ABSTRACT

Topic: Hospital-E-tocken management

Project nature:New topic

Completion level: 0%

Hospitals currently use a manual system for visiting Doctor Slip as a token.The system requires numerous paper forms, with data stores spread throughout the hospital management infrastructure. Often information is incomplete, or does not follow management standards.

Forms are often lost in transit between departments requiring a comprehensive auditing process to ensure that no vital information is lost.Through this project we are aiming an application/website that helps the hospital function and managements, it also help the people who want to visit hospital.An efficient and cost wise fashion so that an institution's resources may be effectively utilized Hospital E-Token management will automate the management of the hospital making it more efficient and error free for outdoor patient. It aims at standardizing data, consolidating data ensuring data integrity and reducing inconsistencies.

MODULES

1. Admin
2. Patient
3. Department

1.Admin

✓ Login
✓ Hospital management
✓ Department mangment
✓ Managing doctor information
✓ Managing token session
✓ Payment system
✓ Laboratory session
✓ Emergency sevices

2.Patient

✓ Login
✓ Registeration
✓ Hospital details view
✓ Doctor details view
✓ Booking of token
✓ Lab test booking
✓ Payment
✓ notification

3.Department

- ✓ Login
- ✓ Register
- ✓ Hospital management
- ✓ Managing doctor information
- ✓ Managing token session
- ✓ Payment system
- ✓ Laboratory session

Existing system:

The hospital management use a manual system for visiting Doctor Slip as a token.The system requires numerous paper forms, with data stores spread throughout the hospital management infrastructure. Often information (on forms) is incomplete, or does not follow management standards.

- ✓ Take more time
- ✓ Less security
- ✓ High manpower
- ✓ Less efficient
- ✓ Loss and multiple copies of information etc..

Proposed system:

Through this project we are aiming an application/website that helps the hospital function and managements, it also help the people who want to visit hospital.An efficient and cost wise fashion so that an institution's resources may be effectively utilized Hospital E-Token management will automate the management of the hospital making it more efficient and error free for outdoor patient. It aims at standardizing data, consolidating data ensuring data integrity and reducing inconsistencies.

- ✓ We aim to make the hospital procedures easier.
- ✓ We can reduce the rush at the hospitals thus we can reduce spreading of contagious diseases through patients In hospitals.
- ✓ People can conveniently book the ticket and pay the bill through their phone.
- ✓ People can check the availability of the doctors easily.

Research Paper:-

Fernet system for symmetric encryption

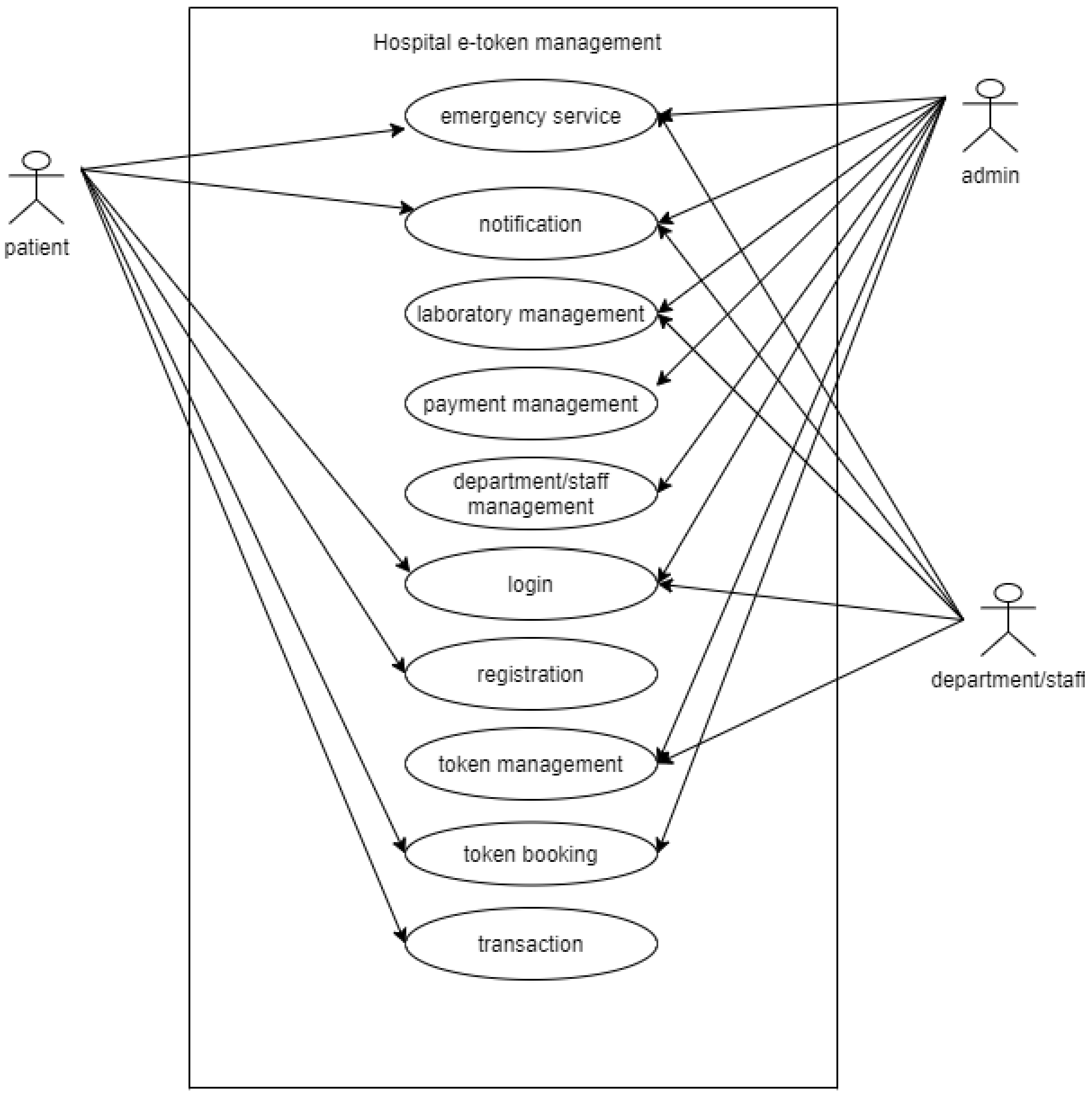
Data can be secured using Fernet System .Cryptography is the practice of securing useful information while transmitting from one computer to another or storing data on a computer. Cryptography deals with the encryption of plaintext into ciphertext and decryption of ciphertext into plaintext. Python supports a cryptography package that helps us encrypt and decrypt data. The fernet module of the cryptography package has inbuilt functions for the generation of the key, encryption of plaintext into ciphertext, and decryption of ciphertext into plaintext using the encrypt and decrypt methods respectively. The fernet module guarantees that data encrypted using it cannot be further manipulated or read without the key.

