# Design

## Introduction

Design is the way to solve the problem. This process can be replicated and can be used for various problem for small scale products to complex service. It is more creative than analysis. The way to solve the problem is to initial requirement where we gather data on user requirement, analyze requirement data, high level design and refine and document the design. This method we can obtain the answer and which also validate the design against the requirement.

There are two types of design conceptual design and technical design where conceptual mean customer and technical mean system builders.

## Structural design

Structural modelling captures the static features of the system. They consists of

* Class diagram
* Object diagram
* Data flow diagram

The purpose of structural model is to reduce the “semantic gap” between the real word and the world of the system. And it creates a vocabulary for analysts and users. And it represents things, ideas and concepts of importance in the application domain.

### Class diagram

Class is the description of a set of objects having a similar attributes operation, relationships and behaviors. Class diagram is the mostly used structural diagram. And essential elements of a UML class diagram is

* Class
* Attributes
* Operations
* Relationship
* Constraint rules and notes.

Relationships is class diagram are:

* **Association:**
* it is a structural relationship, specifying the object of one class are connected to object of second class. For example: an employee work for a company
* directed association, reflection association multiplicity association.

* Generalization
* It indicates the object of the specialized class (sub class) one substitutable for object of the generalized class.
* Is a kind of relationship
* Dependency
* The life time of the part is dependent upon the whole.
* Realization
* It indicates that one class implements a behaviors specified by another class
* An interface can be realized by many classes.

