**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: 4](#_Toc2981867)

[Functional and Non-Functional: 4](#_Toc2981868)

[Requirement Analysis: 5](#_Toc2981869)

[USE CASE DIAGRAM 7](#_Toc2981870)

[Class Diagram: 9](#_Toc2981871)

[Initial Class Diagram: 10](#_Toc2981872)

[Final Class Diagram: 11](#_Toc2981873)

[ER Diagram: 12](#_Toc2981874)

[13](#_Toc2981875)

[Activity Diagram: 14](#_Toc2981876)

[Sequence Diagram: 18](#_Toc2981877)

[Prototype: 21](#_Toc2981878)

[UI Design: 25](#_Toc2981879)

[Conclusion: 30](#_Toc2981880)

[References 30](#_Toc2981881)

**List of Figures:**

[Figure 1: Use Case Diagram 8](#_Toc2962295)

[Figure 2: Initial Class Diagram 10](#_Toc2962296)

[Figure 3: Final class diagram 11](file:///D:\ReconditionHouseManagementSystem\Final_Report\Final%20Report.docx#_Toc2962297)

[Figure 4: ER Diagram 13](file:///D:\ReconditionHouseManagementSystem\Final_Report\Final%20Report.docx#_Toc2962298)

[Figure 5: Login Activity Diagram 15](file:///D:\ReconditionHouseManagementSystem\Final_Report\Final%20Report.docx#_Toc2962299)

[Figure 6: Item Activity Diagram 16](file:///D:\ReconditionHouseManagementSystem\Final_Report\Final%20Report.docx#_Toc2962300)

[Figure 7: Bill Activity Diagram 17](#_Toc2962301)

[Figure 8: Login Sequence Diagram 19](file:///D:\ReconditionHouseManagementSystem\Final_Report\Final%20Report.docx#_Toc2962302)

[Figure 9: Bill Sequence Diagram 19](file:///D:\ReconditionHouseManagementSystem\Final_Report\Final%20Report.docx#_Toc2962303)

[Figure 10: Item Sequence Diagram 20](file:///D:\ReconditionHouseManagementSystem\Final_Report\Final%20Report.docx#_Toc2962304)

[Figure 11: Login Page Prototype 21](#_Toc2962305)

[Figure 12: Sign Up Page Prototype 21](#_Toc2962306)

[Figure 13: Home Page Prototype 22](#_Toc2962307)

[Figure 14: Gallery Page Prototype 22](#_Toc2962308)

[Figure 15: Bookings Page Prototype 23](#_Toc2962309)

[Figure 16: Categories Page Prototype 23](#_Toc2962310)

[Figure 17: About Us Page Prototype 24](#_Toc2962311)

[Figure 18: Admin Page Prototype 24](#_Toc2962312)

[Figure 19: Home Page User Interface 25](#_Toc2962313)

[Figure 20: Gallery Page User Interface 25](#_Toc2962314)

[Figure 21: Bookings Page User Interface 26](#_Toc2962315)

[Figure 22: Categories Page User Interface 26](#_Toc2962316)

[Figure 23: Sign Up Page 27](#_Toc2962317)

[Figure 24: Billing page 27](#_Toc2962318)

[Figure 25: About Us Page User Interface 28](#_Toc2962319)

[Figure 26: Login User Interface 28](#_Toc2962320)

[Figure 27: Sign Up User Interface 29](#_Toc2962321)

# Introduction:

The documentation consists of full information about the project named “Recondition House Management System”. The goal of the project is to develop an applicable website. The project is developed using ‘Waterfall Model’. Waterfall model is one of the popular approach used in software development and manufactures or industries as well. It consists of six stages or phases they are; ***Requirement Analysis***, ***Design***, ***Implementation***, ***Testing, Deployment and Maintenance.***

In the ***Requirement Analysis*** phase of the project 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 models such as use case diagrams, class diagrams, E-R diagrams, activity diagrams, sequence diagrams and Prototype. In this phase, the tools I have used are Star UML and Balsamiq.

After the completion of ***Requirement Analysis*** phase I have started designing the user interface. In this ***Design*** phase, I have designed the interfaces with HTML5, Bootstrap4 and Javascript.

In the ***Implementation*** phase, I have started the code using PHP language on Laravel framework. The tools that I have used in this phase are XAMPP and Sublime (editor).

# 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 also they are prioritized with MoSCoW prioritization.

Indexes 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** | **Description** |
|  | Login | **M** | Login is necessary to verify and validate the users (All the users). |
|  | Registration | **M** | User (Not for admin) should be registered in order to get access. |
|  | Insert Item | **M** | Items are to be inserted by admin only to provide the available items for users. |
|  | Update Item | **S** | Items information should be able to be updated by the admin only. |
|  | Delete Item | **S** | Admin should be able to delete the unavailable items. |
|  | View Item Information | **M** | Both admin and users should be able to view item information. |
|  | Book Item | **M** | Users (Not for admin) should be able to book items they want. |
|  | Search Item | **S** | Items should be able to view information in search if the item exists. |
|  | Home Page | **S** | Home page should be available after the user authentication is verified. |
|  | Edit Profile | **S** | Users (All the users) should be able to edit their profile or information. |
|  | User Review | **C** | User review is useful to know the product’s quality. |
|  | Log Out | **M** | Users should be able to logout their account for their security. |
|  | Generate Bill | **M** | Users are provided bills after they buy the products. Therefore, bill should be able to be generated. |
|  | Forgot Password | **C** | It is one of the good factor that increases the customer satisfaction. User sometimes forgot their passwords, in that case it is very useful. |
|  | Contact Information | **S** | Contact information are provided to the users in order to provide communication service. |
|  | Visitor Counter | **C** | It is just used for analysis purpose that counts how many times the site is visited. |
|  | Manage Inventory | **M** | Inventory system of the data should be managed in a proper way that reduces the future risks. |

**Functional Requirements Table with MoSCoW prioritization:**

Table 1: Requirement Analysis Table

**Non-Functional Requirements Table with MoSCoW prioritization:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Requirements** | **MoSCoW** | **Description** |
|  | Security | **S** | Users should be provided security for their accounts, activities they perform, and also the data in the system. |
|  | Performance | **S** | User should not feel that the system (server) is slow. There are many options to increase the performance. |
|  | Availability | **S** | The system should be available to the users whenever they want. |
|  | Reliability | **S** | The system should be reliable to the users no matters what or how many time they performs any activities. |
|  | 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.

