ONLINE HEALTH CARE PORTAL

SOFTWARE ENGINEERING PROJECT

SUPERVISIOR:	SUBMITTED BY:
DR.APARNA DATT	VARSHA G SHENOY-8872
	HARSHITA SRIVASTAVA-8816

COURSE: B.Sc(H) COMPUTER SCIENCE

SEMESTER: IV

DEPARTMENT OF COMPUTER SCIENCE

PGDAV COLLEGE (DAY)

UNIVERSITY OF DELHI

CERTIFICATE

This is to certify that the project entitled, “ONLINE HEALTH CARE PORTAL” has been done by: VARSHA G SHENOY AND HARSHITA SRIVASTAVA of Bachelor of Science in Computer Science during semester IV from PGDAV College (Day) University of Delhi under the supervision of Dr. APARNA DATT.

DECLARATION

I hereby declare that this Project Report titled "ONLINE HEALTH CARE PORTAL" submitted to the Department of Computer Science, PGDAV College (Delhi University) is a record of original work done by the team under the guidance of Dr. Aparna Datt.

The information and data given in the report is authentic to the best of the team knowledge.

This Project Report is not submitted to any other university or institution for the award of any degree, diploma or fellowship or published any time before.

ACKNOWLEDGEMENT

The contentment that is achieved on the successful completion of any task is incomplete without mentioning the names of the people who made it possible with their consistent guidance, support and indispensable encouragement.

The Project was jointly undertaken by HARSHITA SRIVASTAVA AND VARSHA G SHENOY as their 4th semester Software Engineering Project, under the able guidance and supervision of DR. APARNA DATT. Our primary thanks to her, who poured over every inch of our project with painstaking attention and helped us throughout the working of the project.

It’s our privilege to acknowledge our deepest sense of gratitude to her for her inspiration which has helped us immensely. We are extremely grateful to her for unstilted support and encouragement in the preparation of this project.

PROBLEM STATEMENT

A lot of patients nowadays are seeking a more convenient way to avail healthcare services where in the current scenario they have to visit the doctor and wait for long periods of time to get an appointment.

Doctors, nowadays, find it uneasy to manually maintain and view the records of the patients who visit them on a regular basis.

To solve the above stated problems, we propose to develop a software which will provide an online platform for :

- Patients to book their appointments on a prior basis with the doctor of their choice
- Doctors to maintain patient details and history by storing the information online on the software

Current Scenario

- Long queues of patients in hospitals/clinics waiting for an appointment with the doctor.
- Patients do not get real time updates of whether a doctor will be available on a particular time.
- Doctors have to manually maintain patient records which is time consuming and inconvenient to view in future.

What the proposed system will do ?

- Patients can book their appointments online with a desired doctor as per their requirements from a vast list of doctors in the town.
- They can view feedback, doctor online profiles, consultation fees and other details.
- They will get real time search results as per the availability of a doctor at the time the patient requires.
- Doctors can enter the patient details and history records which will get stored in a computer database and can be viewed later.

How will it work ?

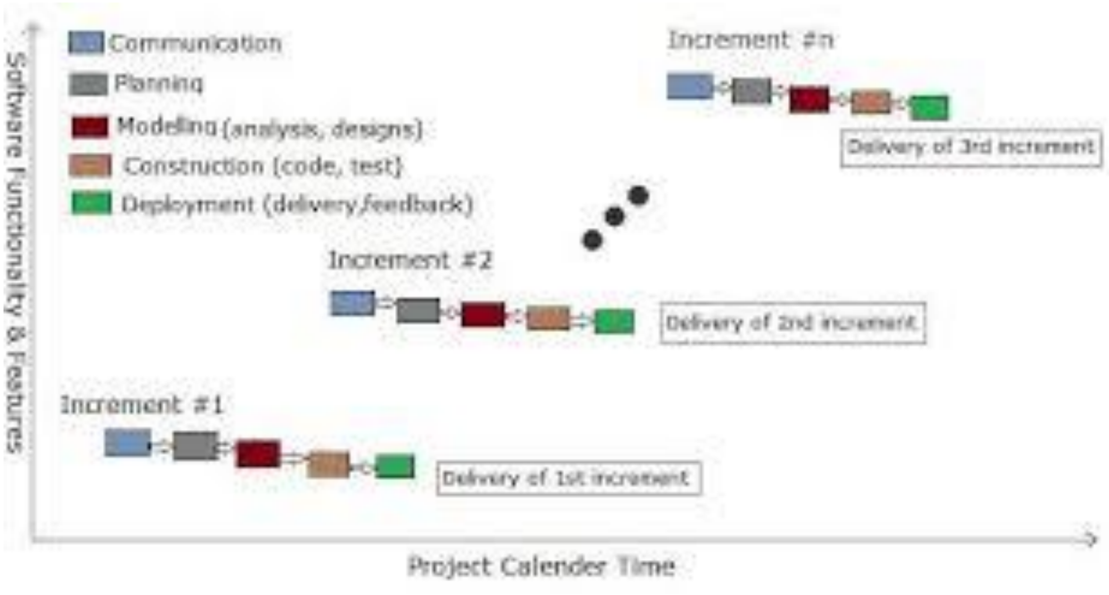
- Patients and doctors can login into the application by providing login details.
- Patients will enter their requirements like location, specialization of doctor and can view various booking slots available and select the date and time.
- The already booked space will be marked yellow.
- There will also be an option to cancel the appointment anytime.
- When a doctor enter the application, he/she can enter the patient details which will be stored in a computerized database.
- Both patients and doctors can view the data for further requirements.

SOFTWARE PROCESS MODEL

INCREMENTAL MODEL

The incremental model combines elements of the waterfall model applied in an iterative fashion. Referring to the figure, the incremental model applies linear sequences in a staggered fashion as calendar time progresses. Each linear sequence produces deliverable “increments” ofthe software. When an incremental model is used, the first increment is often a core product. That is, basic requirements are addressed, but many supplementary features (some known, others unknown) remain undelivered. The core product is used by the customer (or undergoes detailed evaluation). As a result of use and/or evaluation, a plan is developed for the next increment.

The plan addresses the modification of the core product to better meet the needs of the customer and the delivery of each increment, until the complete product is produced.


Communication
Planning
Modeling (analysis, designs)
Construction (code, test)
Deployment (delivery,feedback)
Increment #n
Delivery of 3rd increment
Increment #2
Delivery of 2nd increment
Increment #1
Delivery of 1st increment
Software Functionality & Features
Project Calender Time

Reason for using this model:

The reason for using this that the concept of our software is vast and there are a lot of modules that are required to be built. This increases the chances of an error. Use of an increment model will allow us to analyze our project after completion of each module. Thus we can change our requirements at the start of each module.Moreover after our first increment; we can launch our core product for the users. After use, evaluation and feedback by the users we can modify our software and take it to a more advanced level.

Functional requirements

1. **Login Module** - This module is for both the type of users - patients and doctors. There is a login interface asking for basic details from both type of users and will register them.

2**. Doctor Module**: Doctors can register themselves via this module and can make their account here. They can access anything related to patient after making account into this module. Also doctors can post any status update too whenever they wish, in order to inform all the patients some important information together.

3. **Patient Module**: Accounts of the patient are made up in this module and patients can themselves add their status and other queries into this module. Patients can take appointments by choosing a doctor suitably according to their requirements.

4. **Data Module**: All the data which the patient cannot enter into the system by themselves can be entered into this module and only doctors can access those data via their account.

5. **Admin Module**: All the data and information which is uploaded for the easy access by the doctors is done via the administration panel i.e. Only by the admin

Software Requirement Specification OR SRS

It is the means of translating ideas [inputs in the minds of people] into a formal document [output] which consist of a set of formally specified requirementS which are complete and consistent.

Need of SRS

- Needed due to the changing needs of the user and even for complex applications, needs are not known.
- Errors can be reduced by properly performing the requirement phase.
- Provides the agreement between the client and the supplier.
- Provides a reference for validation of the final product.
- A high quality SRS can also reduce the development cost.

Requirement Analysis

- It states what we need from the system for the problem
- Understanding the problem that the software system is going to solve
- The ideas are transformed from client's mind to a proper formal document known as SRS [Software Requirement Specification]
- The main emphasis is on what we need from the system not how we will achieve the goal
- The task is complicated due to the involvement of two parties i.e. the client and the software developer.

Use in our project

The system will be used as the application that serves hospitals, clinic, dispensaries or other health institutions. The intentions of the system are to increase the number of patients that can be treated and managed properly.

If the hospital management system is file based, management of the hospital has to put much effort on securing the files. They can be easily damaged by fire, insects and natural disasters. Also could be misplaced by losing data and information.

Product Perspective

This Online Health Care portal is a self-contained system that manages activities of the hospital along with patient details and feedbacks. Due to improperly managed details medical center faces quite a lot of difficulties in accessing past data as well as managing present data. The fully functional automated online health portal which will be developed through this project will eliminate the disadvantages caused by the manual system by improving the reliability, efficiency and performance. This also help patient to easily find their nearby doctor respective of their issues. The usage of a database to store patients, doctors etc. will accommodate easy access, retrieval, search and manipulation of data. The access limitations provided through access privilege levels will enhance the security of the system. The system will facilitate concurrent access and convenient management of activities of the medical center.

