# Design

## Introduction

Design is the phase of third phase of Software Development Life Cycle. This phase describes in detail, the necessary specification, features and operation that will satisfy the functional requirements of the proposed system which will be in place. It includes the design of application, network, databases, user interfaces, and system interfaces.

The lists of importance of using this phase in my project are:

* It includes the design of application, network, database, user interface and the system interfaces.
* It helps to transform SRS documents into logical structure.
* It helps to clarify the requirements of the system in efficient way.
* It helps to prevent from duplication.
* It helps to identify classes and object which will be use in a developing phase.

For my project, in this phase I have used ***Star UML*** software for activity, sequence, flow chart, class diagram, etc. ***Visual Paradigm*** is used for ER-diagram. ***Balsamiq Mockups 3*** is used for designing prototype design of the system.

## 3.1 – Structural Modeling

Structural design is a static model which represents the classes and objects of the system with their relationship. It shows the description of the entities and relationship between the entities. It doesn’t explain about the dynamic behavior of the system.

The below diagram is a structural modeling:

### 3.1.1 – Final Class Diagram

Class diagram is a UML diagram which represents the relationship and dependencies of the system classes, their attributes, and methods of the system objects. Class diagram is mostly used for OOP’s programming. It helps develop to code the system in a efficient way as the class diagram is represent. (Anon., 2019)

Class diagram consists of box with 3 parts: They are:

* First part: Name of class.
* Second part: Attributes of class.
* Third part: Operation to be applied.

Justification of using Class diagram on my project are listed below:

* It helps to analysis and design of the static view of my system.
* It will describe the responsibilities of a system.
* It helps to clarify the relationship between the entities, their attributes and operation of the system.

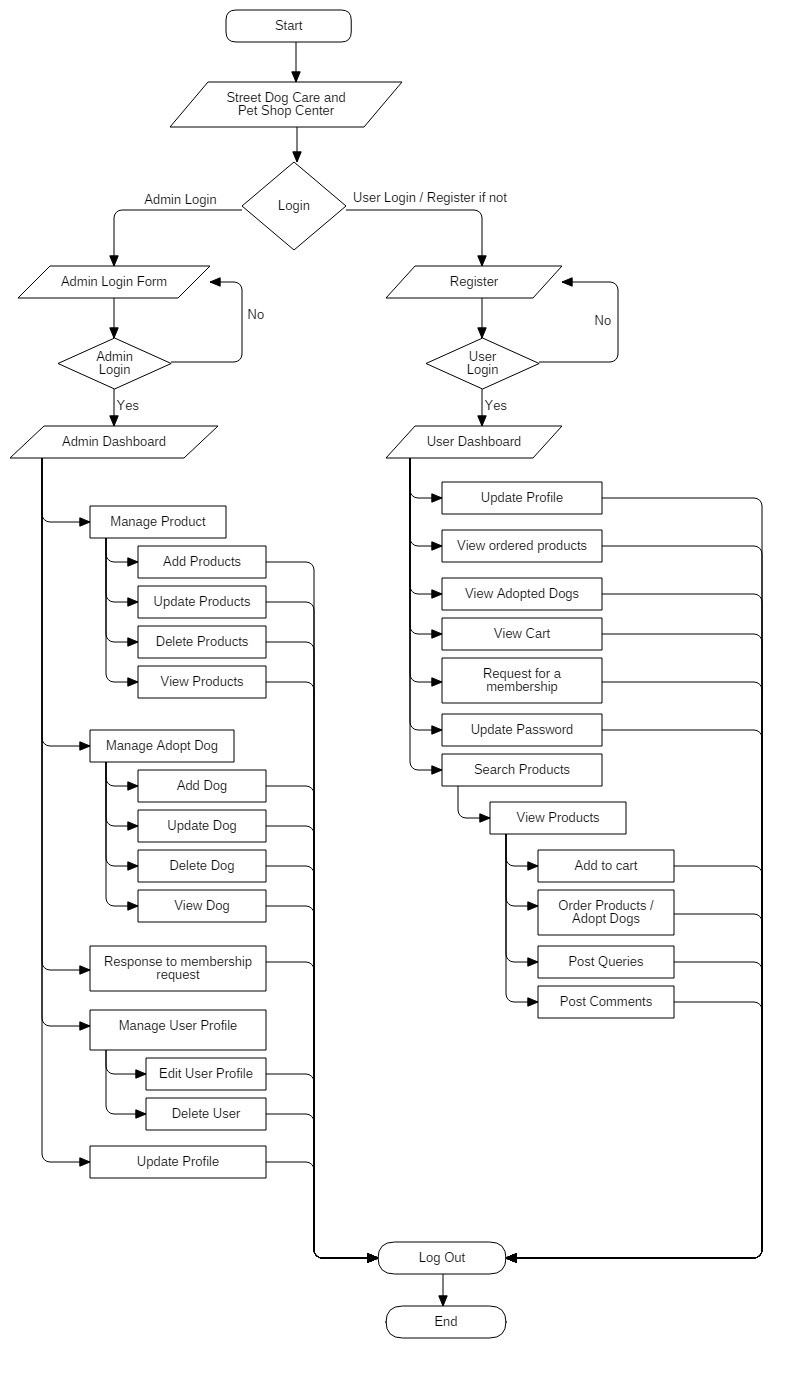
### 3.1.2 – Flow chart

Flow chart is a type of diagram which represent an algorithm, workplace or a process. It includes different boxes which is connected by the arrow symbols. It is very easy and helpful for coding and easy to understand and explain the flow of the system.

Justification of using Flow Chart in my project are listed below:

* It helps developer to code system in efficient way.
* It shows the process of the system.
* It will help in coding.
* Easy to understand the workflow.

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Notation Used** | **Notation Name** | **Description** |
| 1 | Flowline symbol in flowchart of programming | Flow Line | Helps to specify the flow of logic by joining the symbols. |
| 2 | Terminal symbol in flowchart of programming | Terminal | Helps to specify the start and end of flow. |
| 3. | Input/Output symbol in flowchart of programming | Input/output | Helps to indicate input and output of operation. |
| 4. | Processing symbol in flowchart of programming | Processing | Helps to perform arithmetic operation and data manipulation. |
| 5. | Decision making symbol in flowchart of programming | Decision | Helps to decide the operation is true or false. |



## 3.2 – Behavioral Modeling

Behavioral modeling is a representation of dynamic behavior of the system in a diagram. It explains the communication between the objects. It represents the Create, Retrieve, Update and Delete (CRUD) operation of the system.

