# Chapter 3:

# 3.1 Design:

Project design is early phase of the project where key’s features, structure, criteria for the success and all the major deliverable are planned out. By using different types of tools, technique, models first implemented in the paper or in the design tool and later implemented in the real system. we have design the system for the grocery store. Design is the most important part and its play the vital role in my project.

In my project design phase generate a variety of different output, including final class diagram, behavioral model, data flow diagram, database modelling, prototype of the system, and architectural modelling.

# 3.2 structural model:

Structural modelling of the software displays the group of the system in terms of the works that make up their system or attributes. It may be static models which show the structure of the system or the dynamic models which show the group of system when its executing.

## 3.2.1 class diagram:

Class diagram is static diagram which represent static view of the application. It describes the attributes and the operation of a class and also constraints on the system. It is more describable then the initial phase class diagram. Its shows a collection of classes, interfaces, association and constraints. It referred as the blueprint of the system.

The main purpose of the class diagram is:

* Analysis and design of the static view of an application.
* Describes the responsibilities of a system and act as a structural base for writing the codes.

Notation used in the class diagram:

|  |  |
| --- | --- |
| **Notation used** | **Description** |
| Association | It is used to attach an association class to an association relationship to provide an information about the relation ship |
| Dependency |  |
|  |  |

## 3.2.2 Dataflow diagram

A dataflow diagram maps out the flow of the data for any process or system. It is the traditional visual representation of the data flow in the system. Its show how the information enters and leaves the system, what changes the information and where does the data stores.

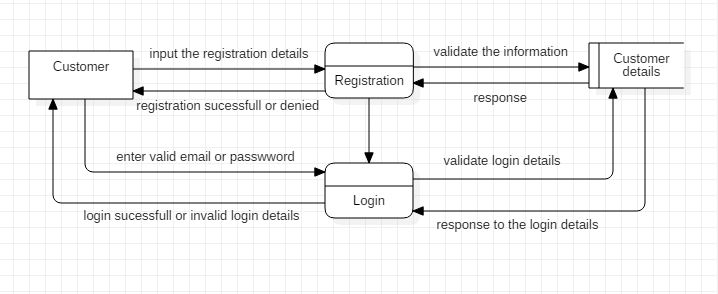
The reason why I create a Data flow diagram in my project are as follow:

* It is easy to understand what the system represents by people because it is well-organized.
* It can be understanding by both technical and non-technical side of the people.

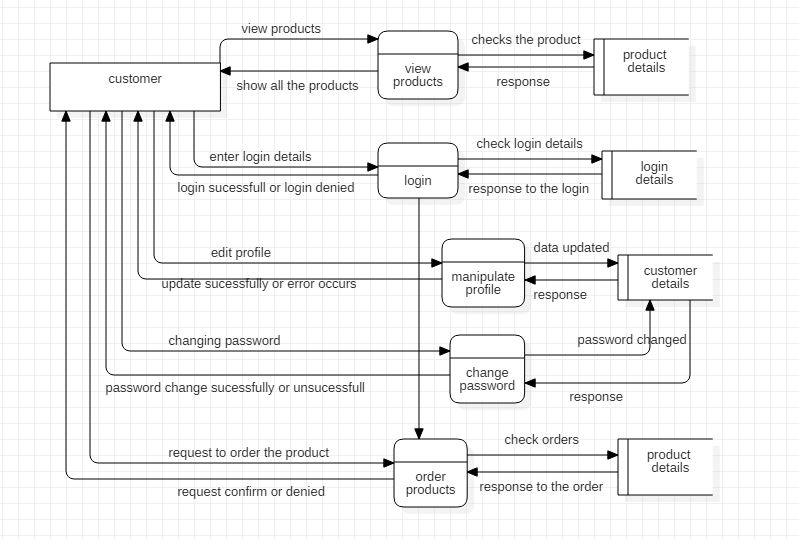
Here are the data flow diagram notation used in my project.