Methods Used:

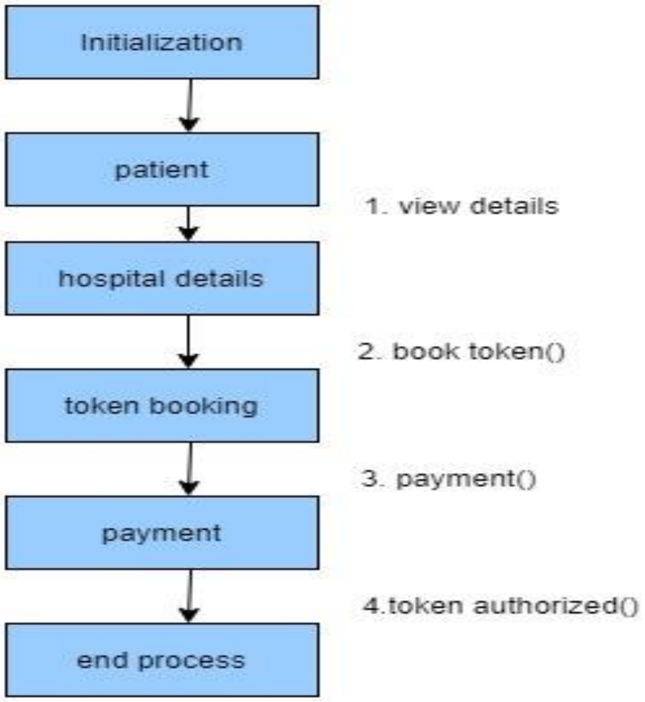
- generate_key() : This method generates a new fernet key. The key must be kept safe as it is the most important component to decrypt the ciphertext. If the key is lost then the user can no longer decrypt the message. Also if an intruder or hacker gets access to the key they can not only read the data but also forge the data.

- encrypt(data) : It encrypts data passed as a parameter to the method. The outcome of this encryption is known as a “Fernet token” which is basically the ciphertext. The encrypted token also contains the current timestamp when it was generated in plaintext. The encrypt method throws an exception if the data is not in bytes.