Product Functions performed:

- Provide access to registered users only
- Registration of new patients
- Enable patient to view their record
- Enable patient to update their record
- Generate appointment date and timing
- Confirmation by doctor
- Modification in schedule by doctor
- Admin access to patient’s record

- Admin access to doctor’s record

Assumptions and Dependencies

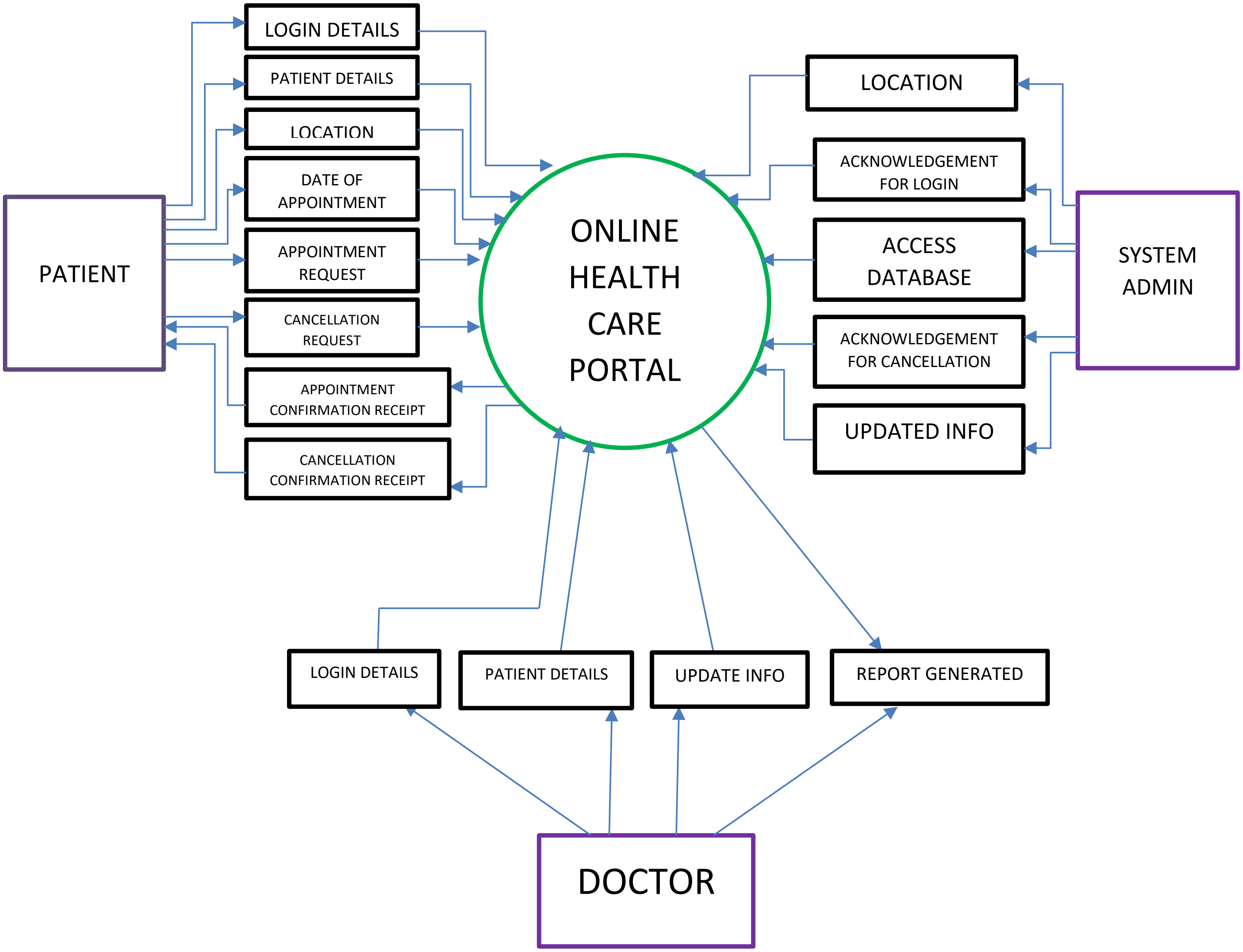
- Each user must have a valid user id and password
- Server must be running for the system to function
- Users must log in to the system to access any record.
- Only the Administrator can delete records.

DATA FLOW DIAGRAM

A Data Flow Diagram is a graphical representation of the “flow” of the data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated.

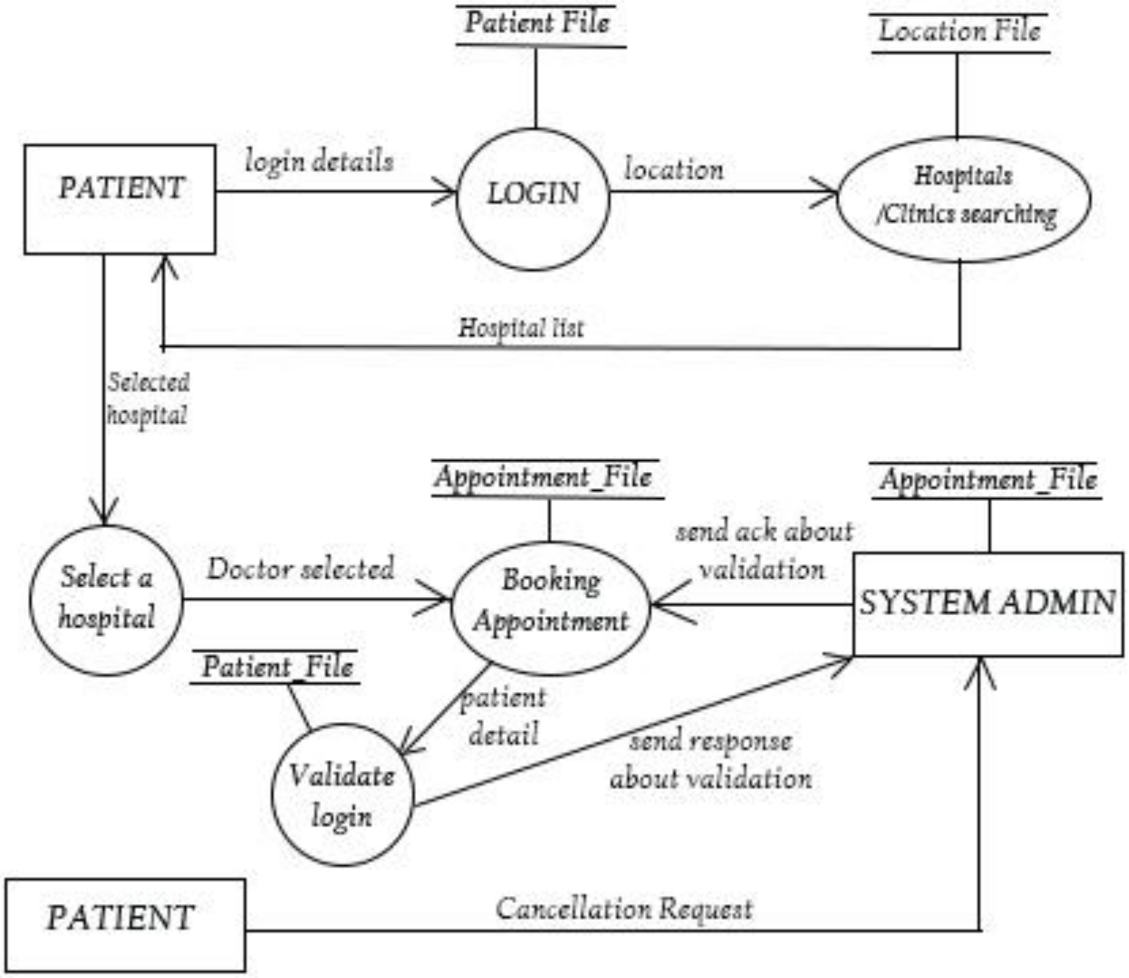
- DFD can also be used for the visualization of data processing.
- A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to and where the data will be stored .It doesn't show information about the timing of process or information about whether processes will operate in sequence or in parallel.

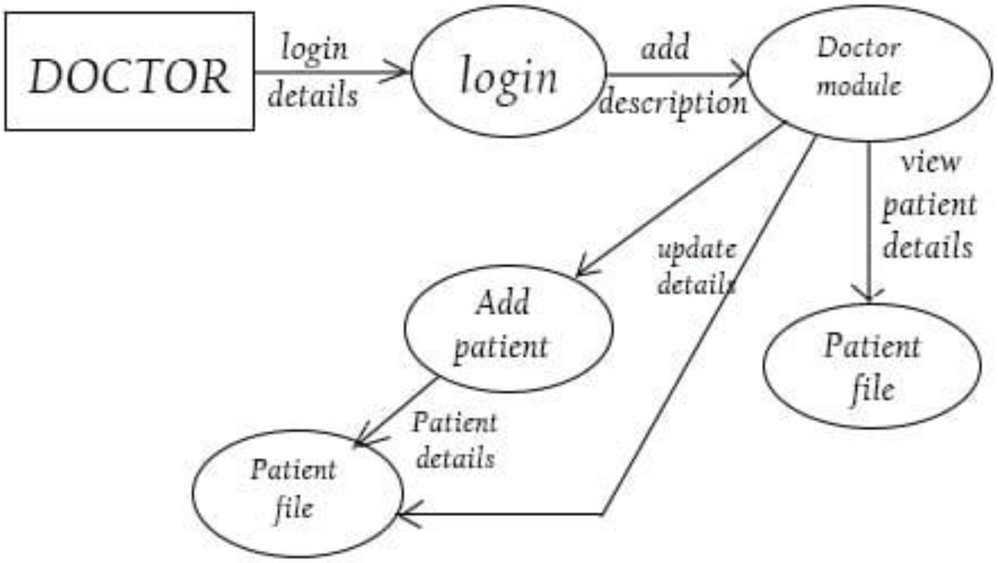
CONTEXT DIAGRAM (LEVEL 0)



LEVEL 1 DFD

(FOR PATIENTS)





LEVEL I DFD (FOR DOCTORS)

DATA DICTIONARY (level 0)

Login=username+password

Patient detail=name+sex+address+email+mobile no.

DATA DICTIONARY (level 1 for patients)

Patient detail=name+sex+address+email+mobile no.

Appointment=patient_detail+doctor_detail+appointment_date

DATA DICTIONARY(level 1 for doctors)

Doctor_details=Did+name+moblile no.+speciality

Appointment=patient_detail+doctor_detail+appointment_date

SYSTEM FEATURES

PATIENT

REGISTRATION

DESCRIPTION- The new patient can register themselves and add their details like name, age sex etc. The patient entry will be made in the patient database.

PRE -CONDITION – The patient must be a new patient.

MAIN FLOW OF EVENTS- 1. Patient selects sign up in login module.
2. A registration form get
displayed 3.Patient fills the
required details.

EXTENTSIONS- if necessary fields left by user then prompt user to fill the

necessary fields. POST CONDITIONS- Patient record is added to patient

database.

UPDATION

DESCRIPTION-The patient should be enabled to update his/her details and the changes should reflect in patient database.

PRE-CONDITION – The patient must be a

registered patient. MAIN FLOW OF EVENTS- 1.

Patient logs in to the system.

2. Patient view his record
3. Patient selects update details.
4. Now patient may change the necessary fields.
5. Pop of update details.

EXTENSIONS – The patient cannot update details after

treatment starts. POST CONDITION- The record of

patient is updated in patient database.

