PROJECT REPORT

ON

"HOSPITAL MANAGEMENT SYSTEM"

AT

ECIL-ECIT

(ELECTRONIC CORPORATION OF INDIA LIMITED)

BY

V. LOHYA SUJITH

UNDER THE ESTEEMED GUIDANCE OF

V.Y.BHARADWAJ, M.Tech

IN PARTIAL FULFILLMENT FOR THE AWARD OF THE COURSE

B-TECH-2017-21

DECLARATION

I **V.LOHYA SUJITH** hereby declare that the project entitled **“HOSPITAL MANAGEMENT SYSTEM”** has been successfully completed by us at **ECIL-ECIT.**

V. LOHYA SUJITH

DATE:

PLACE:

ACKNOWLEDGEMENT

I take this opportunity to express our deep gratitude to all the people who cooperated in various works during our project work and played a big role in completing this project successfully.

I also extend our gratitude to **Mr.BHARADWAJ** project guide for his valuable guidance throughout this project to achieve our goal.

At last but not the least, we would like to thank the entire respondents for extending their helping all circumstance

V. LOHYA SUJITH

CONTENTS

1. INTRODUCTION:

Hospital Management System helps to maintain total patients in the Hospital.

Patient records management system is a comprehensive patient care related clinical information system meant for doctors to enable quick review of previous medical history of patients and provide better quality treatment to the patients.

1.1 ABSTRACT:

Project Analysis:

This application consists of following modules

1. Registration Module
2. Out Patient Scheduling Module
3. In Patient Monitoring Module
4. Patient Feedback
5. Admin Module

Module I: Registration module consists of sub modules namely:

- Maintains New Patient Registration.
- Advance Payments, Appointment Facility from Doctors, Admission, Discharge (or) Transfer Details.

Module II: Out Patient scheduling module consists of sub modules namely:

- Maintains Daycare patient schedule and Referal patient schedule.

Module III: In-patient monitoring module consists of sub modules namely:

- Maintains Casualty patient schedule, patient health status.

Module IV: Patient Feedback module consists of sub modules namely:

- Maintains feedback from the patient.

Module V: Admin module consists of sub modules namely:

- Provide access rights.

2. ORGAINIZATION PROFILE:

Electronics Corporation of India Limited

1914 - 2003

"Let us work up the members of national pride latent in all of us and build up our morale so that we can confidently aim high and achieve greater goals"

Dr. A.S.Rao,
Founder MD of ECIL

ECIL was setup under the Department of Atomic Energy on 11th April, 1967 with a view to generate a strong indigenous capability in the field of professional grade electronics. The initial accent was on total self-reliance and **ECIL** was engaged in the Design, Development, Manufacture and Marketing of several products with emphasis on three technology lines viz. Computers, Control Systems and Communications. Over the years, **ECIL** pioneered the development of various complex electronics products without any external technological help and scored several 'firsts' in these fields prominent among them being country's

- First Digital Computer
- First Solid State TV
- First Control & Instrumentation of Nuclear Power Plants
- First Earth Station Antenna
- First Computerized Operator Information System
- First Radiation Monitoring & Detection Systems
- First Automatic Message Switching Systems
- First Operation & Maintenance Center For E-108 Exchange
- First Programmable Logic Controller
- First Solid State Cockpit Voice Recorder
- First Electronic Voting Machines

The company played a very significant role in the training and growth of high caliber technical and managerial manpower especially in the fields of **Computers and Information Technology**. Though the initial thrust was on meeting the **Control & Instrumentation** requirements of the Nuclear Power Program, the expanded scope of self-reliance pursued by ECIL enabled the company to develop various products to cater to the needs of Defense, Civil Aviation, Information & Broadcasting, Telecommunications, Insurance, Banking, Police, and Para-Military Forces, Oil & Gas, Power, Space Education, Health, Agriculture, Steel and Coal sectors and various user departments in the Government domain. ECIL thus evolved as a multi-product company serving multiple sectors of Indian economy with emphasis on import of country substitution and development of products & services that are of economic and strategic significance to the country.

OBJECTIVES

- To continue services to the country's needs for the peaceful uses Atomic Energy. Special and Strategic requirements of Defense and Space, Electronics Security Systems and Support for Civil Aviation sector.
- To establish newer technology products such as Container Scanning Systems and Explosive Detectors.
- To explore new avenues of business and work for growth in strategic sectors in addition to working for realizing technological solutions for the benefit of society in areas like Agriculture, Education, Health, Power, Transportation, Food, Disaster Management etc.
- To progressively improve shareholder value of the company.
- To strengthen the technology base, enhance skill base and ensure succession planning in the company.
- To re-engineer the company to become nationally and internationally competitive by paying particular attention to delivery, cost and quality in all its activities.
- To consciously work for finding export markets for the company's products.

VISION:-Contribute to the country in achieving ***self reliance in strategic electronics.***

MISSION:- ECIL's mission is to consolidate its status as a valued national asset in the area of strategic electronics with specific focus on ***Atomic Energy, Defence, Security*** *and such* ***critical sectors of strategic national importance.***

3. PROBLEM DEFINITION:

In the existing system there are following disadvantages:

- The existing is a manual process.
- Here the patient’s registration is entered in the registers.
- Lots of paper work is done, which may lead to loss of information.
- The doctors may forget their appointments with the patient’s.
- There is no separate user to interact with database.
- Registration of patient’s process is manual.
- Difficult to maintain details of all staff working in the hospital and also patient’s details at the same time.

So in order to reduce all the draw backs of manual system we are going for automated system.

4. SYSTEM ANALYSIS

4.1 EXISTING SYSTEM:

The existing system is a manual process. Here the patient's registration is entered in the registers. Lots of paper work is done, which may lead to loss of information. The doctors may forget their appointments with the patients.

DISADVANTAGES:

- Doctors have to check the reports of the patients to know their previous history which consumes a lot of time.
- They have to come to the receptionist to check their daily appointments
- Registration of patient's process is manual.
- The patient does have to wait for two or three days to get their lab reports and their total bill details.
- Difficult to maintain details of all staff working in the hospital and also patient's details at the same time.

So in order to reduce all the draw backs of manual system we are going for automated system.

4.2 PROPOSED SYSTEM:

To overcome the difficulties in our existing system we use this system. The advantages of this system are:

ADVANTAGES:

- HMS helps to keep a track of the patient's registering in a hospital
- This system supports the accessing of the patient's previous case sheets and history.
- Helps the doctors to check their appointments for the day.
- Doctors will also use the system to keep track of the patient assigned to them.
- Flexible for future enhancements.
- User friendly interfaces for data entry and easy to navigate.
- Make our work easier.
- Searching of data is easy.
- High security is provided.

5. FEASIBILITY STUDY

A feasibility study is an evaluation of a proposal designed to determine the difficulty in carrying out a designated task. Generally, a feasibility study precedes technical development and project implementation. In other words, a feasibility study is an evaluation or analysis of the potential impact of a proposed project.

5.1 TYPES OF FEASIBILITY:

- **TECHNOLOGY AND SYSTEM FEASIBILITY:**

  The assessment is based on an outline design of system requirements in terms of Input, Processes, Output, Fields, Programs, and Procedures. This can be quantified in terms of volumes of data, trends, frequency of updating, etc. in order to estimate whether the new system will perform adequately or not. Technological feasibility is carried out to determine whether the company has the capability, in terms of software, hardware, personnel and expertise, to handle the completion of the project.

- **ECONOMIC FEASIBILITY:**

  Economic analysis is the most frequently used method for evaluating the effectiveness of a new system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. An entrepreneur must accurately weigh the cost versus benefits before taking an action.

- **LEGAL FEASIBILITY:**

  Determines whether the proposed system conflicts with legal requirements, e.g. a data processing system must comply with the local data protection acts.

➢ OPERATIONAL FEASIBILITY:

Is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

➢ SCHEDULE FEASIBILITY:

A project will fail if it takes too long to be completed before it is useful. Typically this means estimating how long the system will take to develop, and if it can be completed in a given time period using some methods like payback period. Schedule feasibility is a measure of how reasonable the project timetable is.

OUTPUT:

The feasibility study outputs the feasibility study report, a report detailing the evaluation criteria, the study findings, and the recommendations.

6. PROJECT OVERVIEW

INTRODUCTION:

This HMS system is a self-contained system that manages activities of the hospital such operations scheduling, personal management, and administrative issues.

Patient Management System helps to maintain total patients in the Hospital. Patient records management system is a comprehensive patient care related clinical information system meant for doctors to enable quick review of previous medical history of patients and provide better quality treatment to the patients.

After login to the system there are three different actors’ administrator, doctor, receptionist.

ADMINISTRATOR:

After login to the system if the user is administrator then they are allowed to perform certain activities:

- Adding & deleting information & staff.
- Viewing the staff details.
- Updating the staff.
- Viewing the patient details
- Updating and deleting the patient details.
- Viewing the patient status etc...

DOCTOR:

If the user is doctor then they may perform any of activities like:

- Viewing the outpatient details
- Viewing the inpatient details
- Viewing the patient health status
- Viewing the patient discharge form

RECEPTIONIST:

After login to the system if the user is receptionist then they are allowed to perform certain activities:

- Insert all the details in the registration form
- Insert all the details of the patients
- Insert all the details regarding patient status
- Insert all the details in the patient discharge form

- Viewing the outpatient details
- Viewing the inpatient details
- Viewing the patient health status
- Viewing the patient discharge form

7. DEFINITIONS, ACRONYMS, ABBREVATIONS:

HMS – Hospital Management System

Data Base – Collection of information in a structured form

Login ID – a user identification number to enter the system

Password – a word that enables one to gain admission into the system

Web-based Application – an application that runs on the internet

Windows – an operating system produce by Microsoft Corporation that is used to operate the computer using GUI

Oracle – a query language to interrogate the system

PID – Patient Identification Number

GUI – Graphical User Interface

DBMS (Database Management System):

DatabaseManagement system is used to store data in the form of records (or) tables.

HTTP- Hyper Text Transfer Protocol

HTML- Hyper Text Mark-up Language

HS- Host System

JSP- Java Server Pages

8. SYTEM REQUIREMENTS

8.1 SOFTWARE REQUIREMENTS:

OPERATING SYSTEM	Windows XP Service Pack2
TECHNOLOGY	JAVA/J2EE(JDBC,JSP,HTML,JSCRIPT)
WEB SERVER	TOMCAT 5.5
DATABASE	ORACLE 10g
SOFTWARE	JDK 1.6

8.2 HARDWARE REQUIREMENTS:

HARDWARE	Pentium based systems with a minimum of p4
RAM	512MB(minimum)

9. TECHNOLOGIES

LANGUAGES USED:

In my project, I have chosen *Java* language for developing the code.