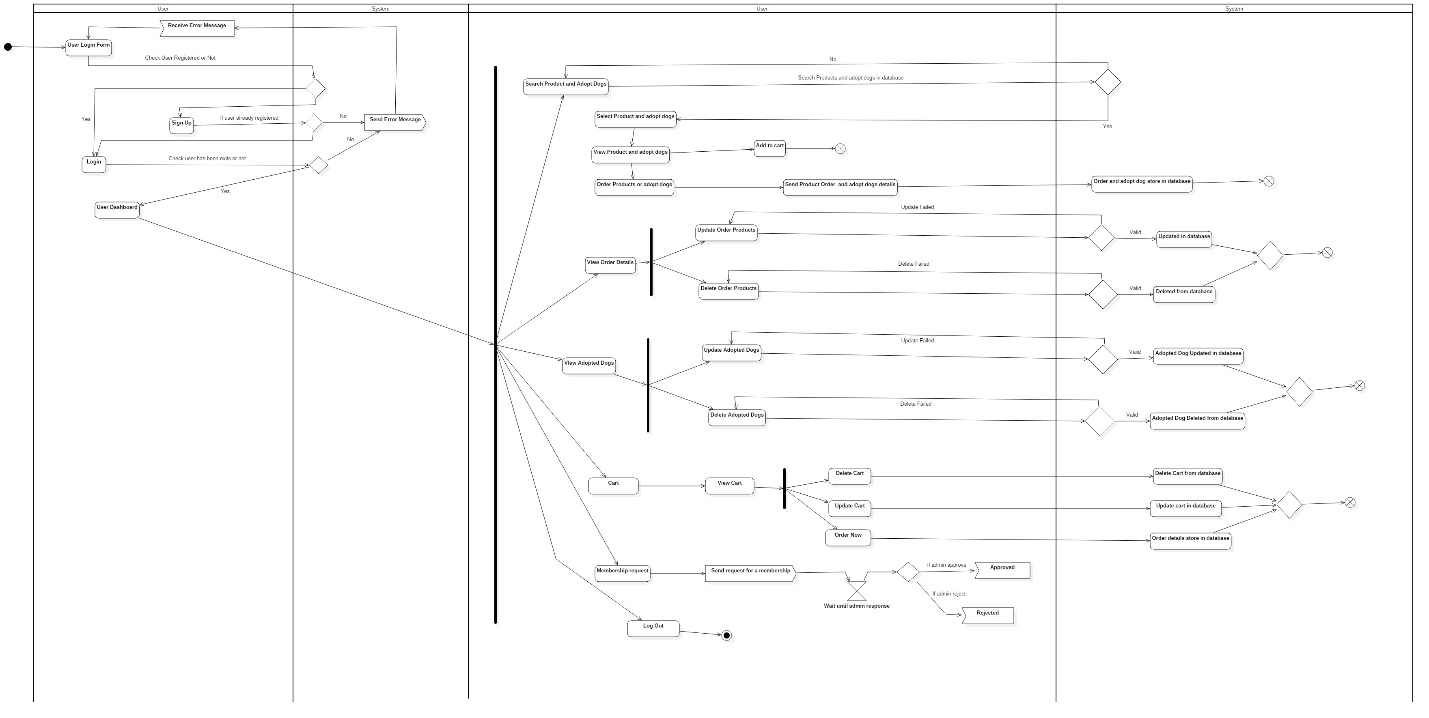
### 3.2.1 – Activity Diagram

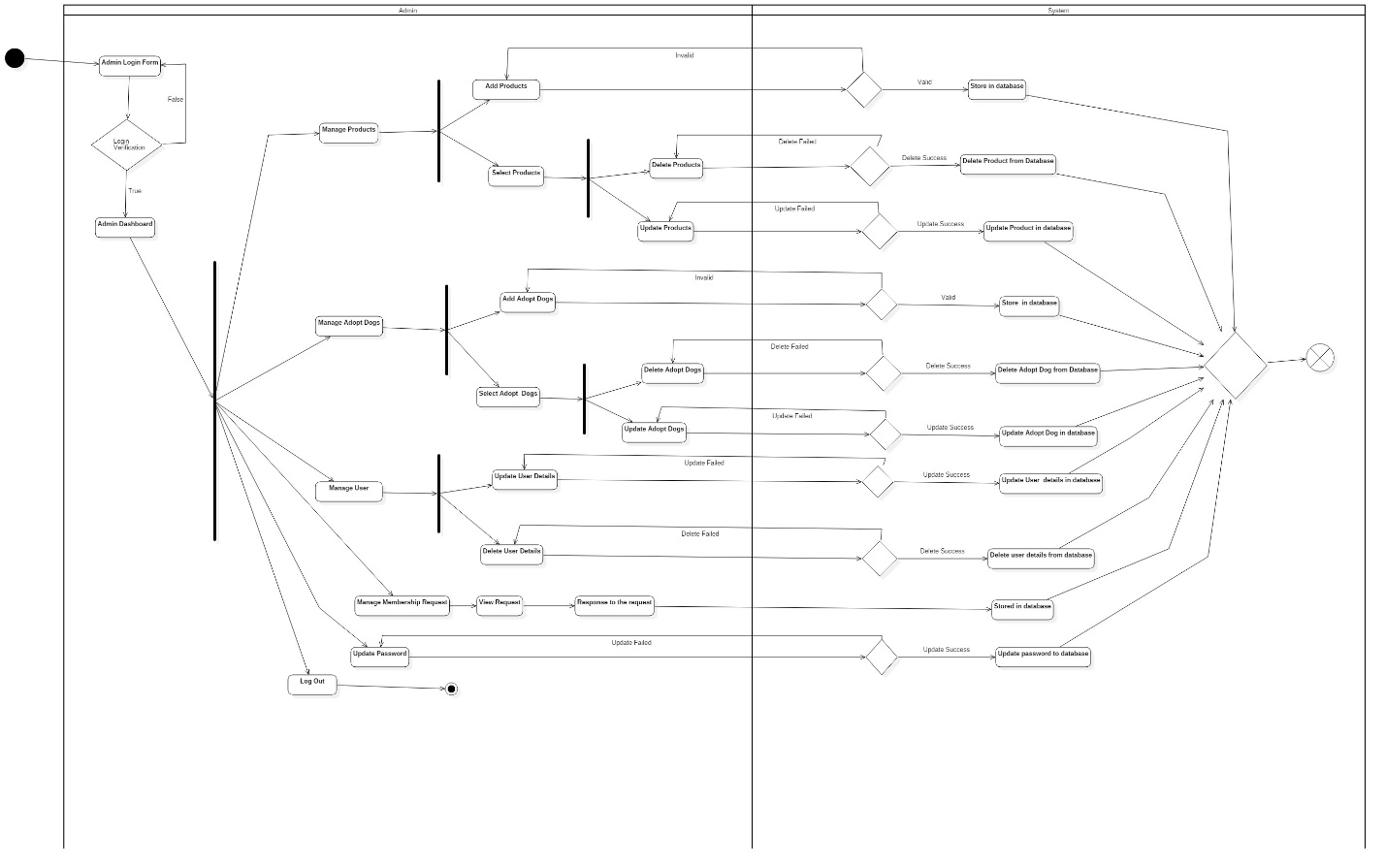
Activity is a type of flowchart that illustrates the flow from one activity to another activity. Activity is also known as the operation of the system. It shows the dynamic aspects of the system with the flow of operation from one to another.

Justification of using “Activity Diagram” on my project are listed below:

* It helps to clarify the flow of operation in multiple system and subsystems.
* It illustrates concurrent, parallel and branching flow of the system.
* It helps to explore and study business requirement as a larger scale.
* It provides an overview of the work flow of activities, that may help for the development phase.

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Notation Used** | **Notation Name** | **Description** |
| **1.** |  | Initial State | Helps to start the flow. |
| **2.** |  | Sequence Flow / Control Flow | Helps to show the transitions from one action state to another. |
| **3.** |  | Activity | Helps to represent a set of actions. |
| **4.** |  | State | Helps to define current condition of an event. |
| **5.** |  | Decision | Helps to test true or false condition. |
| **6.** |  | Synchronization | Split single behavior into a set of parallel flow of activities. |
| **7.** |  | Time Event | Helps to define the waiting or end period of the operation. |
| **8.** |  | Merge | Helps to bring together multiple flows that are concurrent. |
| **9.** |  | Message Sent | Helps to send some message to the system. |
| **10.** |  | Message Receive | Helps to receive message from the sent signal. |
| **11.** |  | Final State | Helps to indicate the end of flow |





### 3.2.2 – Sequence Diagram

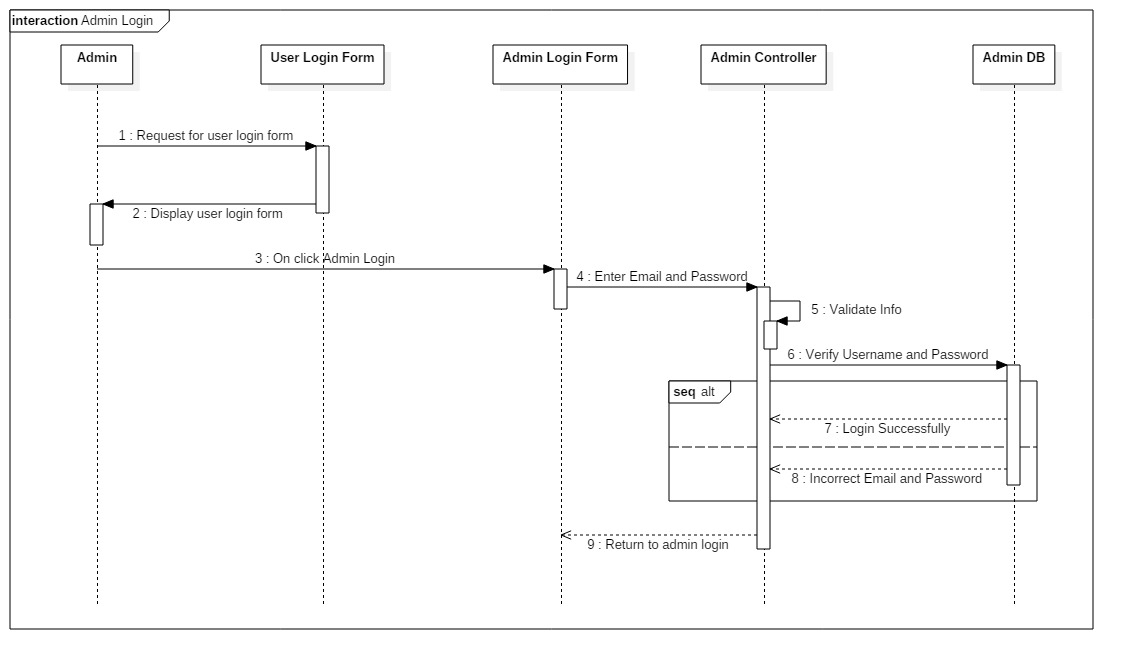
Sequence diagram is a type of ULM diagrams which represents the details interconnection with the object to exchange massage at a time. It shows the dynamic aspects of the flow to system with the message. The main focus of this diagram is on lifeline of an object which transfers the message from one objects to another.