APPOINTMENT

DESCRIPTION- It shows users a list of available doctors, timings, dates and enables patients to select the most suitable appointment date and doctor. The patient may also the cancel the appointment if already available.

PRE-CONDITION- The patient must be a registered patient.

MAIN FLOW OF EVENTS – 1.patient first logs in to system.

2. View his record
3. create a new appointment or cancel the appointment.

EXTENSIONS-Patient can fix only one appointment for a particular department.

POST CONDITIONS- patient details are displayed and a new appointment is fix or a existing appointment is cancelled. The patient database is updated

DOCTOR

DESCRIPTION- The doctor view patient record/ update his details and add description of the treatment given to patient.

PRE-CONDITION – The doctor must be a

registered doctor. MAIN FLOW OF EVENTS –

1.Doctor logs in to the system.

2.Doctor may select view patient.

2.1 patient record is displayed with treatment history.

3 Doctor add description of patient treatment.

4.Doctor update his/her details.

EXTENSIONS- System does not allow the doctor to modify the qualification, hospital managed details.

POST CONDITION – The patient and doctor ‘s database are updated.

ADMIN

DESCRIPTION- The admin add doctor, view patient record and appointments

PRE -CONDITION- Admin must first log in with his/her

credentials.

MAIN FLOW OF EVENTS-

1. Admin logs in the system.
2. admin may add doctor new doctor.
   - 2.1 admin fills the doctor's details
3. admin view patient record.
4. admin view appointments for the current date.
   - 4.1appointments for current date are displayed.

POST CONDITION- The doctor and admin database are updated.

EXTERNAL INTERFACE REQUIREMENTS

User Interfaces

This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.

The protocol used shall be HTTP.

The port number will be 80.

There shall be logical address of the system in IPV4 format.

Hardware Interfaces

Laptop/Desktop PC
Purpose of this is to give information when Patients ask information about doctors, medicine available lab tests etc. To perform such Action it need very efficient computer otherwise due to that reason patients have to wait for a long time to get what they ask for.

Laser Printer (B/W)
Simply this device is for printing patients info etc.

Wi-Fi router
Wi-Fi router is used to for internetwork operations inside of a hospital and simply data transmission from pc's to sever.

Software Interfaces

- JDK 1.8 - Java is fast, secure, and reliable. From laptops to data centers, game consoles to scientific supercomputers, cell phones to the Internet,
- Netbeans 8.1 - IDE for Java developing.
- MySQL server - Database connectivity and management
- OS Windows 7/8/8.1- Very user friendly and common OS
- JRE 1.8 - JAVA Runtime Environment for run Java Application and System
- MySQL server - Database connectivity

Communication interfaces

- NIC (Network Interface Card) – It is a computer hardware component that allows a computer to connect to a network . NICs may be used for both wired and wireless connections.
- CAT 5 network cable- for high signal integrity
- TCP/IP protocol- Internet service provider to access and share information over the Internet
- Ethernet Communications Interface- Ethernet is a frame-based computer network technology for local area networks (LANs)
- Ubiquitous, easy to set up and easy to use. Low cost and high data transmission rates.

Other Nonfunctional Requirements

Performance Requirements

Response time-The system will give responses within 1 second after checking the patient information and other information.

Capacity-The system must support 1000 people at a time

User interface- User interface screen will response within 5 seconds

Safety Requirements

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed upto archival storage and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure. All the administrative and data entry operators have unique logins so system can understand who is login in to system right now no intruders allowed except system administrative nobody cannot change record and valuable data.

Software Quality Attributes

AVAILABILITY: The system shall be available all the time.
CORRECTNESS: A bug free software which fulfill the correct need/requirements of the client.

MAINTAINBILITY:The ability to maintain ,modify information and update fix problems of the system
USABILITY: software can be used again and again without distortion.
ACCESSIBILITY: Administrator and many other users can access the system but the access level is controlled for each user according to their work scope.
ACCURACY: The reliability on the information/output. Can depend/be sure of the outcome.
STABILITY:The system outcome/output won't change time to time. Same output.Will be given always for a given input.

Security Requirements

1. Want take the responsibility of failures due to hardware malfunctioning.
2.Warranty period of maintaining the software would be one year.
3.Additional payments will be analysed and charged for further maintenance
4.If any error occur due to a user's improper use. Warranty will not be allocated to it. 5.No money back returns for the software.

ESTIMATION AND SCHEDULING

PROJECT MANAGEMENT

Project management involves the planning, monitoring, and control of the people, process, and events that occur as software evolves from a preliminary concept to an operational implementation. Project managers plan, monitor, and control the work of a team of software engineers. Effective software project management focuses on the four P‟s: people, product, process, and project.

FUNCTION POINTS

Function-oriented software metrics use a measure of the functionality delivered by the application as a normalization value. Since 'functionality' cannot be measured directly, it must be derived indirectly using other direct measures. Function points are derived using an empirical relationship based on countable (direct) measures of software‛s information domain and assessment of software complexity. Information domain values are defined in the following manner:

- **Number of user inputs:** Each user input that provides distinct application oriented data to the software is counted. Inputs should be distinguished from inquiries, which are counted separately.
- **Number of user outputs:** Each user output that provides application oriented information to the user is counted. In this context output refers to reports, screens, error messages, etc. Individual data items within a report are not counted separately.
- **Number of user inquiries:** An inquiry is defined as an on-line input that results in the generation of some immediate software response in the form of an online output. Each distinct inquiry is counted.
- **Number of files:** Each logical master file (i.e., a logical grouping of data that may be one part of a large database or a separate file) is counted.
- **Number of external interfaces:** All machine readable interfaces (e.g., data files on storage media) that are used to transmit information to another system are counted.

Once these data have been collected, the following table is completed and a complexity value is associated with each count.

COMPUTING FUNCTION POINTS

Information Domain Value	Count		Weighting factor				
			Simple	Average	Complex		
External Inputs(EIs)	5	*	**3**	4	6	=	15
External Outputs(EOs)	7	*	**4**	5	7	=	28
External Inquiries(EQs)	3	*	**3**	4	6	=	9
Internal Logical Files(ILFs)	4	*	**7**	10	15	=	28
External Interface Files(EIFs)	0	*	**5**	7	10	=	0
Count Total							**80**

Formula:-

To compute function points (FP), the following formula is used:

$$\boxed{FP = Count\_Total * [\ 0.65 + 0.01 * \sum (\ F_i\ )\ ]}$$

where,

- Count Total = Sum of the products obtained from multiplying counts of each information domain value by a **weighting factor** depending on their

- complexity level (simple, average or complex).We have Considered the complexity level as simple.

- **F**$_i$(i=1 to 14) = "Complexity Adjustment Values" based on responses to a set of 14 questions; each answered using a scale ranging from 0 to 5:

The *Fi (i = 1 to 14) are value adjustment factors (VAF) based on responses to the* following questions :

1. Does the system require reliable backup and recovery?

ANS.5

2. Are specialized data communications required to transfer information to or from the application?

ANS. 2

3. Are there distributed processing functions?
ANS. 3

4. Is performance critical?

ANS. 5

5. Will the system run in an existing, heavily utilized operational environment?

ANS. 4

6. Does the system require online data entry?

ANS. 5

7. Does the online data entry require the input transaction to be built over multiple screens or operations?

ANS. 3

8. Are the ILFs updated online?

ANS. 3

9. Are the inputs, outputs, files, or inquiries complex?

ANS. 1

10. Is the internal processing complex?

ANS. 3

11. Is the code designed to be reusable?

ANS. 3

12. Are conversion and installation included in the design?

ANS. 4

13. Is the system designed for multiple installations in different organizations?

ANS. 3

14. Is the application designed to facilitate change and ease of use by the user?

ANS. 5

Total: 49

Once this much data is collected now we head on to Function points:-
Formula:-

$$\mathbf{FP = Count\_Total \times [0.65 + 0.01 \times \sum (\mathit{Fi})]}$$

Where, Count_Total=80

$\sum(Fi) = 49$

Thus FP is = 80 * [0.65 + (0.01 * 49)]

= **91.2**

SCHEDULING

Scheduling of a software project does not differ greatly from scheduling of any multitask engineering effort. Therefore, generalized project scheduling tools and techniques can be applied with little modification for software projects.

Program evaluation and review technique (PERT) and the critical path method (CPM) are two project scheduling methods that can be applied to software development. Both techniques are driven by information already developed in earlier project planning activities: estimates of effort, a decomposition of the product function, the selection of the appropriate process model and task set, and decomposition of the tasks that are selected.

Interdependencies among tasks may be defined using a task network. Tasks, sometimes called the project work breakdown structure (WBS), are defined for the product as a whole or for individual functions.

Both PERT and CPM provide quantitative tools that allow us to (1) determine the critical path—the chain of tasks that determines the duration of the project, (2) establish “most likely” time estimates for individual tasks by applying statistical models, and

(3) Calculate “boundary times” that define a time “window” for a particular task.

Schedule table

WORK TASKS	SCHEDULE
1. IDENTIFY NEEDS AND BENEFITS	
Meet with concerned members	Week 1
Identify needs and project constraints	Week 1
Establish problem statement	Week 2
Milestone	Week 2
2. REQUIREMENT ANALYSIS	
Detailed discussion of the project	Week 3
Creating Data flow Diagram	Week 4
Data Dictionary	Week 5
Milestone	Week 5
3. PROJECT MANAGEMENT	
Computing F.P. and Effort	Week 5
Schedule table	Week 6
Risk table	Week 7
Timeline Chart	Week 8

Milestone	Week 8
4. DESIGN ENGINEERING	
Architectural Design	Week 8
Data Design	Week 9
Pseudo Code	Week 9
Milestone	Week 10
5. TESTING	Week 12

RISK ANALYSIS

A table provides a project manager with a simple technique for risk production. A risk table is sorted by probability and impact to rank risks. A project team begins by listing all risks in the 1st column of the table. This can be accomplished with the help of the risk item checklist referenced. Each risk is categorized in the 2nd column. The probability of occurrence of each risk is entered in the next column of the table.
Next, the impact of each risk is assessed. Each risk component is assessed using the characterization presented and an impact category is determined. The categories for each of the four risk components-performance, support, cost and schedule-are averaged to determine an overall impact value.

Once the first four columns of the risk table have been completed, the table is sorted by probability and by impact. High-probability, high-impact, risk-impact risks percolate to the top of the table and low-probability risks drop to the bottom.

