**INDUSTRY ORIENTED PROJECT TRAINING REPORT**

**HOSPITAL MANAGEMENT SYSTEM**

Submitted in partial fulfilment of the Requirements for the project training of degree of

**BACHELOR OF TECHNOGY**

In

**Computer Science & Engineering**

****

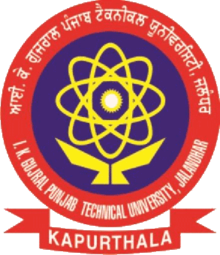
**AS PART OF COURSE WORK OF B.TECH (8TH SEM)**

**SUBMITTED BY**

DHANAKSHI JAIN

(CSE B)

1535447



**Affiliated to I.K. Gujral Punjab Technical University, Jalandhar**

**(BATCH: 2015-2019)**

**CERTIFICATE OF ORIGINALITY OF WORK**

I hereby certify that the work which is being presented in the project report entitled by **‘HOSPITAL MANAGEMENT SYSTEM’** in partial fulfillment of requirements for the award of degree of B.Tech Computer Science & Engineering submitted in the Department of Computer Science & Engineering at **CGC TECHNICAL CAMPUS, Jhanjeri, Mohali** under **I.K. Gujral** **PUNJAB TECHNICAL UNIVERSITY, JALANDHAR** is an authentic record of my own work under the supervision of **Mr. SABBIR POONAWALA**.

**Dhanakshi Jain**

**1535447**

**Date:**

This is to certify that the above statement made by the student is correct to the best of our knowledge and belief.

**Signature of the Supervisor**

**Mr. Sabbir Poonawala**

**Signature of H.O.D**

**Dr. Raman Chadha**

### **CERTIFICATE**

This is to certify that Ms. **DHANAKSHI JAIN** has completed the 6 Months Industrial Training during the period from FEB to MAY in our Organization as a Partial Fulfillment of Degree of Bachelor of Technology in Computer science & Engineering. She was trained in the field of **FULL STACK DEVELOPER.**

**Signature & Seal of Training Manager**

**DECLARATION**

I hereby declare that the work which is being presented in this project entitled **HOSPITAL MANAGEMENT SYSTEM** in partial fulfilment of degree of **Bachelor of Technology** in **COMPUTER SCIENCE AND ENGINEERING** is an authentic record of my own work carried out under the supervision and guidance of **Mr. Sabbir Poonawala, Trainer.**

I am fully responsible for the matter embodied in this project in case of any discrepancy found in the project and the project has not been submitted for the award of any other degree.

**DATE:**

**PLACE:**

**DHANAKSHI JAIN**

**ACKNOWLEDGEMENT**

For the successful completion of this project, I'd extend a sincere thanks to our project guide as well as trainer **Mr.** **Sabbir Poonawala** sir, who has been there with us while building the complete code from scratch. Without his guidance and teaching, it'd have been impossible of us to create the project.

I'm also grateful to our batch owners and people who've helped us out in every way possible, Shilpa Mahajani Ma’am, Mohit Kariya Sir, Gerard Thomas Sir and Komal Sanjay Pawar Ma’am. Without them, internship at such a huge company would have not been completed successfully. They ensured our smooth functioning and handled all the quirks and doubts while the sessions which helped us a lot in order to calmly finish the project.

I'd also like to thank **Dr. RAJNEESH TALWAR,** **Principal** CGC Technical Campus, Jhanjeri, Mohali and **MANAGEMENT** for providing encouragement and for giving us such an amazing opportunity to work in **COGNIZANT** through the college semester in order to get a corporate experience and education. It is only because of his thought that we were able to gain such a training with a parallel balance of college academics.

