**Project Proposal**

**On**

**E-commerce**

****

**Admond Tamang**

**00172885**

**Computing Project**

**Level 5 Diploma in Computing**

**Softwarica College of IT and E-Commerce**

**Kathmandu, Nepal**

**2018/12/28**

**Submitted to: Kiran Rana**

**Abstract**

In this modern world E-commerce is well known place where you can purchase, sells and exchange your goods over internet via online mechanism. So, it can be described as a site which allows transaction of purchasing and selling online.

The BigHeart is an online e-commerce site which is developed for online transaction of goods and promoting local as well as industrial brands. This website will allow them in cheaper marketing, online trading and improving productivities. Likewise, customers will have direct easy, fast and reliable purchasing functionality. Which will further help in preventing unnecessary time and costs.

Web project was developed using 3-tire architecture approach and LARAVEL and MySQL was used as developing language and database respectively.

**Acknowledgement**

The BigHeart is a project which is completed not just by me but with efforts of group of people who understands and helped me in any obstacles that comes on my way in order to complete this project. And, I feel that it’s my great honor to thank those helping hands, who helped me in successful completion of my project.

I would like to express my sincere gratitude to **Softwarica College of IT and e-commerce**, for giving the opportunity to submit this e-commerce project.

I would like to thank **Mr. Kiran Rana**, our project advisor, for guiding us your support and advise in every steps of the process.

Lately, I would like to thank my colleagues' friends for helping me throughout research process whenever in need.

Contents

[Introduction 5](#_Toc5651950)

[Problem background to the system 5](#_Toc5651951)

[Justification 5](#_Toc5651952)

[Aim 5](#_Toc5651953)

[Objective 6](#_Toc5651954)

[Overview of the project design 6](#_Toc5651955)

[Analysis 9](#_Toc5651956)

[Introduction to analysis 9](#_Toc5651957)

[Object oriented analysis 9](#_Toc5651958)

[Merits of your project 9](#_Toc5651959)

[Pitfalls 9](#_Toc5651960)

[Requirements 10](#_Toc5651961)

[Functional requirement 10](#_Toc5651962)

[Non-Functional requirement 10](#_Toc5651963)

[MoSCoW prioritization 10](#_Toc5651964)

[NLA 11](#_Toc5651965)

[Initial class diagram 11](#_Toc5651966)

[Use case 12](#_Toc5651967)

[Architecture 15](#_Toc5651968)

[Design 16](#_Toc5651969)

[Justification(main) 16](#_Toc5651970)

[Dynamic modelling 16](#_Toc5651971)

[Sequence diagram 16](#_Toc5651972)

[Activity diagram 17](#_Toc5651973)

[Structural modelling 20](#_Toc5651974)

[DFD or Context diagram 20](#_Toc5651975)

[Database modelling 20](#_Toc5651976)

[ER Diagram: 20](#_Toc5651977)

[UI modelling (prototype) 23](#_Toc5651978)

[Implementation 28](#_Toc5651979)

[Testing 28](#_Toc5651980)

[Black box testing 28](#_Toc5651981)

[Unit Testing. 28](#_Toc5651982)

[Other project issues 29](#_Toc5651983)

[Conclusion 29](#_Toc5651984)

[References and bibliography 29](#_Toc5651985)

[References 29](#_Toc5651986)

[Appendix 29](#_Toc5651987)

**Tables of figure**

[Figure 1: Home Page 8](#_Toc5651988)

[Figure 2: Login page 8](#_Toc5651989)

[Figure 3: Add to cart 9](#_Toc5651990)

[Figure 4: Check out 9](#_Toc5651991)

[Figure 5: Initial Class diagram 13](#_Toc5651992)

[Figure 6: Use case diagram of e-commerce 15](#_Toc5651993)

[Figure 7: Three tier architecture 16](#_Toc5651994)

[Figure 8: Sequence diagram of user placing an order 18](#_Toc5651995)

[Figure 9: Activity diagram of ordering product 19](#_Toc5651996)

[Figure 10: Edit user profile 20](#_Toc5651997)

[Figure 11: Adding product 21](#_Toc5651998)

[Figure 12: ER - diagram 22](#_Toc5651999)

**Keywords**

DFD - Data Flow Diagram

E-commerce - Electronic commerce

ERD - Entity Relationship Diagram

NLA - Natural Language Analysis

UX – User Experience

WAMP – Window Apache MySQL Php

# Introduction

Modern and most trending sites for transaction like buying and selling goods and services over internet is E-commerce site. E-commerce doesn’t hold barrier between client and business but acts as a bridge which will helps them connect and processed business activity in easy and effective (cost and time) way. Functionality which supports online transaction over internet directly attracts customers and it became more and more common now a days.

The primary objective of this project is to develop a user-friendly e-commerce store where any kind of products and services can be bought or sell through the internet.

## Problem background to the system

In our country, customers are not very aware of products that is currently in local market all because of their busy life and stressful shopping process. Similarly, those local brands lack in knowledge of effective way of marketing their products. Which is the major reason that creates a distance between Nepalis customers and local businesses. Also, availabe products are came across different vendors and routes which makes products to be costly and time captivating.

So, I came up with a solution (BigHeart E-commerce site) which will break this barrier and reduces all buying and selling process and helps in growth of productivity.

## Justification

Now a days, peoples have not enough spare time for personal needs as they are chasing over their goal of their life. But, technology kicks in here and offers facilities that makes life easier. Online transaction specially shopping is one of them. Which mean you can buy and sell your products online and saves a lot of time and money from it just that easily. BigHeart is also a solution that I came up with and support in online transaction facility reliable, secure and effective way possible.

This E-commerce site includes authentication facility (restrict unregistered customers) which is made possible with registration and login process, products and stocks are maintained by admins of website and customer can view product details and make their orders, queries and other transaction processes.

## Aim

The primary aims that I want to achieve are listed below:

* Developers should permit for placing goods and services
* Customer should permit easily interact with the website and buy websites according to their needs.

## Objective

The main objective is to satisfy the local manufacture and customer. Action that I will take to achieve my aim are:

* Customer will get a registration facility where they can set up their accounts and browse content of website easily.
* Developers will have the facility to post their website and provide services to the customer.

## Overview of the project design

In today's competitive market there are lots of product available but none of those products are local product which is sad to see. My project helps to eliminate those problems by providing market to the local product which they aren't getting in the current context.

This application was built the problem of high price that local buyer is facing. Local product owner can add their product to our web site which will make the price of the product comparatively low that of imported products. Only the low price is not the advantageous thing, buyer can get authentic Nepali product which is of high-quality materials. Ordering products from this website will save the time for buyers.

I myself feel blessed to provide the test of local product to local people and feel great that I have taken the burden to provide market for local business man.

# Analysis

## Introduction to analysis

The e-commerce system is not an easy process to set up as it asks for warehouses, customers to browse, and administrator works of maintaining products and their details. But this project is mainly focused upon local production businesses and marketing.

Browsing product information and placing their order online is primary task that I want to achieve from this project where the database will maintain user and product details.

This system involves three types of users:

1. **Customer**

The customer is the person who buys products from the website.

1. **Store/Warehouse**

The store is the shop where customer can buy the products.

1. **Admin**

Admin administrate the behavior of both customer and store.

## Object oriented analysis

It is an iterative stage of analysis, which takes place during the software Development life cycle. Object-oriented analysis is a process that groups items that interact with one another, typically by class, data or behavior, to create a model that accurately represents the intended purpose of the system. Object-oriented analysis does not factor implementation limitations into the model. (Anon., 2019)

## Merits of your project

* User experience (UX) is simple and interactive which focused on compatibility and usability with users.
* Sellers does not have to rent a physical store; hence it saves both time and money.

## Pitfalls

* Not having online payment functionality.
* Solution is focused on local manufactures so it can’t go outside country.

## Requirements

### MoSCoW prioritization

MoSCoW is a prioritization technique for categorizing list of requirements. I have prioritized my features using MoSCoW prioritization technique because it helps to rank and classify requirement to achieve a successful product.

**Note**

**M**=must have - It describes that the requirement must include in the final solution. I should cover 60% of requirements.

**S**=should have - The priority is higher but not critical. Such requirement is priorities in second place in priority list. Covers around 20% of tasks.

**C**=could have – This type of requirement is desirable but no necessary. This covers 20% of requirement.

**W**=won’t have – Requirement is not implemented in current release but may include in future development.

### Functional requirement

Those requirements which are related to behavioral aspect of the system. It specifies what the system should do. It includes business rule, administrative function and transaction correction.

|  |  |  |  |
| --- | --- | --- | --- |
| **S/N** | **Requirement** | **MoSCoW** | **Dependencies** |
|  | Login | M | NF1 |
|  | Register | M | NF1 |
|  | Post product | M | NF2 |
|  | Add to Wishlist | C |  |
|  | Search product | M |  |
|  | Filter product | S |  |
|  | View user profile | M |  |
|  | Edit profile | M |  |
|  | Chat with owner | S |  |
|  | Buy product | M |  |
|  | Rate product | C |  |
|  | View product | M |  |
|  | Comment product | C |  |
|  | Generate bills | M |  |
|  | Verify store owner/ seller | M |  |
|  | Delete user | S |  |

### Non-Functional requirement

These requirements specify how the system should behave and the limits on requirements.

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Requirement** | **Description** |
| NF1 | Verification | To verify customer and store. |
| NF2 | Security | Protect user and system from hacking. |
| NF3 | User friendly | To provide smooth experience. |
| NF4 | Efficient | Make system fast and reliable. |
| NF5 | Availability | Should be available 24 hours. |
| NF6 | Scalability | Handle load of the system |
| NF7 | Maintainability | System should be monitored and documentation must be available. |

### **Justification**

## NLA

Natural Language Analysis is a process that helps in identification of candidate classes, their attributes and relationship with each other.

A BIGHEART is an IT company that focuses in buying and selling products online. The main objective is to promote local market.

The list the of candidate classes and candidate operations are:

|  |  |
| --- | --- |
| **Candidate classes** | **Candidate operations** |
| Admin | Update password, add staff, update staff, delete staff  View staff and view student |
| Store | Add product, Delivery, delete product, update product |
| User | edit profile, update profile |
| Order | Order product. |
| Product | Add product |
| Category | Add category |
| Shipping | Add shipping |
| Cart | Create, store, delete product |

## 

## Initial class diagram

**Class Diagram**

A structural diagram which describes classes and their relationships, providing static model and design of system itself is known as class diagram. It provides information of classes and its attributes, operations, associations and constraints between them. (Anon., n.d.)

**Class**

Class is a blueprint of objects of OOAD system. In UML notation it contains three parts. Class Name, class attributes and class operations

**Class Notation**

A class notation consists of three parts:

1. Class Name
   * Represent class name and placed at top section.
2. Class Attributes
   * It contains attributes of that class
   * Present at middle section of class.
3. Class Operations
   * It describes behavior of a class.
   * Present at middle section of class

**Initial class diagram**

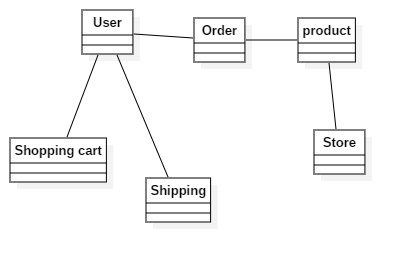
****

Figure 5: Initial Class diagram

x

**Justification of Initial class diagram**

I designed class diagram for my project because it helps to describes the responsibilities of the system which allows me to understand working mechanism of the system. As it holds class, attribute and operations which is required for the development of application. (Anon., n.d.)

## Use case

Represents the functionality of a system. Also, describes how user interact with a system. So, it is a diagrammatic representation of a user's interaction with the system that shows the interaction between the user and their respective use cases. Use case if developed using different specific symbols. Actors, use case and interactions are represented with help of stick figure, over shaped box and straight line respectively.

Construction of use case for BigHeart is made possible with help of requirement gathering processes and MoSCoW prioritization.

**Use case notation**

Actors: Users that interacts with the system.

System: That consist different use cases.

Use case: action that is performed by actor.

**Advantages:**

* Identification of functional requirements of a solution.
* Planning and scheduling of functionality development.
* Bigger picture of system yet simpler way

**Disadvantages:**

* Non-functional requirements don’t show here.
* Lacks in ordering rules doesn’t helps in understanding work flow of system

I have used use case because it helps to represent action performed by actor via use-case.

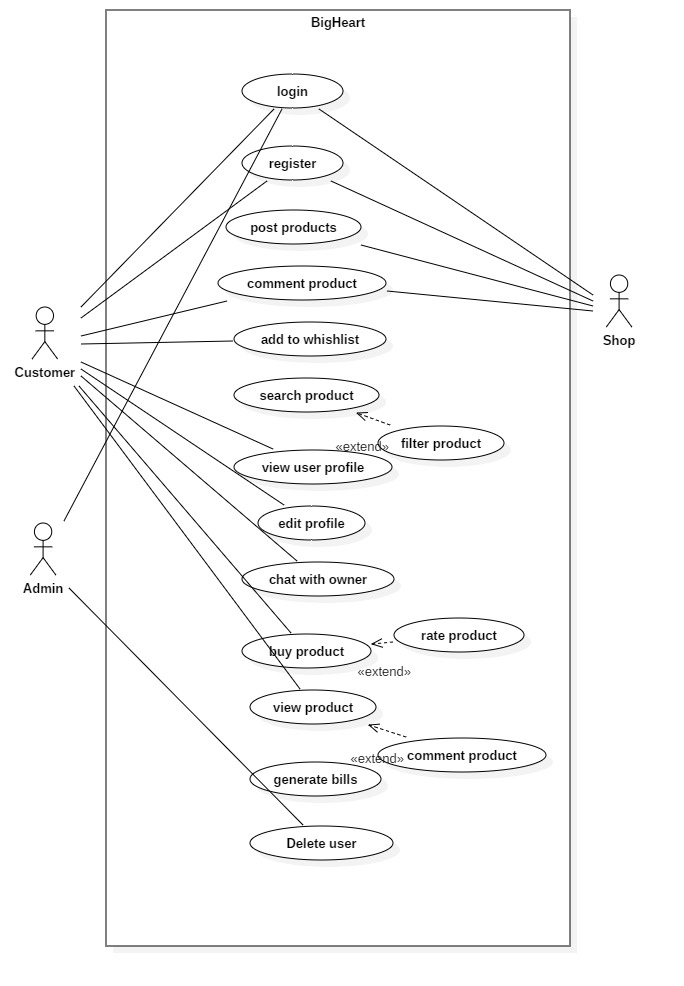


Figure 6: Use case diagram of e-commerce

## Architecture

3-tier architecture was used in the development of the project. It is divided into three layers. They are:

1. **Presentation**

Handles the interaction between user and the client business. For example, HTML5, CSS.