JAVA:

Java is a programming language originally developed by James Gosling at Sun Microsystems (which is now a subsidiary of Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to byte code (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is general-purpose, concurrent, class-based, and object-oriented, and is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere". Java is considered by many as one of the most influential programming languages of the 20th century, and is widely used from application software to web applications.

The original and reference implementation Java compilers, virtual machines, and class libraries were developed by Sun from 1995. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of its Java technologies under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java and GNU Classpath.

PRINCIPLES:

There were five primary goals in the creation of the Java language: It should be "simple, object oriented, and familiar".

- It should be "robust and secure".
- It should be "architecture neutral and portable".

- It should execute with "high performance".
- It should be "interpreted, threaded, and dynamic.

Performance:

Programs written in Java have a reputation for being slower and requiring more memory than those written in some other languages. However, Java programs' execution speed improved significantly with the introduction of Just-in-time compilation in 1997/1998 for Java 1.1, the addition of language features supporting better code analysis (such as inner classes, StringBuffer class, optional assertions, etc.), and optimizations in the Java Virtual Machine itself, such as Hotspots becoming the default for Sun's JVM in 2000.

Syntax:

The syntax of Java is largely derived from C++. Unlike C++, which combines the syntax for structured, generic, and object-oriented programming, Java was built almost exclusively as an object oriented language. All code is written inside a class and everything is an object, with the exception of the intrinsic data types (ordinal and real numbers, boolean values, and characters), which are not classes for performance reasons.

Java suppresses several features (such as operator overloading and multiple inheritance) for *classes* in order to simplify the language and to prevent possible errors and anti-pattern design.

Java uses similar commenting methods to C++. There are three different styles of comment: a single line style marked with two slashes (//), a multiple line style opened with a slash asterisk (/*) and closed with an asterisk slash (*/) and the Javadoc commenting style opened with a slash and two asterisks (/**) and closed with an asterisk slash (*/). The Javadoc style of commenting allows the user to run the Javadoc executable to compile documentation for the program.

FEATURES OF JAVA:

PLATFORM INDEPENDENT:

The concept of Write-once-run-anywhere (known as the Platform independent) is one of the important key feature of java language that makes java as the most powerful language. Not even a single language is idle to this feature but java is more closer to this feature. The programs written on one platform can run on any platform provided the platform must have the JVM.

SIMPLE:

There are various features that make the java as a simple language. Programs are easy to write and debug because java does not use the pointers explicitly. It is much harder to write the java programs that can crash the system but we cannot say about the other programming languages. Java provides the bug free system due to the strong memory management. It also has the automatic memory de-allocation.

OBJECTORIENTED:

To be an Object Oriented language, any language must follow at least the four characteristics.

- **Inheritance:**

  It is the process of creating the new classes and using the behavior of the existing classes by extending them just to reuse the existing code and adding the additional features as needed.

- **Encapsulation:**

  It is the mechanism of combining the information and providing the abstraction.

- **Polymorphism:**

  As the name suggest one name multiple form, Polymorphism is the way of providing the different functionality by the functions having the same name based on the signatures of the methods.

- **Dynamic binding:**

  Sometimes we don't have the knowledge of objects about their specific types while writing our code. It is the way of providing the maximum functionality to a program about the specific type at runtime. As the languages like Objective C, C++ fulfills the above four characteristics yet they are not fully object oriented languages because they are structured as well as object oriented languages. But in case of java, it is a fully Object Oriented language because object is at the outer most level of data structure in java. No stand alone methods, constants, and variables are there in java. Everything in java is object even the primitive data types can also be converted into object by using the wrapper class.

- **Distributed**:

  The widely used protocols like HTTP and FTP are developed in java. Internet programmers can call functions on these protocols and can get access the files from any remote machine on the internet rather than writing codes on their local system.

- **Portable:**

  The feature Write-once-run-anywhere makes the java language portable provided that the system must have interpreter for the JVM. Java also have the standard data size irrespective of operating system or the processor. These features makes the java as a portable language.

- **Dynamic:**

  While executing the java program the user can get the required files dynamically from a local drive or from a computer thousands of miles away from the user just by connecting with the Internet.

- **Secure:**

  Java does not use memory pointers explicitly. All the programs in java are run under an area known as the sand box. Security manager determines the accessibility options of a class like reading and writing a file to the local disk. Java uses the public key encryption system to allow the java applications to transmit over the internet in the secure encrypted form. The bytecode Verifier checks the classes after loading.

- **Performance:**

Java uses native code usage, and lightweight process called threads. In the beginning interpretation of byte code resulted the performance slow but the advance version of JVM uses the adaptive and just in time compilation technique that improves the performance.

- **Multithreaded:**

As we all know several features of Java like Secure, Robust, Portable, dynamic etc; you will be more delighted to know another feature of Java which is Multithreaded. Java is also a Multithreaded programming language. Multithreading means a single program having different threads executing independently at the same time. Multiple threads execute instructions according to the program code in a process or a program. Multithreading works the similar way as multiple processes run on one computer. Multithreading programming is a very interesting concept in Java. In multithreaded programs not even a single thread disturbs the execution of other thread. Threads are obtained from the pool of available ready to run threads and they run on the system CPUs.

- **Interpreted:**

We all know that Java is an interpreted language as well. With an interpreted language such as Java, programs run directly from the source code. The interpreter program reads the source code and translates it on the fly into computations. Thus, Java as an interpreted language depends on an interpreter program. The versatility of being platform independent makes Java to outshine from other languages. The source code to be written and distributed is platform independent. Another advantage of Java as an interpreted language is its error debugging quality. Due to this any error occurring in the program gets traced. This is how it is different to work with Java.

- **Architecture Neutral:**

  The term architectural neutral seems to be weird, but yes Java is an architectural neutral language as well. The growing popularity of networks makes developers think distributed. In the world of network it is essential that the applications must be able to migrate easily to different computer systems. Not only to computer systems but to a wide variety of hardware architecture and operating system architectures as well. The Java compiler does this by generating byte code instructions, to be easily interpreted on any machine and to be easily translated into native machine code on the fly.The compiler generates an architecture-neutral object file format to enable a Java application to execute anywhere on the network and then the compiled code is executed on many processors, given the presence of the Java runtime system.Hence Java was designed to support applications on network. This feature of Java has thrived the programming language.

JAVA SERVER PAGES:

Java server Pages is a simple but powerful technology used to generate dynamic web pages on the server side. JSP's are direct extension of java servlets and provide a way to separate content generation from content presentation.

FEATURES OF JSP:

Portability:

Java Server Pages files can be run on any web server or web-enabled application server that provides support for them. Dubbed the JSP engine, this support involves recognition, translation, and management of the Java Server Page lifecycle and its interaction components.

Components:

It was mentioned earlier that the Java Server Pages architecture can include reusable Java components. The architecture also allows for the embedding of a scripting language directly into the Java Server Pages file. The components current supported include Java Beans, and Servlets.

Processing:

A Java Server Pages file is essentially an HTML document with JSP scripting or tags. The Java Server Pages file has a JSP extension to the server as a Java Server Pages file. Before the page is served, the Java Server Pages syntax is parsed and processed into a Servlet on the server side. The Servlet that is generated outputs real content in straight HTML for responding to the client.

JSP 2.0:

The new version of the JSP specification includes new features meant to improve programmer productivity. Namely:

- An Expression Language (EL) which allows developers to create Velocity-style templates (among other things).
- A faster/easier way to display parameter values.
- A clearer way to navigate nested beans.

HTML:

HTML, which stands for Hyper Text Markup Language, is the predominant markup language for web pages. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It allows images and objects to be embedded and can be used to create interactive forms. It is written in the form of HTML elements consisting of "tags" surrounded by angle brackets within the web page content. It can embed scripts in languages such as JavaScript which affect the behavior of HTML webpages. HTML can also be used to include Cascading Style Sheets (CSS) to define the appearance and layout of text and other material. The W3C, maintainer of both HTML and CSS standards, encourages the use of CSS over explicit presentational markup

HTML TAGS:

Basic HTML Tags:

<!-- -->	Specifies comments
<A>……….</A>	Creates hypertext links
<B>……….</B>	Formats text as bold
<BIG>……….</BIG>	Formats text in large font.
<BODY>…</BODY>	Contains all tags and text in the HTML document
<CENTER>...</CENTER>	Creates text

<DD>…</DD>	Definition of a term
<DL>...</DL>	Creates definition list
<FONT>…</FONT>	Formats text with a particular font
<FORM>...</FORM>	Encloses a fill-out form
<FRAME>...</FRAME>	Defines a particular frame in a set of frames
<H#>…</H#>	Creates headings of different levels
<HEAD>...</HEAD>	Contains tags that specify information about a document
<HR>...</HR>	Creates a horizontal rule
<HTML>…</HTML>	Contains all other HTML tags
<META>...</META>	Provides meta-information about a document
<SCRIPT>…</SCRIPT>	Contains client-side or server-side script
<TABLE>…</TABLE>	Creates a table
<TD>…</TD>	Indicates table data in a table
<TR>…</TR>	Designates a table row
<TH>…</TH>	Creates a heading in a table

Advantages:

- A HTML document is small and hence easy to send over the net. It is small because it does not include formatted information.
- HTML is platform independent.
- HTML tags are not case-sensitive.

JDBC:

JDBC is an API for the Java programming language that defines how a client may access a database. It provides methods for querying and updating data in a database. JDBC is oriented towards relational databases.

JDBC was first introduced in the Java 2 Platform, Standard Edition, version 1.1 (J2SE), together with a reference implementation JDBC-to-ODBC bridge, enabling connections to any ODBC-accessible data source in the JVM host environment.

FUNCTIONALITY:

JDBC allows multiple implementations to exist and be used by the same application. The API provides a mechanism for dynamically loading the correct Java packages and registering them with the JDBC Driver Manager. The Driver Manager is used as a connection factory for creating JDBC connections.

JDBC connections support creating and executing statements. These may be update statements such as SQL's CREATE, INSERT, UPDATE and DELETE, or they may be query statements such as SELECT. Additionally, stored procedures may be invoked through a JDBC connection. JDBC represents statements using one of the following classes:

- Statement – the statement is sent to the database server each and every time.
- Prepared Statement – the statement is cached and then the execution path is pre determined on the database server allowing it to be executed multiple times in an efficient manner.
- Callable Statement – used for executing stored procedures on the database.

Update statements such as INSERT, UPDATE and DELETE return an update count that indicates how many rows were affected in the database. These statements do not return any other information.

Query statements return a JDBC row result set. The row result set is used to walk over the result set. Individual columns in a row are retrieved either by name or by column number. There may be any number of rows in the result set. The row result set has metadata that describes the names of the columns and their types.

10. SYSTEMDESIGN

UML DIAGRAMS:

A diagram is a graphical presentation of a set of elements, most often rendered as a connected graph. UML includes nine such diagrams. Unified Modeling Language (UML) is a probably the most widely known and used notation for object-oriented methods. The Unified Modeling Language (UML) is a standard language for writing software blueprints.

