**EleCommerce**

# 

# ACKNOWLEDGEMENTS

I would like to express my deep sense of gratitude and convey thanks to everyone who helped me and supported during the completion of this project.

First, I would like to express a deep sense of gratitude to \_\_\_for helping, guiding, and supporting me throughout my bachelor’s degree.

I appreciate my department for offering the courses and a welcoming environment. I want to express my gratitude to \_\_\_my supervisor, for his encouragement of my ideas and for giving me the chance to develop the "EleCommerce" project. And assisting me in acquiring the skills required to create this project, as well as offering me complete guidance all the way through.

# 

# 

# UNDERTAKING

This is to inform you that the project "EleCommerce" represents a solid effort on my part. This project is a requirement for the completion of my degree "BSCS (Bachelor of Science in Computer Science)". I completed all project requirements, design, and development. This project is the result of my day and night efforts. This project is entirely unique and has never been presented to any other institution or university.

# ABSTRACT

Business growth plays a vital role in strengthening the economy of Pakistan. The business industry drives economic development through the creation of goods and services, employment opportunities, and the generation of income and profits. Ecommerce paved a new way for business growth by providing more and more opportunities for business firms. This project is a means of contributing to the company's growth, and the target market is electronics. Despite the fact that there are many electronic web applications currently available. However, all of them have flaws and faults that influence the overall user experience. I designed this web application with all the required user's needs in mind, as well as the difficulties that existed on the prior web application. In order to develop an e-commerce web application, a number of Technologies must be studied and understood. These include multi-tiered architecture, server and client side scripting techniques, implementation technologies such as ASP.NET, programming language (such as C#) and relational databases.

**TABLE OF CONTENTS**

[ACKNOWLEDGEMENTS](#_bookmark0)

[UNDERTAKING](#_bookmark1)

[ABSTRACT](#_bookmark2)

[CHAPTER 1: INTRODUCTION](#_bookmark3) 1

* 1. [Problem Statement](#_bookmark5) 2
  2. [Advantages](#_bookmark6) 2
  3. [Motivation 3](#_TOC_250009)
  4. Definition of Terms 4
  5. Goal of project 4
  6. Project Member 5
  7. Stakeholder 5

[CHAPTER 2: EXISTING SYSTEM 6](#_bookmark8)

* 1. Related web application projects 7
  2. [Novelty](#_bookmark10) 7
  3. [System Requirements Specification](#_bookmark12) 
     1. [User Interfaces](#_bookmark13) 8
     2. Functional and Non Functional Requirements 8
     3. Requirements Attributes 10
     4. Hardware Details 13
     5. Software Details 13
  4. [Use Case Diagram](#_bookmark14)  13

[CHAPTER 3: SOFTWARE PROCESS MODEL 16](#_bookmark15)

* 1. [Software Process Model 17](#_TOC_250008)
  2. [Proposed Model 17](#_TOC_250007)

[CHAPTER 4: PROPOSED SYSTEM](#_bookmark16) 19

* 1. [Existing](#_bookmark17) System 20
  2. [Need for Proposed System 20](#_TOC_250006)

[CHAPTER 5: SYSTEM DESIGN](#_bookmark18) 22

* 1. [UI/UX Design Pages](#_bookmark19) 24
  2. [ER Diagram](#_bookmark21)  27
  3. [DFD Diagram](#_bookmark22)  29

[5.4 Activity Diagram](#_bookmark23)  35

[5.5 Class Diagram](#_bookmark24)  36

5.6 Sequence Diagram 37

[5.7 Software Architectural Design](#_bookmark25) 38

[CHAPTER 6: DEVELOPMENT](#_bookmark26) 39

* 1. [Development](#_bookmark27) 40
  2. [Functional Requirements](#_bookmark28) 40
  3. [Non-Functional Requirements 42](#_TOC_250004)
  4. [Selection of tools and technology 42](#_TOC_250003)
  5. [System Specification 43](#_TOC_250002)

[CHAPTER 7: SOFTWARE TESTING](#_bookmark29) 44

* 1. [Testing Process 45](#_TOC_250001)
  2. [Test Case Design](#_TOC_250000)  46
  3. [Test Data](#_bookmark32) 47
  4. [Black Box Testing (behavioral testing)](#_bookmark33) 50
  5. [White Box Testing (structural testing)](#_bookmark34) 53

[CHAPTER 8: IMPLEMENTATION AND TRAINING 54](#_bookmark35)

* 1. [Implementation](#_bookmark36) 55

[Conclusion](#_bookmark38) 57

[References](#_bookmark39) 58

**LIST OF FIGURES**

**Fig 2.1:** Use Case (Admin) **15**

**Fig 2.2:** Use Case (Customer) **15**

**Fig 3.1:** The Iterative Waterfall Model **18**

**Fig 5.1:** Home Page **24**

**Fig 5.2:** Sign in **25**

**Fig 5.3:** Admin Dashboard **25**

**Fig 5.4:** Customer Dashboard **26**

**Fig 5.5:** Product Page **27**

**Fig 5.6:** Cart **27**

**Fig 5.7:** Checkout **28**

**Fig 5.8:** ER Diagram **30**

**Fig 5.9:** 0 Level DFD-Admin Side **31**

**Fig 5.10:** 1st Level DFD Diagram (Admin Side) **31**

**Fig 5.11:** 2nd Level DFD-Admin Side – Manage Product **32**

**Fig 5.12:** 2nd Level DFD-Admin Side – Manage Orders **32**

**Fig 5.13:** 2nd Level DFD-Admin Side – View Reports **33**

**Fig 5.14:** 1st level User side DFD **34**

**Fig 5.15:** 2nd level User side DFD – Manage Account Details **35**

**Fig 5.16:** 2nd level User side DFD - Purchase **36**

**Fig 5.17:** Activity Diagram Admin Side **37**

**Fig 5.18:** Activity Diagram Customer Side **38**

**Fig 5.19:** Class Diagram **39**

**Fig 5.20:** Sequence Diagram **40**

**Fig 7.1:** Flow Of Login **53**

**Fig 7.2:** Flow Of Add Sub Category **54**

**Fig 7.3:** Flow Of Add Product **55**

**LIST OF TABLES**

**Table 2.1:** Related projects **7**

**Table 2.2:** Functional and Non Functional Requirements **8-10**

**Table 2.3:** Hardware Requirements **13**

**Table 2.4:** Software Requirements **13**

**Table 7.1:** Login Object Table **49**

**Table 7.2:** Valid Input Data Table **50**

**Table 7.3:** Invalid Input Data Table **50**

**Table 7.4:** Add new Sub Category Object Table **51**

**Table 7.5:** Valid Input Data Table **51**

**Table 7.6:** Add new Product Object Table **51-52**

**Table 7.7:** Valid Input Data Table **52**

**CHAPTER 1: INTRODUCTION**

**INTRODUCTION**

E-commerce is becoming more and more necessary with the rise in technological use. Traditional brick-and-mortar stores are experiencing greater competition as a result of the growth of e-commerce, and many are turning to e-commerce to remain relevant and draw in new consumers. Ecommerce are also simpler to operate and less expensive to set up. Compared to a conventional business, an e-commerce web application may access a considerably wider audience. E-commerce is becoming highly common in the sector, and in the years to come, online sales are predicted to increase.

This is a web application that offers various electronic appliances online. The use of these appliances will simplify our daily lives. This project intends to create an online platform that users and the owner may access from anywhere. Users may purchase appliances from the comfort of their own homes with a single click and without any physical practice. All business activities may be tracked by the owner. Products and customer data may be easily managed.

A vast variety of goods and services will be available to browse in the virtual store, which will be divided into several categories and subcategories. When ready to make a purchase, customers can explore a variety of items, add them to their basket, and then proceed to the checkout. Online payments can be done in a number of ways. The company will reach a wider audience than a physical store, which is only open to local customers. Sales will automatically rise when the company expands its customer base. The business will get greater advantages as a result of the higher sales volume.

* 1. **Problem Statement**

Buying electronics at a real store is a difficult process. It becomes tough and time-consuming to visit crowded markets and different retailers and then compare costs. An electronic virtual store will offer people a platform from which they can purchase electronics while sitting in the comfort of their own homes. They can use their credit cards to make payments after purchasing. The appliances they purchase will be delivered to their homes.

* 1. **Advantages**
* **Shopping with comfort and ease:**

You won't have to spend your weekends or nights going to several places to conduct things if you purchase online. From the comfort of your sofa, you may purchase goods from any city.

* **Product Information:**

All the important information regarding products will be displayed. You don't have to ask the salesman again and again for product specifications.

* **Comparison of price:**

Because all of the items will be available on a virtual store, you may compare them to similar products on other web applications to see if there are any cheaper possibilities, available.

* **Carrying Appliances:**

It's really tough to get appliances to your home. However, if you purchase them from an online store, the appliances will be delivered right to your door.

* **Advertising and marketing at a low cost**

Sellers do not need to spend a lot of money on advertising their products. There are various inexpensive and quick ways to promote online in the world of ecommerce. Sellers can truly show off their product on ecommerce marketplaces since they are visual platforms.

* 1. **Motivation**

In our ever-changing technological world, opening an e-commerce business is one of the smartest and most important steps toward success. An E-commerce store is easy for the owner to manage and can be viewed at any time. Customers' and stock records will be carefully stored and quickly accessed or modified. Business advancement more people will learn about your company, the number of clients will double, and more profit will be made with less effort.

**1.5 Definition of Terms**

**Terms used in our project are following:**

**Admin:** The highest level of access to the e-commerce web application will be in the hands of the admin. Admin can add, modify or delete any information regarding brands, products, and categories. Admin can make any modification in web application layout or design.

**Order:** Orders will include all data indicating whether the order is fulfilled, in process, or cancelled. Order details like number of products their quantity and costs will be kept here as well.

**Category:** Products have different categories. If the customer wants to browse a specific category this feature will take the user to that page.

**Sub Category:** Subcategories will provide users with more precise information. Through subcategories, users can easily find their desired product.

**Vouchers:** Vouchers will be issued to customers they have an expiry date. Customers can these vouchers before expiry and get discount on their purchased products.

**Customer Dashboard:** Customers will also be provided with a dashboard once they register themselves. The customer dashboard will manage their account information and keep track of their purchases.

**Invoices:** An invoice is a document issued by the seller to the customer in order to collect payment. It includes the price and amount of the acquired goods. Invoices may also include the names and addresses of the client, a description and price of the products or services, and the terms of payment. Assist admin and customers with order management.

* 1. **Goal of project:**

**Responsive Design:** Ecommerce web application will have a responsive design will dynamically change web application appearance depending upon screen size.

**User Friendly:** It is a user-friendly and professional web application any user can operate with a minimum amount of knowledge.

**Virtual electronic store:** Users can buy appliances of different brands or categories according to their convenience.

**Stock management:** Stocks record will be secured and safe.

**Reporting:** Reports will generated based upon the information available like stock report, purchase, sales and profit/loss.

**Advancement of business:** Through ecommerce web application business services will advanced to different cities.

**1.7 Project Member:**

The project titled “EleCommerce” is an individual project that does not have a group of members. Maria Qureshi roll no 2K19-BSCS-219 from section G developed this e-commerce web application. All the responsibilities such as:

• Requirement Gathering

• Front-end

• Back-end

• Documentation

Are fulfilled and completed by an individual person.

**1.8 Stakeholder:**

The stakeholder is the person for whom this e-commerce web application is created by fulfilling all the requirements provided by him.

The stakeholder of project “EleCommerce” is the owner of Ali Electronics which is an electronic shop located in Multan. This project is the virtual store for Ali Electronics.