1. **Application**

Takes request from presentation tier and returns the output to presentation tier. For example, Java, .NET.

1. **Data tier**

Responsible for storing data and sending it to business tier. For example, MySQL, Oracle.

The reason for using 3 tier architecture are as follow:

1. **Scalability**

Middle tier can be added to make system run smoother. The system can work even hardware needs to be added and updated to load load-balance the presentation tier.

1. **Security**

Client is not directly access to the database. Middle layer protects the database tier ensuring strong security.

1. **Performance**

Presentation tier can cache requests, so network utilization is minimized and run smoothly.

1. **Maintainability**

It manages presentation code and business logic separately so change in business logic does not affect presentation layer.

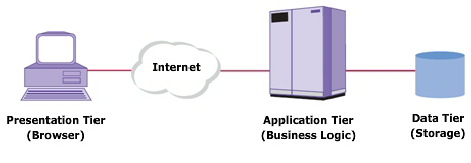


Figure 7: Three tier architecture

# Design

One of the most important phase of software development is Design where structures of project is developed, which further helps in solution development. It is done after analysis steps. it is also the highest critical stage in development of the system development where business need gets built.

Flowchart, Activity Diagram, sequence diagram, DFD are design model which are constructed here.

## Dynamic modelling

### Sequence diagram

A sequence diagram describes interaction between objects in a sequential order. Sequence diagrams describe how and in what order the objects in a system function.

So, following sequence diagram is constructed for identifying sequential work flow of Bigheart and another reason includes:

* Model the logic of a sophisticated procedure, function, or operation.
* See how objects and components interact with each other to complete a process.
* Plan and understand the detailed functionality of an existing or future scenario.

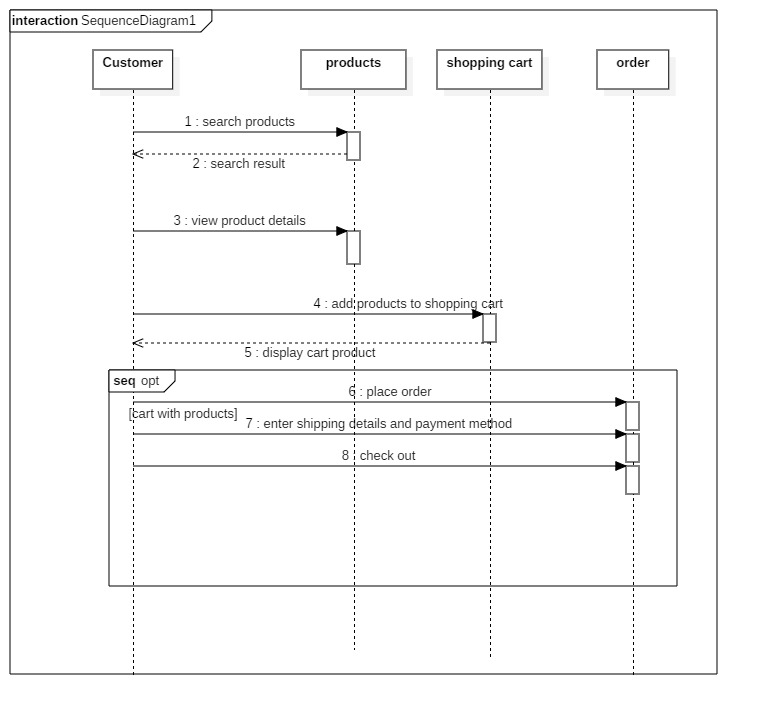


Figure 8: Sequence diagram of user placing an order

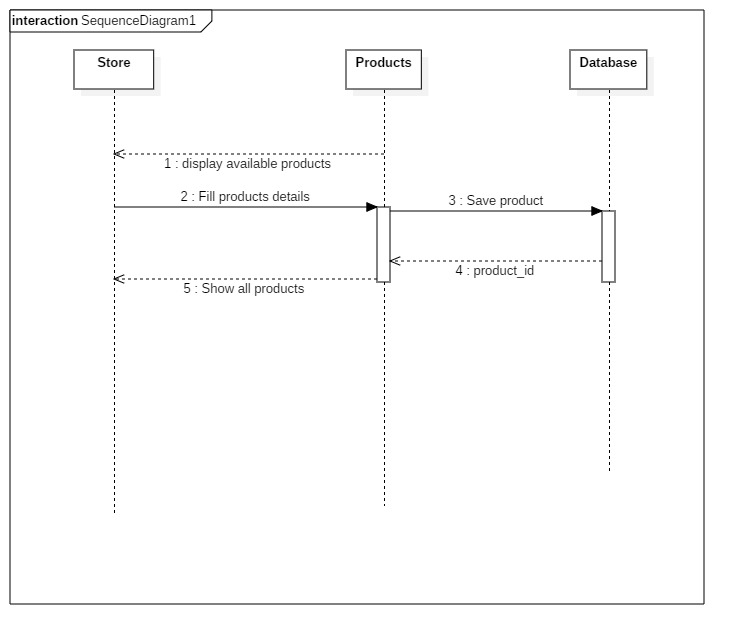


Figure 9: Sequence diagram to add a product



Figure 10: Sequence diagram to authenticate user

### Activity diagram

They ae similar to flow charts, which illustrate the flow of system. An activity diagram focuses on condition of flow and the sequence of specific event. It supports to comprehend the flow of programs on a high level. Similarly, enables to figure out constraints and conditions that cause events.

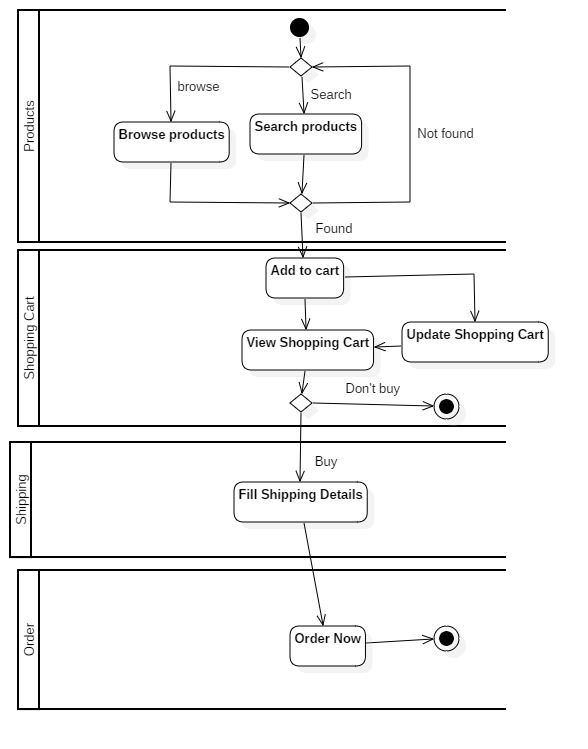


Figure 11: Activity diagram of ordering product

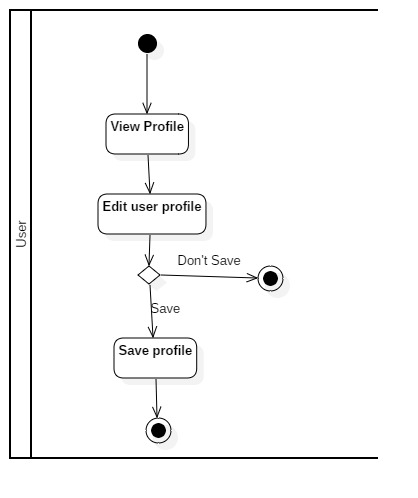


Figure 12: Edit user profile

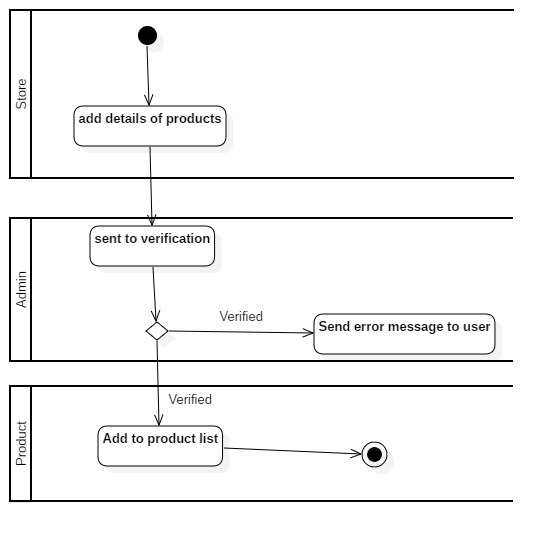


Figure 13: Adding product

I created activity diagram because it helps to show the flow operation in a system.

## Structural modelling

Architecture level class diagram (final class diagram)

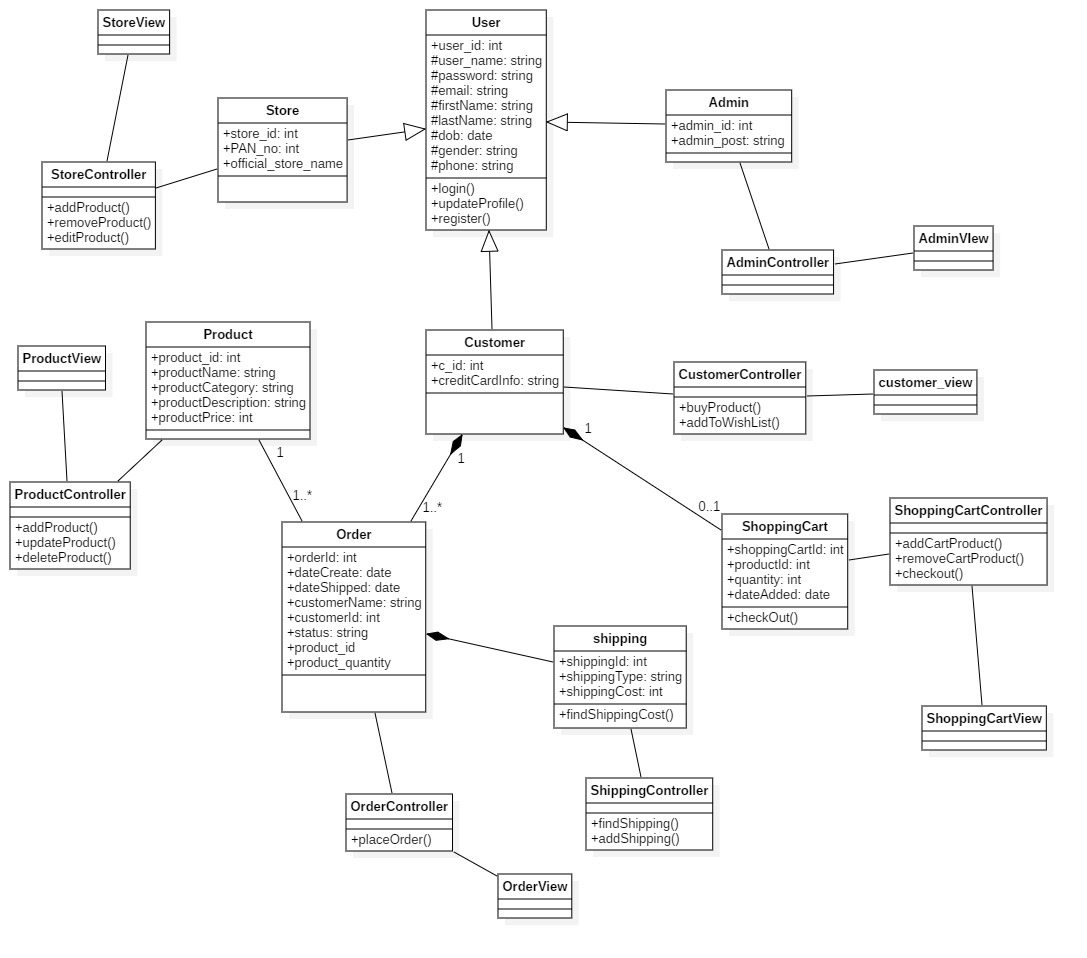
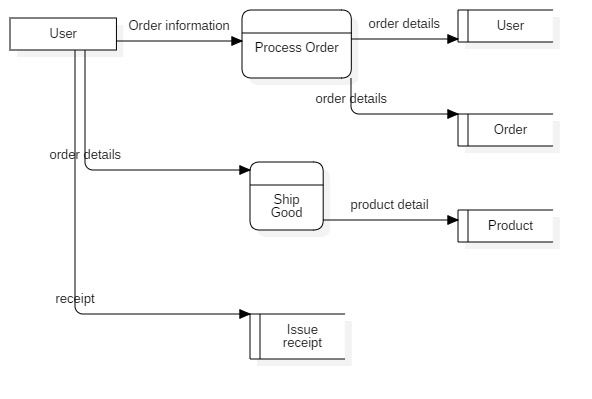


Figure 14: Final class diagram

## DFD or Context diagram

Diagram which illustrate any process or flow of data through a system is known as DFD or Context diagram. It shows the flow of information in a process based on the inputs and outputs. A DFD also referred to as a Process Model.



