**Recondition House Management System**



**Amit Pradhan**

**00172888**

**Computing Project**

**Level 5 Diploma in Computing**

**Email :** [amitpradhan195@gmail.com](mailto:amitpradhan195@gmail.com)

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

***Kathmandu, Nepal***

01/27/2019

**Submitted to**: Kiran Rana

Table of Contents

[Introduction: 3](#_Toc536316458)

[Functional and Non-Functional: 3](#_Toc536316459)

[Requirement Analysis: 4](#_Toc536316460)

[USE CASE DIAGRAM 5](#_Toc536316461)

[Initial Class Diagram 6](#_Toc536316462)

[ER Diagram 7](#_Toc536316463)

[Prototype: 8](#_Toc536316464)

[UI Design: 12](#_Toc536316465)

[Conclusion: 16](#_Toc536316466)

[References 16](#_Toc536316467)

**List of Figures:**

[Figure 1: Use Case Diagram 6](#_Toc2020502)

[Figure 2: Initial Class Diagram 8](#_Toc2020503)

[Figure 3: Final class diagram 9](file:///F:\%23%23Softwarican%2021st%20Batch\4th%20semester%20Notes\CP\MSWord\Analysis.docx#_Toc2020504)

[Figure 4: ER Diagram 11](file:///F:\%23%23Softwarican%2021st%20Batch\4th%20semester%20Notes\CP\MSWord\Analysis.docx#_Toc2020505)

[Figure 5: Activity Diagram 13](#_Toc2020506)

[Figure 6: Login Page Prototype 14](#_Toc2020507)

[Figure 7: Sign Up Page Prototype 14](#_Toc2020508)

[Figure 8: Home Page Prototype 15](#_Toc2020509)

[Figure 9: Gallery Page Prototype 15](#_Toc2020510)

[Figure 10: Bookings Page Prototype 16](#_Toc2020511)

[Figure 11: Categories Page Prototype 16](#_Toc2020512)

[Figure 12: About Us Page Prototype 17](#_Toc2020513)

[Figure 13: Admin Page Prototype 17](#_Toc2020514)

[Figure 14: Home Page User Interface 18](#_Toc2020515)

[Figure 15: Gallery Page User Interface 18](#_Toc2020516)

[Figure 16: Bookings Page User Interface 19](#_Toc2020517)

[Figure 17: Categories Page User Interface 19](#_Toc2020518)

[Figure 18: Sign Up Page 20](#_Toc2020519)

[Figure 19: Billing page 20](#_Toc2020520)

[Figure 20: About Us Page User Interface 21](#_Toc2020521)

[Figure 21: Login User Interface 21](#_Toc2020522)

[Figure 22: Sign Up User Interface 22](#_Toc2020523)

# Introduction:

In this document, I have analyzed the Functional and Non-Functional requirements, and filtered the requirements with MoSCoW Prioritization.

After completing the requirements analysis, I have started designing modeling such as Use Case Diagrams, Initial Class Diagrams, E-R Diagrams and Prototype. At last I have started designing UI design. In this phase, the tools that I have used are Star UML, Balsamiq and Mobirise for the outcome.

# Functional and Non-Functional:

Functional requirements are the requirements that specifies a particular behavior of function of the system when certain conditions are met .It defines what a system is supposed to do.

Non-Functional requirements are the requirements that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. It defines how a system is supposed to be. (Anon., n.d.)

In the below table, the functional and non-functional requirements are identified and then they are prioritized with MoSCoW prioritization.

Index used in the table below are:

**F**=Functional requirements

**NF**=Non-Functional requirements

**M**=Must Have

**S**=Should Have

**C**=Could Have

# Requirement Analysis:

|  |  |  |
| --- | --- | --- |
| **S. No** | **Requirements** | **MoSCoW** |
|  | User Login | **M** |
|  | Insert Item | **M** |
|  | Update Item | **S** |
|  | Delete Item | **S** |
|  | View Information | **M** |
|  | Book Item | **M** |
|  | Search Item | **S** |
|  | Purchase Item | **M** |
|  | Sell Item | **M** |
|  | Exchange Item | **M** |
|  | Registration | **M** |
|  | Home Page | **S** |
|  | Edit Profile | **S** |
|  | Gallery | **C** |
|  | User Review | **C** |
|  | Log Out | **M** |
|  | Generate Bill | **M** |
|  | Store Customer Information | **M** |
|  | Forgot Password | **C** |
|  | Slider | **C** |
|  | Contact Information | **S** |
|  | Visitor Counter | **C** |
|  | Store Bill | **M** |
|  | Manage Inventory | **M** |
|  | Authentication | **M** |

**Functional Requirements Table:**

Table 1: Requirement Analysis Table

**Non-Functional Requirements Table:**

|  |  |  |
| --- | --- | --- |
| **S. No** | **Requirements** | **MoSCoW** |
|  | Security | **S** |
|  | Performance | **S** |
|  | Availability | **S** |
|  | Reliability | **S** |
|  | Maintainability | **S** |
|  | Usability | **S** |
|  | Data Integrity | **S** |
|  | Supportability | **S** |

Table 2: Non-Functional Requirements

# USE CASE DIAGRAM

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.

