1. Project problem statement feasibility assessment using NP-Hard and NP-Complete

Introduction to NP Hard & NP Complete Problems

NP class are divided into two groups:

1. **P class Problems** - Problems that can be solved within a polynomial time.
2. **NP class Problems**- Problems that can be solved within a non-polynomial time.
    - **NP Complete problem** – A problem that is NP complete can be solved in a polynomial time iff all other NP complete problems can also be solved in polynomial time.
    - **NP Hard problems** – If an NP Hard problem can be solved in polynomial time then all NP complete problems can be solved in polynomial time.

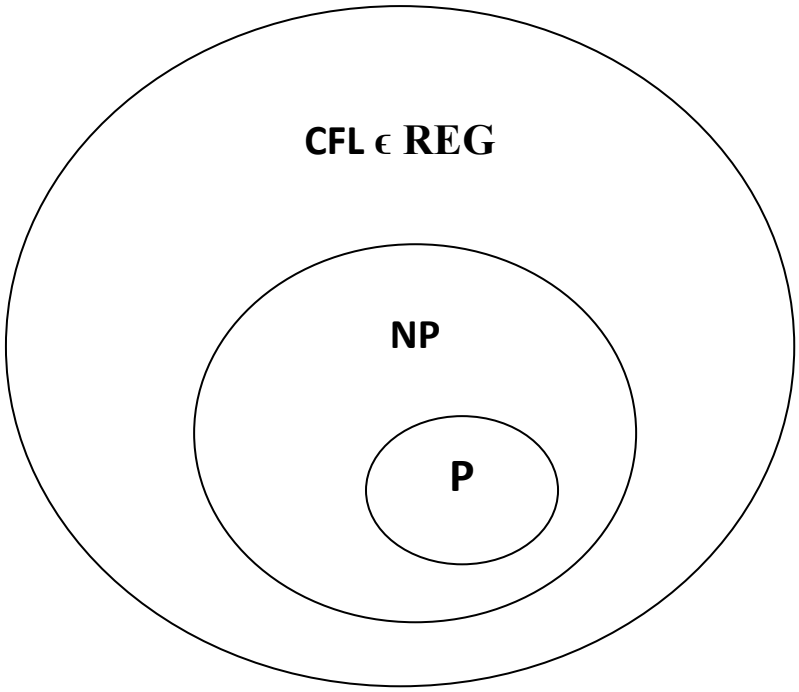


Figure1

- P is a set of problems that can be solved in polynomial time using deterministic method.
- Context free language (CFL) is a set of problems that cannot be solved in polynomial time using non-deterministic method.
- NP is a set of problems that can be solved in polynomial time using non-deterministic method.

Project Statement- E-Health Care Facilities

Module1: Sign-up/Sign-in

In this module the user has to first sign-up if he is not an existing user.

This module has limited steps and does not go into an infinite state which makes it feasible and thus NP-complete.

Module2: Schedules & Reminders

In this module the user can check schedules of the doctor to take or cancel appointments and also the user also gets reminders for his appointments and his medicines.

This module has limited steps and does not go into an infinite state which makes it feasible and thus NP-complete.

Module3: Disease Prediction

In this module the user has to enter his symptoms with reference to which the disease is predicted by mapping these symptoms to the dataset.

This module has limited steps and does not go into an infinite state which makes it feasible and thus NP-complete.

Conclusion:

As are project is composed of the three modules mentioned above we conclude that our project is of type NP-complete.

2. Functional relations, dependencies & Mathematical models

Mathematical Model

Set Theory

Z = {U, D, S}

Where,

U = Set of Users = {P, H}

P = {patient 1, patient 2 . . .}

H = {hospital 1, hospital 2 . . .}

U = P ∪ H;

D = Set of Database = {Dt }

Dt = U ∪ S

S = Set of Services = {Access Records, Take Appointment, Reminders, Predict Diseases}

Functional Decomposition of System

Different Functions

- f(g) = function of sign up
- f(h) = function of appointment scheduling
- f(i) = function of appointment confirmation
- f(j) = function of accessing medical records
- f(k)= function of reminders
- f(l) = function of disease prediction
- f(m) = function of uploading records