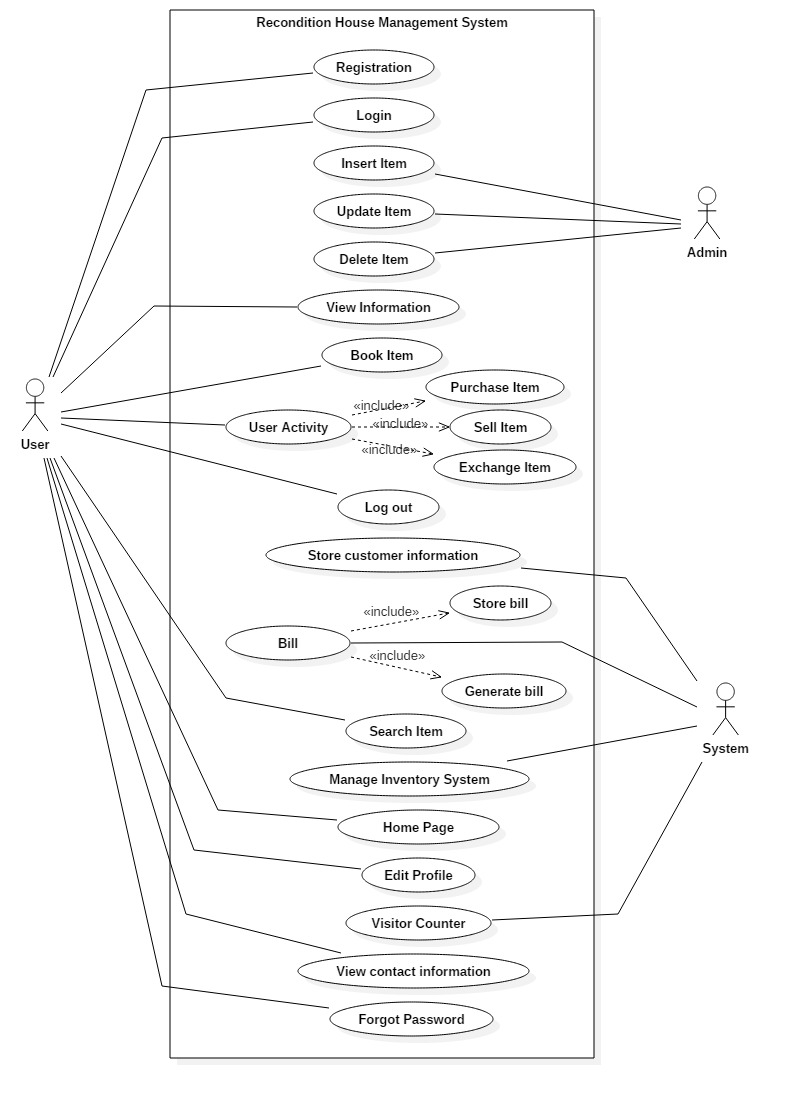
****

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:

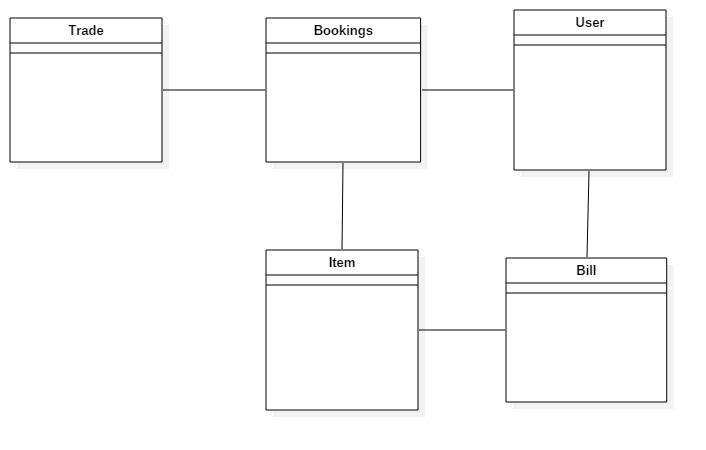


Figure 2: Initial Class Diagram

# 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 : 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.

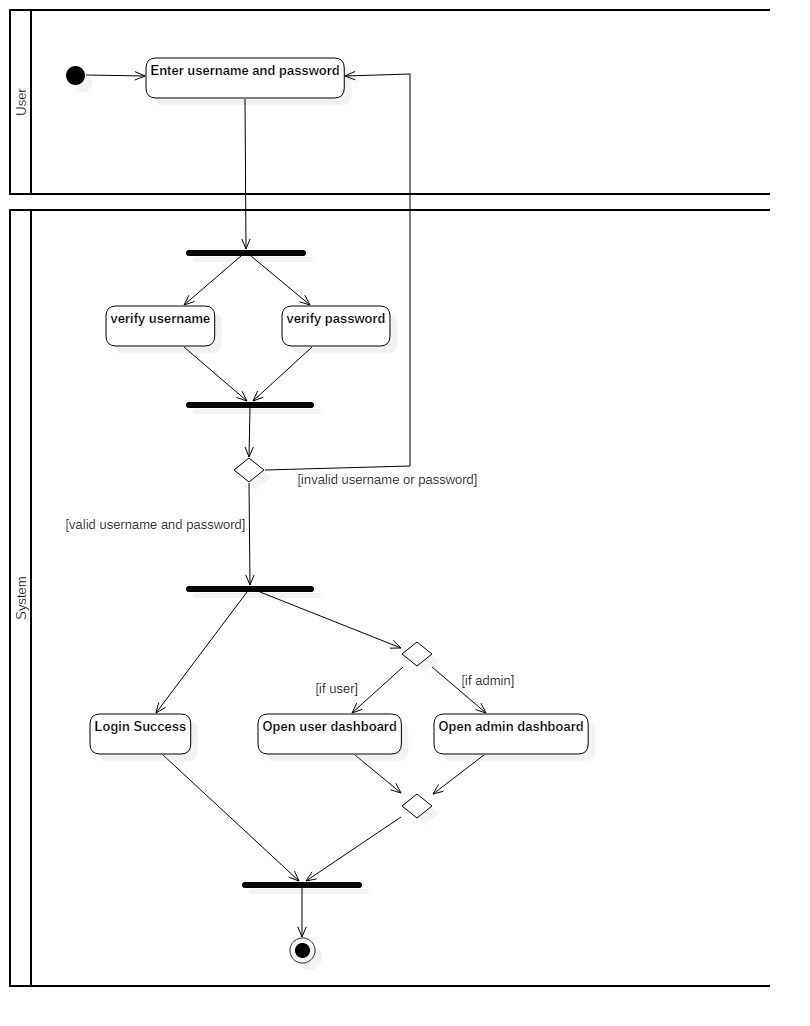
****

Figure 5: Login Activity Diagram

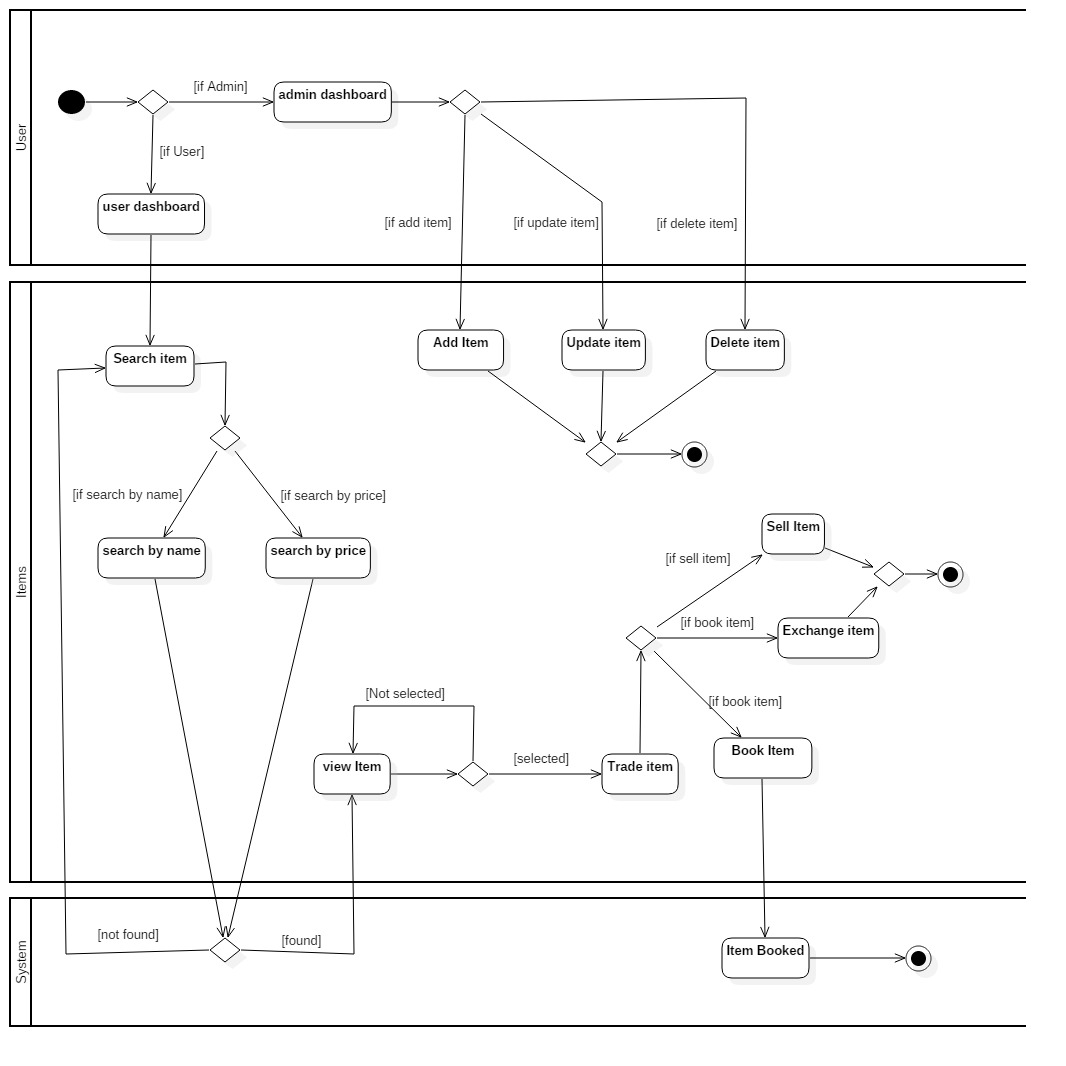


Figure 6: Item Activity Diagram

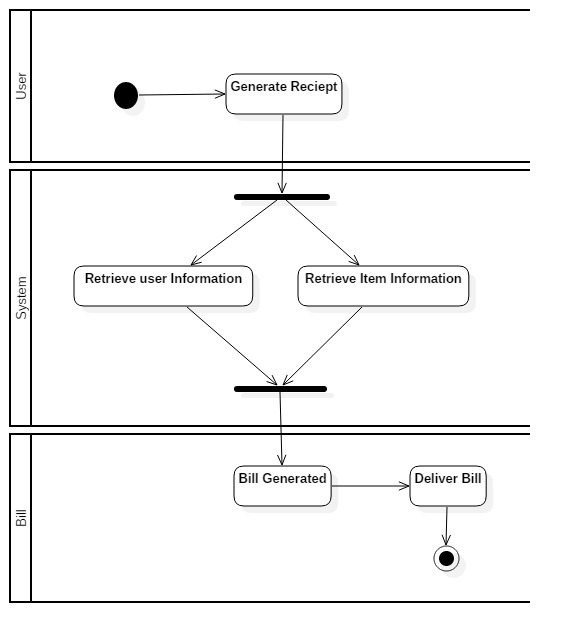


Figure 7: Bill Activity Diagram

# Sequence Diagram:

Sequence Diagrams are interaction diagrams that detail how operations are carried out. Sequence Diagrams show elements as they interact over time and they are organized according to object (horizontally) and time (vertically). The sequence diagram is used primarily to show the interactions between objects in the sequential order that those interactions occur.

**Justifications:**

Here, I have drawn sequence diagram to show the high-level interaction between active objects in a system, model the interaction between object instances within a collaboration that realizes a use case, the interaction between objects within a collaboration that realizes an operation and to show all the possible paths through the interaction.

**Advantages:**

* The sequence diagram may be used in object-oriented analysis to validate class diagrams against use cases, or to show the timing of interactions between entities within the system scope.

**Disadvantages:**

* A sequence diagram must be defined for each possible scenario. Strictly speaking, a sequence diagram requires a fully defined class model.

