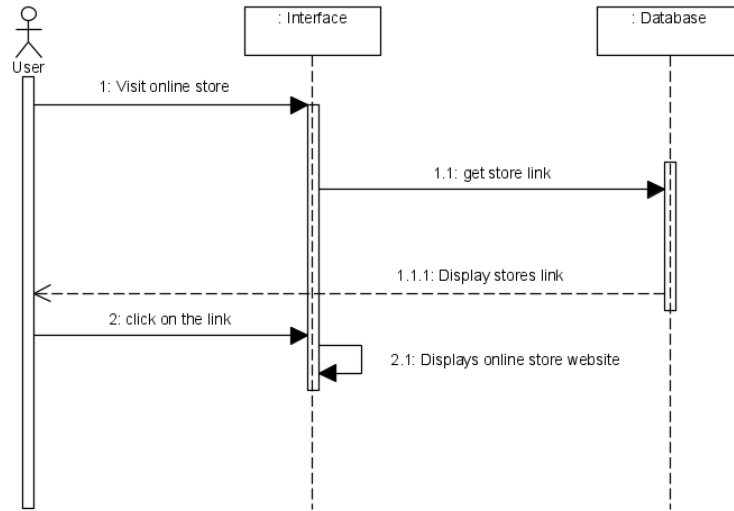


Figure 4.11: Sequence Diagram - Select Product

Explanation

When the user desires to visit the online store of the recommended product, the system will fetch the providers store link and present it to the user.

If the user clicks on the link, the system will take the user to the providers website, where the user might purchase the product.

4.5.1.9. UC -09: logout

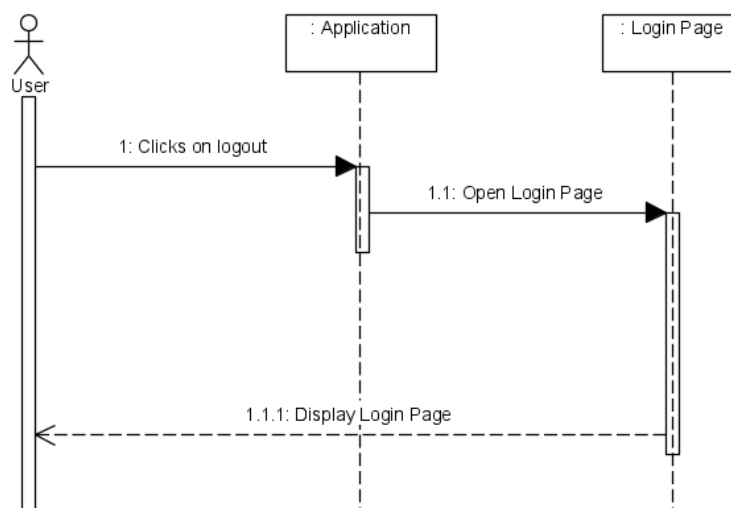


Figure 4.12: Sequence Diagram - Logout

Explanation

User clicks on Logout button at each page navbar. It will logout user from account and will display the Login Page.

4.5.2. State Machine Diagram

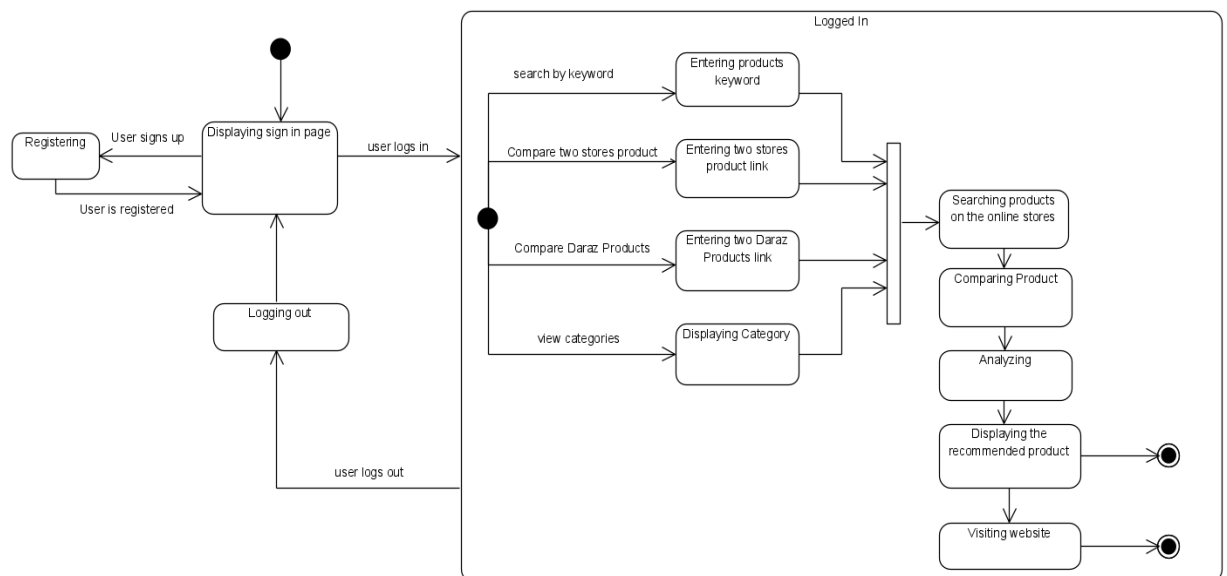


Figure 4.13: State Transition Diagram

Explanation

As the user performs various actions while using the system, the system makes transitions to various states depending upon the user's actions.

First the user is provided with a sign in page, while on the sign in, the user can Sign up if he/she is not registered, Login to the system if the user has already signed up.

If the user is not registered and is Signing Up, the system gets into Registering state because a new user is being added into the database.

If the user logs into their account, they can search a product by its keyword, compare products of Daraz and iShopping stores, view categories or compare Daraz products. All these states are substates of the logged in state because the user performs all these actions while being logged in.

Any state the user opts for, they are joined into the same state of searching the products on the online stores, followed by comparing the products on the basis of reviews, ratings and price. Then analyzing which is better than the other and then finally displaying the recommended product. After this it is up to the user whether he wants to exit or visit the website of the particular online store and then exit.

The system provides the opportunity of logging out of the account at any time after logging in. Once the logging out state is successfully executed, the system displays the log in page again.

4.5.3. Activity Diagram

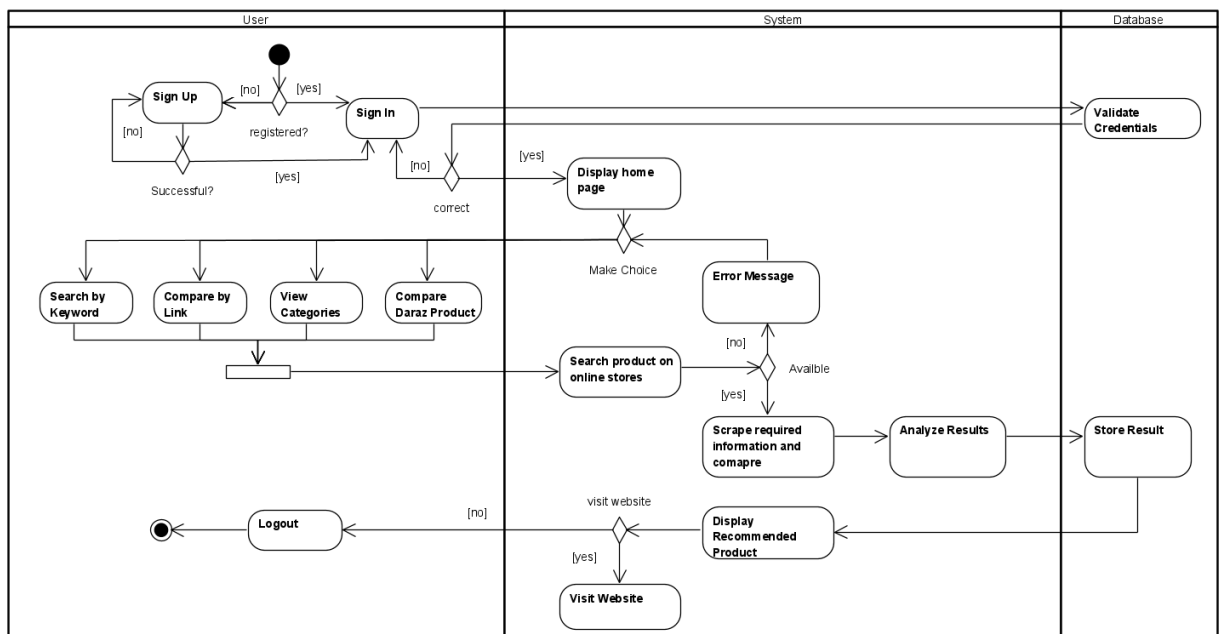


Figure 4.14: Activity Diagram

Explanation

The activity diagram is made by using Swimlanes. Swimlane simply describes who is responsible for the activities being performed in the activity diagram and how they are responsible. The activity diagram only represents the activities being performed, but the Swimlane describes who does what in a process or activity performed.

In the diagram there are three Swimlanes: User, System, and Database. The User performs the activities visible to them on the screen. The User either signs up or logs in to the system. If user does not have an account, then he is required to signs up,

if the user has already signed up then they can login their account. While logging in, the credentials of the user are validated by the Database. If the credentials are valid then the User is logged in to their account, however if not, the login error is displayed, and the User enters their credentials again.

When the user logs in to their account, a home page is shown to them, where the user has to make a choice among four options either he can search by keyword, or compare products by link, or view categories or compare two Daraz products. All these activities are joined to the activity search products on the respective stores. If the product is unavailable, an error message is prompted and the user remains on the same page, but if the product is available, the system scraps required information from the respective stores, analyze their results. These results are stores in the database. And then the system retrieves the result from the database and display t he recommended product. Now the user makes a decision whether, they want to visit the webstore or not, if not, then the user logouts, if the user wants to visit, he is taken to the desired web store.