The UML may be used to visualize, specify, construct and document the artifacts. A Modeling Language is a language whose vocabulary and rules focus on the conceptual and the physical and the physical representation of a system. Modeling is the designing of the software applications before coding. It is an essential part of large software projects, and helpful to medium and even small projects as well. A model plays an analogous role in software development project's success can assure themselves that business functionality is complete and correct. Care should be taken that end-user's needs are met.

The underlying premise of UML is that no one diagram can capture the different elements of a system at different points of time in the software life cycle of a system.

The nine UML diagrams are as follows:

- Class Diagram
- Object Diagram
- Use case Diagram
- Sequence Diagram
- Activity Diagram
- State chart Diagram
- Collaboration Diagram
- Component Diagram
- Deployment Diagram

Class diagram:

The class diagram is used to define a detailed design of the system. It classifies the actors into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provide by the class are termed “methods” of the class. A part from this, each class may have certain “attributes” that uniquely identify the class.

Object diagram:

The object diagram is a special kind of class diagram. An object is an instance of a class. This essentially means that an object represents the state of a class at a given point of time while the system is running. The object diagram captures the state of different classes in the system and their relationships or associations at a given point of time.

Use case diagram:

The use case diagram is used to identify the primary elements and processes that form the system. The primary elements are termed as “actors” and the processes are called as “use cases”. The use case diagram shows which actoes interact with each use case.

Sequence diagram:

A sequence diagram represents the interaction between differentobjects in the system. The important aspect of a sequence diagram is that it is time-ordered. This means that the exact sequence of the interactions between the objects is represented step by step. Different objects in the sequence diagram interact with each other by passing “messages”.

Activity diagram:

The process flows in the system are captured in the activity diagram. Similar to a state diagram, an activity diagram also consists of activities, actions, transitions, initial and final states, and guard conditions.

State chart diagram:

A state chart diagram, as the name suggests, represents the different states that objects in the system undergo during their life cycle. Objects in the system change states in response to events. In addition to this, a state diagram also captures the transition of the object’s state from an initial state to a final state in response to events affecting the system.

Collaboration diagram:

A collaboration diagram groups together the interactions between different objects. The interactions are listed as numbered interactions that help to trace the sequence of the interactions. The collaboration diagram helps to identify all the possible interactions that each object has with other objects.

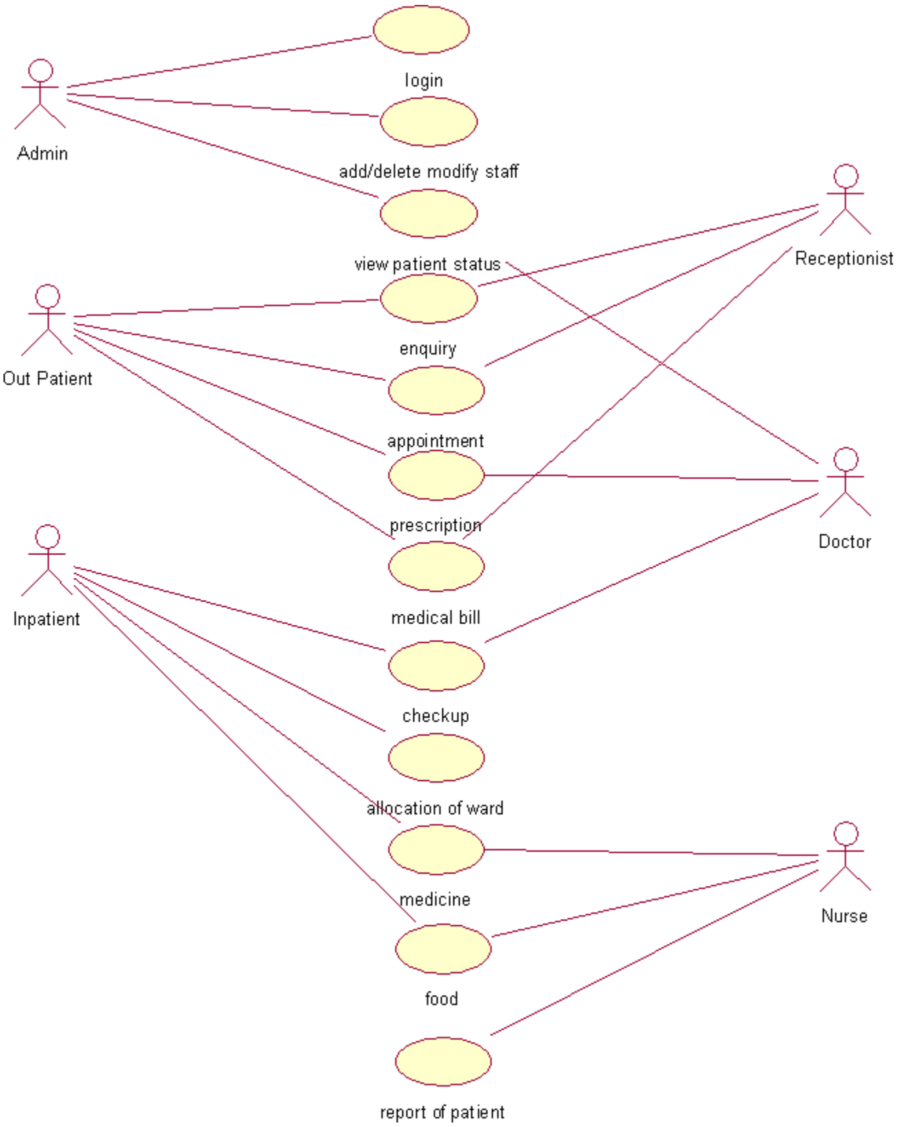
Component diagram:

The component diagram represents the high level parts that make up the system. This diagram depicts, at a high level, what components form part of the system and how they are interrelated. A component diagram depicts the components culled after the system has undergone the development or construction phase.

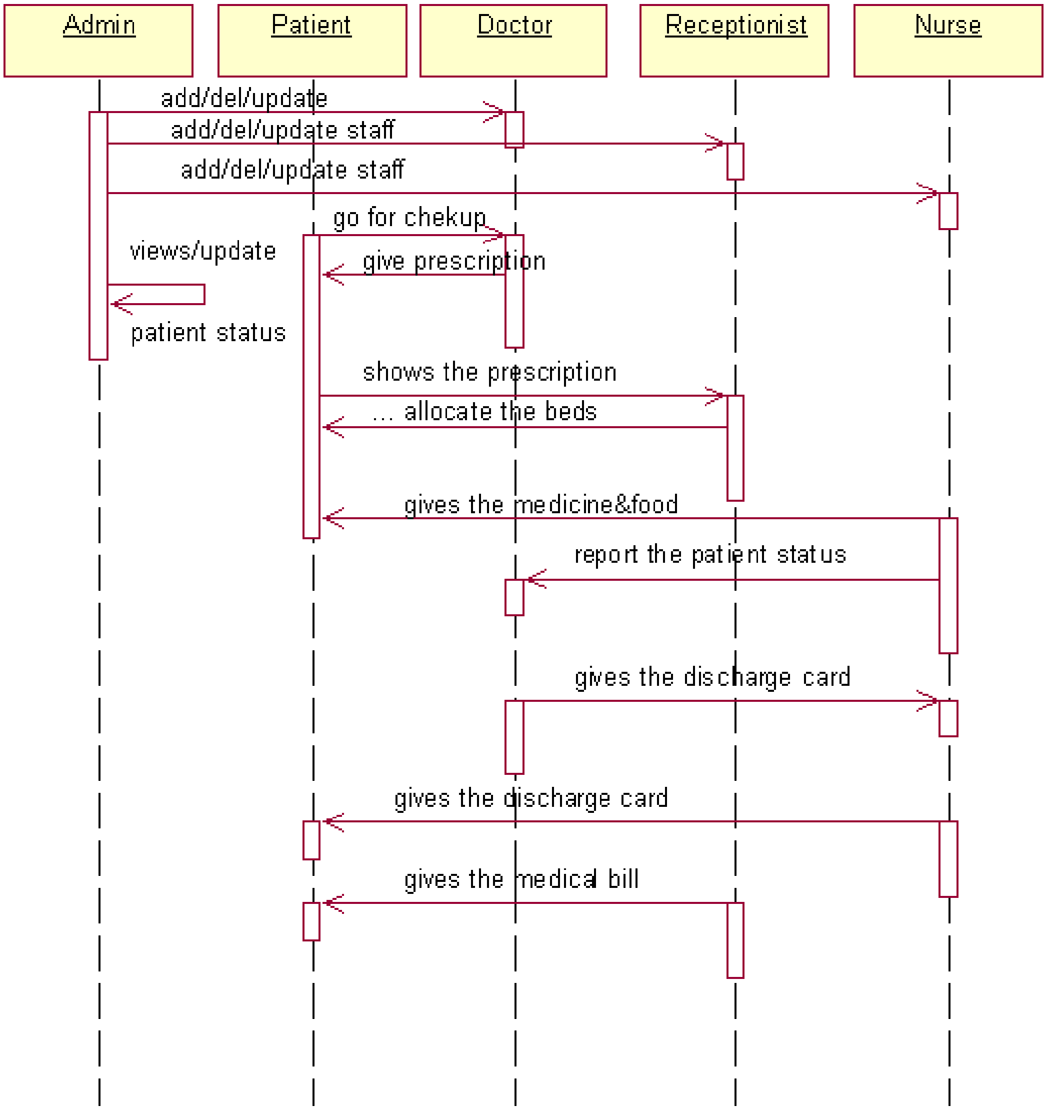
Deployment diagram:

The deployment diagram captures the configuration of the runtime elements of the application. This diagram is by far most useful when a system is built and ready to be deployed.

