**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**

**Table of Contents**

[**1** **Introduction** 4](#_Toc5170750)

[**1.1** **Project Introduction** 4](#_Toc5170751)

[**1.2** **Justification for project** 4](#_Toc5170752)

[**1.2.1** **Background of the project** 4](#_Toc5170753)

[**1.2.2** **Problem Statement** 4](#_Toc5170754)

[**1.3** **Description of project** 4](#_Toc5170755)

[**1.3.1** **Features** 4](#_Toc5170756)

[**2** **Project Scope** 5](#_Toc5170757)

[**2.1** **Scope and limitation of project** 5](#_Toc5170758)

[**2.2** **Aims and Objectives** 5](#_Toc5170759)

[**3** **Development Methodology** 6](#_Toc5170760)

[**3.1** **Methodology used** 6](#_Toc5170761)

[**3.2** **Design Pattern** 7](#_Toc5170762)

[**3.3** **System Architecture** 8](#_Toc5170763)

[**4** **Work Breakdown Structure (WBS) / Scheduling** 9](#_Toc5170764)

[**4.1** **Work Breakdown Structure** 9](#_Toc5170765)

[**4.2** **Milestones** 9](#_Toc5170766)

[**4.3** **Scheduling / Gantt Chart** 11](#_Toc5170767)

[**5** **Risk Management** 12](#_Toc5170768)

[**6** **Configuration Management** 14](#_Toc5170769)

[**6.1** **Requirements** 16](#_Toc5170770)

[**6.2** **MoSCoW prioritization** 16](#_Toc5170771)

[**7** **Conclusion of the project** 36](#_Toc5170772)

[8 References 37](#_Toc5170773)

[Figure 1 Phases of waterfall model 6](#_Toc534496190)

[Figure 2: MVC design pattern 7](#_Toc534496191)

[Figure 3: Three tier architecture 8](#_Toc534496192)

[Figure 4: Work break down structure 9](#_Toc534496193)

[Figure 5: Scheduling time for tasks 11](#_Toc534496194)

[Figure 6: Gantt chart 11](#_Toc534496195)

[Figure 7: Github root directory 13](#_Toc534496196)

[Figure 8: Github proposal directory 13](#_Toc534496197)

[Figure 9: Local directory of project 14](#_Toc534496198)

# **Introduction**

E-commerce is fast gaining ground as an accepted and used business paradigm. More and more business houses are implementing web sites providing functionality for performing commercial transactions over the web. It is reasonable to say that the process of shopping on the web is becoming commonplace.

## **Project Introduction**

The objective of this project is to develop a general-purpose e-commerce store where any kind of product can be bought from the comfort of home through the Internet. However, for implementation purposes, this paper will deal with an online shopping for baby product An online store is a virtual store on the Internet where customers can browse the catalog and select products of interest. The selected items may be collected in a shopping cart. At checkout time, the items in the shopping cart will be presented as an order. At that time, more information will be needed to complete the transaction. Usually, the customer will be asked to fill or select a billing address, a shipping address, a shipping option, and payment information such as bKash, or Cash on Delivery. An e-mail notification is sent to the customer as soon as the order is placed.

## **Justification for project**

### **Background of the project**

In today’s busy world, people don’t have time for their personal needs. And the technology is so fast that anyone can do anything by just sitting in a room. The internet is the way that helps a person in all aspects. If someone wish to buy and view things, he can buy online with the help of internet. Today there are very least organizations which are manual. Everything is going to be computerized and online whether it is banking, advertising or shopping. We are trying to help people to make their life easier by proving online shopping facility. In this we have introduced many modules like admin module and customer module. The customer has to register for any enquiry related to Nepalis products. The unregistered person can’t access this application. The registered customer can view details of clothes and he can buy of his choice and need. He must pay the price of products he/she wishes to buy. The admin module contains the access of admin on the application. The admin can change everything in the application. He can add, delete, update any information regarding the clothes. The project’s home page includes the registration link. The registered users can login to their account for their queries or buy products. And the unregistered users have first to register. The registration can be done by following the sign-up link.

### **Problem Statement**

This application will reduce the problems arise in previous types of e-commerce system. Previously user do not have know about the products arising in nepali market. The main problem of Nepal is there is not enough market and knowledge to sell the products. And people don’t know local nepali products. To problem can be solved by the e-commerce providing people the right products.

## **Description of project**

### **Features**

The features of the project are listed below:

* **User can sign up and login**

User can easily create their account and login into the website.

* **User can upload their products**

Developers can upload their products in the application and make available for customers.

* **Add products to wish list**

User can add the product to wish list if he/she like the product.

* **Rate and comment products**

User can rate and comment about product.

* **Get help from developer via online chat**

If user get confused, then they can chat with product developer.

* **User friendly interface**

Anyone can use the application interface as it is easy to navigate, and simple design helps to understand easily.

* **Share products with friends**

It facilities to share products via social medias.

* **Filter and search products**

User can use filter products to find the exact product they are looking.

# **Project Scope**

## **Scope and limitation of project**

## **Aims and Objectives**

**Aims:** The aims that I want to achieve are listed below:

* Developer would post their website and provide services to the customer.
* Customer should easily interact with the website and buy websites according to their needs.

**Objectives:** Action that I will take to achieve my aim are:

* Customer will have the ability to sign-up their account and browse different products.
* Developers will have the facility to post their website and provide services to the customer.

# **Development Methodology**

## **Methodology used**

Waterfall methodology was used to develop the application. It is sequential life cycle model. It is sequence of process which we cannot overtake or skip until the previous phase has been completed.

The waterfall methodology was used because of following reasons:

1. It is simple and easy to understand.
2. Progress of my project can be tracked easily.
3. My project has clear requirement, so I think waterfall model is suitable for my project.

Figure 1 Phases of waterfall model

## **Design Pattern**

I followed MVC design pattern which stands for Model-View-Controller.

The reason of choosing MVC design pattern are: (interserver, 2018)

* **Faster development process**

It supports parallel development which means that one programmer can work on application design and other can work on business logic simultaneously.

* **Modification does not affect entire model**

Model part does not depend upon view part, so any modification doesn’t entirely affect the architecture.

* **High cohesion and low coupling**

It makes low coupling among models and enables grouping of related logical action on a controller.

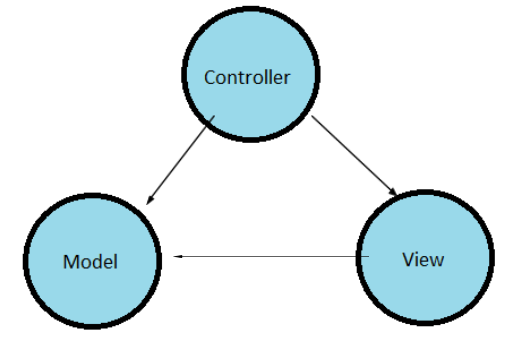


Figure 2: MVC design pattern

## **System 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.

I have used 3 tier architectures because of following reasons:

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.