RISK TABLE

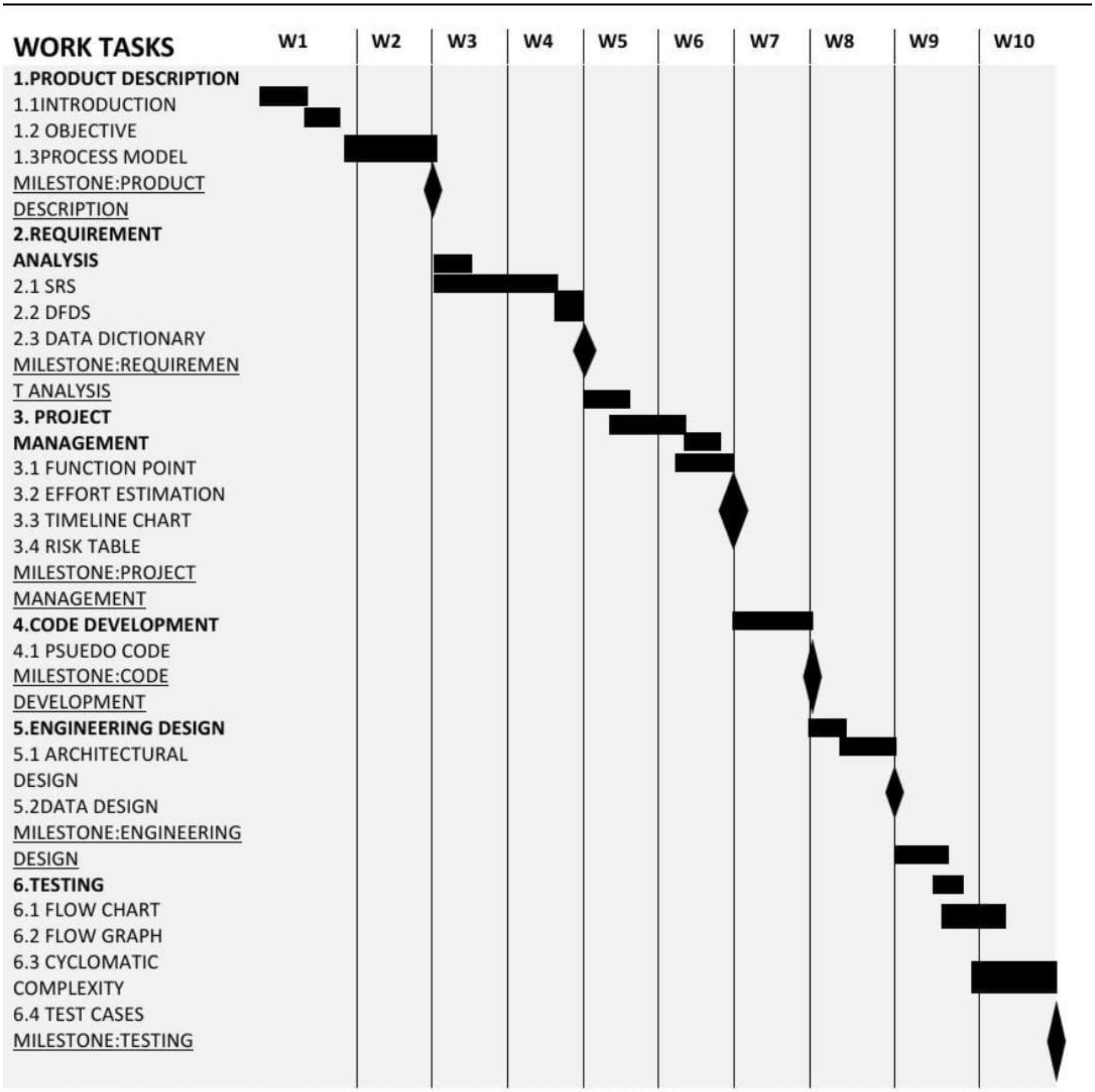
S.NO	RISK	CATEGORY	PROBABILITY(P)	IMPACT(I)	EXPOSURE E=P*I	RMMM PLAN
1	SOME TEAM MEMBERS LEAVE THE PROJECT IN BETWEEN	TECHNICAL RISK	20%	2	0.4	USE BACKUP STAFFS WHO KNOWS WHAT WAS GOING ON PROJECT
2	DELIEVERY DEADLINE TIGHTENED	PROJECT RISK	30%	1	0.3	TEAM MAY USE EXTRA MEMBERS TO DO JOB ON SCHEDULED TIME
3	LOSING OF ALL PROJECT DATA THIS MAY HAPPEN DUE TO HARD DISK FAILURE	PROJECT RISK	20%	2	O.2	CARRY OUT BACKUP OF ESSENTIAL DATABASES,SOURCE CODE ETC.
4	TEAM DISTENSION/LACK OF COHESION	PROJECT RISK	10%	3	0.3	WE MAKE SOME RULES ON HOW WE CONSULT EACH OTHER

Gantt Chart

When creating a software project schedule, you begin with a set of tasks (the work breakdown structure). If automated tools are used, the work breakdown is input as a task network or task outline. Effort, duration, and start date are then input for each task. In addition, tasks may be assigned to specific individuals.

As a consequence of this input, a time-line chart, also called a Gantt chart, is generated. A time-line chart can be developed for the entire project. Alternatively, separate charts can be developed for each project function or for each individual 0working on the project.

The following figure illustrates the format of a time-line chart. It depicts a part of a software project schedule that emphasizes the concept scoping task for a word-processing (WP) software product. All project tasks (for concept scoping) are listed in the left-hand column. The horizontal bars indicate the duration of each task. When multiple bars occur at the same time on the calendar, task concurrency is implied. The diamonds indicate milestones.


WORK TASKS
W1
W2
W3
W4
W5
W6
W7
W8
W9
W10
1.PRODUCT DESCRIPTION
1.1INTRODUCTION
1.2 OBJECTIVE
1.3PROCESS MODEL
MILESTONE:PRODUCT DESCRIPTION
2.REQUIREMENT ANALYSIS
2.1 SRS
2.2 DFDS
2.3 DATA DICTIONARY
MILESTONE:REQUIREMENT ANALYSIS
3. PROJECT MANAGEMENT
3.1 FUNCTION POINT
3.2 EFFORT ESTIMATION
3.3 TIMELINE CHART
3.4 RISK TABLE
MILESTONE:PROJECT MANAGEMENT
4.CODE DEVELOPMENT
4.1 PSUEDO CODE
MILESTONE:CODE DEVELOPMENT
5.ENGINEERING DESIGN
5.1 ARCHITECTURAL DESIGN
5.2DATA DESIGN
MILESTONE:ENGINEERING DESIGN
6.TESTING
6.1 FLOW CHART
6.2 FLOW GRAPH
6.3 CYCLOMATIC COMPLEXITY
6.4 TEST CASES
MILESTONE:TESTING

Architectural design

Requirements of the software should be transformed into an architecture that describes the software's top-level structure and identifies its components. This is accomplished through architectural design (also called **system design),** which acts as a preliminary 'blueprint' from which software can be developed. **IEEE** defines architectural design as 'the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system.' This framework is established by examining the software requirements document and designing a model for providing implementation details. These details are used to specify the components of the system along with their inputs, outputs, functions, and the interaction between them. An architectural design performs the following functions.

It defines an abstraction level at which the designers can specify the functional and performance behaviour of the system.

1. It acts as a guideline for enhancing the system (whenever required) by describing those features of the system that can be modified easily without affecting the system integrity.
2. It evaluates all top-level designs.
3. It develops and documents top-level design for the external and internal interfaces.
4. It develops preliminary versions of user documentation.
5. It defines and documents preliminary test requirements and the schedule for software integration.

Though the architectural design is the responsibility of developers, some other people like user representatives, systems engineers, hardware engineers, and operations personnel are also involved. All these stakeholders must also be consulted while reviewing the architectural design in order to minimize the risks and errors.

Architectural Design Representation

Architectural design can be represented using the following models.

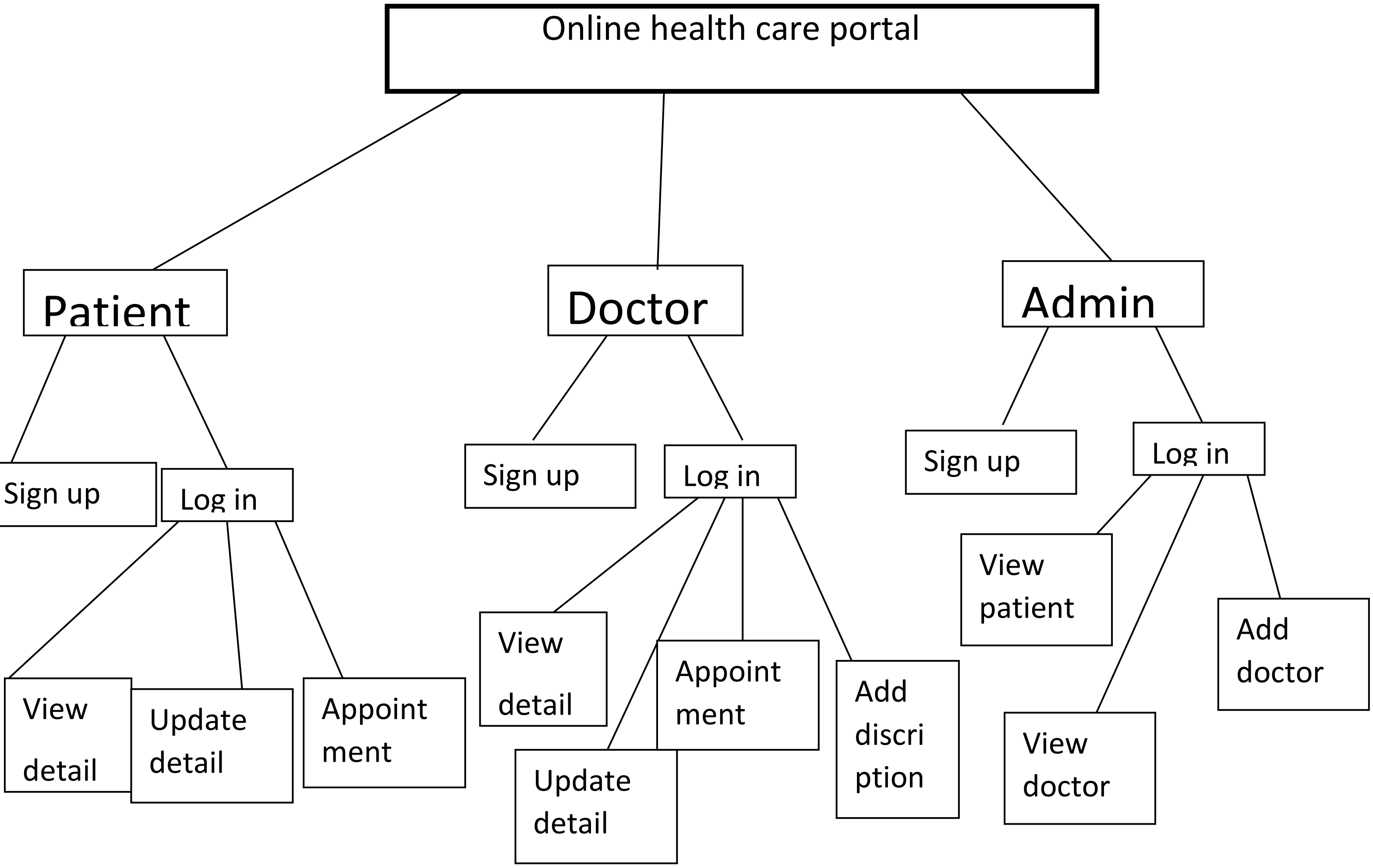
Structural model: Illustrates architecture as an ordered collection of program components