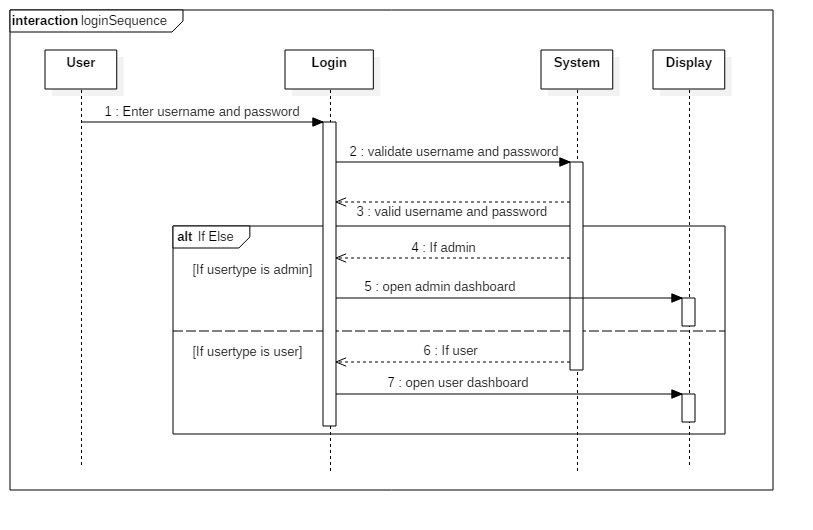


Figure 8: Login Sequence Diagram

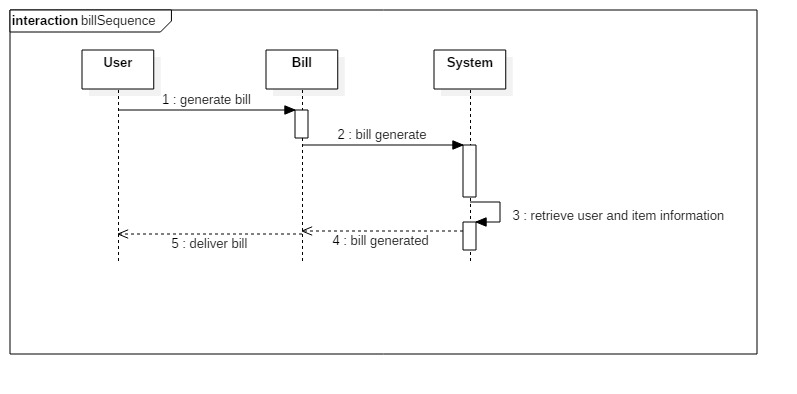


Figure 9: Bill Sequence Diagram

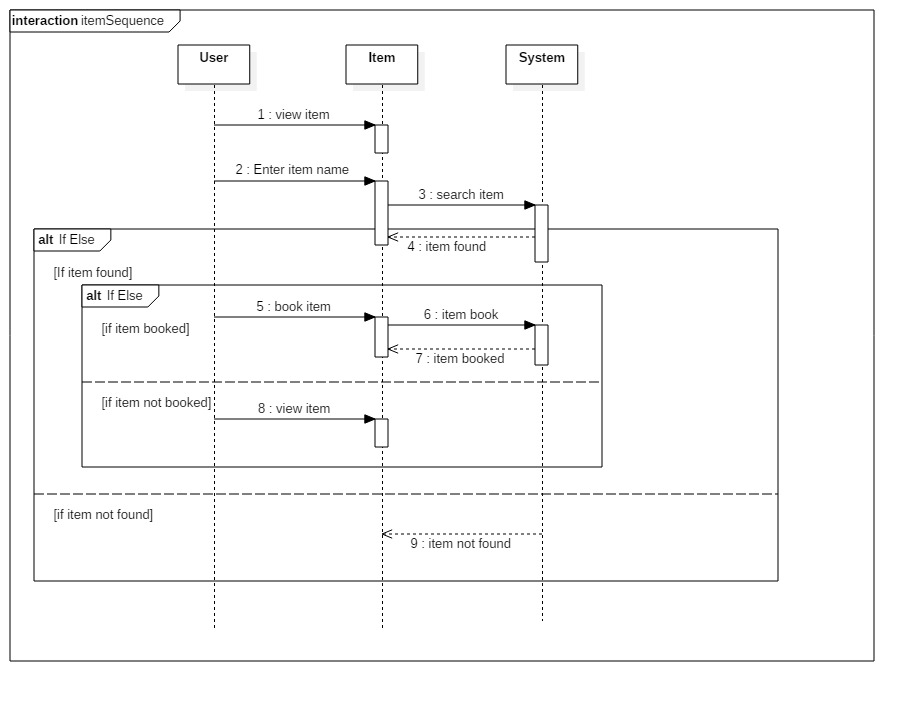


Figure 10: Item Sequence Diagram

# Prototype:

1. **Login Page:**

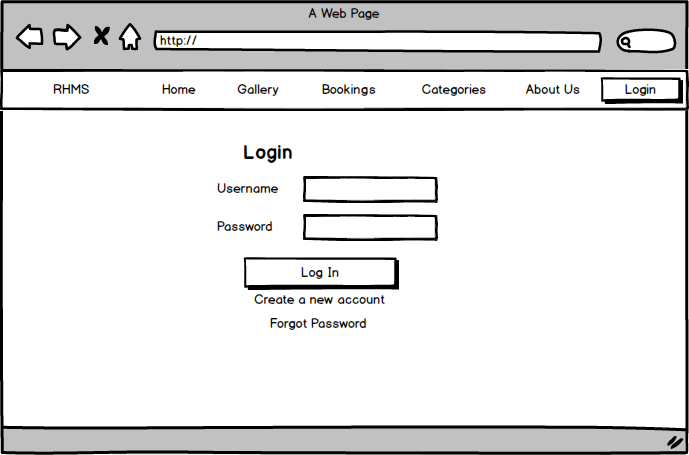
****

Figure : Login Page Prototype

1. **Sign Up Page:**

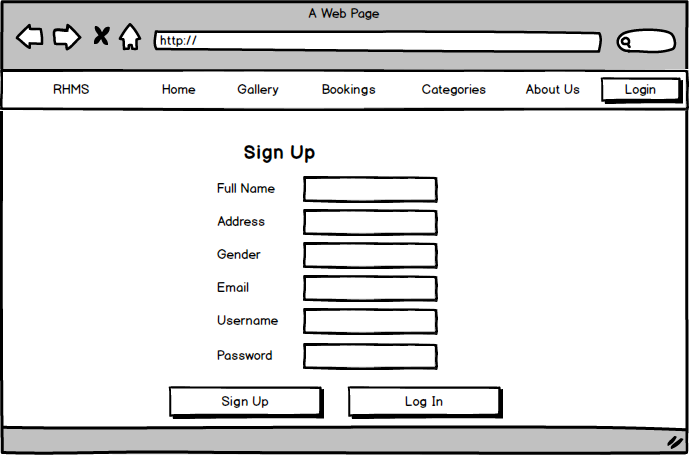