## Database modelling

### ER Diagram:

An ER Diagram shows the entities and relations between each other in a given system in a graphical way. It shows how data of system gets stored in a database and relationships among them. So, it’s a conceptual model representing entity framework infrastructure.

**Justifications:**

In the ER diagram provided below, I have added all the entities and attributes that may be used as database table and columns in BigHeart’s database with respective attributes and datatypes and contrsaints.

**Advantages:**

* Simple and easy way to understanding entities and their relationships
* graphical representation.
* Works as an communication tools for Designers.

**Disadvantages:**

* Constraints are complex
* Not all information gets represented.

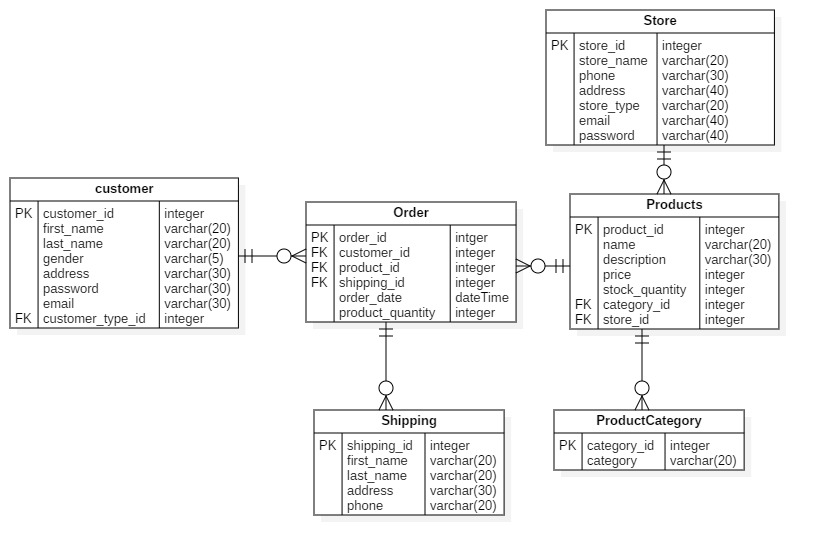


Figure 15: ER - diagram

### Data dictionary

A data dictionary is a file or a set of files that contains a database's metadata. The data dictionary contains records about other objects in the database, such as data ownership, data relationships to other objects, and other data.

**Table: Users**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column name** | **Data type** | **Length** | **Nullable** | **Key** | **Constraint** |
| **Id** | Integer | 10 | No | Primary key | PK\_id |
| **Name** | Varchar | 101 | Yes | - | - |
| **Phone** | Varchar | 101 | Yes | - | - |
| **Email** | Varchar | 101 | No | - | - |
| **Password** | Varchar | 101 | No | - | - |
| **Address** | Varchar | 101 | No | - | - |

**Table: Stores**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column name** | **Data type** | **Length** | **Nullable** | **Key** | **Constraint** |
| **Id** | Integer | 10 | No | Primary key | PK\_id |
| **Name** | Varchar | 101 | Yes | - | - |
| **Phone** | Varchar | 101 | Yes | - | - |
| **Email** | Varchar | 101 | No | - | - |
| **Password** | Varchar | 101 | No | - | - |
| **Address** | Varchar | 101 | No | - | - |

**Table: Products**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column name** | **Data type** | **Length** | **Nullable** | **Key** | **Constraint** |
| **Id** | Integer | 10 | No | Primary key | PK\_id |
| **Name** | Varchar | 101 | Yes | - | - |
| **description** | Varchar | 101 | Yes | - | - |
| **price** | integer | 10 | Yes | - | - |
| **Stock\_quantity** | integer | 10 | Yes | - | - |
| **Category\_id** | integer | 10 | No | Foreign key | Fk\_id |

**Table: Shippings**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column name** | **Data type** | **Length** | **Nullable** | **Key** | **Constraint** |
| **Id** | Integer | 10 | No | Primary key | PK\_id |
| **Name** | Varchar | 101 | Yes | - | - |
| **address** | Varchar | 101 | Yes | - | - |
| **Postal\_code** | integer | 10 | No | - | - |
| **user\_id** | integer | 10 | No | Foreign key | Fk\_id |
| **Email** | Varchar | 191 | No | - | - |

**Table: Order**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column name** | **Data type** | **Length** | **Nullable** | **Key** | **Constraint** |
| **Id** | Integer | 10 | No | Primary key | PK\_id |
| **User\_id** | Varchar | 191 | Yes | Foreign key | FK\_user\_id |
| **Product\_id** | Varchar | 191 | Yes | Foreign key | FK\_product\_id |
| **Order\_date** | datetime | - | No | - | - |
| **Name** | integer | 10 | No | - | - |
| **Email** | varchar | 191 | No | - | - |
| **Address** | varchar | 191 | No | - | - |
| **Postal\_code** | integer | 10 | No | - | - |
| **Phone** | varchar | 191 | No | - | - |
| **Product\_quantity** | integer | 10 | No | - | - |
| **Status** | boolean | 1 | No | - | - |
| **payment** | varchar | 191 | No | - | - |

**Table: categories**

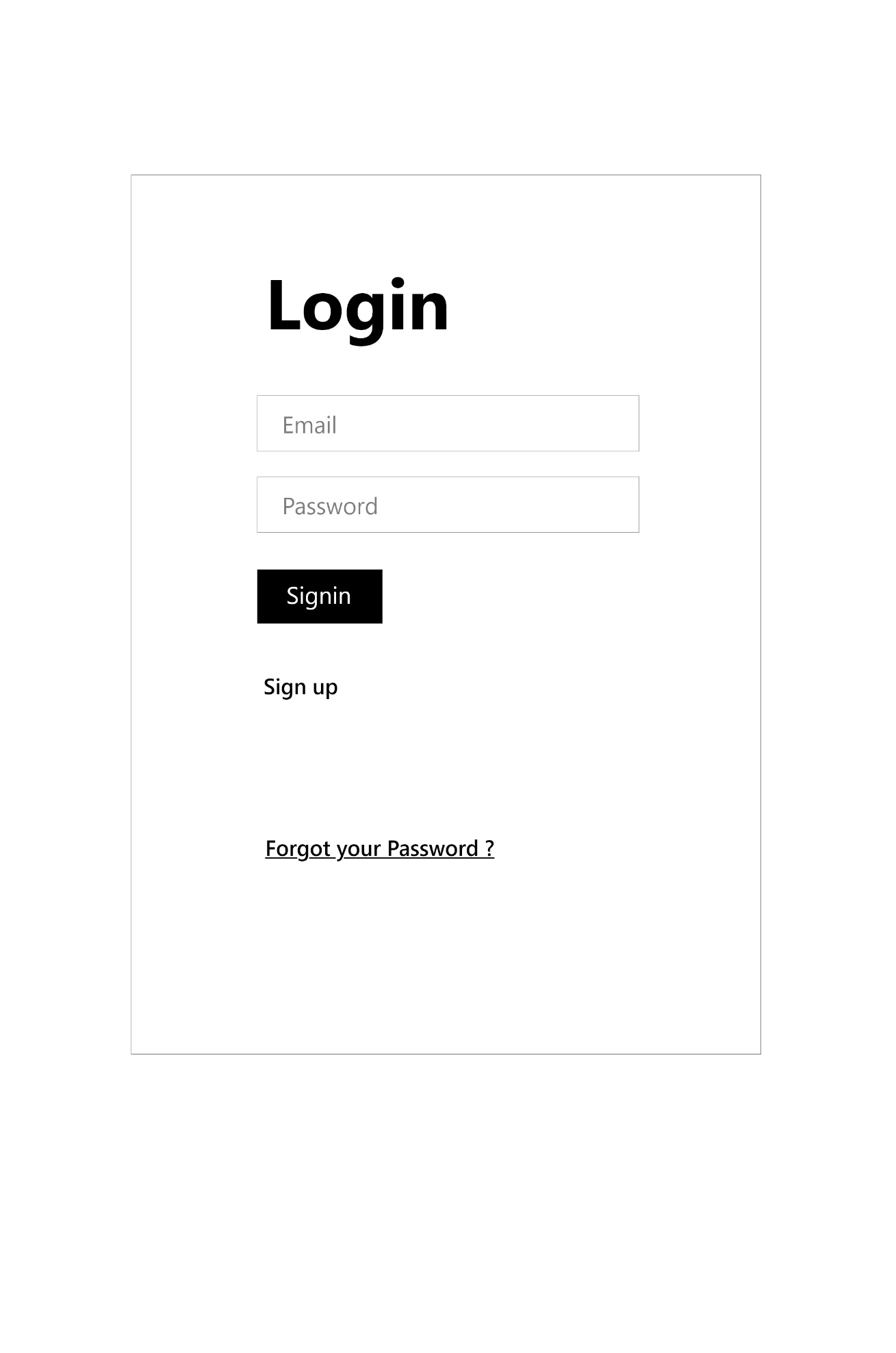
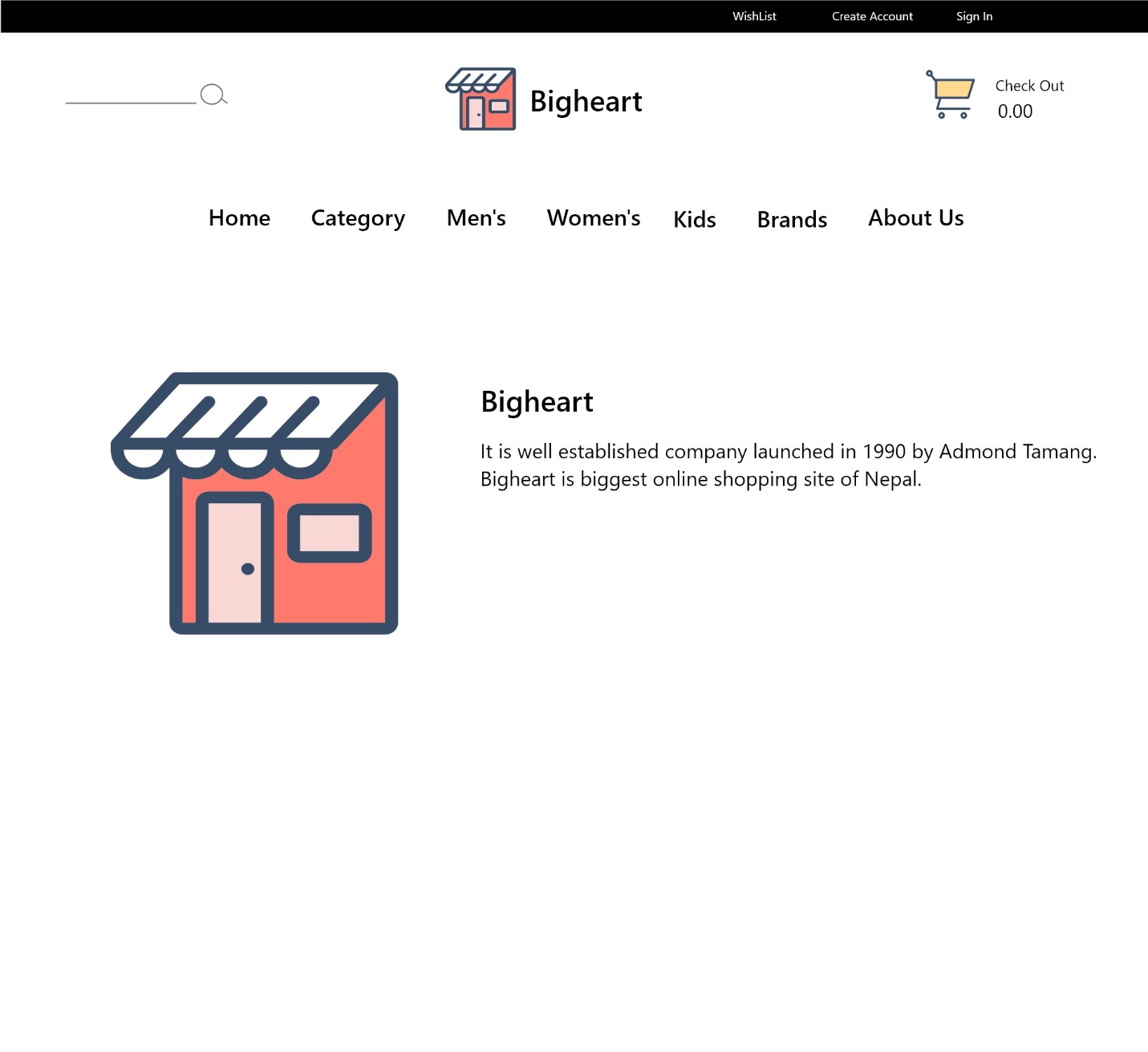
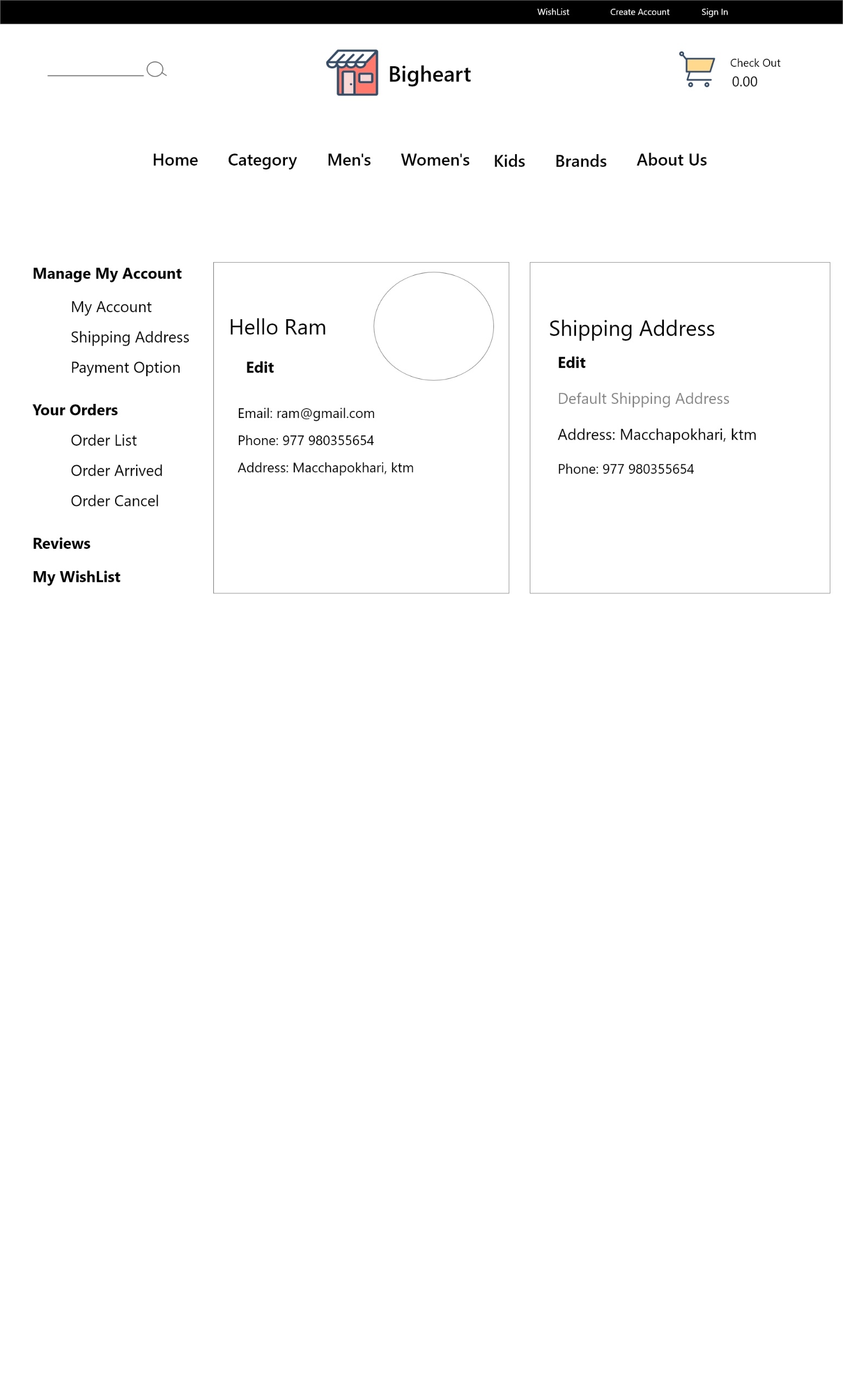
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column name** | **Data type** | **Length** | **Nullable** | **Key** | **Constraint** |
| **Id** | Integer | 10 | No | Primary key | PK\_id |
| **Category** | Varchar | 191 | No | - | - |
| **Parenti\_id** | Varchar | 11 | No | - | - |
| **url** | Varchar | 191 | No | - | - |