4.6. Component Design

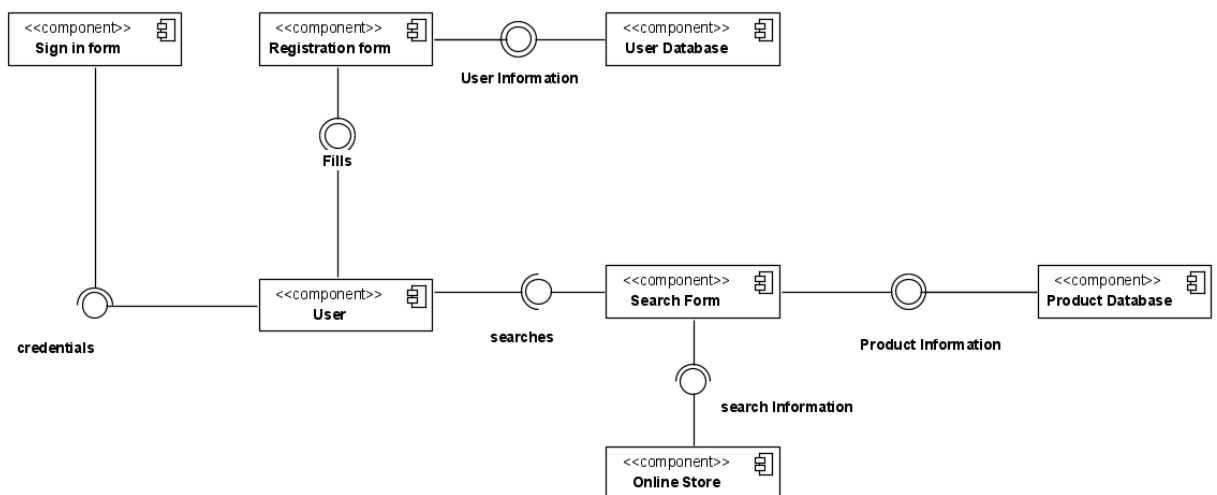


Figure 4.15: Component Diagram

Explanation

Component Diagram provides static view of the system and relationship between different components of the system.

User component needs to provide User Information at the time of registration that will be stored in User Database component. User component credentials will be need for the sign in form component.

The user searches will be stored in the search form component. As stated earlier in the thesis, there are four ways to search and compare for a product, they all are combined as the search form component. The search information is looked in the online store component. The search form component products information will be stored in the Product Database component.

4.7. Deployment Diagram

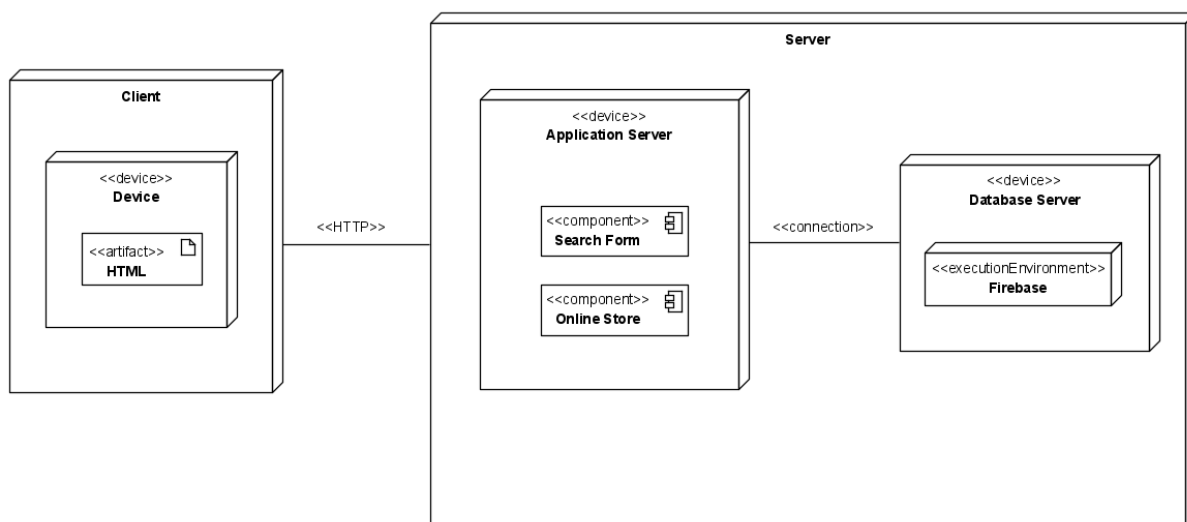


Figure 4.16: Deployment Diagram

Explanation

Deployment diagram show how the hardware and software components run in database/web application and how are connected the different pieces with one another. It contains nodes and relationships of nodes.

A user can use any device (desktop, laptop or smartphone) and can request for the ShopSmart application. The application is based on the client – server architecture that is already explained in the Software Architecture section.

4.8. Data Models

4.8.1. ER Diagram

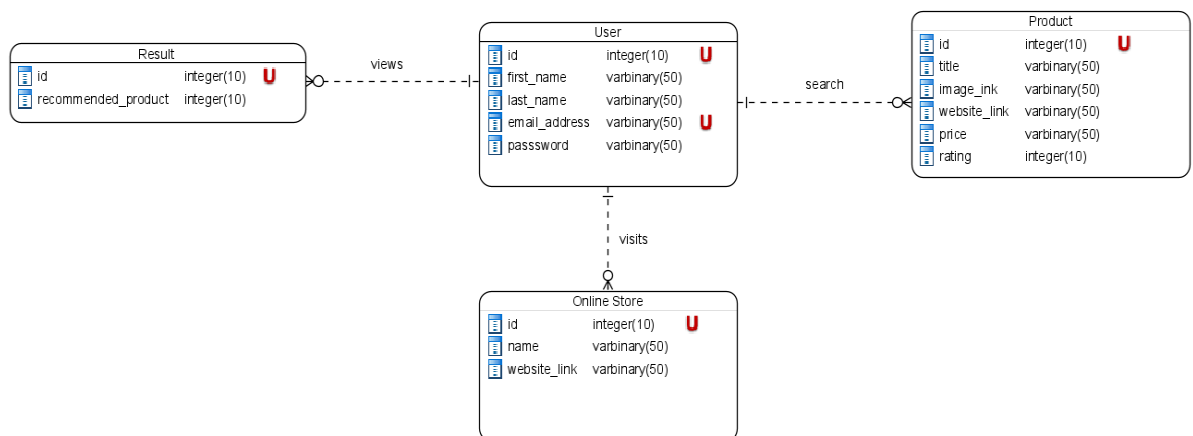


Figure 4.17: Deployment Diagram

Explanation

An entity relationship diagram gives a picture of how the entities relate to each other. It is the blueprint that underpins your business architecture, offering a visual representation of the relationships between different sets of data (entities).

The user can view one to many results and also can visit the online stores and search for one-to-many products.

4.9. User Interface Design

The ShopSmart application consists of the following pages:

- Homepage
- Sign Up
- Sign In
- Reset Password
- Search and compare by keyword
- Compare Products by Links
- Categories
- Compare Daraz Products

4.9.1. Low Fidelity Prototypes - Web Wireframes

Low fidelity wireframes were created, they are static nature means the user cannot interact with them and are just used to get the basic design idea of the user interface. we have created the following pages:



Figure 4.18: Low fidelity - Sign Up

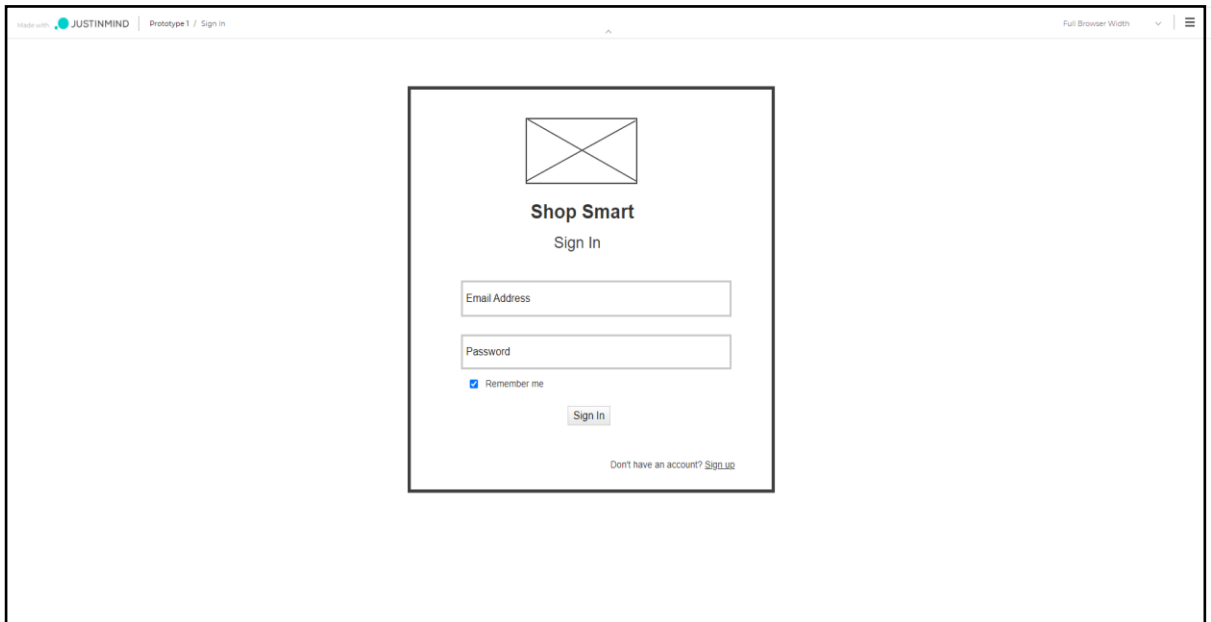


Figure 4.19: Low fidelity - Sign In

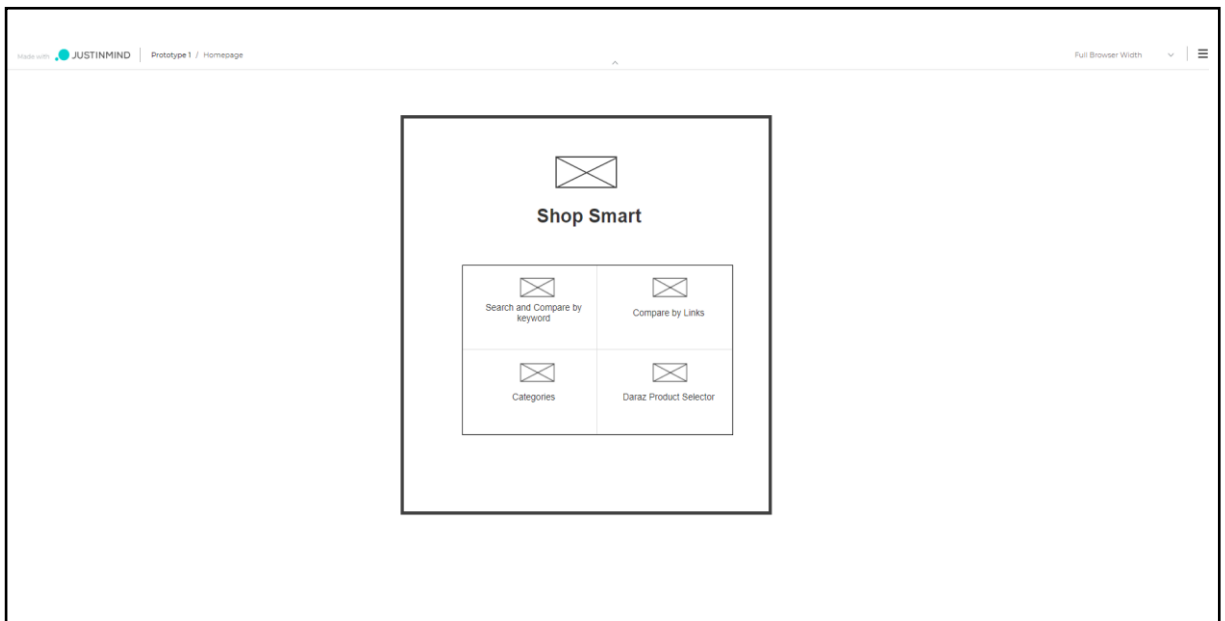


Figure 4.20: Low fidelity – Homepage

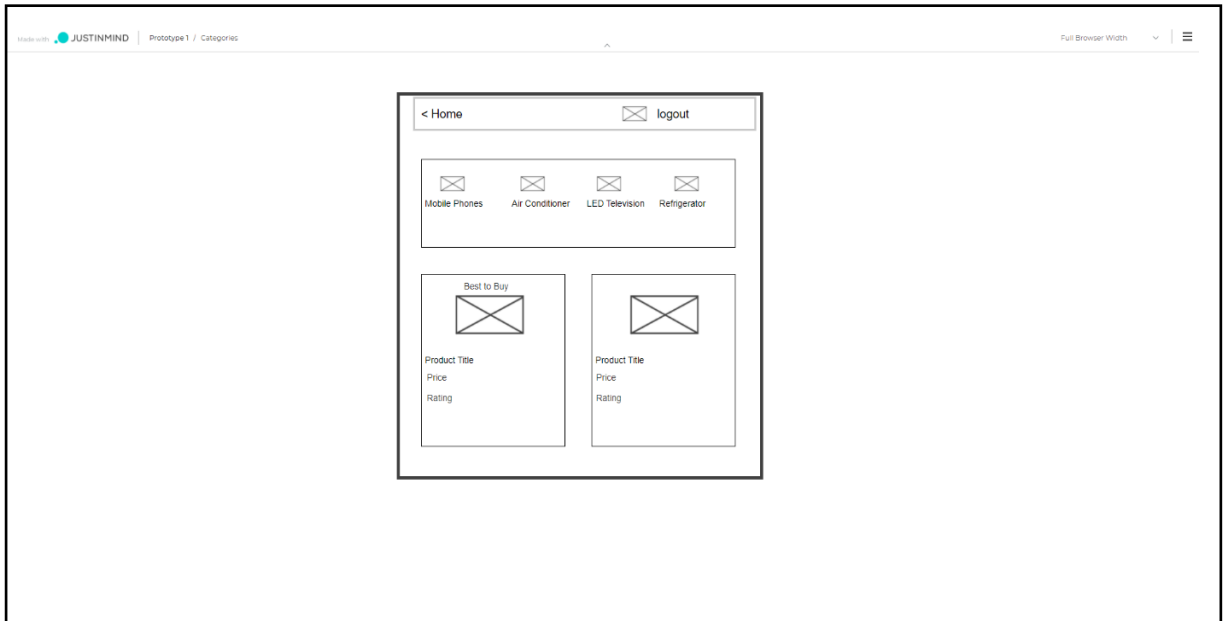


Figure 4.21: Low fidelity – Categories

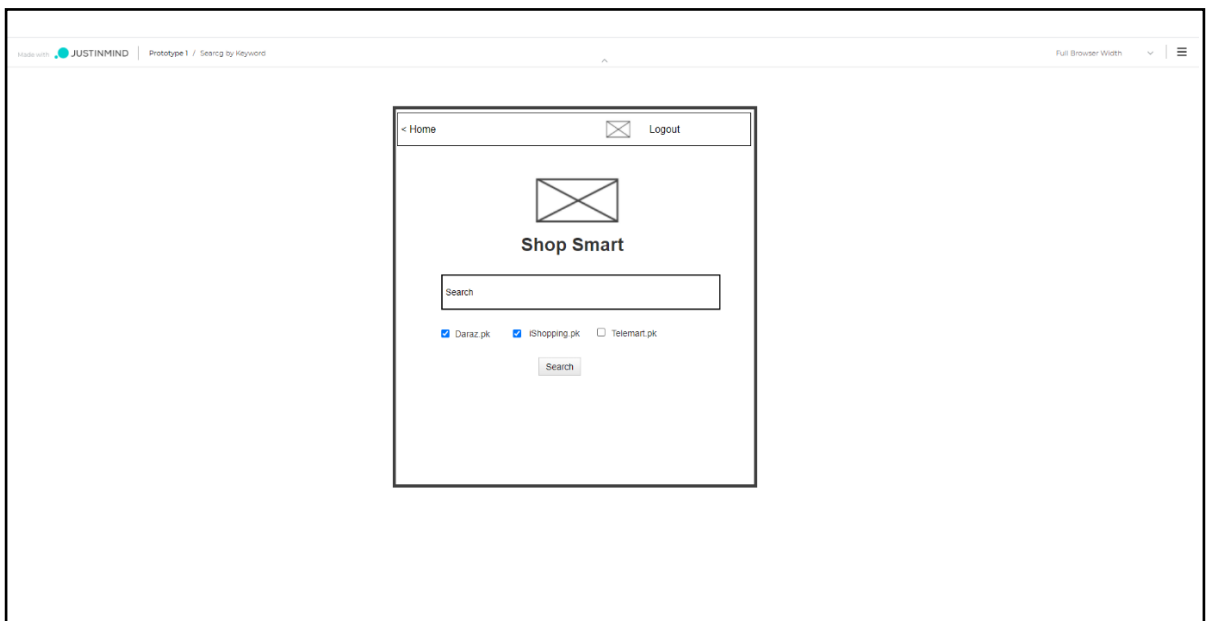


Figure 4.22: Low fidelity – Search and Compare by Keyword

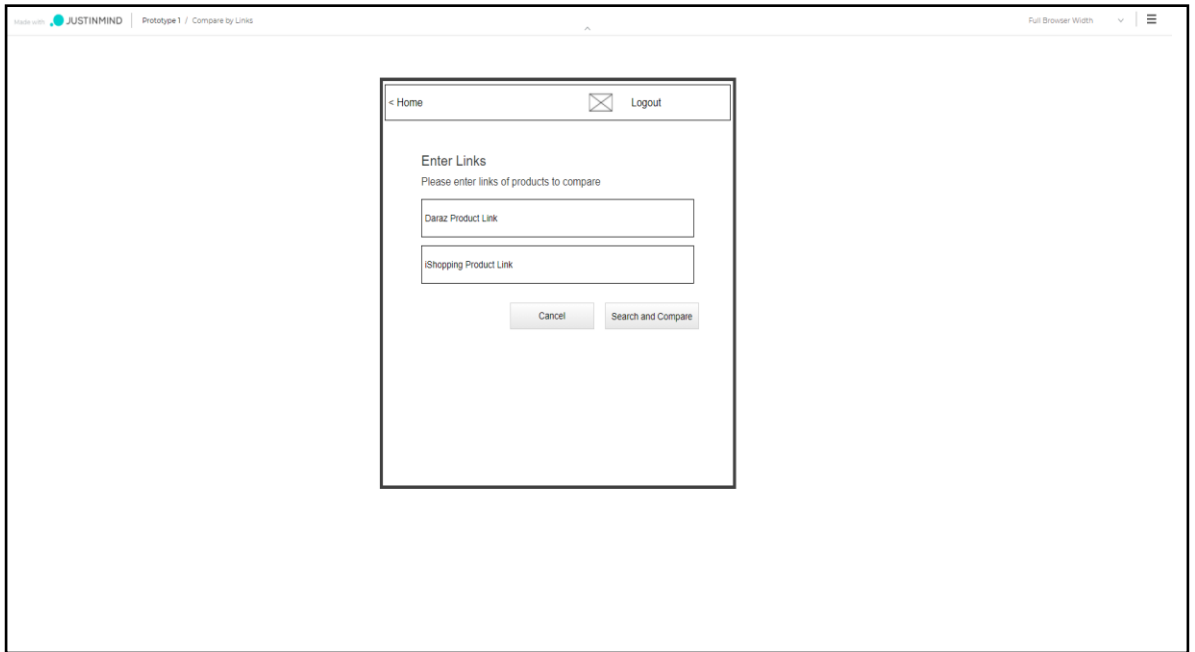


Figure 4.23: Low fidelity – Compare by Links

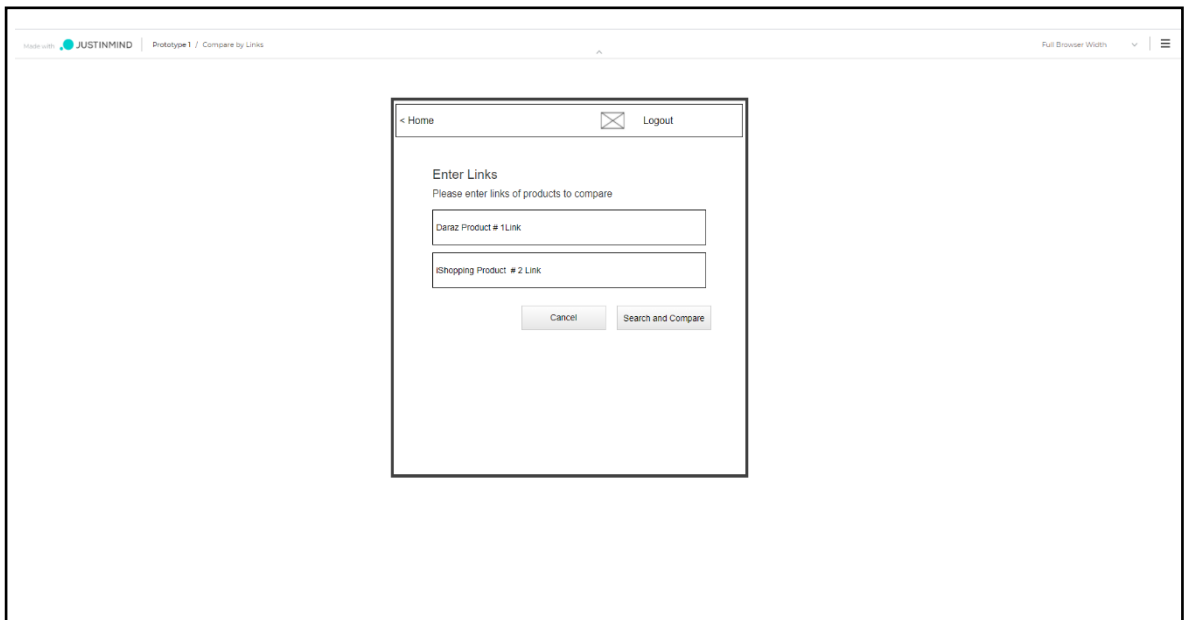


Figure 4.24: Low fidelity – Daraz Product Selector



Figure 4.25: Low fidelity - Searching and Comparing

4.10. System Prototype

4.10.1. High Fidelity Prototype

The following figures show the final interface of the application. The description of each interface is already stated in the user interface section.