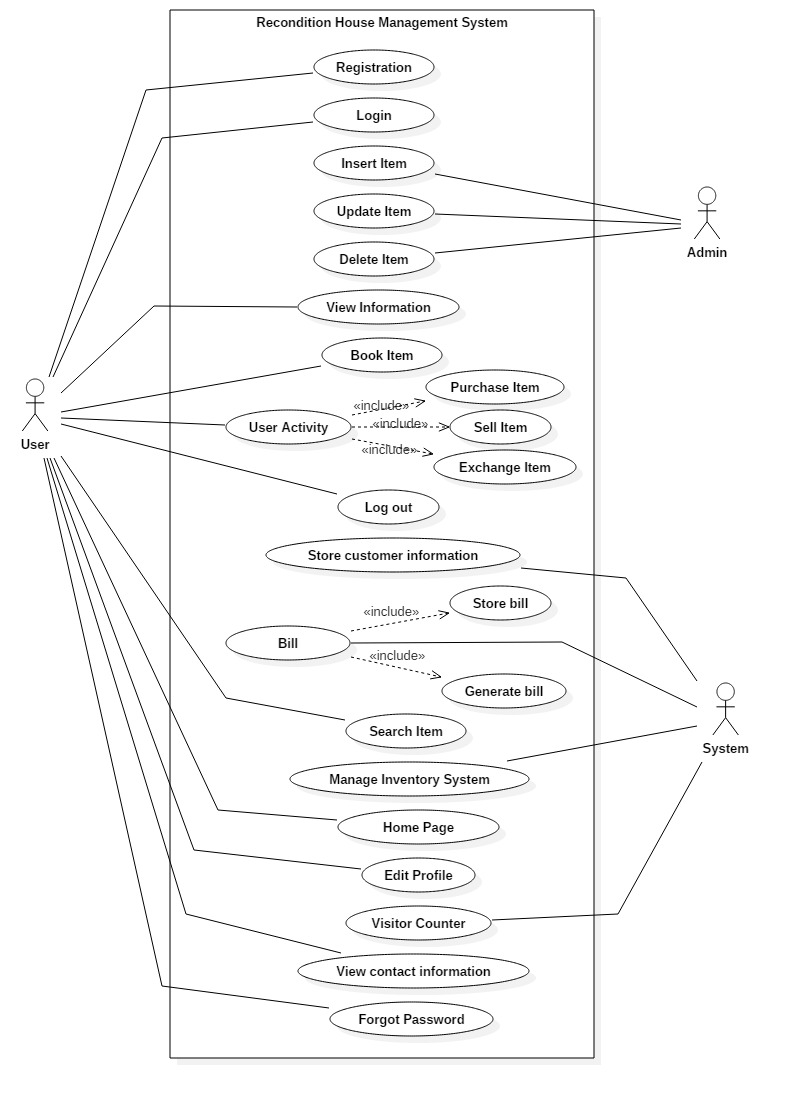
****

Figure 1: Use Case Diagram

# Class Diagram:

Class diagram is a static diagram. 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.

**Advantages:**

* Class diagram are simple and fast to read.
* It gives you a sense of orientation.
* They provide detailed insight into the structure of our systems.

**Disadvantages:**

* They do not have dynamic model.
* It shows only collaboration among the elements of the static view.

# Initial Class Diagram:

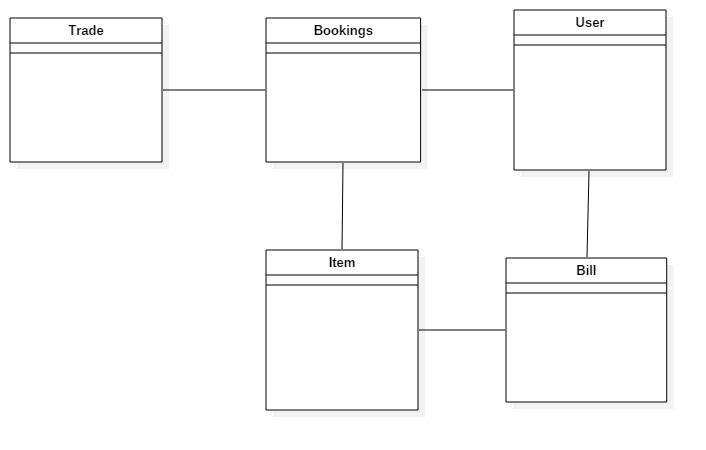


Figure 2: Initial Class Diagram

# Final Class Diagram:

Figure 3: Final class diagram

# ER Diagram:

An **entity-relationship diagram (ERD**) is a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities. An **ERD** 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.

# 

Figure 4: ER Diagram

# Activity Diagram:

An **activity diagram** is a graphical representation of an executed set of procedural system activities and considered a state chart diagram variation. Activity diagrams describe parallel and conditional activities, use cases and system functions at a detailed level. (Anon., n.d.)

**Justifications:**

Here, I have drawn activity diagram to show message flow from one activity to another. It captures the dynamic behavior of the system. It is also used to draw the activity flow of a system, to describe the sequence from one activity to another, and to describe the parallel, branched and concurrent flow of the system.

**Advantages:**

* Since it is the most user-friendly diagram. So, generally regarded as an essential tool.
* It helps to display multiple conditions and actors within a work flow through the use of swim lanes.
* These diagrams are normally easily comprehensive for both analysts and stakeholder.

**Disadvantages:**

* These diagrams can lead the over complex which might affect the user-friendly nature.
* These diagrams do not give the detail about how object behave or collaborate.