**CHAPTER 2: EXISTING SYSTEM**

# 2.1 Related projects

# Table 2.1: Related projects

|  |  |
| --- | --- |
| Store Name | Web application |
| HAQ Electronics | https://haqelectronics.com/ |
| Yasir Electronics | https://yasirelectronics.com/ |
| Home Appliances.pk | https://homeappliances.pk/ |
| Surmawala.pk | https://surmawala.pk/ |
| Al fatah Electronics | https://www.alfatah.com.pk/ |

**2.2 Novelty**

I visited all of the web applications listed in the table above, as well as many more, to gather user experience and identify flaws. My study has taught me how to give consumers with a fully unique experience via this web application. A one-of-a-kind experience will be accomplished via originality and by resolving concerns seen on other web applications.

The following are some faults that other web applications have, but this web application does not:

* Many web applications are not search engine optimized, which means they were not created or constructed with search engine optimization (SEO) in mind.
* Many web applications offer a difficult and nonprofessional interface for the user that causes difficulty to navigate, confusion, and frustration for users.
* Many web applications take more than a couple of seconds to load, which may irritate users and cause them to quit the page.
* Many web applications are charging shipping cost for delivering the appliances on your doorsteps.
* Many web applications don’t provide the option for reviews and ratings on products.

**2.3 System Requirement Specification**

Software Requirement Specifications (SRS) is a detailed description of the software system to be developed, including its features, constraints, and interfaces. The purpose of SRS is to provide a common understanding of the software requirements and to ensure that the final product meets the customer's needs and expectations.

**2.3.1 USER INTERFACES**

This web application will have three user interfaces:

* **Main Web application Interface**
* **Admin User Interface**
* **Customer User Interface**

**2.3.2 Functional and Non Functional Requirements**

**Table 2.2: Functional and Non Functional Requirements**

|  |  |  |
| --- | --- | --- |
| Requirement | Category | Attribute |
| Sign Up | Functional | System should allow users to sign up on the system |
| Sign In | Functional | System should allow users to sign in on the system |
| View Profile | Functional | System should allow users to view their profile |
| View Product | Functional | System should allow users to view products |
| View Brands | Functional | System should allow users to view brands |
| View Categories | Functional | System should allow users to view categories |
| Add to Cart | Functional | System should allow users to add product to cart |
| Delete from Cart | Functional | System should allow users to delete product from cart |
| Edit Cart | Functional | System should allow users to edit product from cart |
| Add rating to product | Functional | System should allow users to give review on products |
| Add review on product | Functional | System should allow users to add rating to a product |
| View other users review/rating | Functional | System should allow users to view other users review/ratings |
| User Verification | Functional | System should allow users to verify themselves by entering valid credentials |
| View Invoices | Functional | System should allow users to download and view invoices. |
| View Reports | Functional | System should allow admin to view reports |
| Search product through text | Functional | System should allow users to search product through text |
| View Total Purchases | Functional | System should allow admin to see total purchases |
| Add Update/Delete category | Functional | System should allow admin to add , delete or update category |
| Add/Update/Delete product | Functional | System should allow admin to add , delete or update product |
| Add Update/Delete subcategory | Functional | System should allow admin to add , delete or update sub category |
| Security | Non-Functional | System should be able to secure data of users |
| Maintainability | Non-Functional | System should be able to back up data to ensure data integrity |
| Performance | Non-Functional | System should be responsive to make sure user experience is not being compromised |

**2.3.3 Requirements Attributes**

The requirements shell is a guide to writing each atomic requirement. The components of the shell (also called a “snow card”) are identified below.

|  |  |
| --- | --- |
| **Name** | **Description** |
| **Id** | Unique identifier within the document |
| **Heading** | Short name of the document section or the requirement |
| **Type** | Type of the document object (Section, Information, Functional Requirement, Constraint, &dots;) |
| **Rationale** | Explanation why the requirement is important and how it contributes to the product's purpose |
| **Originator** | Name of the person who raised the requirement in the first instance, or the person to whom it can be attributed |

**Requirement# 1**

**Requirement Type:** Functional

**Description:** System should allow users to sign in on the system.

**Rationale:** Signing in to the system and gaining its access is the important requirement for both user and admin.

**Originator:** Stakeholder

**Fit Criterion:** We achieve this goal by making an interface in the ecommerce web application.

**Requirement# 2**

**Requirement Type:** Functional

**Description:** System should allow admin to add and edit products.

**Rationale:** Admin controls the complete system and only admin is allowed to add or edit the details of the products.

**Originator:** Stakeholder

**Fit Criterion:** We achieve this goal by making an interface in the ecommerce web application.

**Requirement# 3**

**Requirement Type:** Functional

**Description:** System should allow visitor and customers to view product.

**Rationale:** For an ecommerce web application product are the most essential part because user will buy something once they can see them.

**Originator:** Stakeholder

**Fit Criterion:** We achieve this goal by making an interface in the ecommerce web application.

**Requirement# 4**

**Requirement Type:** Functional

**Description:** System should allow users to add product to cart.

**Rationale:** Customers can buy from the web application when they can add the products to the cart.

**Originator:** Stakeholder

**Fit Criterion:** We achieve this goal by making an interface in the ecommerce web application.

**Requirement# 5**

**Requirement Type:** Functional

**Description:** System should allow users search product by name.

**Rationale:** Customers and visitor can search a particular product in the web application.

**Originator:** Stakeholder

**Fit Criterion:** We achieve this goal by making a search product function in the ecommerce web application.

**Requirement# 6**

**Requirement Type:** Functional

**Description:** System should allow users to edit and delete.

**Rationale:** Customers can able to change their products and delete them from cart.

**Originator:** Stakeholder

**Fit Criterion:** We achieve this goal by making an interface in the ecommerce web application.

**Requirement# 7**

**Requirement Type:** Functional

**Description:** System should allow users to add rating and reviews.

**Rationale:** Customers can able to add their review about a certain product and rate it according to their satisfaction level.

**Originator:** Stakeholder

**Fit Criterion:** We achieve this goal by making an interface in the ecommerce web application.

**Requirement# 8**

**Requirement Type:** Functional

**Description:** System should admin to view the reports.

**Rationale:** Different reports like stock and sales will be generated by the system in order to keep the record.

**Originator:** Stakeholder

**Fit Criterion:** We achieve this goal by making an interface in the ecommerce web application.

**Requirement# 9**

**Requirement Type:** Non Functional

**Description:** Performance.

**Rationale:** The web application should load fast and be able to manage large traffic levels without slowdowns.

**Originator:** Developer

**Requirement# 10**

**Requirement Type:** NonFunctional

**Description:** Usability.

**Rationale:** The web application should be simple to use, with distinct categories and product descriptions, as well as simple search and filtering features.

**Originator:** Developer

**2.3.4 Hardware Requirements**

**Table 2.3: Hardware Requirements**

|  |
| --- |
| **Hardware Requirement** |
| Processor 1 GHz |
| RAM 512 MB |
| Minimum disk space (32-bit) 4.5 GB |
| Minimum disk space (64-bit) 4.5 GB |

**2.3.5 Software Requirements**

**Table 2.4: Software Requirements**

|  |  |
| --- | --- |
| **Software** | **Description** |
| Operating System | Windows. |
| Server | IIS web server |
| Languages | C#, HTML, CSS, BOOTSTRAP, JavaScript |
| Tools | Visual Studio, MS SQL |
| Framework | .NET Framework |

**2.4** [**Use Case Diagram for System**](https://meeraacademy.com/use-case-diagram-for-online-shopping/)

**List of Actors:**

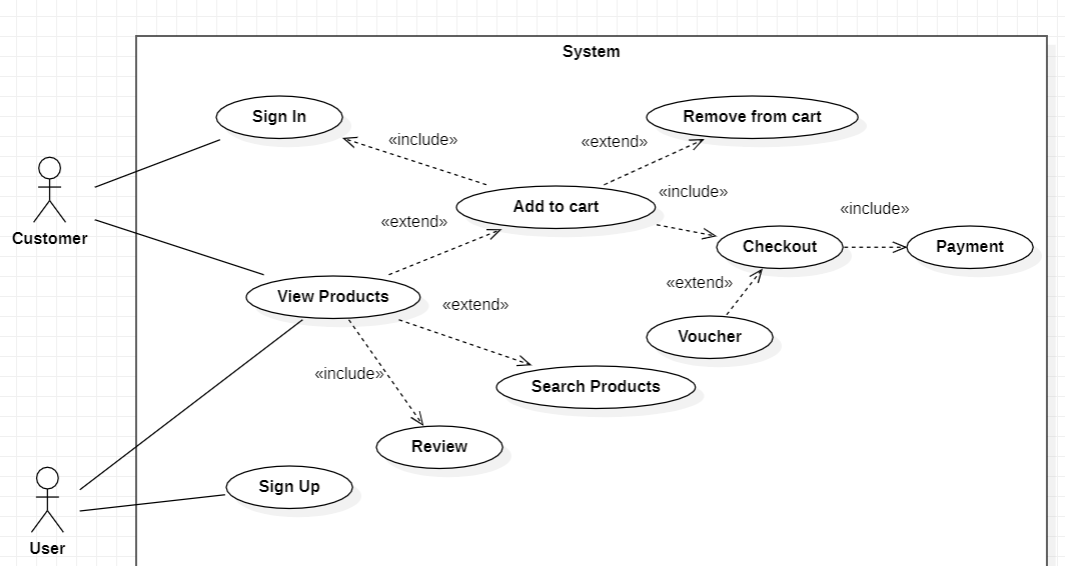
**Visitor:** This actor will be able to take overview of the web application

**Customer**: This actor will be able to make purchases on the web application

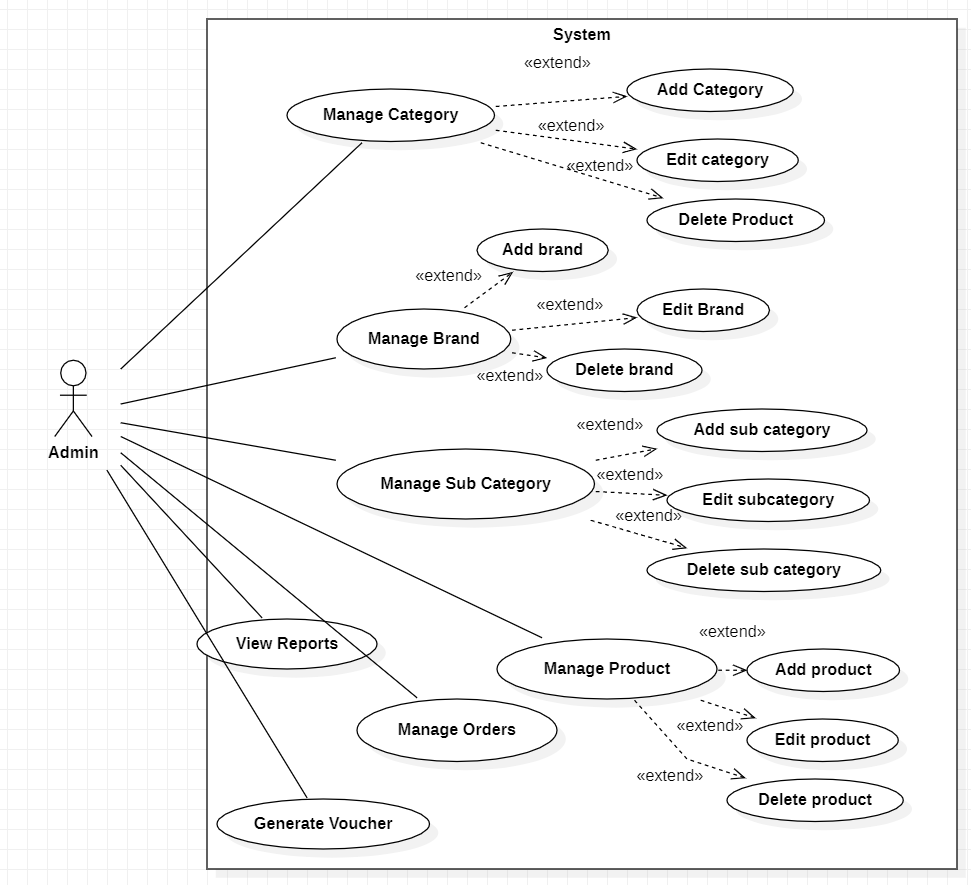
**Admin:** This actor will be monitoring the whole web application

## **List of use cases:**

* **Sign Up:** Sign up will allow the visitor coming to the web application to sign up themselves so they will become registered users.
* **Sign In:** Allow the registered customers of the web application and the admin to sign in to the web application.
* **Add to Cart:** When customers want to buy products that products will added into the cart.
* **Remove from Cart**: In case some unwanted products will added so those products can be removed from the cart.
* **View Product**: Allow visitors and customers to view all the products available on the web application.
* **Search Product**: Allow visitors and customers to search specific products on the web application.
* **Add product**: Admin can add new products to the web application with its images, description and price.
* **Delete products**: Allow admin to delete products
* **Edit products**: Allow admin to edit products
* **Add Category**: Allow admin to add category
* **Delete Category**: Allow admin to delete category
* **Edit Sub Category**: Allow admin to edit subcategory
* **Add Sub Category**: Allow admin to add subcategory
* **Delete Sub Category**: Allow admin to delete sub category
* **Edit Brand**: Allow admin to edit brand
* **Add brand**: Allow admin to add brand
* **Delete Brand**: Allow admin to delete brand



**Fig 2.1:** Use Case (Customer)



**Fig 2.2:** Use Case (Admin)

**CHAPTER 3: SOFTWARE PROCESS MODEL**

**3.1 Software Process Model**

A software process model is a structured approach to software development that defines how software is developed and maintained.