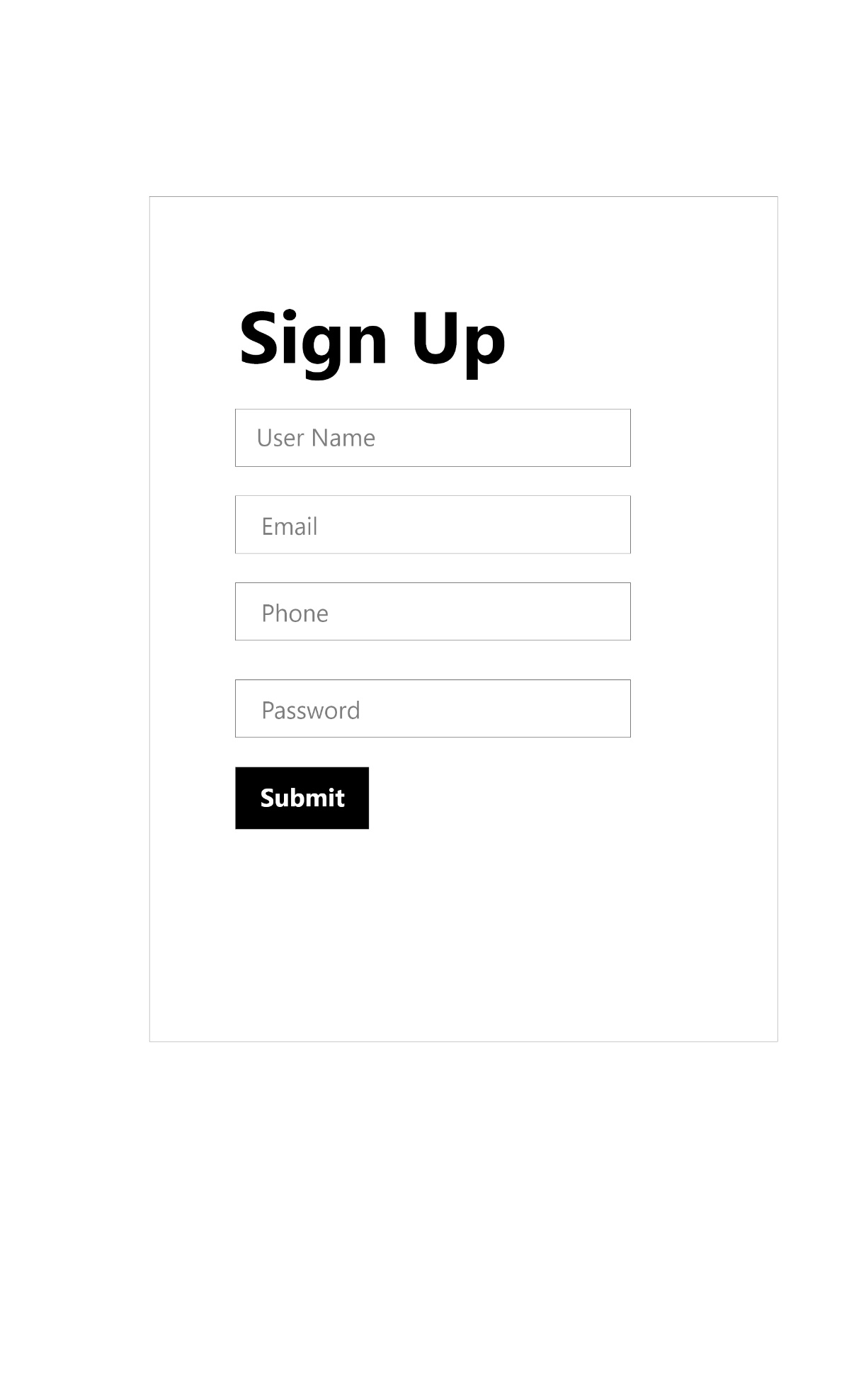
**Table: Wishlists**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column name** | **Data type** | **Length** | **Nullable** | **Key** | **Constraint** |
| **Id** | Integer | 10 | No | Primary key | PK\_id |
| **Product\_id** | Integer | 191 | No | - | - |
| **User\_id** | Integer | 11 | No | - | - |
| **Name** | Varchar | 191 | No | - | - |
| **Price** | Integer | 11 | No | - | - |
| **Store\_id** | Integer | 11 | No |  |  |
| **Image** | Varchar | 191 | No |  |  |

## UI modelling (prototype)







# Implementation

Phase where developers bring up our ideas to life i.e. actual coding process of the BigHeart begins in this phase. I created different types of diagrams, models etc. and gather all the ideas and implemented that idea in this phase of software development.

## Programming language

I have used Laravel framework which is based on PHP language. PHP is very user friendly and platform independent language. Since, I'm developing my application on web combination of PHP and Laravel is the best as compare to other format (Programming languages).

Laravel provides wide range of facilities and plugins which makes programming easier and more fun. It also helps to simplify the tasks by using its inbuild library like Auth, guard, middleware, routes etc. Laravel also helps to manages complex folder structure by using MVC pattern which helps to divide code according to their main functionality. For example: routes. Migration etc. have their own functionality which is managed by Laravel. Laravel has provide access to different commands of PHP that makes our work easier.

*All UI design and code are included in the appendix below.*

# Testing

It is the process of evaluating the function in the application in order to find the bugs in system. SDLC has six phases among them testing comes after implementing in SDLC.it is an integral part of software quality and it is important activity for supporting entire SDLC. testing must be done thorough strategic and showed throughout the SDLC for the finest outcomes which eventually aid in charge the cost of bug setting identical little. Testing not only shows a important part in SDLC for conference the values of an application but it also imitates in structure up the status of an organization.

The objective of testing is to ensure

* Software reliability
* Software quality
* System Assurance
* Optimum performance and capacity utilization

There are many types of testing to test software application like as: -

* Black box testing
* White box testing
* Unit testing
* Integration testing
* Usability testing
* System testing

### Black box testing

It is also known as UI testing. It tests the software without knowing the internal structure of program. All testing is done as customer’s point of view. Tester is only aware of what is software is supposed to do and unaware about how these requests are being processed. While tester is only known about the input and expected output of the software. The main purpose of black box testing is to check whether the software is working as expected as requirement. (admin, 2012)

**Test 1**

|  |  |
| --- | --- |
| **Test case no** | 1 |
| **Purpose of test** | To check route is working |
| **Test data** | / |
| **Expected result** | Visit home page |
| **Actual Result** | Successfully visited |
| **Test log (Status)** | Pass |

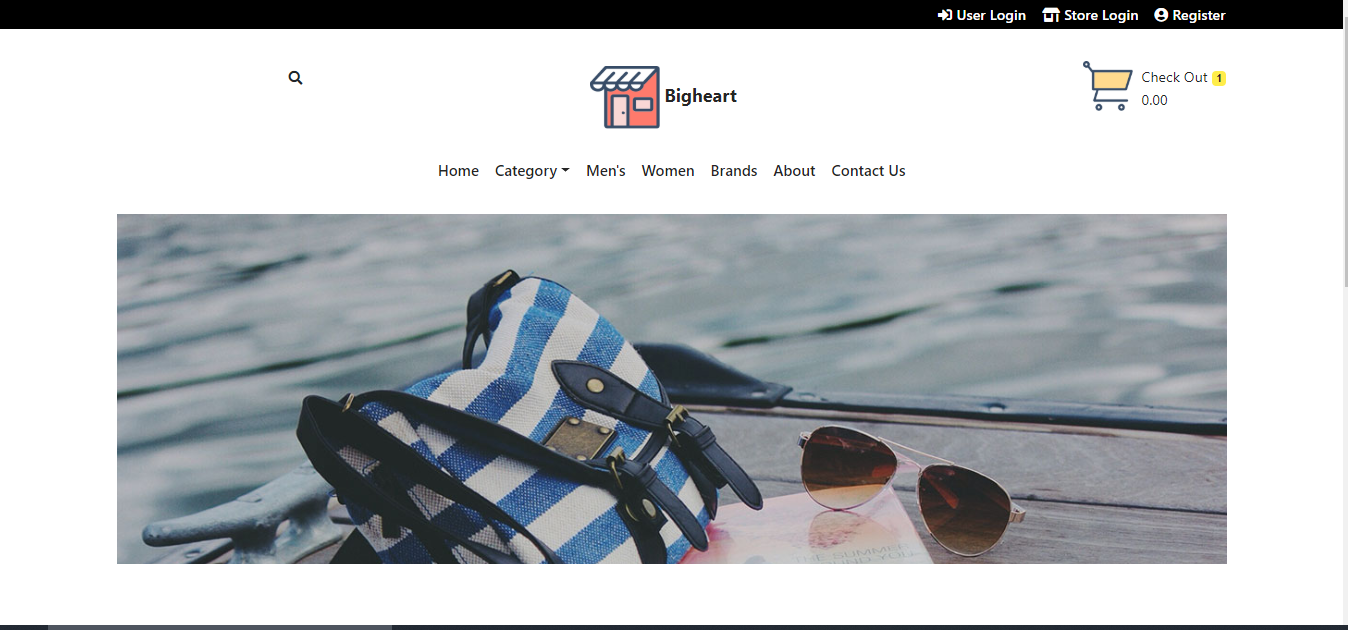
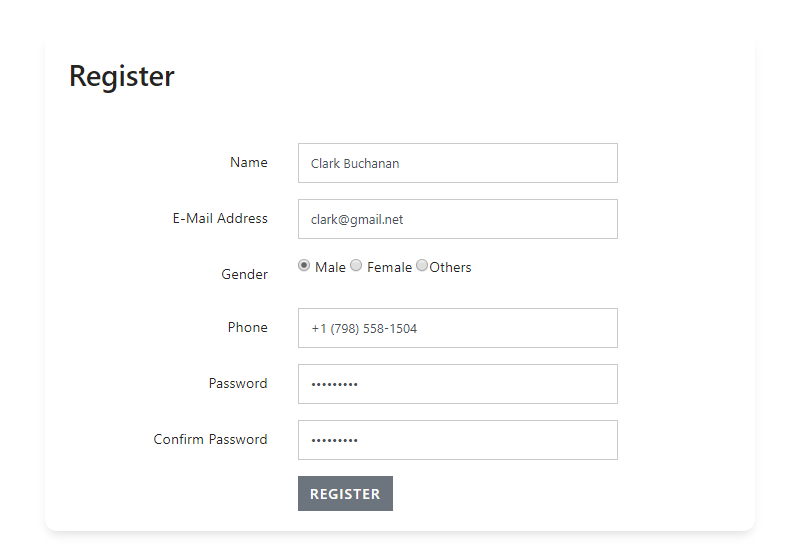
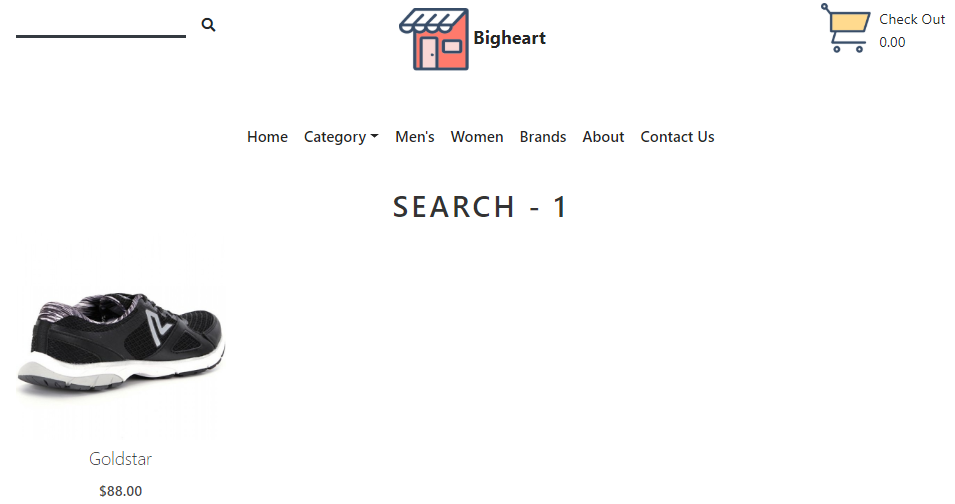
****