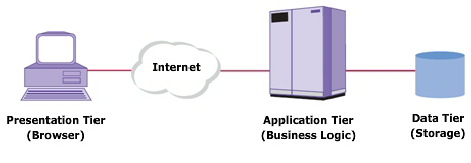


Figure 3: Three tier architecture

# **Work Breakdown Structure (WBS) / Scheduling**

## **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 4: 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:**

* **Project Management (12 days)**
* **Analysis (24 days)**
* **Design (30 days)**
* **Implementation (32 days)**
* **Testing (11 days)**
* **Deployment (10 days)**

## **Scheduling / Gantt Chart**

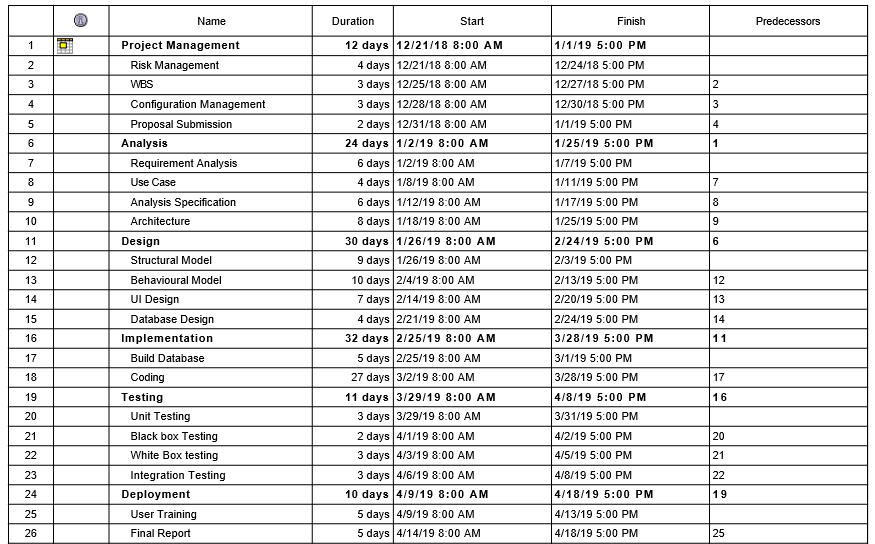
****

Figure 5: Scheduling time for tasks

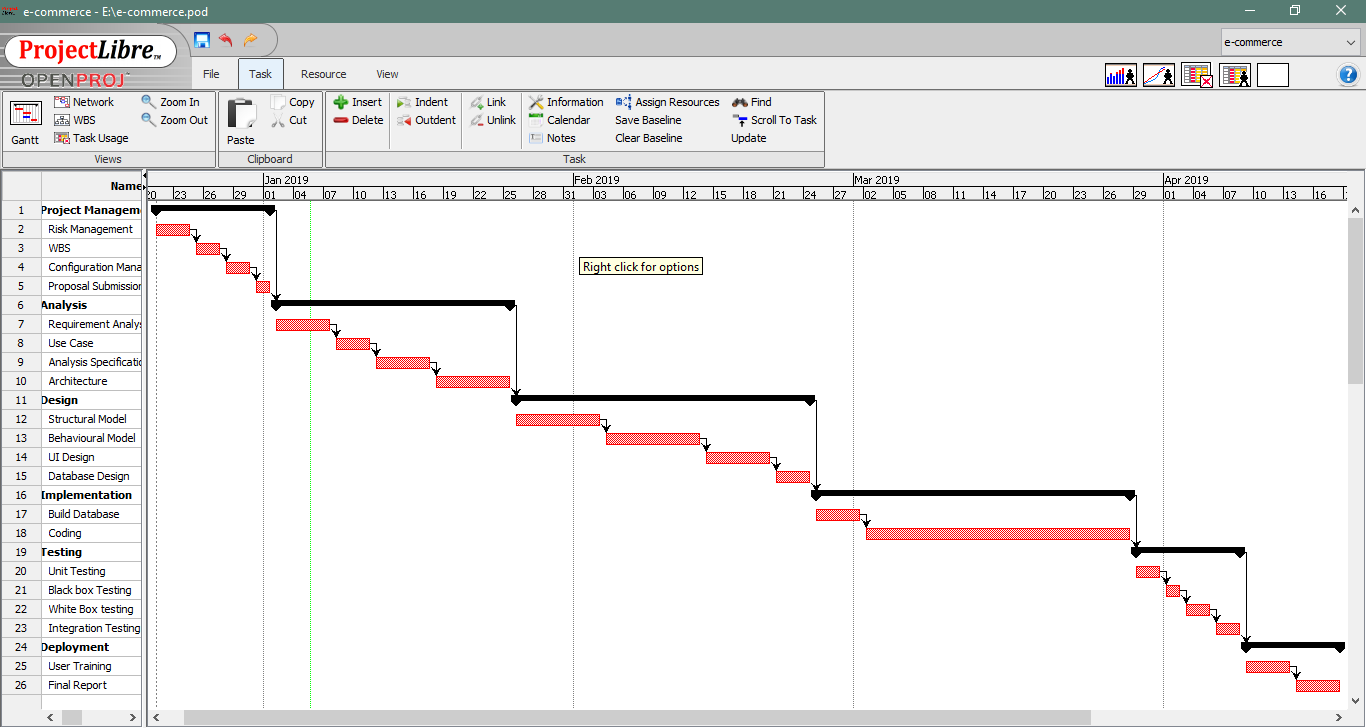
****

Figure 6: 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 to keep track of application and related information which includes software version and updates. It keeps track of modification, changes and update of any projects.

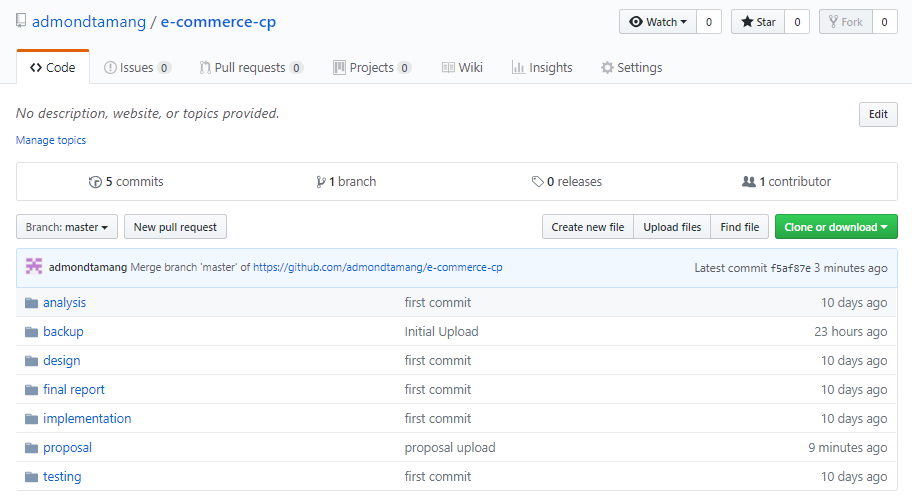


Figure 7: Github root directory

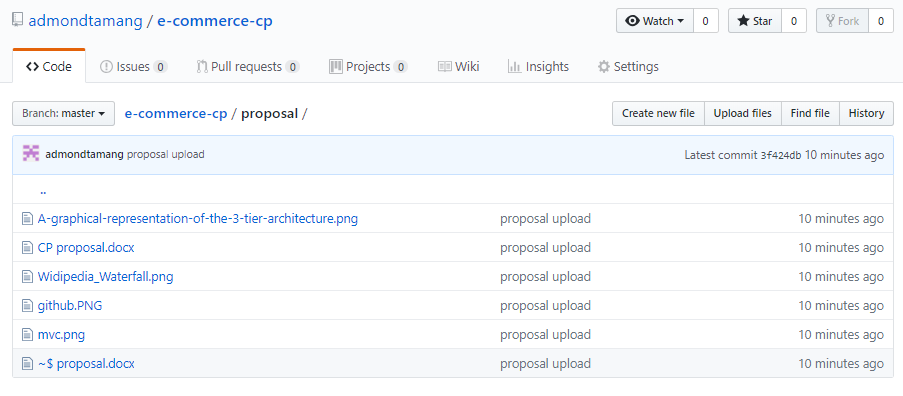


Figure 8: Github proposal directory

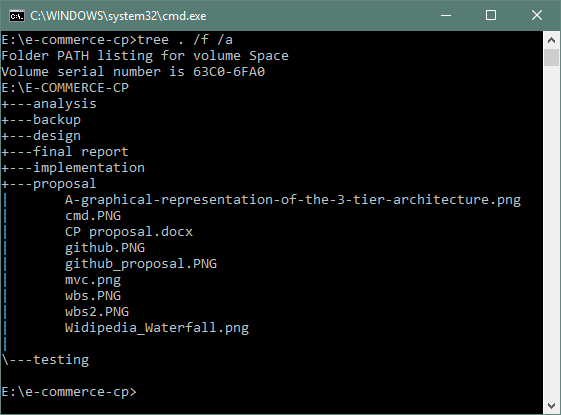


Figure 9: Local directory of project

## **Requirements**

**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.

**Non-Functional requirement**

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

## **MoSCoW prioritization**

I have prioritized my features using MoSCoW prioritization technique.