**A model will define the following:**

* The tasks to be performed
* The input and output of each task
* The pre and post conditions for each task
* The flow and sequence of each task

Different software process models appropriate for developing web applications:

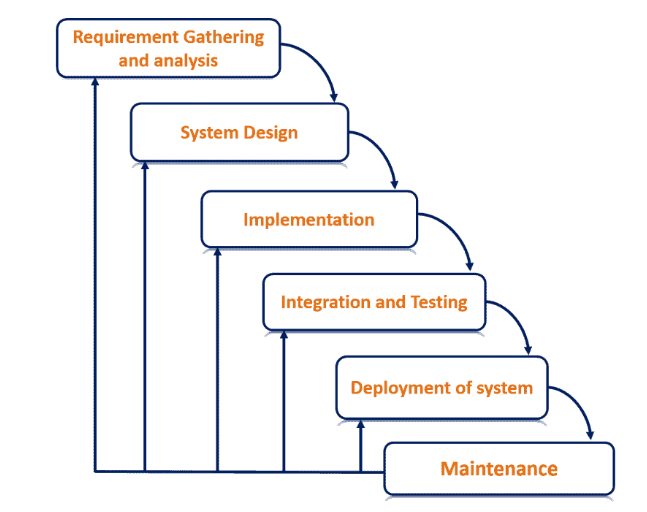
1. Waterfall Model
2. Agile Model
3. Spiral Model
4. V-Model
5. Iterative and Incremental Model

Each model has a different approach, with its own set of phases, steps, and activities.

**3.2 Proposed Model**

The **iterative waterfall model** was proposed to overcome the shortcomings of the classical waterfall model. In this model development process flows in linear sequential manner but developer is allowed to go to previous phase to make any sort of modification or changes.

This is possible because, in this model, there are feedback paths provided for every phase and no matter in which phase you are working, you can always go back to the previous phases and make the necessary changes.



**Fig 3.1: The Iterative Waterfall Model**

#### **Advantages of Iterative Waterfall Model:**

**Flexibility:** The model allows for changes and modifications to be made during the development process.

**Better risk management:** The model helps to identify and mitigate risks early in the development process.

**Increased transparency:** The clear and distinct phases of the model provide better visibility and clarity for stakeholders.

**Better stakeholder engagement:** The model provides opportunities for stakeholders to provide feedback and input throughout the development process.

**CHAPTER 4: PROPOSED SYSTEM**

**4.1 EXISTING SYSTEM**

The current system for shopping of electronic appliances have a lot of flaws and issues. They make it difficult for the users to make purchases.

**4.1.1 Limitations of existing system:**

* It is less user friendly.
* Product information is incomplete.
* Sell products more than the market price.
* Offers limited shipping methods.
* Bad user experience.
* Shipping Rates
* Loading speed is too low

**4.2 NEED FOR PROPOSED SYSTEM**

In the proposed system, the customer does not need to go to the store to buy the products. They may order the items he desires using web application on his smartphone or laptop. The web application will save money by buying online since the prices are lower than market pricing and purchases will be received at home. You will get a variety of items online and pick which ones we want. There are many such web applications currently accessible, but they all have certain flaws. This specific web application is devoid of all of these flaws.

**4.2.1 Merits of Proposed System**

* **Convenience:** This is one of the main reasons that online shopping has become so popular, as it allows you to switch stores and products by clicking a button rather than traveling to a new store.
* **Selection:** Of course, a large selection means that your decision making process may be a bit more difficult, but it also makes it more likely that you will find a high quality product that truly pleases you.
* **Immediacy:** When you purchase a new product, whether for yourself or for another person, it is always nice to have that product in your possession immediately.
* **Saving Money:** Another very important aspect of any shopping experience is trying to save as much money as possible. One reason that people enjoy online shopping is that you can often find a product more cheaply online than you can in stores.
* **Improved User Experience:** The proposed system intends to improve the user experience by providing a modern and user-friendly interface, simple navigation, and personalized suggestions. Customers will be more satisfied and retained if they can simply find and get the items they want.
* **Comprehensive product record management:** For electronic appliances, the system include a strong product record management feature that allows administrators to easily add, update, and organize products, including product details, images, pricing.
* **CMS:** The proposed system includes a user-friendly CMS that allows administrators to quickly manage web application material such as product descriptions, photos, and other content to keep the web application up to date and entertaining for customers.

**CHAPTER 5: SYSTEM DESIGN**

**Brief Description**

This project is an ASP.NET MVC ecommerce web application for electronic appliances that would be an online store that sells a wide range of electronic appliances such as Televisions, laptops, smartphones, home theatre systems, and more. Customers will be able to simply search and explore items, check product information and specs, and make online purchases due to the web application's user-friendly layout. In addition, numerous payment methods including cash on delivery are available. Separate dashboards for admin and customers to help them manage their data. The web application would also have a customer review section where users could offer feedback and ratings on items and services, so assisting other customers in making educated purchasing decisions.

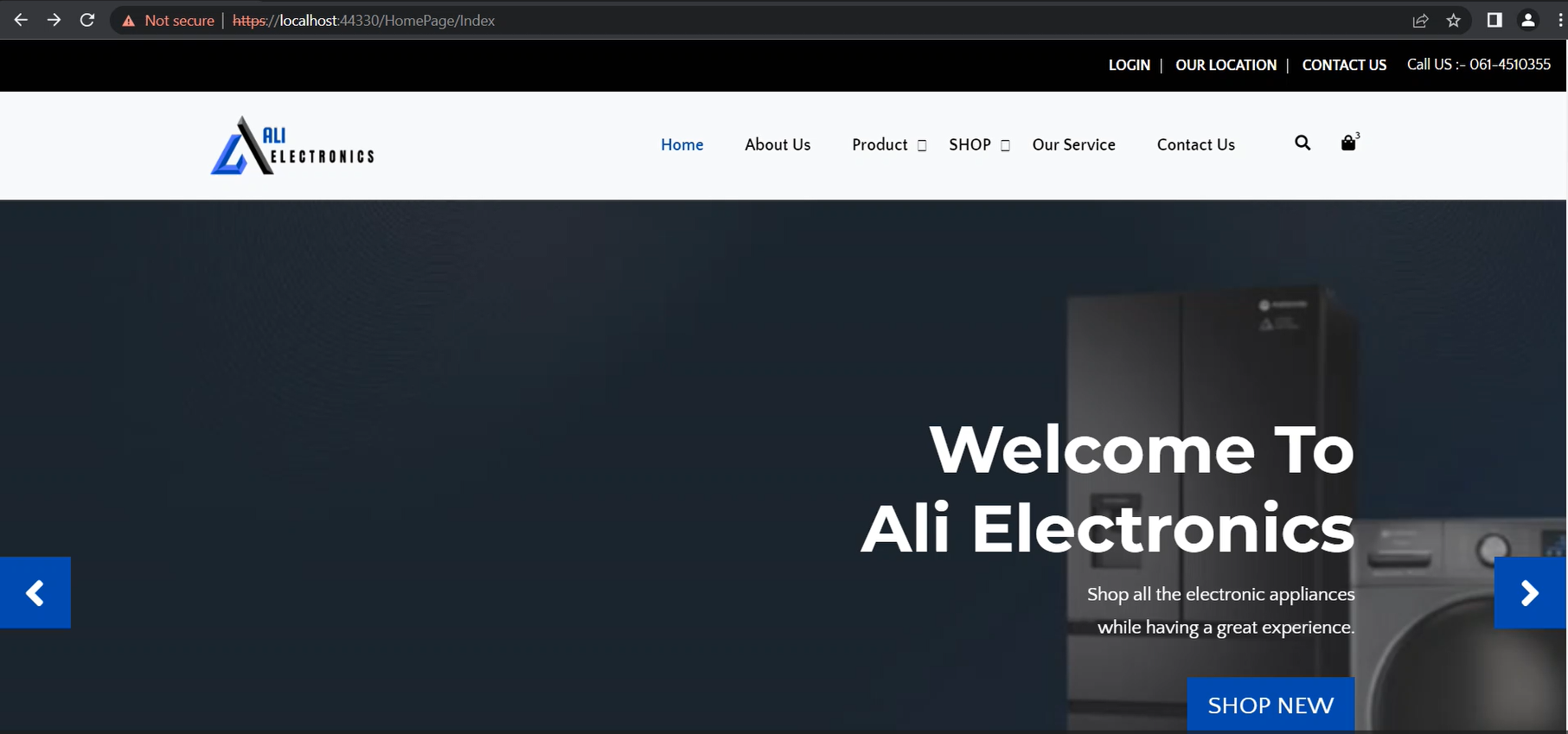
**Fully Dressed Format**

It is an ecommerce web application designed for a single shop that offers a variety of electronic equipment from several brands. Customers may simply examine and purchase things from the comfort of their own homes, eliminating the need to travel markets and other stores. The goal is to create a convenient and safe online platform for customers to purchase electrical appliances and associated items. With his dashboard, Admin may control all of the goods, their supplies, and order information. Consumers may see and amend their order information and account data via their dashboard.