Justification of using Sequence Diagram in my project are listed below:

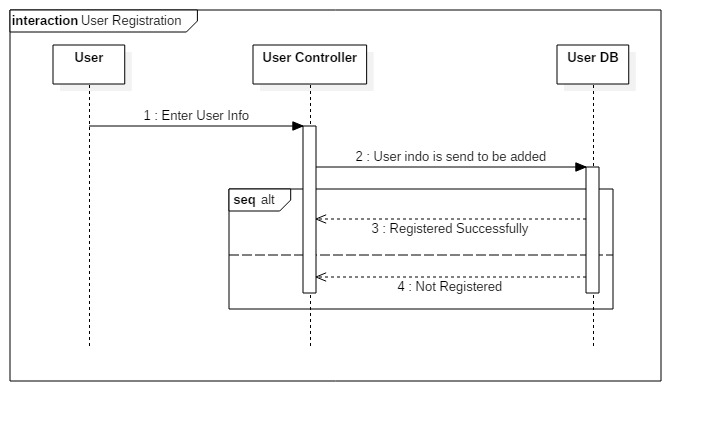
* It helps to define the whole process of a use case.
* It helps to illustrates the high level interaction between the objects in a system.
* It helps to plan and understand the details functionality of an existing system.

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Notation Used** | **Notation Name** | **Description** |
| **1.** |  | Object | Helps to indicate the object. |
| **2.** | Actor | Actor | Helps to indicate the actor like user, admin, etc. |
| **3.** |  | Lifeline | Helps to show the lifeline of the object. |
| **4.** |  | Activation | Helps to represent the period during which an element is performing an operation. |
| **5.** | Call Message | Message Arrow | Helps to define a particular communication between lifelines of an interaction. |
| **6.** | Return Message | Reply Message | Helps to define a particular communication between lifelines of an interactions. |