**Note**

**M**=must have

**S**=should have

**C**=could have

**W**=won’t have

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

**Non-functional requirement**

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Requirement** | **Description** |
|  | **Verification** | **To verify customer and store.** |
|  | **Security** | **Protect user and system from hacking.** |
|  | **User friendly** | **To provide smooth experience.** |
|  | **Efficient** | **Make system fast and reliable.** |

**Class Diagram**

Class diagram describes the attributes and operations of a system. It represents the static view of an application. Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram. (Anon., n.d.)

**Justifications:**

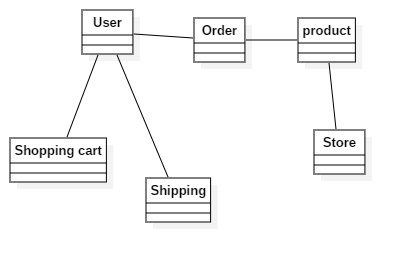
The Class Diagram is to model the static view of an application. It is forward and reverse engineering. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction. It describes the responsibilities of a system.

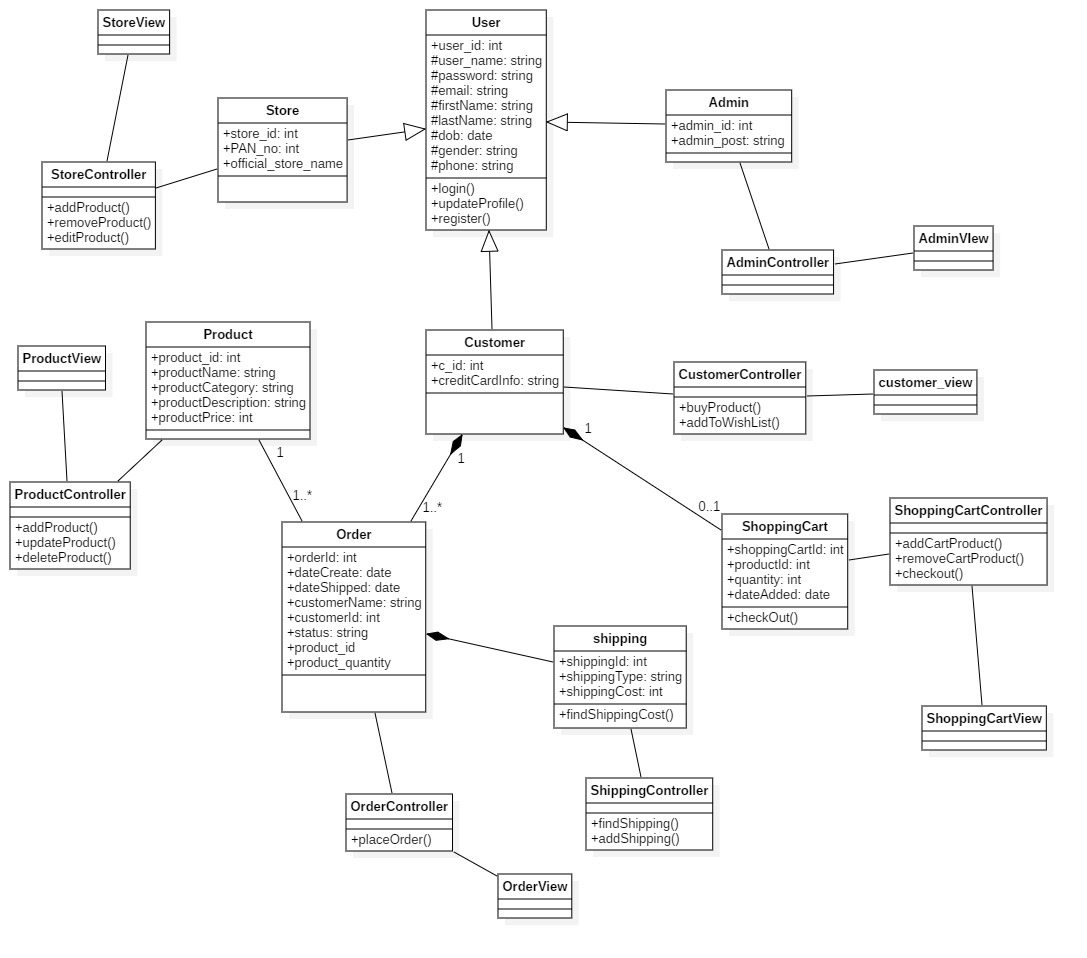
**Class Notation**

A class notation consists of three parts:

1. Class Name
   * Determines the name of class and shown at top.
2. Class Attributes
   * Attributes are variables of any class (data members) in code
   * Data type of variable are shown after colon.
3. Class Operations
   * Operations are shown in the third partition. It defines behavior of a class.
   * After colon return type of method parameters are shown following the parameter name.

**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. (htt)

**Use case diagram**

It shows the functionality of a system. It describes how a user's react with a system.

A use case diagram is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. To build the use case diagram, a set of specialized symbols and connectors are to be used. (Anon., n.d.)

**Justifications:**

The use case diagram describes a set of actions that some system can perform in collaboration with one or more external users of the system, gather the requirements of a system, identifies and clarifies the relationships between and among the actors and the use cases and identifies the external and internal factors influencing the system.

**Advantages:**

* It helps to collect the functional requirements of a system.
* It can serve as the basis for the estimating, scheduling and validating effort.
* It can also collect additional behavior that can improve system robustness.

**Disadvantages:**

* They do not capture the non-functional requirements easily.
* Difficult to manage scenario.
* It has poor identification of structure and flow.

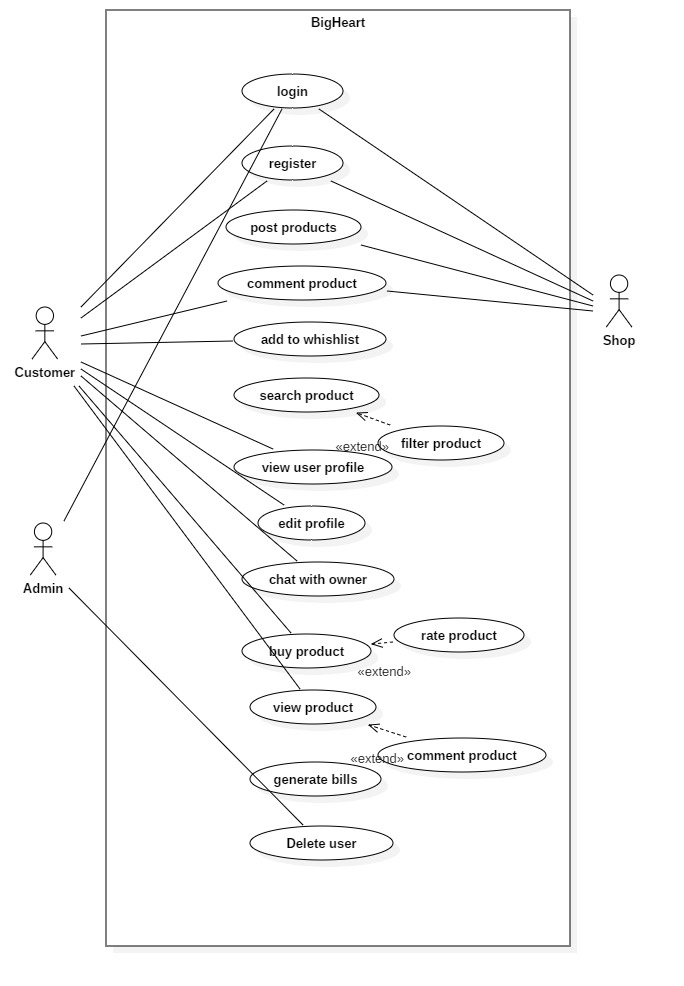
**Use case notation**

Actors: Users that interacts with the system.