Figure 16: test case result

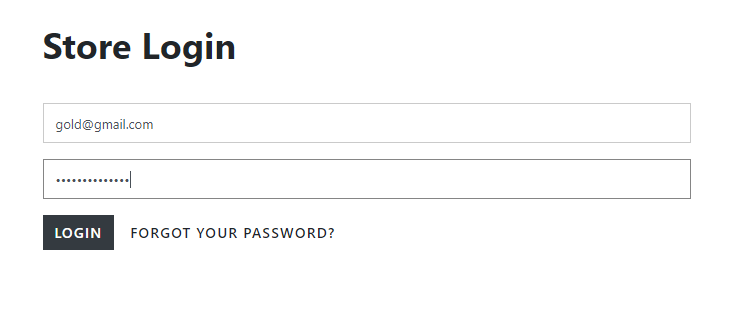
|  |  |
| --- | --- |
| **Test case no** | 2 |
| **Purpose of test** | To check whether user can register themselves or not |
| **Test data** | Clark buchanan, clark@gmail.net |
| **Expected result** | User should registered. |
| **Actual Result** | User is registered |
| **Test log (Status)** | Passed |



|  |  |
| --- | --- |
| **Test case no** | 3 |
| **Purpose of test** | To check whether user can search products or not |
| **Test data** | goldstar |
| **Expected result** | User should be able to search products |
| **Actual Result** | User can search products |
| **Test log (status)** | Pass |



|  |  |
| --- | --- |
| **Test case no** | 4 |
| **Purpose of test** | To check store user can login |
| **Test data** | Email: [gold@gmail.com](mailto:gold@gmail.com)  Password: [gold@gmail.com](mailto:gold@gmail.com) |
| **Expected result** | Login in to dashboard |
| **Actual Result** | Logged into user dashboard |
| **Class** | testLogin, UserFactory |



|  |  |
| --- | --- |
| **Test case no** | 5 |
| **Purpose of test** | To check whether product can be added or not |
| **Test data** | Product Name: Cathleen Durham  Description: Nobis placeat volup  Stock Quantity: 892  Price: 853 |
| **Expected result** | Product added |
| **Actual Result** | Product added successfully |
| **Test log (status)** | Passed |

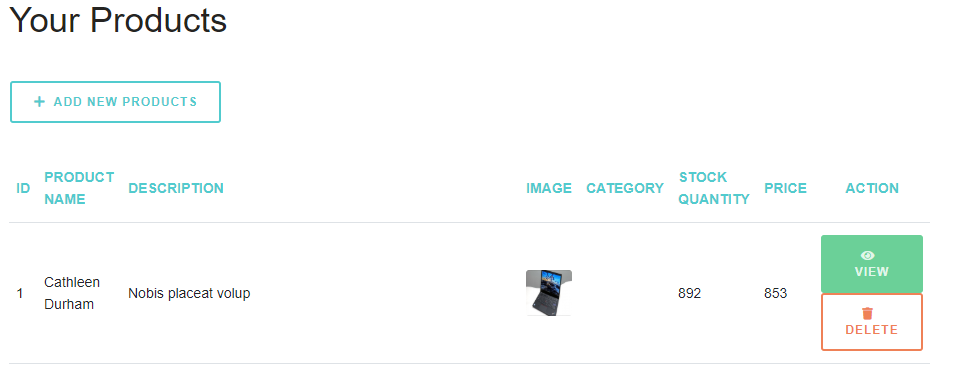
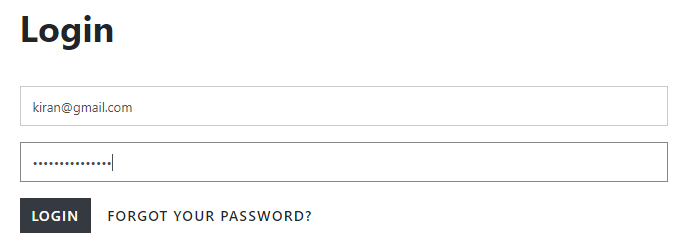
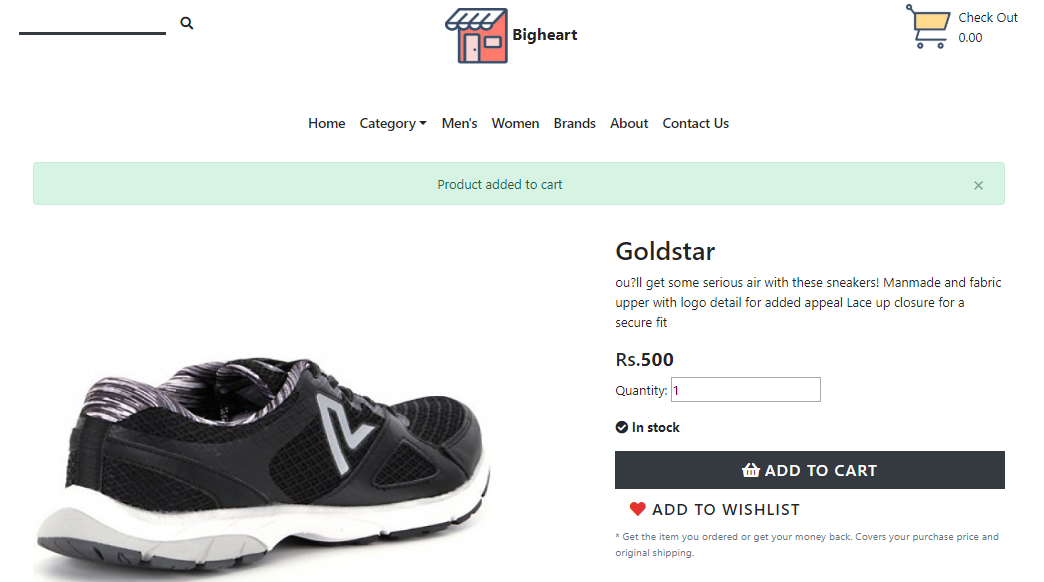


Figure 17: product added successfully

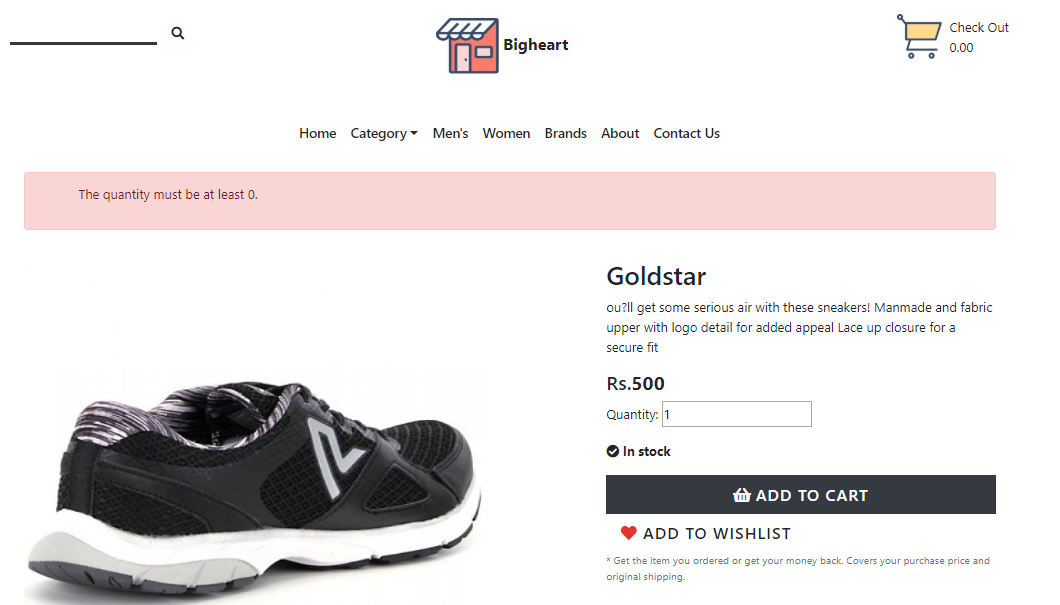
|  |  |
| --- | --- |
| **Test case no** | 6 |
| **Purpose of test** | To check customer can login |
| **Test data** | Email: [kiran@gmail.com](mailto:kiran@gmail.com)  Password: [kiran@gmail.com](mailto:kiran@gmail.com) |
| **Expected result** | Login in to home with authentication |
| **Actual Result** | Logged into home |
| **Test log (status)** | Passed |



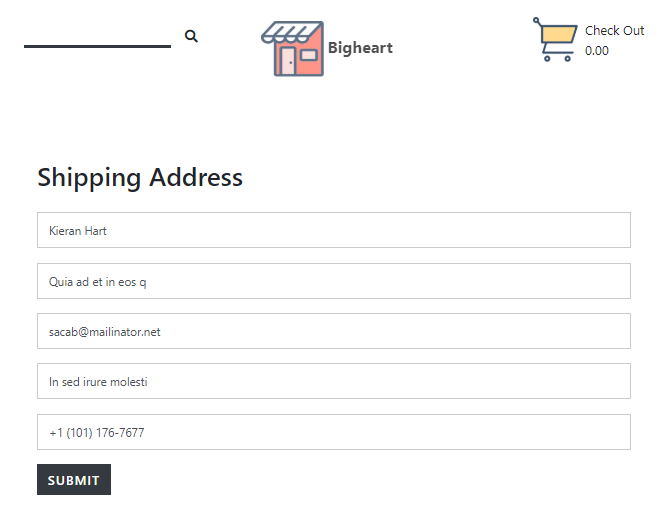
|  |  |
| --- | --- |
| **Test case no** | 7 |
| **Purpose of test** | To check whether user can add product to cart |
| **Test data** |  |
| **Expected result** | User should be able add product to cart |
| **Actual Result** | User can add product to cart |
| **Test log (status)** | passed |



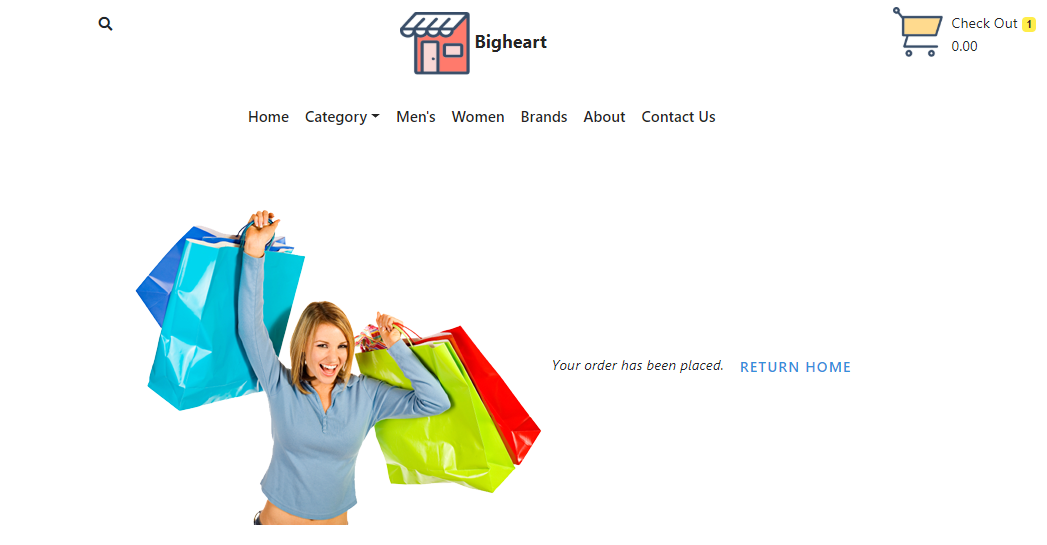
|  |  |
| --- | --- |
| **Test case no** | 8 |
| **Purpose of test** | To check whether user can input negative value in product quantity |
| **Test data** | -1 |
| **Expected result** | User should not be able to add negative value |
| **Actual Result** | User cannot add negative value. |
| **Class** | Product |



|  |  |
| --- | --- |
| **Test case no** | 9 |
| **Purpose of test** | To check whether user can add shipping address or not |
| **Test data** |  |
| **Expected result** | User should be able add shipping data |
| **Actual Result** | User can add shipping data |
| **Test log (status)** | passed |



|  |  |
| --- | --- |
| **Test case no** | 10 |
| **Purpose of test** | To check whether user can order product |
| **Test data** |  |
| **Expected result** | User should be able add order product |
| **Actual Result** | User can add order product |
| **Test log (status)** | pass |

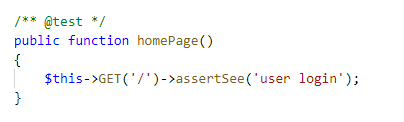
****

### Unit Testing.

It is level of testing where individual unit are tested to determine whether they are working or not. The testing is done by developer to make sure individual components are working as expected. The components like functions, procedures, classes, etc. are tested. The white box testing is used to test the unit testing.