Functional Dependency

f(g) = Dt → U

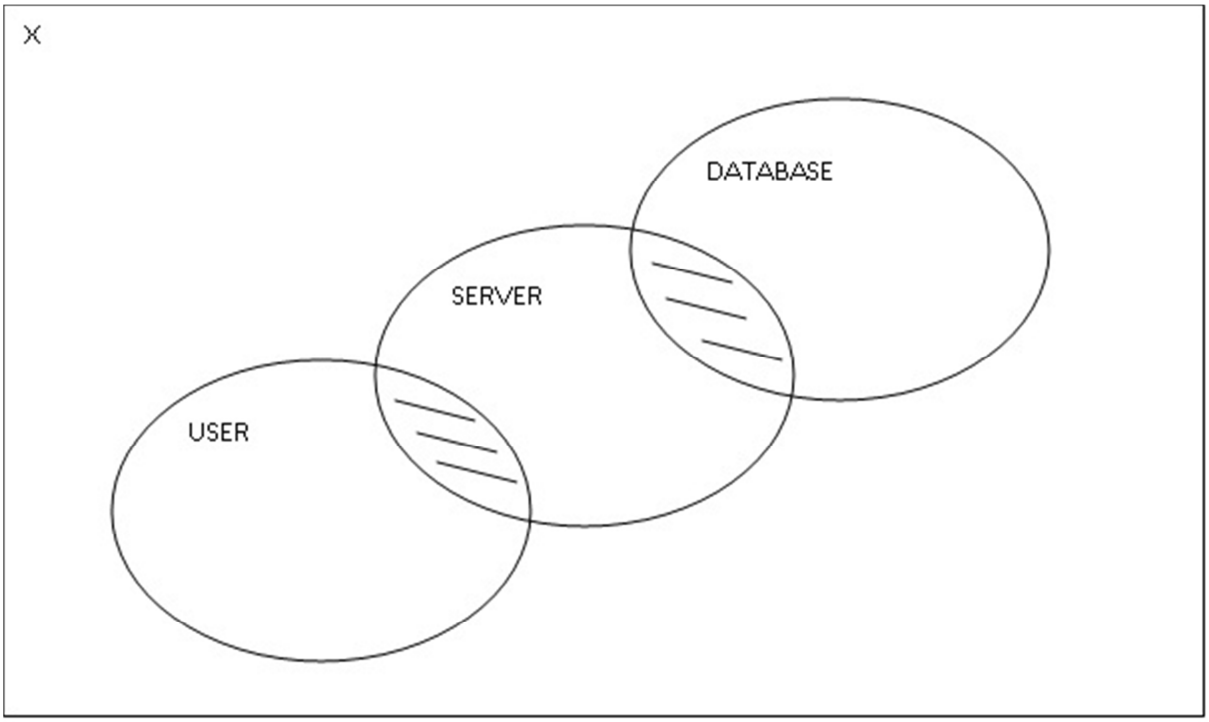
A: f(i)→f(h)

B: f(j)→f(m)

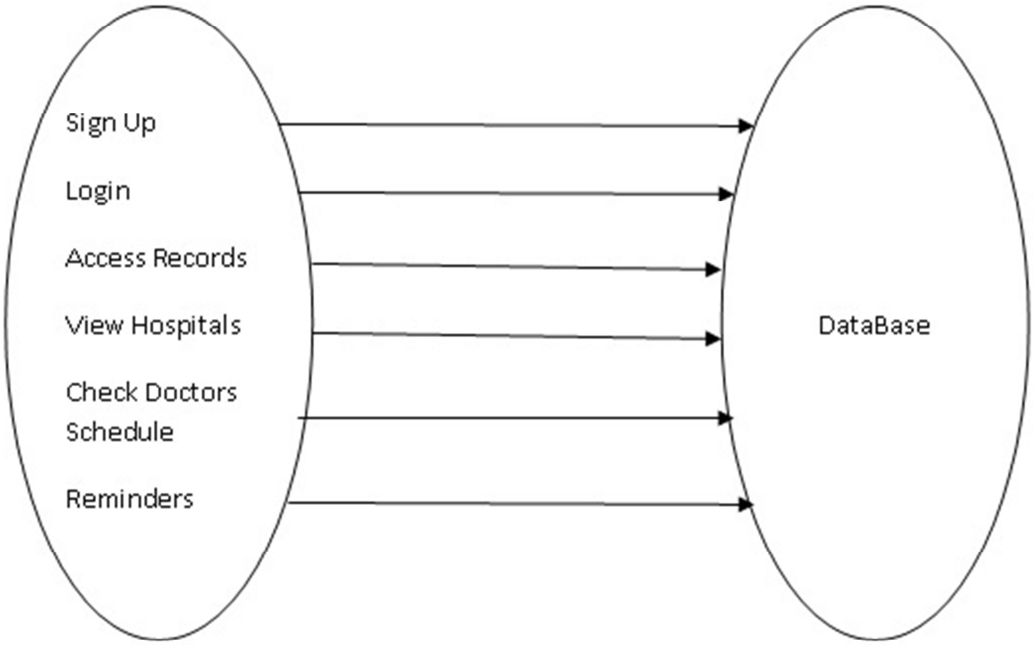
C: f(k)→f(i) ∪ f(m)