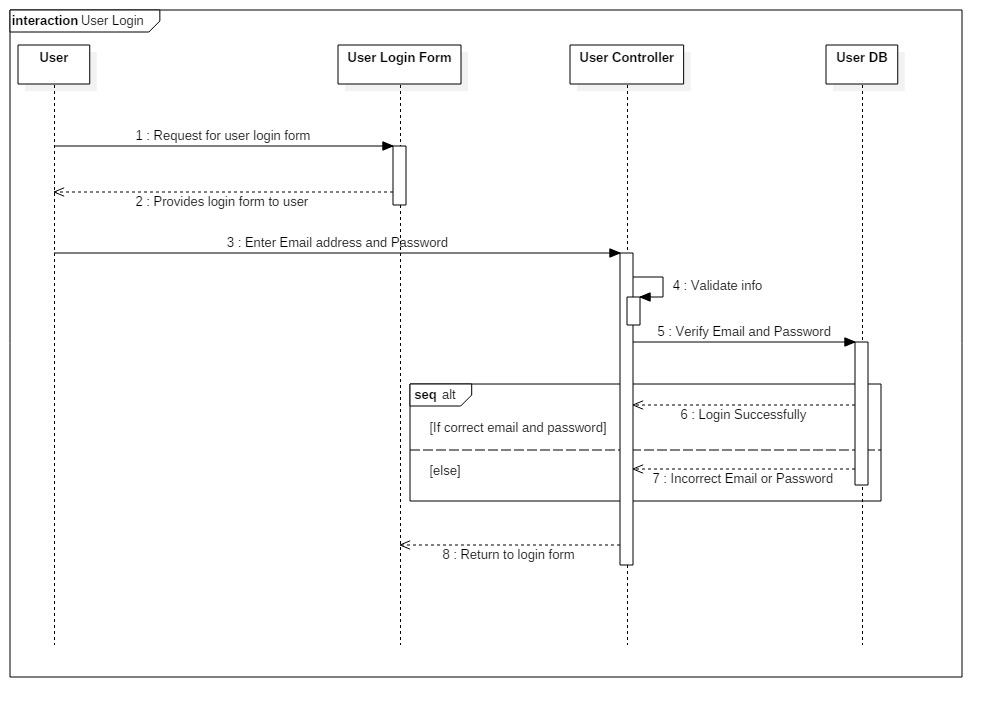
Admin Login



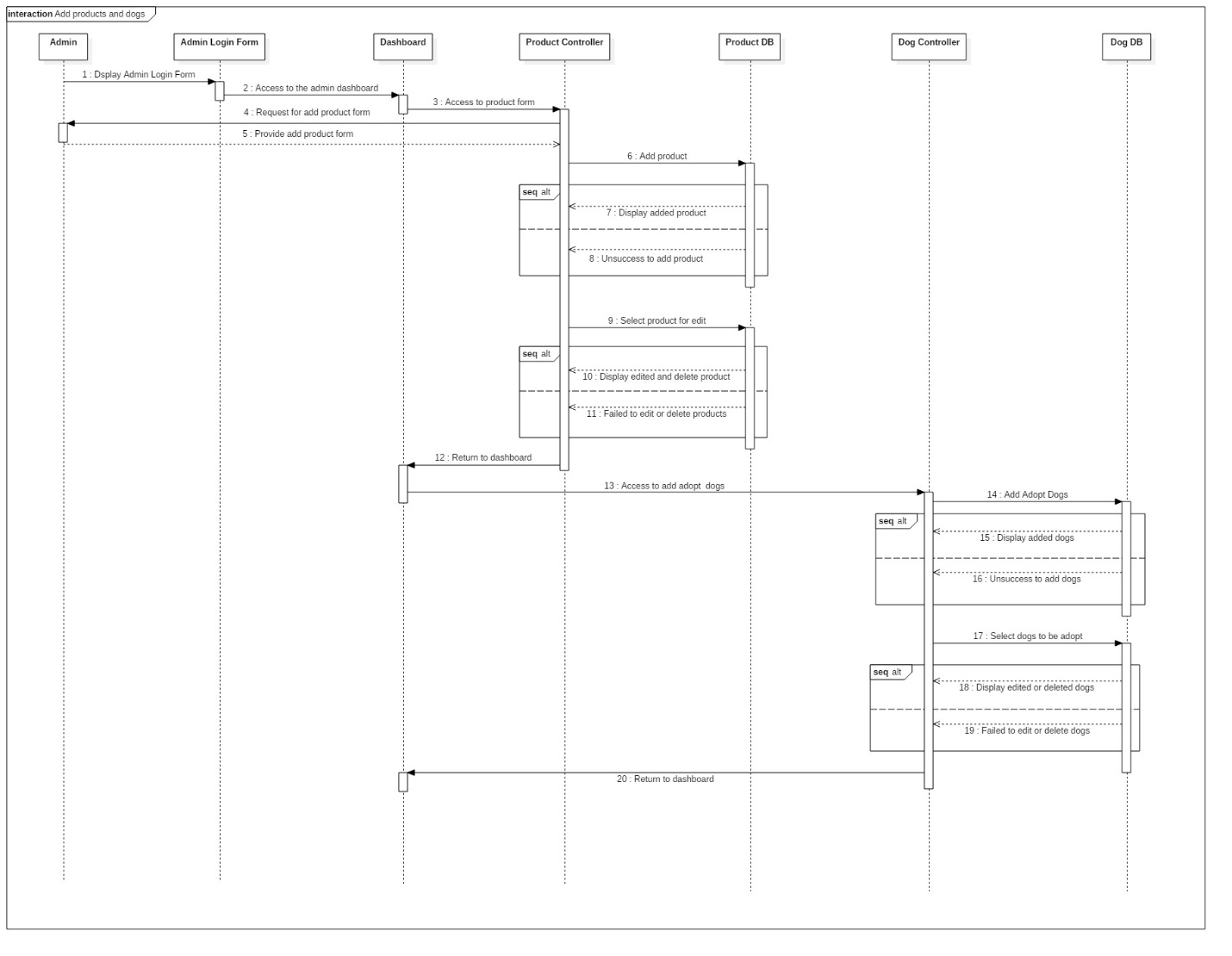
User Registration



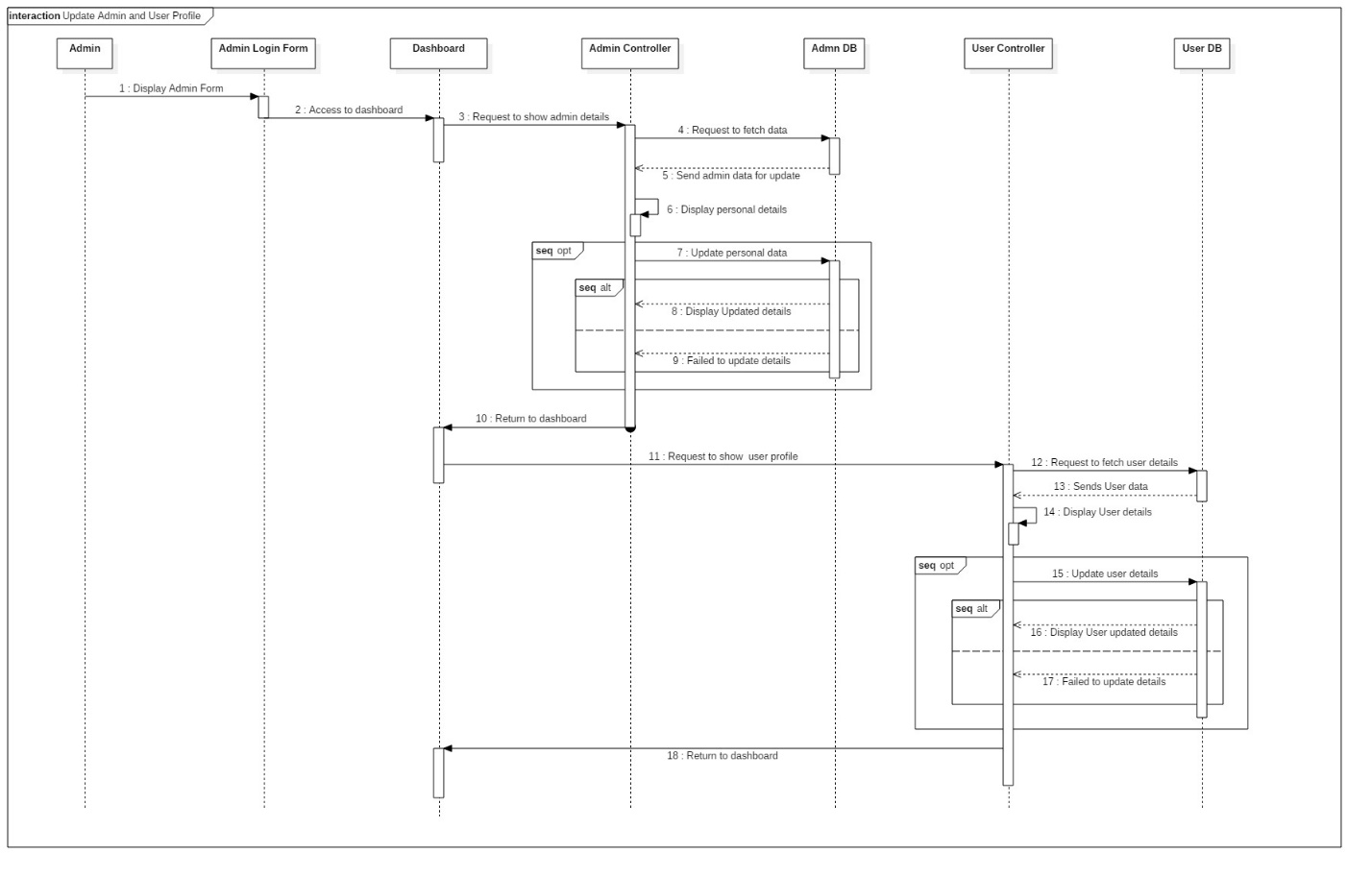
User Login



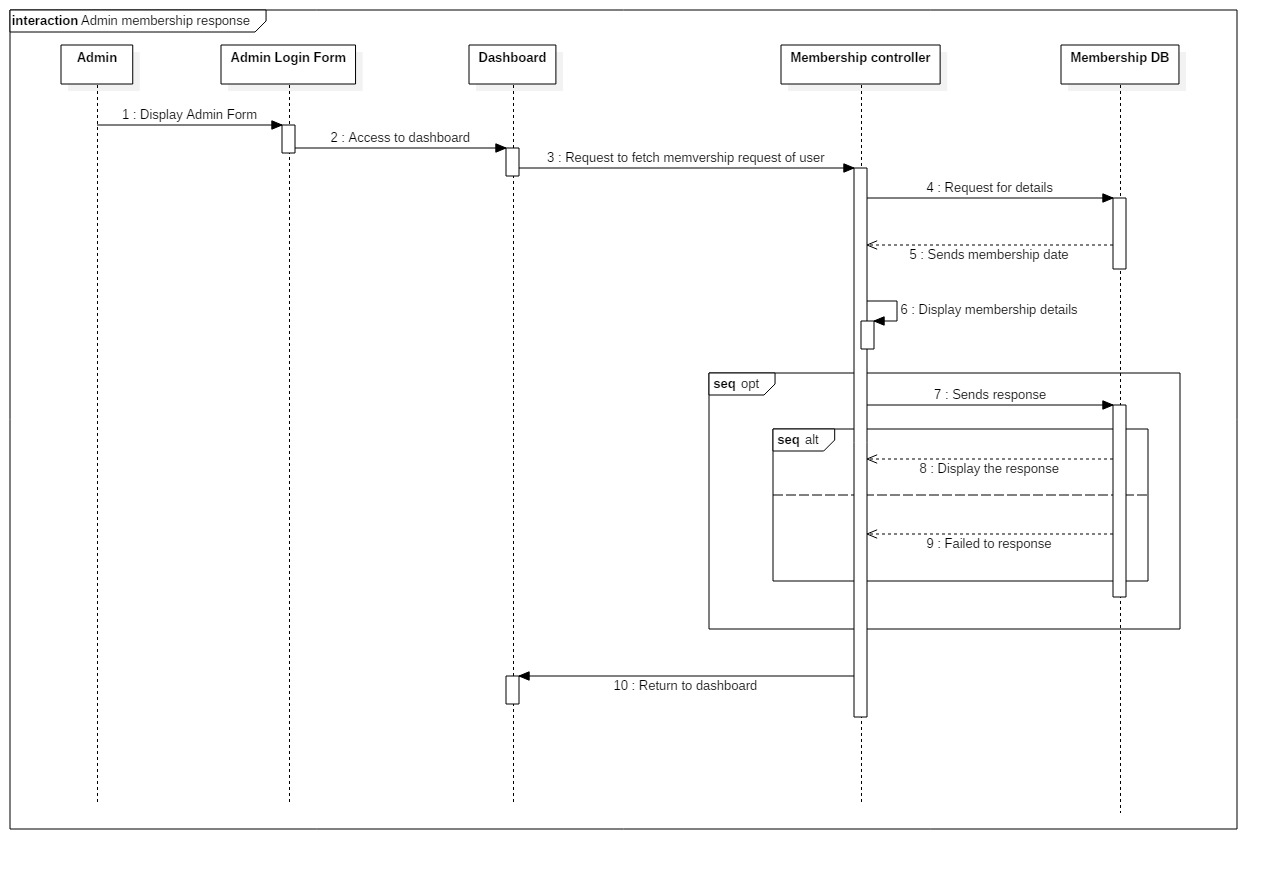
Add products and dogs:



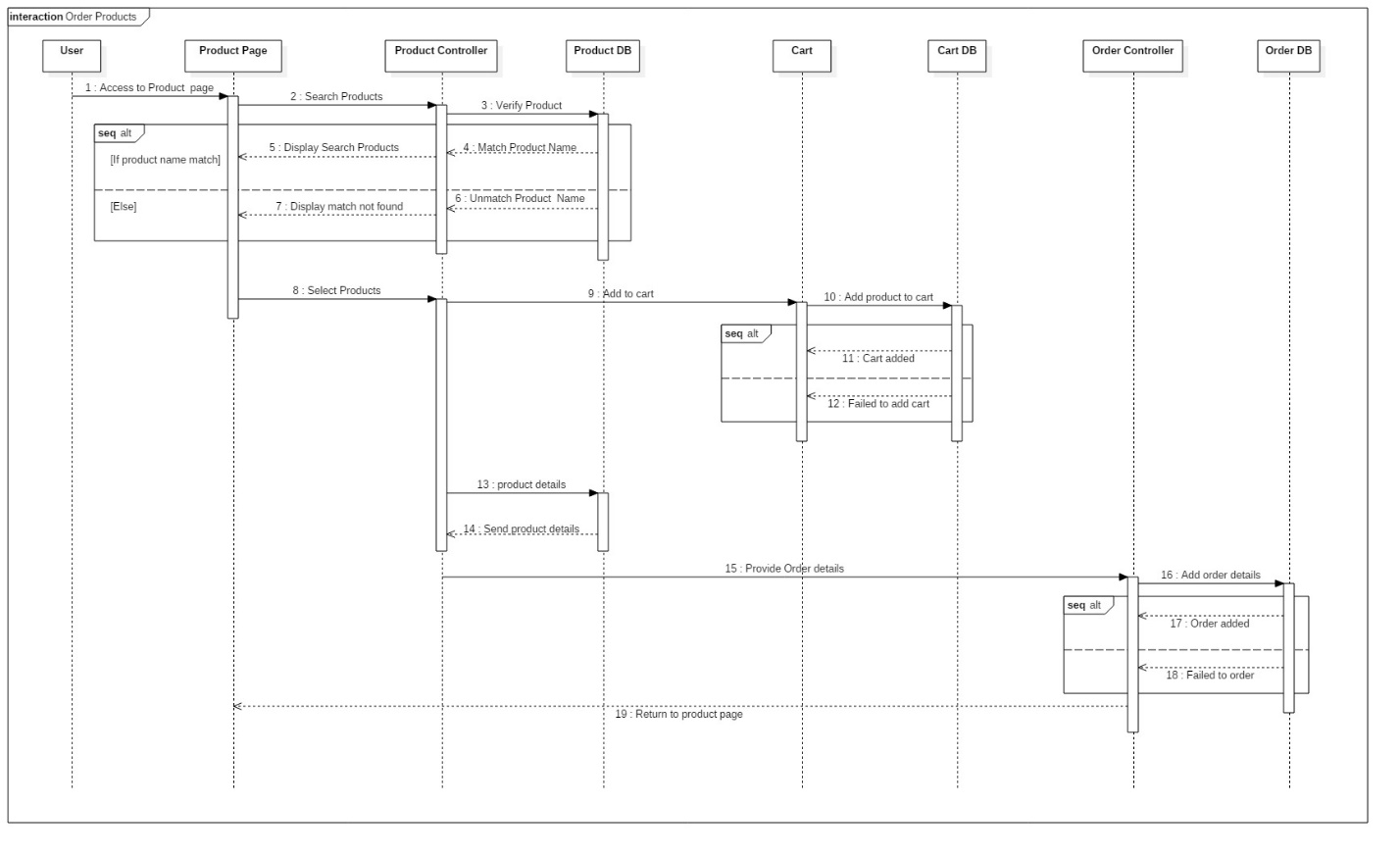
Update Admin and User Profile



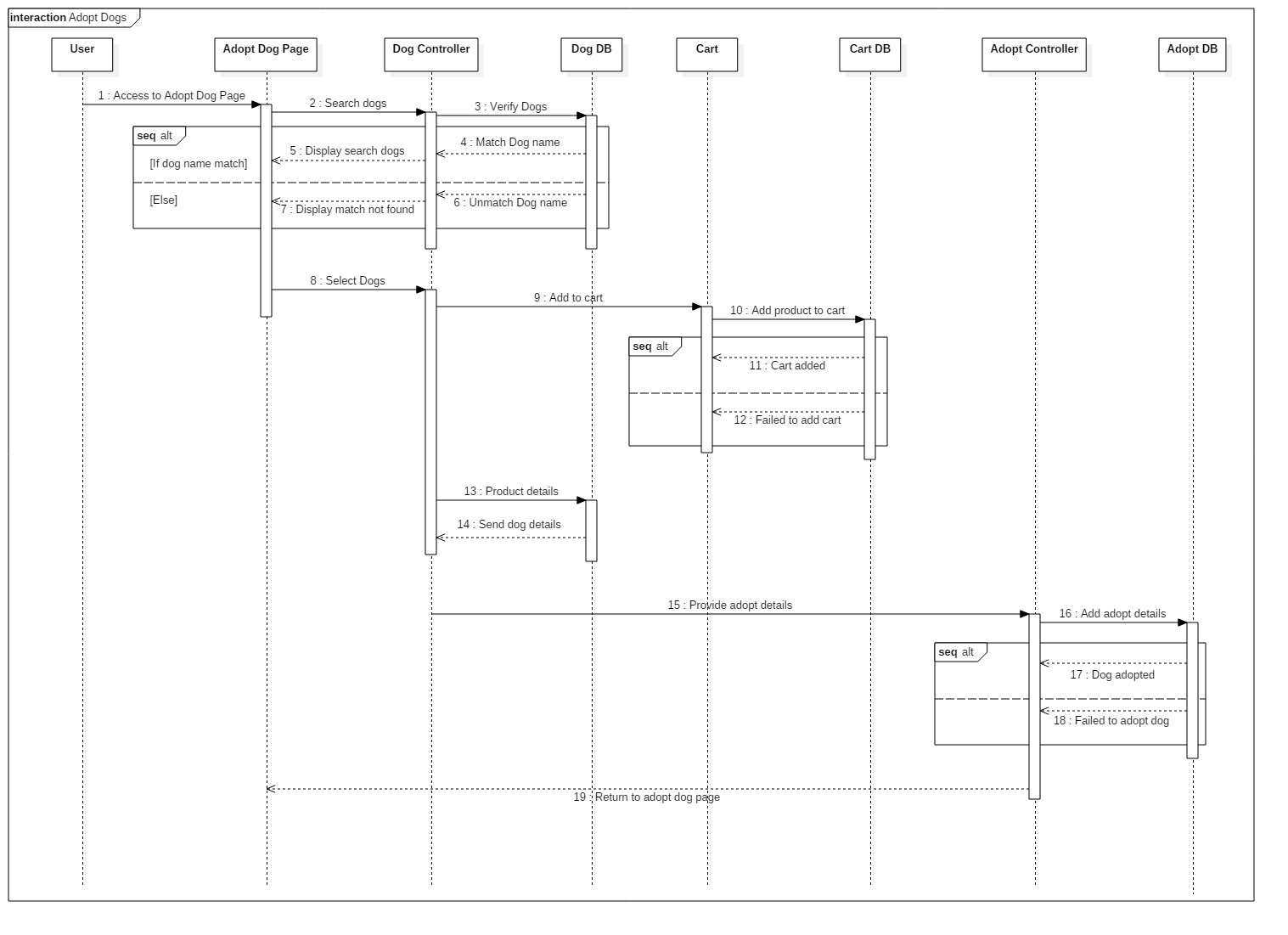
Admin reponse to membership request



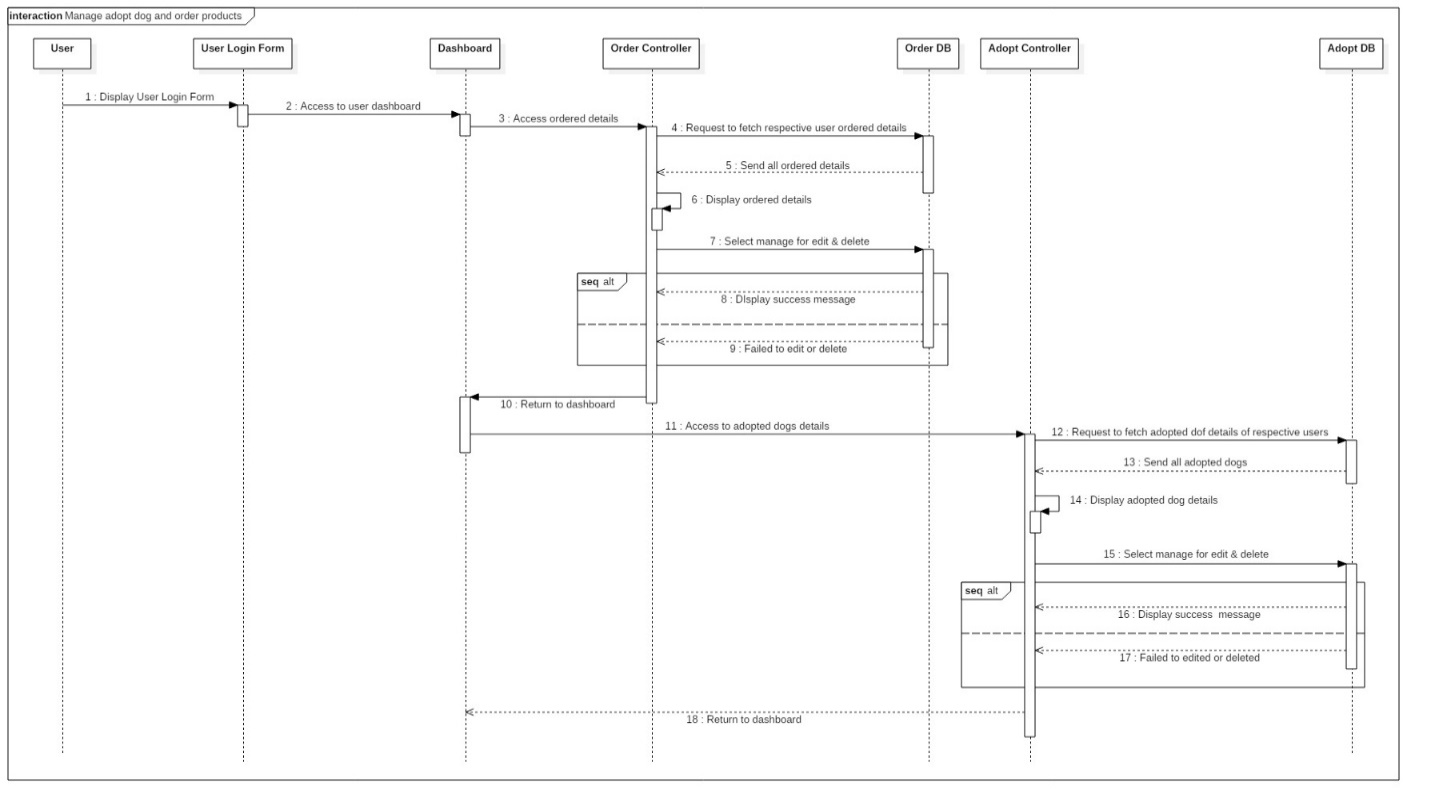
Order Products



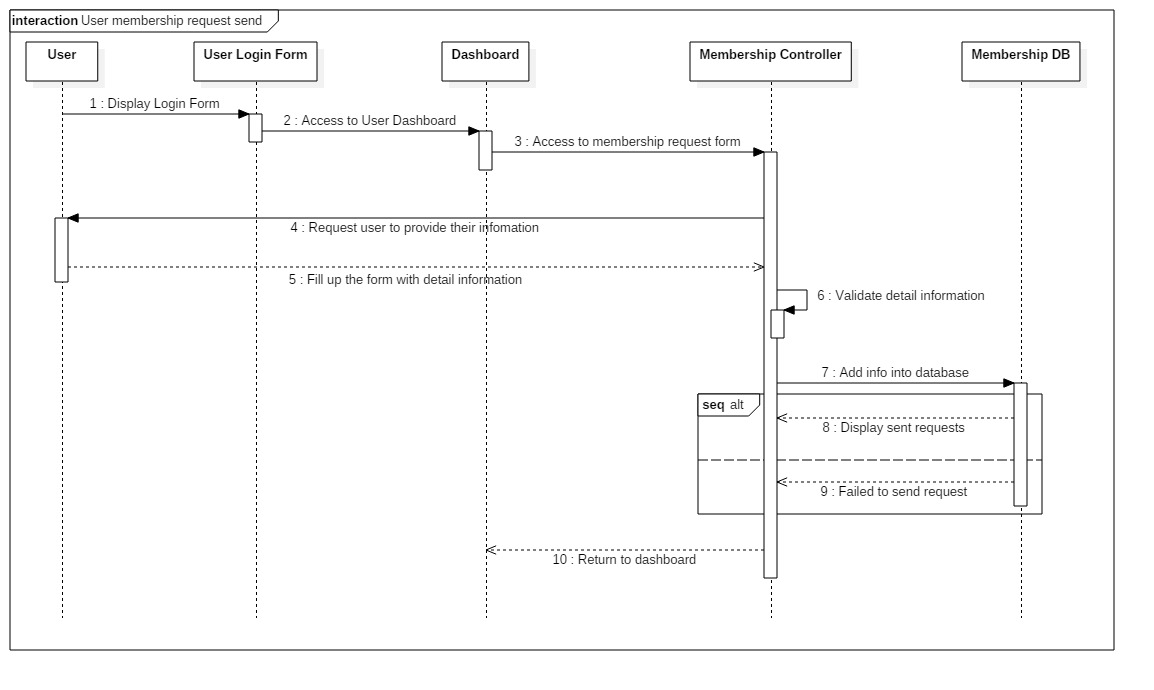
Adopt Dogs



Manage Adopt dogs and order products



User Membership request send



## 3.3 – Database Modeling

Database Modeling is process of illustration of the data in a logical structure. It shows the relationship of entity and shows the method of the data to be store and retrieve.

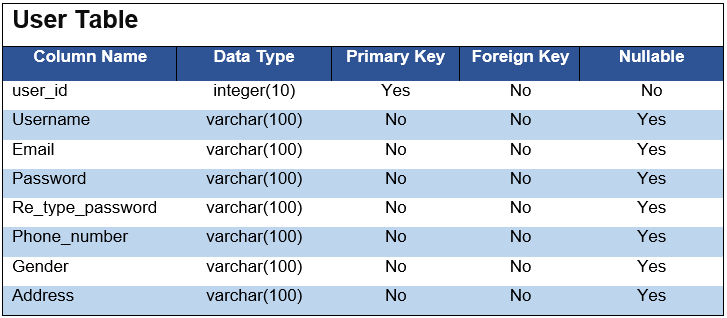