System: That consist different use cases.

Use case: action that is performed by actor.

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



# **ER Diagram:**

An entity-relationship diagram is a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities. It is a conceptual and representational model of data used to represent the entity framework infrastructure.

**Justifications:**

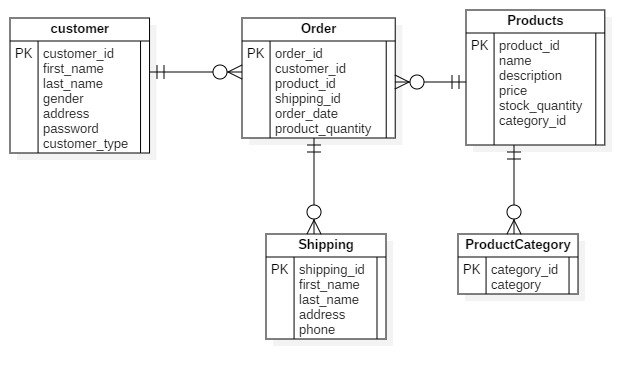
The ER Diagram is to visualize database design ideas, so we have a chance to identify the mistakes and design flaws, and to make correction before executing the changes in database. By visualizing a database schema with an ERD, we have a full picture of the entire database schema through which we can easily locate entities, view their attributes and to identify the relationships they have with others.

**Advantages:**

* It is very simple if we know relationship between entities and attributes.
* It is better visual representation.
* It is an effective communication tool for database designer.

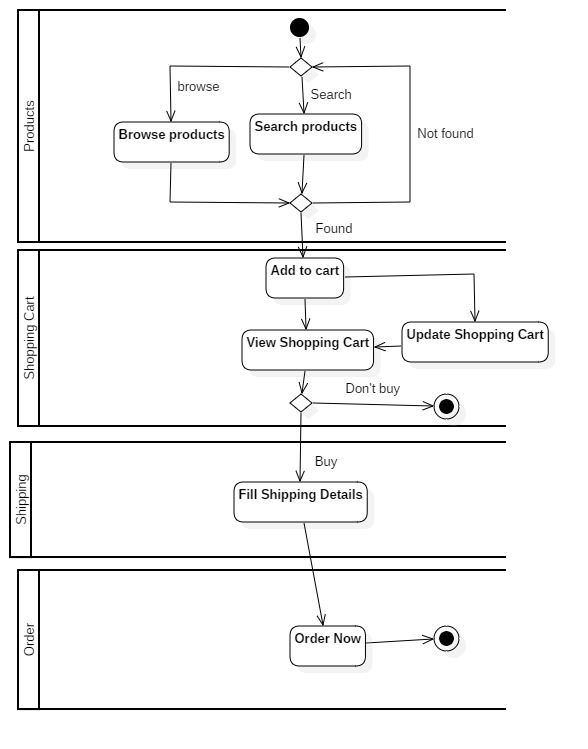
**Disadvantages:**

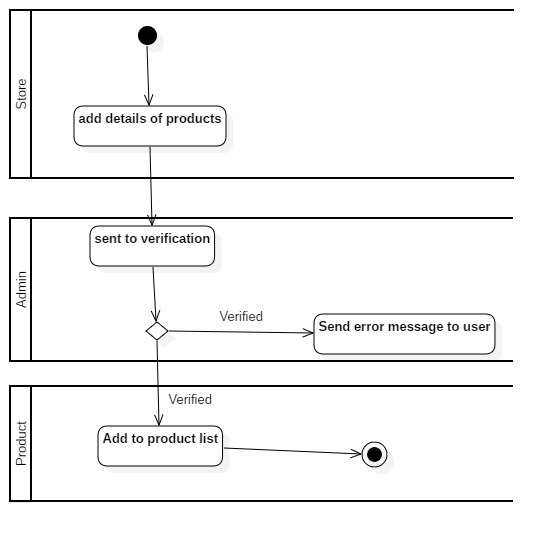
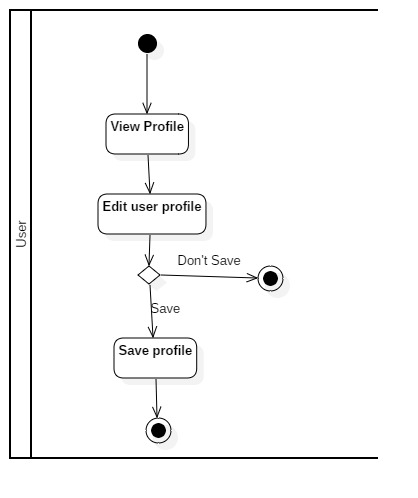
* It has limited constraints and specification.
* Information can be hidden in ER model.
* It is difficult to show data manipulation in ER model.



**Activity diagram**

I had used activity diagram to illustrate the flow of system. An activity diagram focuses on condition of flow and the sequence in which it happens. It helps to understand the flow of programs on a high level. It also enables them to figure out constraints and conditions that cause events.