D: f(l)→Dt

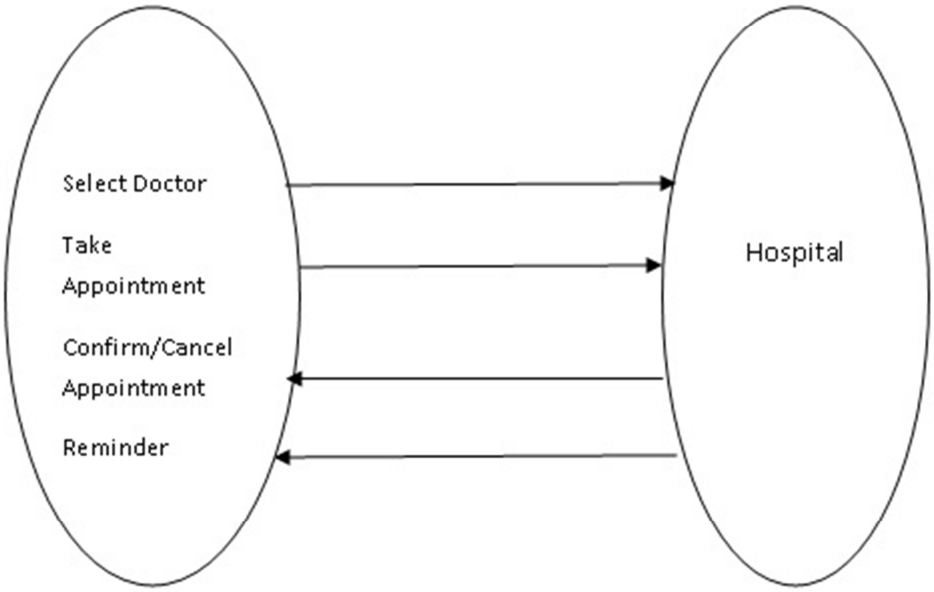
Venn Diagram

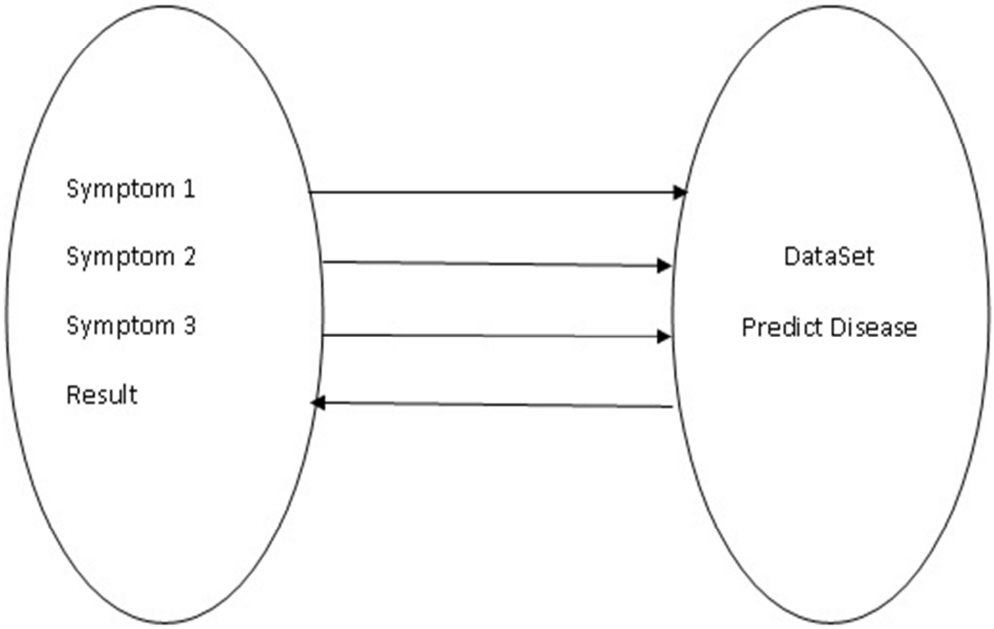