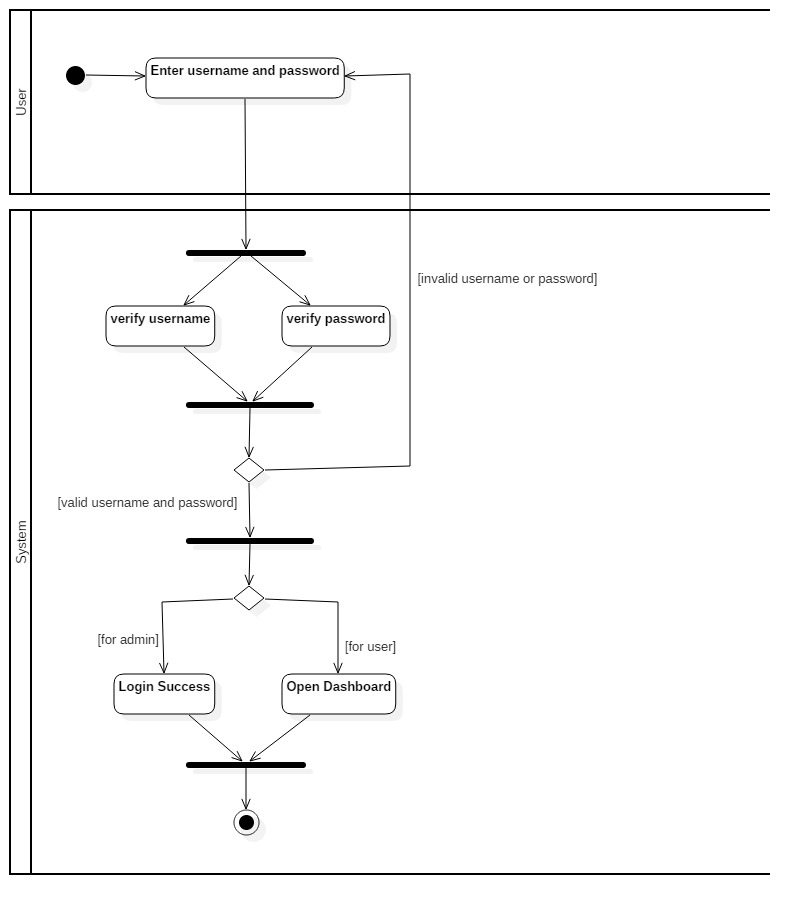
****

Figure 5: Login Activity Diagram

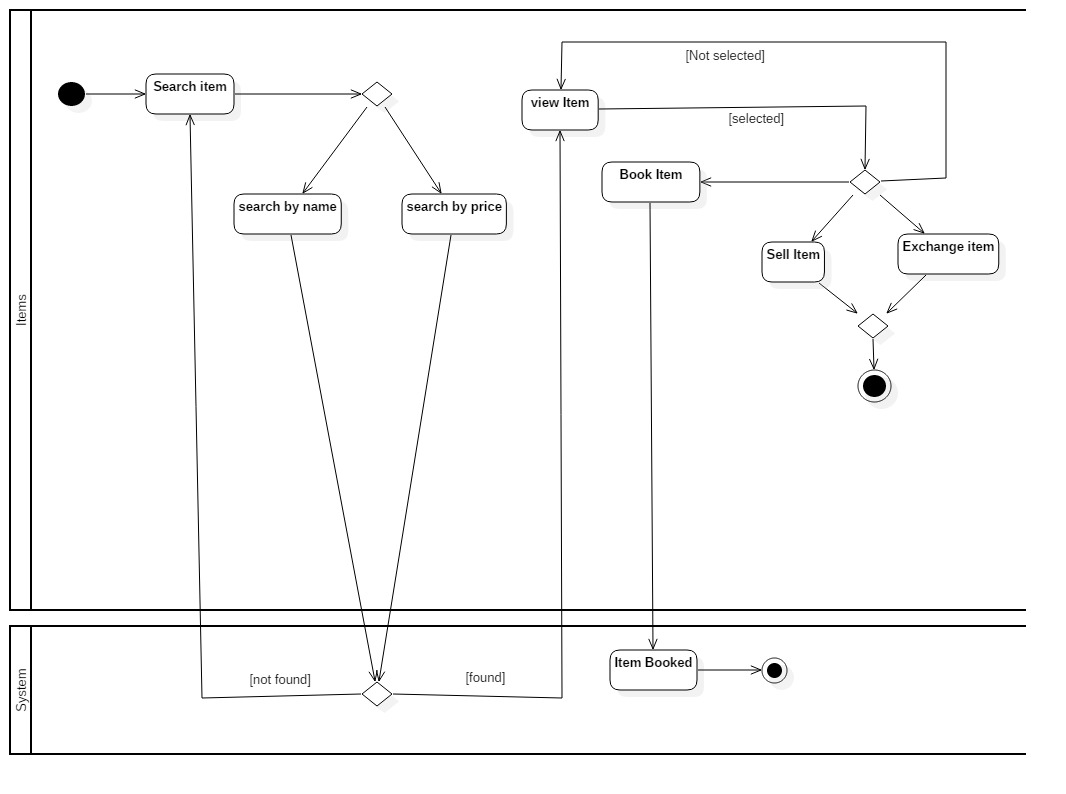


Figure 6: Item Activity Diagram

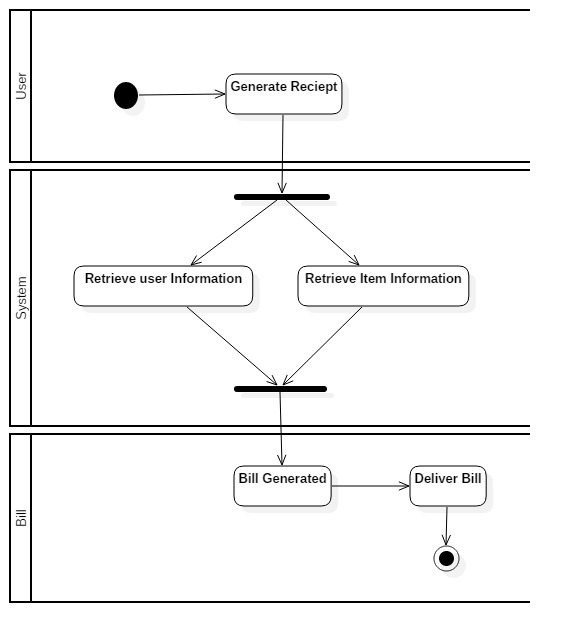


Figure 7: Bill Activity Diagram

# Prototype:

1. **Login Page:**

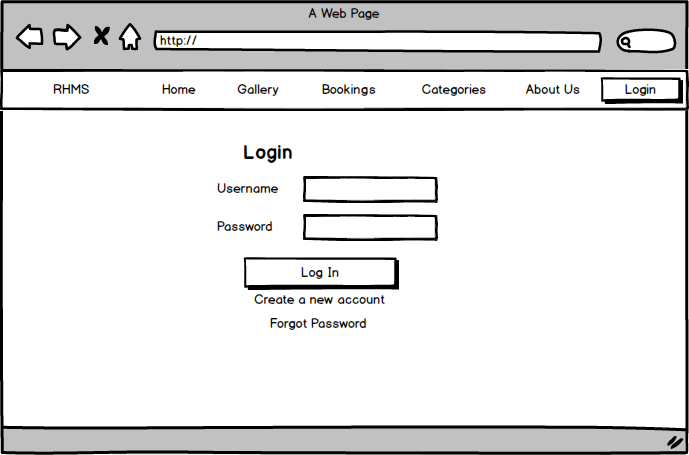
****

Figure 8: Login Page Prototype

1. **Sign Up Page:**

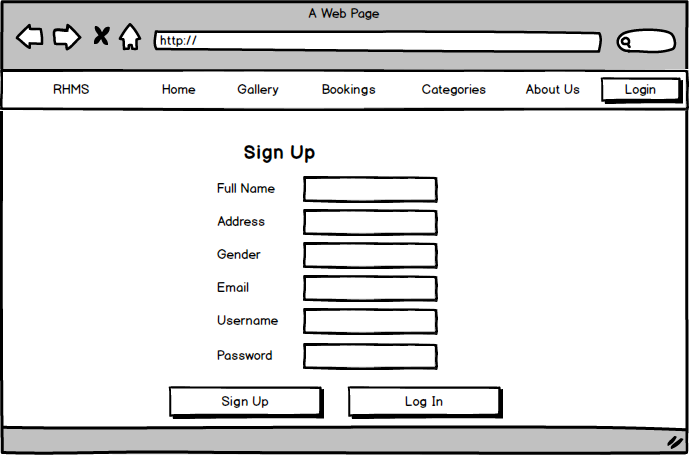
****

Figure 9: Sign Up Page Prototype