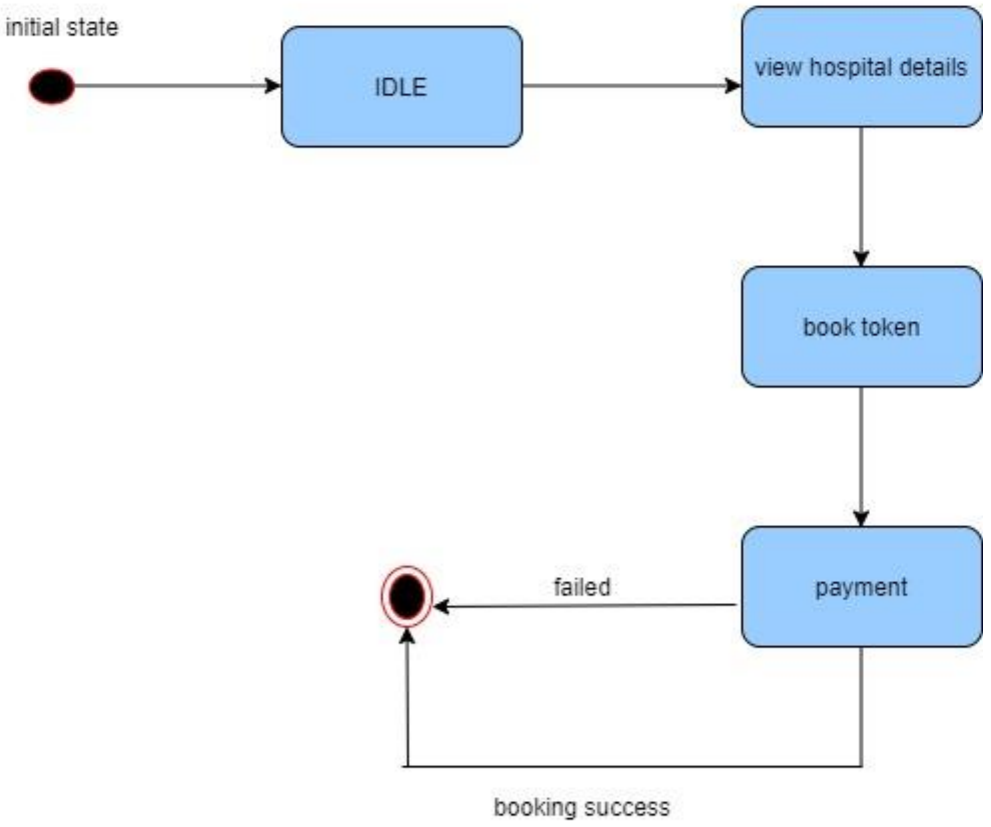
UML DAIGRAMS



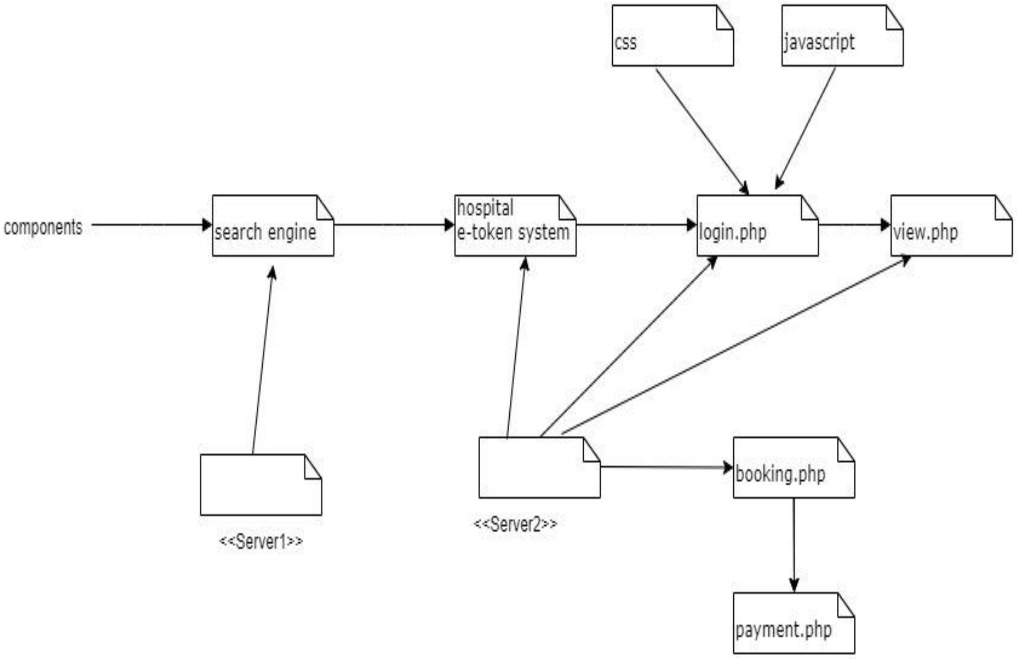
Collaboration diagram



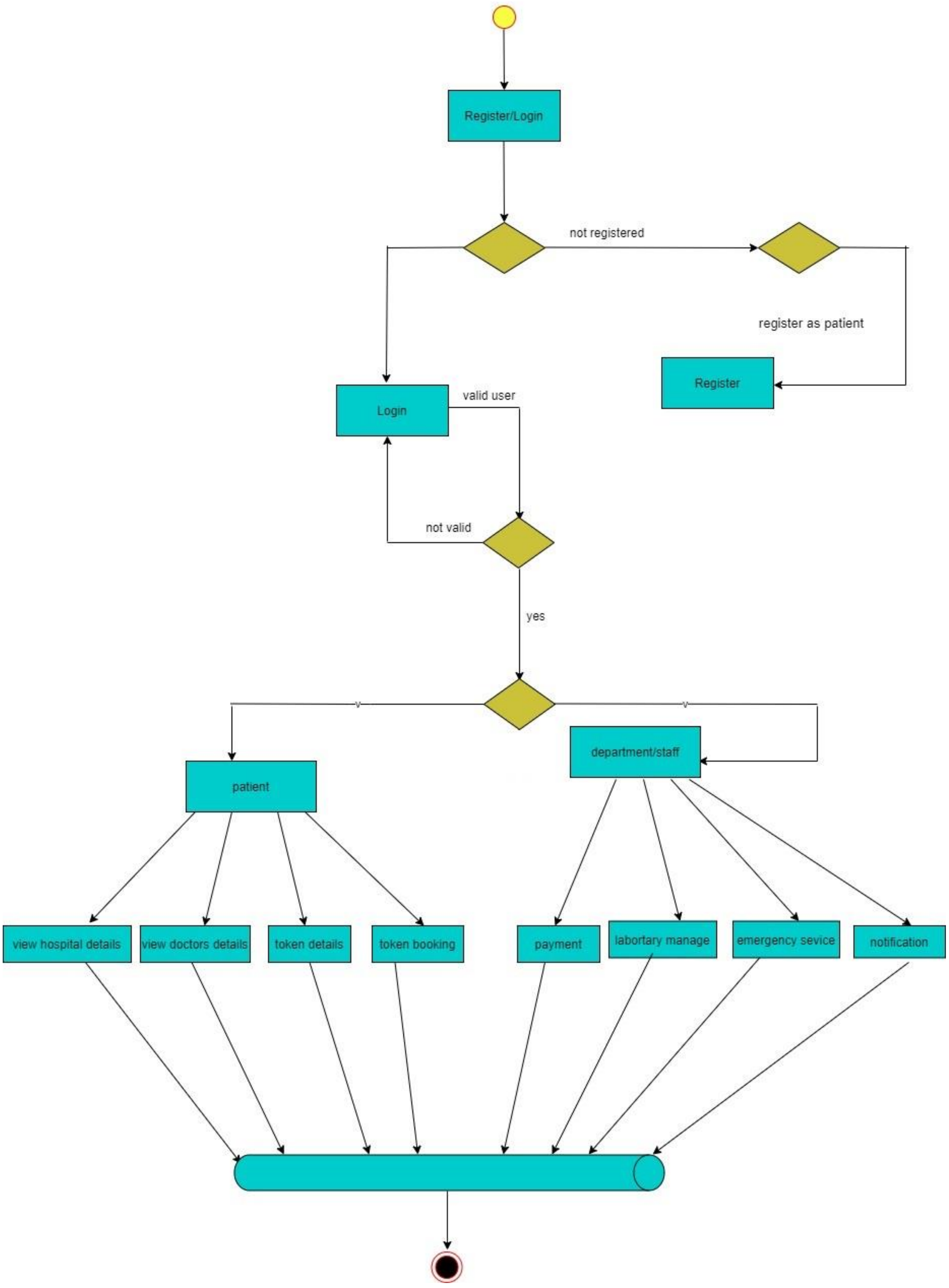
State chart Diagram



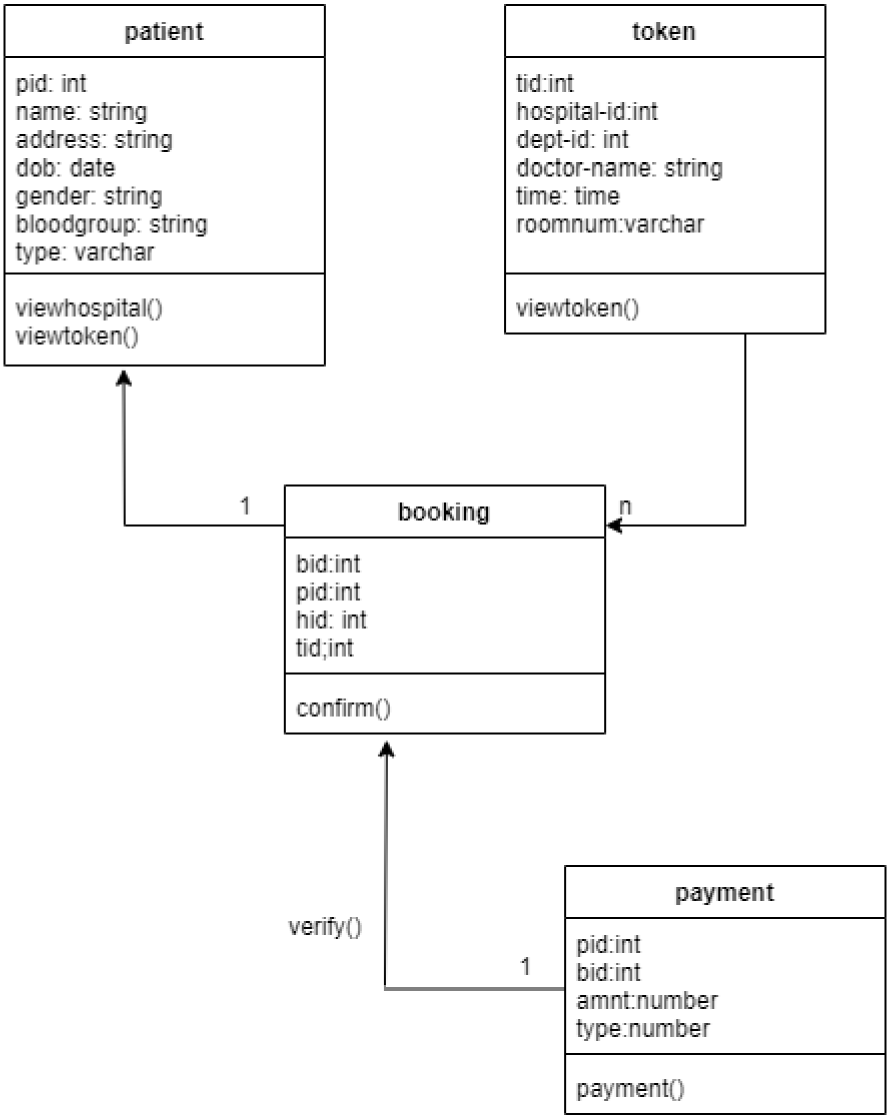