|  |  |
| --- | --- |
| **Notations** | **Descriptions** |
| External Entity | It can represent a human, system or subsystem. It is where the certain data comes from or goes to. |
| Process | A process is a function where the transformation and the manipulation of the data takes place. A process can be decomposed to the finer level of data, for representing how data is being processed within the process. |
| Data Store | A data store represents the storage of the data required or produced by the process. |
| Data Flow | A data flow represents the flow of information with its direction represented by an arrow head that show the end of flow connector. |

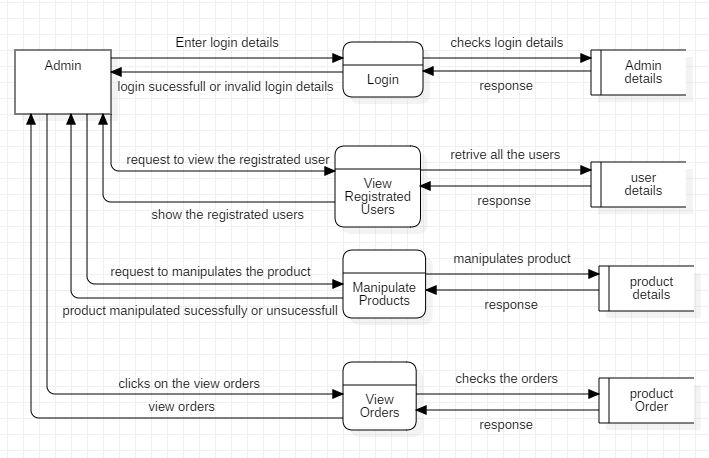
Here are the dataflow diagrams of my system:



Above mentioned **DFD** is defined about the data flow in the system about registration and login system in my project where customer inputs their registration details and gets their username and password to gets login into the system. If the valid details or the unique emails were input by the customer, then registration will success if not, login will have denied or username already exist. Then, in the login system if the correct username or password is filled then login successful otherwise invalid login details.



Above mentioned **DFD** defined about system flow from the customer side where customer can view the product on the website and if they are registered into the website then they are able to login into it with their email and password after login into the system they can edit their profile like, update their name, address, phone number. They are also able to change their password. They are able to order the product if they want after login into the system.



Above **DFD** describe about the functionality of the admin after login, they are able to change the product or add the new one with details information about it. They will able to view the registered user and able to manipulates the products. They can update the product details and add the new product and delete the unwanted product from the site. Check the orders made by the customers.

# Behavioral diagram

Behavioral diagram visualizes, specify, construct, and document the dynamic aspects of a system. There are different categories of the behavioral diagram but I have used activity and sequences diagram for my project.

# Activity diagram:

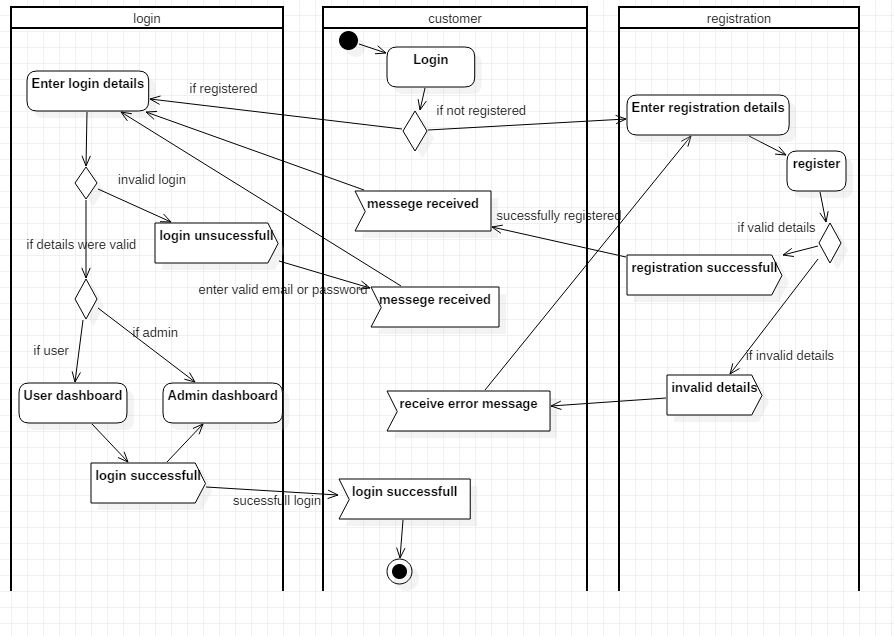
Activity diagram is another important part of UML diagram to describe the dynamic aspect of the system. It is important part in the project, it shows the dynamic representation of the system in my project. It shows the flow of one activity to another activity in the system. It can be described as the operation of the system.