1. **Home Page:**

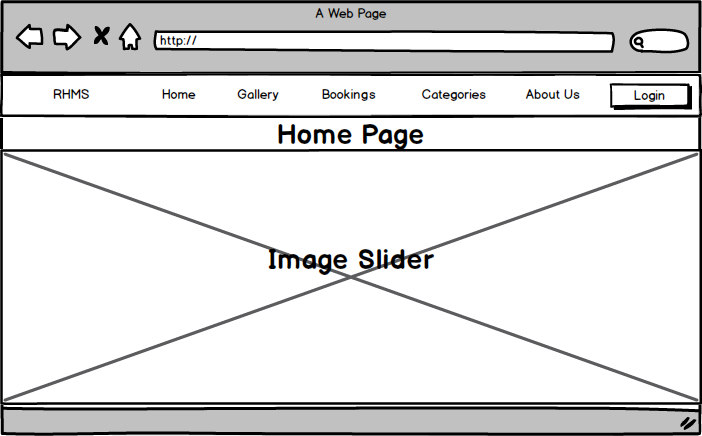
****

Figure 10: Home Page Prototype

1. **Gallery Page:**

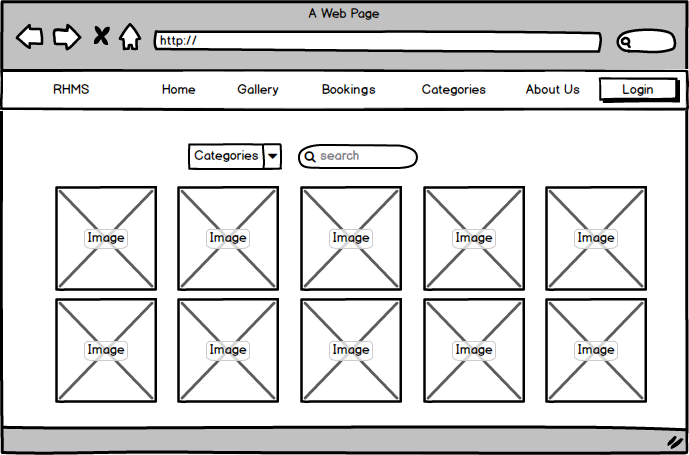
****

Figure 11: Gallery Page Prototype

1. **Bookings Page;**

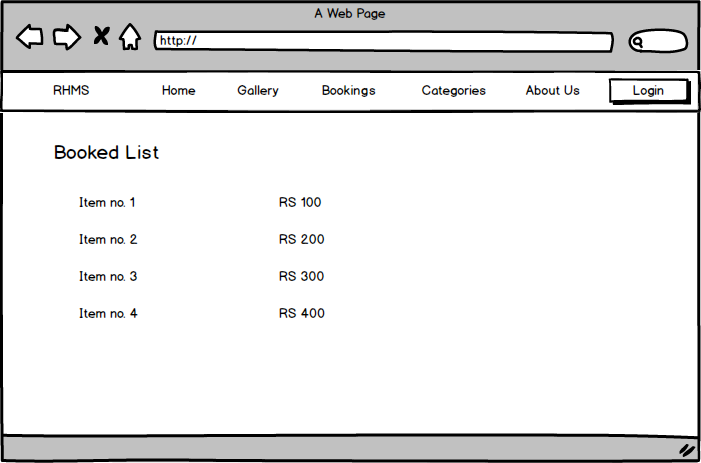
****

Figure 12: Bookings Page Prototype

1. **Categories Page:**

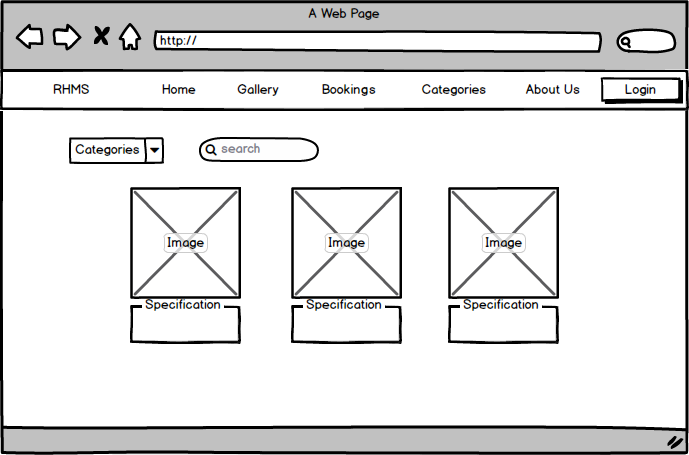
****

Figure 13: Categories Page Prototype

1. **About Us Page:**