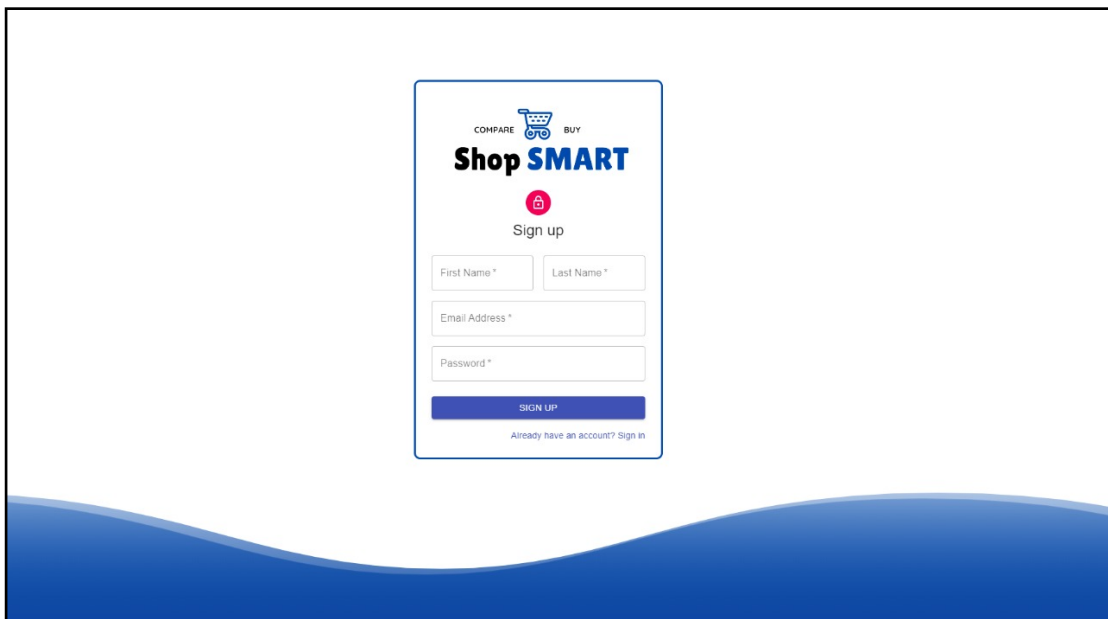


Figure 4.26: High fidelity - Sign up

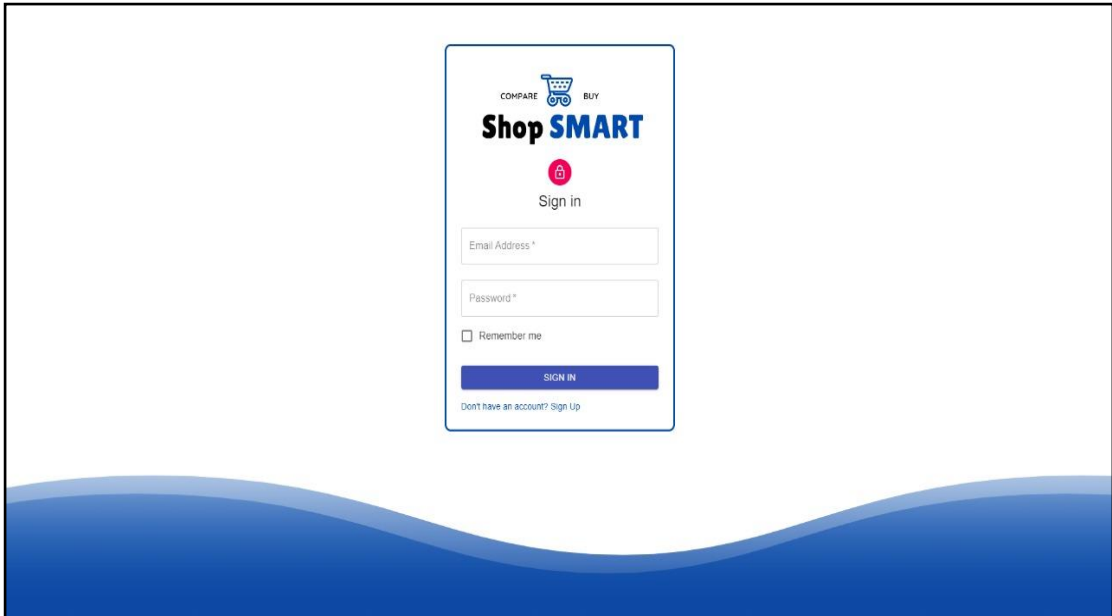


Figure 4.27: High fidelity - Sign In

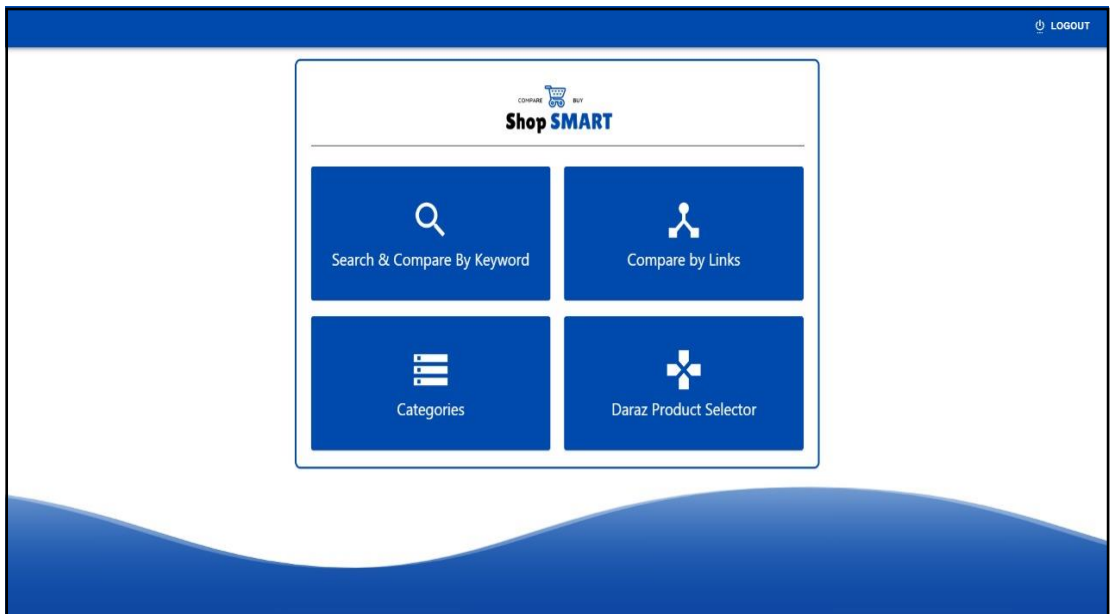


Figure 4.28: High fidelity - Homepage

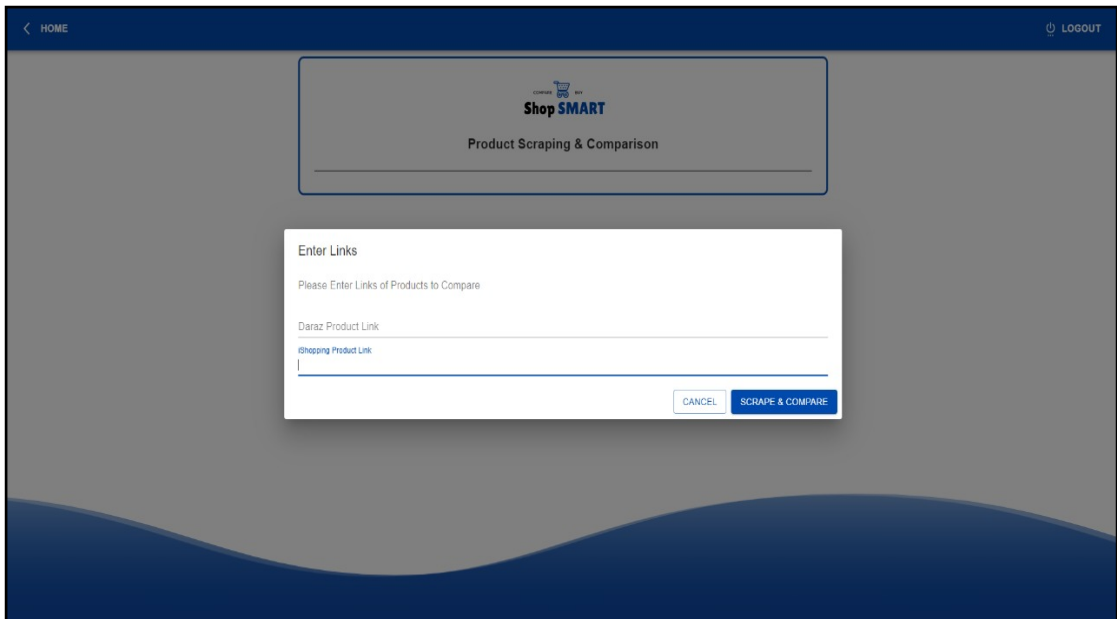


Figure 4.29: High fidelity – Compare Products

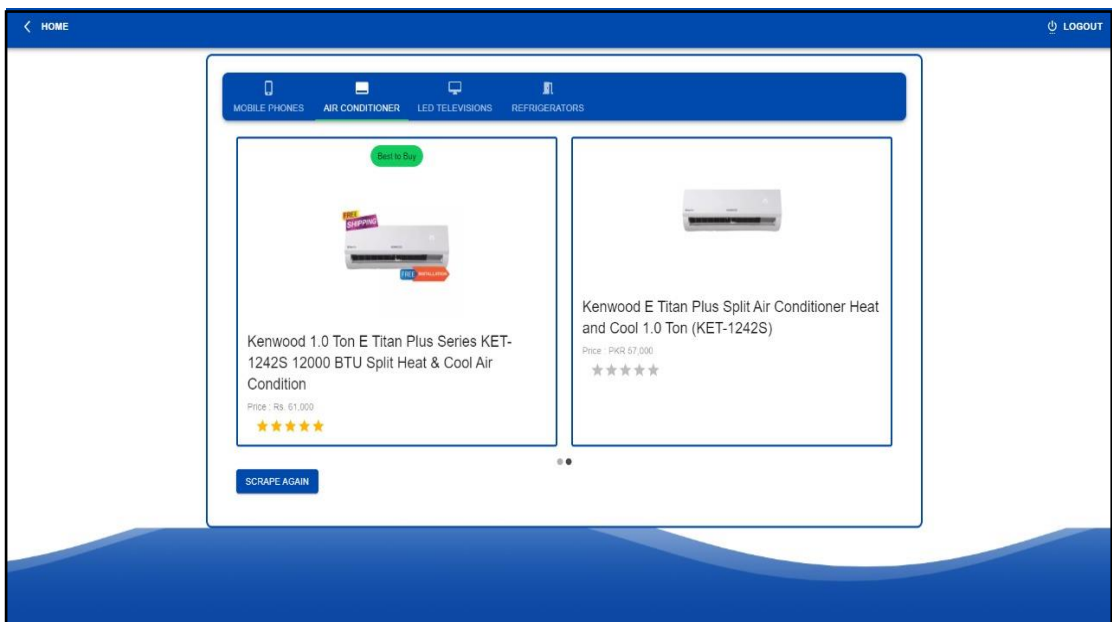


Figure 4.30: High fidelity - Categories

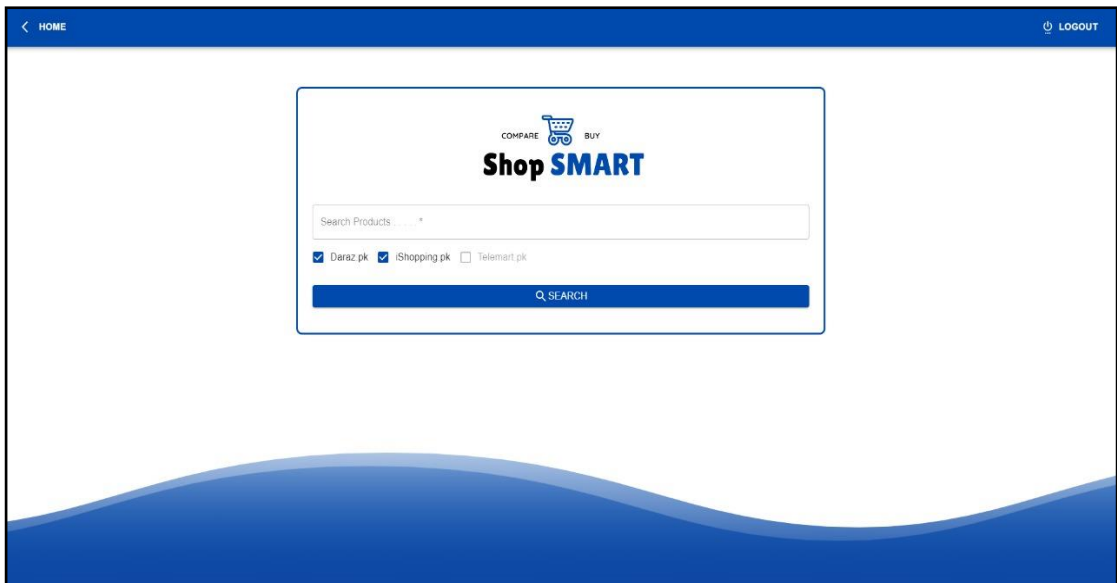


Figure 4.31: High fidelity – Search and Compare by Keyword

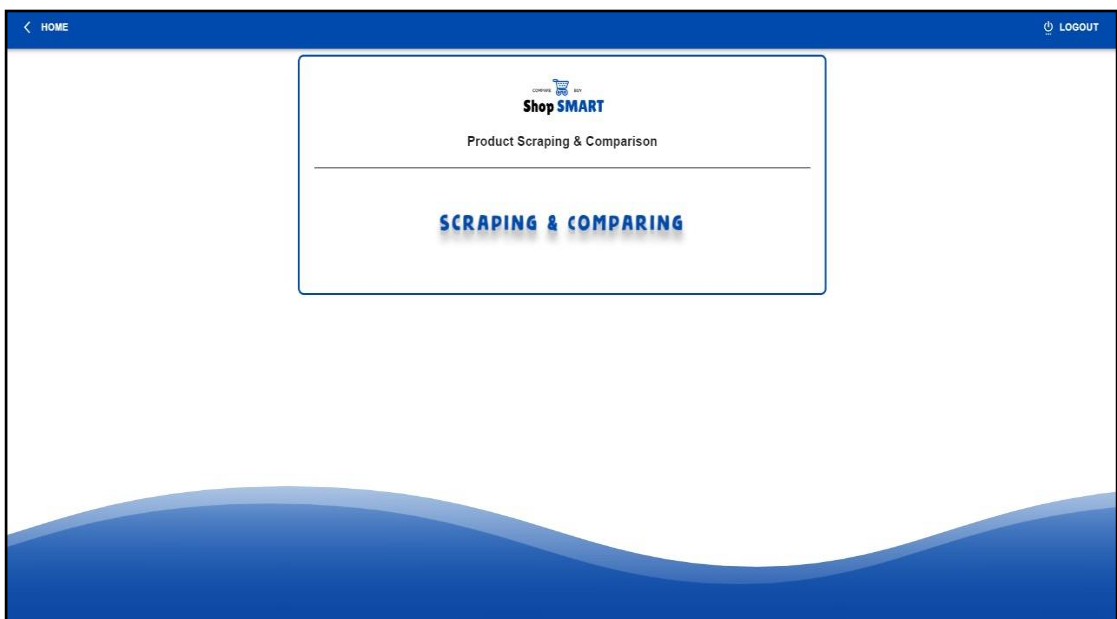


Figure 4.32: High fidelity – Scraping and Comparing Information

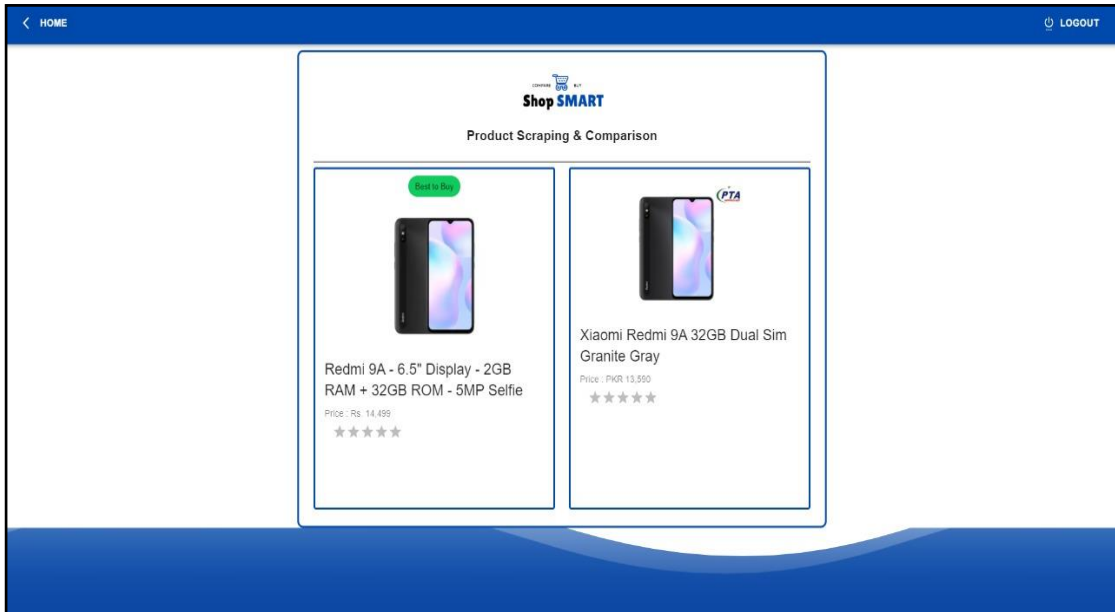


Figure 4.33: High fidelity - Best to Buy

4.11. Conclusion

This chapter focuses on the design of the system. It includes all the pictorial representations of our project, the design strategies, and design models. The chapter includes the system architecture, logical design such as domain model and class diagram, dynamic view such as the sequence diagrams, state transition diagrams and activity diagrams. The screenshots of high-fidelity prototype are also included in the chapter.

Chapter 5

System Implementation

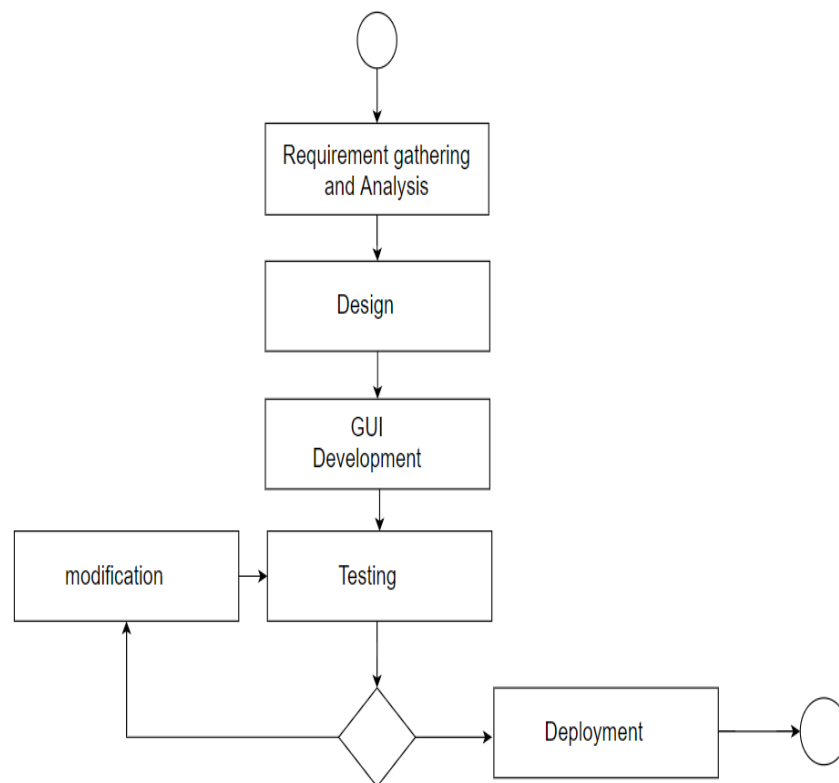
5.1. Chapter Overview

This chapter will cover all the implementation strategies, the tools, technologies and methodologies used for the implementation of ShopSmart,

5.2. Strategy

The basic strategy that we used for our project is to integrate all the components being developed independently. Our application was divided into components or modules based on different user choice. We have developed and tested each module individually and we then integrated them with the actual system in order to get required and expected results. This methodology led us to understand and develop each module individually as well as efficiently with least chances of possible errors.

Figure 5.1: Strategy Flow



5.3. Tools Used

5.3.1. VS Code

VS Code is used as a code editor for the development of our project. It is a freeware source code editor which provides various features for debugging code, syntax highlighting, various IntelliSense extensions for smart code completion, code refactoring and showing problems in the code.

A variety of languages are supported by visual studio code including Java, JavaScript, Go, Node.js, Python, and C++. Visual studio code is based on the Electron framework, which is used to develop Node.js Web applications. ShopSmart applications front end is developed in React.js, thus we used this platform for the debugging of our application.