**DHANAKSHI JAIN**

**TABLE OF CONTENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| S. NO. | CHAPTER NO. | SUB TOPICS | PAGE NO. |
| 1. | Chapter-1 | ABOUT |  |
|  |  | 1.1. The company |  |
|  |  | 1.2. Purpose of this document |  |
| 2. | Chapter-2 | INTRODUCTION TO PROJECT |  |
|  |  | 2.1. Purpose |  |
|  |  | 2.2. Objectives |  |
|  |  | 2.3. Intended Audience |  |
| 3. | Chapter-3 | TECHNOLOGIES AND SOFTWARES |  |
|  |  | 3.1. Technologies |  |
|  |  | 3.1.1. Java |  |
|  |  | 3.1.2. Maven |  |
|  |  | 3.1.3. Spring MVC |  |
|  |  | 3.1.4. Hibernate |  |
|  |  | 3.1.5. Html |  |
|  |  | 3.1.6. CSS |  |
|  |  | 3.1.7. JavaScript |  |
|  |  | 3.1.8. Mockito |  |
|  |  | 3.1.9. Junit |  |
|  |  | 3.1.10. Logger |  |
|  |  | 3.2. Softwares |  |
|  |  | 3.2.1. Apache Tomcat |  |
|  |  | 3.2.2. Oracle |  |
|  |  | 3.2.3. Eclipse |  |
| 4. | Chapter-4 | PROJECT OVERVIEW |  |
|  |  | 4.1. Problem Introduction |  |
|  |  | 4.2. Business Case |  |
|  |  | 4.3. Project Scope |  |
|  |  | 4.4. Modules |  |
|  |  | 4.5. Requirement Engineering |  |
| 5. | Chapter-5 | BUSINESS REQUIREMENTS |  |
|  |  | 5.1. High Level Requirements |  |
|  |  | 5.2. Detailed Business Requirements |  |
|  |  | 5.2.1. functional requirements |  |
| S. NO. | **CHAPTER NO.** | **SUB TOPICS** | **PAGE NO.** |
| 6. | Chapter-6 | ANALYSIS |  |
|  |  | 6.1. Hardware Requirements |  |
|  |  | 6.2. Software Requirements |  |
| 7. | Chapter-7 | TABLES |  |
|  |  | 7.1. Admin Table |  |
|  |  | 7.2. Patient Table |  |
|  |  | 7.3. Physician Table |  |
|  |  | 7.4. Diagnosis Table |  |
| 8. | Chapter-8 |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**LIST OF FIGURES:**

|  |  |  |
| --- | --- | --- |
| S. NO. | NAME OF FIGURE | PAGE NO. |
| 1. |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**CHAPTER-1**

**ABOUT**

* 1. THE COMPANY

**Cognizant** is an American multinational corporation that provides IT services, including digital, technology, consulting, and operations services. It is headquartered in Teaneck, New Jersey, United States of America. Cognizant is included in the NASDAQ-100 and the S&P 500 indices. It is also one of the fastest growing Fortune 500 companies. It was founded as an in-house technology unit of DUN & BRADSTREEIN in 1994 and started serving external clients in 1996.

Cognizant had a period of fast growth during the 2000s, becoming a Fortune 500 company in 2011. In 2015, the FORTUNE Magazine named it as the world's fourth most admired IT Services company. In 2017, Cognizant was named in Fortune’s Future 50 list.

Cognizant provides information technology, information security, consulting, ITO and BPO services. These include business and technology consulting, system integration, application development & maintenance, IT infrastructure services, analytics, business intelligence data, warehousing, customer relationship management, supply chain management, engineering & manufacturing solutions, enterprise resource planning, research and development, outsourcing, and testing solutions.

The company has more than 281,600 employees globally, of which over 150,000 are in India across 10 locations with a plurality in Chennai. The other centers of the company are in Bangalore, Coimbatore, Gurgaon, Noida, Hyderabad, Kochi, Kolkata, Mangalore, Mumbai, and Pune. The company has local, regional, and global delivery centers in the UK, Hungary, The Netherlands, Spain, China, Philippines, Canada, Brazil, Argentina, Mexico etc.

Cognizant has three key practice areas that span its business — Digital Business, Digital Operations, and Digital Systems & Technology.

* 1. PURPOSE OF THIS DOCUMENT

This document is aimed at:

* Providing the necessary inputs to the detailed requirements gathering phase and further on for the SDLC processes.
* This document also serves to establish the traceability between the Business Objectives and the requirements identified in the proposed solution and how they satisfy the stated objectives.
* Provide expectation traceability in terms of the requirements and the user expectation
* Serves as a formal template for documenting the Business Requirements which also includes statutory and regulatory requirements.

The purpose of this document is to systematically capture requirements for the project and the system to be developed. Functional requirements are captured in this document. It also serves as the input for the project scoping.

**CHAPTER-2**

**INTRODUCTION TO PROJECT**

2.1. PURPOSE

Hospitals currently use a manual system for the management and maintenance of critical information. The current 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. Forms are often lost in transit between departments requiring a comprehensive auditing process to ensure that no vital information is lost. Multiple copies of the same information exist in the hospital and may lead to inconsistencies in data in various data stores.