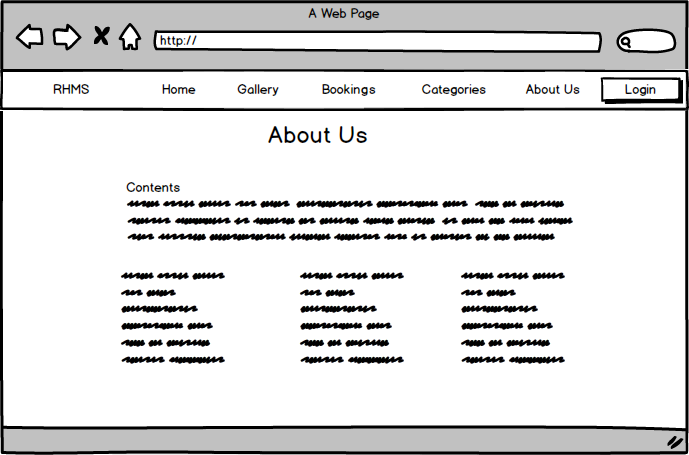
****

Figure 14: About Us Page Prototype

1. **Admin Page:**

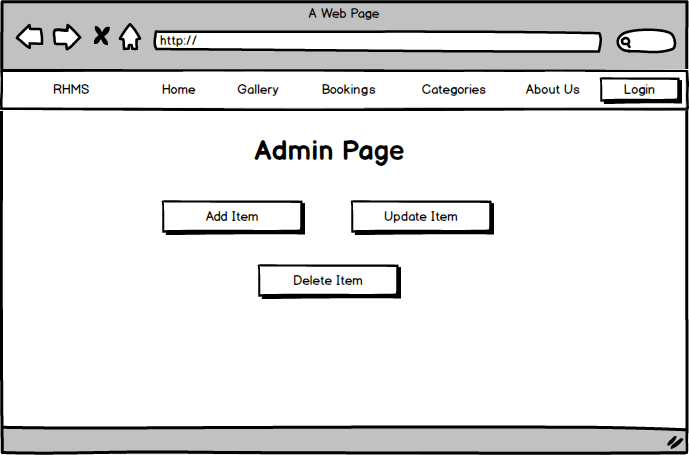
****

Figure 15: Admin Page Prototype

# UI Design:

1. **Home Page:**

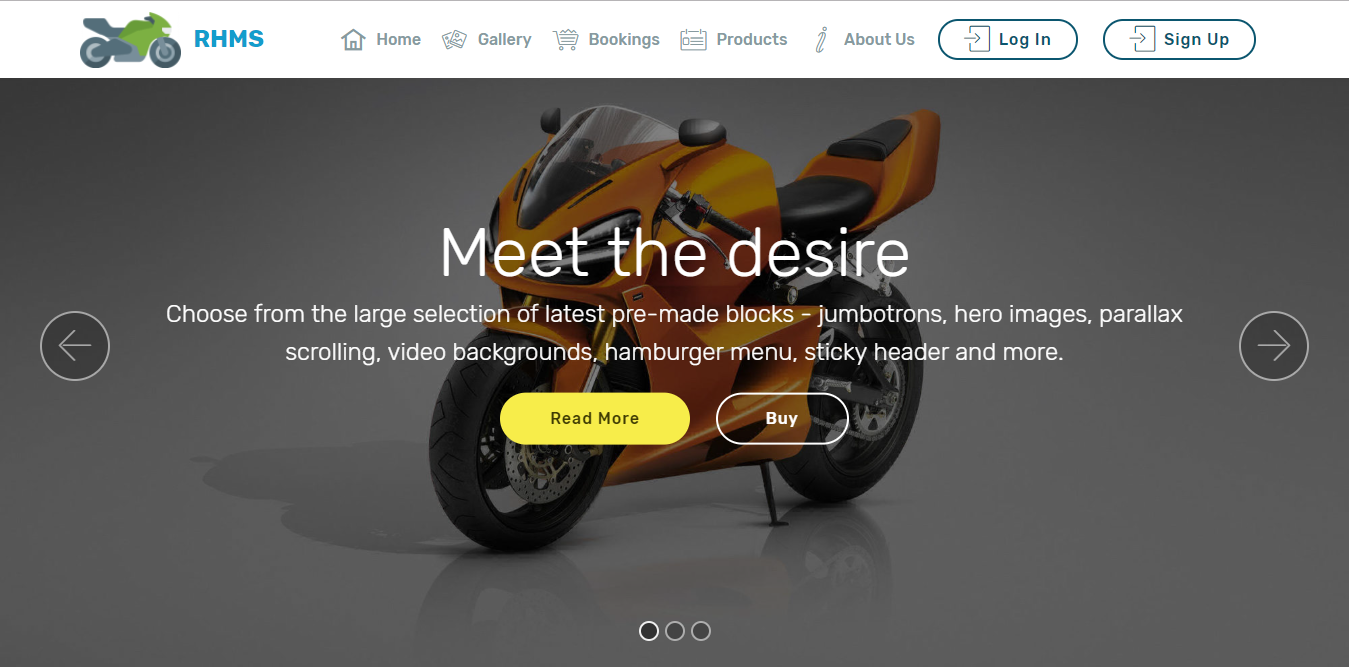
****

Figure 16: Home Page User Interface

1. **Gallery Page:**

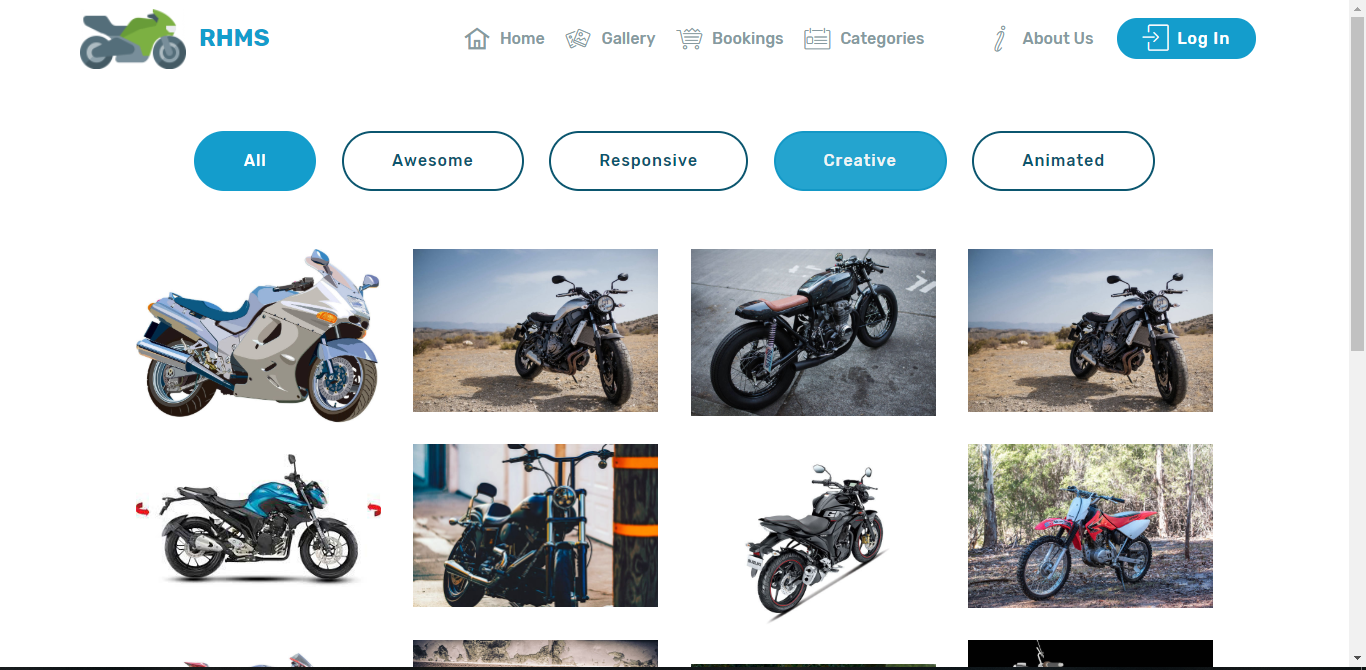
****

Figure 17: Gallery Page User Interface

1. **Bookings Page:**

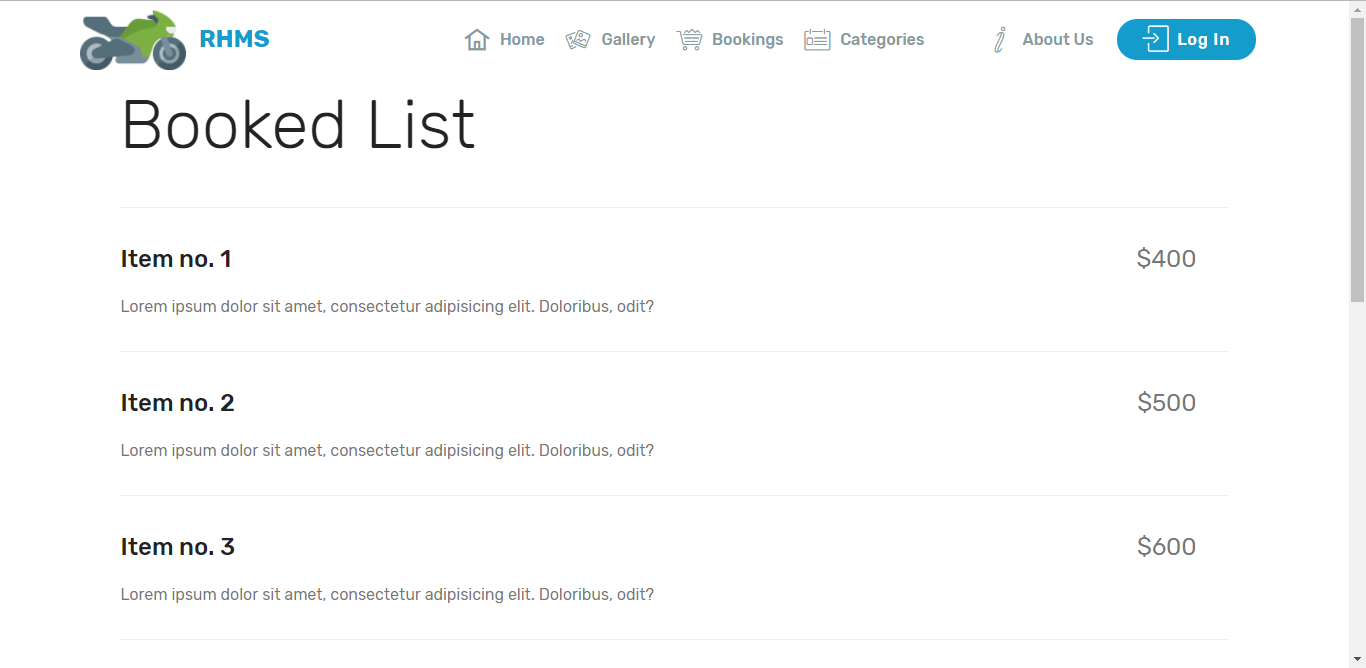