5.3.2. Firebase

Firebase is Google's application development platform that helps to build, improve and grow applications. Firebase can act as the server, API, and datastore. The purpose of using firebase in our project is for authentication purposes and storing the result of the scraped product.

It provides backend services, easy-to-use SDKs, and readymade UI libraries to authenticate users. Firebase authenticates the user with their email and passwords. The firebase authentication Software Development Kit (SDK) provides methods to create and manage users that use their email addresses and passwords to sign in. it also handles that sending the password reset emails.

It is useful to connect the application's existing sign-in system to the firebase authentication SDK and gain access to Firebase Realtime Database and other firebase services. In the real time Database, the products image link, link of the website, price, ratings and title is stored.

5.3.3. NMP

NPM is the world's largest software registry. Open-source developers use npm to share and borrow packages, and many organizations use npm to manage private

development as well. NPM – or "Node Package Manager" – is the default package manager for JavaScript's runtime Node.js. NPM consists of two main parts: A CLI (command-line interface) tool for publishing and downloading packages, and an online repository that hosts JavaScript packages. Every project in JavaScript whether it's Node.js or a browser application – can be scoped as a npm package with its own package information. As ShopSmart is based on React and Node.js thus to install the libraries we used npm in our project.

5.3.4. Selenium

Selenium WebDriver is a collection of open-source APIs which are used to automate the testing of a web application. WebDriver is a browser automation framework that works with open-source APIs. The framework operates by accepting commands, sending those commands to a browser, and interacting with applications. It supports many browsers such as Firefox, Chrome, IE, and Safari. Selenium Web driver is platform-independent since the same code can be used on different Operating Systems like Microsoft Windows, Apple OS and Linux. Selenium supports most code scripts such as Java, C#, PHP, Python, Ruby, Perl, and Groovy. In our project, it used for automation.

5.4. Technologies

5.4.1. Frontend Technologies

The coding languages, frameworks and libraries that were used for the development of frontend of ShopSmart include HTML, CSS, Bootstrap, JavaScript and React.js.

1. HTML (Hypertext Markup Language) is the coding language that is used to structure a web page and its contents.