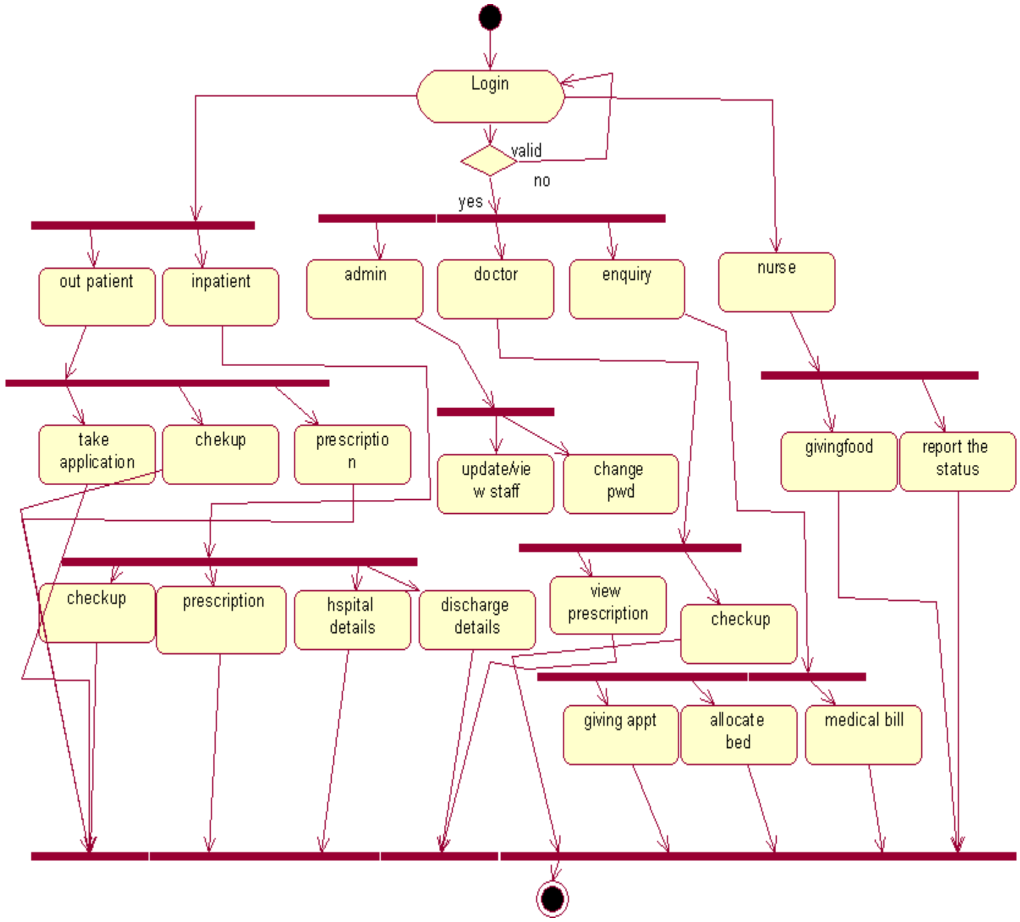
USE CASE DIAGRAM:



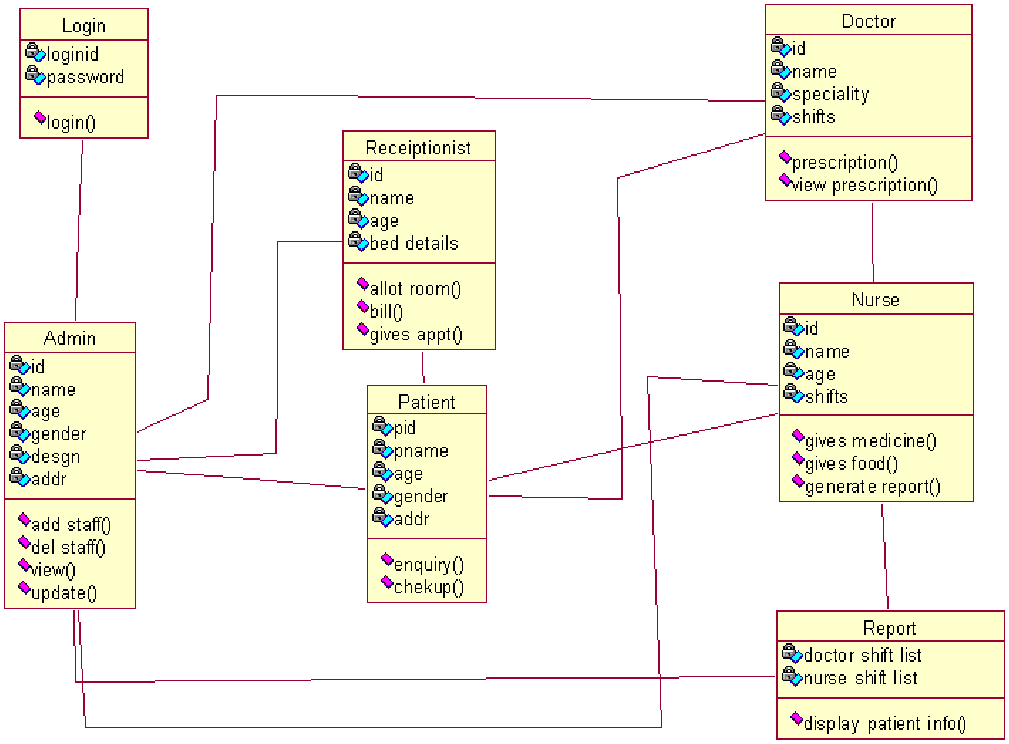
SEQUENCE DIAGRAM:



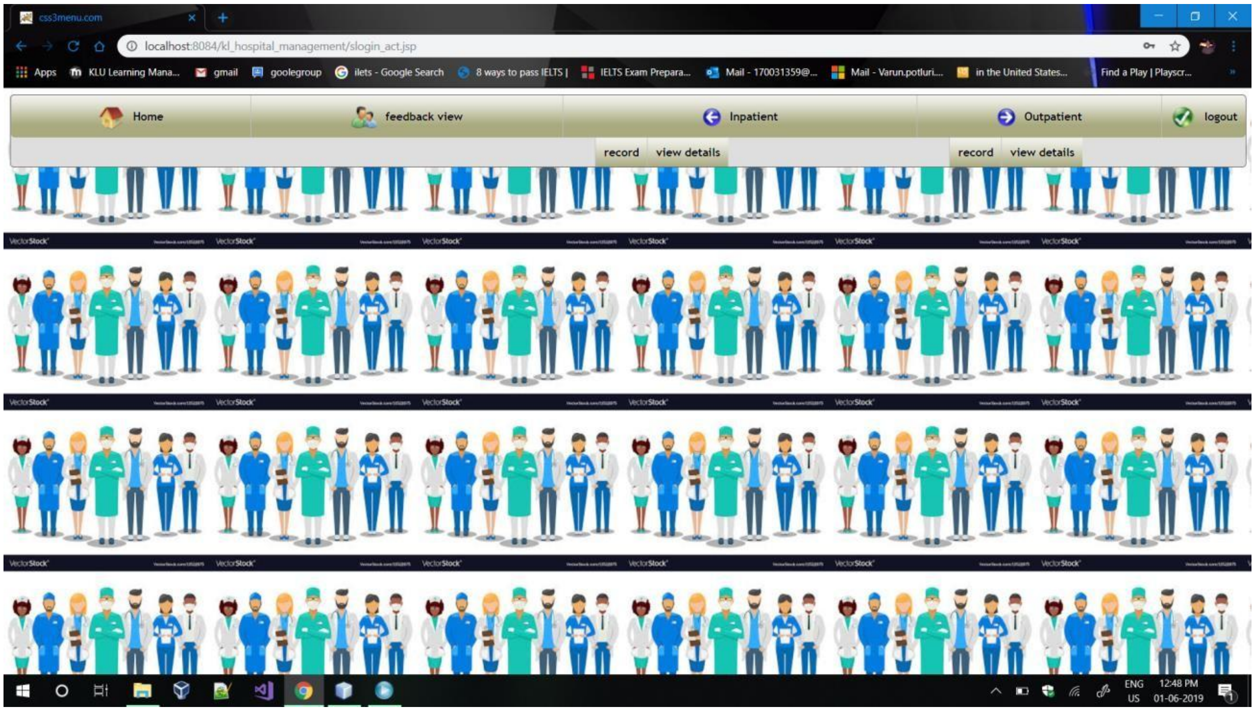
ACTIVITY DIAGRAM

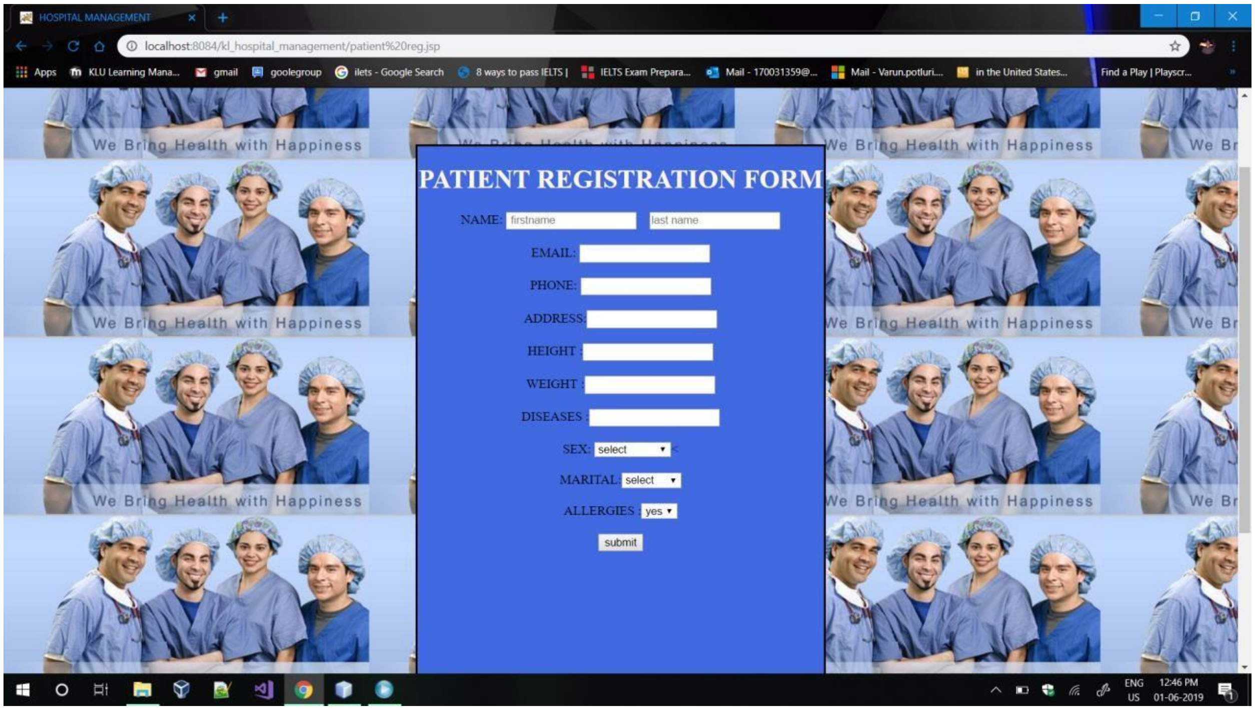


CLASS DIAGRAM



11. SCREENS


css3menu.com
localhost:8084/kl_hospital_management/slogin_act.jsp
Apps
KLU Learning Mana...
gmail
goolegroup
ilets - Google Search
8 ways to pass IELTS |
IELTS Exam Prepara...
Mail - 170031359@...
Mail - Varun.potluri...
in the United States...
Find a Play | Playscr...
Home
feedback view
Inpatient
record
view details
Outpatient
record
view details
logout
VectorStock
ENG
US
12:48 PM
01-06-2019



PATIENT REGISTRATION FORM

NAME: firstname last name

EMAIL:

PHONE:

ADDRESS:

HEIGHT :

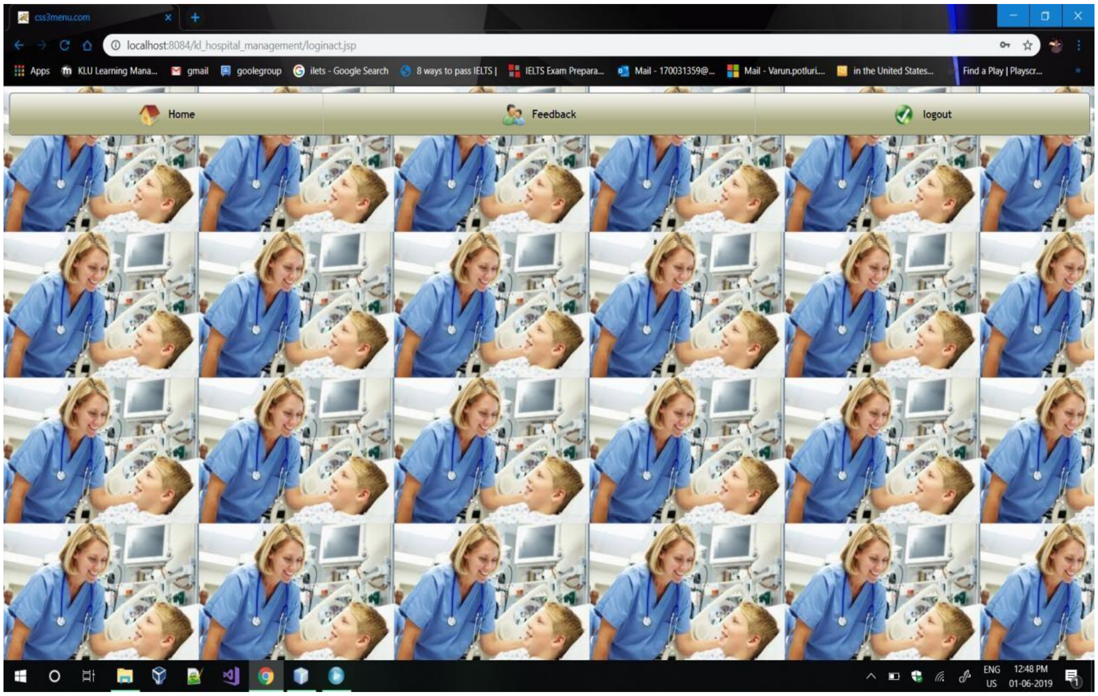
WEIGHT :

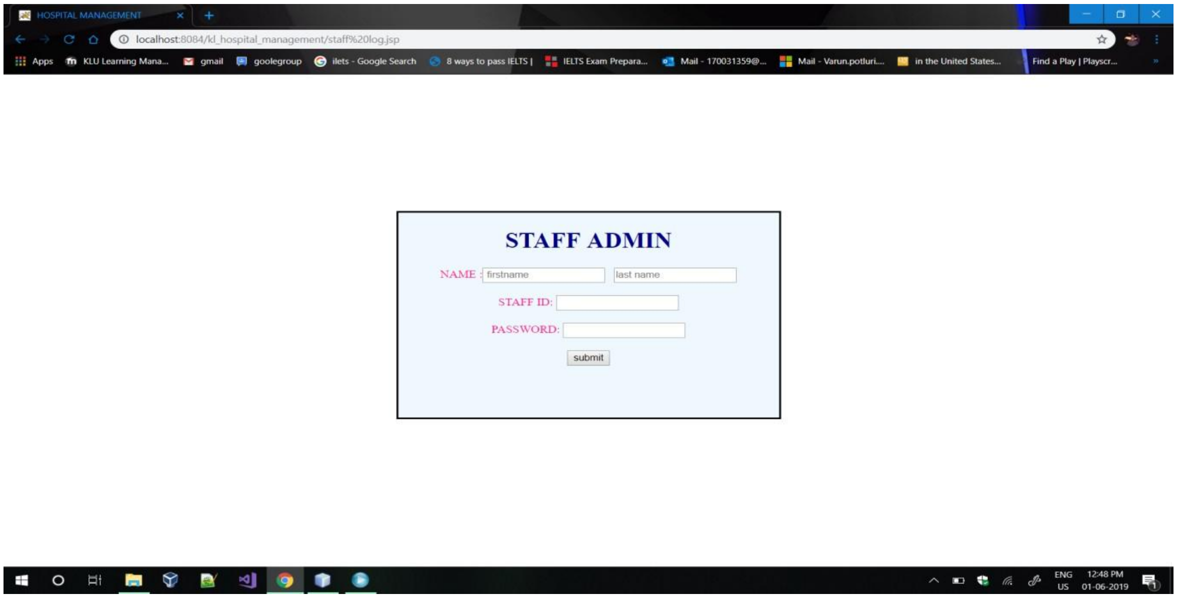
DISEASES :

SEX: select

MARITAL: select

ALLERGIES : yes


css3menu.com
localhost:8084/kl_hospital_management/loginact.jsp
Apps
KLU Learning Mana...
gmail
goolegroup
ilets - Google Search
8 ways to pass IELTS |
IELTS Exam Prepara...
Mail - 170031359@...
Mail - Varun.potluri...
in the United States...
Find a Play | Playscr...
Home
Feedback
logout
ENG
US
12:48 PM
01-06-2019



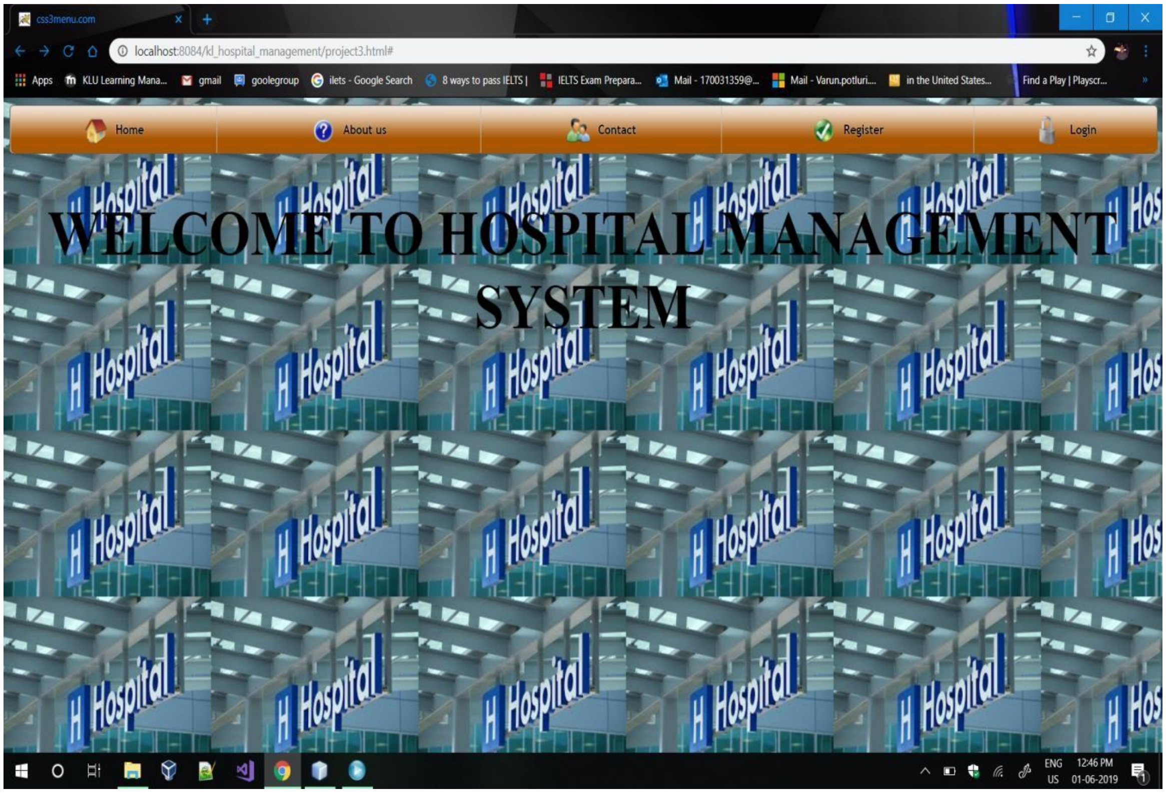
STAFF ADMIN

NAME : firstname last name

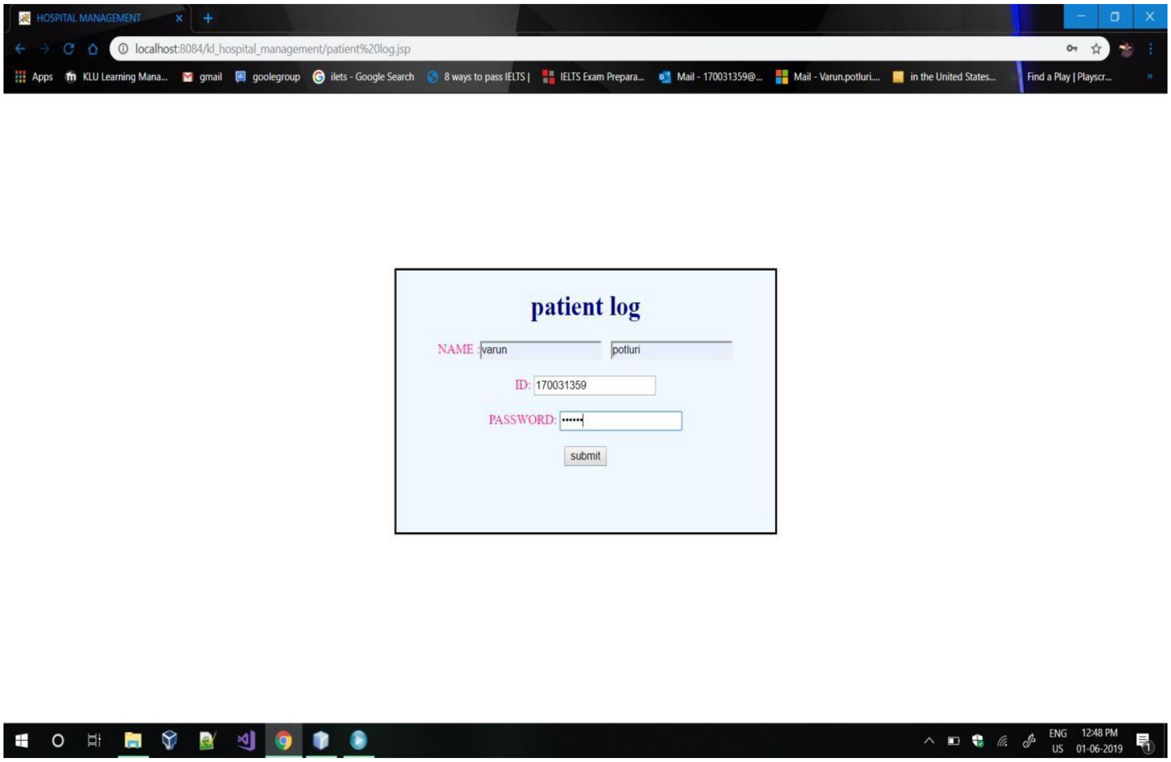
STAFF ID:

PASSWORD:

submit



WELCOME TO HOSPITAL MANAGEMENT SYSTEM



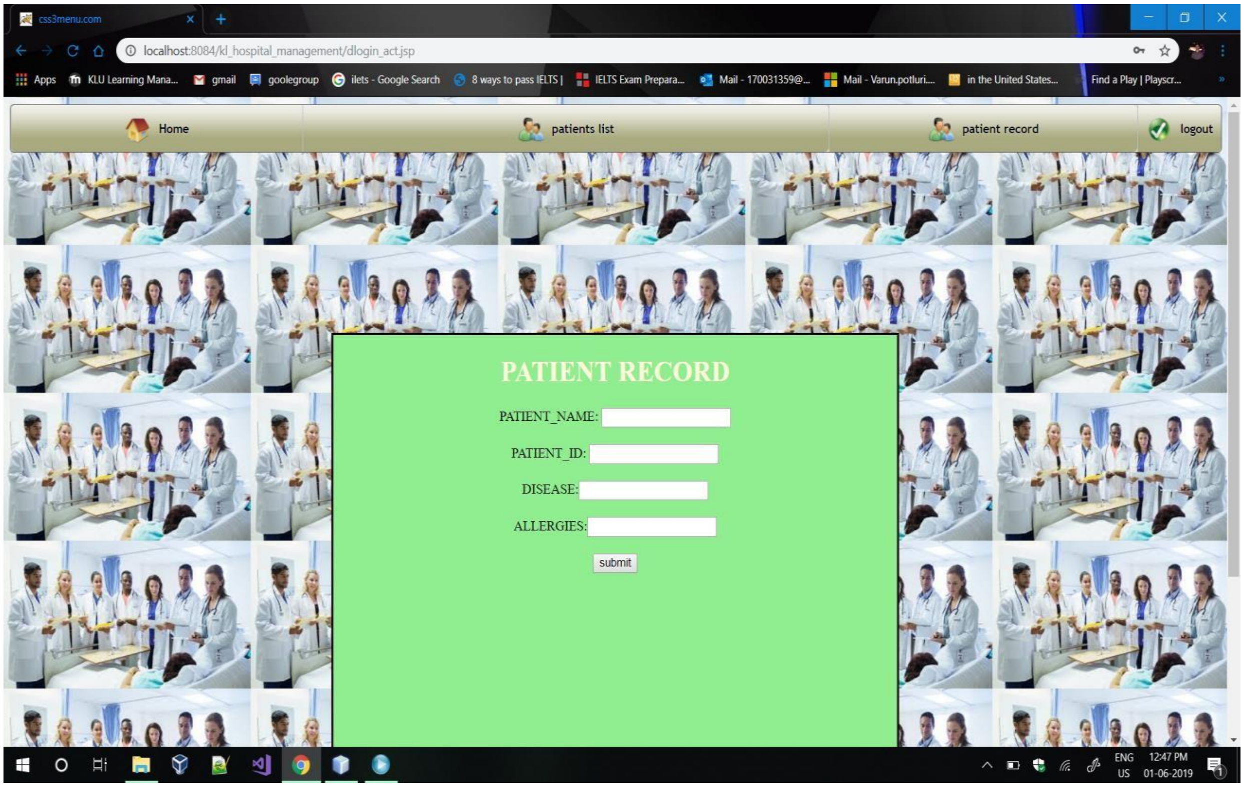
patient log

NAME : varun potluri

ID: 170031359

PASSWORD: ••••••

submit



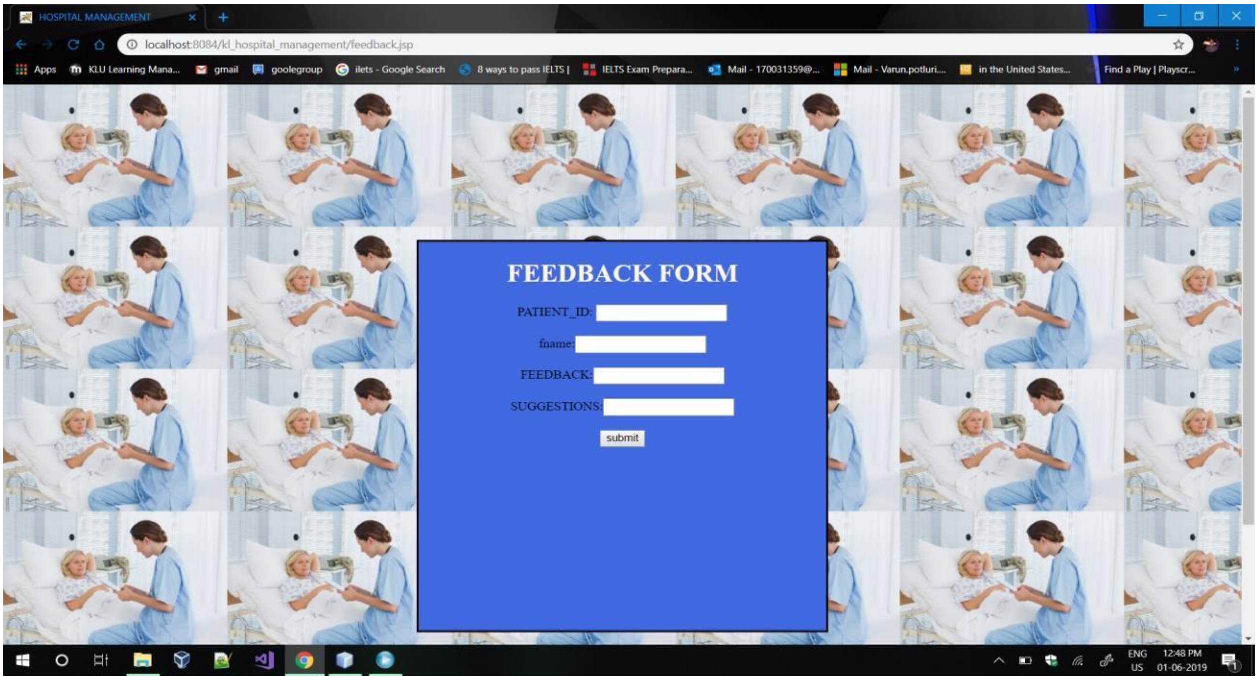
PATIENT RECORD

PATIENT_NAME:

PATIENT_ID:

DISEASE:

ALLERGIES:



FEEDBACK FORM

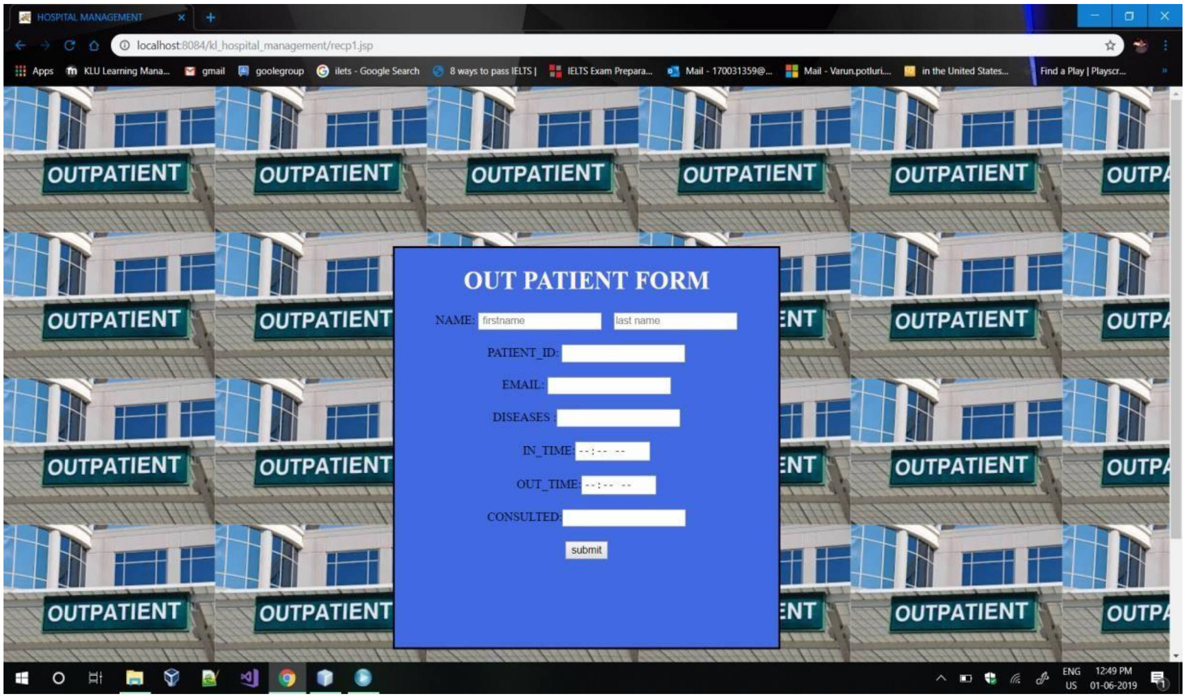
PATIENT_ID:

fname:

FEEDBACK:

SUGGESTIONS:

submit



OUT PATIENT FORM

NAME: firstname last name

PATIENT_ID:

EMAIL:

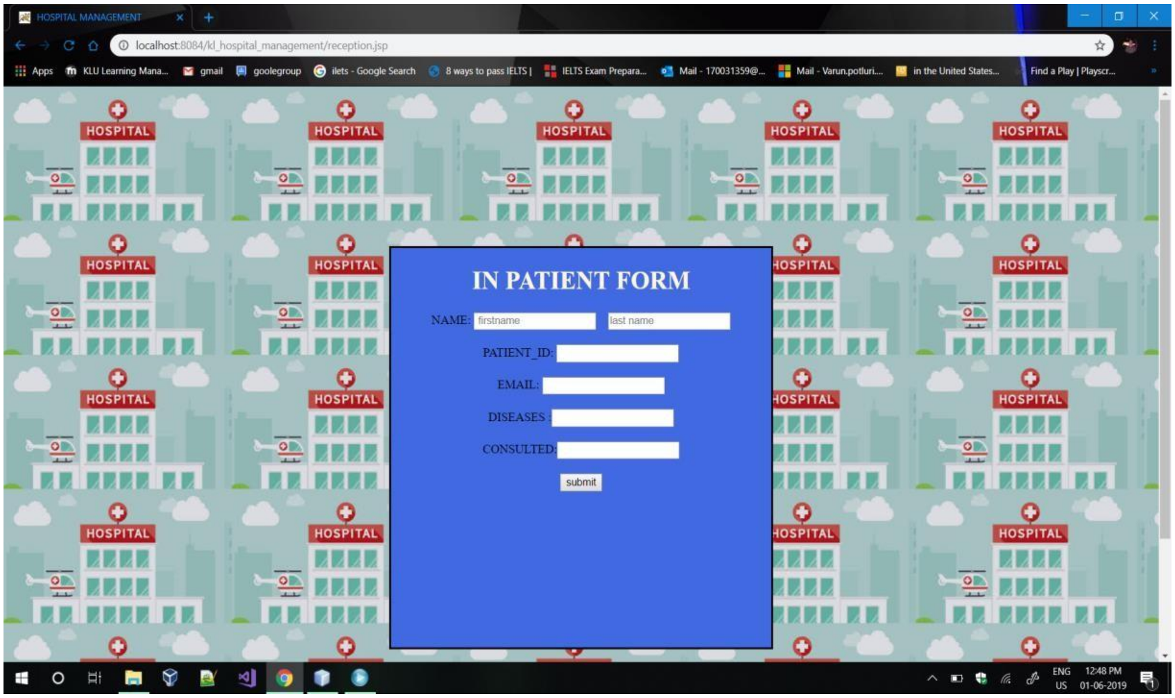
DISEASES :

IN_TIME: --:-- --

OUT_TIME: --:-- --

CONSULTED:

submit



IN PATIENT FORM

NAME: firstname last name

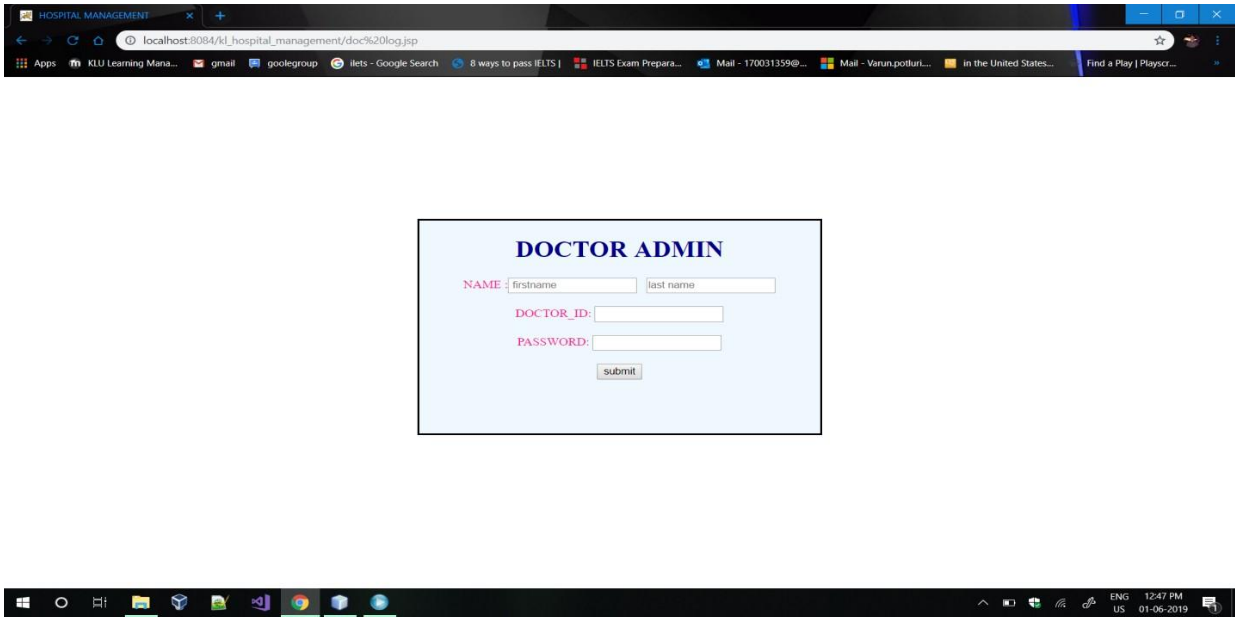
PATIENT_ID:

EMAIL:

DISEASES :

CONSULTED:

submit



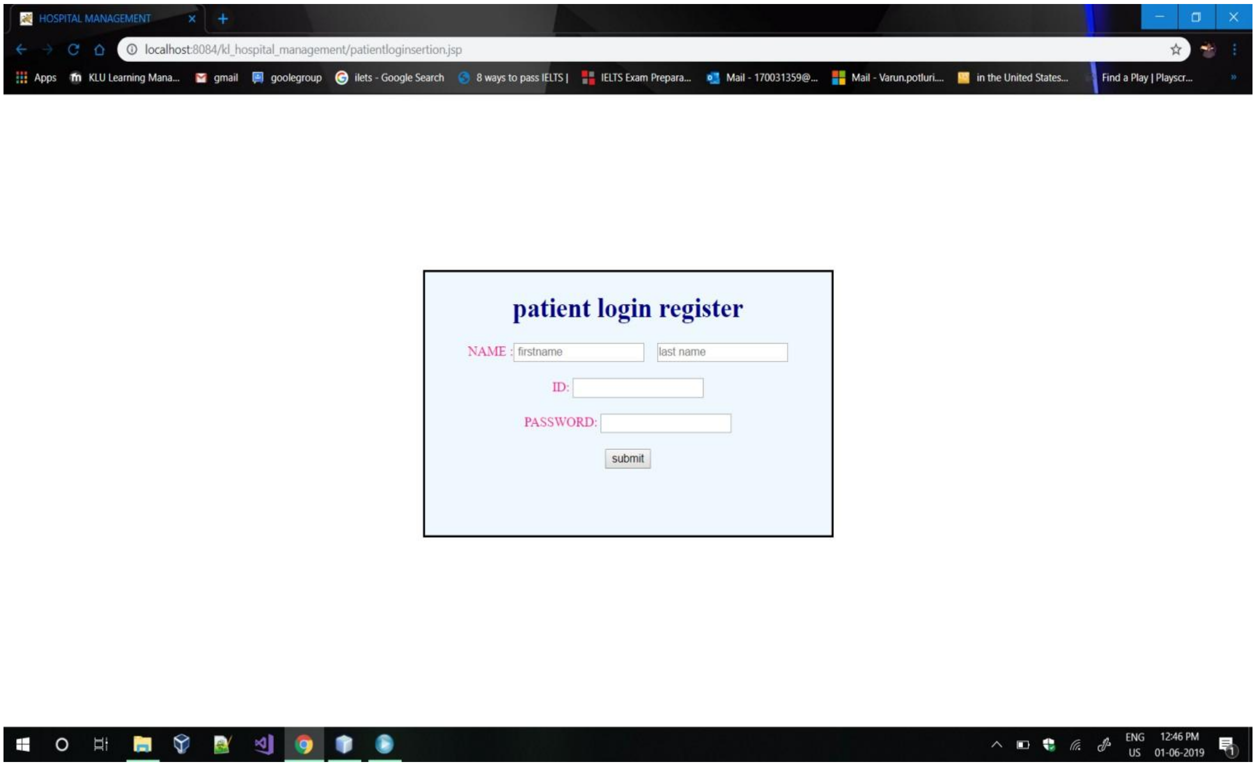
DOCTOR ADMIN

NAME : firstname last name

DOCTOR_ID:

PASSWORD:

submit

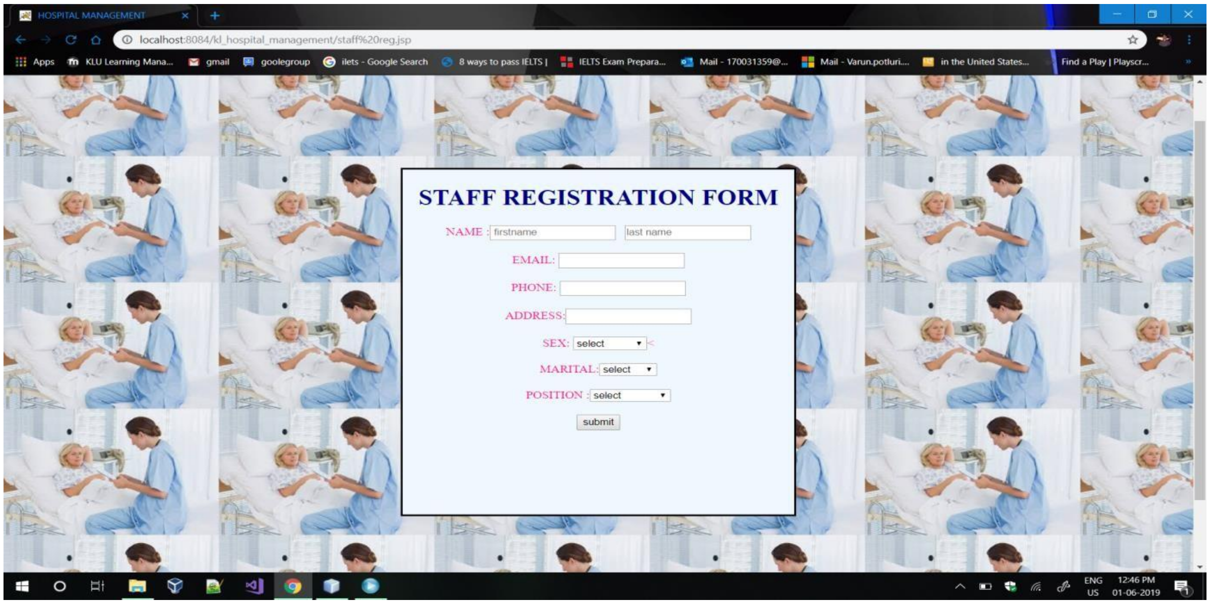


patient login register

NAME : firstname last name

ID:

PASSWORD:



STAFF REGISTRATION FORM

NAME : firstname last name

EMAIL:

PHONE:

ADDRESS:

SEX: select

MARITAL: select

POSITION : select

12. DATABASE DESIGN

12.1 TABLES

LOGIN:

Field Name	Data Type	Constraints
Userid	Varchar2(10)	Null
Password	Varchar2(10)	Null
Roles	Varchar2(20)	Null
Permission	Varchar2(20)	Null

REGISTRATION:

Field Name	Data Type	Constraints
PID	Varchar2(10)	Primary key (not null)
Name	Varchar2(20)	Null
Age	Number(5)	Null
Gender	Varchar2(10)	Null
Occupation	Varchar2(25)	Null
Address	Varchar2(20)	Null
Email	Varchar2(20)	Null
Cellphoneno	Varchar2(20)	Null
Ref_doctor	Varchar2(20)	Null
Patient_type	Varchar2(20)	Null
If_patient_bedno	Number(10)	Null
Dob	Varchar2(20)	Null
Form_no	Number(10)	Null