**Features:**

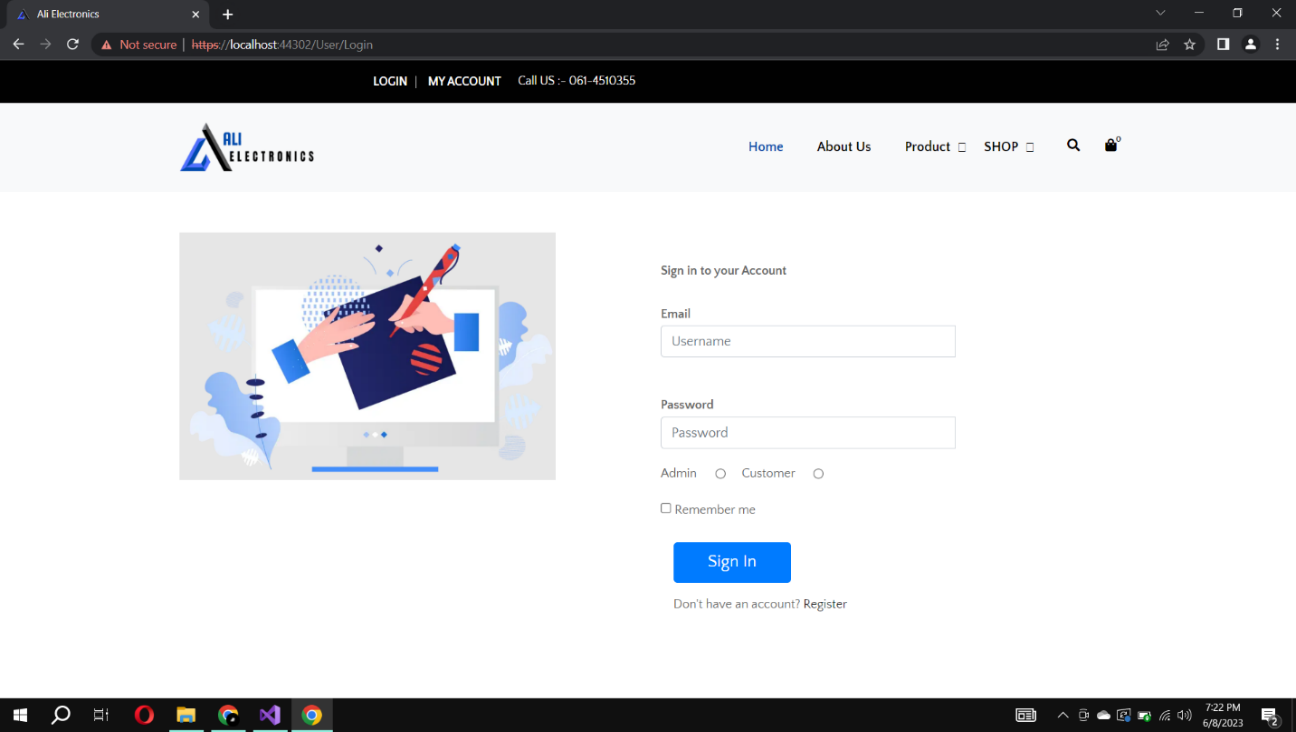
1. **User-friendly interface:** The web application will have an easy-to-use interface that allows customers to browse products, view details, and make purchases.
2. **Product catalog:** The web application will feature a comprehensive catalog of electronic appliances, including product descriptions, specifications, and images.
3. **Search functionality:** Customers will be able to search for products by brand, category, and other parameters.
4. **Shopping cart:** The web application will have a shopping cart feature that allows customers to add products, view prices, and proceed to checkout.
5. **Reviews and ratings:** The web application will allow customers to leave product reviews and ratings to help other customers make informed purchase decisions.
6. **Deals and discounts:** The web application will offer deals and discounts on selected products to attract and retain customers.
   1. **UI/UX Design Pages**