2. CSS (Cascading Style Sheet) is the coding language used for designing the web pages like giving different colors to different HTML elements, fonts, layouts, and responsiveness of different elements according to the dimensions of different devices.
3. Bootstrap is the framework that was used to design various elements of web pages faster and easier with the help of HTML and CSS based templates like

forms, navbars, buttons etc. It can also be used to set the responsiveness of various elements of web pages for various devices.

4. JavaScript is the coding language used to make web pages dynamic and interactive.
5. React.js is an open-source JavaScript library that is used for building user interfaces specifically for single-page applications. It's used for handling the view layer for web and mobile apps. React also allows us to create reusable UI components.

5.4.2. Backend Technologies

The backend technologies that were used for the development of the project are Python, Node.js and Firebase.

1. Python is the programming language used to write the scrapper code.
2. Node.js was used to interact the front end written in React.js and the backend written in Python. The frame work of Node.js which was used is express.js.
3. Firebase is used as the database of the ShopSmart to store the users' credentials and the scrapped product information.

5.5. Methodology

5.5.1. Iterative Development Approach

ShopSmart was developed using iterative approach. It is the approach in which whole project is developed in repeated cycles.

Unlike traditional waterfall model which follows linear sequential approach to complete different phases of the development of the project. In iterative approach, the initial phase is the gathering of requirements in which we have planned, gathered, and written all major requirements of the project. Although, some of the new functionalities were identified later which enhanced the requirements of the project. After initial phase of requirements, we have designed, implemented, and tested our project in several iterations. In each iteration, a new feature was added, or existing feature was improved until all stated requirements of the project were implemented.

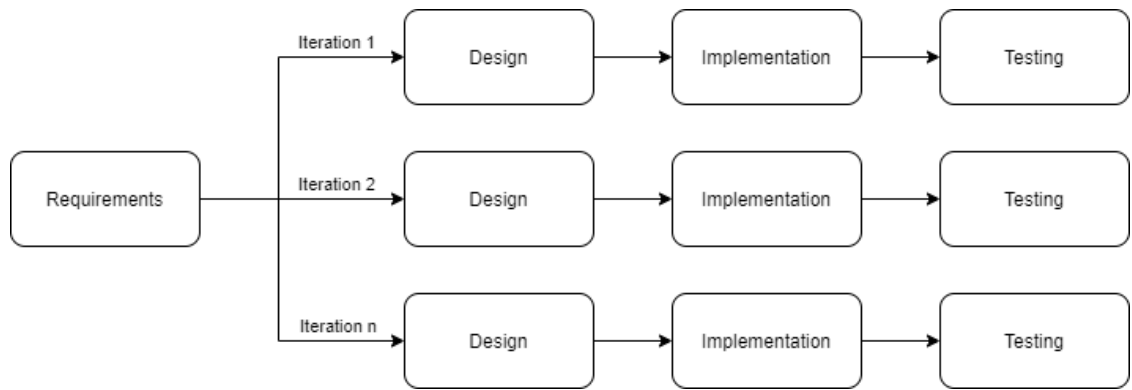


Figure 5.2: Iterative Development Approach

5.6. Conclusion

In this chapter we discussed all the implementation details, strategy, the tools, techniques, and methodology used to implement the project. The iterative development approach was used for different frontend and backend technologies.

