# Chapter 3

# Design

It is the third stage of software development life cycle (SDLC) which is very important to start implementation or coding stage. It is the most crucial phase in SDLC. In this stage SDLC process continues to move from what questions of analysis stage to the how. In the design stage the programming language, hardware, software platform in which new system will run are also decided.

Importance of design in my project are as follows:

* It helps in contingency, training, maintenance and make operational plan for my project.
* It helps my projects SRS documents into logical structure that contains details and complete set of specification that can be implemented.
* It helps in ensuring the final design has meet the requirements in SRS document.
* Finally it helps in preparing the design documents which will be used in next stage.

For my Project I will be designing following models:

* Structural Model.
* Behavioral Model.
* Architectural Model.
* Database Model.
* Prototype.

## 3.1 Structural Model

Structural Model is a model that exhibits the boundary level that would cause a default considering value of assets and capital. It shows the relations between different components.

### 3.1.1 Final Class Diagram

Class diagram is an illustration of relationships and source code dependencies among classes in UML (Unified Modeling Language). Here classes are arranged in groups which have common characteristics. It resembles a flowchart where classes are kept in boxes which have three box inside where top contains name of class, middle contains attributes and bottom contains operations. Lines is used to connect many boxes which have relationships between the classes.

It’s important for project because of following reasons:

* It helps in analysis and design of static view of my system.
* It helps to describe the responsibilities of individual system.
* It helps to deploy the diagrams we made easier.

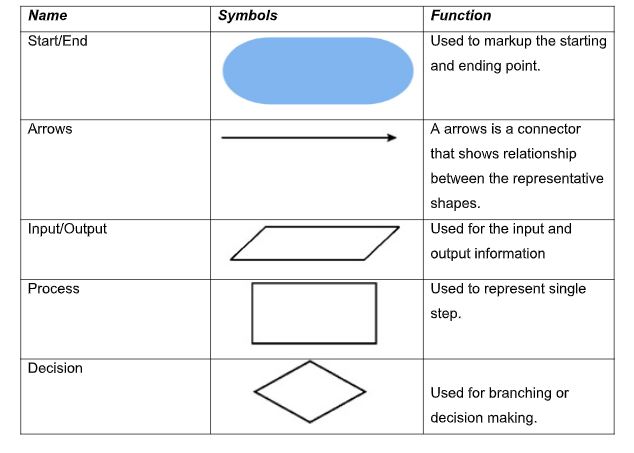
### 3.1.2 Flow Chart Diagram

Flowchart is a diagram that depicts a process, system or computer algorithm. It is sometimes spelled as flowcharts, which uses rectangles, oval, diamonds and potentially numerous other shapes to define the type of step.

I have used it in my project because of following advantages in my project:

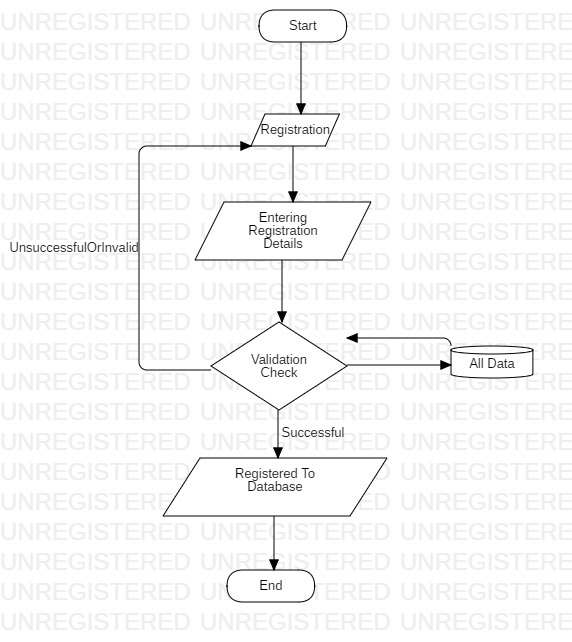
* It helps to visualize the multiple progress and their sequence in single document.
* It helps to eliminate the unnecessary steps.
* It helps to show only the required steps to reach the end fast.

Symbols used for Flowchart diagram are:



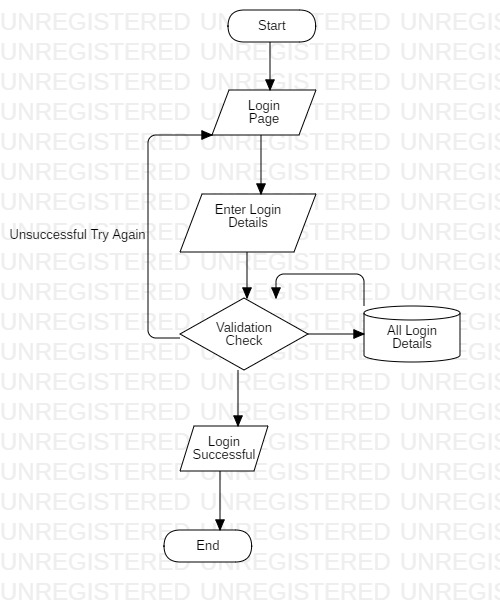
Screenshot: Symbols Details

All the flowcharts for my projects are:



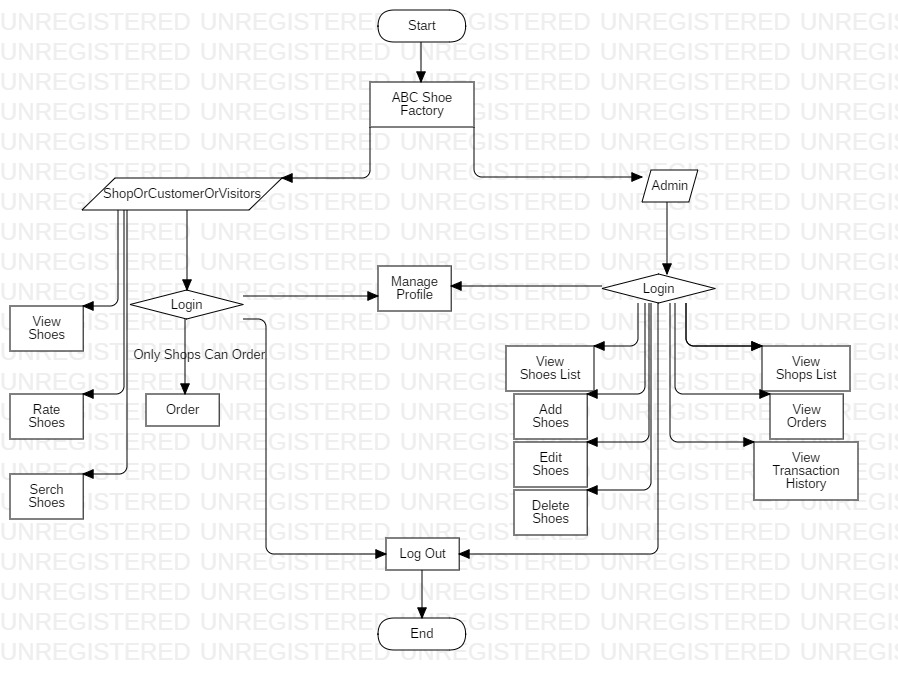
Screenshot: Flowchart for Registration

In the above diagram Admin and Shop or Customers can register to get login access. They must fill the form or details required. After that system checks if the given details are valid or not and if it’s not valid it will take back to registration page and if details are valid then registered to database.



Screenshot: Flowchart Login Page

In the above diagram Users can use their details to login but if the details are invalid it will reload login page and if details are valid users will reload to their dashboard.



Screenshot: Flowchart Of Admin And Shops Dashboard

In the above diagram, there are two types of users. Firstly, after Shops or Customers logged in they can order shoes edit profile but for viewing shoes no login required.

Finally after admin are logged in they can View list of shoe, add new shoes, edit shoes, delete shoes. They can also see the list of profile of shops, view orders and transaction made with shops.

## 3.2 Behavioral Model

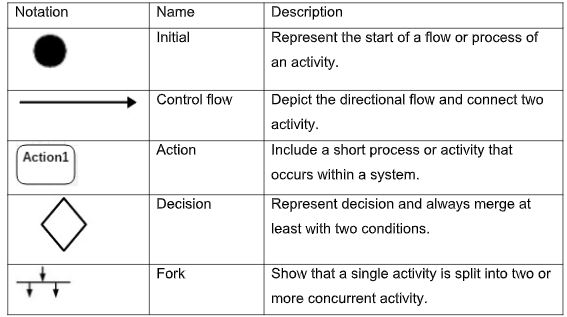
It is a behavioral approach to systems theory and control the theory which was established in 1970 by J. C. Willems. It helps in resolving inconsistencies present in classical approach. Its important feature is that it does not distinguish between input and output variables.

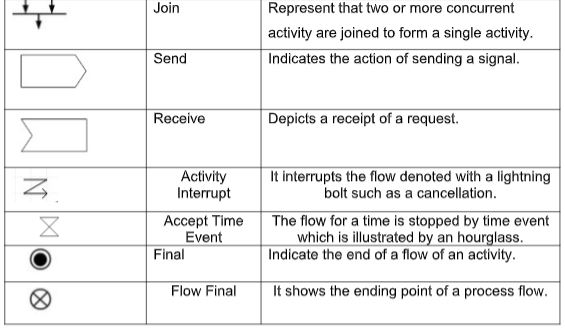
### 3.2.1 Activity Diagram

Activity diagram is a diagram which is also one of important behavioral diagram which describes the dynamic aspects of the system. It is advanced form of flowchart which models the flow of one activity to another.

Features of activity diagram in my project are:

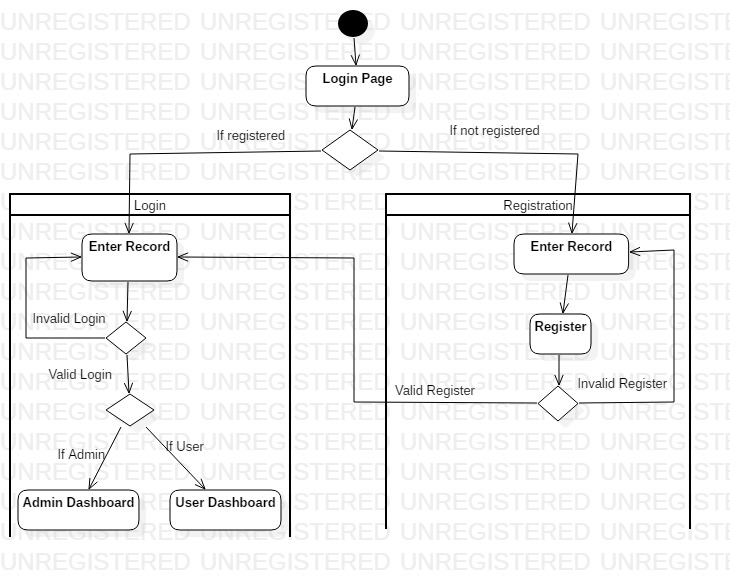
* It helps to describe the parallel, branched and concurrent flow of system.
* It helps to outline the high level activity in system.
* It helps in investing the system in later stage.





Screenshots: Symbols used in Activity Diagram

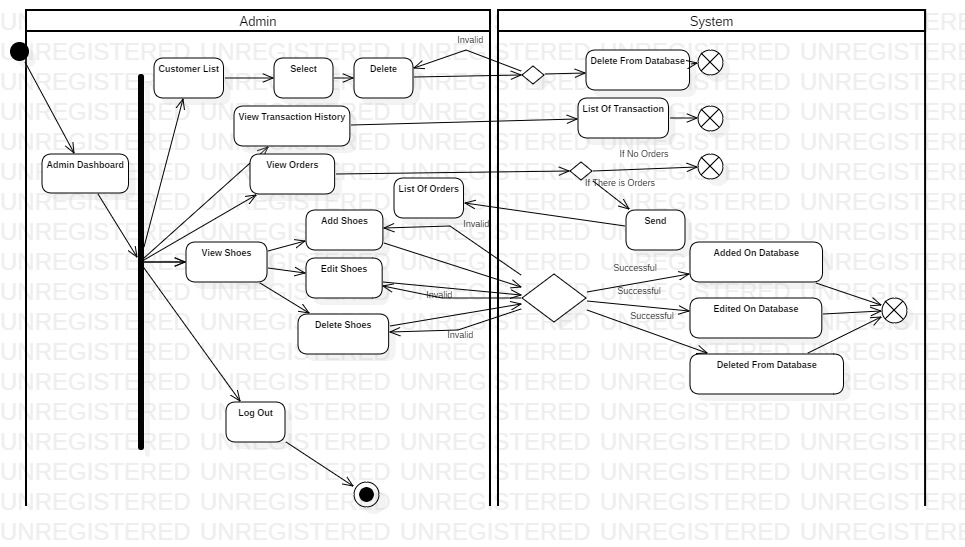
1. Login and Registration

****

Screenshot: Activity Diagram

Here admin and shops or customer can register the account with the detail required which will be saved in database. So they can also login to the system to get the more function where admin member will login to the admin dashboard and shops or customer will login to the customer dashboard.