Dynamic model: Specifies the behavioral aspect of the software architecture and indicates how the structure or system configuration changes as the function changes due to change in the external environment.

Process model: Focuses on the design of the business or technical process, which must be implemented in the system

Functional model: Represents the functional hierarchy of a system

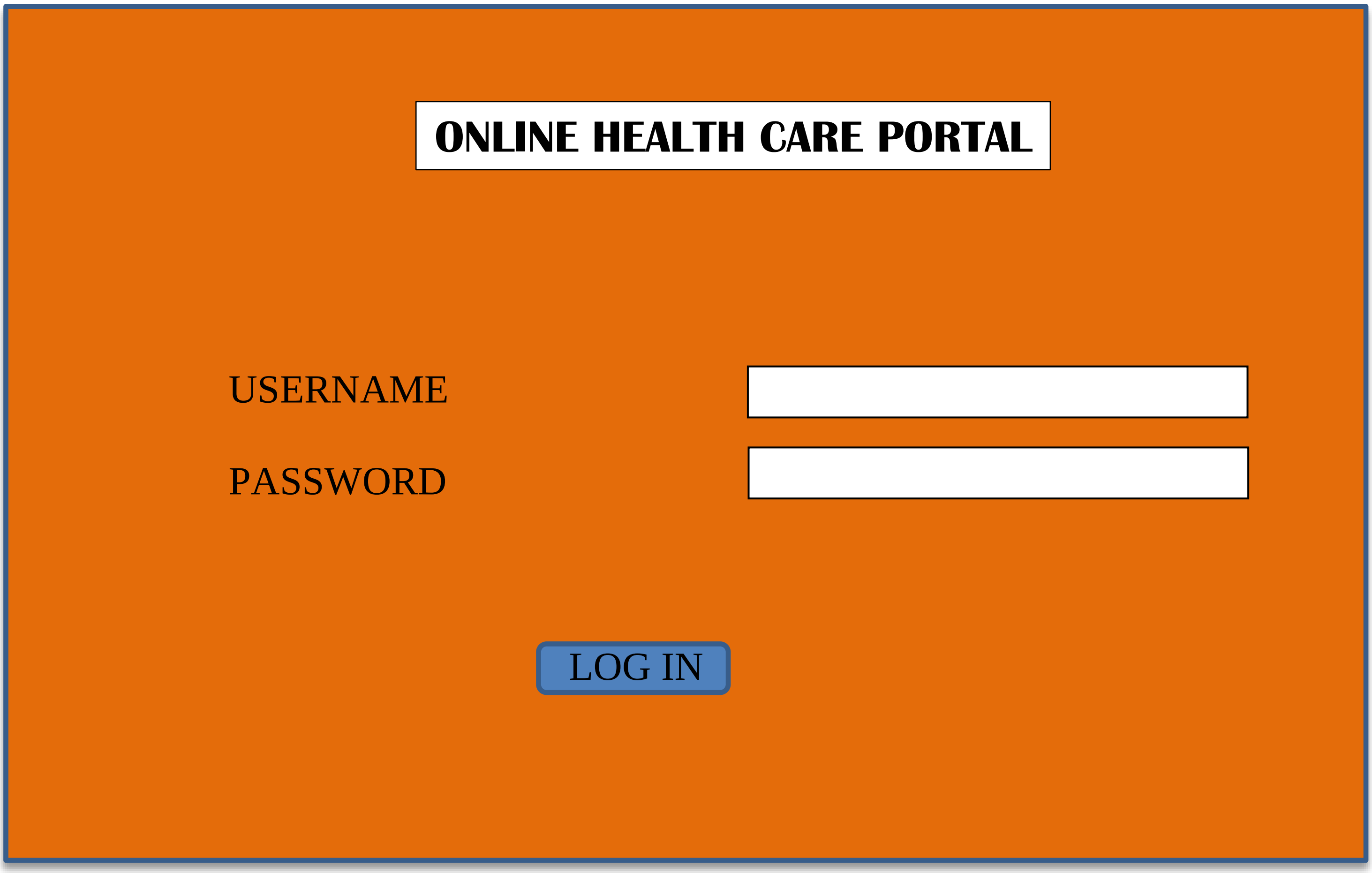
Framework model: Attempts to identify repeatable architectural design patterns encountered in similar types of application. This leads to an increase in the level of abstraction.