Component Diagram



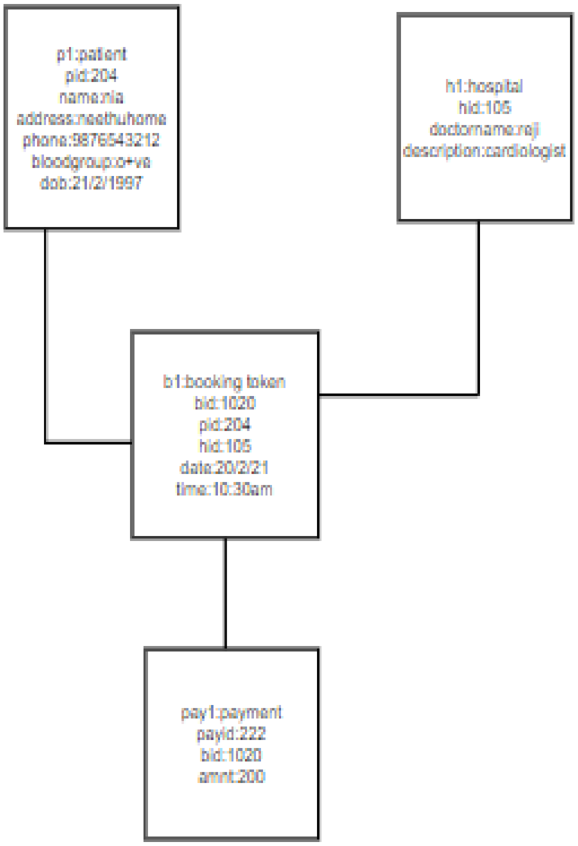
Activity diagram



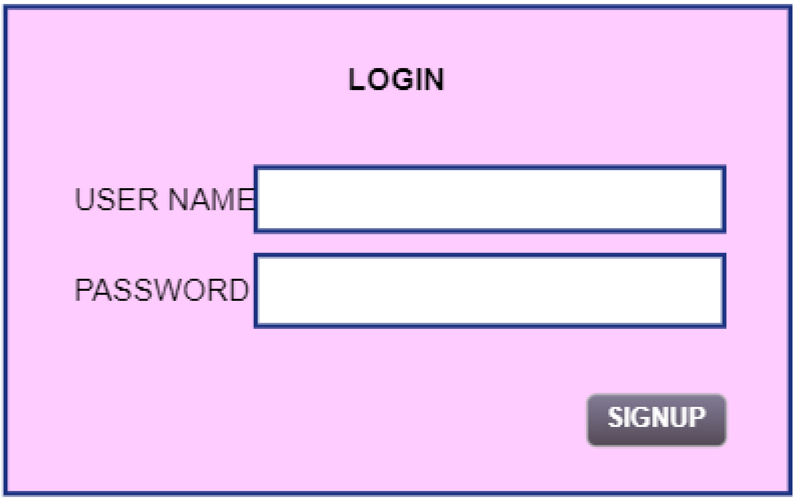
Class Diagram

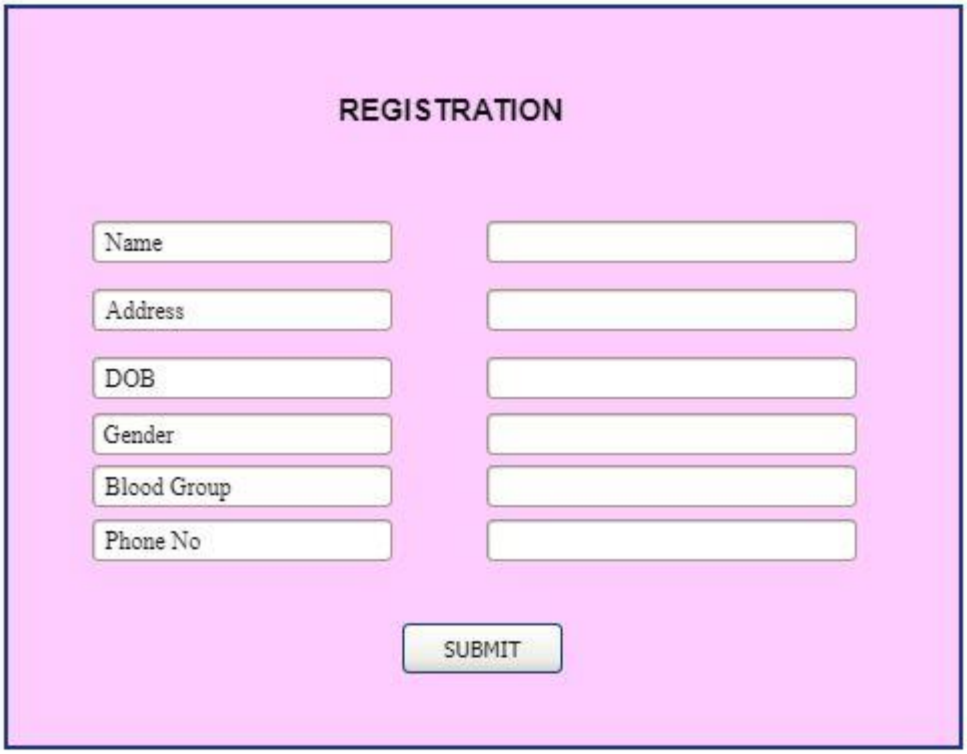


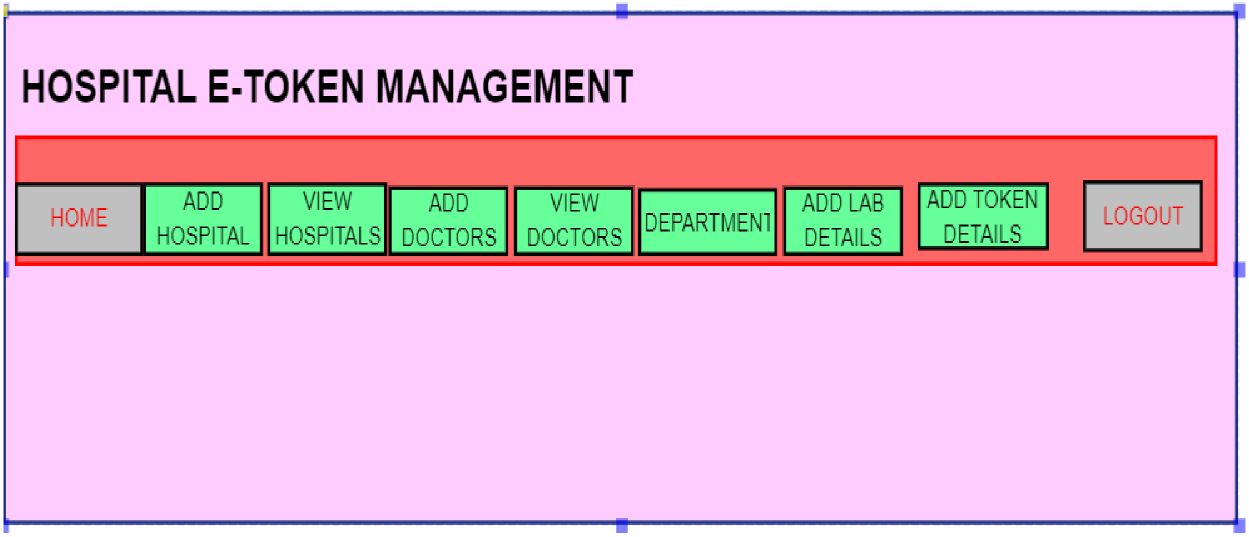