A significant part of the operation of any hospital involves the acquisition, management and timely retrieval of great volumes of information. This information typically involves: patient personal information, patient history, diagnosis details, physician details and searching for the physician. All of this information must be managed in an efficient and cost wise fashion so that an institution’s resources may be effectively utilized HMS will automate the management of the hospital making it more efficient and error free. It aims at standardizing data, consolidating data ensuring, data integrity and reducing inconsistencies.

2.2. OBJECTIVES

Below are the objectives that are fulfilled in this project:

Hospital Management system is a system used for enabling enrollment of patient, add physician, search for physician, update diagnosis details and view Patient history.

* Enroll Patient – Hospital Administrators to register a Patient into the system. The administrator is having the complete details of the patient to be entered into the system.
* Add Physician - Hospital Administrators to register a Physician into the system. The administrator is having the complete details of the physician to be entered into the system including his speciality and qualification details.
* Physician Search - Hospital Administrators search for Physician based on specified search criteria.
* Patient Diagnosis Details - Hospital Administrators to save the patients diagnosis details into the system.
* View Patient History – In case administrator wishes to view the patient case history details, he can retrieve all the past records of the patient.

2.3. INTENDED AUDIENCE

* All members of the project.
* Delivery Assurance Group.

**CHAPTER-3**

**TECHNOLOGIES AND SOFTWARES**

**3.1**. **TECHNOLOGIES**

3.1.1. JAVA

With the invention of microprocessors, the world is scientifically developed with sophisticated equipments, systems, and devices. Microprocessors are used in computers, televisions, and fax machines. Even the hand-held devices such as pagers, PDAs (Personal Digital Assistant), and cell phones make use of microprocessors. All these electronic devices are helpful because of their communication capabilities. With the increasing capabilities and decreasing cost of information processing and networking technologies, the network is growing rapidly for transmitting information through electronic systems. Internet is the network of networks between different types of computers located at different places to transmit information. Information can reach to any place in the world quickly at a cheaper rate through the Internet. Thus, the Internet has made the world a global village for information exchange. The emerging infrastructure of electronic devices and interconnected computer networks create an environment that presents new challenges to software industries. for this emerging computing environment, Java process to be a well – suited programming language. it is found suitable for networked environments involving a great variety of computer and devices. Java has many characteristics that have contributed to its popularity: Platform independence - Many languages are compatible with only one platform. **Java** was specifically designed so that it would run on any computer, regardless if it was running Windows, Linux, Mac, Unix or any of the other operating systems. Simple and easy to use - Java's creators tried to design it so code could be written¬ efficiently and easily. Multi-functional - Java can produce many applications from command-line programs to applets to Swing windows (basically, sophisticated graphical user interfaces). Java does have some drawbacks. Since it has automated garbage collection, it can tend to use more memory than other similar languages. There are often implementation differences on different platforms, which have led to Java being described as a "write once, test everywhere" system. Lastly, since it uses an abstract "virtual machine", a generic Java program doesn't have access to the Native API's on a system directly. None of these issues are fatal, but it can mean that Java isn't an appropriate choice for a particular piece of software.

3.1.2. MAVEN

**Maven** is a powerful project management tool that is based on POM (Project Object Model). It is used for projects build, dependency and documentation. It simplifies the build process like ANT. But it is too much advanced than ANT.

In short terms we can tell maven is a tool that can be used for building and managing any Java-based project. maven make the day-to-day work of Java developers easier and generally help with the comprehension of any Java-based project.

Maven does a lot of helpful task like:

* We can easily build a project using maven.
* We can add jars and other dependencies of the project easily using the help of maven.
* Maven provides project information (log document, dependency list, unit test reports etc.)
* Maven is very helpful for a project while updating central repository of JARs and other dependencies.
* With the help of Maven, we can build any number of projects into output types like the JAR, WAR etc. without doing any scripting.
* Using Maven, we can easily integrate our project with source control system (such as Subversion or Git).