### 3.3.1 – Data Dictionary

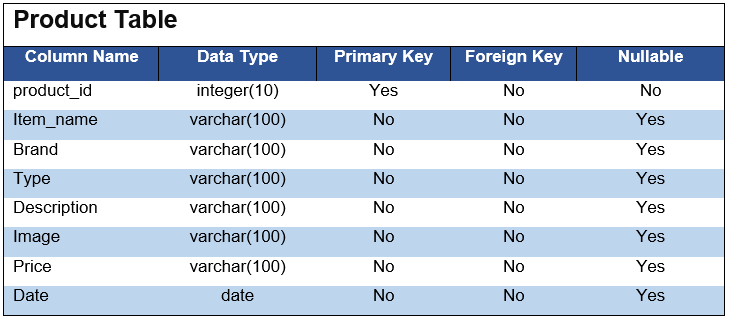
Data Dictionary is a set of information which contains the metadata of the database. It is also known as “Data Definition Matrix(DDM)”. It describes the column name with their length, datatypes, etc.

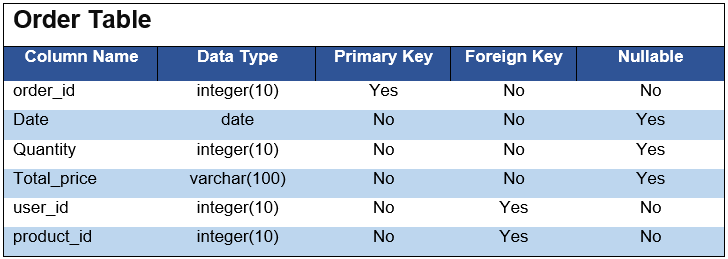
The information/metadata include:

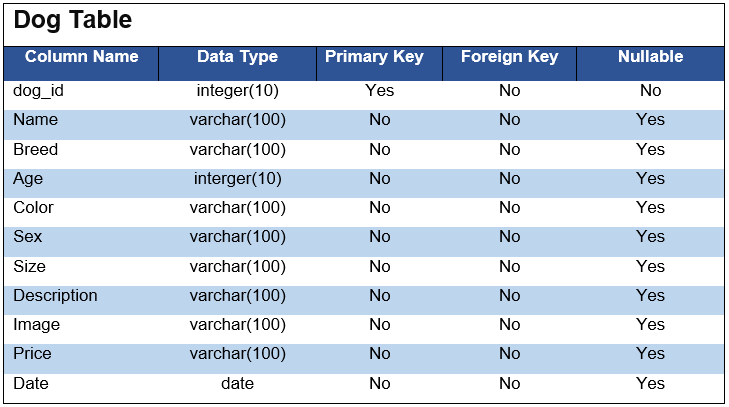
* Attribute name
* Datatype,
* Length
* Constrains
* Default value
* Description etc.

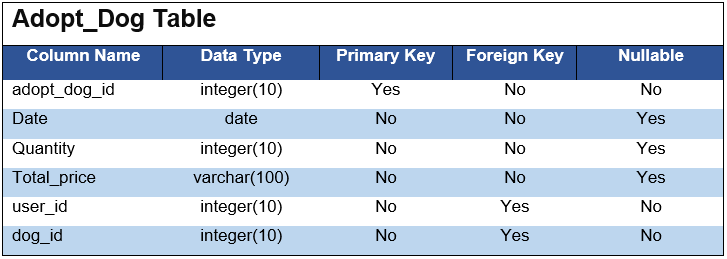
The data dictionary of our database model are as follow:

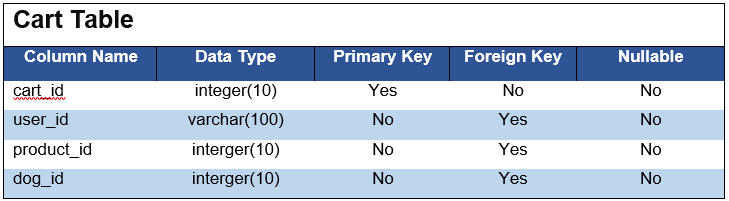


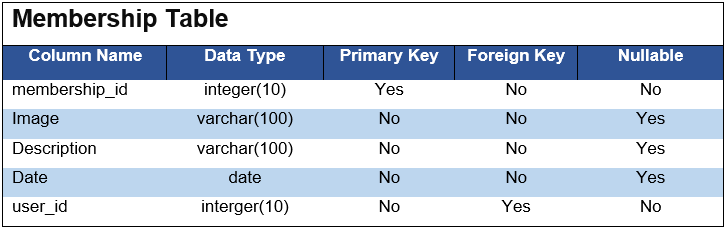


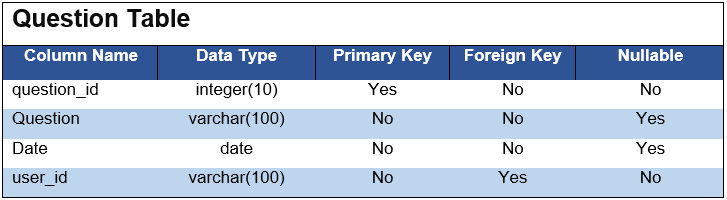


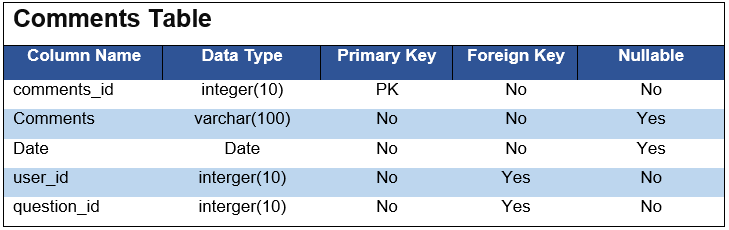












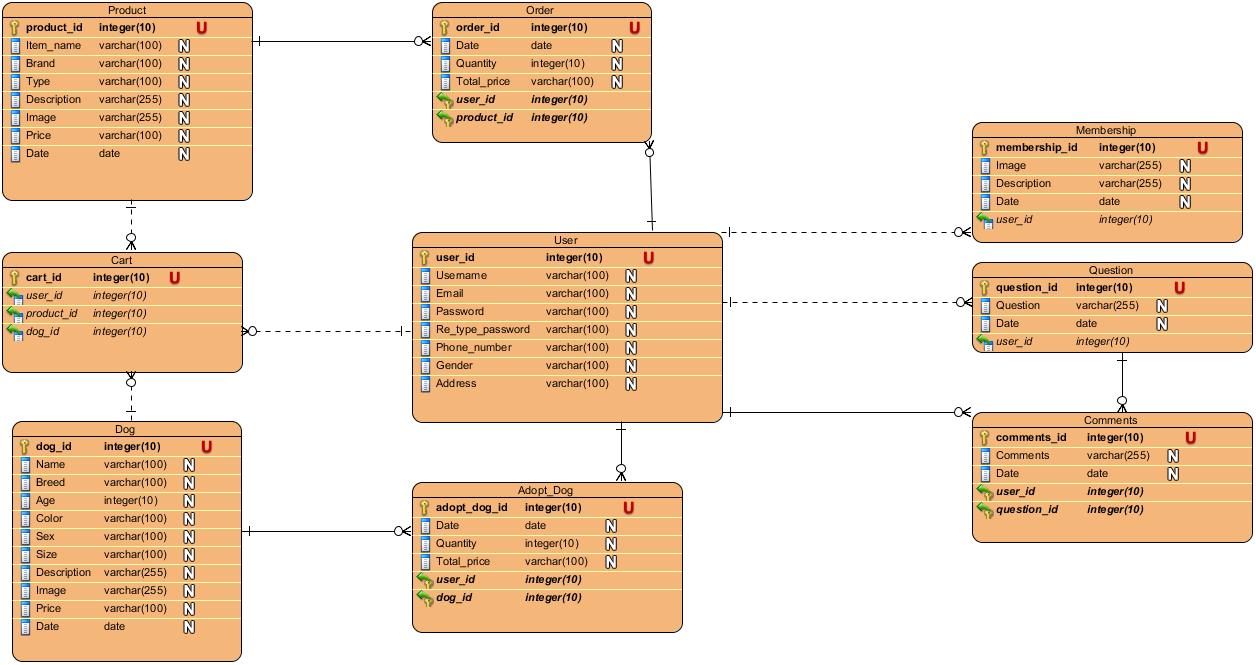
### 3.3.2 – ER-Diagram

Entity Relation Diagram is the graphical representation of the relationship between set of entity stored in a database. It represents the logical structure of the database. It is also known as the data modeling technique.

Justification of using ER diagram in my project are listed below:

* It helps to show the relationship between the entity.
* It helps to represent the logical structure of the database.
* It is easier and understandable diagram.

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Notation Used** | **Notation Name** | **Description** |
| **1.** |  | Entity | Helps to represent the entity of the table. |
| **2.** |  | One to one | Helps to represent the one to one relationship of an entity. |
| **3.** |  | One to many | Helps to represent the one to many relationship of an entity |
| **4.** |  | Many to many | Helps to represent the many to many relationship of an entity |

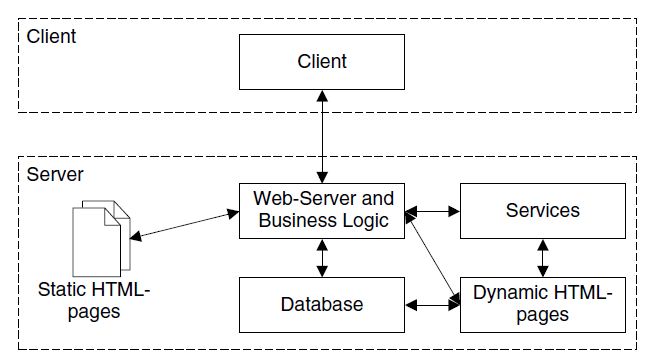


## 3.4 – Architecture Diagram

Architecture diagram is a graphical representation of a set of concepts, that are part of an architecture, including their principles, elements and components. Architecture diagram helps to represents the structure of an object

* **Client-server architecture:**

Client Server Architecture is the model where the client sends the requests and services to the server and then the server manages the services & request and send back to the clients. Servers are a powerful computer so many clients can connect to a single server.



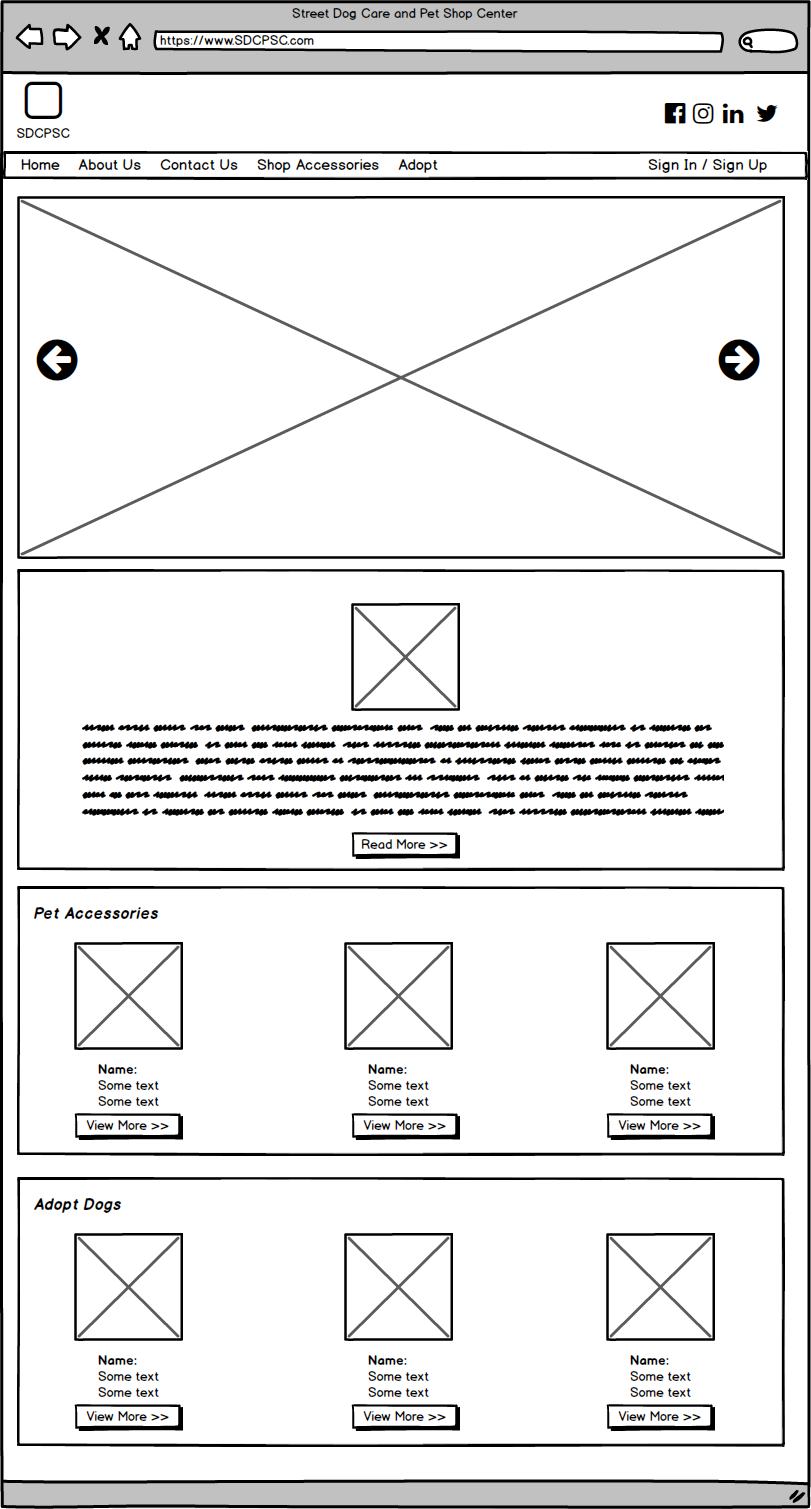
I have chosen Client-Server Architecture because this project is based on the online transactions between the customer and the system. Server have better control access and resources to ensure that only authorized client can access and manipulate data. The clients send the request to the web-server then the web-server forwards the data to the database, services, dynamic HTML pages or Static HTML pages. Then as the client request the server replies the data which the clients want.