Object Diagram

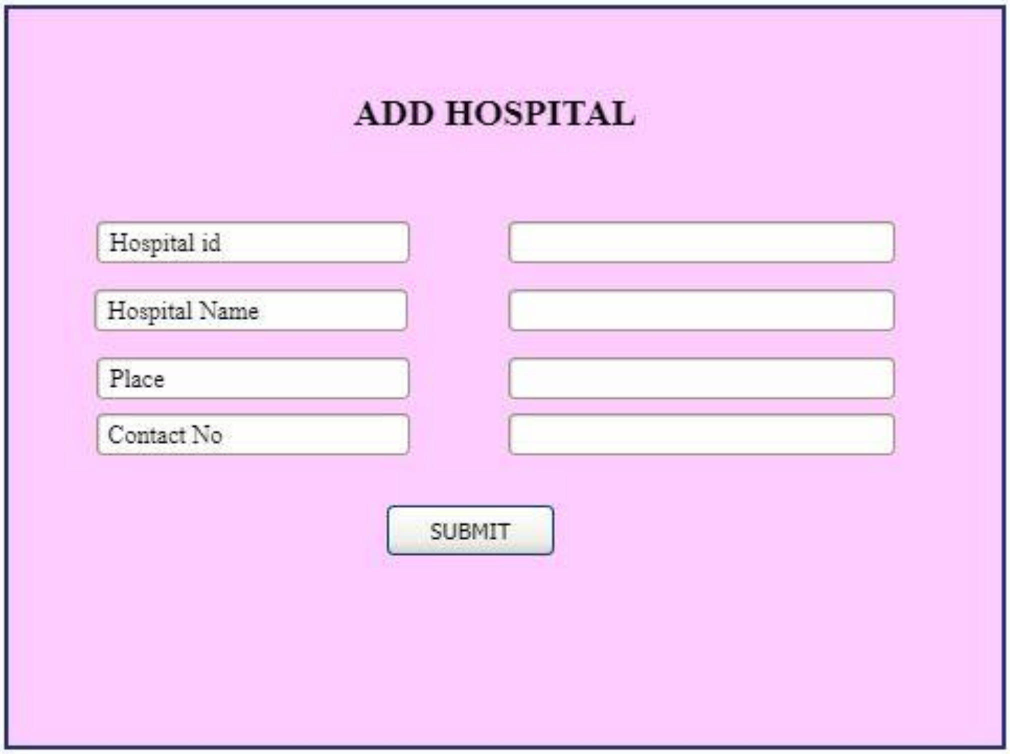


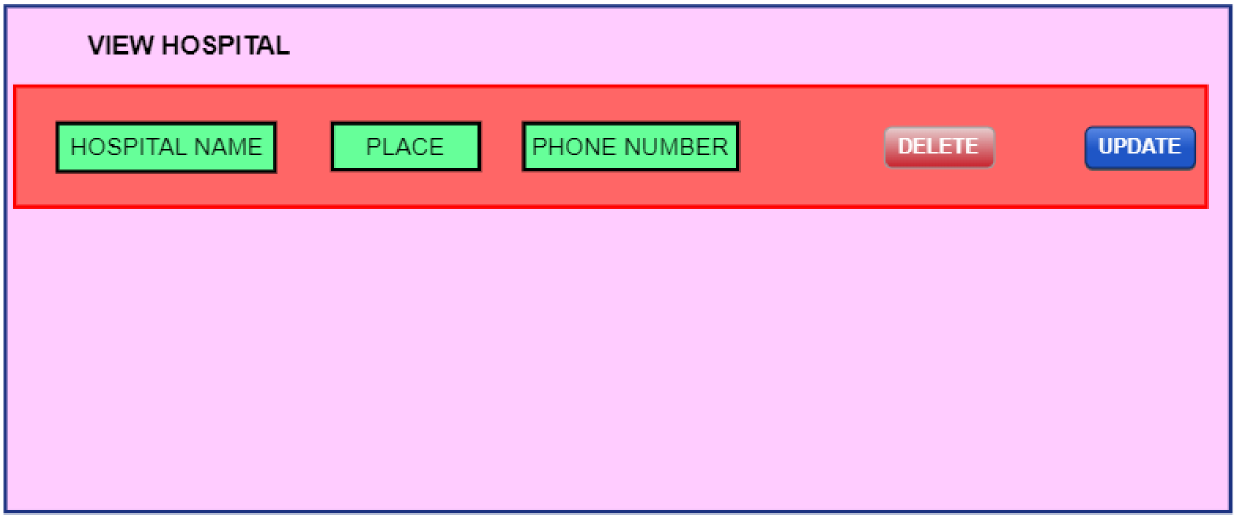
Forms

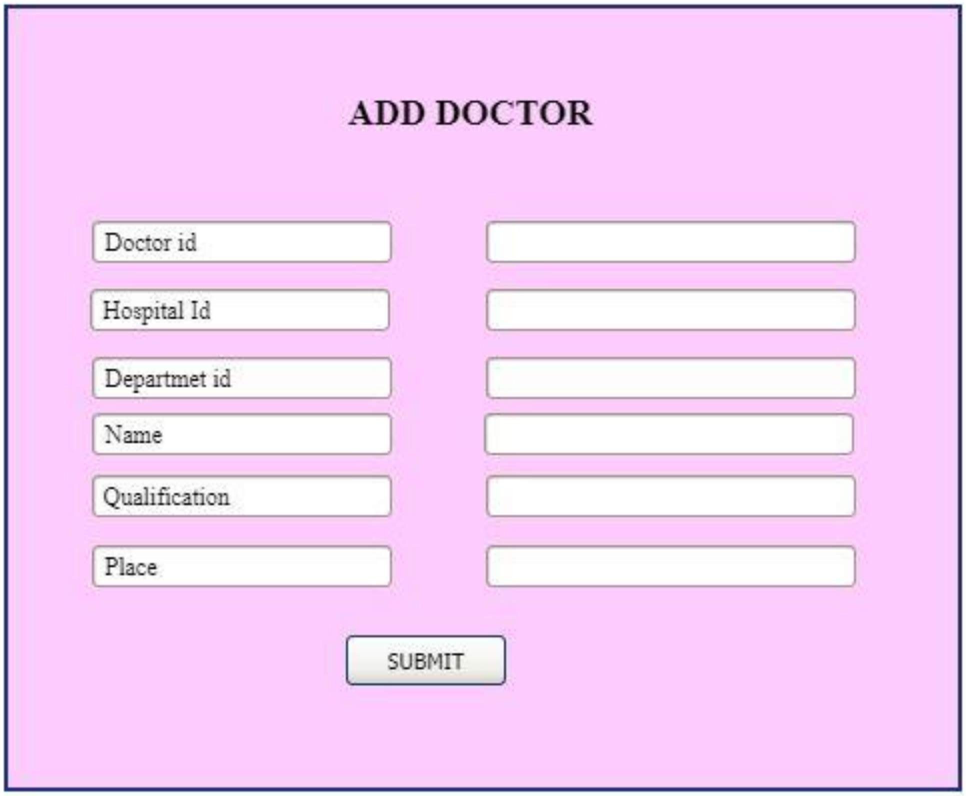

LOGIN
USER NAME
PASSWORD
SIGNUP

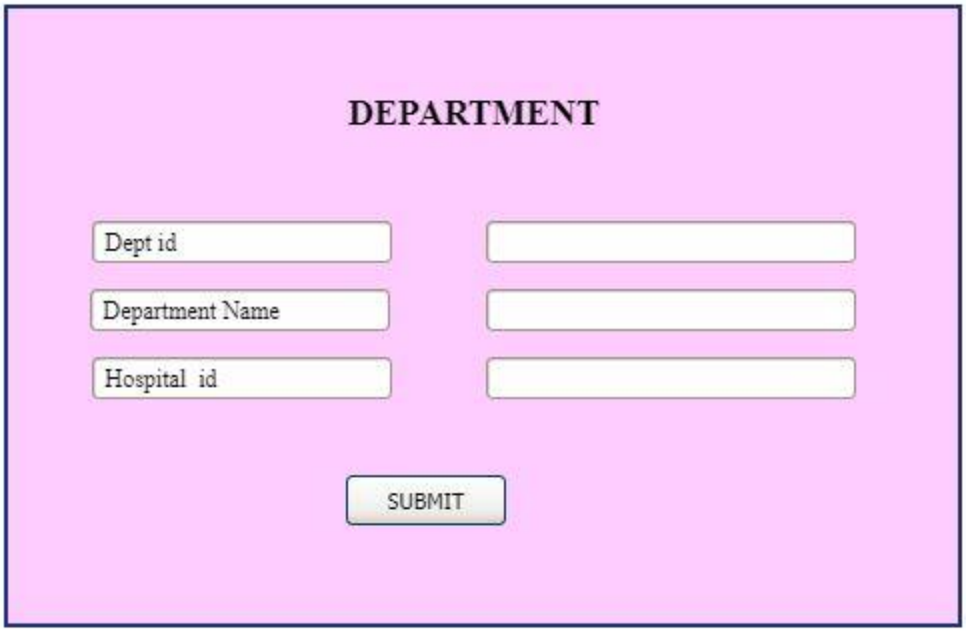

REGISTRATION
Name
Address
DOB
Gender
Blood Group
Phone No
SUBMIT


HOSPITAL E-TOKEN MANAGEMENT
HOME
ADD HOSPITAL
VIEW HOSPITALS
ADD DOCTORS
VIEW DOCTORS
DEPARTMENT
ADD LAB DETAILS
ADD TOKEN DETAILS
LOGOUT


ADD HOSPITAL
Hospital id
Hospital Name
Place
Contact No
SUBMIT


VIEW HOSPITAL
HOSPITAL NAME
PLACE
PHONE NUMBER
DELETE
UPDATE


ADD DOCTOR
Doctor id
Hospital Id
Departmet id
Name
Qualification
Place
SUBMIT

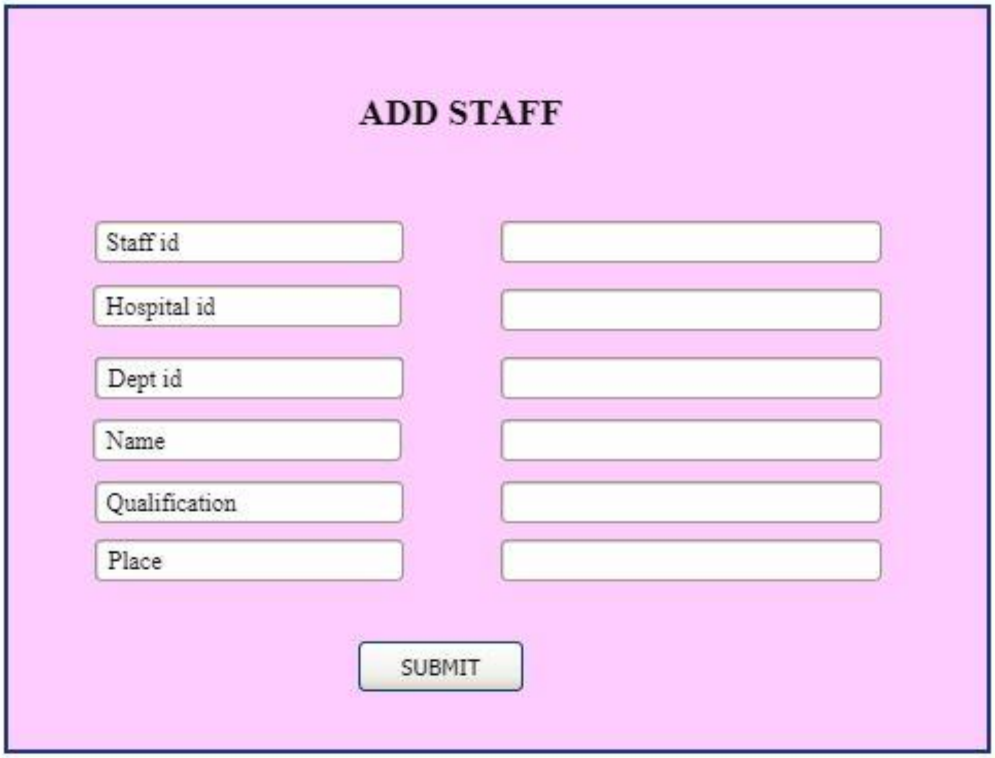


DEPARTMENT

Dept id

Department Name

Hospital id



ADD STAFF

Staff id

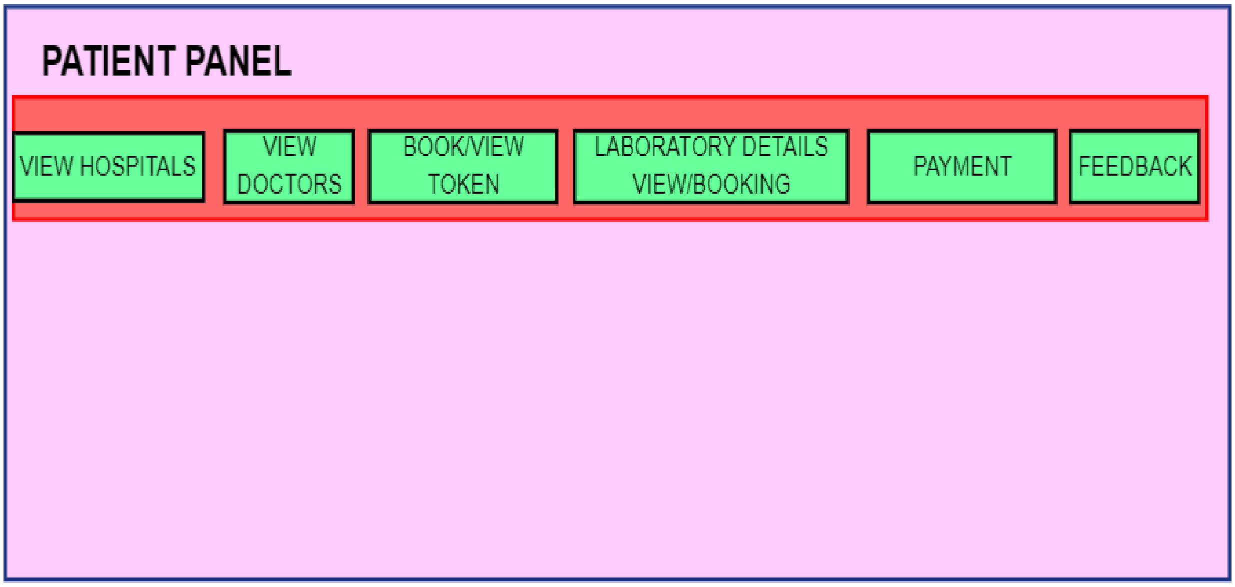
Hospital id

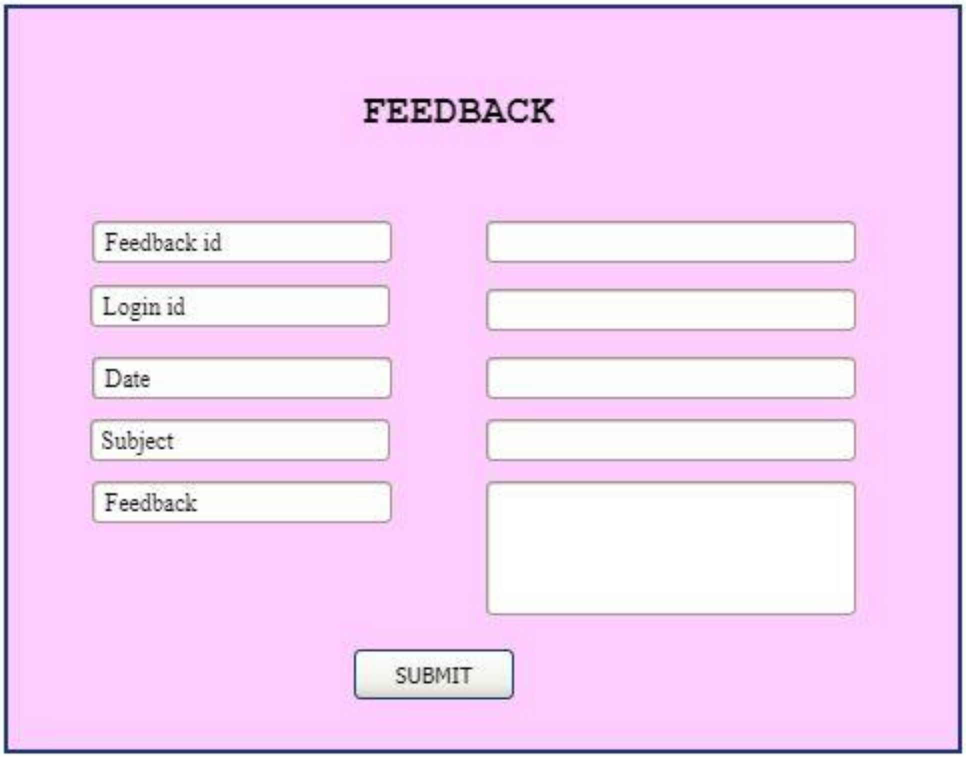
Dept id

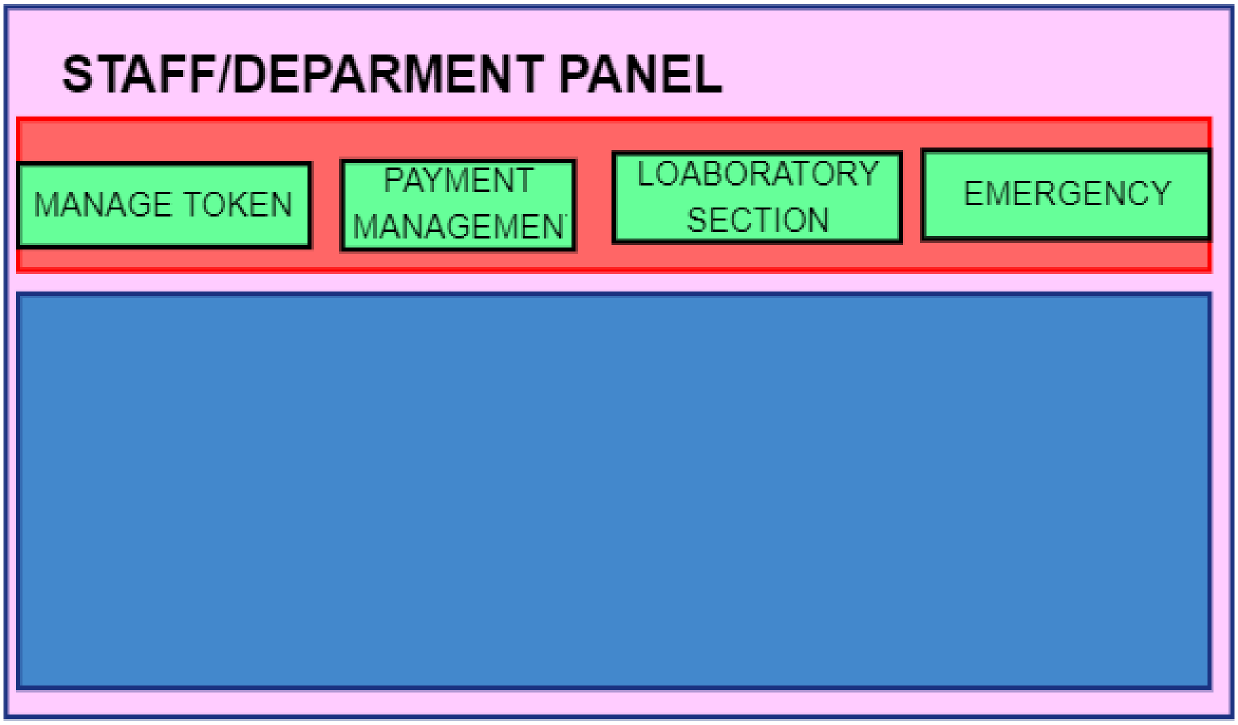
Name

Qualification

Place


PATIENT PANEL
VIEW HOSPITALS
VIEW DOCTORS
BOOK/VIEW TOKEN
LABORATORY DETAILS VIEW/BOOKING
PAYMENT
FEEDBACK


FEEDBACK
Feedback id
Login id
Date
Subject
Feedback
SUBMIT


STAFF/DEPARMENT PANEL
MANAGE TOKEN
PAYMENT MANAGEMEN
LOABORATORY SECTION
EMERGENCY

1.Table name: Login_tbl

Field name	Datatype	Size	Description
login_id	int	10	Primary key of login table
username	Varchar	50	Username
password	Varchar	50	Password for login

2.Table name: patient_tbl

Primary key: patient_id

Field name	Datatype	Size	Description
patient_id	int	10	Primery key of registration table
Name	Varchar	20	Name of patient
Address	Varchar	20	Address of patient
Dob	Date	20	Dob of patient
Gender	Varchar	20	Gender
Bloodgroup	Varchar	20	Blood group
Phone	number	20	Phone number of patient

3.Table name: hospital_tbl

Primary key: Hospital_id

Field name	Datatype	size	Description
Hospital_id	int	11	Primary key of hospital_tbl
Hospital name	varchar	20	Name of hospital
Place	Varchar	20	Place