Notation used in activity diagram.

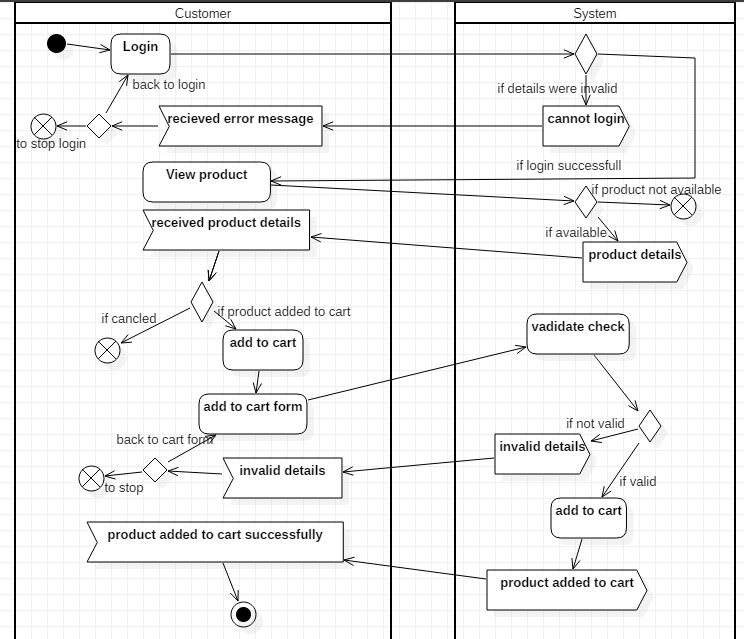
|  |  |
| --- | --- |
| Notation used | Descriptions |
| Start/initial point | A small filled circle followed by the arrow represent the initial action state or start point of any activity diagram. |
| Action or activity state | An activity state represents the non-interruptible action of objects. It is a rectangle with rounded corners. |
| control flow | Action flows, also called edge and paths, explained the transitions from one action state to the another. |
| Decision nodes | A diamond represents a decision with alternate paths. |
| Final state/end point | It represents the final step in activity diagram. It mark as the completion of an all flow of process. |
| Swimlanes | It is a way to group activities by the same actor on an activity diagram or to group activities in a single thread. |
| Send signal | It indicate that the signal is being sent to the receiving or accept signal. |
| Accept signal | It demonstrated the acceptance of an event, after the event is received the flow that comes from this action is completed. |
| Final flow | It represents the end of a specific flow. It does not represent the end of all flow in the activity. |
| Fork | A fork notation control node that splits a flow into the multiple parallel flow. It have one coming edge and multiple outgoing edges. |
| Join | A join notation is a control node that synchronizes multiples flows. This will have multiple incoming edges and one outgoing edge. |

.