****

Figure : Sign Up Page Prototype

1. **Home Page:**

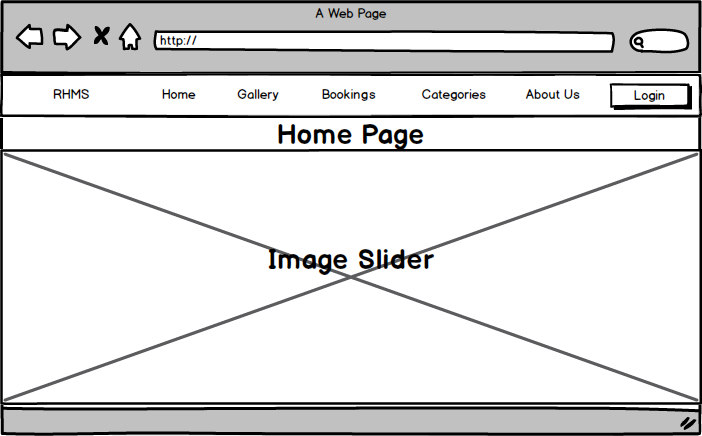
****

Figure : Home Page Prototype

1. **Gallery Page:**

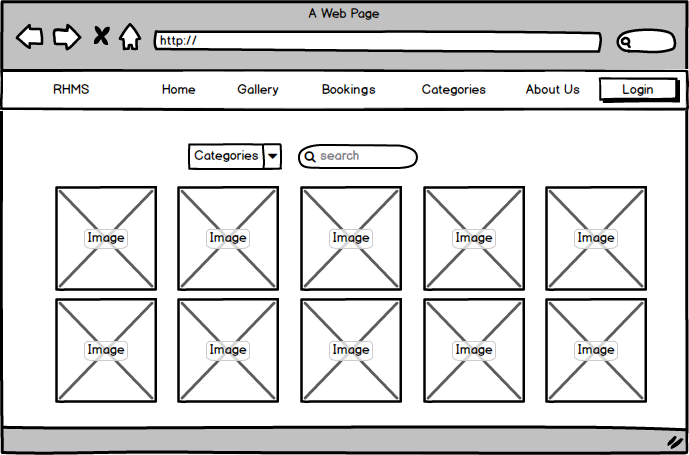
****

Figure : Gallery Page Prototype

1. **Bookings Page;**

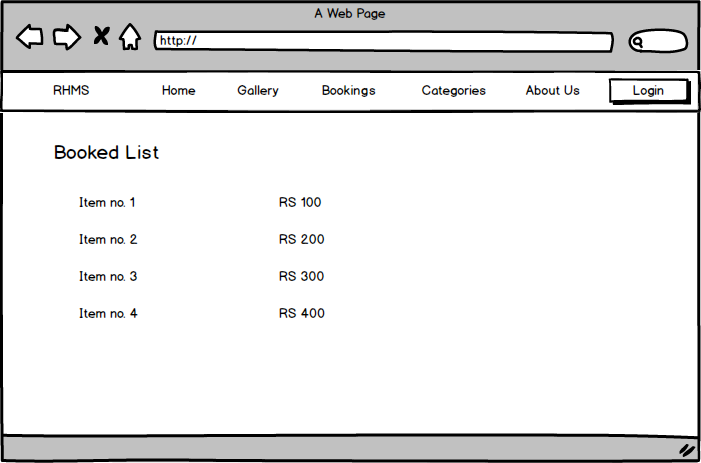
****

Figure : Bookings Page Prototype

1. **Categories Page:**

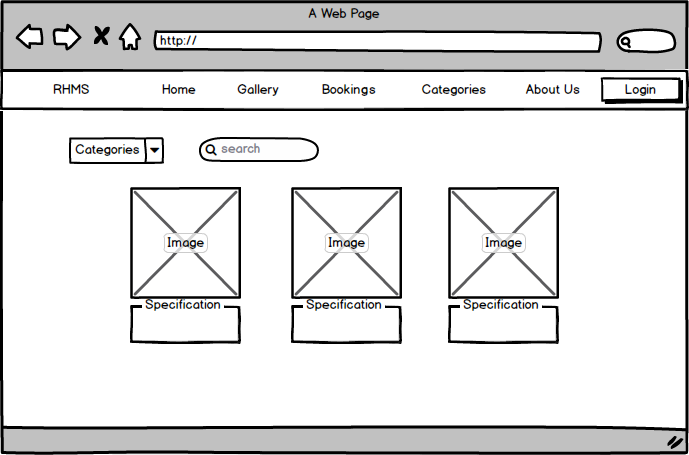