Contact_no	Number	20	Contact number of hospital

4.Table name: dept_tbl

Primary key: Dept_id

Foreign key: Hospital_id

Field name	Datatype	size	Description
Dept_id	int	11	Primary key of dept_tbl
Dept_name	varchar	20	Foreign key
Hospital_id	Int	11	Foreign key of hospital_tbl

5.Table name: doctor_tbl

Primary key: doctor _id

Foreign key: hospital_id,dept_id

Field name	Datatype	size	Description
Doctor_id	Int	11	Primary key
Hospital_id	Int	11	Foreign key of hospital tbl
Dept_id	Int	11	Foreign key of dept tbl
name	Varchar	20	Name of doctor
Qualification	Varchar	20	Qualifications
place	Varchar	20	Place

6.Table name: token_tbl

Primary key: token_id
Foreignkey: patient_id,hospital_id,dept_id

Field name	Datatype	size	Description
Patient_id	int	10	Foreign key of patient reg _tbl
Hospital_id	int	11	Foreign key of hospital tbl
Dept_id	Int	11	Foreign key of dept tbl
Doctor name	Varchar	20	Name of doctor
Time	Varchar	20	Time slot
Room number	Varchar	20	Room number
Token_id	int	11	Primary_key

7.Table name: feedback_tbl

Primary key: feedback_id
Foreignkey: login_id

Field name	Datatype	Size	Description
feedback_id	Int	11	Primary key
Login_id	Int	10	Foreign key of login tbl
Date	Varchar	50	Date
subject	Varchar	50	Subject
feedback	varchar	150	Feedback of patients

8.Table name: ambulance_tbl

Primary key: amb_id

Field name	Datatype	size	Description
Amb_id	int	11	Primary key of ambulance tbl
Service_name	Varchar	20	Which service needs
Location	Varchar	20	Location
Status	Varchar	20	Status