Functional Mapping



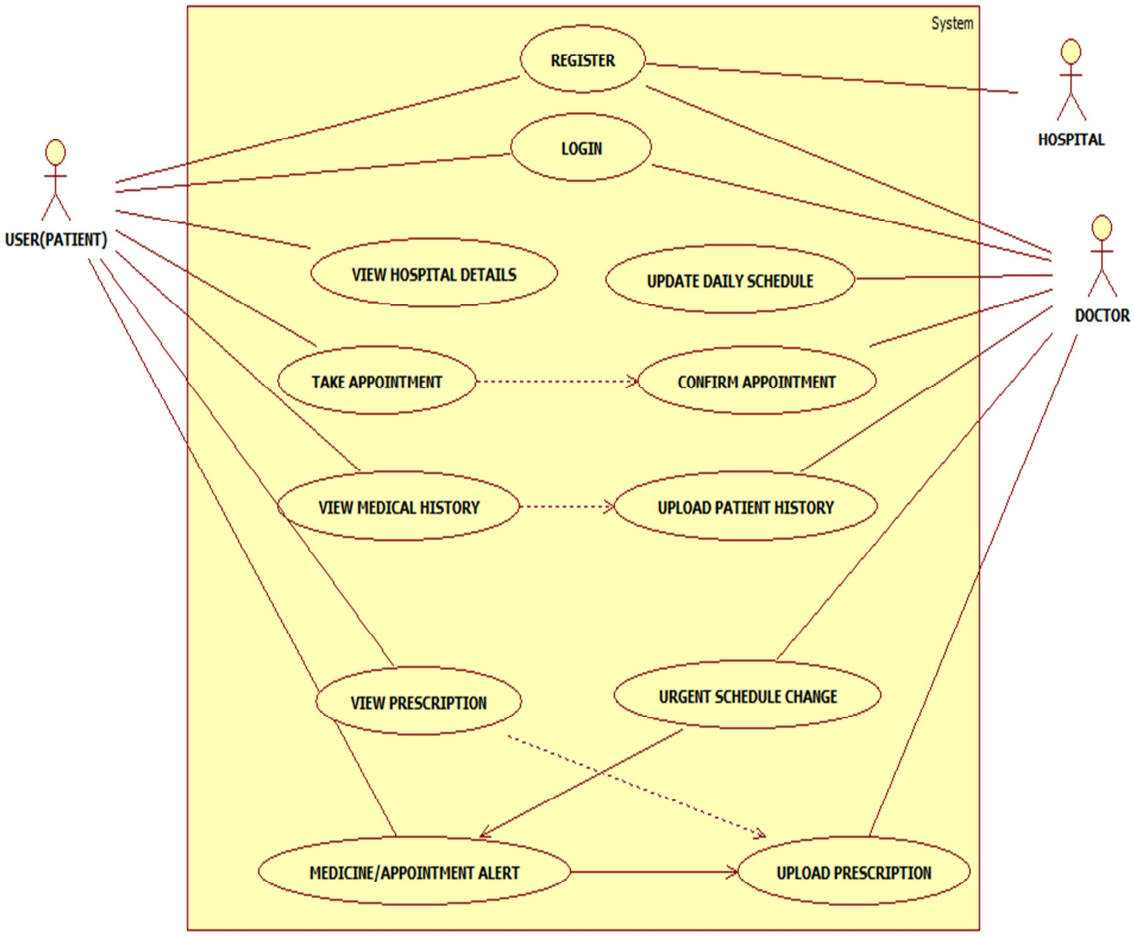
Functional Mapping



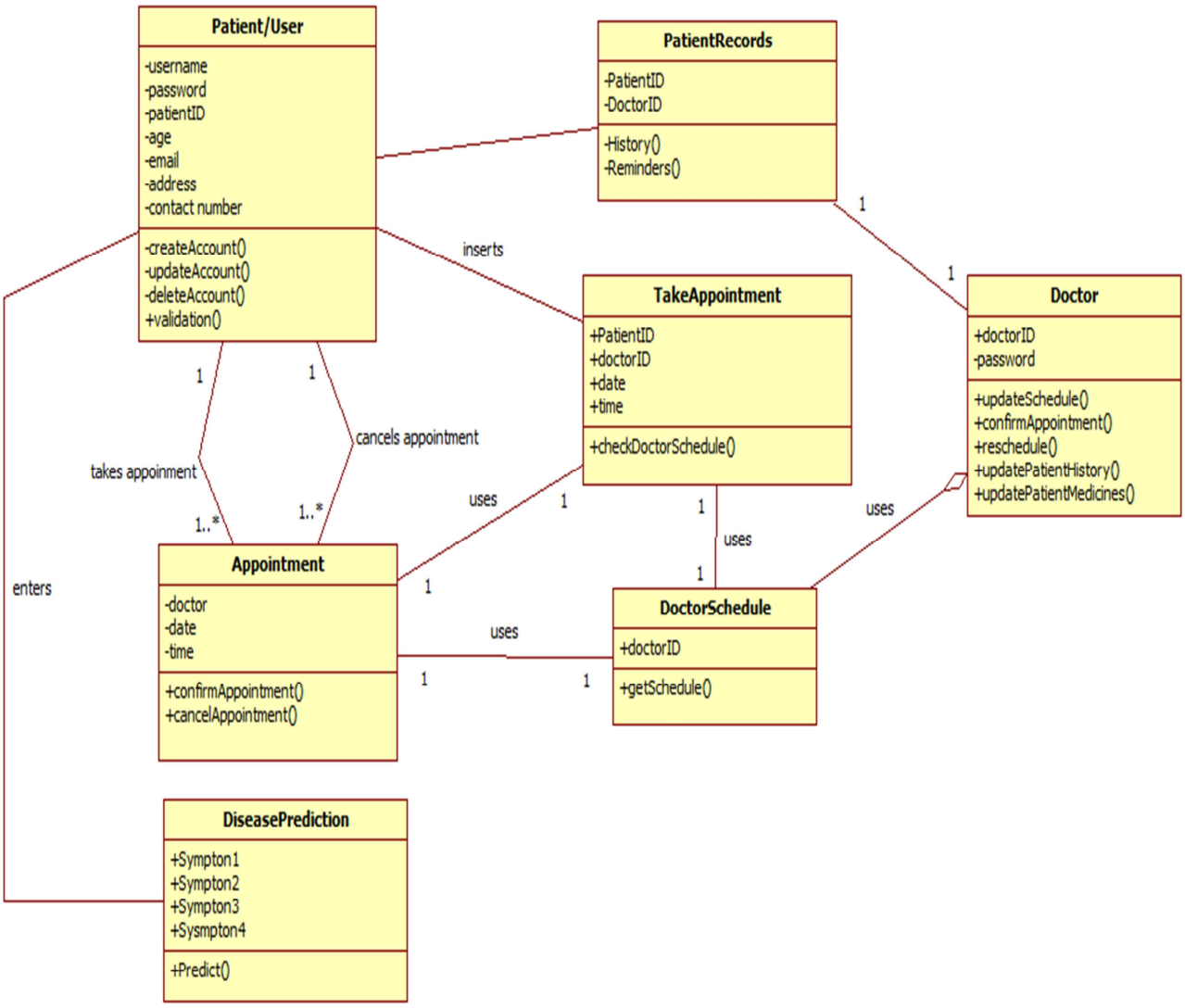


3. UML Diagrams

- **use case diagram**



- **Class diagram**



- **Activity Diagram**