Core Concepts of Maven:

* **POM Files:** Project Object Model(POM) Files are XML file that contains information related to the project and configuration information such as dependencies, source directory, plugin, goals etc. used by Maven to build the project. When you should execute a maven command you give maven a POM file to execute the commands. Maven reads pom.xml file to accomplish its configuration and operations.
* **Dependencies and Repositories:** Dependencies are external Java libraries required for Project and repositories are directories of packaged JAR files. The local repository is just a directory on your machine hard drive. If the dependencies are not found in the local Maven repository, Maven downloads them from a central Maven repository and puts them in your local repository.
* **Build Life Cycles, Phases and Goals:** A build life cycle consists of a sequence of build phases, and each build phase consists of a sequence of goals. Maven command is the name of a build lifecycle, phase or goal. If a lifecycle is requested executed by giving maven command, all build phases in that life cycle are executed also. If a build phase is requested executed, all build phases before it in the defined sequence are executed too.
* **Build Profiles:** Build profiles a set of configuration values which allows you to build your project using different configurations. For example, you may need to build your project for your local computer, for development and test. To enable different builds, you can add different build profiles to your POM files using its profiles elements and are triggered in the variety of ways.
* **Build Plugins:** Build plugins are used to perform specific goal. you can add a plugin to the POM file. Maven has some standard plugins you can use, and you can also implement your own in Java.

3.1.3. SPRING MVC

A **Spring MVC** is a Java framework which is used to build web applications. It follows the Model-View-Controller design pattern. It implements all the basic features of a core spring framework like Inversion of Control, Dependency Injection.

A Spring MVC provides an elegant solution to use MVC in spring framework by the help of DispatcherServlet. Here, DispatcherServlet is a class that receives the incoming request and maps it to the right resource such as controllers, models, and views.

Spring Web Model-View-Controller:

* **Model:** A model contains the data of the application. A data can be a single object or a collection of objects.
* **Controller:** A controller contains the business logic of an application. Here, the @Controller annotation is used to mark the class as the controller.
* **View:** A view represents the provided information in a particular format. Generally, JSP+JSTL is used to create a view page. Although spring also supports other view technologies such as Apache Velocity, Thymeleaf and FreeMarker.
* **Front Controller:** In Spring Web MVC, the DispatcherServlet class works as the front controller. It is responsible to manage the flow of the Spring MVC application.

Advantages of Spring MVC Framework

* **Separate roles:** The Spring MVC separates each role, where the model object, controller, command object, view resolver, DispatcherServlet, validator, etc. can be fulfilled by a specialized object.
* **Light-weight:** It uses light-weight servlet container to develop and deploy your application.
* **Powerful Configuration:** It provides a robust configuration for both framework and application classes that includes easy referencing across contexts, such as from web controllers to business objects and validators.
* **Rapid development:** The Spring MVC facilitates fast and parallel development.
* **Reusable business code:** Instead of creating new objects, it allows us to use the existing business objects.
* **Easy to test:** In Spring, generally we create JavaBeans classes that enable you to inject test data using the setter methods.
* **Flexible Mapping:** It provides the specific annotations that easily redirect the page.

3.1.4. HIBERNATE

**Hibernate** is a framework which provides some abstraction layer means programmer don’t have to worry about the implementations, hibernate do implementations for you internally like Establishing a connection with the database, writing query to perform CRUD operations etc.

It is a java framework which is used to develop persistence logic. Persistence logic means to store and process the data for long use. More precisely Hibernate is a open source, non-invasive, light-weight java ORM (Object relational mapping) framework to develop objects which is independent of the database software and make independent persistence logic in all JAVA, JEE.

Framework means it is a special install-able software that provides abstraction layer on one or more technologies like JDBC, Servlet etc. to simplify or reduce the complexity for development process.

Open Source means:

* Hibernate framework is available for everyone without any cost.
* The source code of Hibernate is also available on the Internet and we can also modify the code.

Light-weight means:

* Hibernate is less in size means the installation package is not big is size.
* Hibernate does not require any heavy container for execution.
* It does not require POJO and POJI model programming.
* Hibernate can be used alone or we can use Hibernate with other java technology and framework.

Non-invasive means:

* The classes of Hibernate application development are loosely coupled classes with respect to Hibernate API i.e. Hibernate class need not to implement hibernate API interfaces and need not to extend from Hibernate API classes.

Functionalities supported by Hibernate framework:

* **Hibernate** framework support Auto DDL operations. In JDBC manually we have to create table and declare the data-type for each and every column. But Hibernate can do DDL operations for you internally like creation of table, drop a table, alter a table etc.
* **Hibernate** supports Auto Primary key generation. It means in JDBC we have to manually set a primary key for a table. But Hibernate can this task for you.
* **Hibernate** framework is independent of Database because it supports HQL (Hibernate Query Language) which is not specific to any database, whereas JDBC is database dependent.
* In **Hibernate**, Exception Handling is not mandatory, whereas in JDBC exception handling is mandatory.
* **Hibernate** supports Cache Memory whereas JDBC does not support cache memory.
* **Hibernate** is a ORM tool means it support Object relational mapping. Whereas JDBC is not object oriented moreover we are dealing with values means primitive data. In hibernate each record is represented as an Object but in JDBC each record is nothing but a data which is nothing but primitive values.