**1). Homepage**



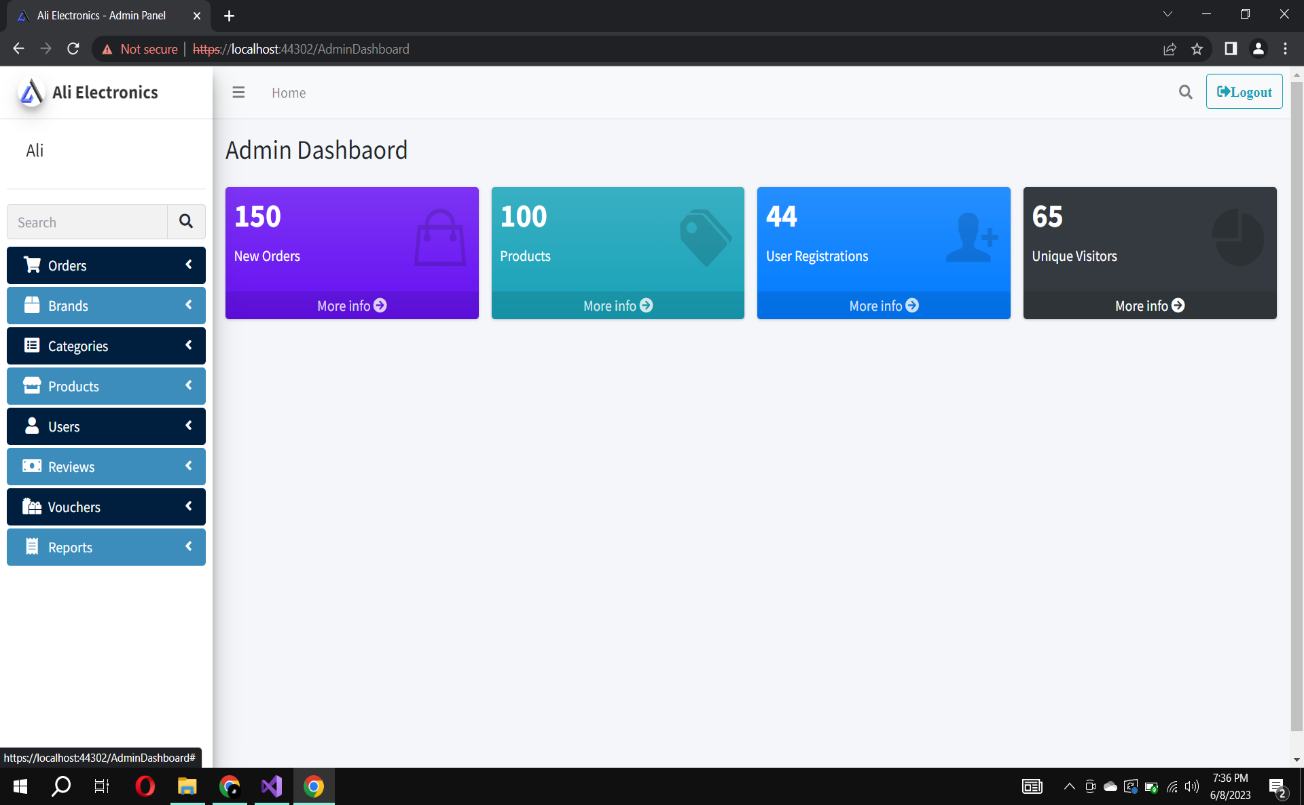
**Fig 5.1: Home Page**

**2). Sign In**

****

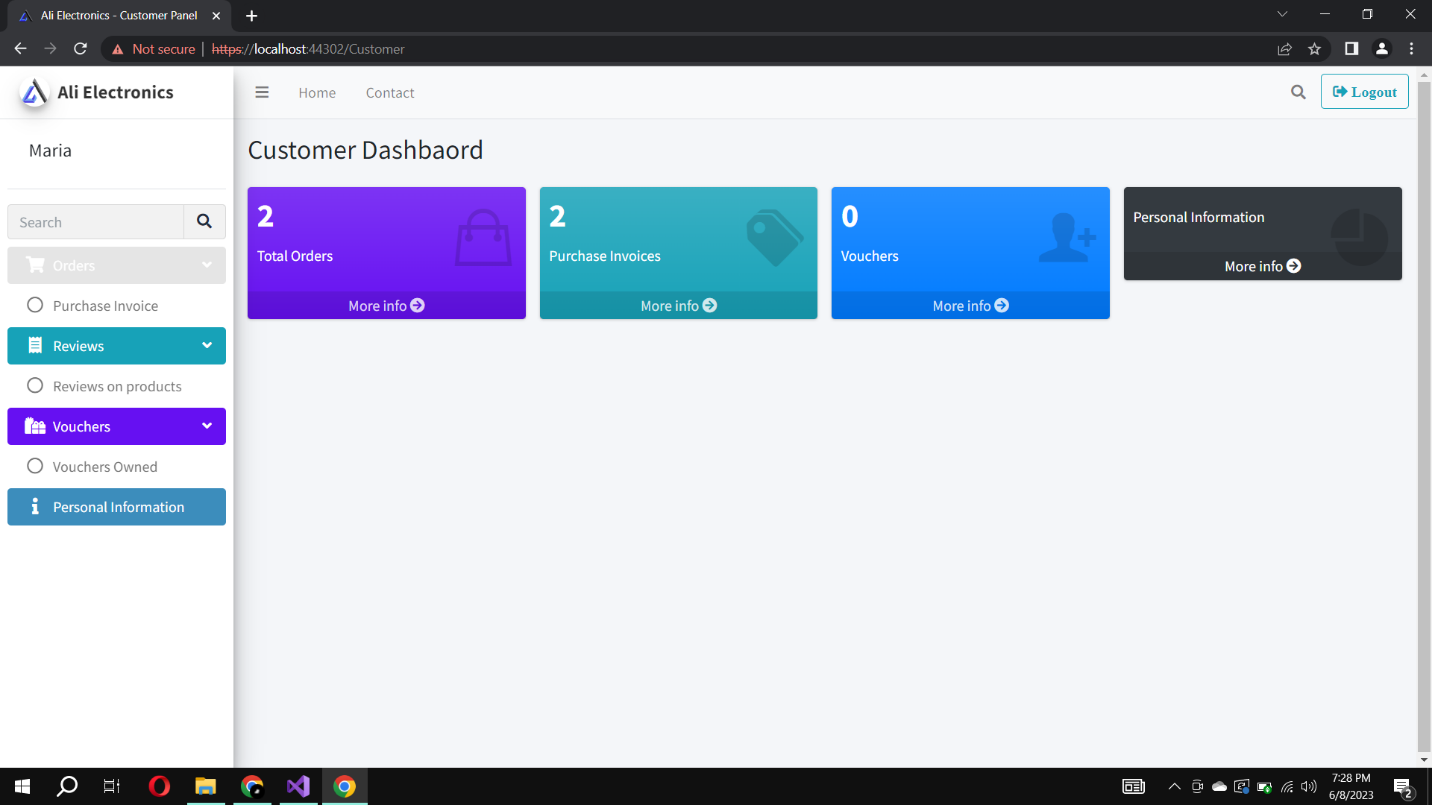
**Fig 5.2: Sign In**

**3). Admin Dashboard**

****

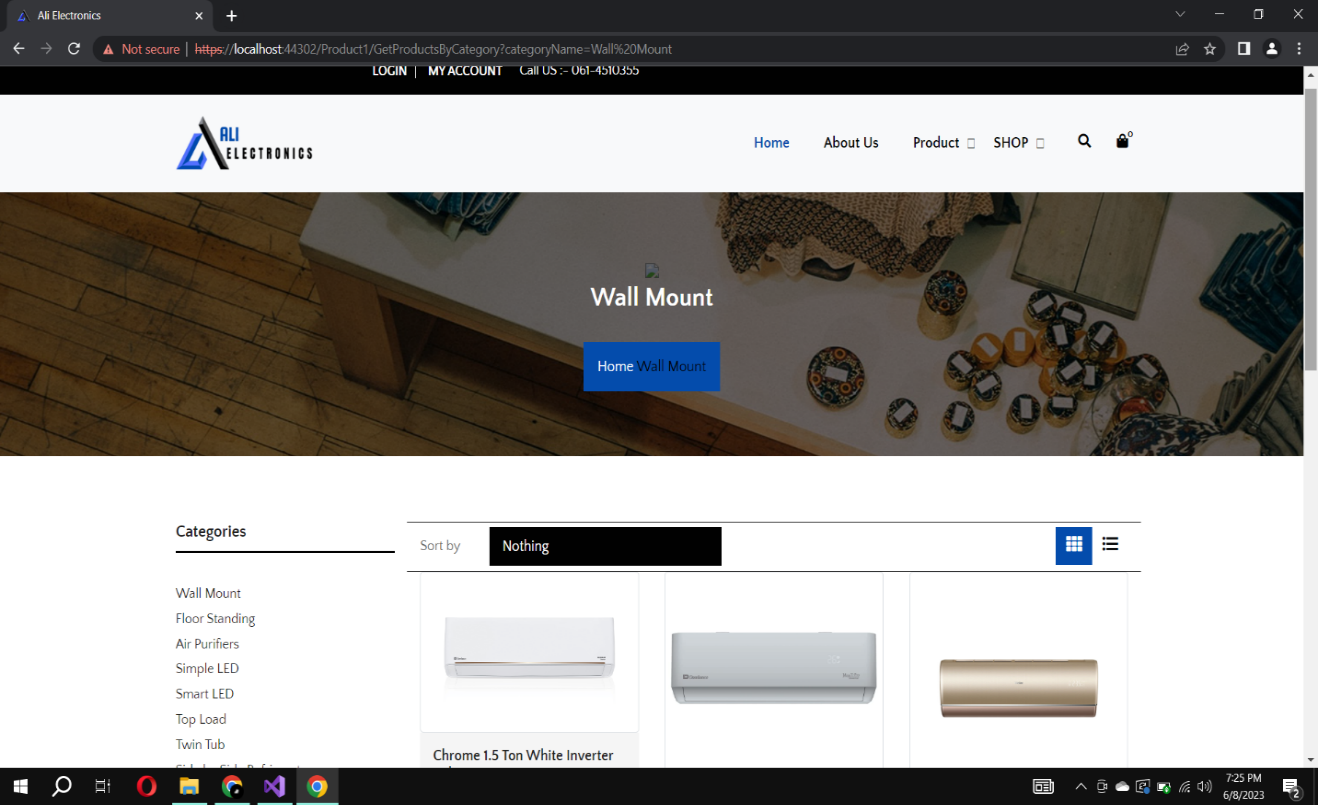
**Fig 5.3: Admin Dashboard**

**4). Customer Dashboard**

****

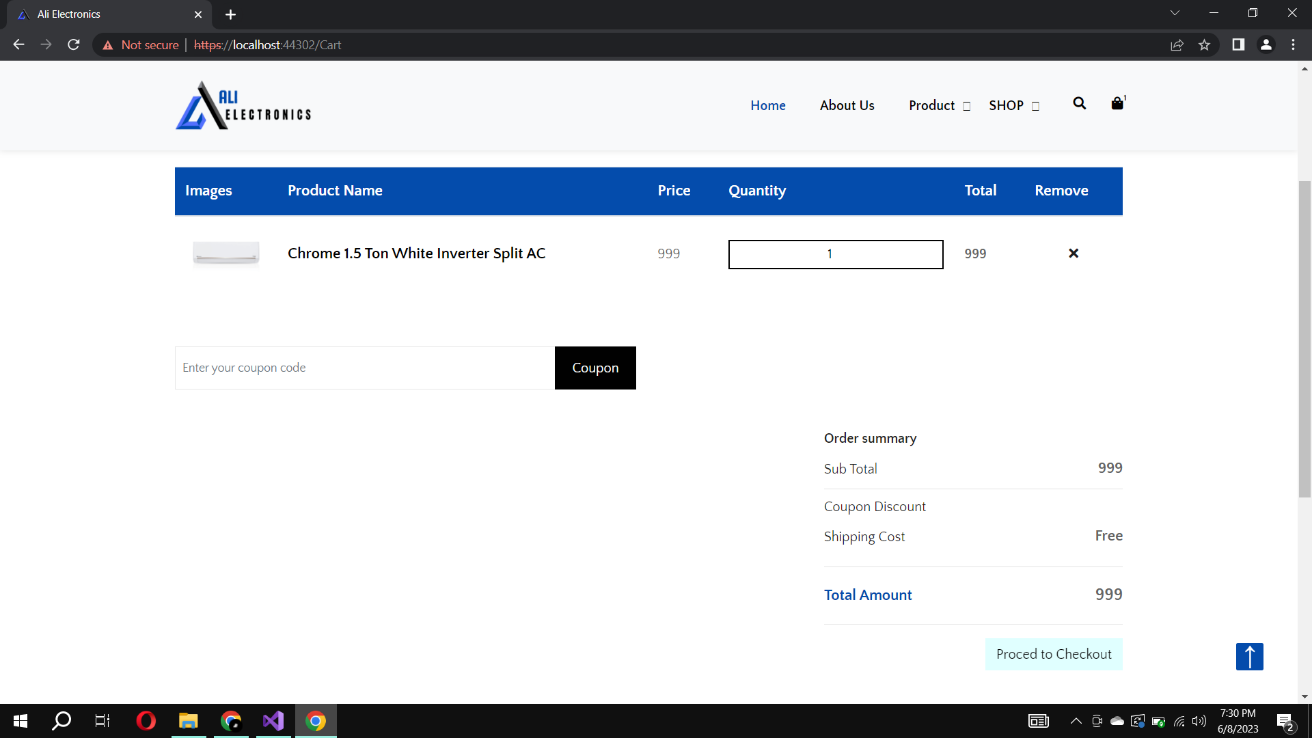
**Fig 5.4: Customer Dashboard**

**5). Products Page**

****

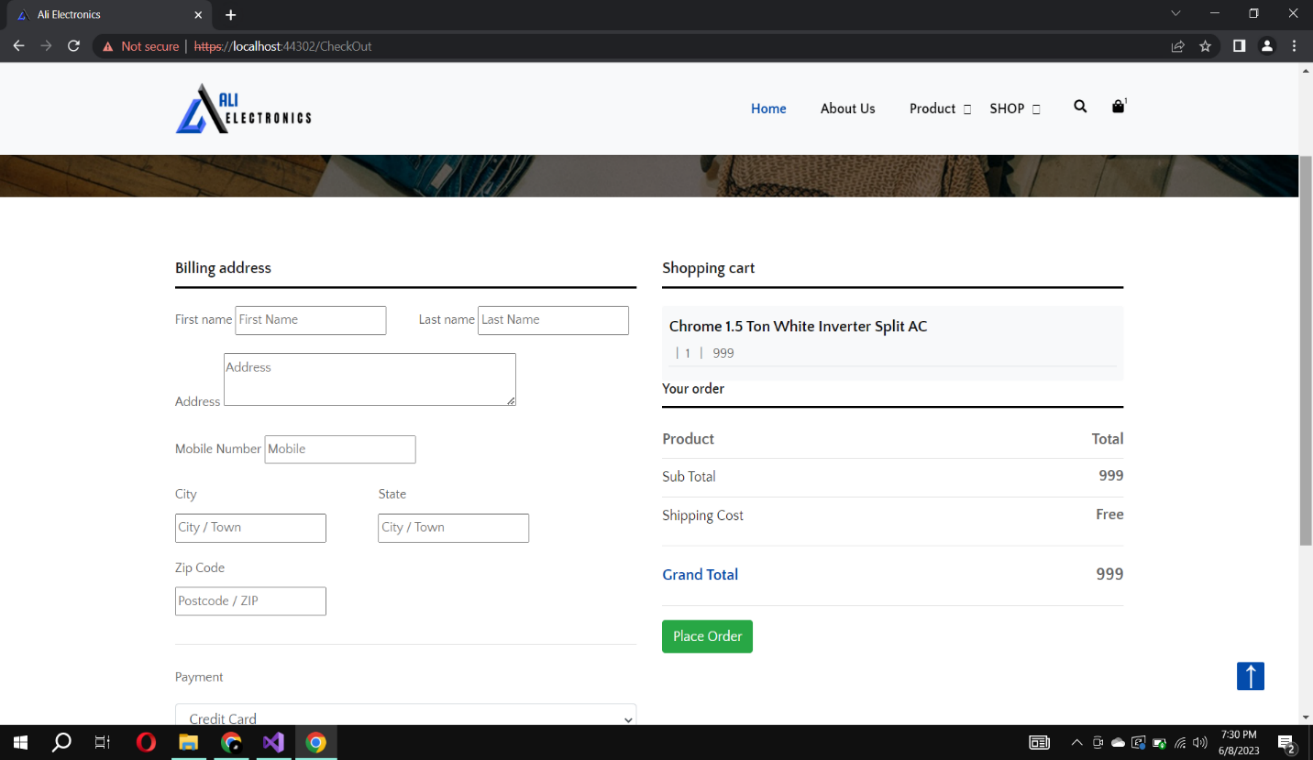
**Fig 5.5: Product Page**

**6). Cart**

****

**Fig 5.6: Cart**

**7). Checkout**

****

**Fig 5.7: Checkout**

**5.2 ER Diagram for Ecommerce Web application**

Entity-Relationship diagram (ER diagram) is a graphical depiction of entities, properties, and the relationships between them in a database. It is used to create or view a database schema, which describes the structure of the data that will be stored in the database. An ER diagram for an ecommerce web application selling electronic equipment would depict the system's entities, characteristics, and relationships.

**Entities**

**We have the following entities in this diagram:**

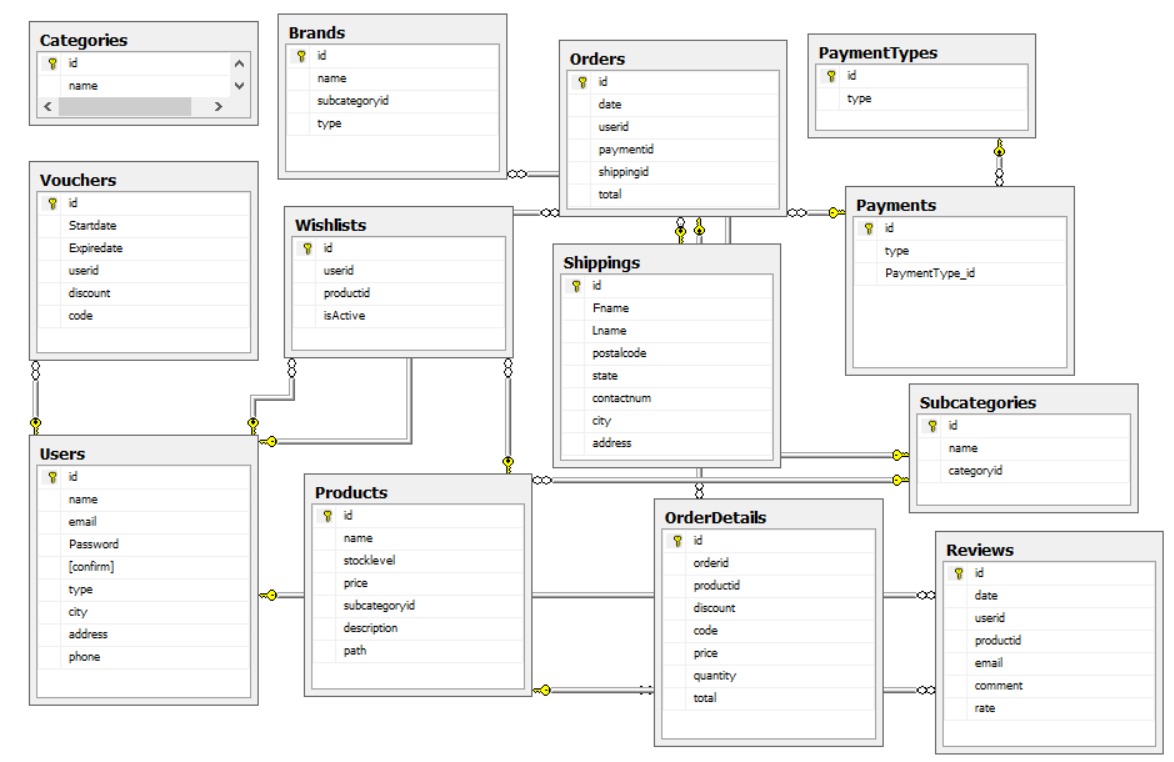
* **Users:** This entity maintains information about customers and admin, such as their name, email address, street address, and phone number. A unique ID is assigned to each user.
* **Products:** This entity maintains information about the ecommerce web application's items, such as the product name, description, price, and subcategory. A unique ID is assigned to each product.
* **Brands:** This entity maintains information about the ecommerce web application's brands, such name and category. A unique ID is assigned to each brand.
* **Categories:** This entity maintains information about the ecommerce web application's categories, such as name. A unique ID is assigned to each category.
* **Sub Categories:** This entity maintains information about the ecommerce web application's sub categories, such as the name and category. A unique ID is assigned to each sub category.
* **Orders:** This entity keeps track of client orders, such as the order date, status, and total price. Each order is assigned a unique Order ID.
* **Order Details:** This entity maintains information about the products included in each order, such as number and price. An Order Details ID is assigned to each order item.
* **Registration:** This entity maintains information about the customer details needed at the time of shipping.
* **Payment:** This entity maintains information about the payment of the order that the customer will pay.
* **Vouchers:** This entity maintains information about the discount vouchers assigned to the user.
* **Reviews:** This entity maintains information about the reviews given by the user on different products.