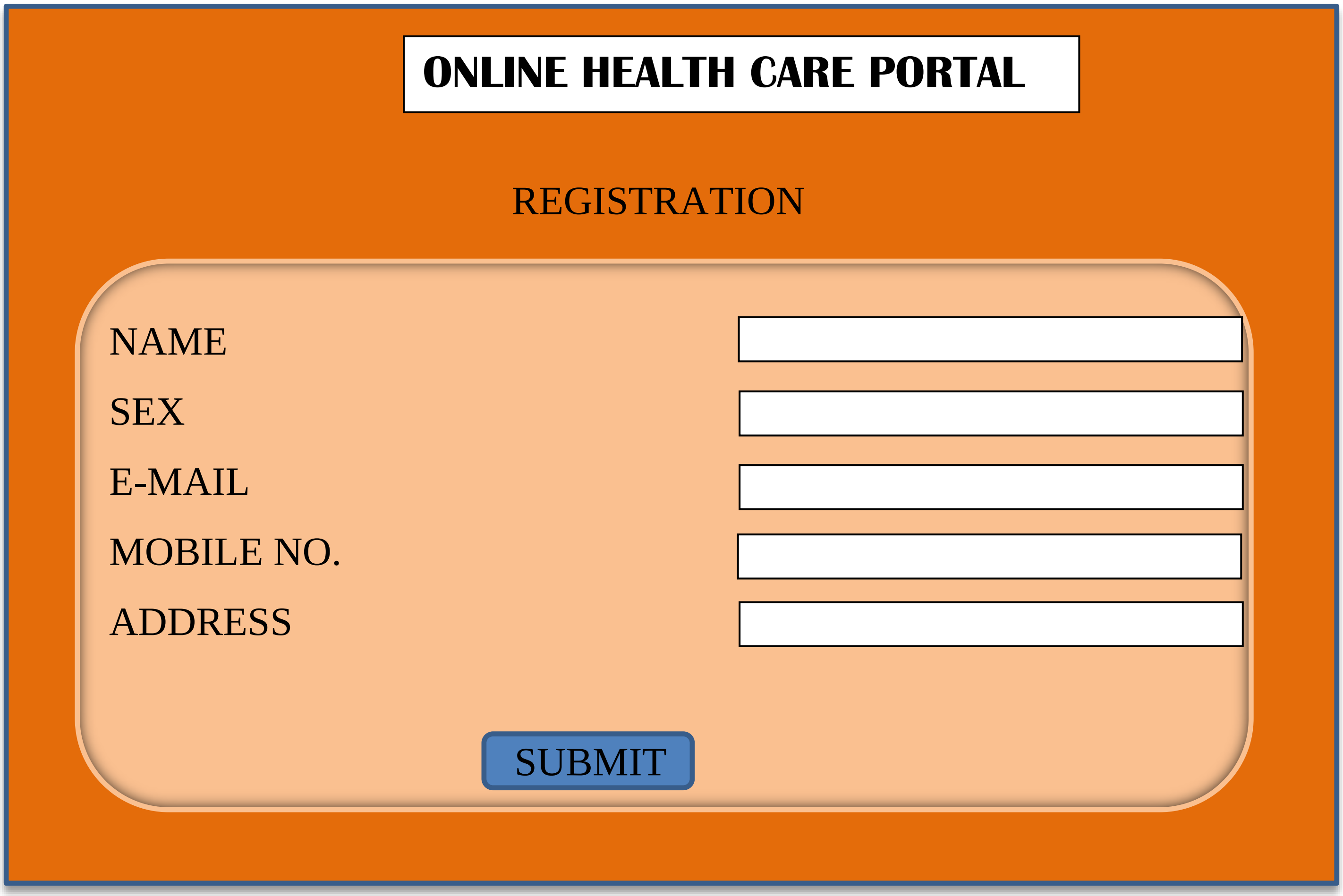
IMPLEMENTATION

MODULES SCREEN

LOG IN PAGE



REGISTRATION PAGE



ONLINE HEALTH CARE PORTAL

REGISTRATION

NAME

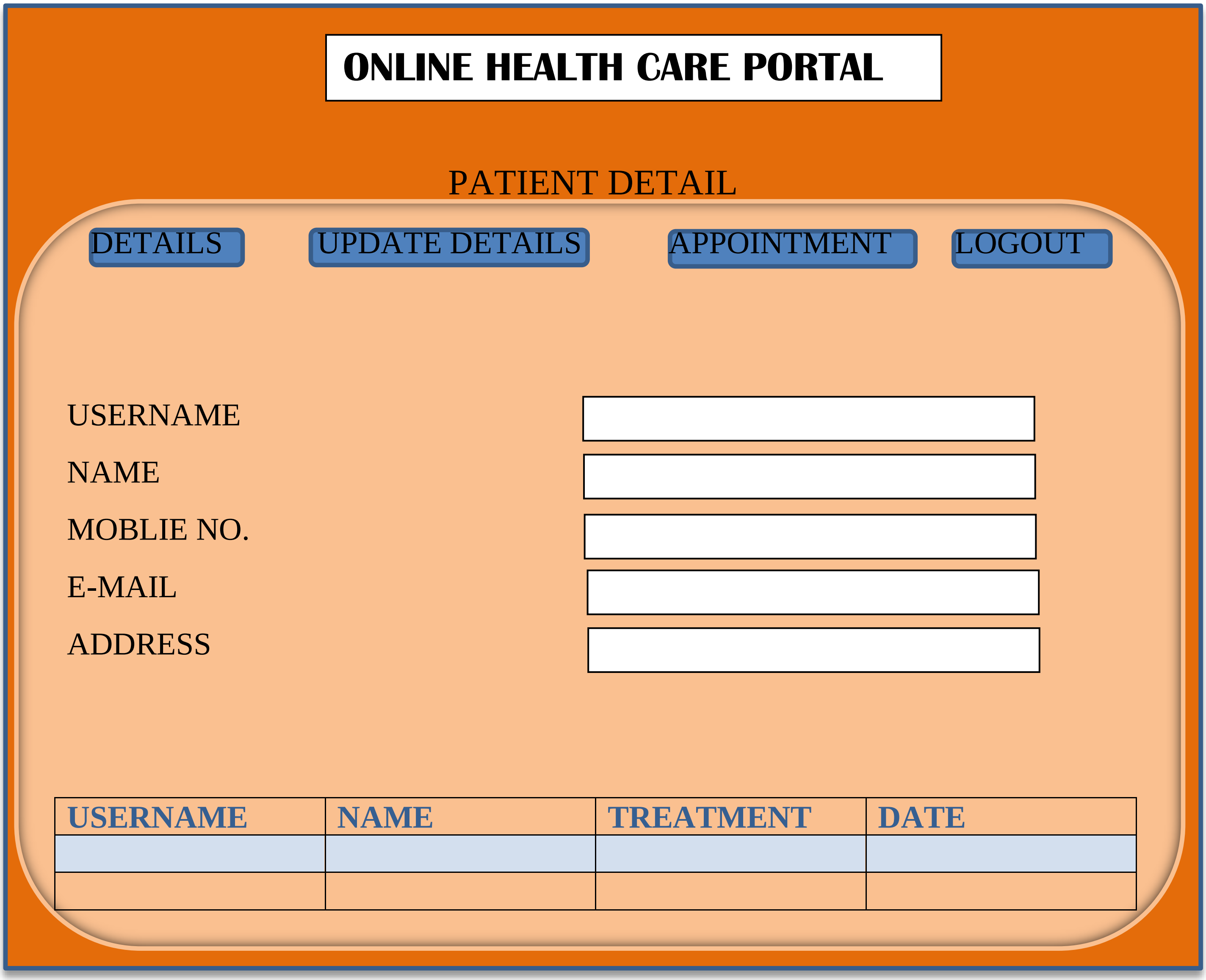
SEX

E-MAIL

MOBILE NO.

ADDRESS

PATIENT PAGE



ONLINE HEALTH CARE PORTAL

PATIENT DETAIL

USERNAME

NAME

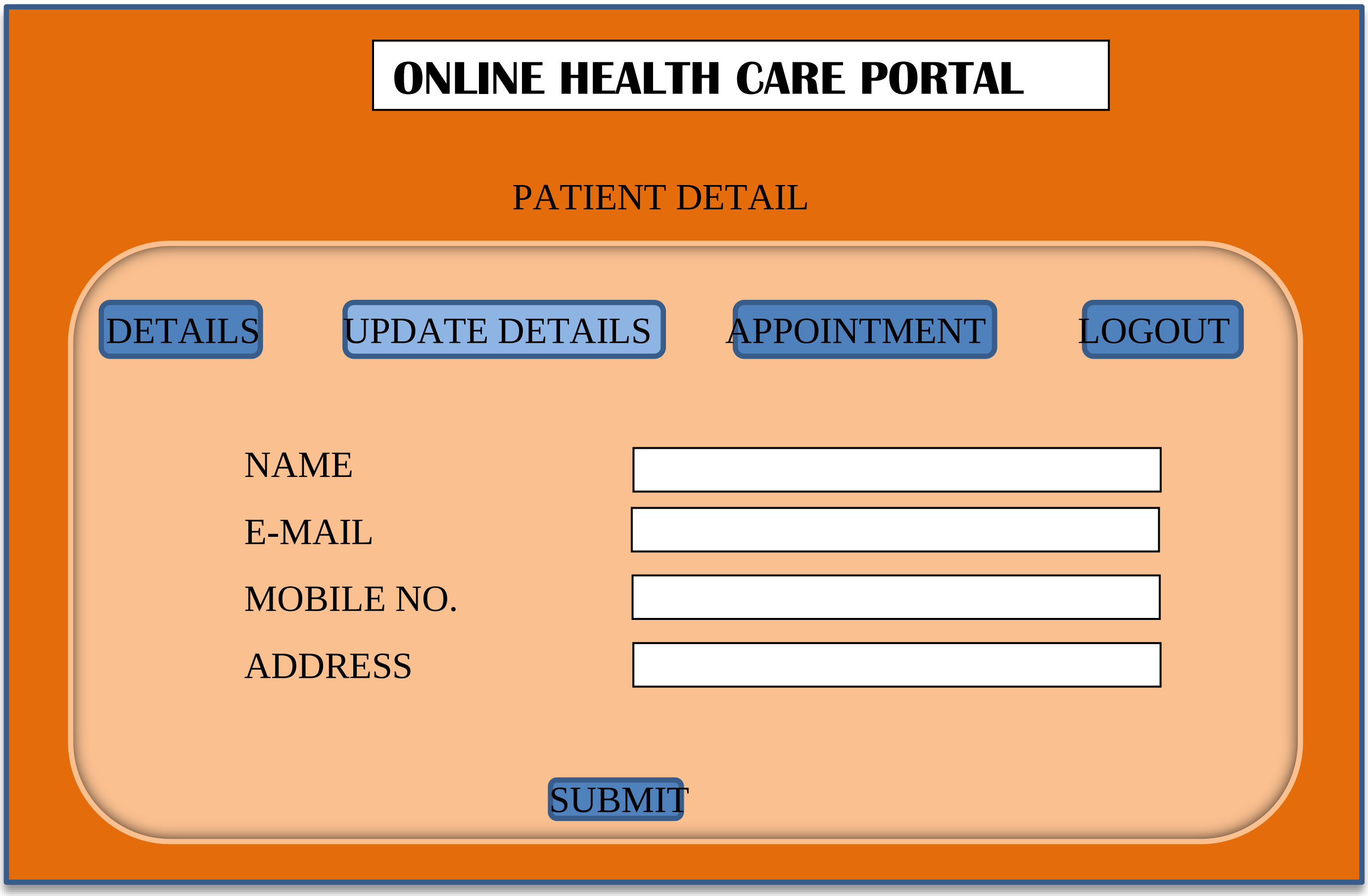
MOBLIE NO.