Chapter 6

System Testing & Evaluation

6.1. Chapter Overview

This chapter contains all the test strategies applied for the testing of ShopSmart application such as component testing, unit testing, integration testing and system testing. This chapter also covers the test cases made for the use cases.

6.2. Test Strategy

We used Bottom-Up integration technique for developing our application, each module was created separately and then integrated all of these modules into a single application. So, we tested each of the modules separately at each point in the development process and in the end, we tested the entire application as a whole to make sure that it has been integrated properly.

We have used requirements based testing and manual testing as our test strategy. The test cases are derived from the functional requirements and then executed manually to check if those requirements are met. The status of each test was recorded once the test was executed either it is a pass or a failure. The tests that were not successful were fixed and then the status of those test cases was updated. We made sure that our system fulfils the functional and non-functional requirements.

6.3. Component Testing

Component testing is the testing of each module or component. When we test each of the modules and components separately it becomes very easy to locate errors and faults. Component testing is black box testing, in which the tester does not know the depth of implementation and tests the system using some form of inputs to get the expected output.

Our application was divided into different modules. After each of the modules were created, we tested them to ensure that they are working properly before integrating them into our application.

6.4. Unit Testing

Testing a small testable part of a component is known as Unit Testing. This type of testing is very important because it is very easy to remove or repair errors at such a low level. It is white box testing because the tester has an access to the source code of the project.

At the beginning of the project, we faced a lot of issues and errors were generated but we performed a unit test on each of the smaller part and that helped in catching and removing these errors.

6.5. Integrated Testing

Once the unit testing has been done on the smaller parts, they are compiled into a single module on which integration testing is performed to make sure that the system works properly after it has been integrated. We tried our very best to ensure that each of the modules were properly tested so that not errors remained.

6.6. System Testing

System testing is performed after the system is completely developed. It is followed by the integration testing. After the integration testing a lot of testing was already done on the system as a whole to ensure that all of the modules worked properly as one and that the application would not crash at any point while it is being used.

6.7. Test Cases

For each use case in the use case diagram, a test case is written. Some of the test cases are of positive intent and some are written with negative intent. All the test cases have been passed in the testing.

6.7.1. Test Case#1 - Sign Up

Use Case	UC-01				
Use Case Name	Sign Up				
Test Priority	High				
Precondition	User is not registered				
Test Cases					
Test ID	Test Steps	Input Data	Expected Output	Actual Output	Status
TC - 01	<ol style="list-style-type: none"> 1. Click on Sign-Up page. 2. Click on Sign-Up button on Sign Up page 	Click on Sign-Up button	Fill in the required fields	Please fill in this field	Pass
TC - 02	<ol style="list-style-type: none"> 1. Click on Sign-Up 2. Enter First name, Last Name, Email Address, Password. 3. Click on Sign-Up button 	First name- Maira Last name - Babar Email Address- mairababar@gmail.com Password- maira123	Account Created. Redirecting to Login in 3 seconds.	Account Created. Redirecting to Login in 3 seconds.	Pass
TC - 03	<ol style="list-style-type: none"> 1. Click on Sign-Up 2. Enter First name, Email Address, Password. 3. Does not enter last name. 4. Click on Sign-Up button 	First name- Maira Last name - Email Address- mairababar@gmail.com Password- maira123	Error Message Generated	Error Message Generated	Pass
TC - 04	<ol style="list-style-type: none"> 1. Click on Sign-Up 2. Enter First name, Last Name, Email Address, Password. 3. Click on Sign-Up button 	First name- Maira Last name - Babar Email Address- mairababar@gmail.com Password- maira123	Account already exists.	Account already exists.	Pass

Table 6.1: Test Case # 1 - Sign Up

6.7.2. Test Case#2 - Sign In

Use Case	UC-02				
Use Case Name	Sign In				
Test Priority	High				
Precondition	User is registered and is on login page				
Test Cases					
Test ID	Test Steps	Input Data	Expected Output	Actual Output	Status
TC - 01	<ol style="list-style-type: none"> 1. Click on Sign-In page. 2. Click on Sign-In button on Sign In page 	Click on Sign-In button	Email is required	Error! Email is required.	Pass
TC - 02	<ol style="list-style-type: none"> 1. Enter email address 2. Enter password 3. Clicks on login button. 	Email Address- mairababar@gmail.com Password- maira	Error: Password is invalid.	Error: Password is invalid.	Pass
TC - 03	<ol style="list-style-type: none"> 1. Enter email address 2. Enter password 3. Clicks on login button. 	Email Address- mairababar@gmail.com Password- maira123	Message: Logged-In Successfully	Message: Logged-In Successfully	Pass
TC - 04	<ol style="list-style-type: none"> 1. Enter email address 2. Clicks on login button. 	Email Address- mairababar@gmail.com Password-	Error: Password is invalid.	Error: Password is invalid.	Pass

Table 6.2: Test Case # 2 - Sign In

6.7.3. Test Case#3 - Search and Compare by Keyword

Use Case	UC-03				
Use Case Name	Search and Compare by Keyword				
Test Priority	Medium				
Precondition	User should be logged to their account.				
Test Cases					
Test ID	Test Steps	Input Data	Expected Output	Actual Output	Status
TC - 01	<ol style="list-style-type: none"> 1. Click on Search and Compare by keyword from main menu. 2. Enter keyword and select checkboxes 3. Click on search button 	Search Product – Air conditioner Checkbox – Daraz.pk and iShopping.pk	Display best to buy product, among the stores.	Display best to buy product, among the stores.	Pass
TC - 02	<ol style="list-style-type: none"> 1. Click on Search and Compare by keyword from main menu. 2. Select Checkboxes 3. Click on search button 	Search Product - Checkbox – Daraz.pk and iShopping.pk	Error: Invalid search	Error: Invalid search	Pass
TC - 03	<ol style="list-style-type: none"> 1. Click on Search and Compare by keyword from main menu. 2. Enter keyword 3. Click on search button 	Search Product – Air conditioner Checkbox –	Error: Invalid search	Error: Invalid search	Pass

Table 6.3: Test Case # 3 - Search and Compare by Keyword

6.7.4. Test Case#4 - Compare Products by Links

Use Case	UC-04				
Use Case Name	Compare Products by Links				
Test Priority	Medium				
Precondition	User should be logged to their account.				
Test Cases					
Test ID	Test Steps	Input Data	Expected Output	Actual Output	Status
TC - 01	<ol style="list-style-type: none"> 1. Click on Compare Products by Links from main menu. 2. Enter the Daraz product link 3. Enter iShopping product link 4. Click on Scrape and compare button. 	<p>Daraz Product Link – https://www.daraz.pk/products/changhong-ruba-sc-12jtw-10-ton-12000btus-wall-mount-split-air-conditioner-white-i107184555-s125548100.html?spm=a2a0e.searchlist.list.1.532777c2bXAw9H&search=1</p> <p>iShopping Product Link – https://www.ishopping.pk/pel-fit-chrome-inverter-air-conditioner-1-ton-prinvo-12k-aex1.html</p>	Display best to buy product, among the two stores.	Display best to buy product, among the two stores.	Pass
TC - 02	<ol style="list-style-type: none"> 1. Click on Compare Products by Links from main menu. 2. Click on cancel button 	None	Returns to the home page.	Returns to the home page.	Pass

TC - 03	<ol style="list-style-type: none"> 1. Click on Compare Products by Links from main menu. 2. Enter the Daraz product link 3. Click on Scrape and compare button. 	<p>Daraz Product Link – https://www.daraz.pk/products/changhong-ruba-sc-12jtw-10-ton-12000btus-wall-mount-split-air-conditioner-white-i107184555-s1255548100.html?spm=a2a0e.searchlist.list.1.532777c2bXAw9H&search=1</p> <p>iShopping Product Link –</p>	Error: Invalid Search	Error: Invalid Search	Pass
TC - 03	<ol style="list-style-type: none"> 1. Click on Compare Products by Links from main menu. 2. Enter incorrect Daraz product link 3. Enter incorrect iShopping product link 4. Click on Scrape and compare button. 	<p>Daraz Product Link – Air conditioner</p> <p>iShopping Product Link – 44</p>	Error: Invalid Search	Error: Invalid Search	Pass

Table 6.4: Test Case # 4 - Compare Products by Links

6.7.5. Test Case# 5 - View Categories

Use Case	UC-05				
Use Case Name	View Categories				
Test Priority	Medium				
Precondition	User should be logged to their account.				
Test Cases					
Test ID	Test Steps	Input Data	Expected Output	Actual Output	Status
TC - 01	<ol style="list-style-type: none"> 1. Click on Categories from the home page 	None	Display different categories	Display different categories	Pass

TC - 02	<ol style="list-style-type: none"> 1. Click on Categories from the home page 2. Clicks on mobile phones category 	None	Display best to buy mobile phone from iShopping and Daraz.	Display best to buy mobile phone from iShopping and Daraz.	Pass
TC - 03	<ol style="list-style-type: none"> 1. Click on Categories from the home page 2. Clicks on air conditioner category 	None	Display best to buy air conditioner from iShopping and Daraz.	Display best to buy air conditioner from iShopping and Daraz.	Pass
TC - 04	<ol style="list-style-type: none"> 1. Click on Categories from the home page 2. Clicks on LED Television category 	None	Display best to buy LED Television from iShopping and Daraz.	Display best to buy LED Television from iShopping and Daraz.	Pass
TC - 05	<ol style="list-style-type: none"> 1. Click on Categories from the home page. 2. Clicks on Refrigerator category 	None	Display best to buy Refrigerator from iShopping and Daraz.	Display best to buy Refrigerator from iShopping and Daraz.	Pass