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

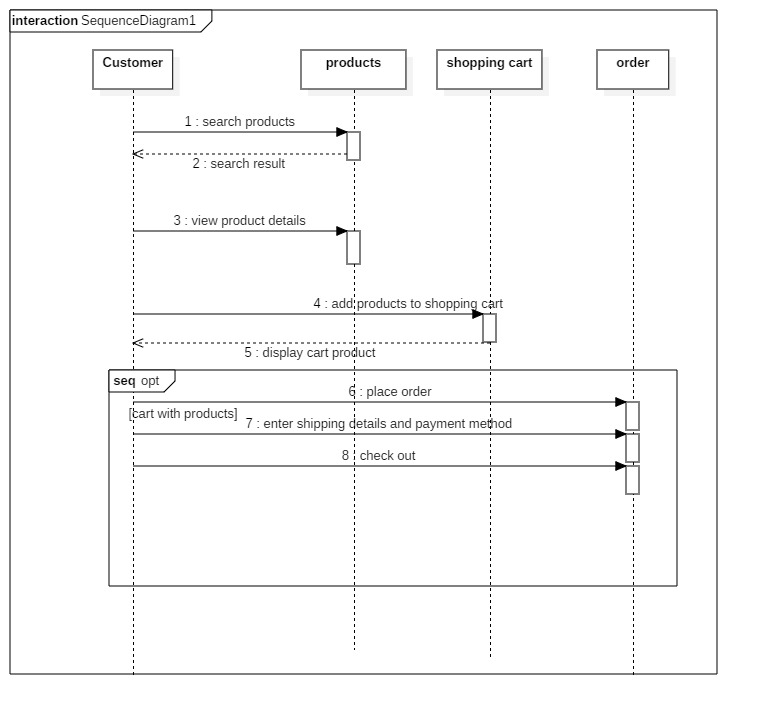
**Sequence Diagram**

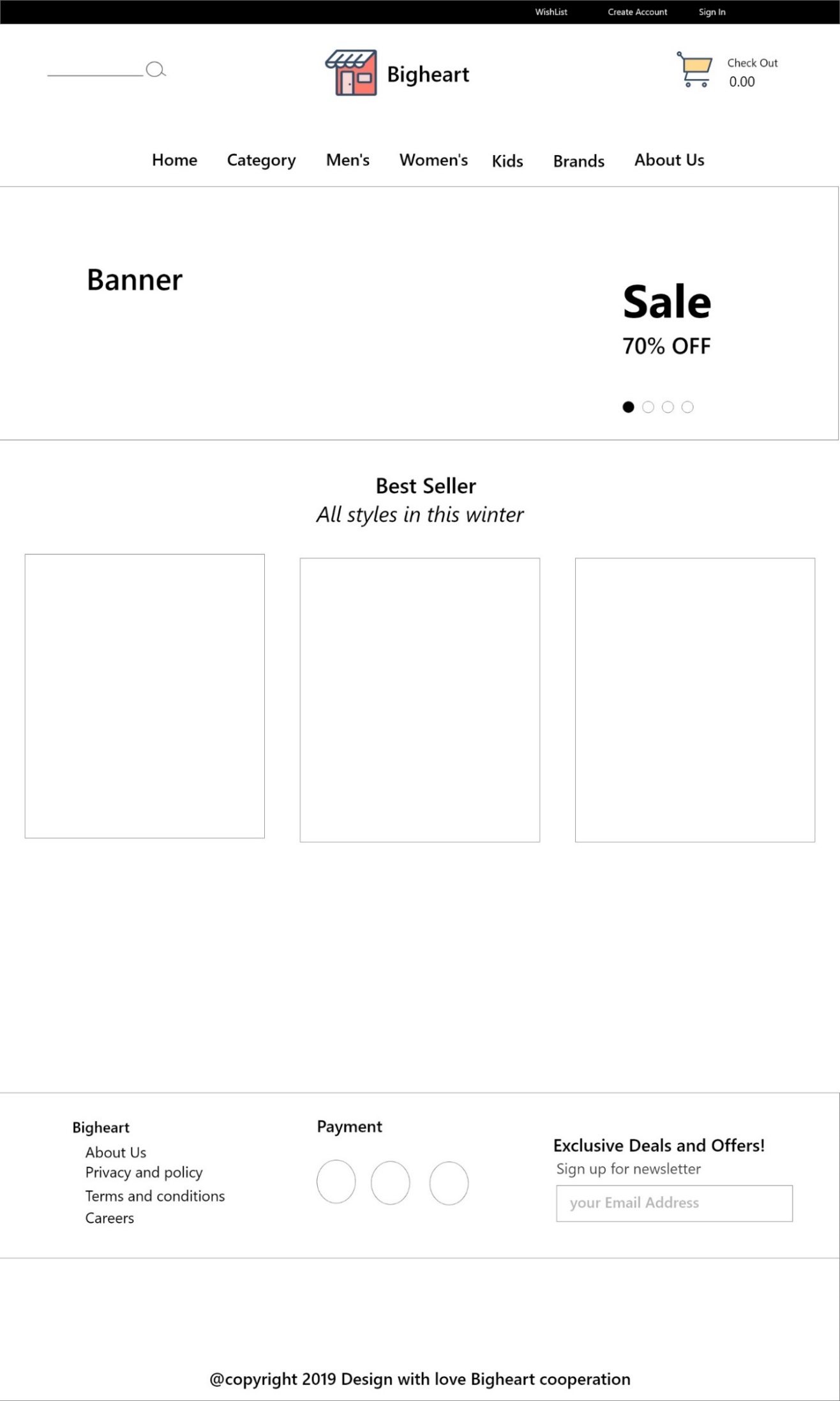
A sequence diagram is interaction between objects in a sequential order. Sequence diagrams describe how and in what order the objects in a system function.