E-MAIL

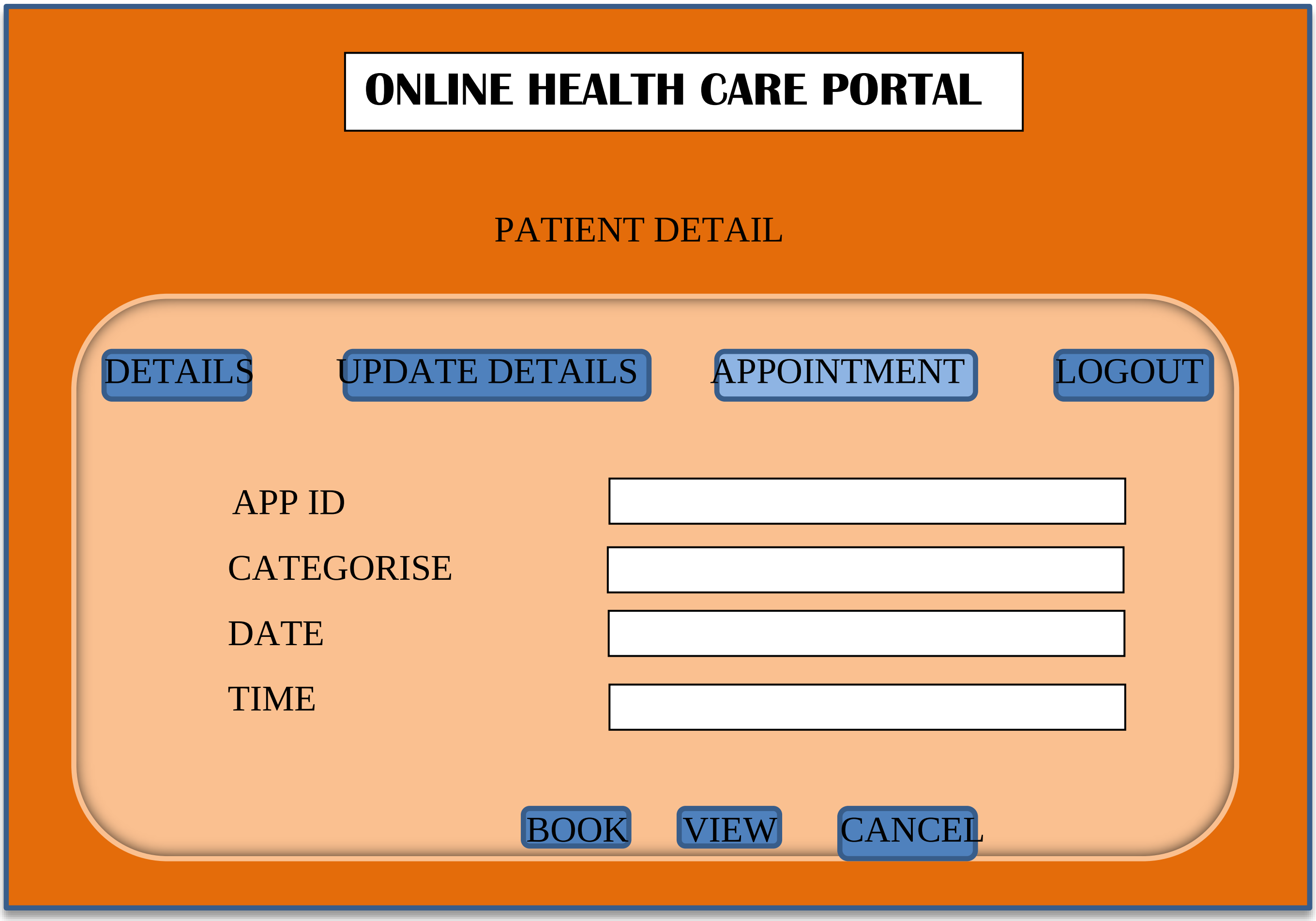
ADDRESS

USERNAME	NAME	TREATMENT	DATE

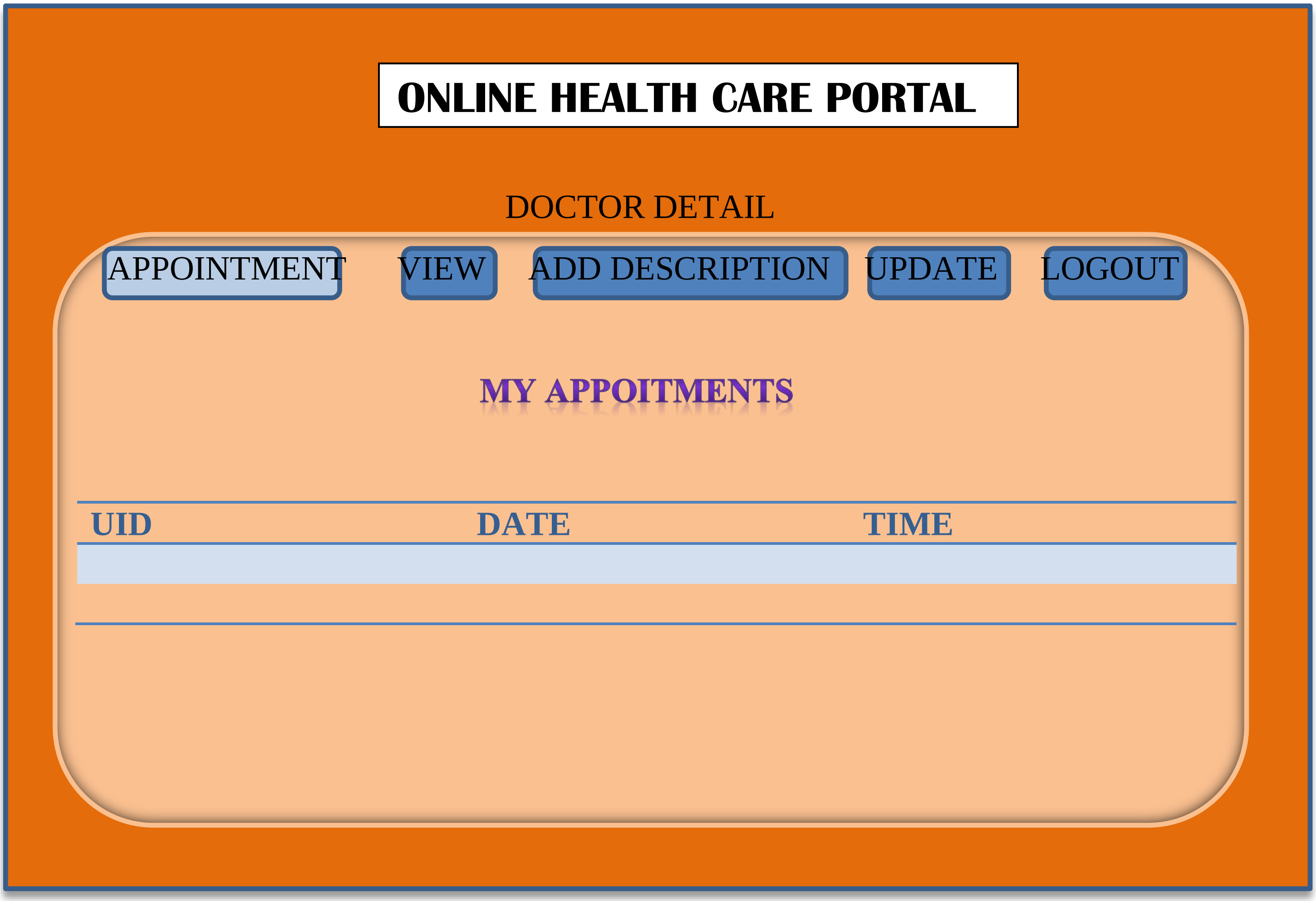
UPDATE DETAIL



APPOITMENT PAGE



DOCTOR APPOINTMENT



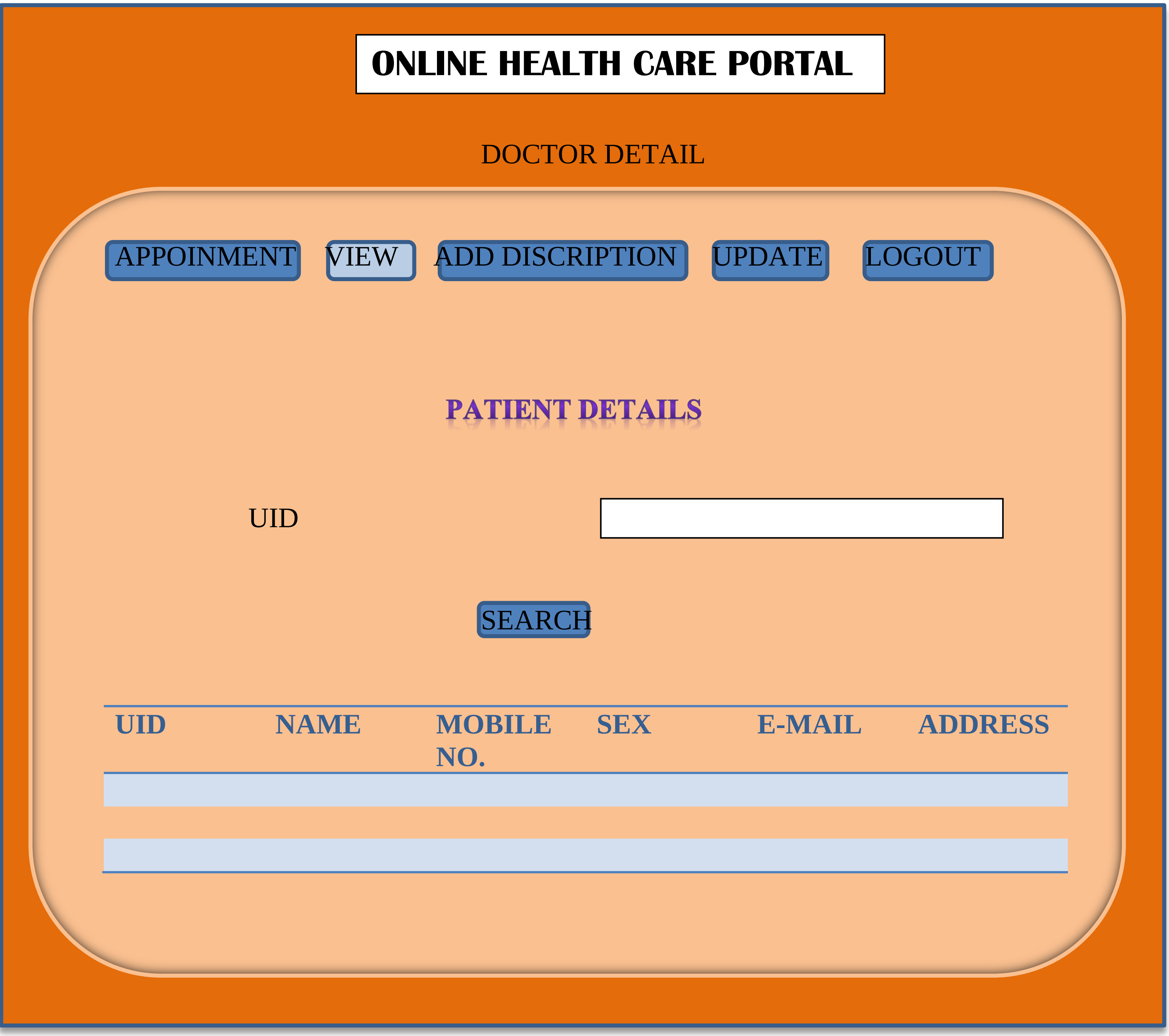
ONLINE HEALTH CARE PORTAL

DOCTOR DETAIL

MY APPOITMENTS

UID	DATE	TIME

DOCTOR VIEW PATIENT



ONLINE HEALTH CARE PORTAL

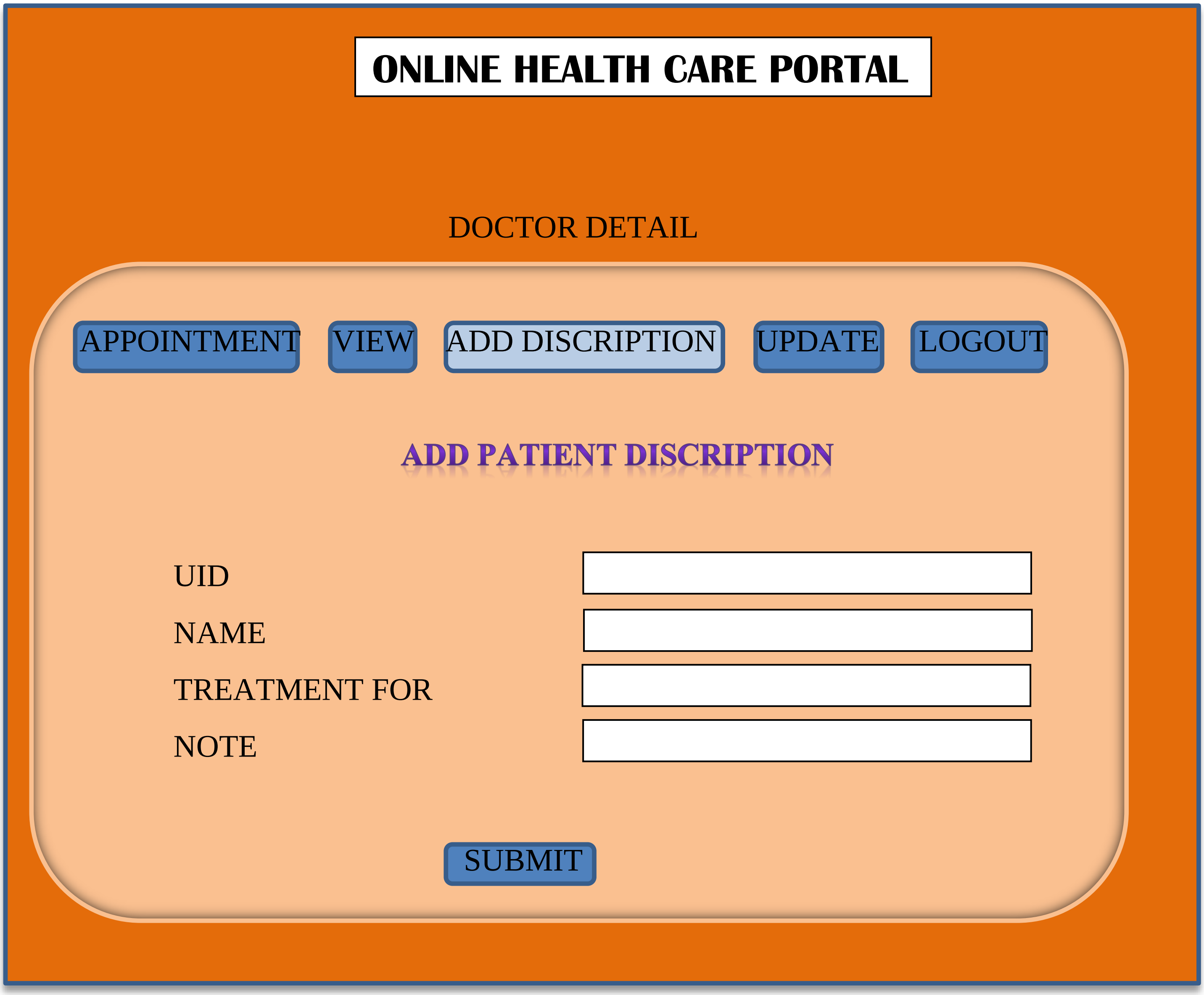
DOCTOR DETAIL

PATIENT DETAILS

UID

UID	NAME	MOBILE NO.	SEX	E-MAIL	ADDRESS

DOCTOR ADD DISCRIPTION



ONLINE HEALTH CARE PORTAL

DOCTOR DETAIL

ADD PATIENT DISCRIPTION

UID

NAME

TREATMENT FOR

NOTE

Testing

- Testing is a process of executing a program with the intent of finding an error. A good test case is one that has high probability of finding -undiscovered error.
- Software testing is a critical element of software quality assurance and represent the ultimate review of specification, design, coding.
- The purpose of product testing is to verify and validate the various work products viz. units, integrated unit, final product to ensure that they meet their requirements.

There are two types of test case design methods-

- Black Box Testing
- White Box Testing

Black Box Testing

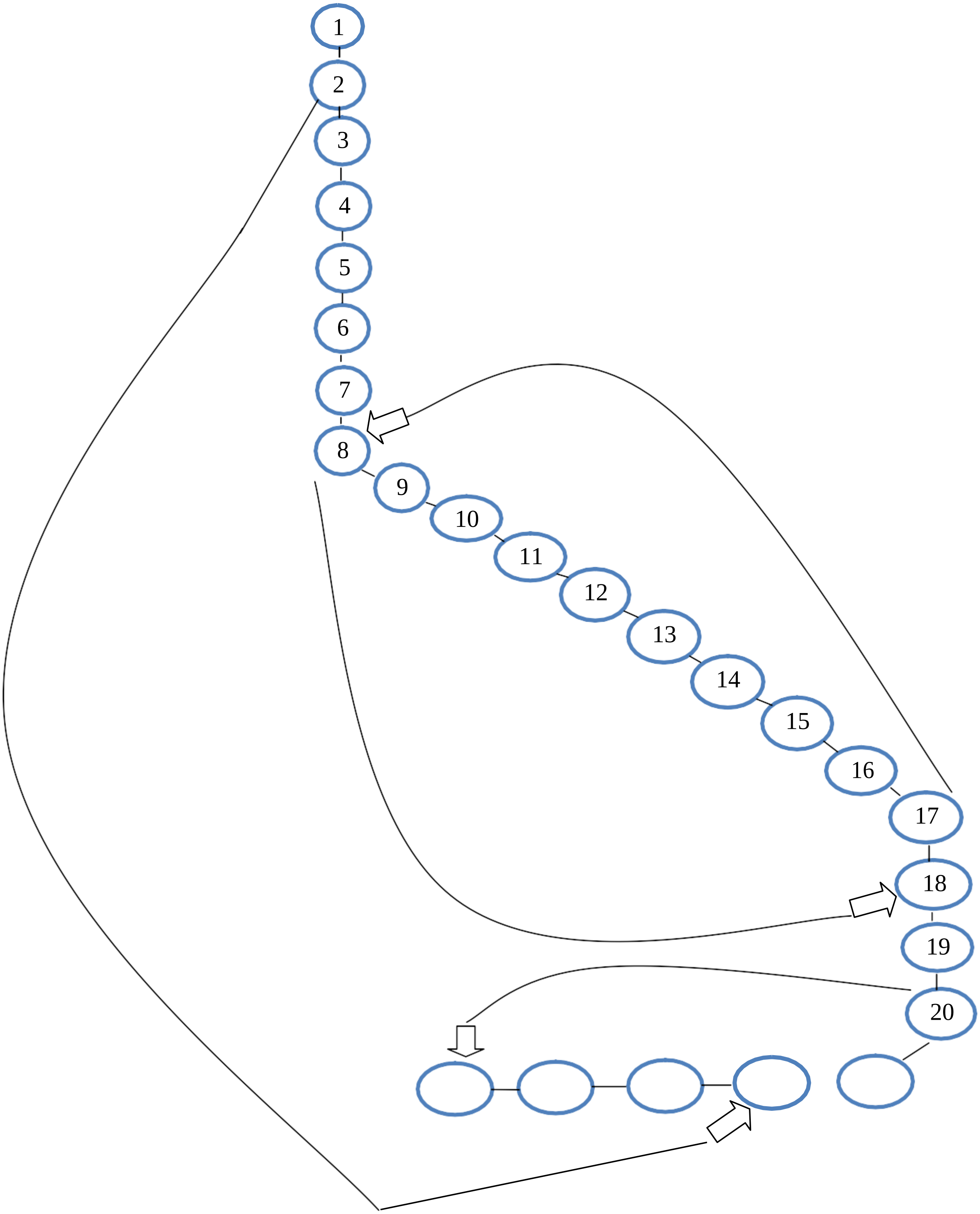
- Black Box testing is also called as functional testing.
- Black Box Testing is a test case design method that focuses on the functional requirements of the software that is it enables the software engineer to derive set of inputs conditions that fully exercise all functional requirements for a program.
- Test the artifacts from the external point of view.
- Specifications are used to test data that is what type of input should be given to the unit or module should be specified.
- We can check the functionality on the basis of the output generated and the input, not looking at the internal coding.
- It attempts to find errors in the following categories
  - Incorrect or missing functions
  - Interface errors
  - Errors in data structure or External database access
  - Behaviour or performance error
  - Initialisation and termination errors

White Box Testing

- It is also called as glass box testing.
- White Box testing is test case design method that uses the control structure of the procedural design to derive test cases.
- Using White Box Testing method, the software engineer can derive test cases that
    - Guarantee that all independent paths within a module have been exercised at least once.
        - Exercise all logical decisions on their true and false sides.
    - Execute all loops at their boundaries and within their operational bounds.