****

Figure 18: Bookings Page User Interface

1. **Categories Page:**

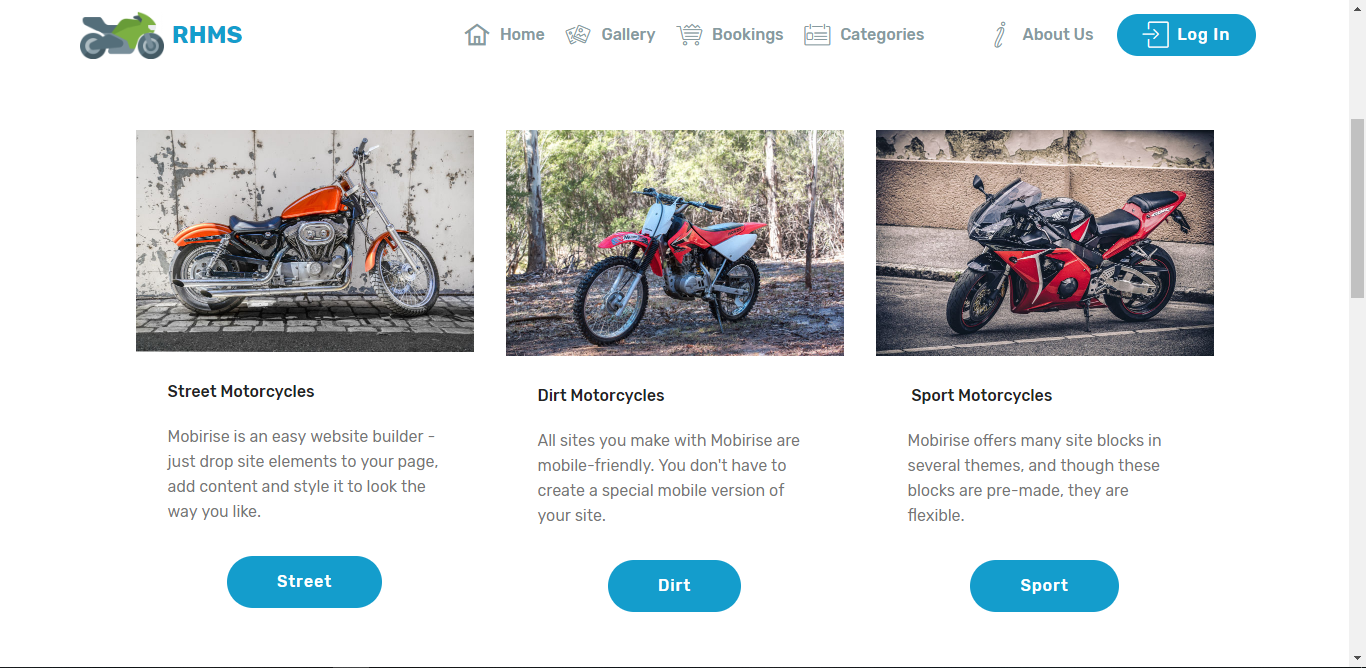
****

Figure 19: Categories Page User Interface

1. **Edit Profile page:**

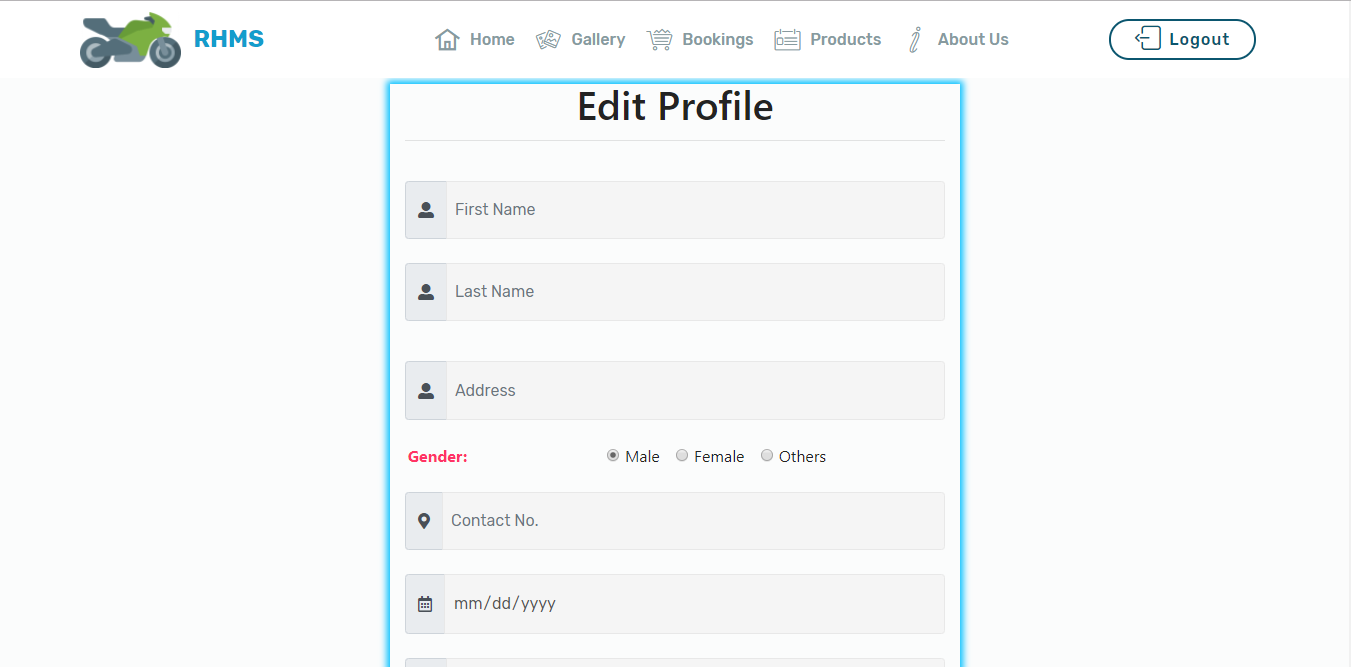
****

Figure 20: Sign Up Page

1. **Billing:**

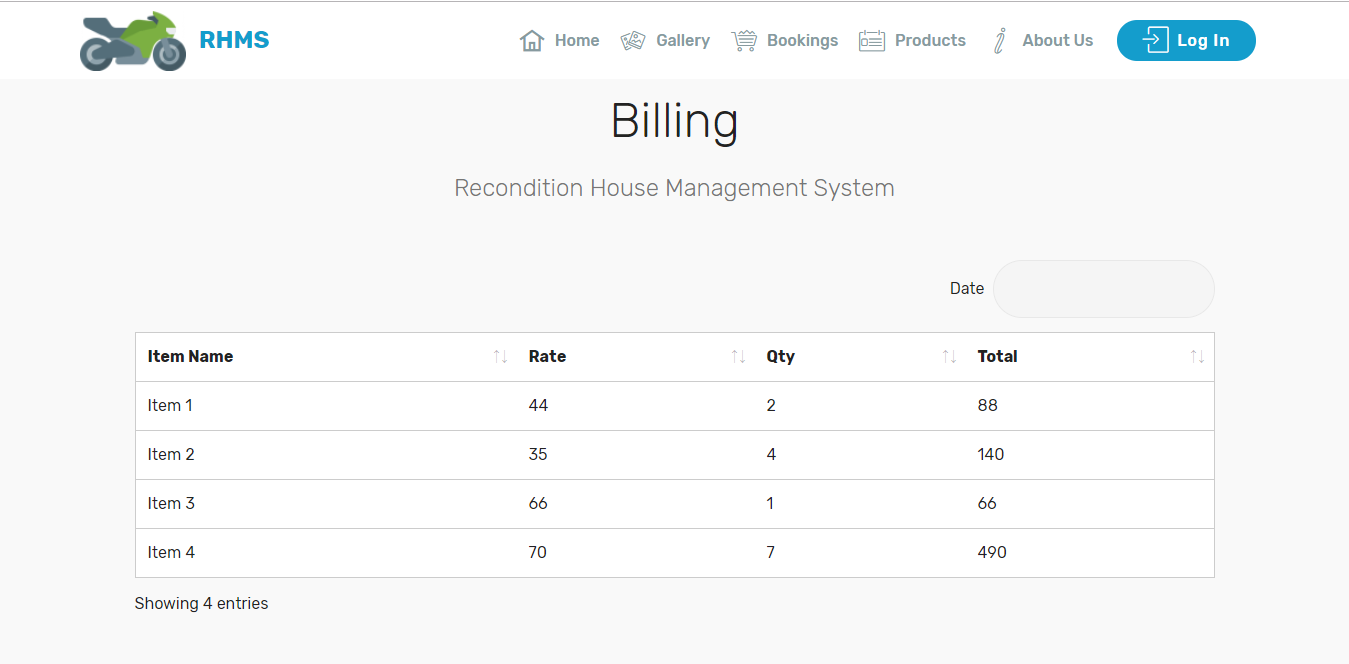
****

Figure 21: Billing page

1. **About Us:**

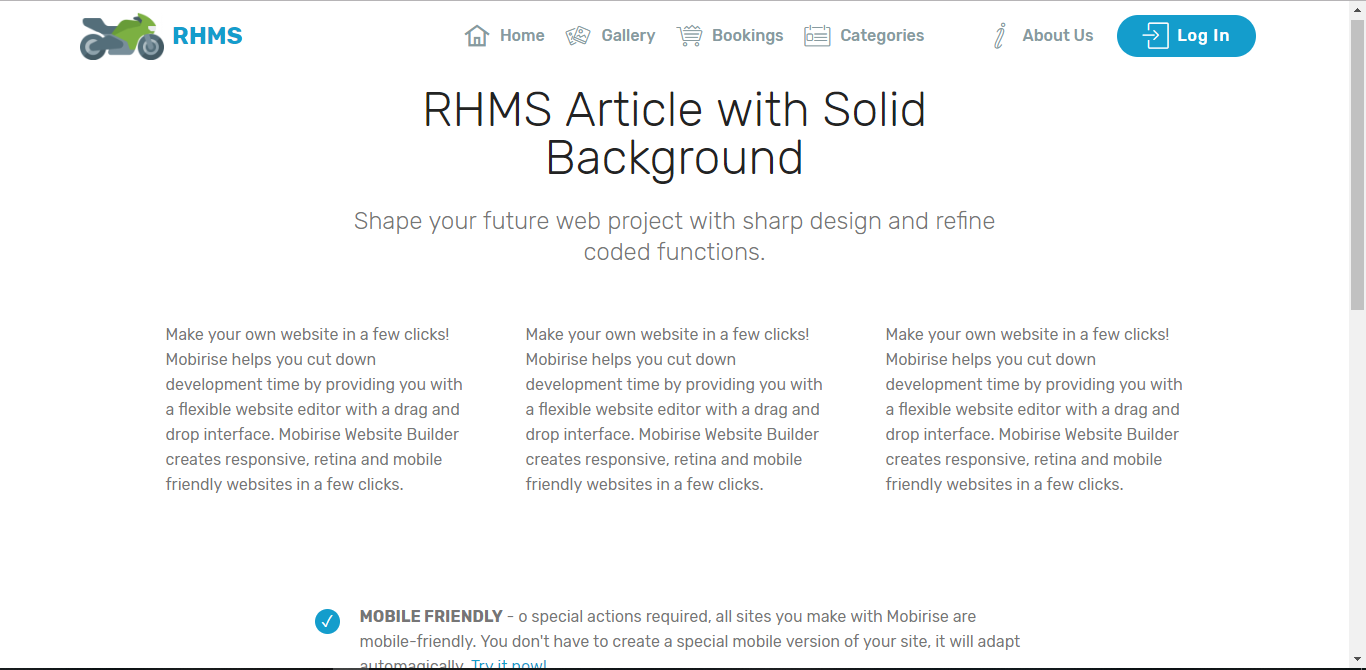
****

Figure 22: About Us Page User Interface