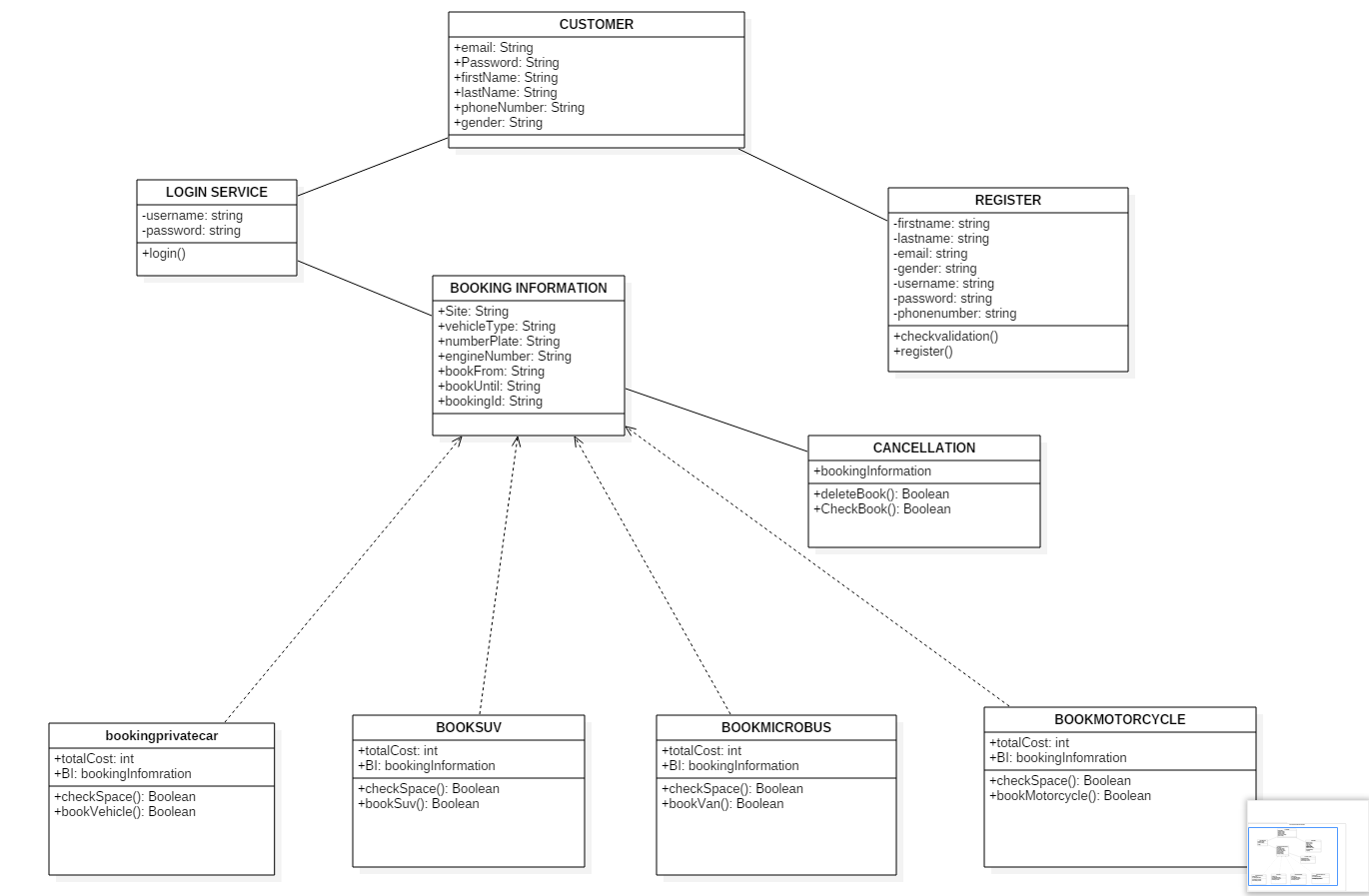
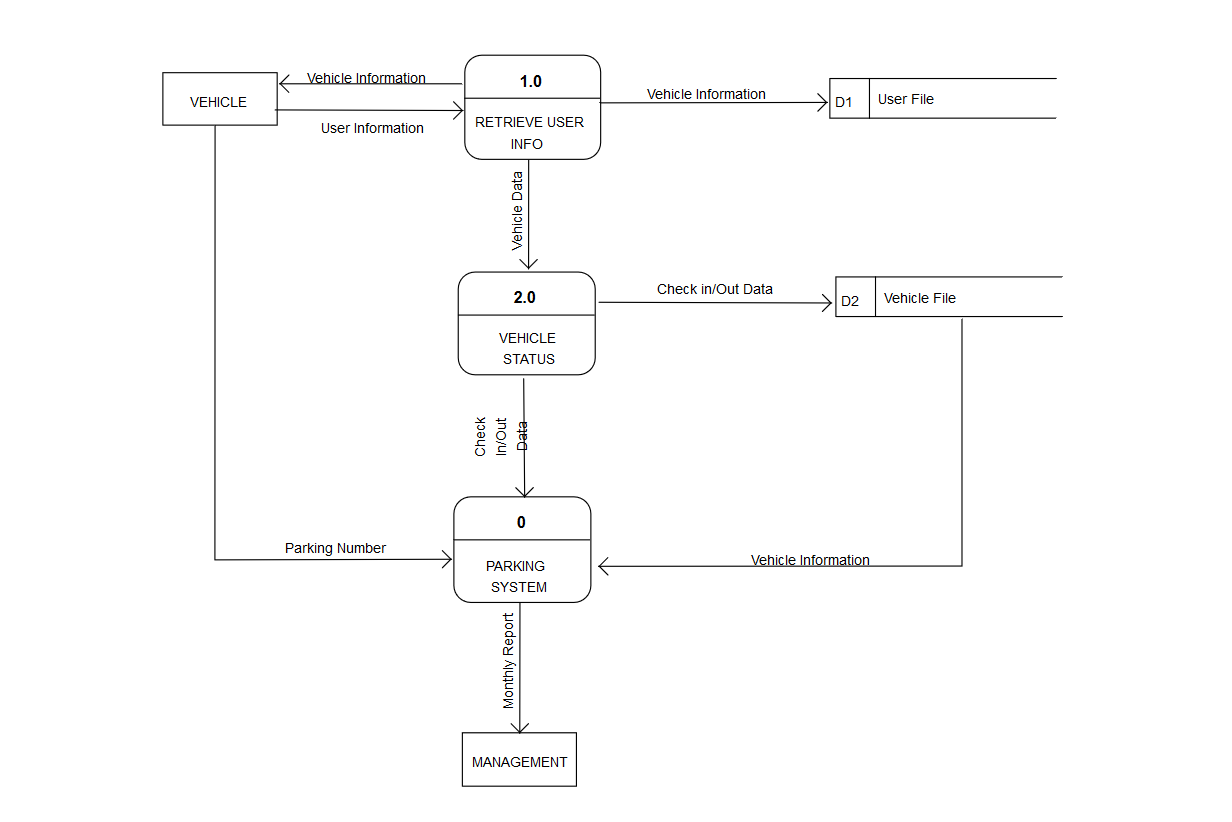