**Relationship**

**The diagram also depicts the following relationships between the entities:**

* A consumer can place many orders, but each order is only placed by one customer.
* An order can contain several order items, and each order item can only belong to one order.
* A product can be included in many order items, but each order item can only include one product.
* Several users can offer reviews on a single product.

**ER Diagram:**



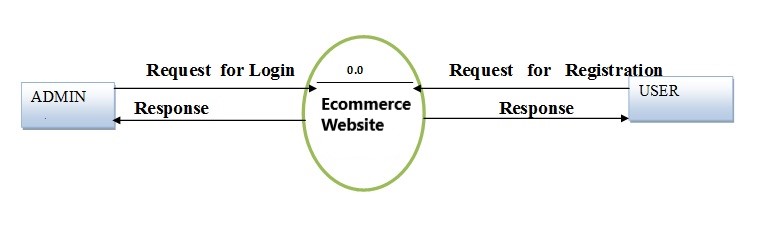
**Fig 5.8: ER Diagram**

**5.3 DFD Diagram**

A data flow diagram (DFD) depicts the movement of data through a system or process graphically. It demonstrates how data flows through various system components, how it is processed or modified, and how it is stored or outputted. Data flow diagrams are useful for modelling a wide range of systems and processes, such as information systems, business processes, and software applications. They are often used in system analysis and design, as well as in system documentation and communication.

**Context level DFD – 0 level**

The context level data flow diagram (dfd) is a diagram that describes the entire system. All user modules that run the system are described by the (0) level dfd. The data flow diagram of an online shopping site shown below demonstrates how two people may control the system admin and users.



**Fig 5.9: 0 Level DFD-Admin Side**

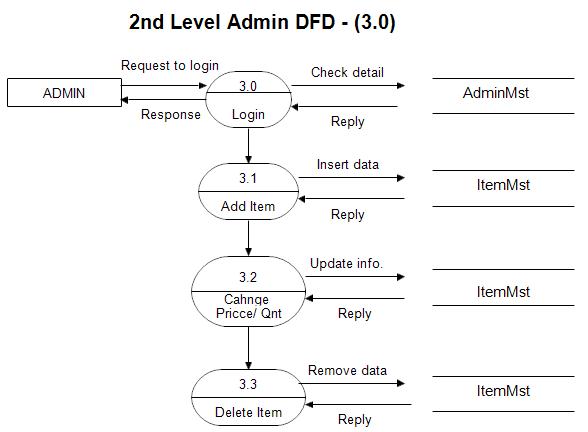
### **1st Level Admin Side DFD**

The admin side DFD describes the capabilities of Admin, who is the owner of the web application. Admin can first add a category of item and then add things by category. and admin may control order and payment information.



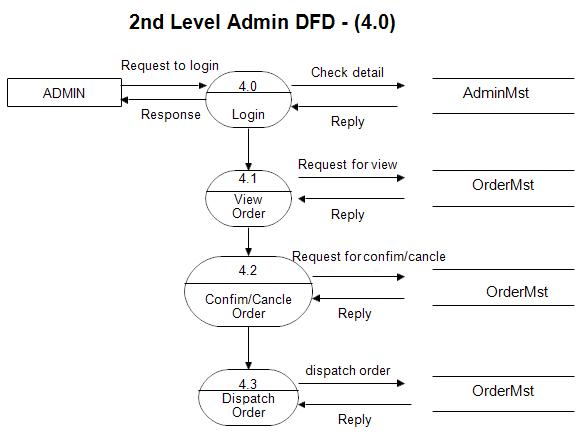
**Fig 5.10: 1st Level DFD-Admin Side**

### **2nd Level – Admin side DFD (3.0) – Manage Product**



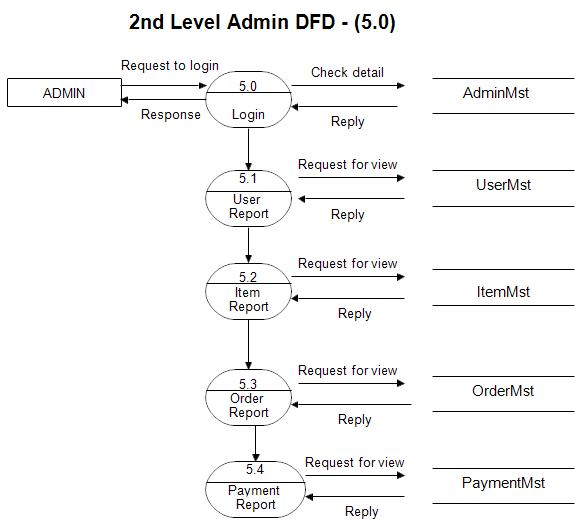
**Fig 5.11: 2nd Level DFD-Admin Side – Manage Product**

### **2nd Level – Admin side DFD (4.0) Manage Orders**



**Fig 5.12: 2nd Level DFD-Admin Side – Manage Orders**

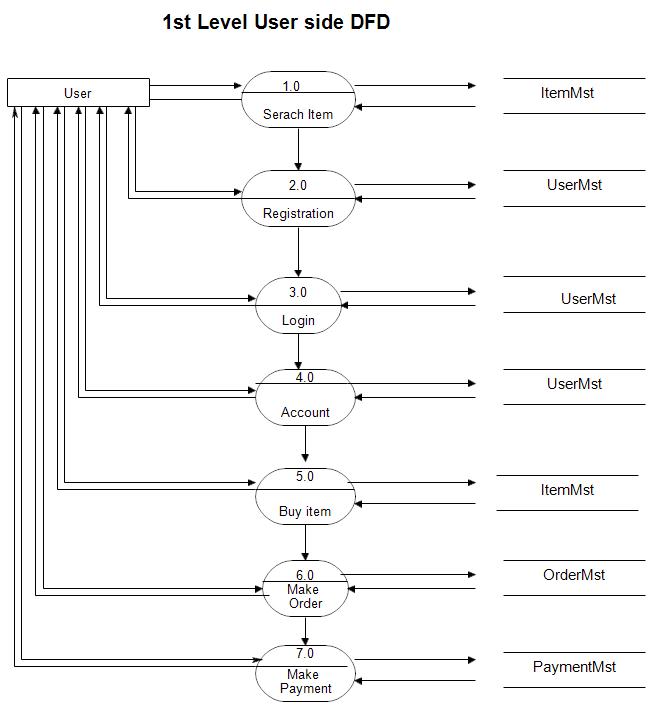
### **2nd Level – Admin side DFD (5.0) – View Reports**



**Fig 5.13: 2nd Level DFD-Admin Side – View Reports**

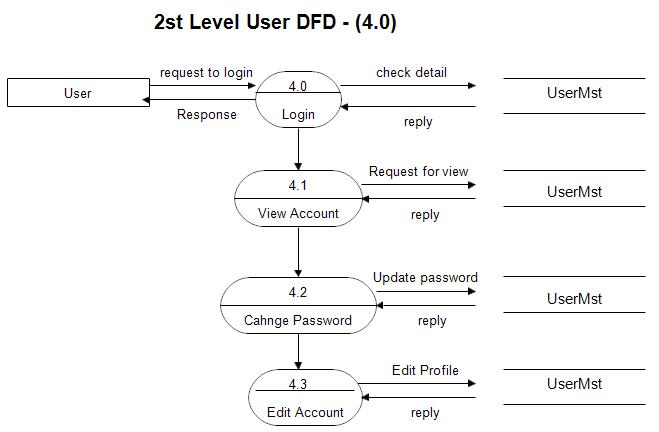
## **1st level – User side DFD**

The user is all people who operate or visit our web application. User is a customer of a web application. User can first select product for buy, user must have to register in our system for purchase any item from our web application. After register he can login to site and buy any electronic appliances



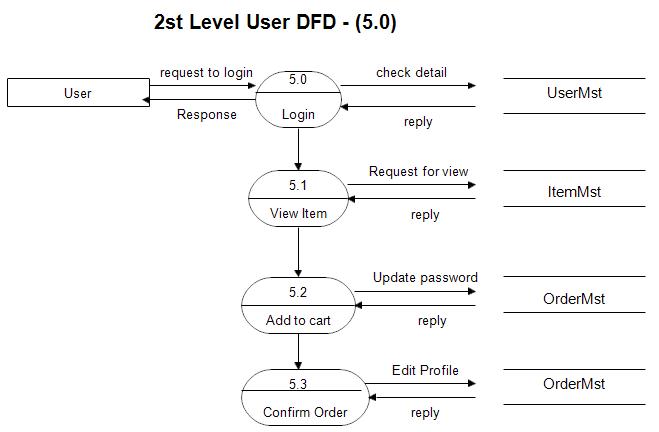
**Fig 5.14: 1st level User side DFD**

### **2nd level – User side DFD (4.0) – Manage Account Details**



**Fig 5.15: 2nd level User side DFD – Manage Account Details**

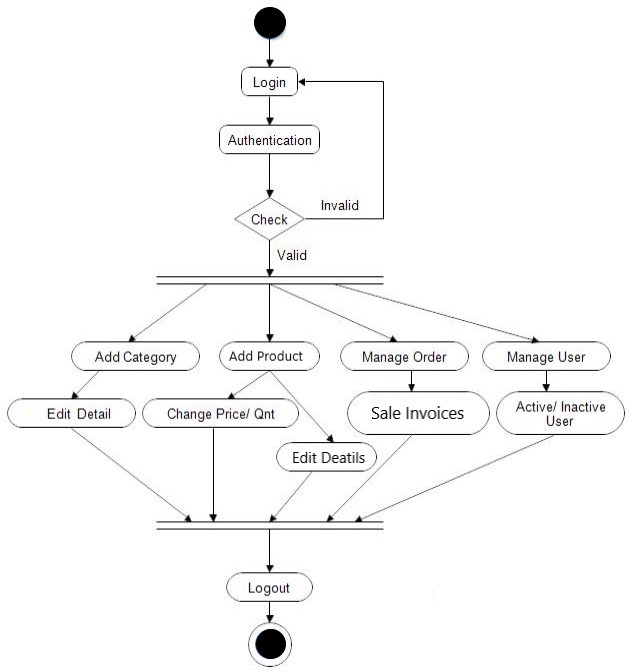
### **2nd level – User side DFD (5.0) - Purchase**



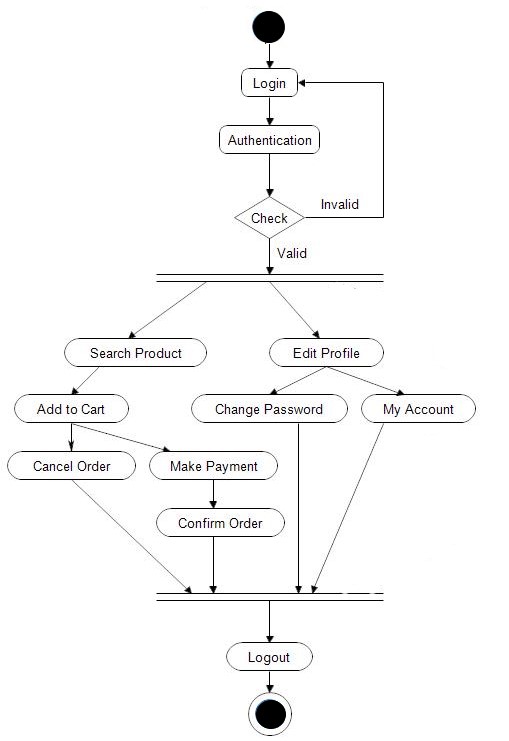
**Fig 5.16: 2nd level User side DFD - Purchase**

**5.4 Activity Diagram**

An activity diagram is a sort of UML diagram that depicts the sequence of activities and actions inside a system or process. An activity diagram may be used to demonstrate the many phases involved in the process of purchasing an item on an ecommerce web application for electrical appliances.