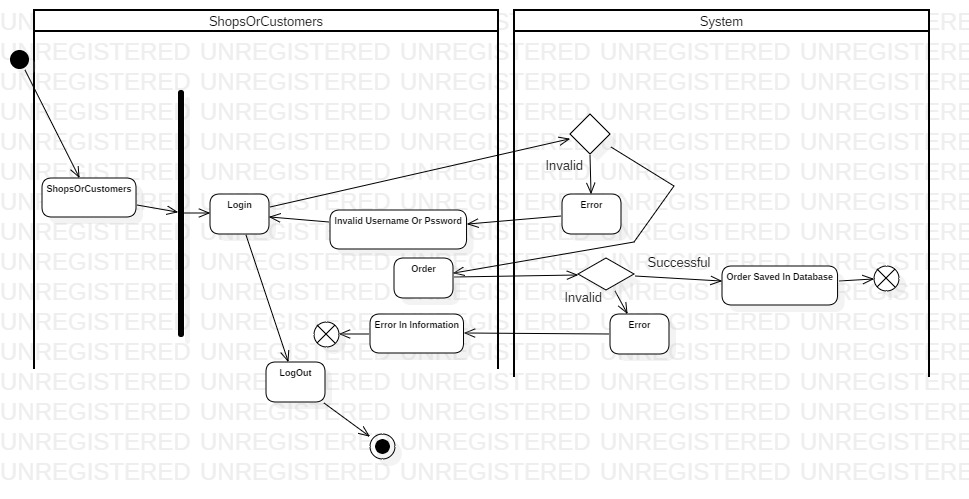
2) Admin Function



Screenshot: Activity Diagram

Admin can do various function after logged in. They can view the list of customers and delete also. They can see all the transaction made with shops also. The view the list of shoes, add new shoes, edit shoes, delete shoes and see all the order also. Finally they can logout from system.

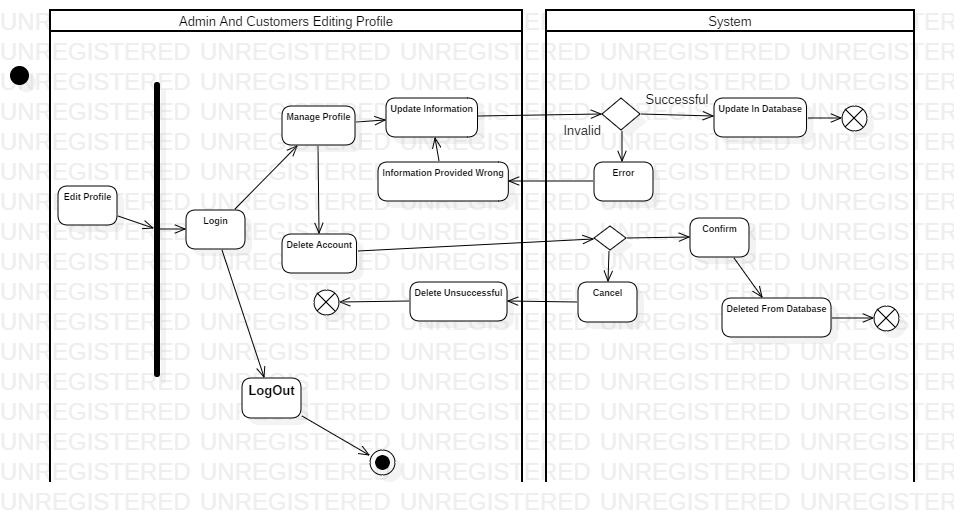
3) Shops function



Screenshot: Activity Diagram

In above diagram after logged in shops can order the shoes. It will process in system and if any error found in information they must do process again and if no error found their order will be saved in database.

4) Manage Profile



Screenshot: Activity Diagram

In above diagram, admin and customers can login their profile. They can manage profile and edit or update the information they want. They can also delete their account from manage profile if they want. And finally they can logout.

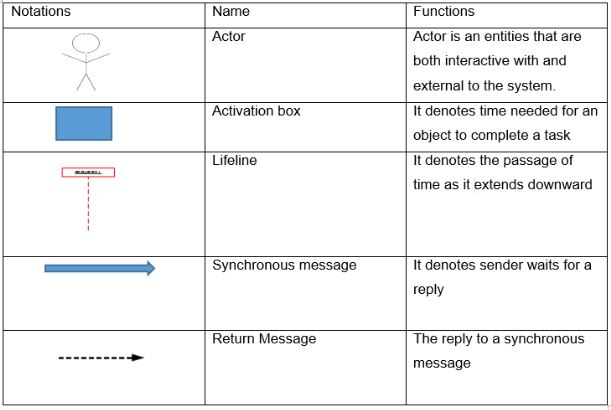
### 3.2.2 Sequence Diagram

Sequence diagram is diagram that shoes the object interactions arranged in time sequence.

Features of sequence diagram in my project are as follows:

* It helps in high level interaction between user of the system and the system, between system and other systems or between sub systems.
* It helps in interaction between objects within a collaboration that realizes an operation.

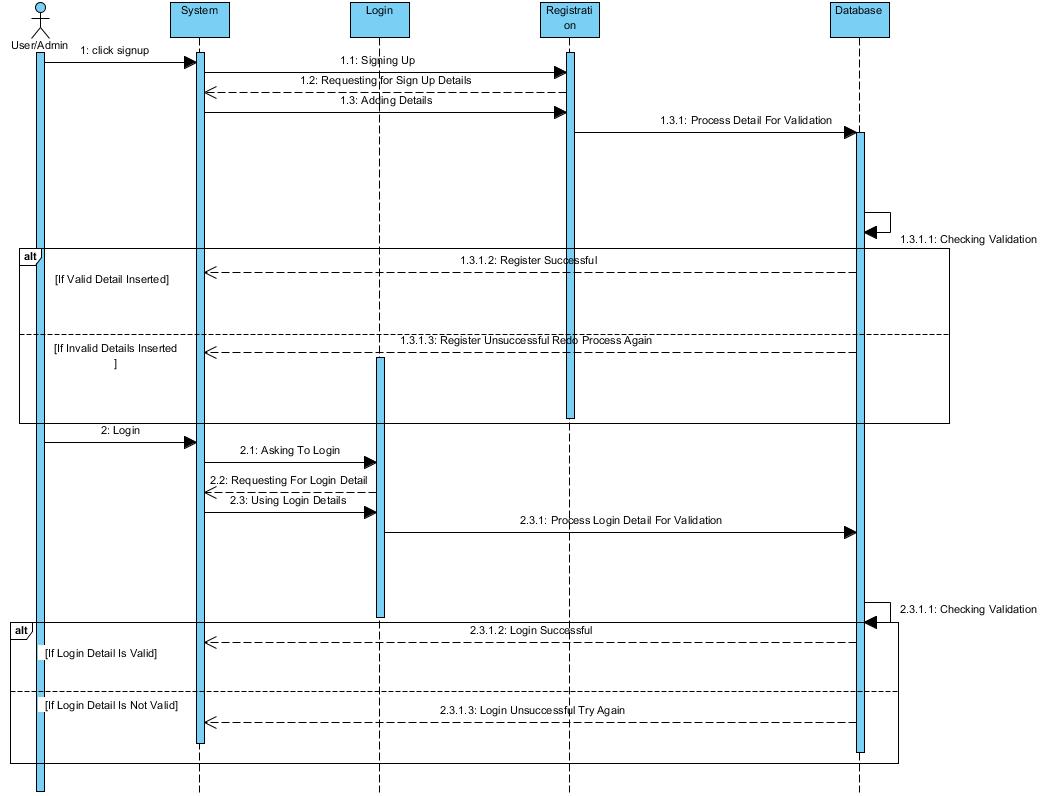
Notation used in my diagram are as follows:



Screenshot: Notation Used in Sequence Diagram

Sequence diagram for my project are as follows:

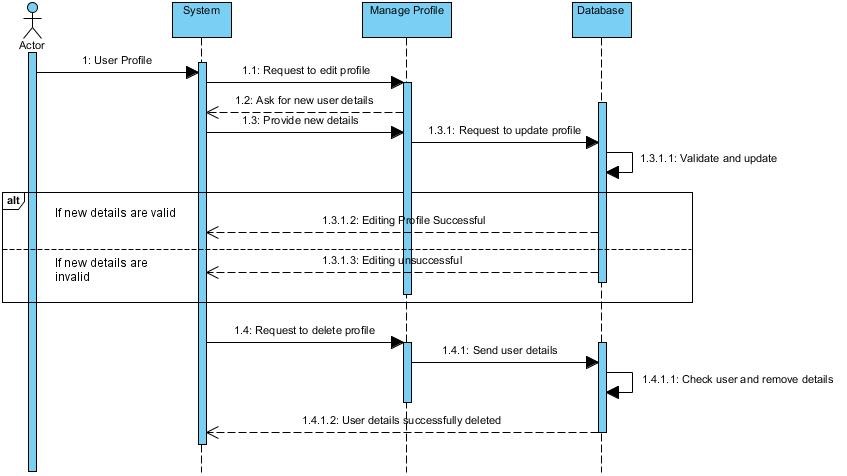
1. Registration



Screenshot: Sequence Diagram

In above diagram, all user admin and shops or customers can register and make profile which will added in data base. They can also login with the details the used to register. They can also logout.

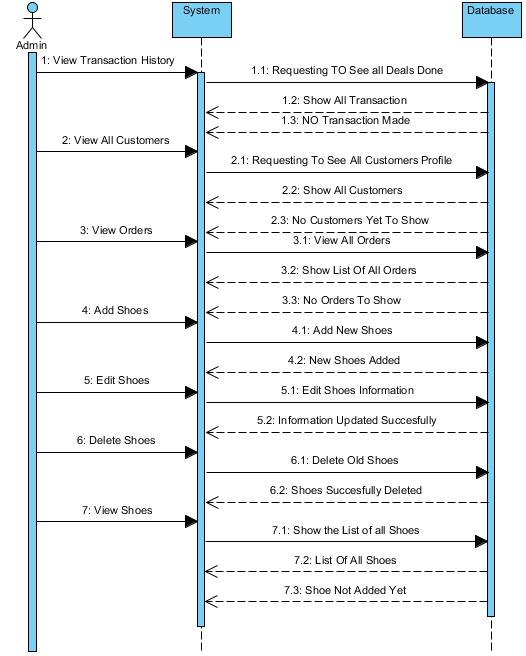
1. User Manage Profile(Admin and Customers)



Screenshot: Sequence Diagram

In the above diagram, all the users admin, shops or customers can manage profile and edit their information and also can delete their profile from database.

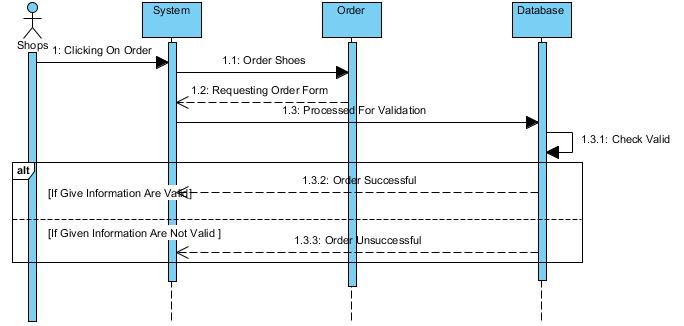
1. Admin Functions



Screenshot: Sequence Diagram

In the above diagram, it show the function of admin where they can view all the transaction made. They can see the accounts of shops or customers, see all the orders made by shops. They can also add new shoes, edit the shoes, delete shoes and see the list of all shoes.

1. Shops Or Customers Function



Screenshot: Sequence diagram

In above diagram, Shops can Order the shoes online and the order will be saved in database where admin can see later.

## 3.3 Database Modeling

Database model is the type of data model that determines the logical structures of a database and determines in that manner where data can be organized, manipulated and stored. One of the popular example of database model is relational model which uses table format.

### 3.3.1 Data Dictionary

A data dictionary or metadata is defined as centralized repository of information about data as meaning relationships to other data, origin, usage and format as defined in IBM dictionary of Computing. Oracles defines it as collection of tables with metadata.