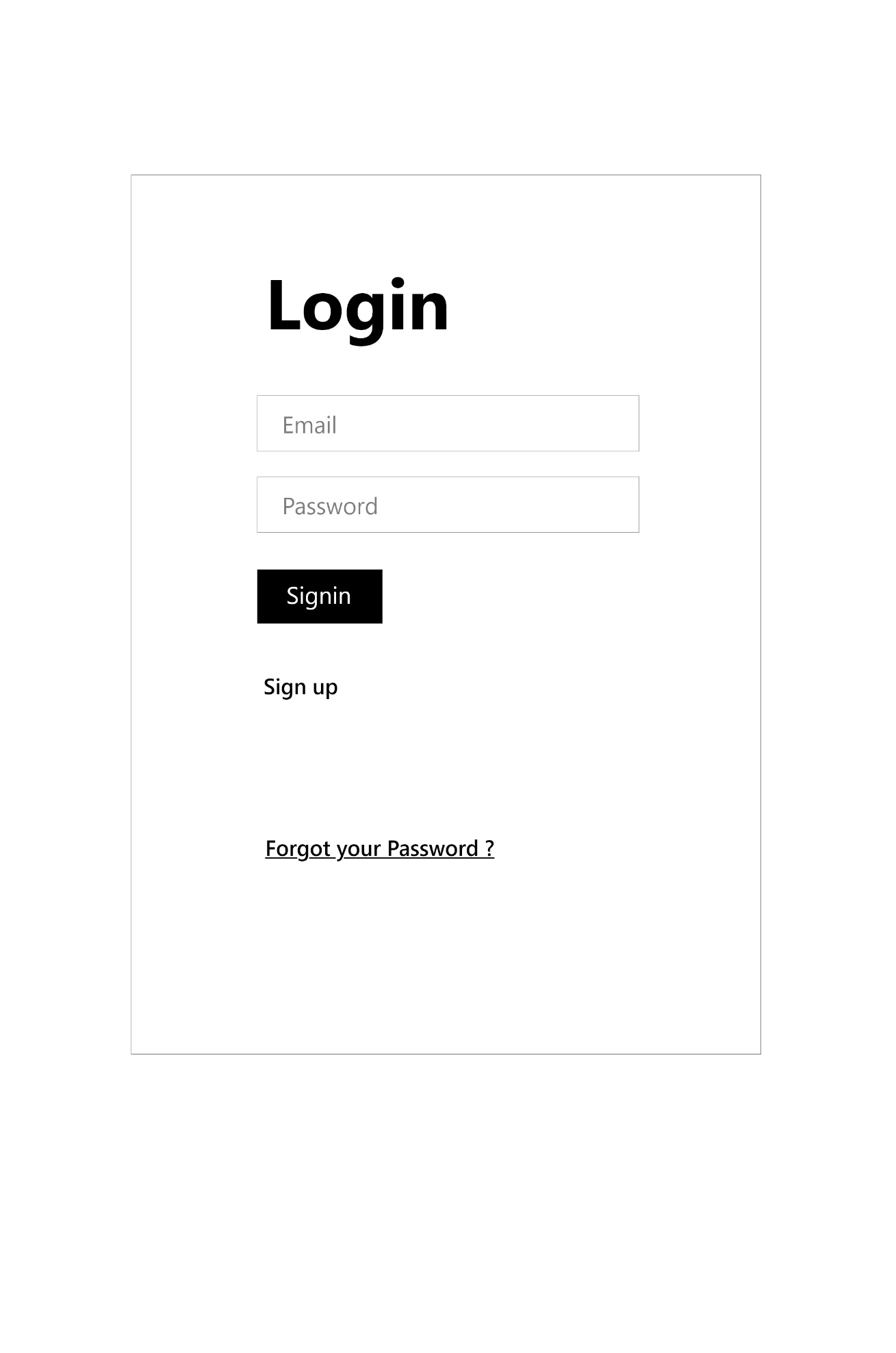
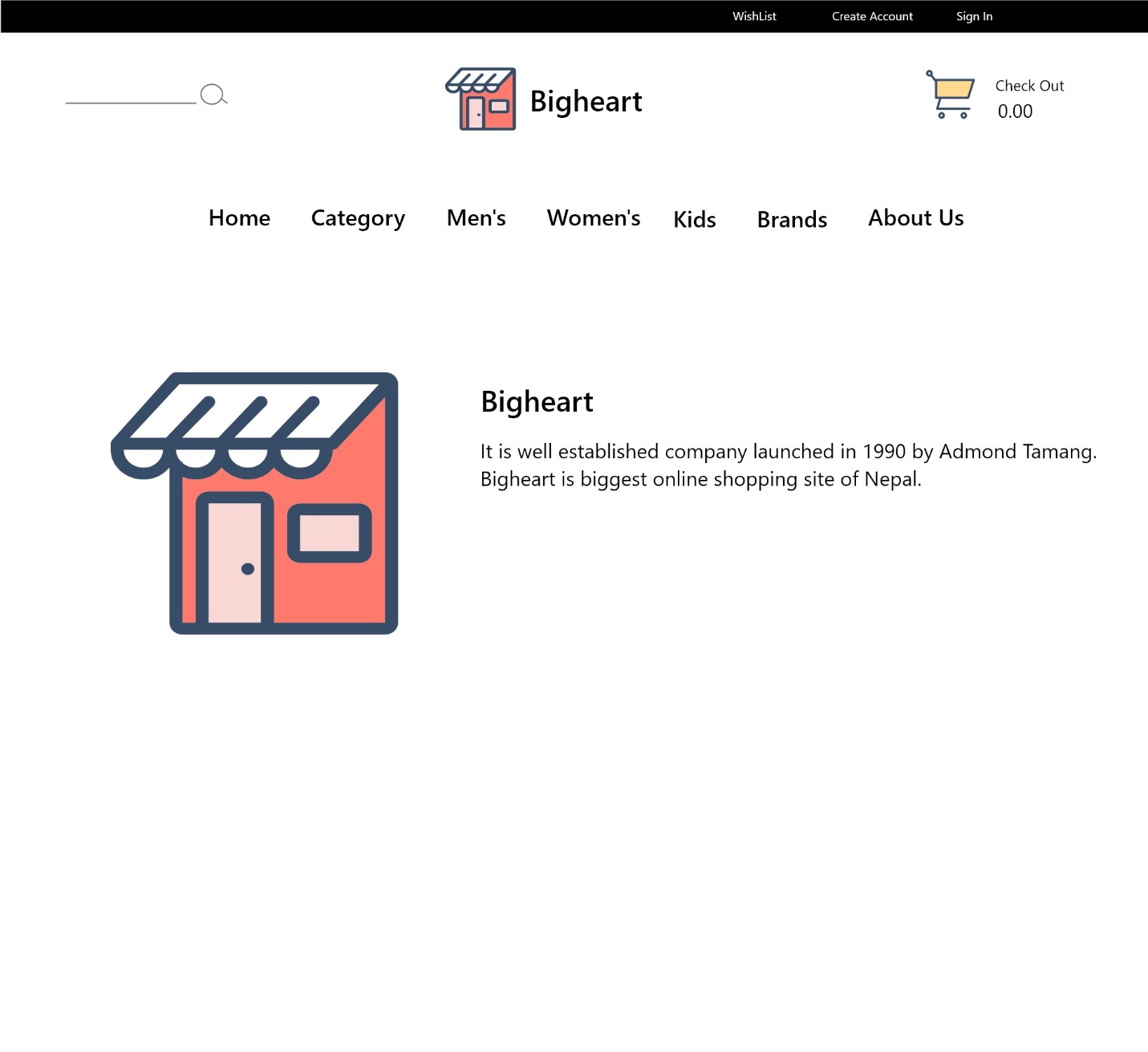
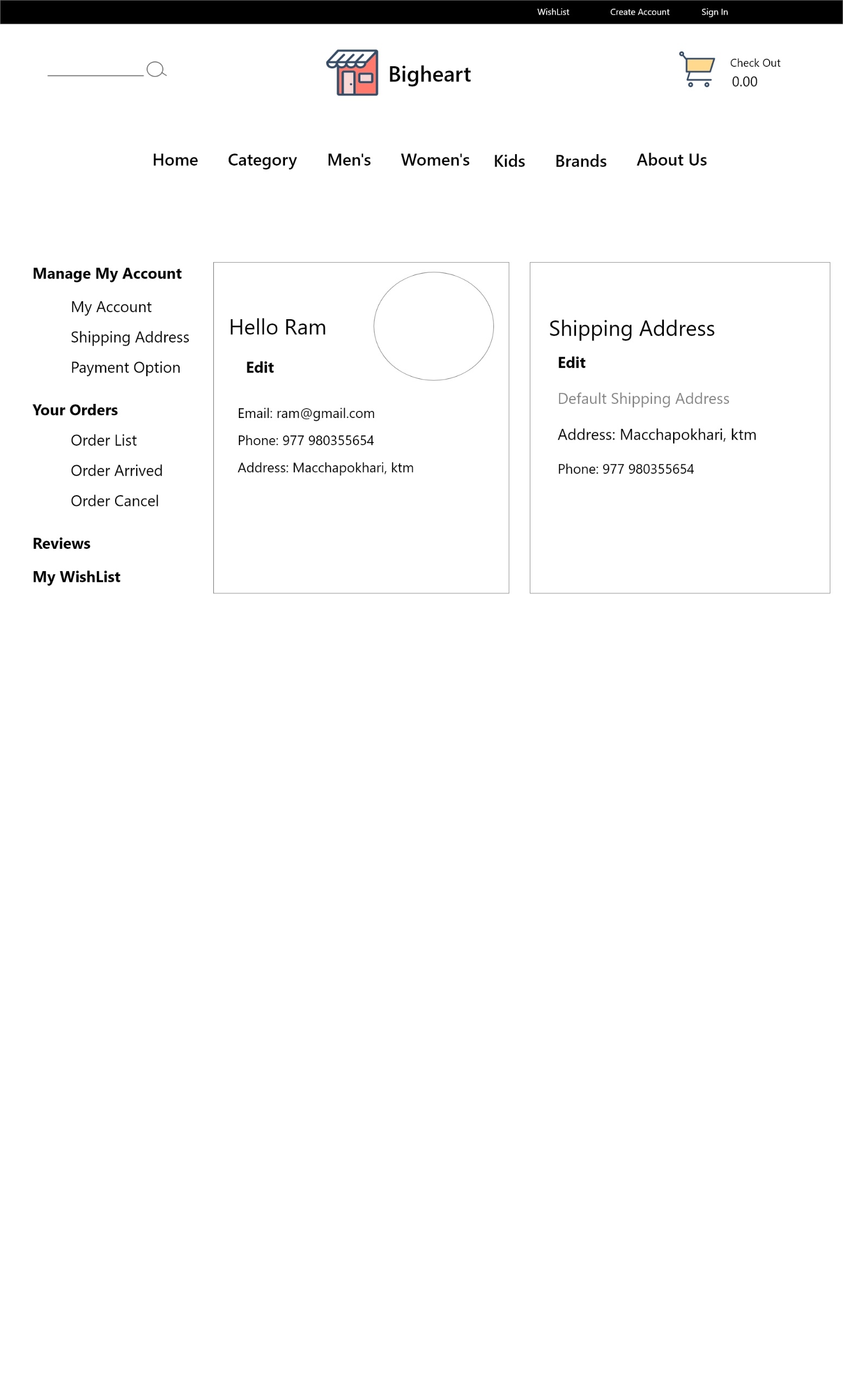
sequence diagrams can be useful references for businesses and other organizations. Try drawing a sequence diagram to:

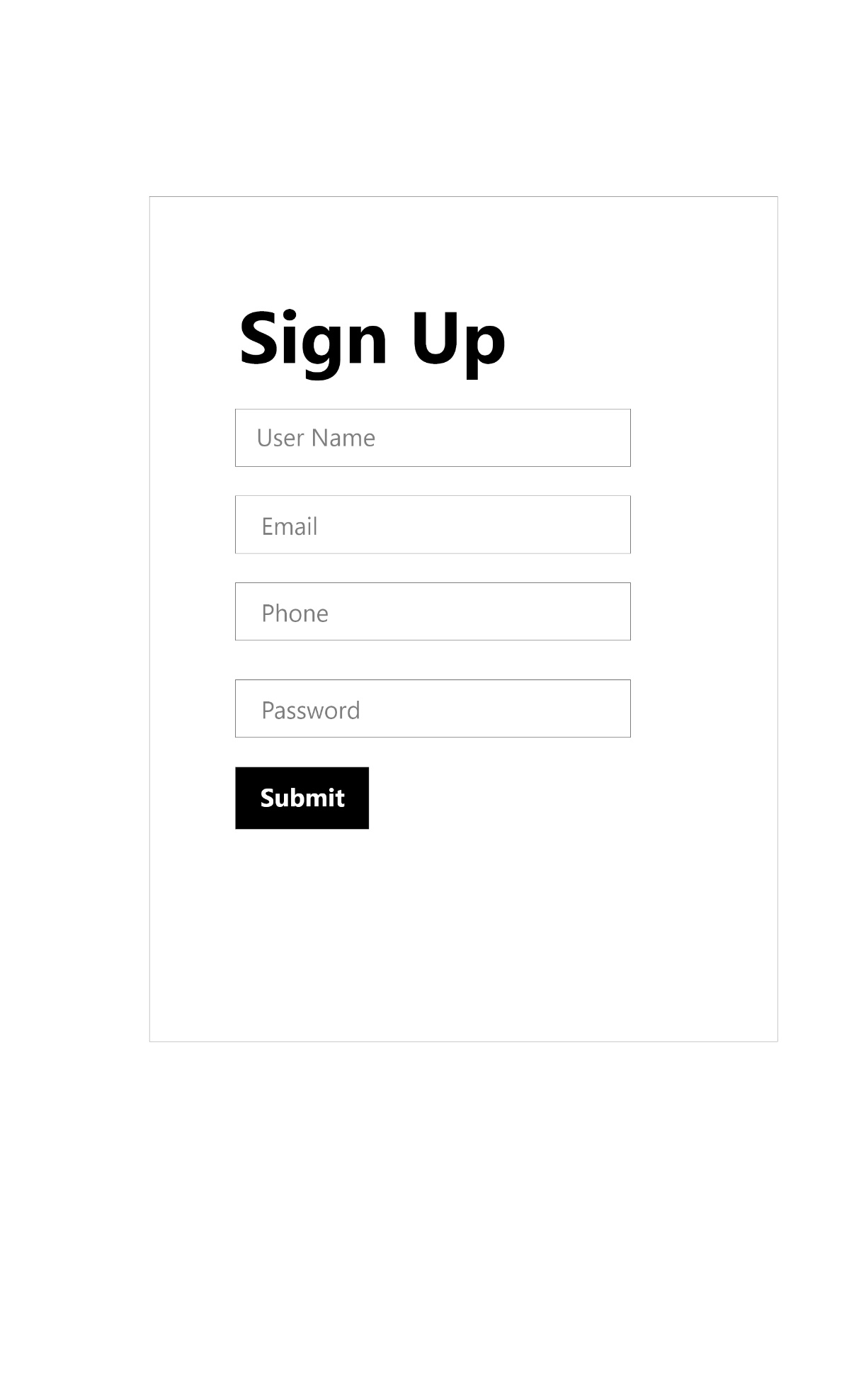
The reason of using sequence diagram are:

* 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.

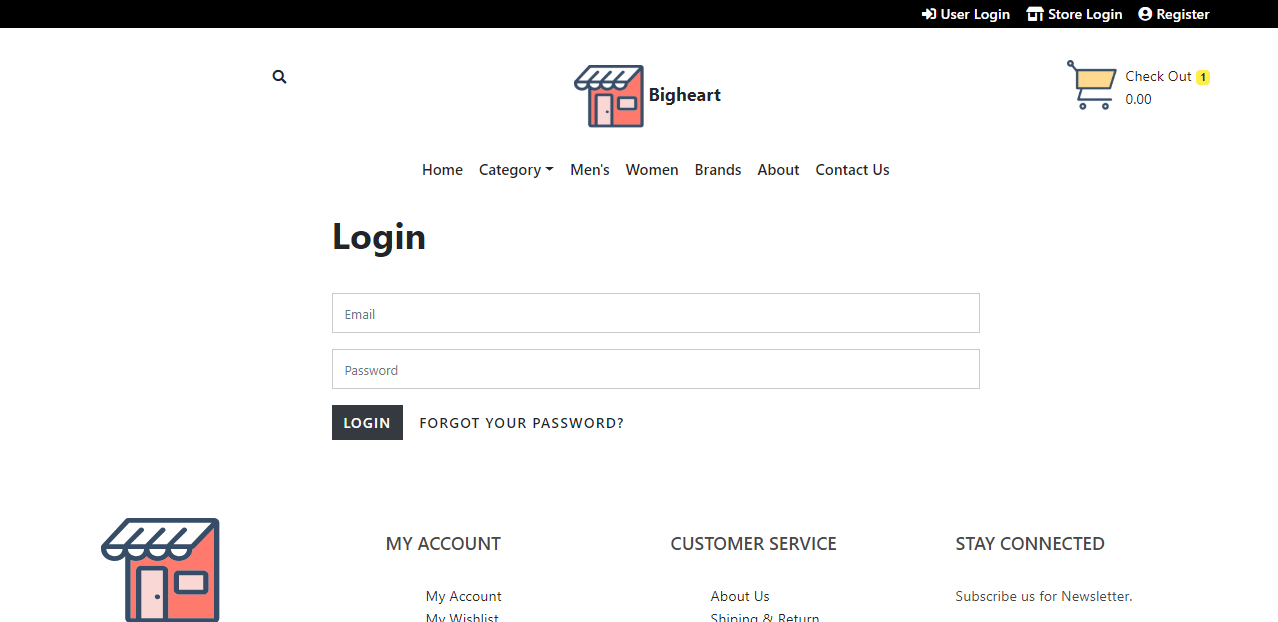
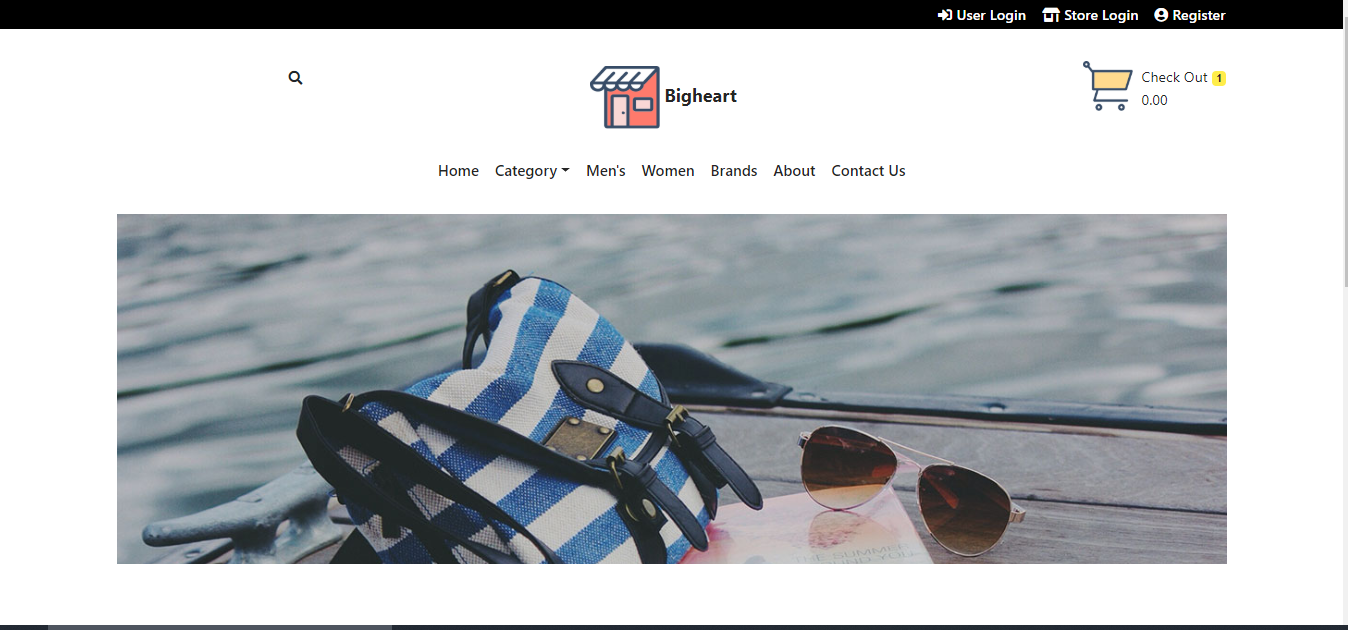
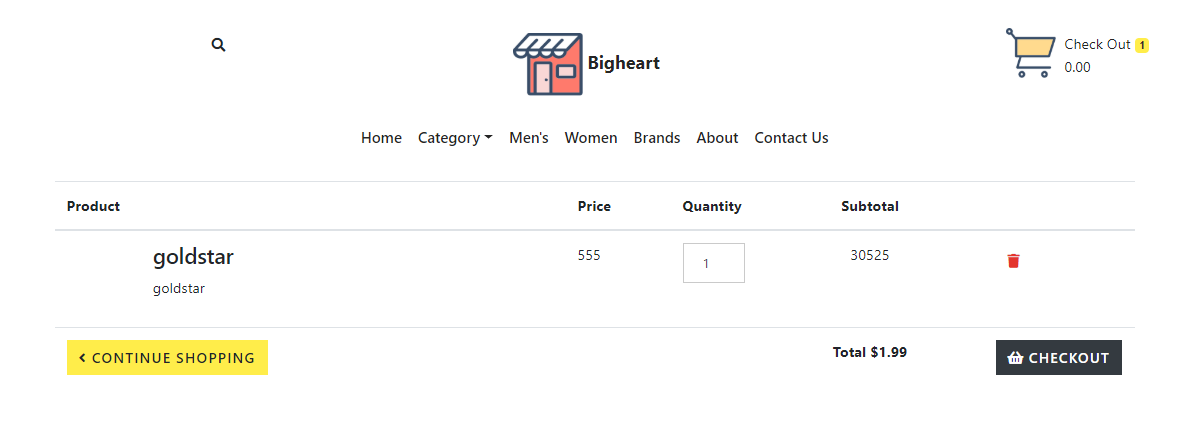