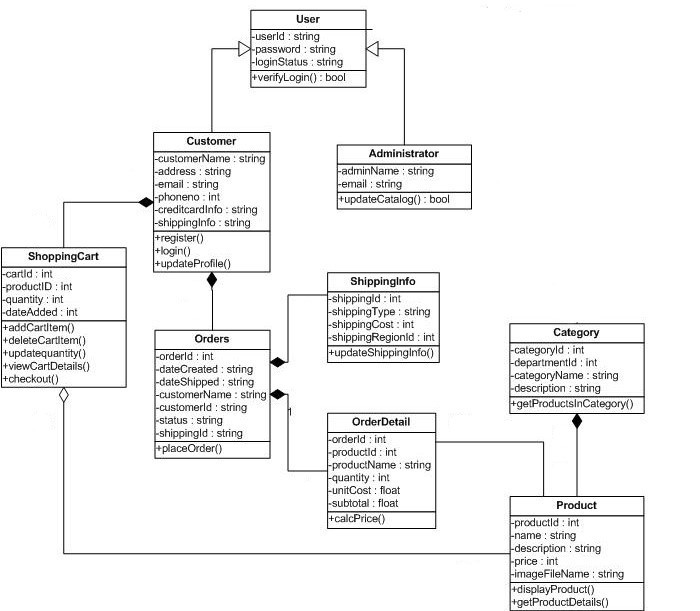
**Fig 5.17: Activity Diagram Admin Side**



**Fig 5.18: Activity Diagram Customer Side**

**5.5 Class Diagram**

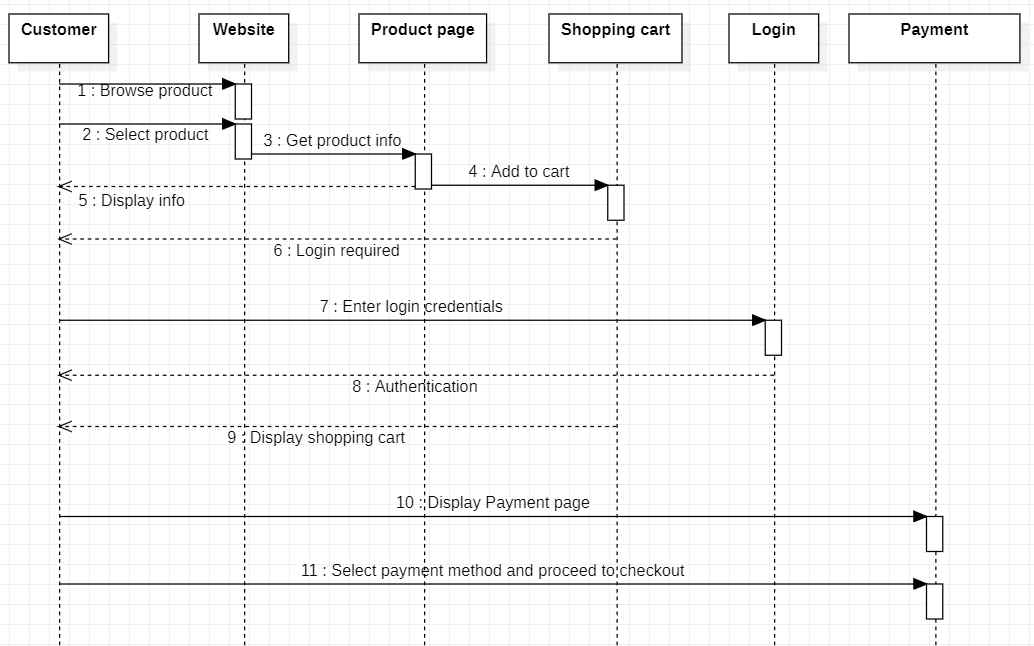
A class diagram is a type of UML diagram used to represent the static structure of a system, including the classes, attributes, methods, and relationships between objects. In the context of an ecommerce web application for electronic appliances, a class diagram can be used to illustrate the different classes and their relationships within the system. It is used to model the system's architecture, to plan its implementation, and to communicate its design.



**Fig 5.19: Class Diagram**

**5.6 Sequence Diagram**

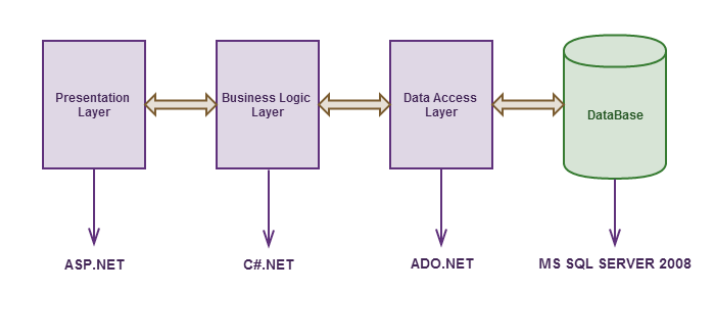
A sequence diagram depicts the interactions between objects on an ecommerce store for electronic appliances, making it easier to comprehend and manage the system over time. It depicts the flow of actions and events between the various objects and aids in the identification of possible issues or bottlenecks in the system. The sequence diagram shows the interactions between the User, Shopping Cart, Product, Payment, and Order objects in a series of steps. The diagram starts with the User logging in to the system and browsing the available products. The User selects a product and adds it to the shopping cart. The User then decides to remove the product from the shopping cart and updates the cart accordingly. Once the User has finished shopping, they proceed to checkout and initiate the payment process. The Payment object processes the payment and verifies it, and then creates a new Order object to record the purchase.



**Fig 5.20: Sequence Diagram**

**5.7 Software Architectural Design**

The software architectural design for an ecommerce web application will include a three-tier architecture with a presentation layer, business logic layer, and data storage layer. The business logic layer contains the essential functionality of the e-commerce web application, whereas the presentation layer is in charge of presenting the user interface and managing user input. Data utilized by the e-commerce web application must be stored and retrieved via the data storage layer. Designing the layers, selecting the right technologies, defining the interfaces, and ensuring security are all crucial when creating an e-commerce web application with a three-tier architecture.



**CHAPTER 6: DEVELOPMENT**

**6.1 Development**

**Coding**

Coding is a crucial stage involves translating the project requirements into code using a programming language. In this phase we create classes, functions, methods, and other components that will be used to build the software. The languages used in this project:

* **C#**

C# is a programming language that is commonly used in the development of web applications using the ASP.NET MVC framework. C# is used in ASP.NET MVC to write the code that makes up the controller and model components of the MVC pattern and define their behavior.

* **HTML & CSS**

HTML and CSS are the building blocks of web development. The structure and content of a web page are provided by HTML, while the display and layout are provided by CSS. They collaborate to develop aesthetically appealing and effective web applications that anybody with an internet connection and a web browser may view.

* **JavaScript**

JavaScript is a web development programming language that is used to give interactivity and dynamic response to web sites. It is an interpreted high-level language that runs in a web browser.

**6.2 Functional Requirements**

A set of requirements that specify what a software system or application must perform to suit the demands of its customers is known as functional requirements. These requirements define the functionality that the application must have in order to work properly. Functional requirements can include things like specific input and output processes, calculations or algorithms that need to be performed, data storage and retrieval mechanisms, and user interface design. Functional requirements are often defined in a software requirements specification (SRS) and serve as the foundation for software system design and development.

**Functional requirements are as follows:**

1. **Login**

There is only one login page both Admin and Customer login through using that

1. **Admin Dashboard**

* Admin can control whole system easily
* Admin can add/delete/edit brand, category, sub categories and product
* Admin can keep track of all the orders

1. **Customer Dashboard**

* Customer can keep track of all his/her orders
* Customer can edit his personal details

1. **Vouchers**

* Admin can generate discount vouchers for customers that can be used before expiry date

1. **Reviews**

* Customers can give reviews and rating on different products

1. **Reports Module**

* View Sales Reports
* Product Ledger
* Profit/Loss Report
* Stock Report

1. **Database**

* MS SQL Server

Microsoft SQL Server is a relational database management system (RDBMS) that was created by Microsoft. This product is designed to provide the fundamental function of storing and retrieving data as required by other applications. It can be executed on the same computer or on another machine over a network. This lesson discusses various fundamental and complex SQL Server principles, such as how to generate and recover data, create logins and backups, assign permissions, and so on.

* 1. **Non-Functional Requirements**

Non-functional requirements are characteristics and attributes of a software system that are unrelated to its exact functioning or features and instead focus on how the system performs, operates, and behaves. These requirements describe the system's quality and performance qualities, and they are often stated in quantitative terms.

**Non Functional requirements are as follows:**

* **Performance:** The web application should load fast and be able to manage large traffic levels without slowdowns or breakdowns.
* **Security:** The web application should be safe, with credit card information, personal information, and purchase history protected. Admin can only execute admin duties on pages to which they have access. Customers will be denied access to the admin pages.
* **Availability:** The system should remain operational in any day and any place.
* **Usability:** The web application should be simple to use, with distinct categories and product descriptions, as well as simple search and filtering features. It should also be mobile device friendly.
* **Compatibility:** The web application should be compatible with a range of browsers and devices.
* **Efficiency and Maintainability:** Page loads should be returned and formatted in a timely fashion depending on the request being made. Admin will have the ability to edit the aspects of product descriptions, prices and web application directly.

**6.4 Selection of tools and technology**

● Hardware

● Software

**Graphical User Interface**

* Login Page
* Home page
* Shopping Cart
* About Us Page
* Product Page
* Wish list Page

**6.5 System Specification**

System specification refers to the detailed description of the requirements, features, and characteristics of a computer system or software application. System specifications are crucial in ensuring that a system or application performs optimally and meets the needs of its users. They also serve as a guide for developers and system administrators to ensure that the system or application is developed and configured to meet the specified requirements. System specifications may vary depending on the type of system or application being developed.

**CHAPTER 7: SOFTWARE TESTING**

**Software Testing**

Software testing is the act of examining a software application to find flaws or problems that can affect its functioning, usability, or performance. It comprises testing the software in multiple scenarios and environments to make sure it lives up to expectations and requirements. Software testing makes ensuring the application is reliable, meets the intended purposes, and is of an exceptional quality. It supports usability and user experience, encourages the identification and rectification of issues or flaws, and enhances the overall performance of the product.

To analyze various software features, functional testing, non-functional testing, performance testing, security testing, usability testing, and other types of software testing approaches can all be utilized. Testers use a range of tools and methodologies to carry out these tests, including automated testing, manual testing, exploratory testing, and regression testing.

Software testing is an important component of the software development lifecycle (SDLC) since it ensures that the software is ready for deployment and use by end users. It also helps to limit the risk of any errors or flaws that might hurt the software's or the company's reputation.

**7.1** **Testing Process**

Software testing ensures that the final product meets the desired requirements. The purpose of software testing is to:

* To test functionality.
* To find bugs and faults.
* Identifying differences between present and expected circumstances.

There are various types of testing each designed to evaluate different aspects of a software application.

1. **Unit Testing:** Checking individual software parts or components to make sure they operate as intended is known as "unit testing." Developers often carry it out when working on a project.
2. **Integration Testing:** To make sure that a software application's different components perform properly when combined, integration testing entails examining how each component interacts with the others.
3. **System Testing:** System testing involves putting the entire system through its tests to make sure it meets with all specifications and that each component functions as it should.
4. **Usability Testing:** Usability testing determines any potential usability problems or hurdles and evaluates how user-friendly and intuitive the software application is.
5. **Performance Testing:** Performance testing is carried out to assess how well a software performs under various loads or situations.
6. **Compatibility Testing:** To make sure that a web application works properly on a variety of platforms and settings, it must be examined for compatibility with various browsers, mobile devices, and operating systems.