Table 6.5: Test Case # 5 - View Categories

6.7.6. Test Case#6 - Compare Daraz Products

Use Case	UC-06				
Use Case Name	Compare Daraz Products				
Test Priority	Medium				
Precondition	User should be logged to their account.				
Test Cases					
Test ID	Test Steps	Input Data	Expected Output	Actual Output	Status
TC - 01	<ol style="list-style-type: none"> 1. Click on Compare Daraz Products from main menu. 2. Enter Daraz product 1 link 3. Enter Daraz product 2 link 4. Click on Scrape and compare button. 	<p>Daraz Product 1 Link – https://www.daraz.pk/products/changhong-ruba-sc-12jw-10-ton-12000btus-wall-mount-split-air-conditioner-white-i107184555-s1255548100.html?spm=a2a0e.searchlist.list.1.532777c2bXAw9H&search=1</p> <p>Daraz Product 2 Link – https://www.daraz.pk/products/candy-15-ton-dc-inverter-turbo-cooling-wide-voltage-energy-efficient-csu-18ht-heat-cool-ac-air-conditioner-free-installation-i213602598-s1422164244.html?spm=a2a0e.searchlist.list.1.4ced77c2bE3k4l&search=1</p>	Display best to buy product, among the Daraz products.	Display best to buy product, among the Daraz products.	Pass

TC - 02	<ol style="list-style-type: none"> 1. Click on Compare Daraz Products from main menu. 2. Enter Daraz product 1 link 3. Click on Scrape and compare button. 	<p>Daraz Product 1 Link – https://www.daraz.pk/products/changhong-ruba-sc-12jwttw-10-ton-12000btus-wall-mount-split-air-conditioner-white-i107184555-s1255548100.html?spm=a2a0e.searchlist.list.1.532777c2bXAw9H&search=1</p> <p>Daraz Product 2 Link –</p>	Error: Invalid Search	Error: Invalid Search	Pass
TC - 03	<ol style="list-style-type: none"> 1. Click on Compare Products by Links from main menu. 2. Enter incorrect Daraz product 1 link 3. Enter incorrect Daraz product 2 link 4. Click on Scrape and compare button. 	<p>Daraz Product 1 Link – Air conditioner</p> <p>Daraz Product 2 Link – 4478</p>	Error: Invalid Search	Error: Invalid Search	Pass
TC - 04	<ol style="list-style-type: none"> 1. Click on Compare Products by Links from main menu. 2. Click on cancel button 	None	Returns to the home page.	Returns to the home page.	Pass

Table 6.6: Test Case # 6 - Compare Daraz Products

6.7.7. Test Case#7 - View Product

Use Case	UC-07				
Use Case Name	View Product				
Test Priority	Medium				
Precondition	User should be logged to their account.				
Test Cases					
Test ID	Test Steps	Input Data	Expected Output	Actual Output	Status
TC - 01	<ol style="list-style-type: none"> 1. Searches a product 2. System displays the product best to buy 	None	Display Product title, image, price and rating.	Display Product title, image, price and rating.	Pass

Table 6.7: Test Case # 7 - View Product

6.7.8. Test Case#8 - Select Product

Use Case	UC-08				
Use Case Name	Select Product				
Test Priority	Medium				
Precondition	User should be logged to their account.				
Test Cases					
Test ID	Test Steps	Input Data	Expected Output	Actual Output	Status
TC - 01	<ol style="list-style-type: none"> 1. Searches a product by keyword 2. System displays the product best to buy along with store link 	Click on store link	Displays product's provider webstore	Displays product's provider webstore	Pass

Table 6.8: Test Case # 8 - Select Product

6.7.9. Test Case#9 - Logout

Use Case	UC-09				
Use Case Name	Logout				
Test Priority	Medium				
Precondition	User should be logged to their account.				
Test Cases					
Test ID	Test Steps	Input Data	Expected Output	Actual Output	Status
TC - 01	1. Click on logout button.	Click on logout button	User should be logged out	User should be logged out	Pass

Table 6.9: Test Case # 9 - Logout

6.8. Results & Evaluation

The functional requirements were analyzed in order to write the test cases. All the test cases are executed manually so check whether the implementation of project matches the requirements defined. All the test cases were successfully passed while performing the testing on the system.

6.9. Conclusion

This chapter consisted of all the strategies and test cases that we made to test every module of our system. This chapter indicates that our project has successfully performed unit testing, component testing, integration test and system testing. All the test cases were performed manually.

Chapter 7

Conclusion

7.1. Chapter Overview

This chapter concludes our project ShopSmart. The contributions it will make for the society. The strengths and the weaknesses of the application and the future work that can be done on it.

7.2. Introduction

In this final chapter we would like to conclude our overall work on the project. The reader of this report will now have a clear and complete picture of the application being developed, its methodologies and usefulness of the application. This application has a lot of practical usage and will help online shopping users. We have tried to meet each and every requirement and details that were provided and required by our department. We tried our very best to achieve quality but there is always room for improvement and this application allows changes and enhancements.

7.3. Contributions

There are many systems that has already been developed with the intent of assisting the online shopping users. Those applications are based on the price comparison system. But at the time of inflation and lock down, people like to spend their money wisely, they like to compare the quality of product before purchasing so that they can choose the better one. There was a need for an application that can tell which store offers the best product as considering the busy lifestyle of the users, it is not feasible to go to each store and do a comparison manually, thus automatic was needed.

ShopSmart will recommend the user which product they should by based on the customers ratings, reviews and price. It offers a variety of categories where same products available on Daraz and iShopping are compared and among them which is best to buy is indicated. The user can also the compare two of Daraz products, by placing their links in the desired fields. This will help them in taking decision in a better way.

7.4. Reflections

In the era of science and technology, people have opted more towards the online shopping. Most of the working people do not have time to go to shopping. As costumers, they have the right to choose which shop is offering the best price and quality for a certain product that they are interested in.

7.4.1. Strengths

Following are the strengths of our system that make it unique:

1. User can search and compare products of three stores (Daraz, Telemart, and iShopping) based on user reviews, ratings and price.
2. The user can compare two of products available on Daraz, and the system will tell on which one is best to buy.
3. The ShopSmart application presents with the user with a category option where the user can view without searching that which product is best to but between Daraz and iShopping.

7.4.2. Weakness

Following are the weaknesses of our system:

1. The system provides the user with an opportunity to compare between three stores on the basis of search by keyword.
2. As there is limited time in Final Year Project (FYP), but the four categories could be enhanced.
3. As the system only offers two store products to be compared by link, this feature can also be enhanced and improved in the future.

These short comings can be covered and improved in the future, if we get the opportunity to enhance the system.

7.5. Future work

The future work that can enhance the functionalities of ShopSmart includes:

1. ShopSmart intends to include more shopping stores while comparing the product. so that a good number of products are analyzed before informing the user about the best to buy.
2. ShopSmart also intends to increase the number of categories provided to the user, the categories that can be added in the future work could be home groceries as it is mainly dealing with the electronics.
3. ShopSmart also intends to allow the user to compare the product of more than two stores via link.
4. Small businesses could also be integrated in the near future.

7.6. Conclusion

In this chapter we concluded our thesis work, by stating the strengths and the weaknesses of our system, what future work that can be performed to the system to enhance the functionalities as said before there is always a room for improvement

REFERENCES

[1] Morgana Gonzalez, "Comparison Sites", 2011.

https://www.google.com/search?q=morgana+gonzalez+research+on+comparison+sites&sxsrf=ALeKk01n44LsITTy3OBgWra6W6T9-kqrjA%3A1622993516144&ei=bOq8YLGdCMiBhbIPxoeogA0&oq=morgana+gonzalez+research+on+comparison+sites&gs_lcp=Cgdnd3Mtd2l6EAMyBwghEAoQoAE6BQghEKABUIMqWNNOYJxPaAFwAHgAgAGgAogB0CeSAQQyLTE5mAEAoAEBqgEHZ3dzLXdpesABAQ&sclient=gws-wiz&ved=0ahUKEwixg4KAqoPxAhXIQEEAHcYDCtAQ4dUDCA4&uact=5

[2] Emily Knight, the use of comparison sites in the uk general insurance market, 2009.

<http://utpedia.utp.edu.my/6281/1/KhamisahBuaimin12612.pdf>

[3] Agus Zainal Arifin, Dinial Utami Nurul Qomariah, Muhammad Riduwan, Addien Haniefardy, Yufis Azhar, Rizka Wakhidatus Sholikhah and Dini Adni Navastara, *Automatic Comparison of Products based on Opinion Features*, Indonesia: Palembang, 2018.

<http://eprints.umm.ac.id/60856/1/Arifin%20Qomariah%20Riduwan%20Haniefardy%20Azhar%20Sholikhah%20Navastara%20-%20Opinion%20Features%20Product%20Comparison%20Jaccard%20Similarity%20Synonym.pdf>

GLOSSARY

Notation	Description
VS CODE	A source code editor which allows the users to debug their code, include many IntelliSense extensions for smart code completion, code refactoring and showing problems in the code.
Firebase	Firebase is Google's application development platform that helps to build, improve and grow applications. Firebase can act as the server, API, and datastore.
Python	An object- oriented programming language that is used to write codes
React.js	An open-source JavaScript library that is used for building user interfaces specifically for single-page applications. It is used for handling the view layer for web and mobile apps.
HTML	Hypertext Markup Language is the coding language that is used to structure a web page and its contents.
CSS	CSS (Cascading Style Sheet) is the coding language used for designing the web pages like giving different colors to different HTML elements, fonts, layouts, and responsiveness of different elements according to the dimensions of different devices.
Bootstrap	Framework that is used to design the web pages faster with HTML and CSS based templates.
Web scrapping	Web scraping refers to the extraction of data from a website. This information is collected and then exported into a format that is more useful for the user.
Database	A database is an organized collection of structured information, or data, typically stored electronically in a computer system.