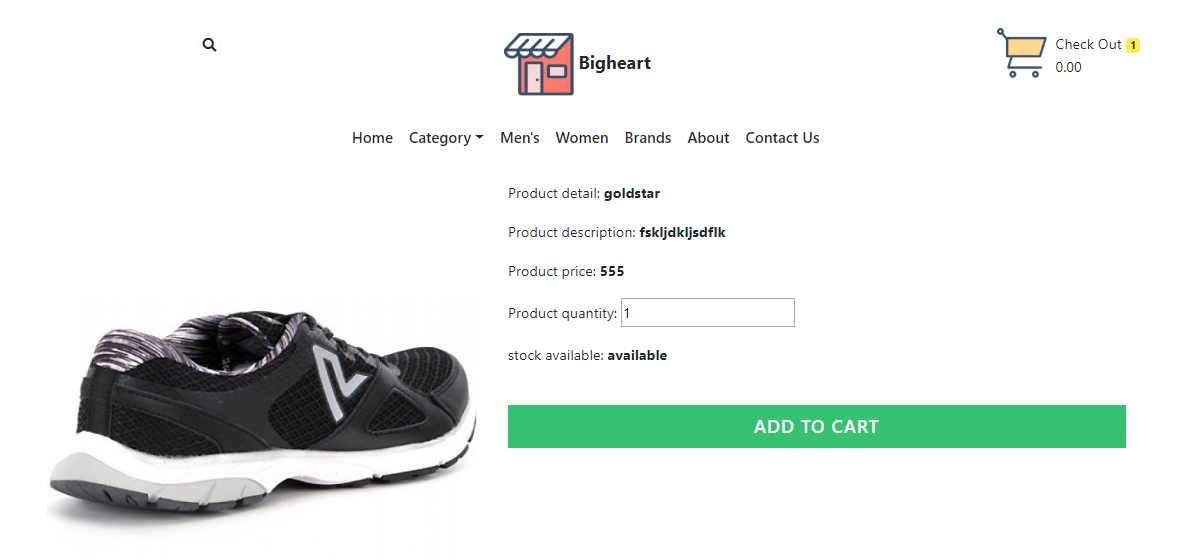
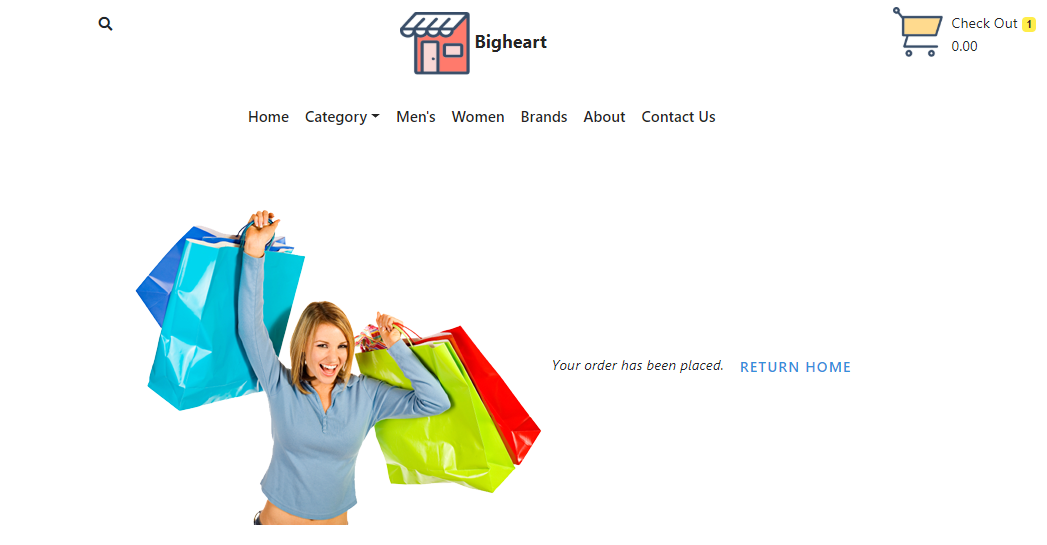
Prototypes 





**UI design and code**

****

****

**Testing**

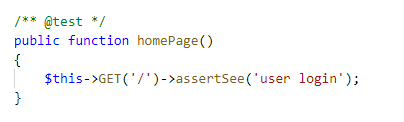
Testing is defined as testing of an eCommerce (online shopping) application. It helps in the prevention of errors and adds value to the product by ensuring conformity to client requirements.

The objective of testing is to ensure

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

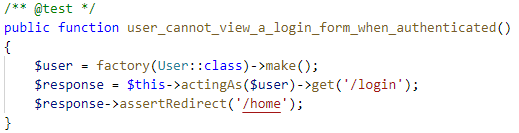
**Test cases**

**Test Case 1**

****

|  |  |
| --- | --- |
| **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 2**



|  |  |
| --- | --- |
| **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 3**

|  |  |
| --- | --- |
| **Test case no** | 3 |
| **Purpose of test** | To check whether product can be added without authentication |
| **Test data** |  |
| **Expected result** | Redirect to store login |
| **Actual Result** | Redirected to store login |
| **Class** | testLogin, UserFactory |

**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 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 |

**Future Work**

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.

# **Conclusion of the project**

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.

In this project the ‘add to cart’ functionality made by raw PHP without using any plugin and payment system is not included properly in this project because to complete payment system we require a payment gateway.

# References

(n.d.). Retrieved from https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/

interserver. (2018, 12 255). Retrieved from interserver: https://www.interserver.net/tips/kb/mvc-advantages-disadvantages-mvc/