|  |  |
| --- | --- |
| **Test case no** | 1 |
| **Purpose of test** | To check route is working |
| **Test data** |  |
| **Expected result** | Visit home page |
| **Actual Result** | Successfully visited |
| **Class** | testLogin |

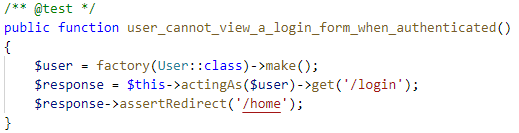
**Test Case 1**

****



|  |  |
| --- | --- |
| **Test case no** | 2 |
| **Purpose of test** | To check user can login without authentication |
| **Test data** |  |
| **Expected result** | Redirect to login |
| **Actual Result** | Redirected to login |
| **Class** | testLogin, UserFactory |

**Test case 2**





**Test case 3**

|  |  |
| --- | --- |
| **Test case no** | 3 |
| **Purpose of test** | To check whether product can be added or not |
| **Test data** |  |
| **Expected result** | Product added |
| **Actual Result** | Product added successfully |
| **Class** | Product, Store |

****



**Test case 4**

|  |  |
| --- | --- |
| **Test case no** | 4 |
| **Purpose of test** | To check whether user can register themselves or not |
| **Test data** |  |
| **Expected result** | User should registered. |
| **Actual Result** | User is registered |
| **Class** | User, UserFactory |

****



**Test case 5**

|  |  |
| --- | --- |
| **Test case no** | 5 |
| **Purpose of test** | To check whether user can update their profile |
| **Test data** |  |
| **Expected result** | User should be able to update their profile |
| **Actual Result** | User can change profile |
| **Class** | User, UserFactory |

****



**Test case 6**

|  |  |
| --- | --- |
| **Test case no** | 6 |
| **Purpose of test** | To check whether user can add shipping address or not |
| **Test data** |  |
| **Expected result** | User should be able add shipping data |
| **Actual Result** | User can add shipping data |
| **Class** | Shipping |

****



**Test case 7**

|  |  |
| --- | --- |
| **Test case no** | 7 |
| **Purpose of test** | To check whether user can order product |
| **Test data** |  |
| **Expected result** | User should be able add order product |
| **Actual Result** | User can add order product |
| **Class** | Product |

****



**Test case 8**

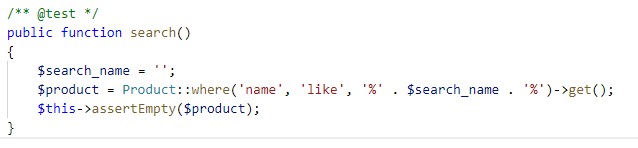
|  |  |
| --- | --- |
| **Test case no** | 8 |
| **Purpose of test** | To check whether user can add product to cart |
| **Test data** |  |
| **Expected result** | User should be able add product to cart |
| **Actual Result** | User can add product to cart |
| **Class** | Product, Cart |

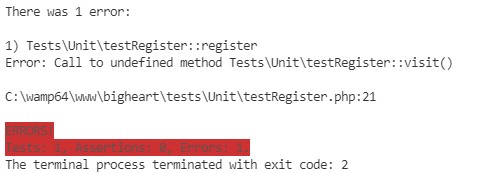
****



**Test case 9**

|  |  |
| --- | --- |
| **Test case no** | 9 |
| **Purpose of test** | To check whether user can search products or not |
| **Test data** | addidas |
| **Expected result** | User should be able to search products |
| **Actual Result** | User can search products |
| **Class** | Product |

****



**Test case 10**

|  |  |
| --- | --- |
| **Test case no** | 10 |
| **Purpose of test** | To check user can delete product or not |
| **Test data** | -1 |
| **Expected result** | Should delete product |
| **Actual Result** | User can delete product |
| **Class** | Product |

****



# Other project issues

## Work Breakdown Structure

All the things a project needs to accomplish in the project is displays graphically which helps to understand the projects activities clearly. The purpose of WBS is to break down complex activities into simple form.

Figure 18: Work break down structure

## **Milestones**

|  |  |
| --- | --- |
| **Milestones** | **Date** |
| **Project Management**  Risk Management  Work Breakdown Structure  Configuration Management  Proposal Submission | **12/21/2018 -1/3/2019**  12/21/2018 - 12/24/2018  12/25/2018 - 12/27/2018  12/28/2018 - 12/30/2018  12/31/2018 - 1/1/2019 |
| **Analysis**  Requirement Analysis  Use Case  Architecture (Initial Class Diagram)  Analysis Specification | **1/2/2019 - 1/25/2019**  1/2/2019 - 1/7/2019  1/8/2019 - 1/11/2019  1/12/2019 - 1/17/2019  1/18/2019 - 1/25/2019 |
| **Design**  Structural Diagram  Behavioral Diagram  UI Design  Database Design | **1/26/2019 - 2/24/2019**  1/26/2019 - 2/3/2019  2/4/2019 - 2/13/2019  2/14/2019 - 2/20/2019  2/21/2019 - 2/24/2019 |
| **Implementation**  Building Database  Coding | **2/25/2019 - 3/28/2019**  2/25/2019 - 3/1/2019  3/2/2019 - 3/28/2019 |
| **Testing**  Unit Testing  Integration Testing  Blackbox Testing  Whitebox Testing | **3/29/2019 - 4/8/2019**  3/29/2019 - 3/31/2019  4/1/2019 - 4/3/2019  4/4/2019 - 4/6/2019  4/7/2019 - 4/8/2019 |
| **Deployment**  User Training  Final Report | **4/9/2019 - 4/18/2019**  4/9/2019 - 4/13/2019  4/14/2019 - 4/18/2019 |

**Description of Milestones:**

1. **Project Management (14 Days)**

* Risk Management (4 Days)
* Work Breakdown Structure (3 Days)
* Configuration Management (3 Days)
* Proposal Submission (4 Days)

1. **Analysis (19 Days)**

* Requirement Analysis (7 Days)
* Use Case (4 Days)
* Architecture (3 Days)
* Analysis Specification (5 Days)

1. **Design (32 Days)**

* Structural Diagram (9 Days)
* Behavioral Diagram (7 Days)
* UI Design (8 Days)
* Database Design (8 Days)

1. **Implementation (33 Days)**

* Build Database (7 Days)
* Coding (26 Days)

1. **Testing (13 Days)**

* Unit Testing (3 Days)
* Integration Testing (3 Days)
* Blackbox Testing (3 Days)
* Whitebox Testing (4 Days)

1. **Deployment (7 Days)**

* User Training (3 Days)
* Final Report (4 Days)

## **Scheduling / Gantt Chart**

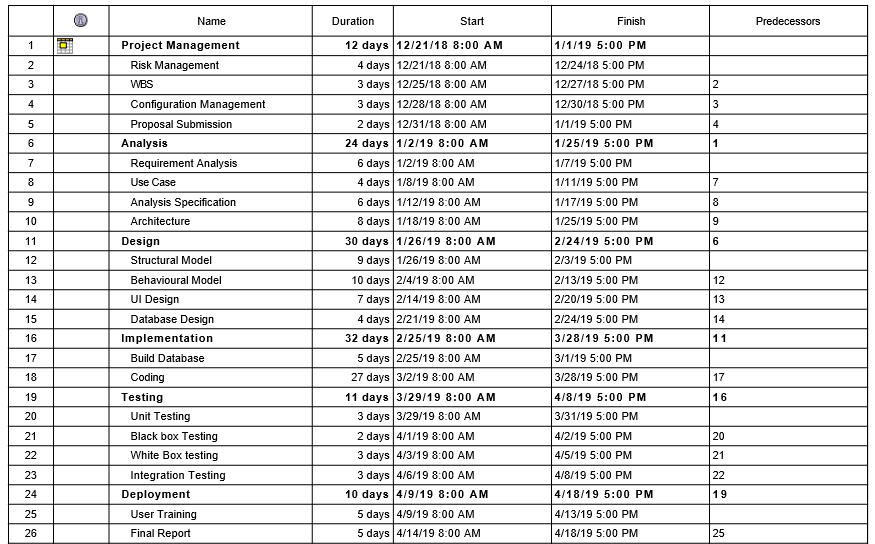
****

Figure 19: Scheduling time for tasks

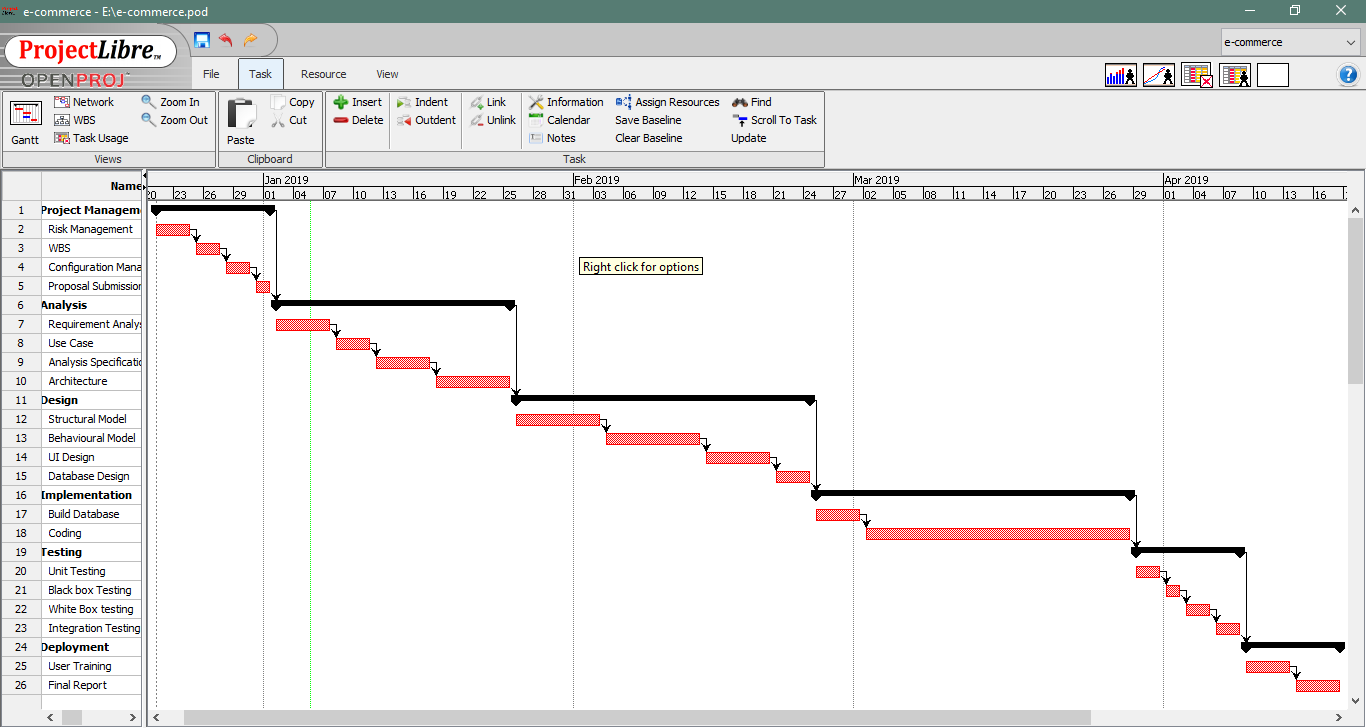
****

Figure 20: Gantt chart

# **Risk Management**

It is the process of identifying and prioritizing possible threats to an organization or system. Harmful risk is analyzed and solution to the threats are prepared.

**Impact = Likelihood \* Consequences**

Risk Likelihood values are shown as follows

|  |  |
| --- | --- |
| **Likelihood** | **Value** |
| Low | 1 |
| Medium | 2 |
| High | 3 |

Risk Consequence values are shown below

|  |  |
| --- | --- |
| **Consequence** | **Value** |
| Very low | 1 |
| Low | 2 |
| Medium | 3 |
| High | 4 |
| Very High | 5 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S. No** | **Risks** | **Likelihood** | **Consequences** | **Impact** | **Solution** |
|  | Lack of staff training | 2 | 3 | 6 | At the stage of deployment, staff should be trained properly to use the system. |
|  | Data theft | 3 | 4 | 12 | Strong company policy should be established. |
|  | Failure to meet requirement | 2 | 4 | 12 | Proper planning is to be done in every phase. |
|  | DDOS attack | 1 | 4 | 4 | Placing the publicly exposed server in DMZ. |
|  | Database error | 1 | 5 | 5 | Analyze the data traffic and upgrade the hardware if necessary. |
|  | Server failure | 1 | 5 | 5 | IT security should be hired to look after the server. |
|  | Change in government policy | 1 | 5 | 5 | Should be aware about the situation of the country and make changes accordingly. |

# 

# **Configuration Management**

Configuration management is used in this project to keep track of application and related information which includes software version and updates. It keeps track of modification, changes and update of my projects. I have used GitHub for configuration management because it is free and easy to use.