Admin and Shop/Customers Registration Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Datatype | Length | Constraint | Null |
| admin\_id/customerid | integer | 1000000 | Primary Key | Not Null |
| name | varchar | 100 | - | Null |
| phone\_num | varchar | 20 | - | Null |
| membertype | varchar | 25 | - | Null |
| email | varchar | 80 | - | Null |
| address | varchar | 100 | - | Null |
| password | varchar | 255 | - | Null |

Shoes Registration Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Datatype | Length | Constraint | Null |
| shoesid | Integer | 1000000 | Primary Key | Not null |
| shoesname | Varchar | 50 | - | Null |
| details | varchar | 255 | - | null |

Online Order Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Datatype | Length | Constraint | Null |
| orderid | integer | 1000000 | Primary Key | not null |
| shoesname | varchar | 255 | - | null |
| shoesqty | varchar | 255 | - | null |
| address | varchar | 100 | - | null |
| shopname | varchar | 100 | - | null |
| phonenum | varchar | 20 | - | null |

Transaction History Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Datatype | Length | Constraint | Null |
| transid | integer | 1000000 | Primary Key | not null |
| details | varchar | 500 |  | null |

Admin\_Customer Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Datatype | Length | Constraint | Null |
| Adminadminid | integer | 1000000 |  | not null |
| Customercustomerid | integer | 1000000 |  | not null |

Admin\_Shoes Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Datatype | Length | Constraint | Null |
| Adminadminid | integer | 1000000 |  | not null |
| Shoesshoesid | integer | 1000000 |  | not null |

Customer\_Online\_Order Table

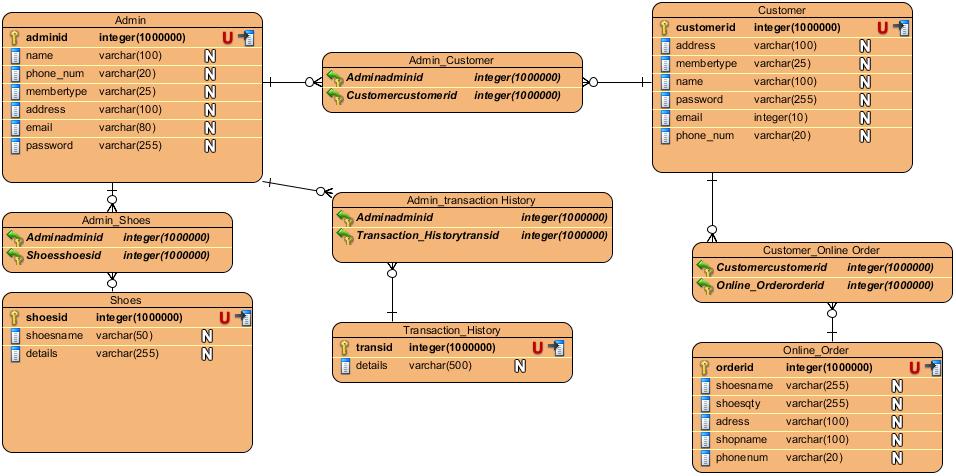
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Datatype | Length | Constraint | Null |
| Customercustomerid | integer | 1000000 |  | not null |
| Online\_Orderorderid | integer | 1000000 |  | not null |

Admin\_Transaction History Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Datatype | Length | Constraint | Null |
| Adminadminid | integer | 1000000 |  | not null |
| Transaction\_Historytransid | integer | 1000000 |  | not null |

### 3.3.2 ER Diagram

ER diagram (Entity Relationship Diagram) describes the interrelated things of interest in a specific domain of knowledge. ER diagram is composed of entity types which specifies the relationships that exist between entities. Each entities holds attributes and its data types.



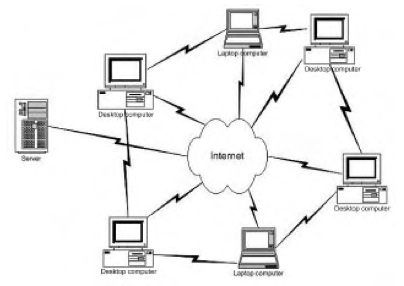
Screenshot: ER Diagram

## 3.4 Architectural Diagram

I have used peer to peer architecture to develop my project. It is one of the commonly used architecture in which each workstation, or node, has the same capabilities and responsibilities. Often it is compared and contrasted to classic client/server architecture.

Advantages of the architecture for my project are as follows:

* It is easy to set up and configure the computer in this network.
* Cost is very low to set up P2P network architecture.
* No need of full time system administrator.
* Central dependency is eliminated so it is more reliable.