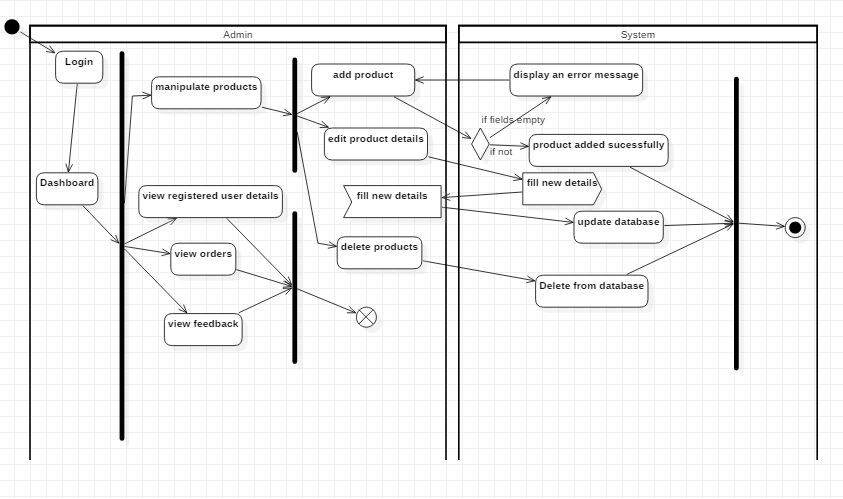
Here is the picture of activity diagram of my project:



In above **activity diagram**, its describe about how the customer or admin able to login or register. Where if they were register or they have their password and email then they are able to login into the website if not then they have to register first with the appropriate details. If they enter valid details they will enter into their dashboard. If not, then error message will come and they will get back into the login or registered page again.



Above **activity diagram** describe about the flow of activity by the customer after login into the website. They are able to view the product and added into the cart if they want otherwise they can cancel it. They will check its validation if customer enter the validate details about the product then it will add to their cart otherwise it will not.



Above **activity diagram** describe about the functionality of the admin where after login, admin can able to enter into the dashboard and manipulates the product such as by adding new product or editing the details of the already exists and also delete the unwanted product from the database. Admin can also able to view the details of the customers and orders made by them. They can also view the feedback given by the customers.

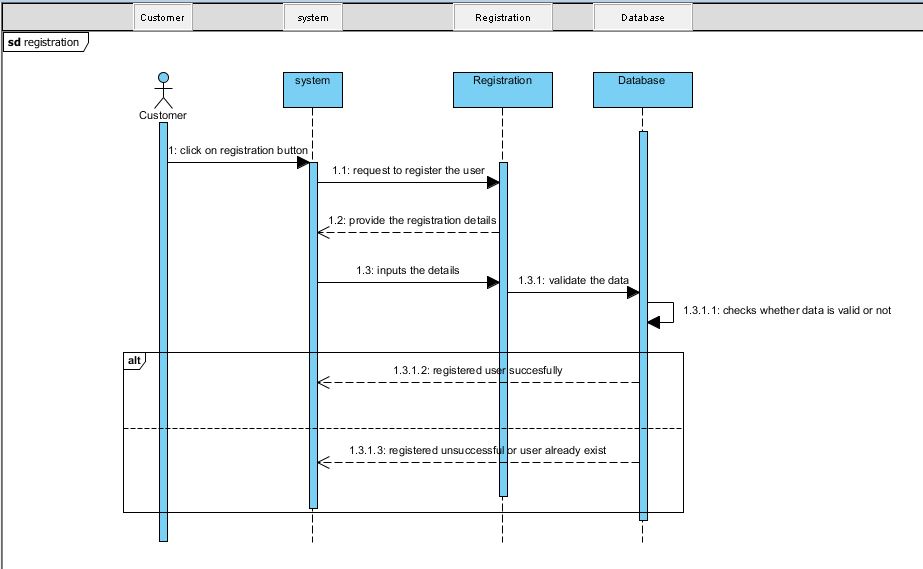
## sequence diagram:

sequence diagram are interaction diagrams that detailed how operation are carried out. it displays and explain the object relationship and interaction between them. It helps to represent the details of use case diagram. The purpose of the sequence diagram is used to show the relations between objects in the sequential order that those interactions occur.