Figure 1:final class diagram

### Dataflow diagram

* The data flow diagram is a hierarchical graphical model of a system that shows a different processing activities or function that the system performs and the data interchange among these functions.
* Symbols used in construction a dataflow diagram is
* External entity: the sources from which information flow into the system and recipients of information leaving the system.
* Process: the activities carried out by a system which use and transform information
* Dataflow: the data flow are notated as named arrow.
* Data store: where information is stored within the system
* A data flow diagram model of a system graphically represent the transformation of the data input to the system to the final result through a hierarchy of levels.

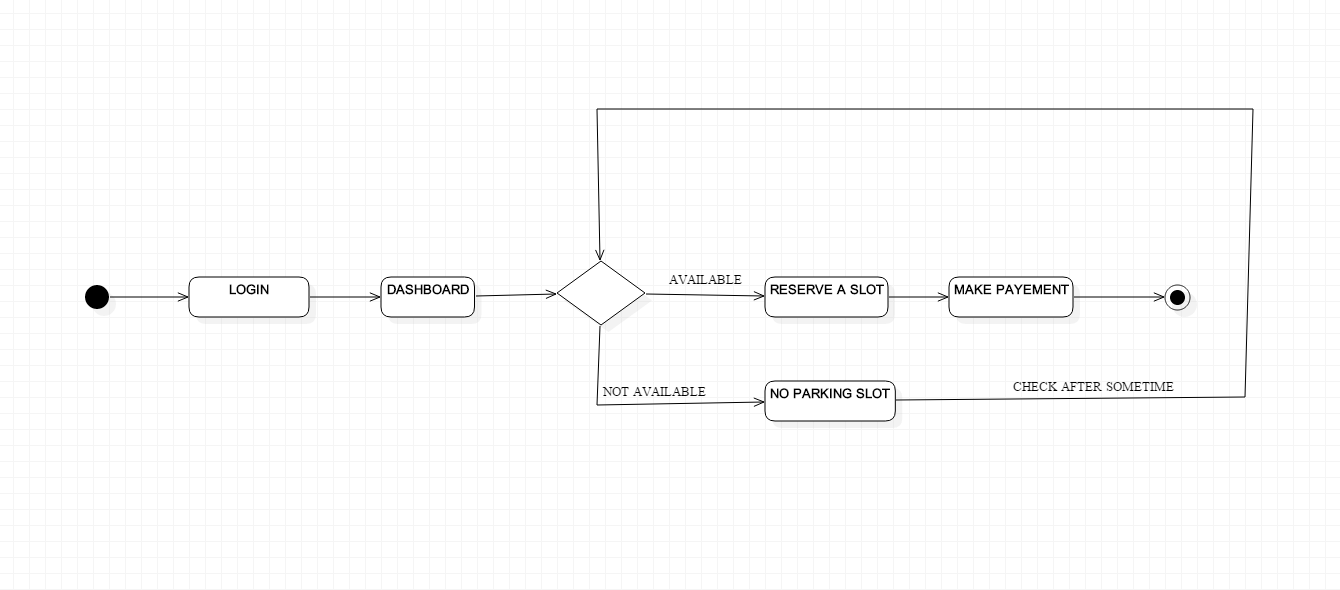


## Behavioral design

Behavioral modeling consists consist of activity diagram and sequence diagram.

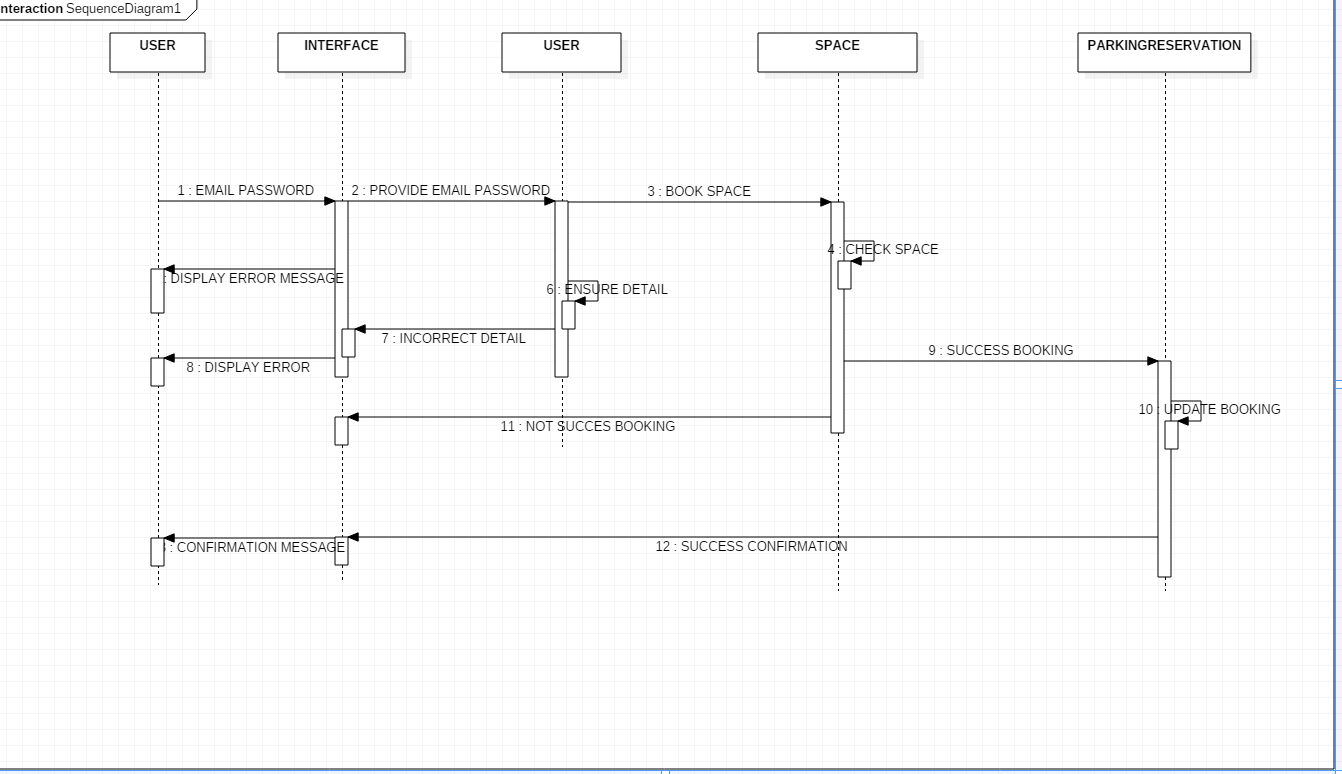
### Activity

* It is a special kind of a state chart diagram that shows the flow from activity to activity within the system
* Activity diagram address the dynamic view of the system
* They are especially important in modeling the function of a system and emphasize the flow of control among objects.



### Sequence

A sequence diagram is an interaction diagram that emphasizes the time-ordering the message.



## Database design

There are two approaches of design that is bottom up and top down where bottom up is design by synthesis where we collect all the information. For example in online parking booking system we collect all the entities and from these entities we show the relationship where as in top down system we break the entities into small entities.

### Data dictionary

* A structure place to keep detail of the contains of data flow process and data store.
* It is a structured repository of data about data
* It is a set of definition of all data flow diagram elements.

The items to be defined in data dictionary is:

* Data elements: smallest unit of data that provides for no further decomposition.
* Data structure: it is a group of element handled as a unit
* Data flows and data stores: data flow are motion where as data stores mean data structure are stored

1. **CUSTOMER**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Datatype | Constraints | Nullable |
| Id | Int | Primary key unique | No |
| vehicleNumber | Varchar(10) |  | Yes |
| engineNumber | Varchar(20) |  | Yes |
| phoneNumber | Varchar(20) |  | Yes |
| Address | Varchar |  | Yes |

1. **REGULAR PASS**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Datatype | Constraints | Nullable |
| id | Int | Primary key unique | No |
| Customer\_id | int | Forign key | No |
| Cost | Float(10) | Null | Yes |

1. **Parking slot**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Datatype | Constraints | Nullable |
| id | Int(10) | Primary key unique | No |
| Booking\_date | date | Null | Yes |
| customerId | Int | Foreign key | No |

1. **Parking space**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Datatype | Constraints | Nullable |
| Id | Int | Primary key unique | No |
| Entry\_time | date | Null | Yes |
| Exit\_time | date | Null | Yes |
| Total\_cost | float(10) | Null | Yes |
| column | int | Null | Yes |
| Parkin slotid | int | null | yes |

1. **Parking lot**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Datatype | Constraints | Nullable |
| Id | Int | Primary key unique | No |
| Number\_of\_block | double(10) |  | Yes |
| Address | Varchar(20) | Null | Yes |
| Slot\_available | Varchar(20) | Null | Yes |
| Parking slotid | integer | Null | Yes |

1. **Block**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Datatype | Constraints | Nullable |
| Id | Int | Primary key unique | No |
| Parking slotid | interger(10) |  | Yes |
| Block\_code | varchar | null | yes |

1. **Space**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Datatype | Constraints | Nullable |
| Blockid | Int | foreign key | No |
| id | Varchar(10) | Primary key | Yes |
| Number\_of\_space | Varchar(20) | null | Yes |

### ERD

**Entity relationship modeling:**

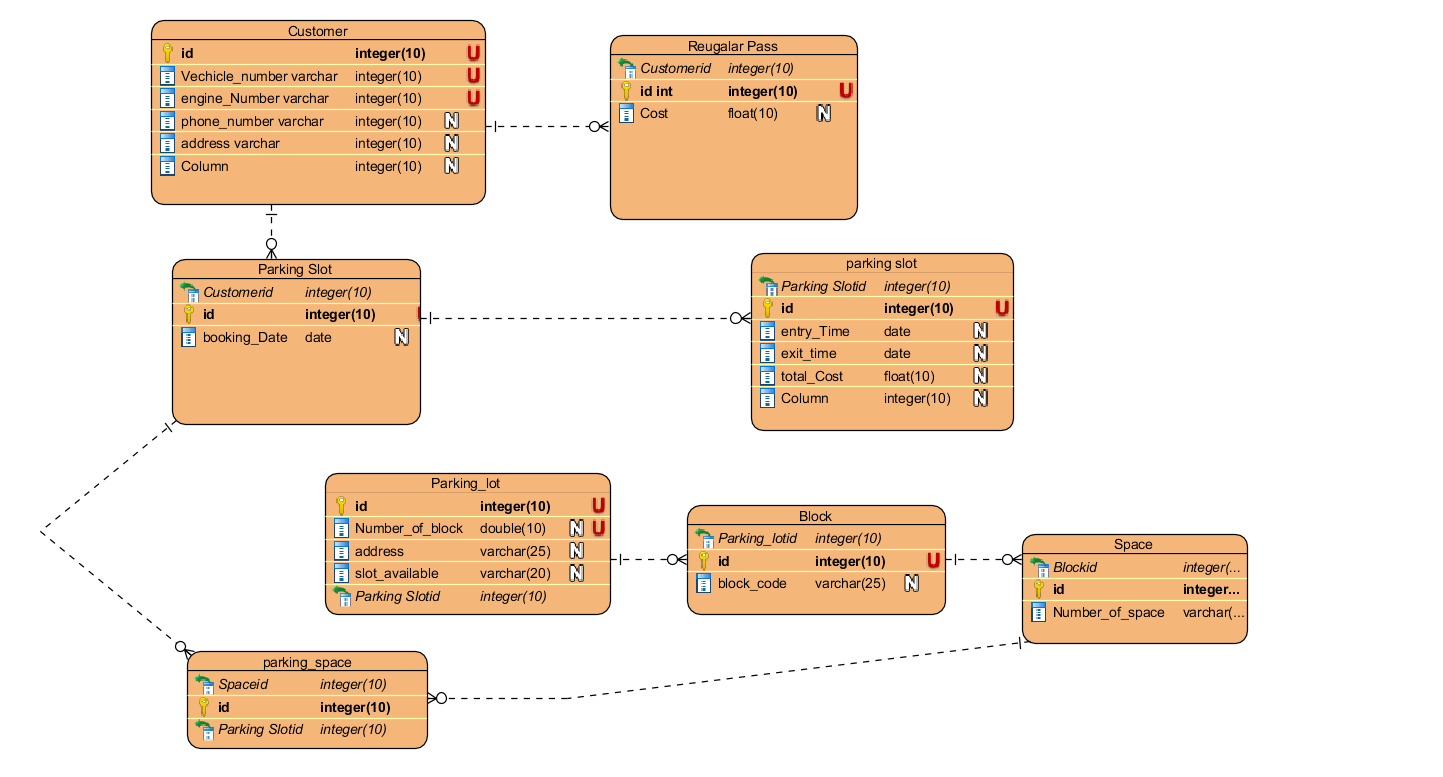
* It is a systematic way of describing and defining a business process
* Entities may have various properties attributes that characterize them.
* Diagram created to represent these entities, attributes and relationship graphically are called entity relationship diagram.

The key components in database design are as explained below.

* Entity: entity may be a person, place, things or it may be an event.
* Attributes: attributes may be the process which describes entity
* Relationship: its shows the relationship between the entities.

ER diagram is modelled in three type of relationship basically as follows:

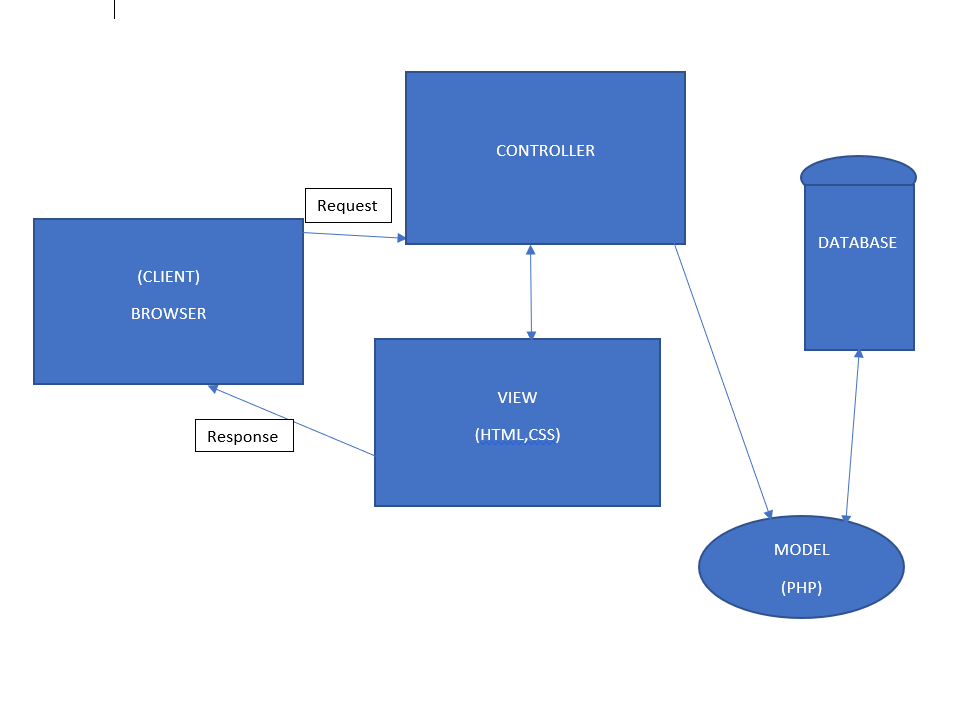
* One to one relationship:
* One to many relationships:
* Many to many relationships:



## Architectural design

* System design architectural design represent the structure of data that are required to build a computer based system.
* Architectural design, representing the system in context
* System enginner must be model the context
* Context diagram model in which the system interacts with external entities.

And I have used model view controller design for this system.



## UI design

* User interface is a front end application view to which user interacts in order to use the software.
* User can manipulate and control software and hardware by means of user interface.
* For example: if my online booking system has bad interface customer cannot use it so it should be simple. We can user interface in computer, mobile, cars ,airplane, music players etc.
* It provides platform for human computer interface.
* The software becomes most popular if its interface is attractive, simple to use, reposive in short time and clear to understand.
* User interface is broadly divided into 2 category called command line interface and graphical user interface.

### Digital Prototype

Interface that is digitally display is digital prototype and I have used Balsamiq for digital prototype