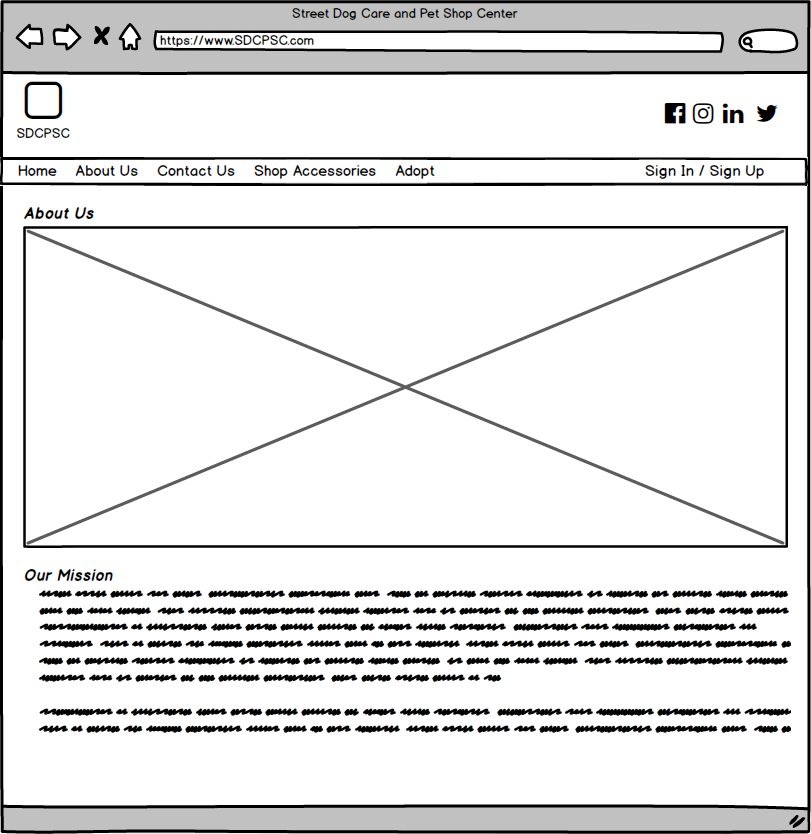
Justification of using Client-Server Architecture in my project are listed below:

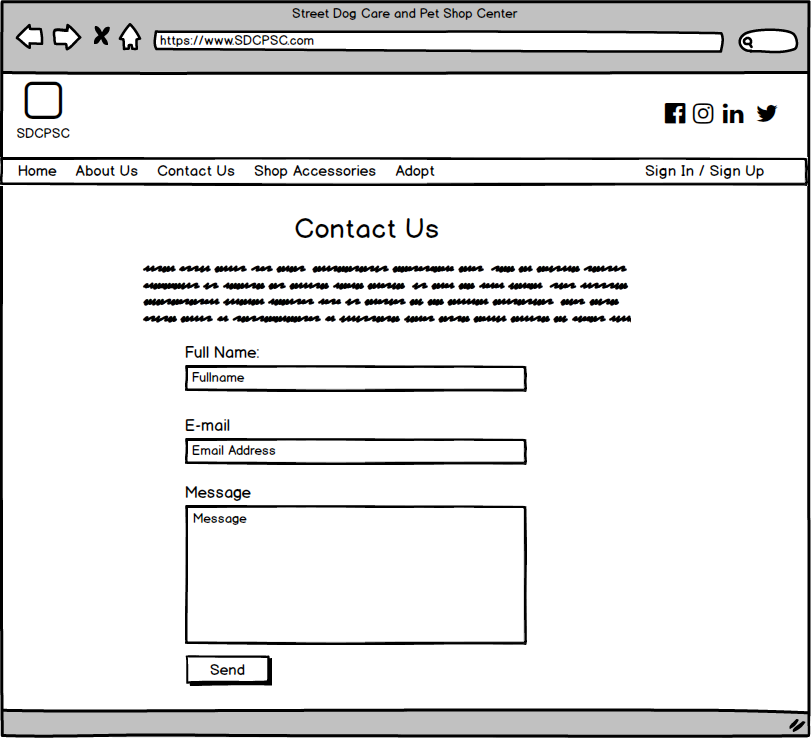
* It can be easily access by the client from any places through internet.
* It helps to protect the data with access control.
* It enforces the security policies on the data.
* The cost is reliable.

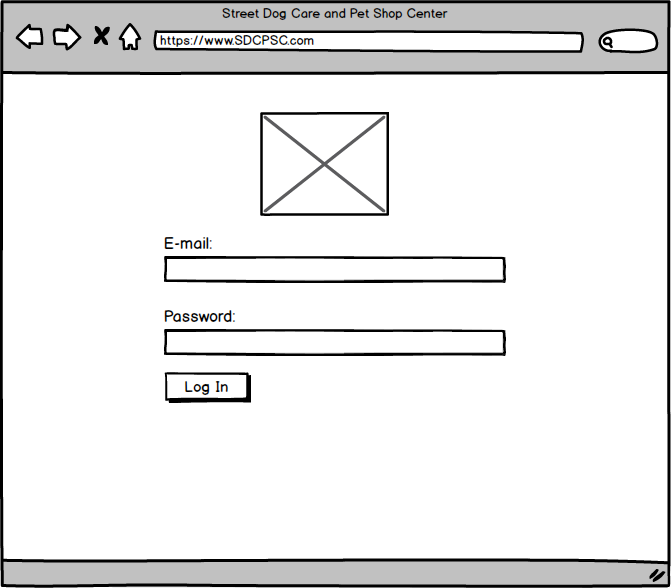
## 3.5 – Prototype Design

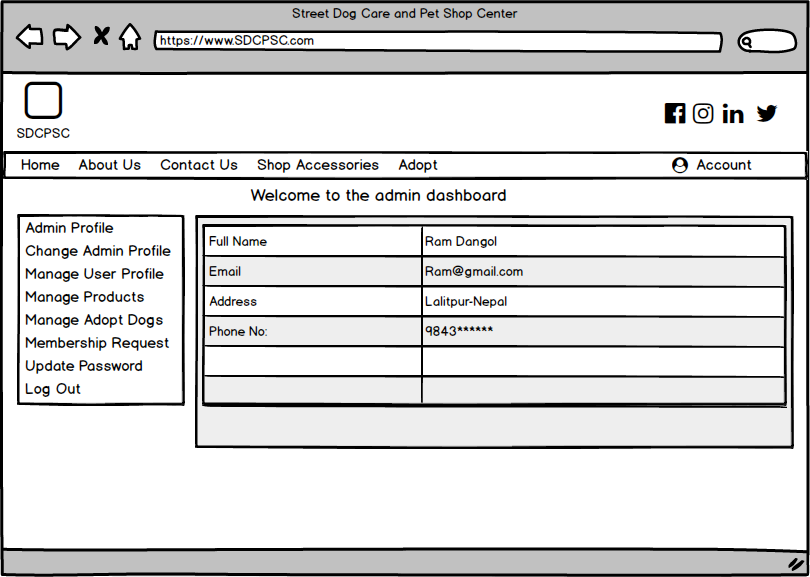
Prototype design is the graphical user interface design of the system. It is also known as the blueprint of the system interface. It shows the interface of a new system. The designer and developer will develop the system as the prototype design of the system. I have used ***Balsamiq Mockups 3*** software to make the below prototyping design











## C:\Users\DELL\Desktop\Diagram\Design\Prototype\View Ordered Details.pngC:\Users\DELL\Desktop\Diagram\Design\Prototype\View Adopted Details.pngC:\Users\DELL\Desktop\Diagram\Design\Prototype\User Login Form.pngC:\Users\DELL\Desktop\Diagram\Design\Prototype\User Dashboard.pngC:\Users\DELL\Desktop\Diagram\Design\Prototype\Update User Profile.pngC:\Users\DELL\Desktop\Diagram\Design\Prototype\Update User Profile by admin.pngC:\Users\DELL\Desktop\Diagram\Design\Prototype\Update User Password.pngC:\Users\DELL\Desktop\Diagram\Design\Prototype\Update Products.pngC:\Users\DELL\Desktop\Diagram\Design\Prototype\Update Adopt Dogs.pngC:\Users\DELL\Desktop\Diagram\Design\Prototype\Update Admin Profile.pngC:\Users\DELL\Desktop\Diagram\Design\Prototype\Update Admin Password.pngC:\Users\DELL\Desktop\Diagram\Design\Prototype\Sign Up Form.pngC:\Users\DELL\Desktop\Diagram\Design\Prototype\Shop Details.pngC:\Users\DELL\Desktop\Diagram\Design\Prototype\Shop Accessories.pngC:\Users\DELL\Desktop\Diagram\Design\Prototype\Membership Request.pngC:\Users\DELL\Desktop\Diagram\Design\Prototype\Membership Request Form.pngC:\Users\DELL\Desktop\Diagram\Design\Prototype\Manage User Profile.pngC:\Users\DELL\Desktop\Diagram\Design\Prototype\Manage Products.pngC:\Users\DELL\Desktop\Diagram\Design\Prototype\Manage Adopt Dogs.pngC:\Users\DELL\Desktop\Diagram\Design\Prototype\Comments.pngC:\Users\DELL\Desktop\Diagram\Design\Prototype\Adopt Dog Details.pngC:\Users\DELL\Desktop\Diagram\Design\Prototype\Adopt .pngC:\Users\DELL\Desktop\Diagram\Design\Prototype\Add to cart.pngC:\Users\DELL\Desktop\Diagram\Design\Prototype\Add Products.pngC:\Users\DELL\Desktop\Diagram\Design\Prototype\Add Adopt Dogs.png