****

Figure : Categories Page Prototype

1. **About Us Page:**

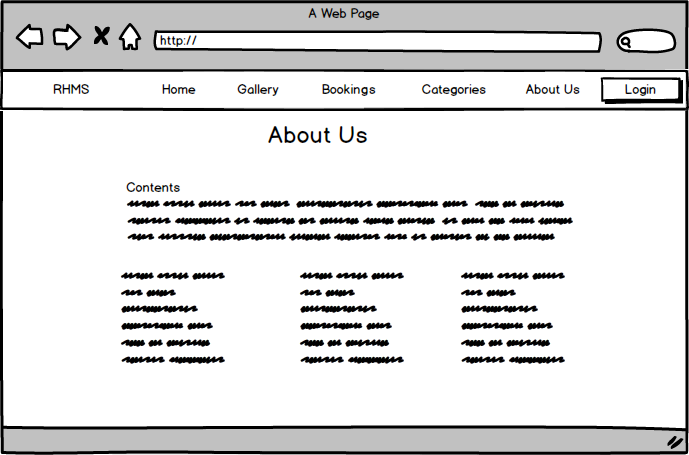
****

Figure : About Us Page Prototype

1. **Admin Page:**

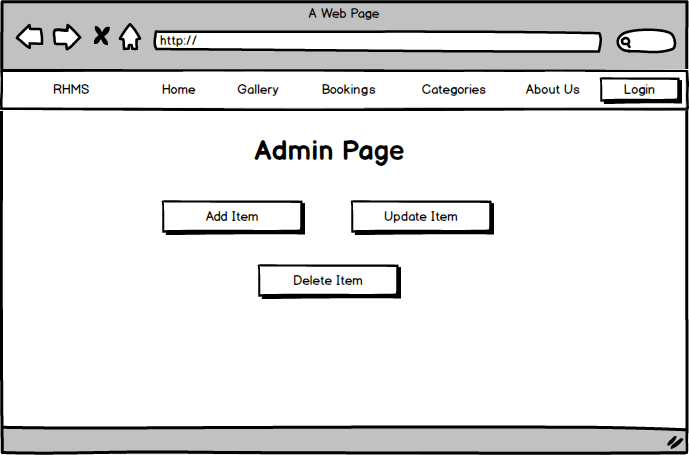
****

Figure : Admin Page Prototype

# UI Design:

1. **Home Page:**

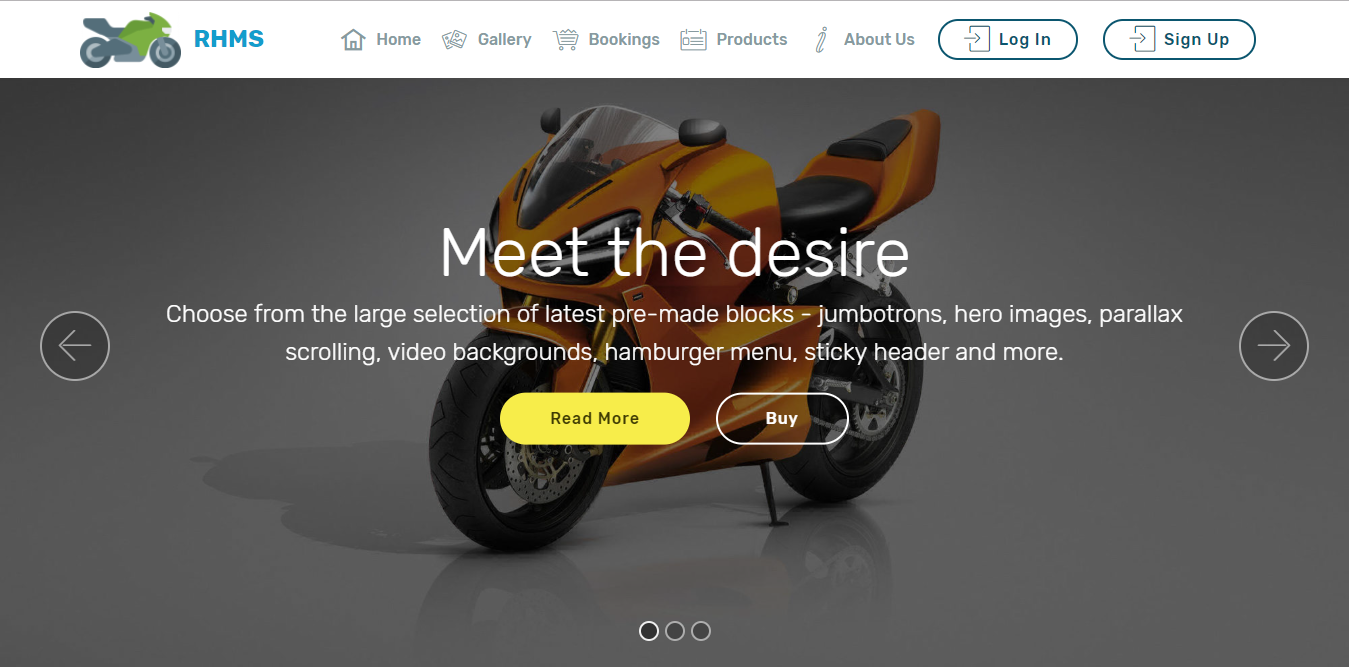
****

Figure : Home Page User Interface

1. **Gallery Page:**