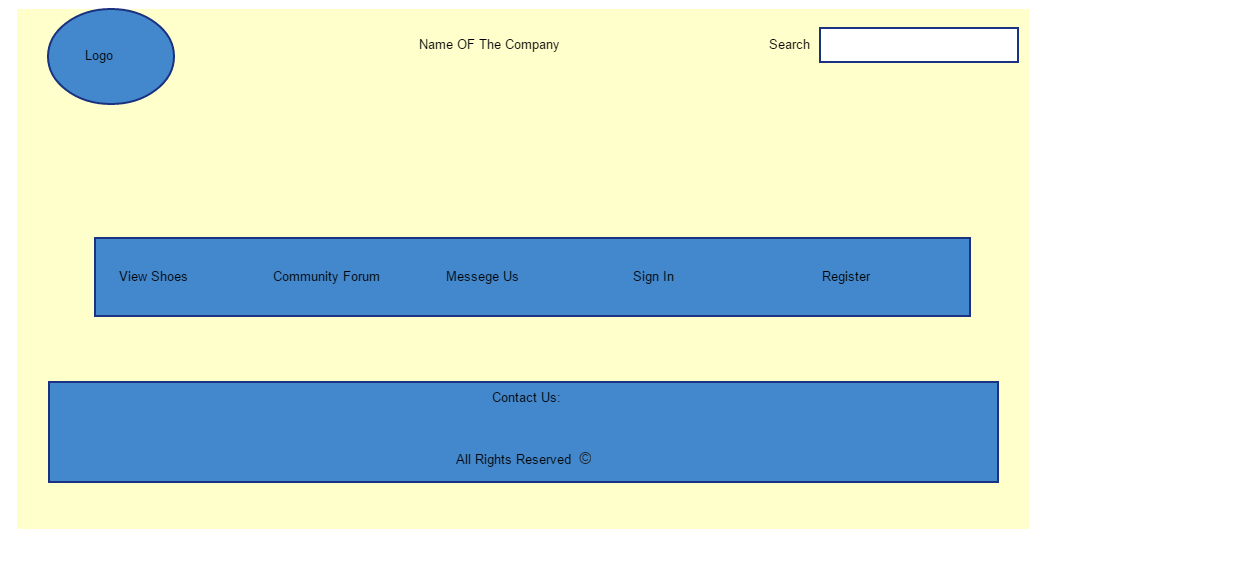


Screenshot: Peer to Peer architecture

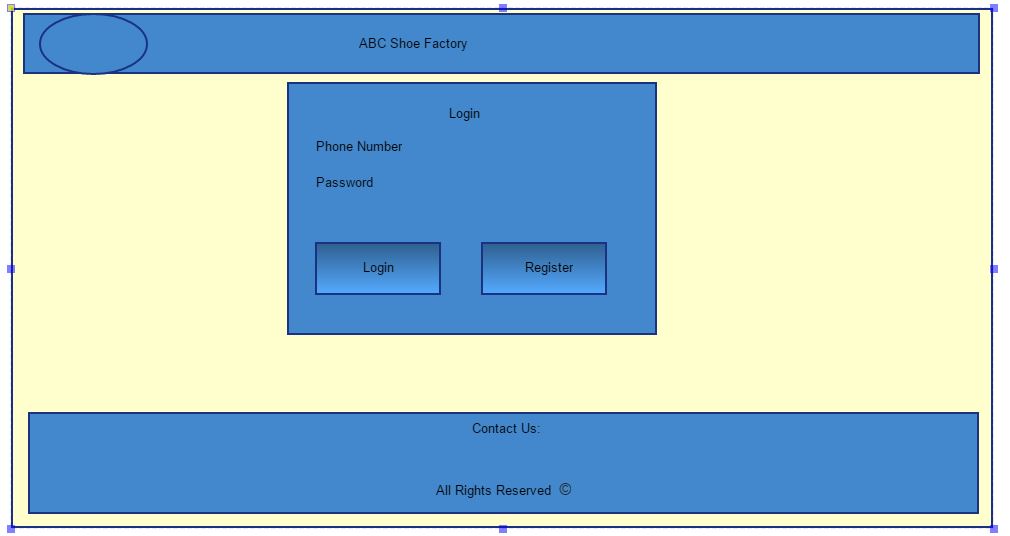
## 3.5 Prototype

Prototype is a sample, model of a system which is made to test a concept or process or to act as a thing to be replicated or learned from. There are many type of prototypes and among them I am using Visual Prototype. It represents the appearance, but not the functionality of intended design.

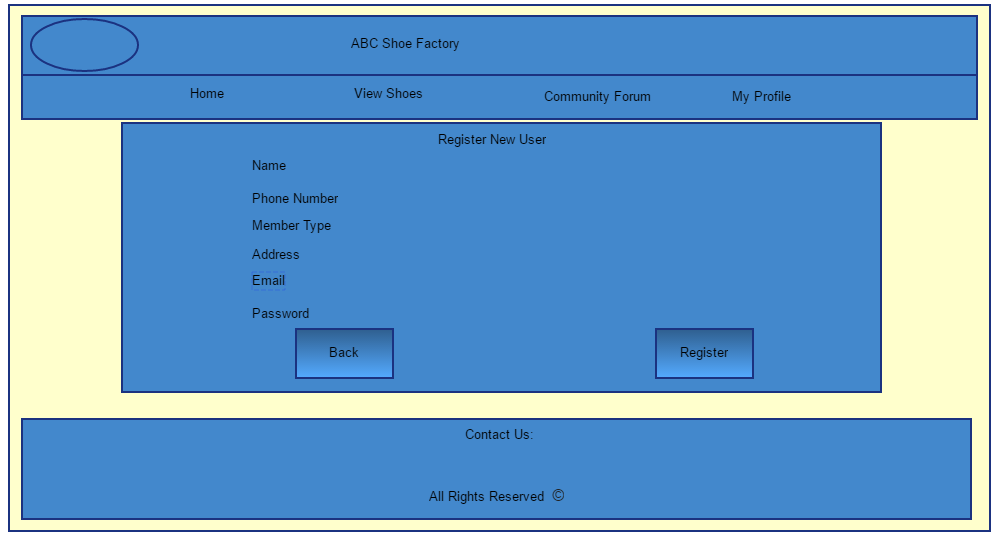
Prototype I have made for my system are as follows:



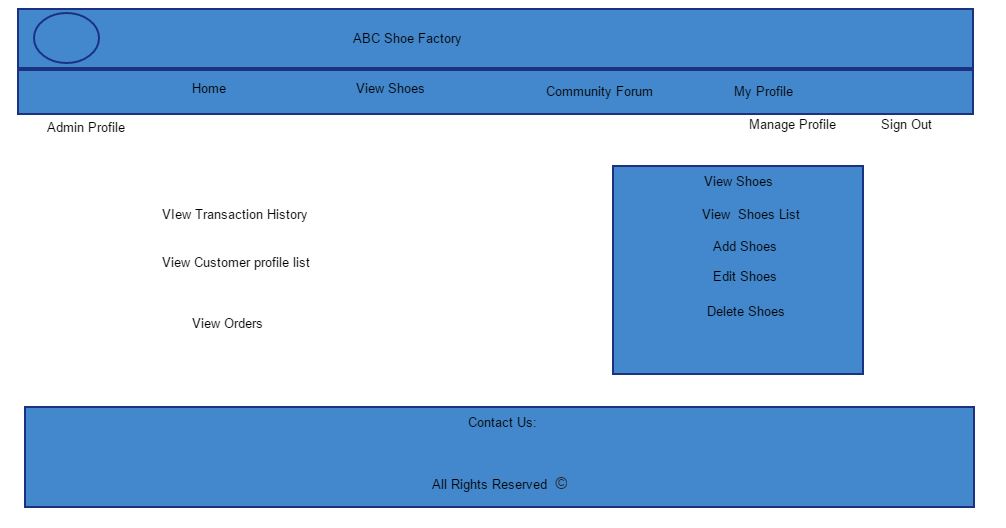
Screenshots: Prototype Home page



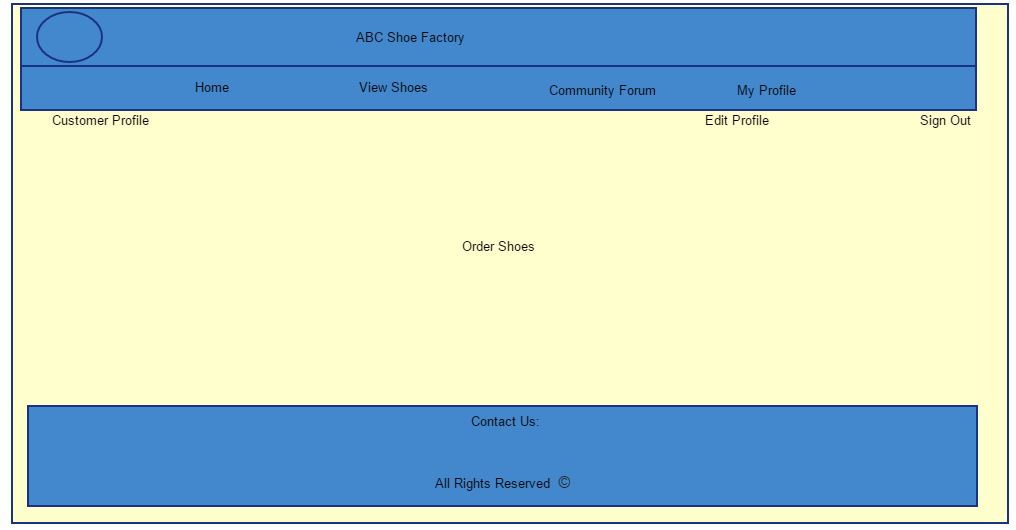
Screenshots: Prototype Login Page



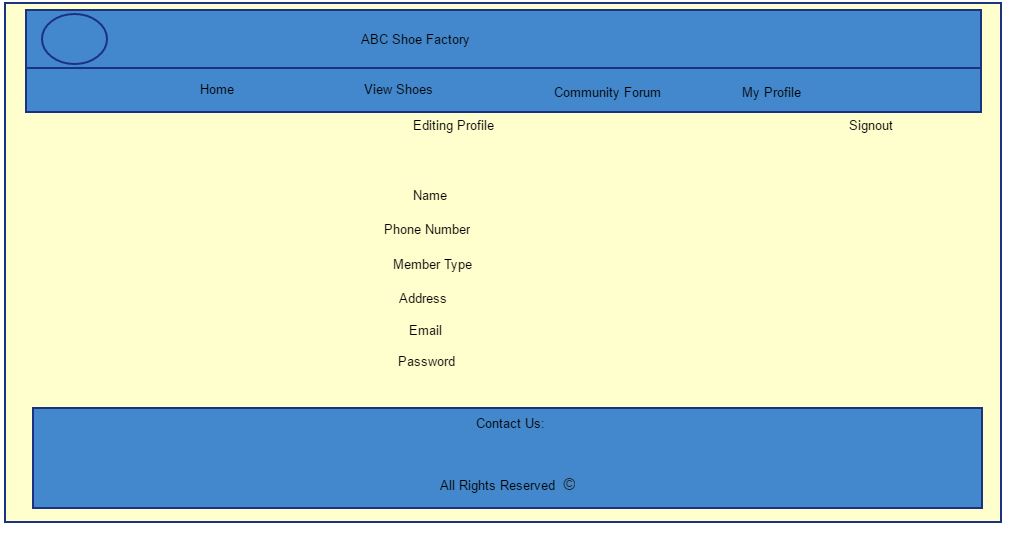
Screenshots: Prototype Register Page



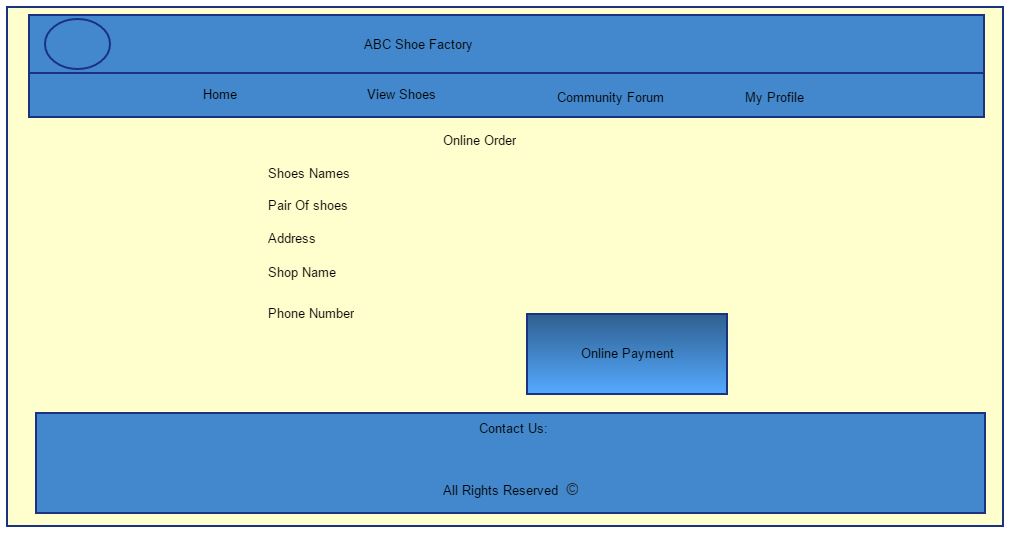
Screenshots: Prototype Admin Dashboard page



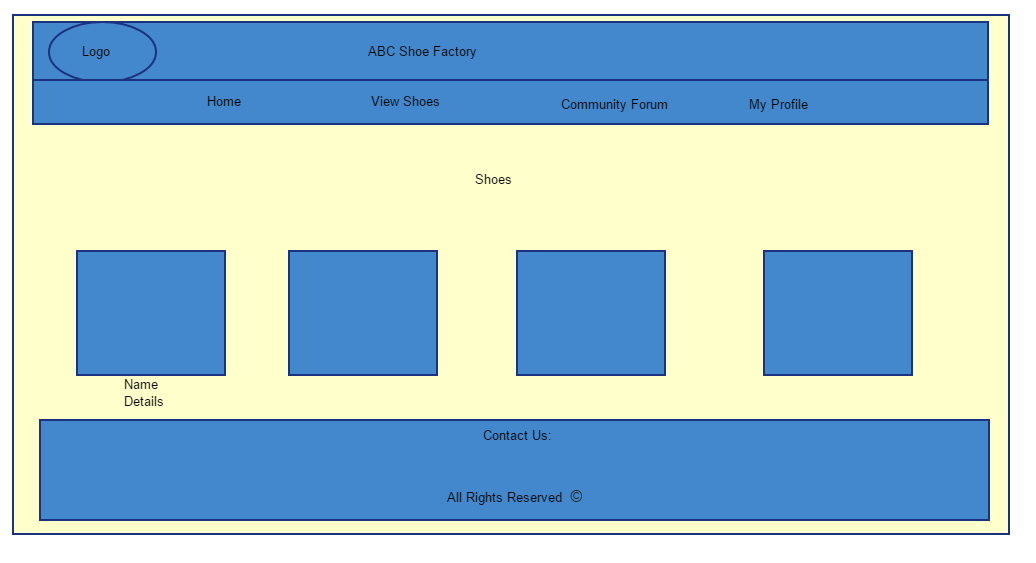
Screenshots: Prototype Customer Dashboard Page