Notation used in sequences diagram:

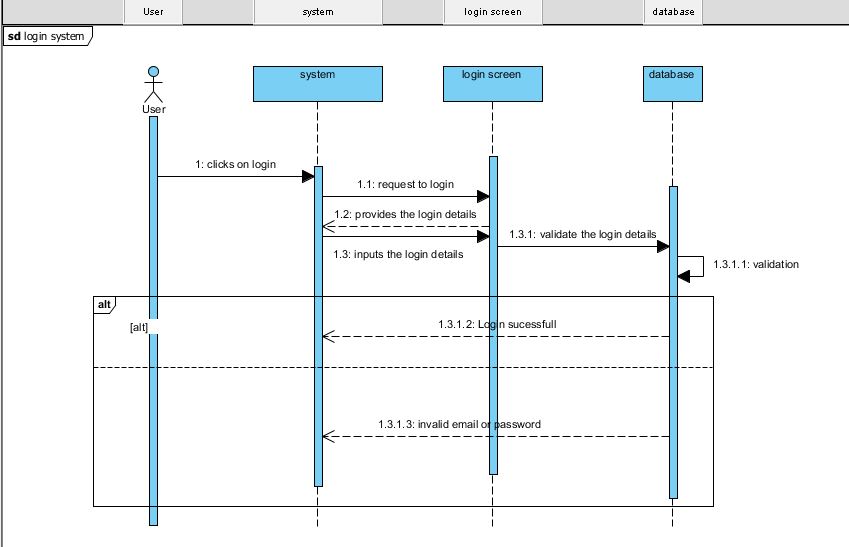
|  |  |
| --- | --- |
| **Notation used** | **Description** |
| Lifeline notation | A sequences diagram is made up of several of these lifeline notation that should be arranged horizontally across the top of the diagram. |
| Activation bars | Activation bars are placed on the lifeline. It used to indicate that an object is active during the interaction between two objects. |
| Asynchronous message | An asynchronous message is used when the message caller does not wait for the receiver to process the message and return before sending other message to other objects with the system. |
| Option combination fragmentation | It is used to indicate the sequences that is only occurs in the certain conditions otherwise it won’t occur. It’s a “if statement” |
| Reply message | Reply message used to indicate that message receiver is done processing the message and is returning control over the messaged caller. |

Here are the pictures of sequences diagram of my project.

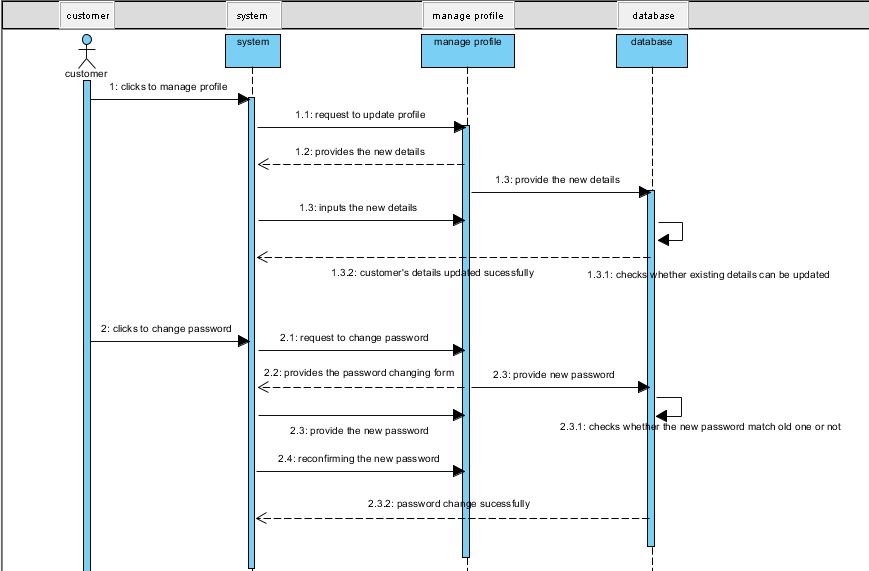


Registration:

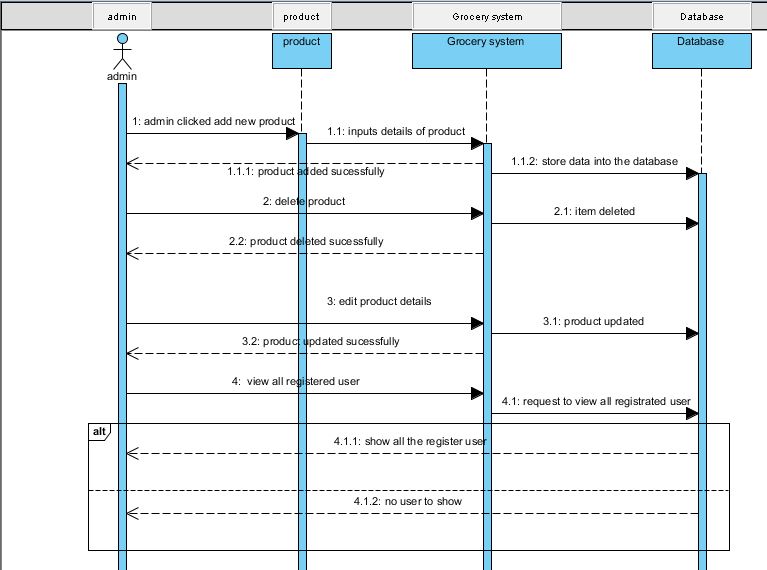
This sequence diagram represents dynamic system of the registration in my project where the customer request to register the user details and inputs the valid details of themselves and if the valid input is given then user gets registered, if not then error message is displayed.



Above mentioned sequences diagram show the sequences of login where registered user inputs their validated email or password. If valid inputs are given to the login system then login will successful, if not then the error message is displayed.



Above mentioned **sequence** show the sequence of the customer managing their profile by updating their details and changing their password. After login into the system customer are allows to edit their profile like, changing their names, address etc. and they are also allowing to change their password with the new one.



Above mentioned sequence diagram show the sequence of the admin added the product and can view the registered user. Its show that the admin can add the products, edits the product detail and delete the product and also admin can view the registered user if there is any registered user in the system.

# 3.4 Database design:

It is a collection of processes that facilitates the designing, developing, implementation and maintenance of the data management system. A good database is important in ensuring reliable data, elimination of data redundancy, efficient execution of queries and high performance application.

## 3.4.1 Data dictionary:

Data dictionary consists of database metadata i.e. data about the database. It records about object in the database. It consists of table name, relationships, keys and the explanation the function elements. It is important as it contain information such as details about all database tables and their representations, physical information about the tables such as where they stored and how.

Why I used data dictionary? Here are the reasons:

* It is use to define format, relationships, meaning of my system.
* It used to manage the all the table in my project like, user registration, product, order etc.

## 3.4.2 Entity relationship diagram

An entity relationship diagram shows relationships entity sets stored in the database. Its defines the entities, their attributes and showing the relationship between them. It’s often used debug or design relational database. The elements of entity relationship diagram are:

* Entities
* Relationships and
* Attributes

Steps to create the entity relationship diagram are:

* First of all, we have identified or define the entities.
* Then, we have to determine the interaction between the entities.
* And after that analyzing the nature of interactions/ determining the cardinality of the relationships.
* At last create, Entity relationship diagram.

# Architectural modelling

# 3.6 prototyping

A prototype is a basic working sample, model, mock-up or just an imitation of the actual product based on which the others forms are developed. Motive behind prototyping is to validate the design of the actual product. It may also have called as it is first step of transforming the virtual or conceptualized design into the real physical form.

Here is some prototype of my project:

