# Design

Design is the period of planning and also called framework which is done after gathering the user requirement and analyzing the system to make a project more reliable. Design is a development that is in the form of prototype. Firstly, we did design on a paper or design tool and then implemented in the system by using various tools, techniques and models. Design is important to make a user more easily understood and convenient.

To implement the various user requirement in the software, I have used various design models such as structural and behavioral models. Structural models is used to show the structure of a system and behavioral model is used to show the workflow of a system being developed. Rather than this, I have used database design which includes data dictionary and ER diagram. To give overview as how the system backend and frontend would be like I have done user interface design.

# Structural design

Structural design shows the static features of a system. It represent the framework for a system where all the components are exists.

# Class diagram

Class diagram is a static diagram which is used to show the system static view. It is used mainly for constructing executable code of software rather than describing, visualizing different aspects of a system. Class is a blueprint of an object which shows the structure of a system.

**Why we used class diagram?**

1. System responsibilities can be described.
2. It depicts the system framework by representing the system classes, their attributes, operations and relationship among the object.

**Notations**

* useable notation of class diagram are:

Name of class



Attributes

Operations

1. Inheritance
2. Association

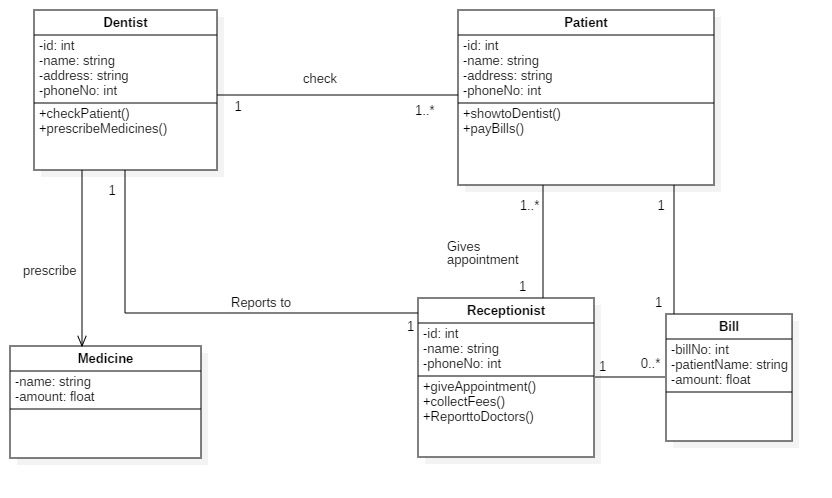


Figure : class diagram

Class diagram represents the structure of a system. For controlling a whole system, model is used to manipulate the data. To access the data, view is used through controller. This above class diagram is followed by MVC design pattern which includes model, view and controller.

# Dataflow diagram

Dataflow diagram is a way of representing a flow of a process or a system. It delivers information about each entity outputs and inputs and the process itself. There are only one rule in DFD i.e. all the flow must be start with and end at a processing step.

DFD is easy to understand the flow of data through the system.

* **Why we used data flow diagram?**

1. There is a good interaction between user and system designer.
2. Flow of system will be logical for the information.

**Notations**

The following are the notations used in data flow diagram.

1. External entities
2. Process

1. Data store
2. Data flow

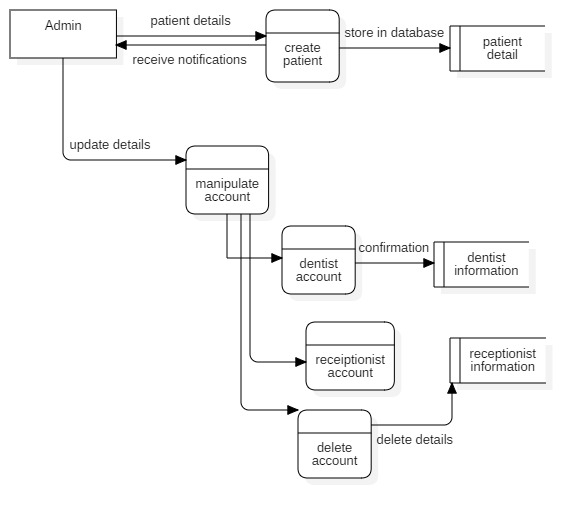


Figure : Admin DFD

Admin store patient details in the database. Admin can update all the manipulate account that is involved in dental clinic management system. i.e. Admin can update dentist account and store all the information in dentist information database. Admin can update receptionist account and store all the information in receptionist information database. Admin can delete all the account.

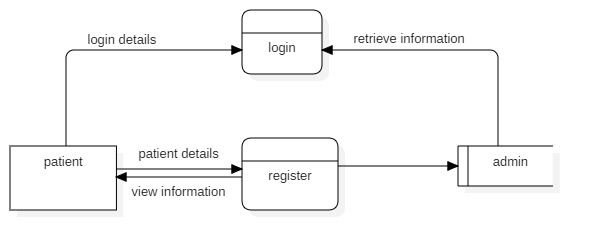


Figure : DFD (registration, login)

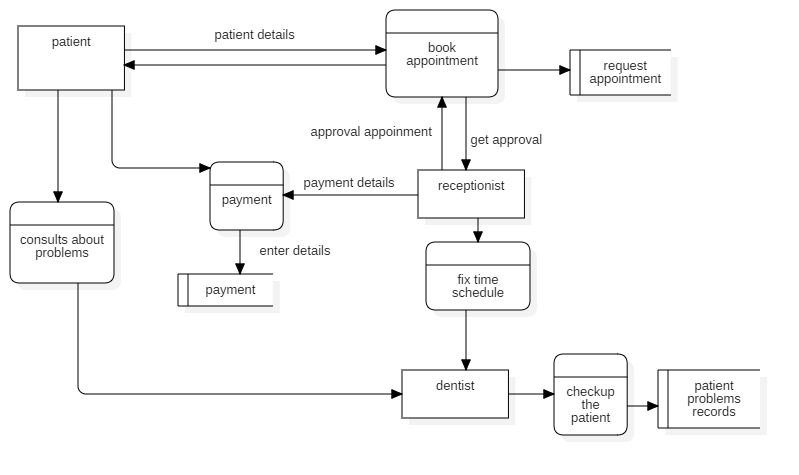


Figure : process of DFD

After, patient login to the system. Then, patient book the appointment by the help of receptionist where patient has to include all the details and this record will be store in database. Then, receptionist gives approval to the patient. On the other hand, receptionist fix a time schedule. Then, patient consults about their problems to the dentist. Dentist check the patient and all the patient problems record will be store in database. After then, receptionist make a payment details. At the end, patient pay the bill.

# Behavioral design

Behavioral design is a sub-category of design, which is concerned with how design can shape, or be used to influence human behaviors. It shows the interaction within the system.

# Activity diagram

It is a figure in UML which explains the whole system about dynamic aspect. Activity diagram is basically a flow chart to represent the flow from one activity to another activity. It also describe as an operation of a system. Activity diagram also called work flow diagram. It describes all the logic of the operation which are exhibits on class diagram.

* **Why we used activity diagram?**

1. Dynamic aspect of a system is taken.
2. Parallel, branched and concurrent flow of the system is explained by this diagram.

* **Notation**

The following are the most useful notation:

1. Start
2. Action
3. Basic flow
4. Decisions

1. End

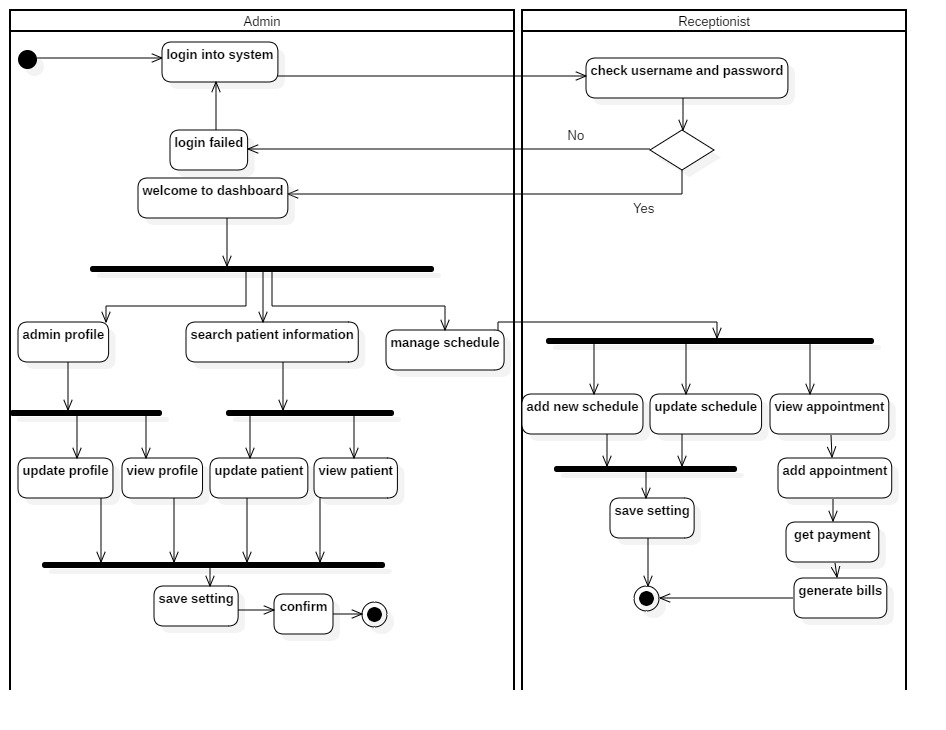


Figure 5: Activity diagram 1

Firstly, admin login to the receptionist system. Then, check the username and password. If username is valid, then directly goes to dashboard. If invalid then again return to login page. After successfully login, admin can see the profile, search patient information and seeing the manage scheduling. Admin can update, view the profile of the patient. At the end admin save the setting and click on confirm then the admin profile is terminated. On the other hand, receptionist manage all the schedule where receptionist can add, update schedule and also can view an appointment. After that, receptionist add the appointment and after completing all the process receptionist get payment after generating bills. Finally, receptionist system has been terminated.

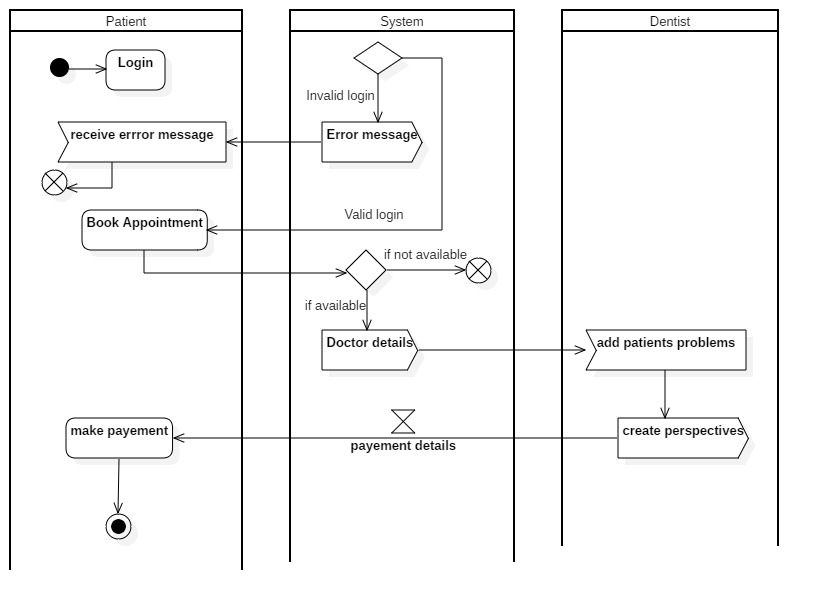


Figure : Activity diagram 2

Here, Patient login to the system. Then, system check the username and password. If valid then patient can book the appointment. If invalid, then error message is seen. After booking the appointment, patient check the doctor details in the system. Then, dentist add the patient problem and create perspectives. After that, system gives the acceptance time event. Then, patient pay the bill and patient profile is terminated.

# Sequence diagram

It is the popular dynamic modelling solution in UML. It focus on lifelines or the process and object that lies concurrently and the message exchange between them to perform or function before lifeline ends. It describes object relationship and interaction between themselves.

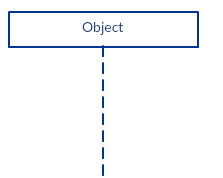
* **Why we used sequence diagram?**

1. It consists of no of lifelines that produces message which make user easy to understand.
2. It shows dependencies of objects and methods but not usually implemented directly.

**Notation**

The following are the notation used in sequence diagram:

1. **Lifeline**



1. **Message**

1. **Reply message**



1. **Self-message**



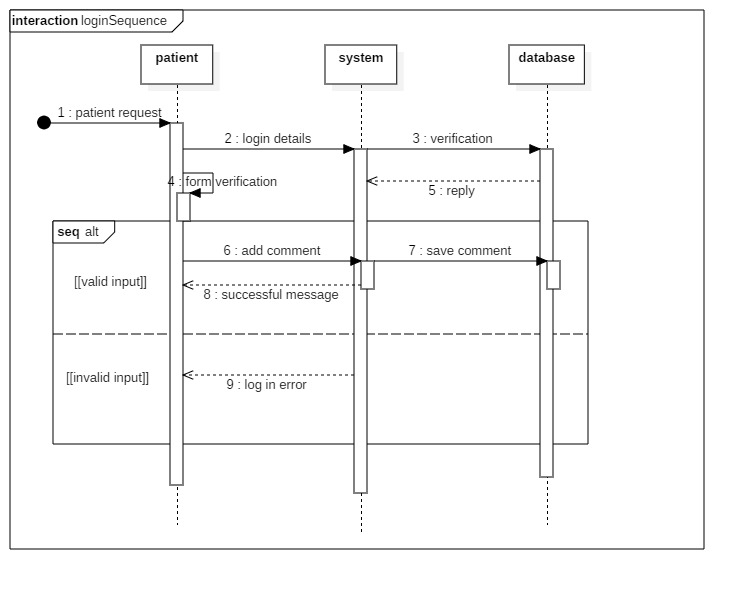


Figure : Login sequence diagram

Here, patient login to the system. For verification, system sends to database and system itself form verification. The system valid information is directly stored in a database. If invalid, then the message is directly goes to the login page.

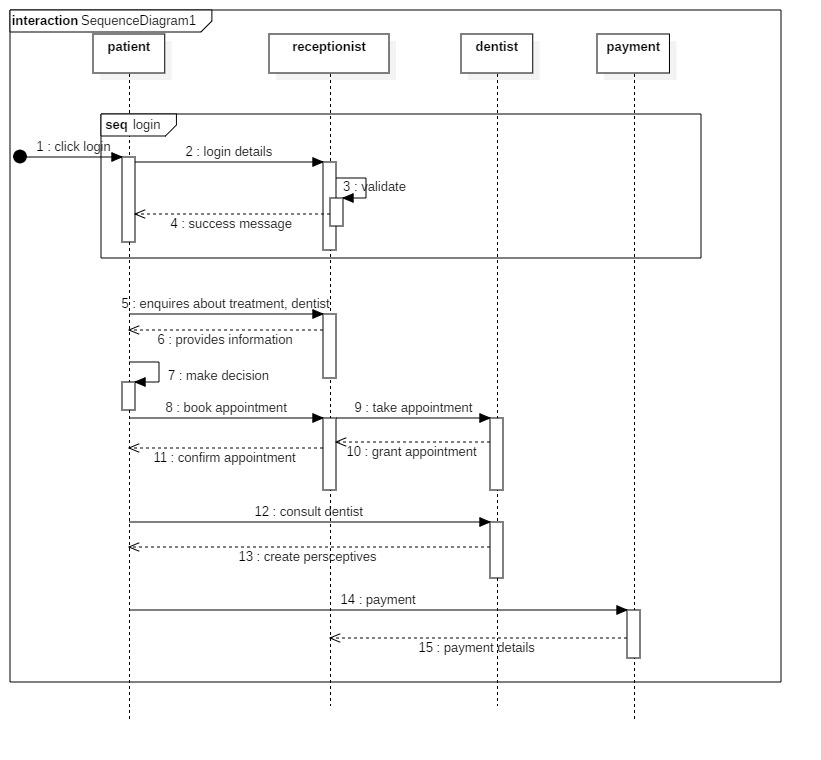


Figure 8: Sequence diagram of flow system

After login, patient send enquires message to receptionist. Then, receptionist provides information in the reply. Patient make decision and book the appointment. After that, receptionist take appointment to doctor and doctor grant appointment to receptionist. For confirmation, receptionist reply message to patients. Then, patient consults with dentist and dentist create perceptive to patient. At the end, patient terminates after paying the bill.