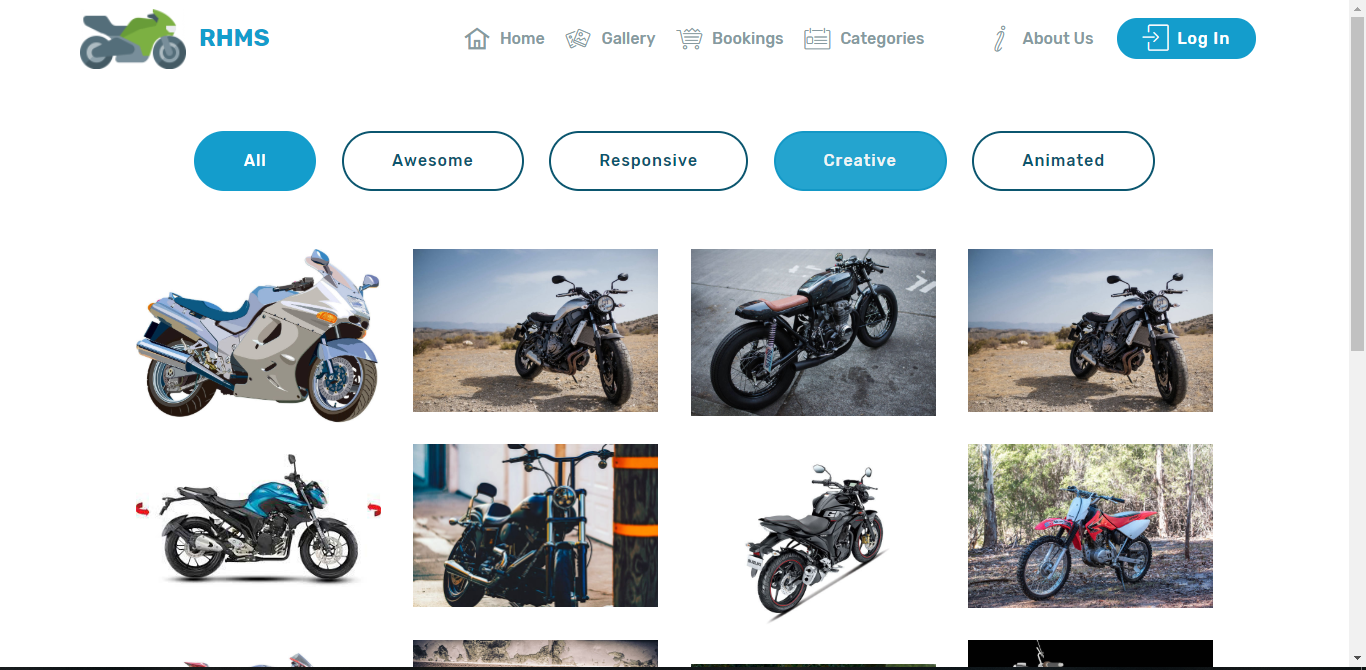
****

Figure : Gallery Page User Interface

1. **Bookings Page:**

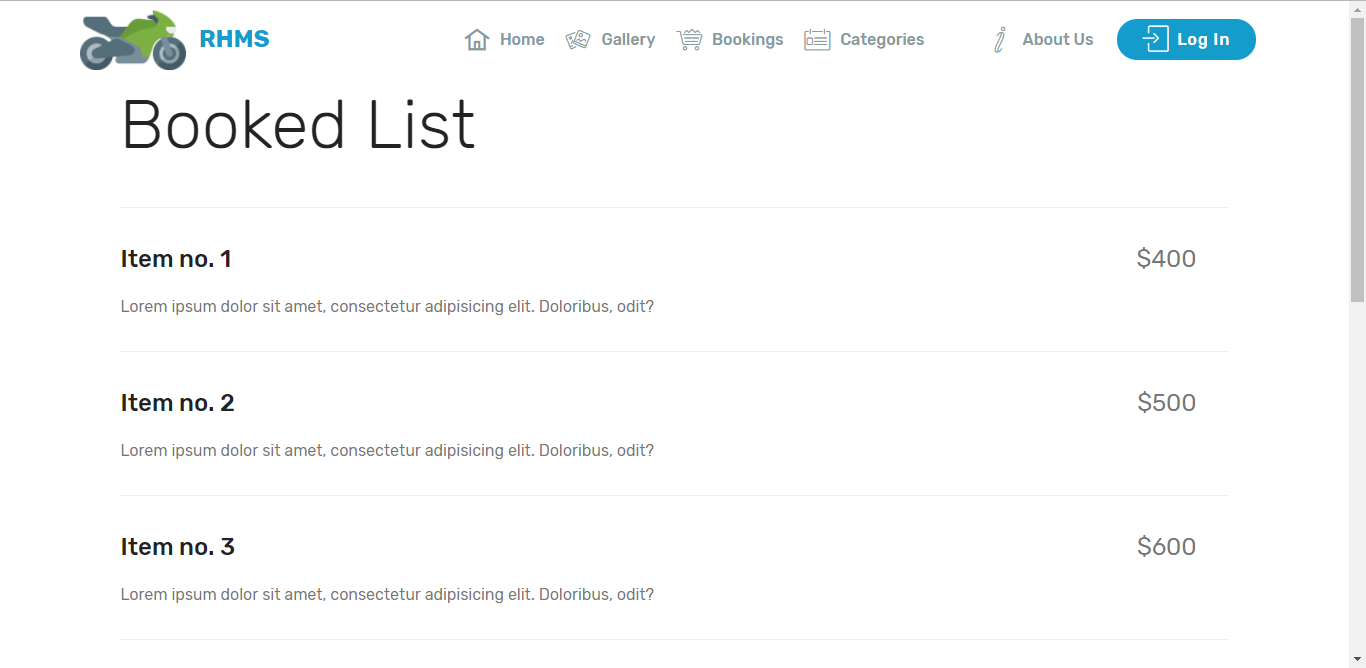
****

Figure : Bookings Page User Interface

1. **Categories Page:**

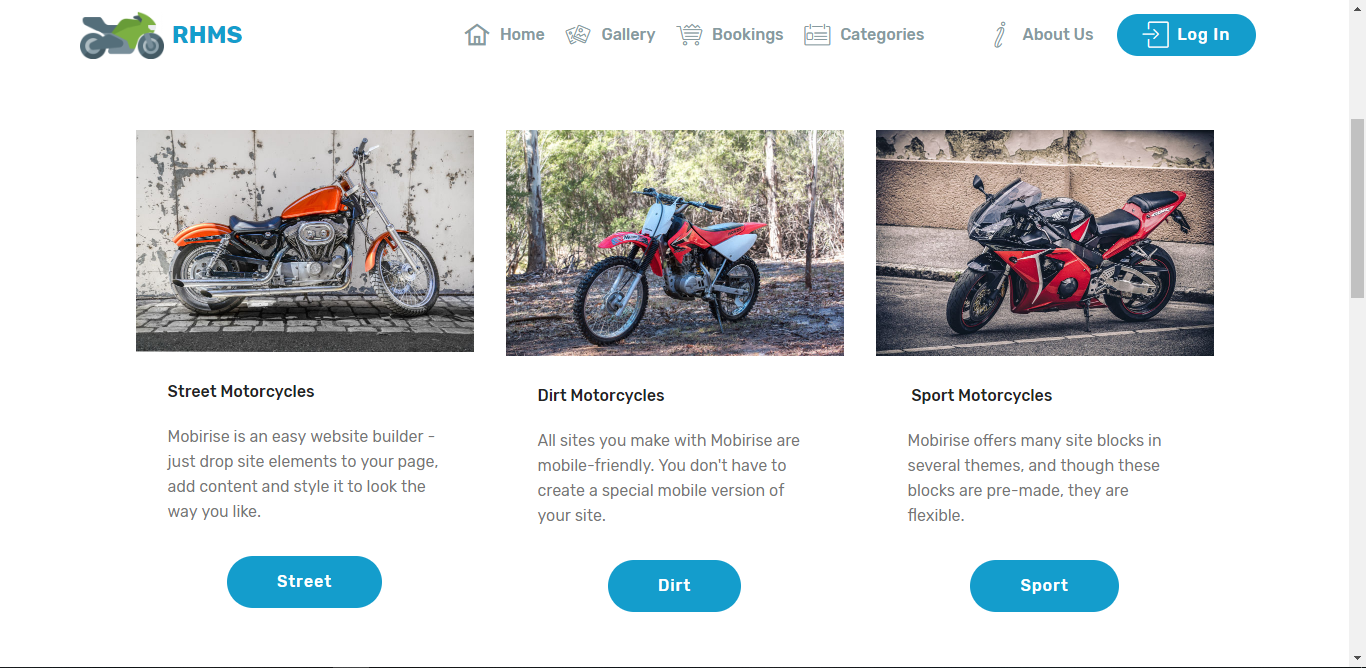
****

Figure : Categories Page User Interface

1. **Edit Profile page:**

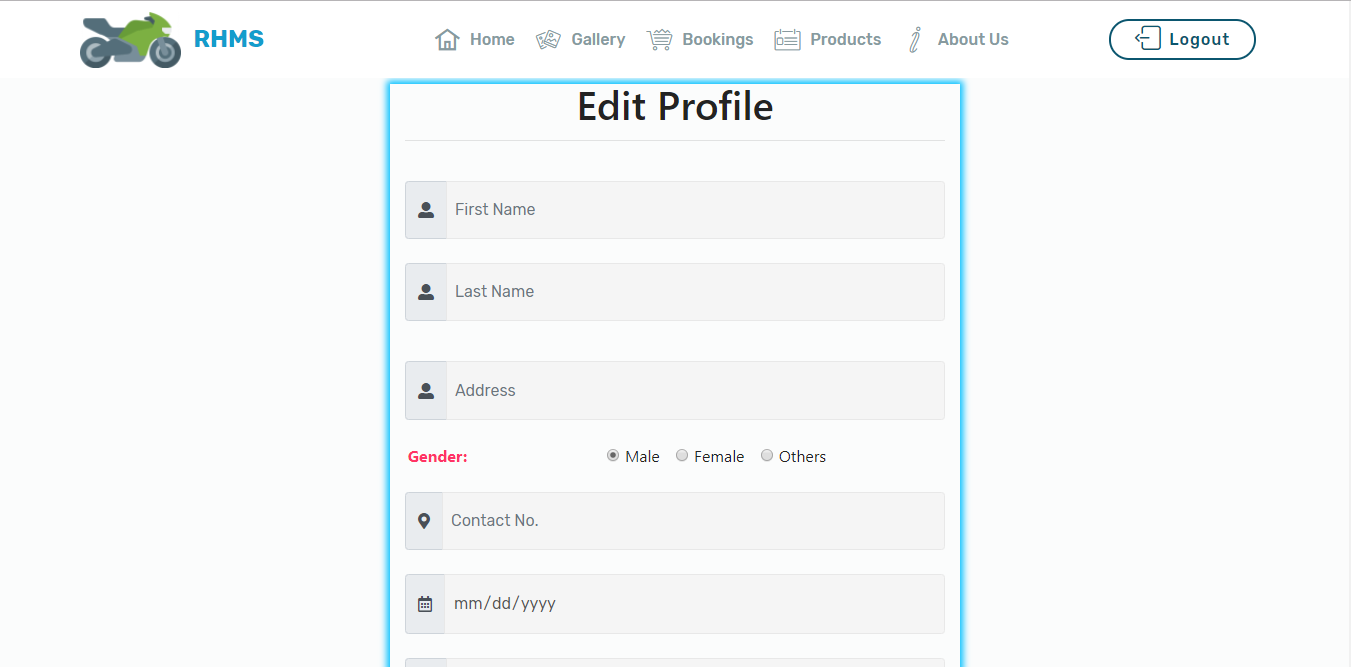
****

Figure : Sign Up Page

1. **Billing:**