STAFF:

Field Name	Data Type	Constraints
ID	Varchar2(20)	Primary key (not null)
Name	Varchar2(20)	Null
Gender	Varchar2(20)	Null
Doj	Varchar2(20)	Null
Designation	Varchar2(20)	Null
Address	Varchar2(20)	Null
Payscale	Number(10)	Null
Role	Varchar2(20)	Null
Permission	Varchar2(20)	Null

OUTPATIENT:

Field Name	Data Type	Constraints
PID	Varchar2(20)	Null
Pname	Varchar2(20)	Null
Age	Number(5)	Null
Address	Varchar2(20)	Null
Gender	Varchar2(20)	Null
Doc_name	Varchar2(20)	Null
Speciality	Varchar2(20)	Null
Complaint	Varchar2(20)	Null
Illness_history	Varchar2(20)	Null
Time	Varchar2(20)	Null
Day	Varchar2(20)	Null
Date_of_appt	Varchar2(20)	Null

INPATIENT:

Field Name	Data Type	Constraints
PID	Varchar2(20)	Null
Room_no	Number(5)	Null
Name	Varchar2(20)	Null
Age	Number(5)	Null
Gender	Varchar2(20)	Null
Address	Varchar2(20)	Null
Name_of_guardian	Varchar2(20)	Null
Ref_doc	Varchar2(20)	Null
Treating_consultant	Varchar2(20)	Null
Date_of_admision	Varchar2(20)	Null
Time_of_admision	Varchar2(20)	Null
Date_of_discharge	Varchar2(20)	Null
Time_of_discharge	Varchar2(20)	Null

PATIENT HEALTH STATUS:

Field Name	Data Type	Constraints
PID	Varchar2(20)	Null
Pname	Varchar2(20)	Null
Ref_doc	Varchar2(20)	Null
Speciality	Varchar2(20)	Null
Wardno	Number(5)	Null
Cond_patient	Varchar2(20)	Null
Stautus	Varchar2(20)	Null
Prescription	Varchar2(20)	Null
Day	Varchar2(20)	Null

PATIENT DISCHARGE:

Field Name	Data Type	Constraints
PID	Varchar2(20)	Null
Room_no	Number(5)	Null
Pname	Varchar2(20)	Null
Gender	Varchar2(20)	Null
Age	Number(5)	Null
Consultant	Varchar2(20)	Null
Date_of_admision	Varchar2(20)	Null
Time_of_admision	Varchar2(20)	Null
Date_of_surgery	Varchar2(20)	Null
Time_of_surgery	Varchar2(20)	Null
Date_od_discharge	Varchar2(20)	Null
Time_of_discharge	Varchar2(20)	Null
Final_diagnosis	Varchar2(20)	Null
Complaints	Varchar2(20)	Null
Investigation	Varchar2(20)	Null
Treatment_given	Varchar2(20)	Null
Cond_discharge	Varchar2(20)	Null
Review	Varchar2(20)	Null

13. TEST REPORT & PLAN

Testing is the process of detecting errors. Testing performs a very critical role for quality assurance and for ensuring the reliability of software. The results of testing are used later on during maintenance also.

Psychology of Testing

The aim of testing is often to demonstrate that a program works by showing that it has no errors. The basic purpose of testing phase is to detect the errors that may be present in the program. Hence one should not start testing with the intent of showing that a program works, but the intent should be to show that a program doesn't work. Testing is the process of executing a program with the intent of finding errors.

Testing Objectives

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say,

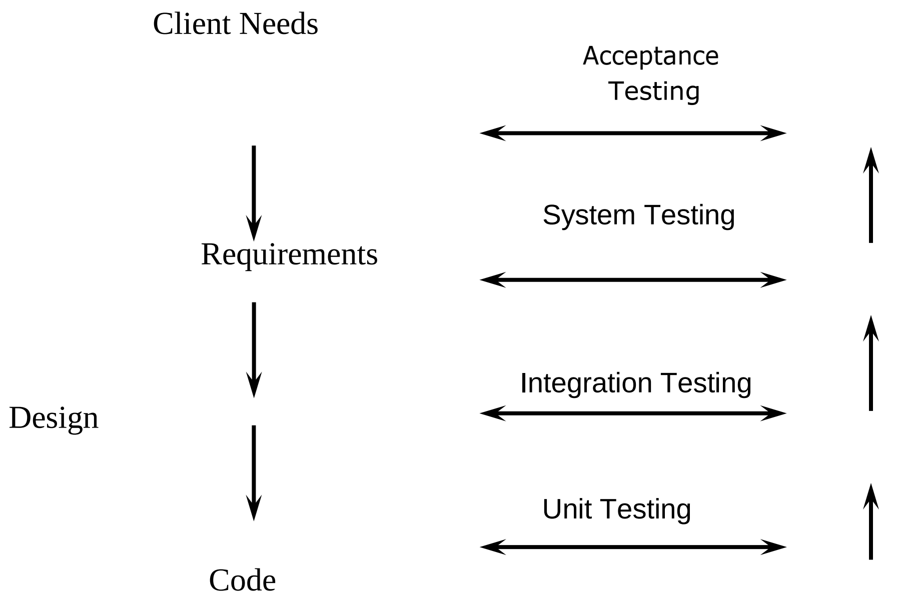
- Testing is a process of executing a program with the intent of finding an error.

A successful test is one that uncovers an as yet undiscovered error.

- A good test case is one that has a high probability of finding error, if it exists.
- The tests are inadequate to detect possibly present errors.
- The software more or less confirms to the quality and reliable standards.

Levels of Testing

In order to uncover the errors present in different phases we have the concept of levels of testing. The basic levels of testing are as shown below…



System Testing

The philosophy behind testing is to find errors. Test cases are devised with this in mind. A strategy employed for system testing is code testing.

Code Testing

This strategy examines the logic of the program. To follow this method we developed some test data that resulted in executing every instruction in the program and module i.e. every path is tested. Systems are not designed as entire nor are they tested as single systems. To ensure that the coding is perfect two types of testing is performed or for that matter is performed or that matter is performed or for that matter is performed on all systems.

Types of Testing

- **Unit Testing**
- **Link Testing**

Unit Testing

Unit testing focuses verification effort on the smallest unit of software i.e. the module. Using the detailed design and the process specifications testing is done to uncover errors within the boundary of the module. All modules must be successful in the unit test before the start of the integration testing begins.

In this project each service can be thought of a module. There are so many modules like Login, HWAdmin, MasterAdmin, Normal User, and PManager. Giving different sets of inputs has tested each module. When developing the module as well as finishing the development so that each module works without any error. The inputs are validated when accepting from the user.

In this application developer tests the programs up as system. Software units in a system are the modules and routines that are assembled and integrated to form a specific function. Unit testing is first done on modules, independent of one another to locate errors. This enables to detect errors. Through this errors resulting from interaction between modules initially avoided.

Link Testing

Link testing does not test software but rather the integration of each module in system. The primary concern is the compatibility of each module. The Programmer tests where modules are designed with different parameters, length, type etc.

Integration Testing:

After the unit testing we have to perform integration testing. The goal here is to see if modules can be integrated proprerly, the emphasis being on testing interfaces between modules. This testing activity can be considered as testing the design and hence the emphasis on testing module interactions.

In this project integrating all the modules forms the main system. When integrating all the modules I have checked whether the integration effects working of any of the services by giving different combinations of inputs with which the two services run perfectly before Integration.

System Testing

Here the entire software system is tested. The reference document for this process is the requirements document, and the goal os to see if software meets its requirements.

Here entire 'ATM' has been tested against requirements of project and it is checked whether all requirements of project have been satisfied or not.

Acceptance Testing

Acceptance Test is performed with realistic data of the client to demonstrate that the software is working satisfactorily. Testing here is focused on external behavior of the system; the internal logic of program is not emphasized.

In this project 'Network Management Of Database System' I have collected some data and tested whether project is working correctly or not.

Test cases should be selected so that the largest number of attributes of an equivalence class is exercised at once. The testing phase is an important part of software development. It is the process of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied.

White Box Testing

This is a unit testing method where a unit will be taken at a time and tested thoroughly at a statement level to find the maximum possible errors. I tested step wise every piece of code, taking care that every statement in the code is executed at least once. The white box testing is also called Glass Box Testing.

I have generated a list of test cases, sample data. which is used to check all possible combinations of execution paths through the code at every module level.

Black Box Testing

This testing method considers a module as a single unit and checks the unit at interface and communication with other modules rather getting into details at statement level. Here the module will be treated as a block box that will take some input and generate output. Output for a given set of input combinations are forwarded to other modules.

TEST REPORT1:

1. **Project Name:**Hospital Management System

2. **Module Name:** login

3. **Unit Name:** User Name

4. **Test Result:** user-id and password is tested and verified.

TEST PLAN FOR HOSPITAL MANAGEMENT SYSTEM:

Unit id: Hospital Management System

Test case id: login page

Test type: unit level testing

Form name: login form

Base Table: login Table

PURPOSE:

Login table is used to save the user id, password and privileges provided by administrator to staff. By using these details they will perform operations.

TEST CASE DESCRIPTION:

Check login for Hospital Patient Management System WITH VALID USERNAME AND PASSWORD

Username varchar 2(20)Primary key

Password varchar 2(20) Primary key

Role varchar 2(20)

TEST DATA:

S.NO	Input specification	Output specification
1	Column name: username& password Valid input: enter valid user name & password. Invalid input: there are different possibilities. Invalid username: valid password Validusername: invalid password Invalid username & password	Valid output: if the username and password are correct then form navigation to page according to the input details. Invalid output: error message is displayed which helps us to re-enter the username & password again.

TEST COMPLETION CRITERIA:

When expected results match the actual results after performing tests the test is considered to be complete.

TEST PROCESS:

Login form will be used for allowing correct user to use the software. Every person will be given a username & password

14. CONCLUSION

Hospital Management System is entry in to an area of information processing that is significantly different from existing ones. Because, of this fact a different approach to Hospital Management System development and implementation is called for business engaged in launching or upgrading hospital management system must contain wide variety of factors that can influence activities.

Hospital Management System make possible to simplify, improve & automate the hospitals activities as well as automatic transfer and exchange of information and management of administrative , handling numerical data without human operation.

15.BIBLIOGRAPHY

References for the Project Development Were Taken From the following Books and Web Sites .

JAVA Technologies
JAVA Complete Reference

Java Script Programming by Yehuda Shiran

Mastering JAVA Security

Head First EJB Sierra Bates

J2EE Professional by Shadab siddiqui

JAVA server pages by Larne Pekowsley

JAVA Server pages by Nick Todd

HTML
HTML Black Book by Holzner

JDBC
Java Database Programming with JDBC by Patel moss.

Software Engineering by Roger Pressman