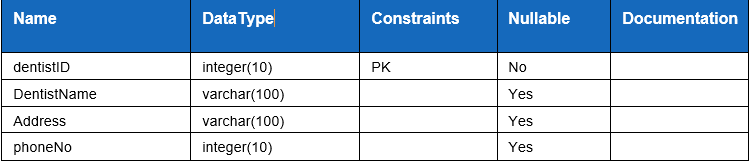
# Database design

Database design is the process of organizing a data in the basis of database model. It is method of managing a data in a database system also used to remove data redundancy.

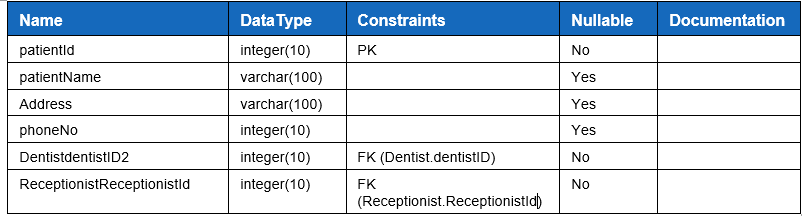
# Data dictionary

Data dictionary is a way in which database keeps information about its own structure. It is used to control access to and manipulation of the database. Metadata is stored in data dictionary. It describe the format and structure of database.

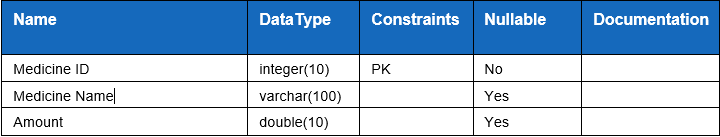
* **Dentist Table**



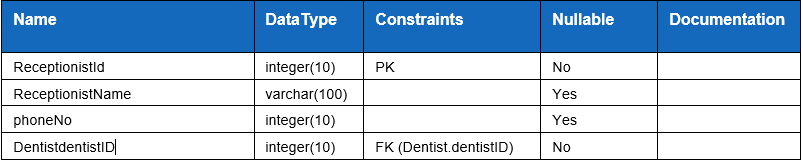
* **Patient Table**



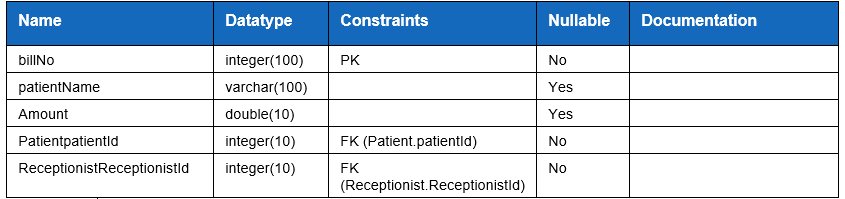
* **Medicine Table**



* **Receptionist Table**



* **Bill Table**



# Entity Relationship Diagram

ER diagram characterize the relationship between entities that is stored in a database. It is a logical structure of a database. Also, there will be defining the entities, their attributes in ER diagram. CROWS FEET, UML or CHEN notation are the ways in which ER diagram can be represented.

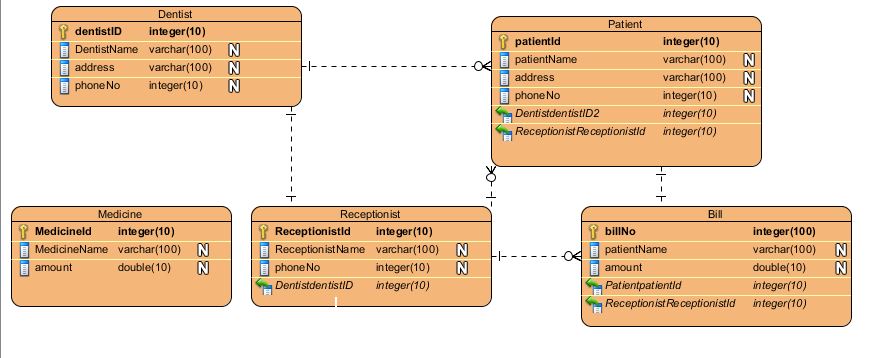


Figure : ER diagram

In this figure, there are five entities. This diagram is based upon data dictionary. Entity dentist has one-to-many relationship with patient. Likewise, dentist has one-to-one relationship with receptionist. Whereas, receptionist has one-to-many relationship with bill entity. On the other hand, patient has many-to-one relationship with receptionist. Again, patient has one-to-one relationship with bill entity.

# 5. UI design