**7.2 Test Case Design**

Test case design is the process of generating and documenting a set of test cases that will be used to validate the functionality of a software application. The design of test cases is an important aspect of the software testing process since it guarantees that all of the software's features are evaluated and that the product satisfies its requirements.

**The following are some of the key steps involved in test case design:**

* Define the testing objectives
* Identify test scenarios
* Determine test inputs
* Identify expected results
* Create test cases
* Review and validate test cases
* Execute test cases
* Analyze results

**TEST CASES:**

A test case is a document containing a set of test data, preconditions, expected results, and post conditions, designed to be used in a specific test scenario to verify compliance with a specific requirement. A test case acts as a starting point for test execution, and after applying a set of input values, the application gets a final result and exits the system at an endpoint, also known as executing a post condition.

**Test cases typically include the following components:**

**Test Case ID:** A unique identifier for the test case.

**Test Case Description:** A brief description of the test scenario.

**Test Steps:** The specific steps that must be taken to run the test scenario.

**Test Data:** The input value or data to use when testing.

**Expected Result:** The expected output or outcome of the test scenario.

**Actual Result:** The actual output or result of the test scenario.

**Pass/Fail Status:** Whether the test scenario passed or failed.

* 1. **Test Data**

**Login Object Table**

The following test case table shows login entries.

**Table 7.1 Login Object Table**

|  |  |  |
| --- | --- | --- |
| Sr. no | Object name | Object type |
| 1 | Name | Text Field |
| 2 | Email | Text Field |
| 3 | Password | Password Field |
| 4 | Type | Radio Button |
| 5 | Login | Button |

**Valid Input Data Table**

The following test case table describes valid input for login.

**Table 7.2 Valid Input Data Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Test Data** | **Expected Result** | **Actual Result** | **Status** |
| **1** | Check Admin Login with valid Data | **Email**  [haqmultan@gmail.com](mailto:haqmultan@gmail.com)  **Password**  1234  **Type:** Admin | Admin  Dashboard | As expected | Pass |
| **2** | Check Customer Login with valid Data | **Email**  [maria2k19bscs@gmail.com](mailto:haqmultan@gmail.com)  **Password**  5678  **Type:** Customer | Homepage | As expected | Pass |

**Table 7.3 Invalid Input Data Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Test Data** | **Expected Result** | **Actual Result** | **Status** |
| **3** | Check Admin Login with Invalid Data | **Email**  [hamid@gmail.com](mailto:hamid@gmail.com)  **Password**  14578  **Type:** Admin | Login | As expected | Pass |
| **4** | Check Customer Login with Invalid Data | **Email**  [maria966865@gmail.com](mailto:maria966865@gmail.com)  **Password**  5678mm  **Type:** Customer | Login | As expected | Pass |

**Add new Sub Category Object Table**

The following test case table shows login entries.

**Table 7.4 Add new Sub Category Object Table**

|  |  |  |
| --- | --- | --- |
| Sr. no | Object name | Object type |
| 1 | Name | Text Field |
| 2 | Category id | Dropdown |
| 3 | Save | Button |

**Valid Input Data Table**

The following test case table describes valid input for login.

**Table 7.5 Valid Input Data Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Test Data** | **Expected Result** | **Actual Result** | **Status** |
| **5** | Add sub category data with valid input in database | **Name:**  Wall mount  **Category id:**  22 | Saved in Database | As expected | Pass |

**Add new Product Object Table**

The following test case table shows login entries.

**Table 7.6 Add new Product Object Table**

|  |  |  |
| --- | --- | --- |
| Sr. no | Object name | Object type |
| 1 | Name | Text Field |
| 2 | Stock level | Text Field |
| 3 | Subcategory id | Dropdown |
| 4 | Description | Text Field |
| 5 | Price | Text Field |
| 6 | Image | Image Box |
| 7 | Save | Button |

**Valid Input Data Table**

The following test case table describes valid input for login.

**Table 7.7 Valid Input Data Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Test Data** | **Expected Result** | **Actual Result** | **Status** |
| **6** | Add new product data with valid input in database | **Name:**  Chrome 1.5 Ton White Inverter Split AC  **Stock level:**  8  **Subcategory id:**  22  **Description:**  Ac 1.5 ton cooling effect  **Price:**  118.77  **Image:** | Saved in Database | As expected | Pass |

# Black Box Testing

Black box testing is a software testing technique that focuses on testing the functionality of an application without understanding its inner workings. It is based on the principle that the tester has no knowledge of the code, architecture or design of the application under test.

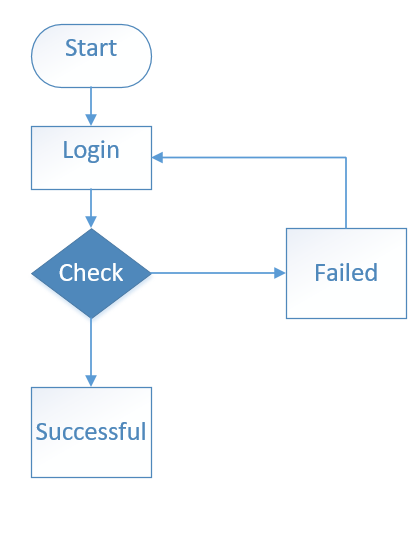
In black box testing, the tester evaluates the behavior of the application by providing input and examining the output without knowing how the application processes the input to generate the output. The goal is to determine whether applications meet their functional requirements, such as B. their ability to process input, produce expected output, and respond correctly to different types of user interactions.

Black box testing is usually done at the conclusion of the software development life cycle (SDLC) to confirm that the software works as planned and to discover any faults or errors. It is also handy for testing apps produced by third-party suppliers or applications where the source code is unavailable.

**Flow of Login**

The process begins with login. The customer/admin inputs a valid email address and password. The credentials are validated against the database. The login is valid if the username and password are correct; otherwise, an incorrect login attempt occurs. The customer/admin must log in again.

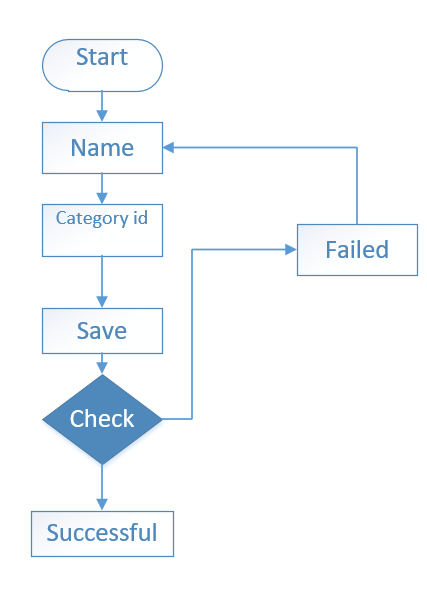
**Figure 7.1 Flow of Login**



**Flow of Add new Sub Category**

First of all, enter the name of the sub category followed by the choosing category id from dropdown. After pressing the “Save” button if the sub category is added into the database record the test is successful otherwise, add again.

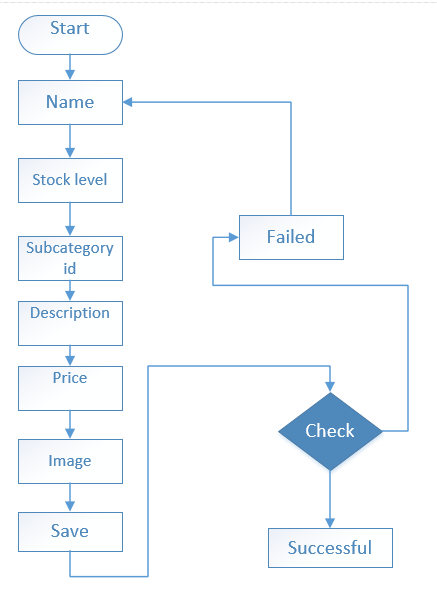
**Figure 7.2 Flow of add new sub category**



**Flow of Add new Product**

Enter the name that is first, followed by the description, price, image, stock level, and subcategory id. If the product is added to the products record after pressing the "Save" button, the test is successful; otherwise, add the product again.

**Figure 7.3 Flow of add new product**



* 1. **White Box Testing**

White box testing creates test cases based on the inner workings of the software. These test cases are designed to test specific areas of the software, such as individual functions or modules, to ensure they work as expected. Some of the techniques used in white box testing include code coverage analysis, branch coverage analysis, and path coverage analysis. These techniques include analyzing code to identify areas not covered by tests and ensuring that all code paths are tested. White-box testing is often used in conjunction with black-box testing, which focuses on the external behavior of the software. Together, these testing techniques can help ensure that software applications and systems work as expected and are free of bugs and defects.

**CHAPTER 8: IMPLEMENTATION AND TRAINING**

**8.1 Implementation**

The process of turning a concept or plan into a working software system or application is known in software development as implementation. It involves developing code, configuring software components, integrating different modules, and testing software to make sure it works properly. Implementation is an important phase in the software development life cycle (SDLC) because it includes the actual coding and development. It occurs after the SDLC planning, design and analysis phases and before the testing and maintenance phases. During implementation, developers work closely with software design specifications, which detail the behavior, functionality, and requirements of the system. To generate code and build software systems, they use programming languages, development frameworks, and tools.

**Project Overview**

This ecommerce project will have following features:

**Product Catalog:** The web application will have a product catalog listing all the items offered. Customers can browse the products and select the ones they want to buy.

**Shopping Cart:** When a customer selects a product, that product will be added to their shopping cart. The shopping cart keeps track of all the products the customer wants to buy and calculates the total cost.

**Checkout:** Customers can checkout after pickup. Shipping and billing information must be entered and a payment method selected during the ordering process.

**User Accounts:** Customers can create user accounts to store financial and personal data and view order history on their dashboard.

**Admin Panel:** The web application will have an admin panel where site administrators can manage products and manage orders.

**Search feature:** Customers can search for items on the web application using keywords, categories or brands.

**Product Reviews and Ratings:** Customers can leave product reviews and ratings on the site, which can help other buyers make more informed purchasing decisions.

**CONCLUSION**

In conclusion, this e-commerce web application provides users wishing to buy electronic equipment online with a streamlined and practical buying experience. We provide a wide range of premium products from reliable manufacturers, and our user-friendly interface enables customers to browse, compare, and buy electrical equipment with confidence. Our broad assortment of electronic products from reputable companies is one of our main advantages.

Our web application places a high priority on security, making sure that user data is protected and transactions are carried out safely. We also offer dependable shipping and attentive customer service to ensure a wonderful experience after the sale.

Our ecommerce web application for electrical appliances is the go-to place for clients looking for a quick and dependable online shopping experience, with an emphasis on customer satisfaction, quality items, and user-friendly features. Shop at EleCommerce to enjoy the convenience and dependability of online purchasing.

**References**

* <https://www.reqview.com/doc/volere-template/>
* <https://www.geeksforgeeks.org/software-engineering-iterative-waterfall-model/>
* <https://www.guru99.com/test-case.html>
* <https://www.guru99.com/functional-vs-non-functional-requirements.html>
* <https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document#:~:text=A%20software%20requirements%20specification%20(SRS)%20is%20a%20document%20that%20describes,stakeholders%20(business%2C%20users)>.
* <https://images.template.net/wp-content/uploads/2016/07/26121721/Online-Shopping-Project-Documentation-Template.pdf>
* <https://www.techtarget.com/whatis/definition/software-testing>
* <https://www.spiceworks.com/tech/devops/articles/black-box-vs-white-box-testing/>
* <https://creately.com/blog/diagrams/uml-diagram-types-examples/>
* <https://opus.govst.edu/cgi/viewcontent.cgi?article=1079&context=capstones>