    - Exercise internal data structures ensure their validity.
- Test the artifacts from the internal point of view.
- It cannot detect absence of features.
- For security purposes the E-mail of user is required in case he/she forgets his/her password and wants to retrieve that.

BASIS PATH TESTING FOR DOCTOR VIEW PATIENT MODULE

```
1.      DefaultTabelmodel model = (DefaultTableModel)tablename.getModel1);

2.      try{
3.      class.forname("java.sql.Driver);

4.      Connection con= DriverManager.getConnection("jdbc:mysql://localhost/test","root","xyz");

5.      Statement stmt= con.createstatement();

6.      string query= "select * from Appointment where vid="+vid+f.getText+";"

7.      Result set rs= stmt.executeQuery(query);

8.      while(rs.next()){
9.      string uid=rs.getstring("uid ");

10.     string name= rs.getstring("Name");

11.     string address= rs.getstring("Address");

12.     string mobile= rs.getstring("Mobile");

13.     string Email= rs.getstring("Email");

14.     string sex=rs.getstring("Sex");

15.     string Age=rs.getstring("Age");

16.     model.addRpw(newobject[ ]{uid,name,address,mobile,email,sex,age})

17,     }

18.     rs.close();

19.     stmt.close();

20.     con.close();

21,     }
22.     catch(Exception()) {

23..    .       JoptionPane.showmessageDialog(null,"Error in Connectivity");
.
24.     }
25.         }
```


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

1) CYCLOMATIC COMPLEXITY

CYCLOMATIC COMPLEXITY=E-N+2

E=NO. OF EDGES IN PROGRAM FLOW GRAPH

N=NO.OF NODES IN PROGRAM FLOW GRAPH

HERE E=27 N=25

SO CYCLOMATIC COMPLEXITY=4

Hence, there will be 4 independent paths.

1) INDEPENDENT PATHS

PATH A: 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-25

PATH B: 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-8-18-19-20-25

PATH C: 1-2-3-4-5-6-7-8-18-19-20-25

PATH D: 1-2-22-23-24-25

CONCLUSION

The entire project has been developed and deployed as per the requirements stated by the user. It is found to be bug free as per the testing standards that are implemented.

The whole system's activities are divided into two major parts like User and admin.

There are also few features which can be integrated with this system to make it more flexible. Below list shows the future points to be consider:

- Getting the current status of patient.
- Including a different module for pharmacy

Finally, we like to conclude that we put all our efforts throughout the development of our project and tried to fulfill most of the requirements of the user.