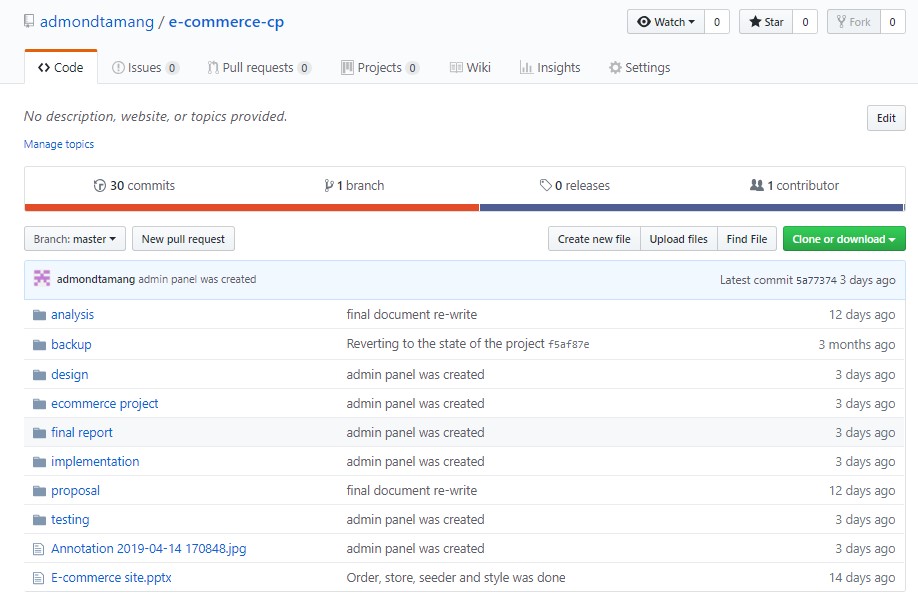


Figure 21: Github root directory

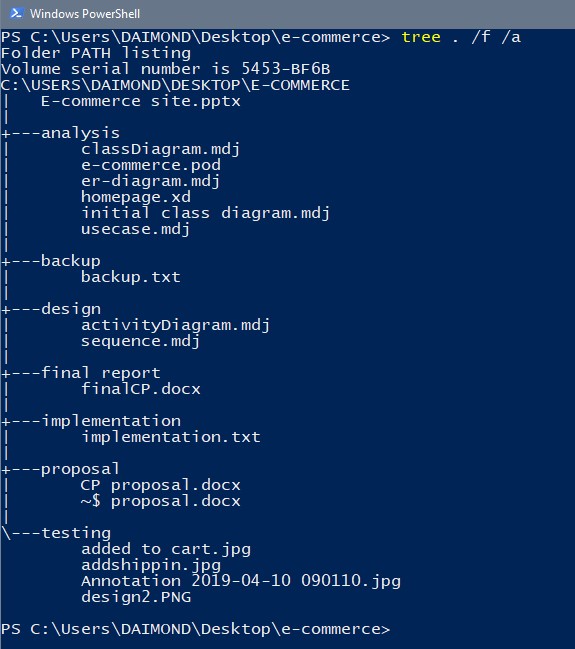


Figure 22: Local directory of project

### User Manual

User manual is a guide that helps user to understand how to make use of the system. It contains all essential information like application features, alternate mode of operations and different step by step procedures of usage of the application.

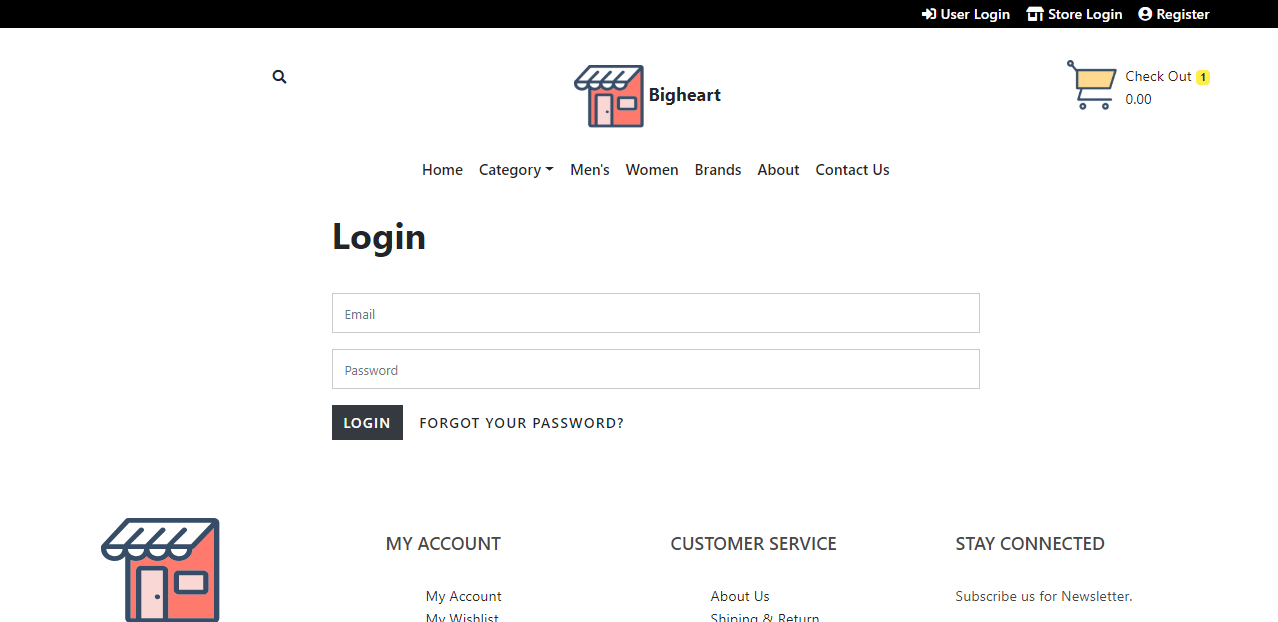
#### System overview

Minimum requirement to run the system:

* Windows OS greater than window 8 or 10
* 500 GB of hard disk and 2 GB of RAM
* A web browser
* XAMP or WAMP installed

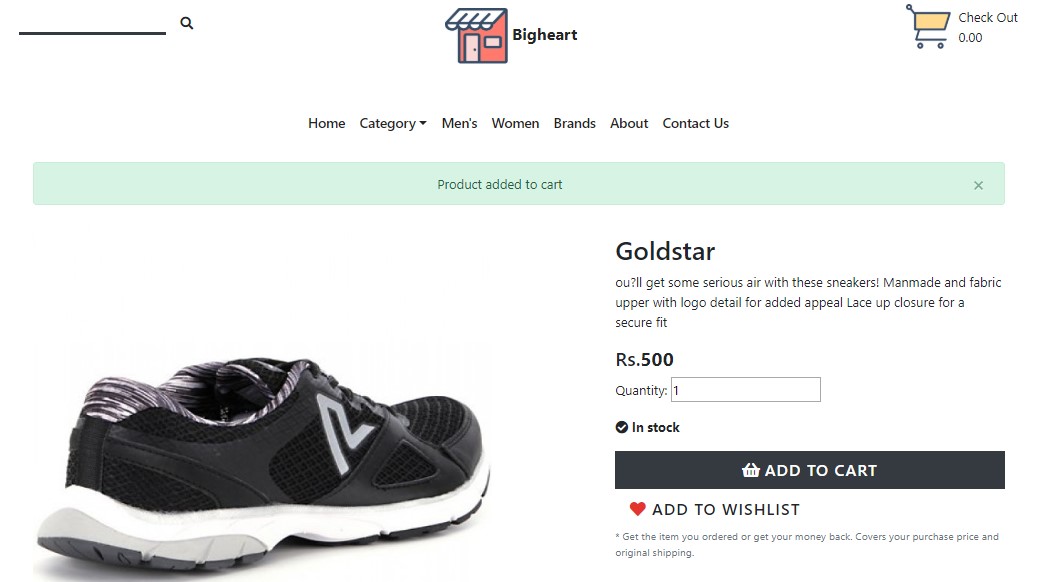
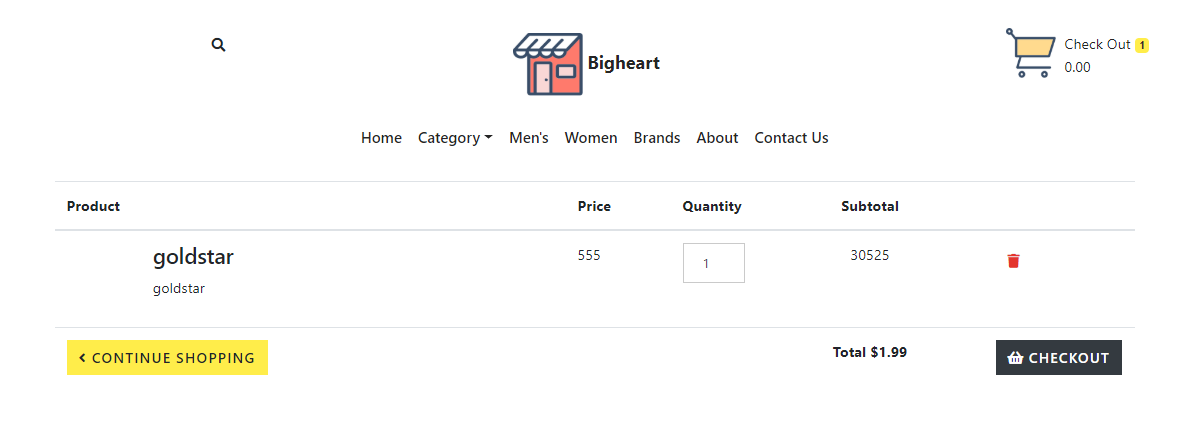
#### For user login

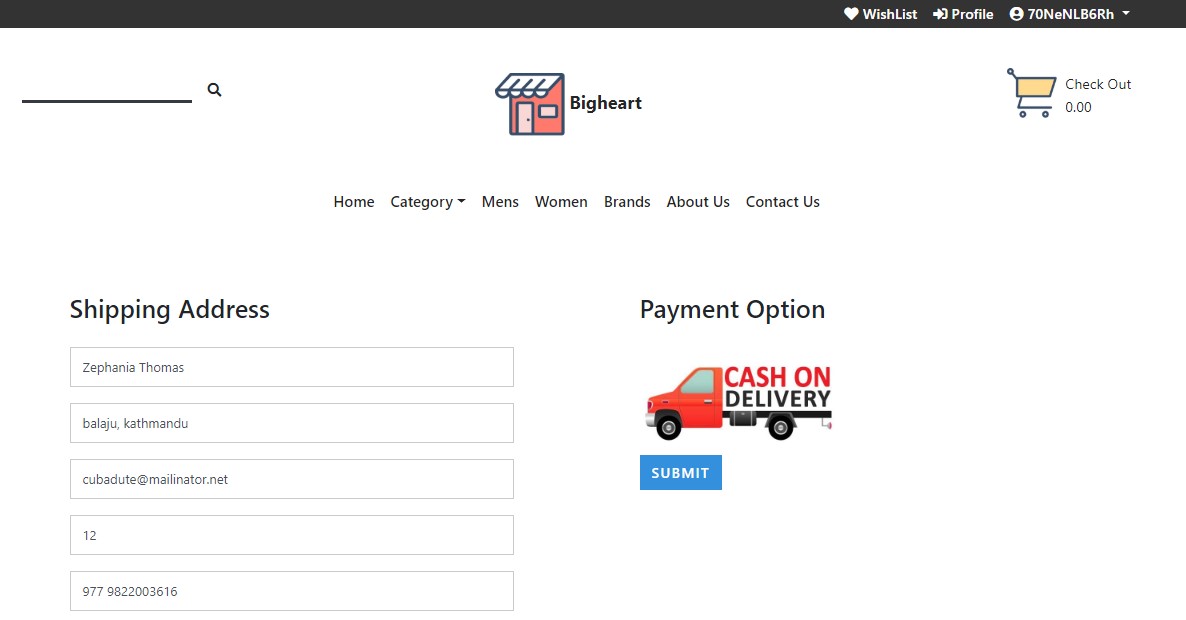
1. Click on login from home page
2. Provide valid email and password
3. Click login button to login



1. Hurray, now you are logged in.

#### To order Product

1. Select products you like
2. Click on add to cart button
3. Now, click on (check out) at the top right corner where you'll find the products you want to buy.
4. Make changes if you don't want to buy
5. Click on checkout to go to order product.
6. Fill up shipping address and payment method.



1. Now, click order to conform all purchase item.
2. Hurray, you've successful order a product from the website.

### Limitation

My project has following limitations:

* Online Payment system is not available
* Admin is not privilege enough to do everything.
* Chatting service is not available yet.

### Future Work

The progress of e-commerce has been completed with some limitations. However, the application has much to accomplish.

Following are the things which can be implement in near future:

* To make smoother experience, I will make modification on GUI.
* User can sign up via social media like facebook, google etc.
* Payment system like E-sewa, Khalti will be implemented in future work.
* Chat system is the priority for upgrading the system.

# Conclusion

The Internet has become a major resource in modern business, thus electronic shopping has gained significance not only from the entrepreneur’s but also from the customer’s point of view. For the entrepreneur, electronic shopping generates new business opportunities and for the customer, it makes comparative shopping possible. As per a survey, most consumers of online stores are impulsive and usually decide to stay on a site within the first few seconds. “Website design is like a shop interior. If the shop looks poor or like hundreds of other shops the customer is most likely to skip to the other site. Hence, we have designed the project to provide the user with easy navigation, retrieval of data and necessary feedback as much as possible. In this project, the user is provided with an e-commerce web site that can be used to buy all types baby products online.