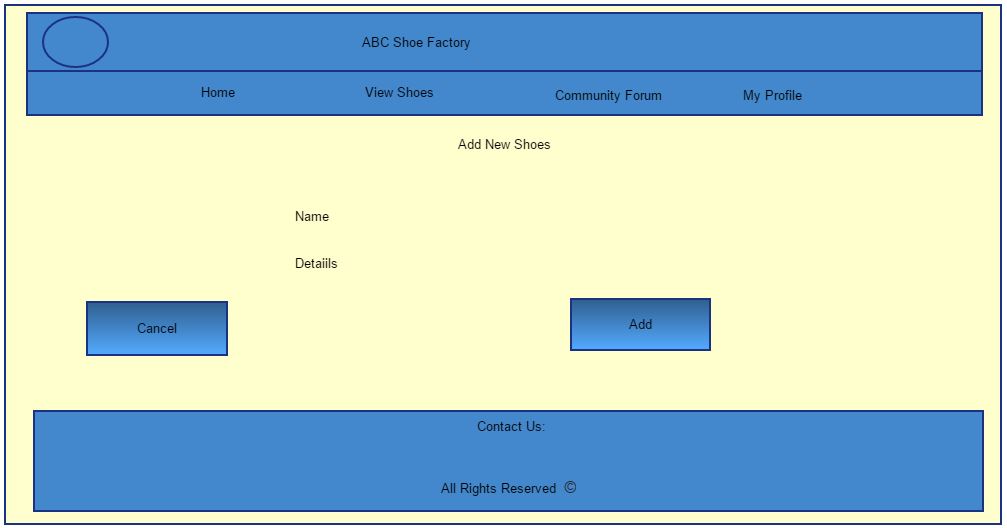
Screenshots: Prototype Manage Profile page



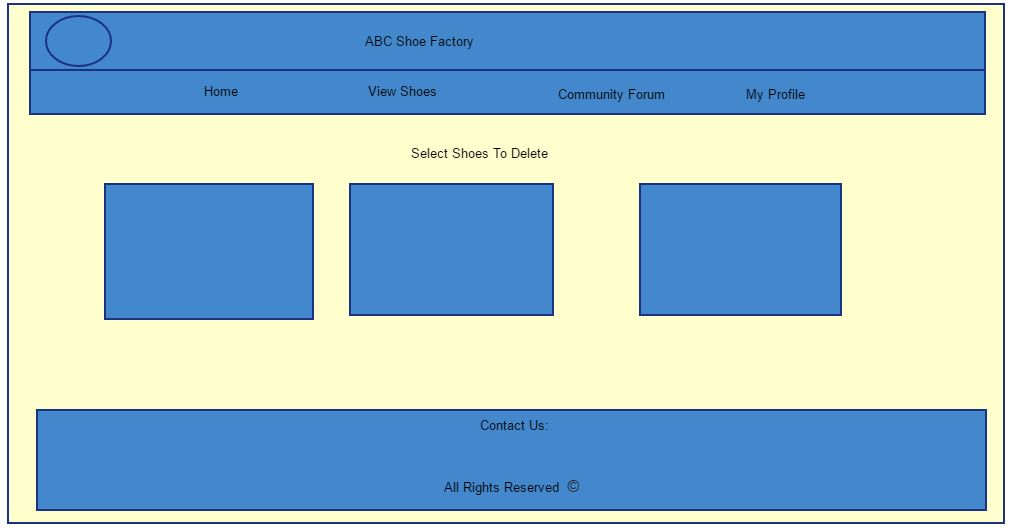
Screenshots: Prototype Order Page



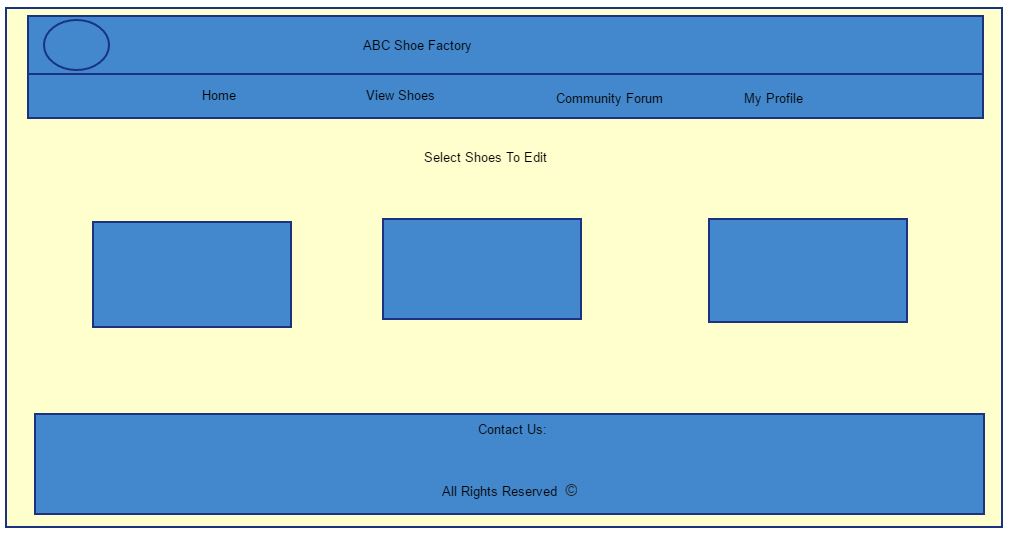
Screenshots: Prototype View Shoes Page



Screenshots: Prototype Add Shoes page



Screenshots: Prototype Select to delete page



Screenshots: Prototype select to edit page