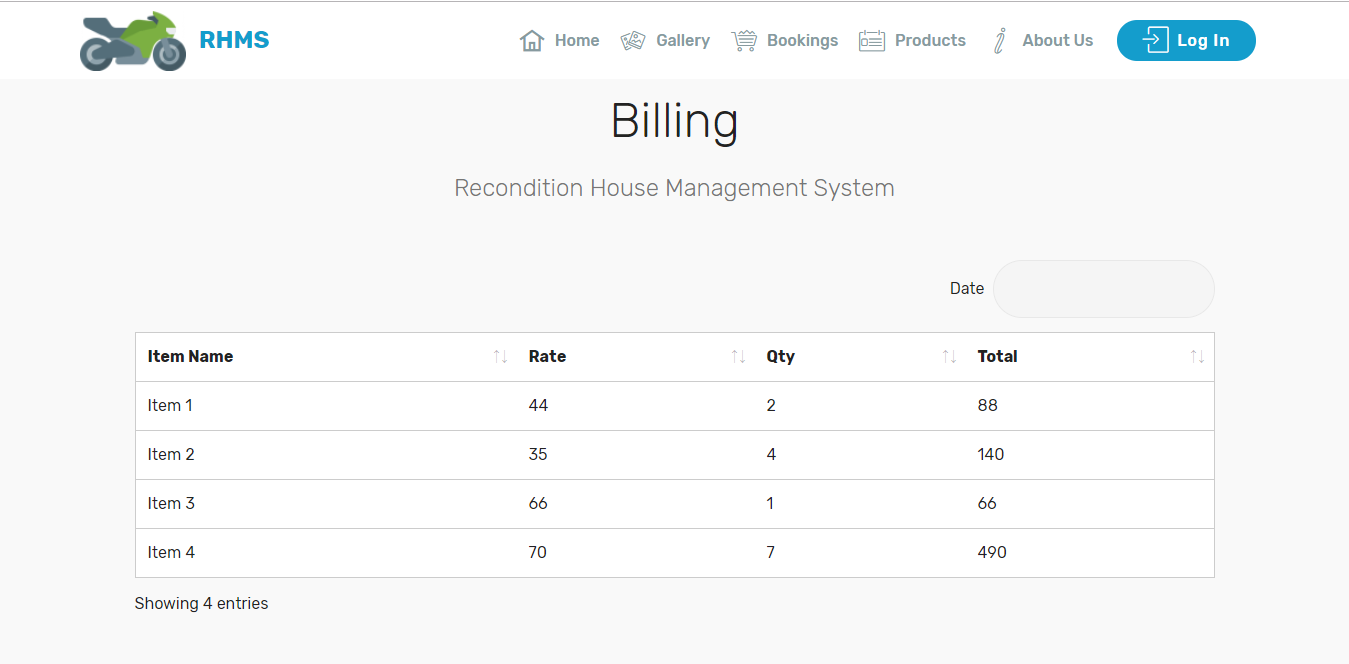
****

Figure : Billing page

1. **About Us:**

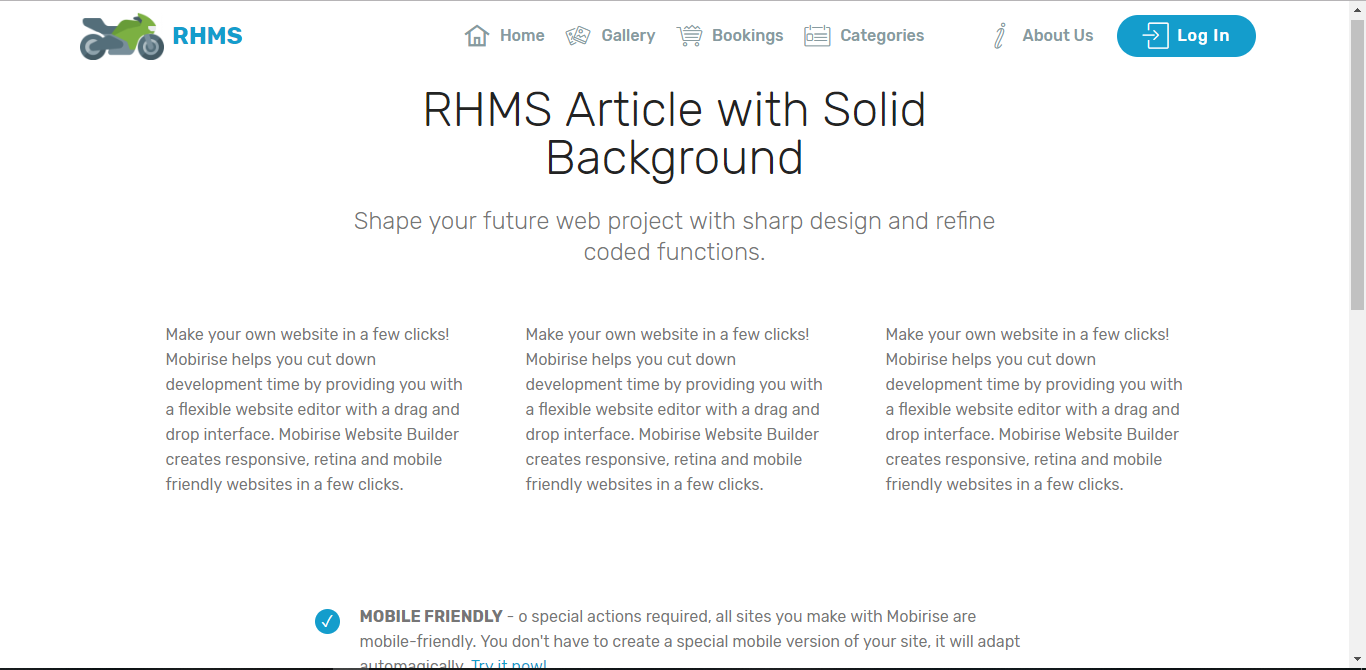
****

Figure : About Us Page User Interface

1. **Log In:**

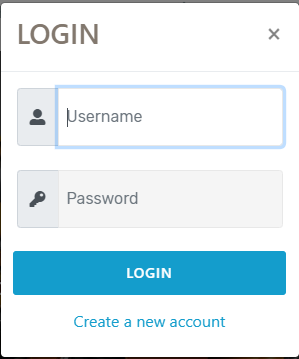
****

Figure : Login User Interface

1. **Sign Up:**

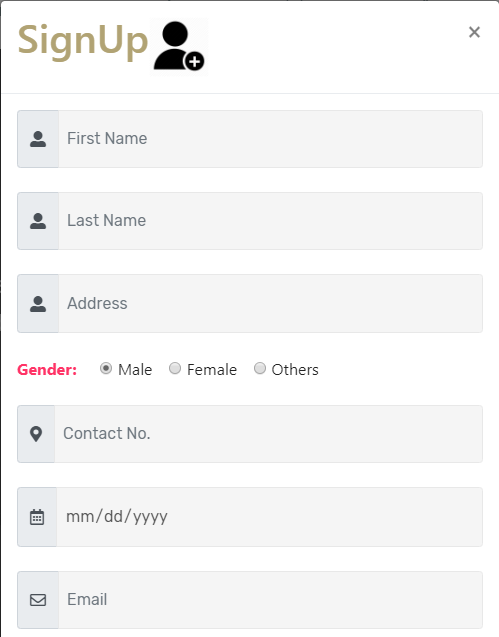
****

Figure : 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