# Appendix

### UI Design

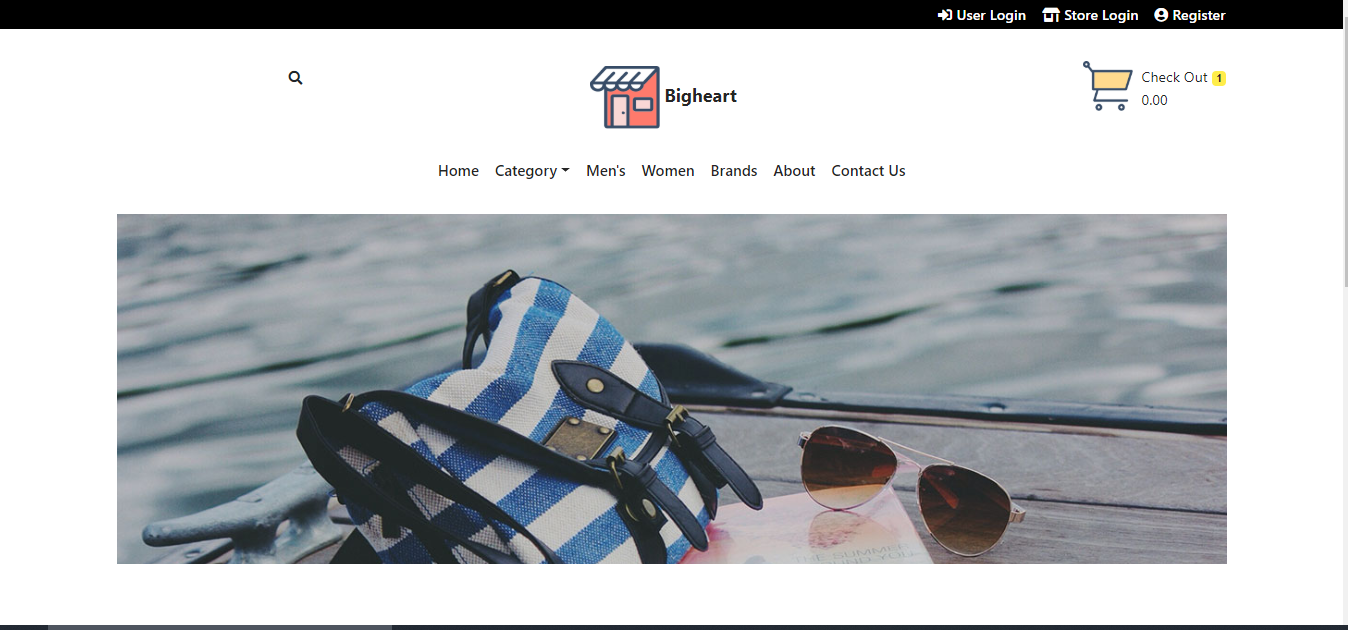
****

Figure 1: Home Page

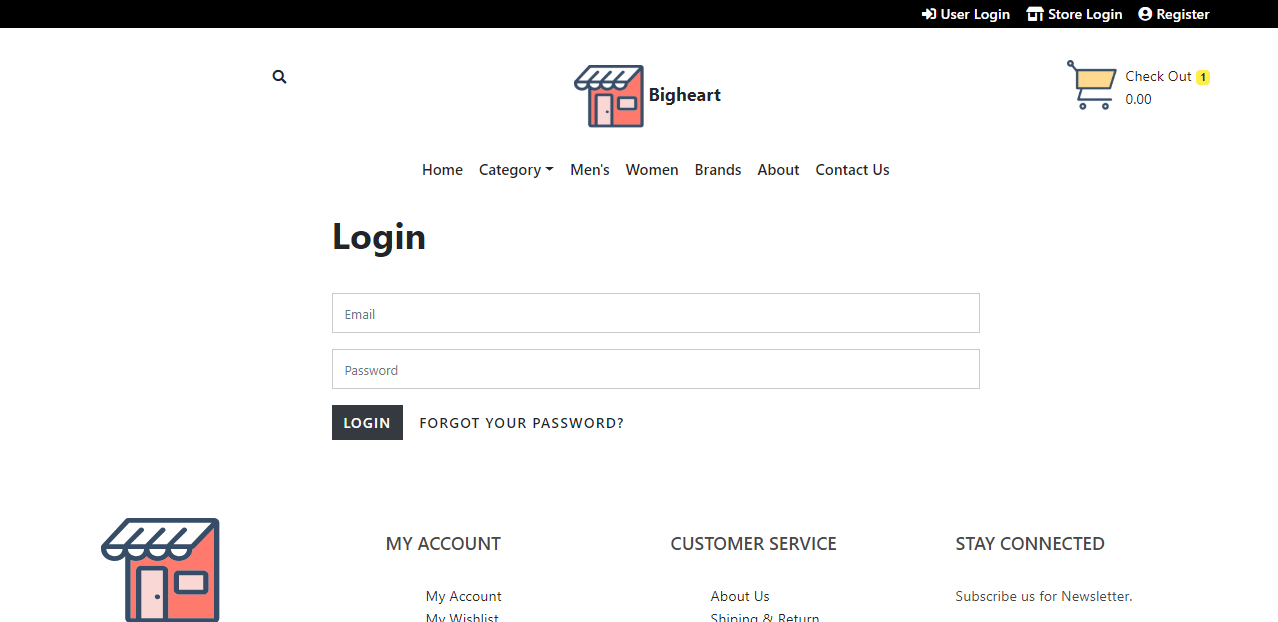
****

Figure 2: Login page

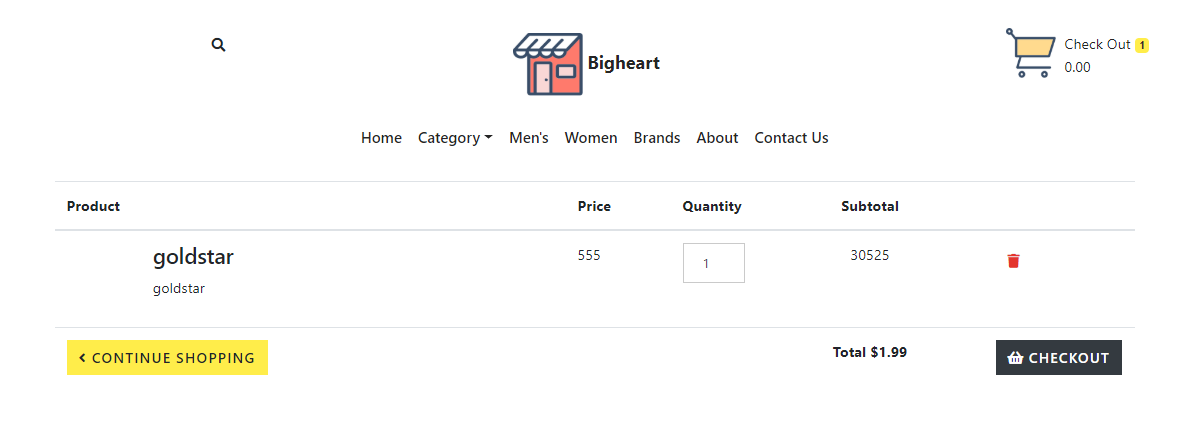
****

Figure 3: Add to cart

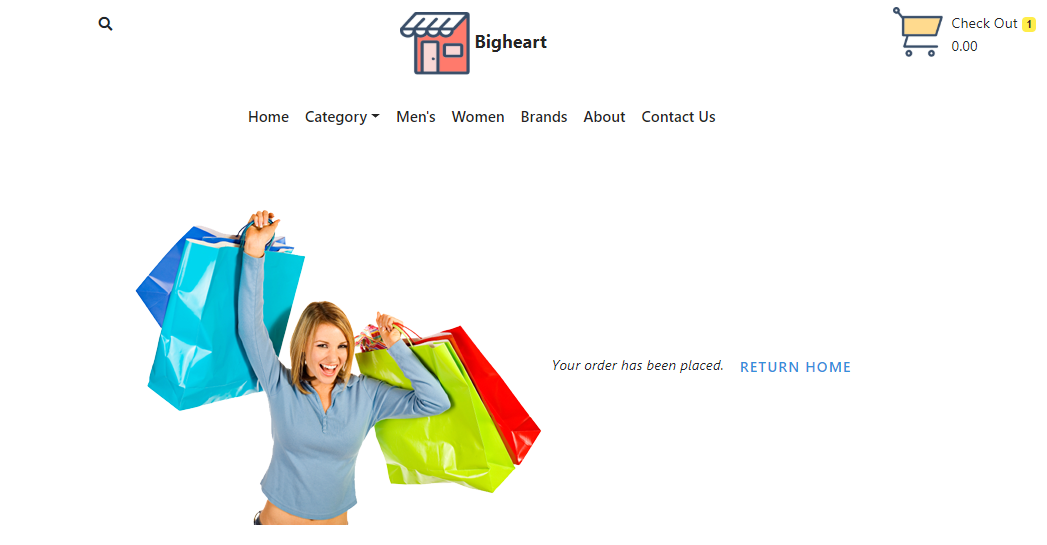
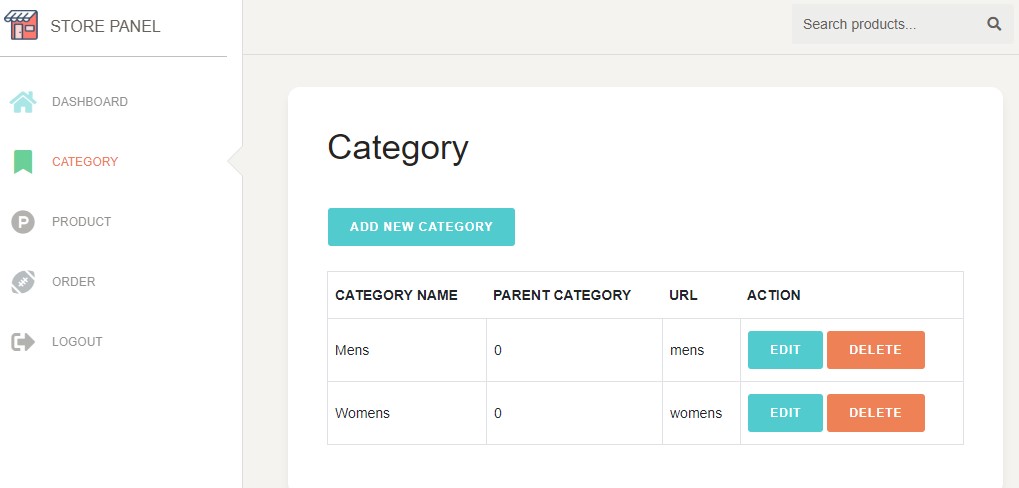
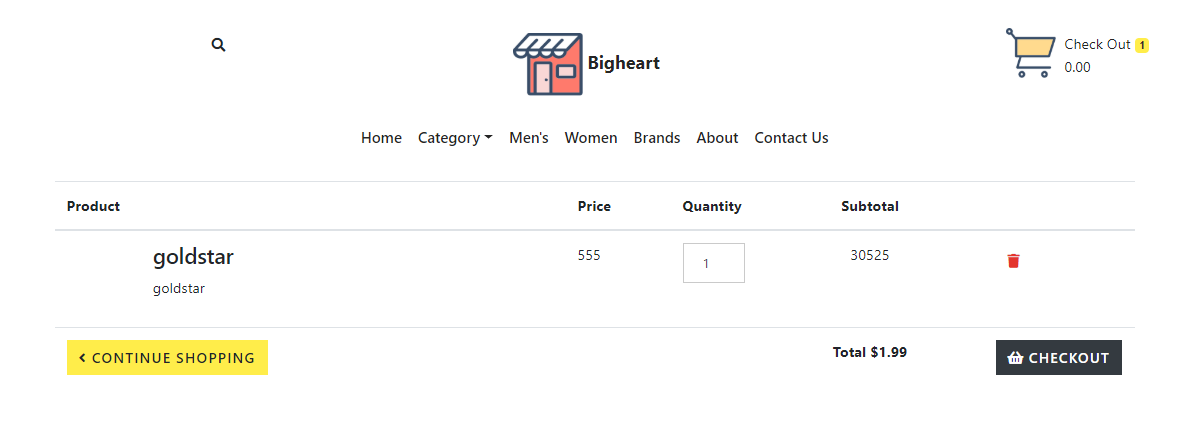
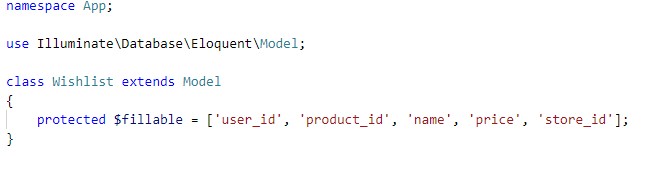
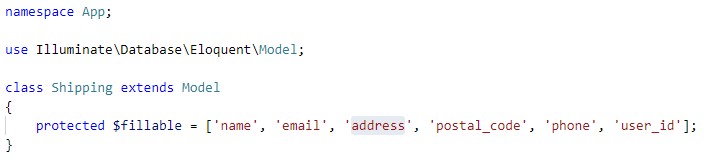
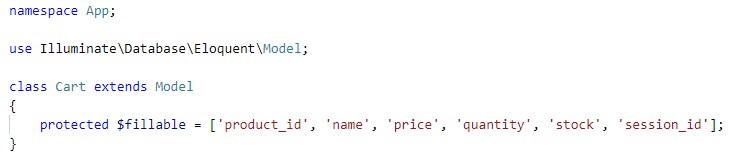
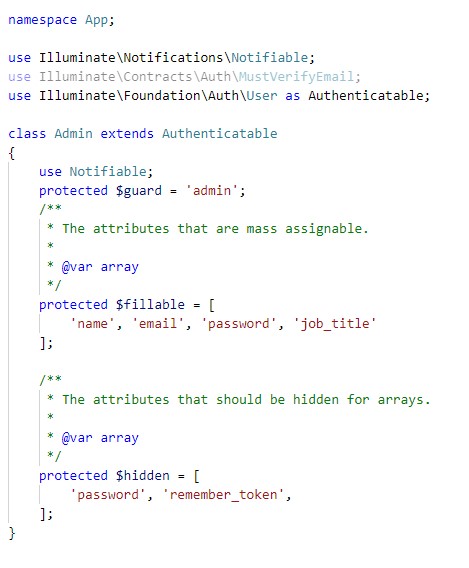
****

Figure 4: Check out





View Code



# References

admin, S., 2012. *softwaretestingclass.* [Online]   
Available at: https://www.softwaretestingclass.com/what-is-black-box-testing/  
[Accessed 8 4 2019].

Anon., 2019. *businessdictionary.* [Online]   
Available at: http://www.businessdictionary.com/definition/object-oriented-analysis.html  
[Accessed 8 4 2019].

Anon., n.d. [Online]   
Available at: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/