1. **Log In:**

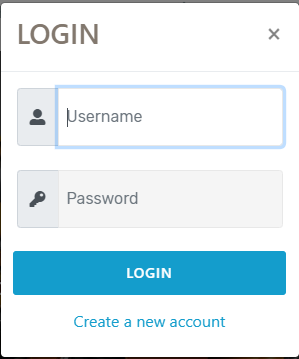
****

Figure 23: Login User Interface

1. **Sign Up:**

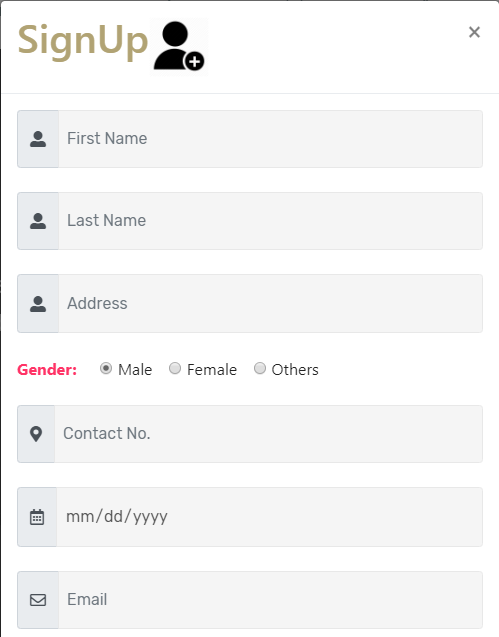
****

Figure 24: Sign Up User Interface

# Conclusion:

In this part, I have completed the analysis phase of developing software. In the completion of this phase, we have outcome the requirements, database model and user interface.

# References

Anon., n.d. [Online]   
Available at: https://www.tutorialspoint.com/uml/uml\_class\_diagram.htm

Anon., n.d. [Online]   
Available at: https://www.tutorialspoint.com/uml/uml\_use\_case\_diagram.htm

Anon., n.d. [Online]   
Available at: https://www.tutorialspoint.com/uml/uml\_activity\_diagram.htm

Anon., n.d. *stack overflow.* [Online]   
Available at: https://stackoverflow.com/questions/16475979/what-is-the-difference-between-functional-and-non-functional-requirement