3.1.5. HTML

**Hypertext Markup Language (HTML)** is the standard [markup language](https://en.wikipedia.org/wiki/Markup_language) for creating [web pages](https://en.wikipedia.org/wiki/Web_page) and [web applications](https://en.wikipedia.org/wiki/Web_application). With [Cascading Style Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets) (CSS) and [JavaScript](https://en.wikipedia.org/wiki/JavaScript), it forms a triad of [cornerstone](https://en.wikipedia.org/wiki/Cornerstone) technologies for the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web).

[Web browsers](https://en.wikipedia.org/wiki/Web_browser) receive HTML documents from a [web server](https://en.wikipedia.org/wiki/Web_server) or from local storage and [render](https://en.wikipedia.org/wiki/Browser_engine) the documents into multimedia web pages. HTML describes the structure of a web page [semantically](https://en.wikipedia.org/wiki/Semantic_Web) and originally included cues for the appearance of the document.

[HTML elements](https://en.wikipedia.org/wiki/HTML_element) are the building blocks of HTML pages. With HTML constructs, [images](https://en.wikipedia.org/wiki/HTML_element#Images_and_objects) and other objects such as [interactive forms](https://en.wikipedia.org/wiki/Fieldset) may be embedded into the rendered page. HTML provides a means to create [structured documents](https://en.wikipedia.org/wiki/Structured_document) by denoting structural [semantics](https://en.wikipedia.org/wiki/Semantics) for text such as headings, paragraphs, lists, [links](https://en.wikipedia.org/wiki/Hyperlink), quotes and other items. HTML elements are delineated by tags, written using [angle brackets](https://en.wikipedia.org/wiki/Bracket#Angle_brackets). Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a [scripting language](https://en.wikipedia.org/wiki/Scripting_language) such as [JavaScript](https://en.wikipedia.org/wiki/JavaScript), which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The [World Wide Web Consortium](https://en.wikipedia.org/wiki/World_Wide_Web_Consortium) (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

3.1.6. CSS

**Cascading Style Sheets (CSS)** is a [style sheet language](https://en.wikipedia.org/wiki/Style_sheet_language) used for describing the [presentation](https://en.wikipedia.org/wiki/Presentation_semantics) of a document written in a [markup language](https://en.wikipedia.org/wiki/Markup_language) like [HTML](https://en.wikipedia.org/wiki/HTML). CSS is a cornerstone technology of the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web), alongside HTML and [JavaScript](https://en.wikipedia.org/wiki/JavaScript).

CSS is designed to enable the separation of presentation and content, including [layout](https://en.wikipedia.org/wiki/Page_layout), [colors](https://en.wikipedia.org/wiki/Color), and [fonts](https://en.wikipedia.org/wiki/Typeface). This separation can improve content [accessibility](https://en.wikipedia.org/wiki/Accessibility), provide more flexibility and control in the specification of presentation characteristics, enable multiple [web pages](https://en.wikipedia.org/wiki/Web_page) to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or [screen reader](https://en.wikipedia.org/wiki/Screen_reader)), and on [Braille-based](https://en.wikipedia.org/wiki/Braille_display) tactile devices. CSS also has rules for alternate formatting if the content is accessed on a [mobile device](https://en.wikipedia.org/wiki/Mobile_device).

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the [World Wide Web Consortium](https://en.wikipedia.org/wiki/World_Wide_Web_Consortium) (W3C). Internet media type ([MIME type](https://en.wikipedia.org/wiki/MIME_media_type)) text/css is registered for use with CSS by [RFC 2318](https://tools.ietf.org/html/rfc2318) (March 1998). The W3C operates a free [CSS validation service](https://en.wikipedia.org/wiki/W3C_Markup_Validation_Service#CSS_validation) for CSS documents.

In addition to HTML, other markup languages support the use of CSS including [XHTML](https://en.wikipedia.org/wiki/XHTML), [plain XML](https://en.wikipedia.org/wiki/Plain_Old_XML), [SVG](https://en.wikipedia.org/wiki/Scalable_Vector_Graphics), and [XUL](https://en.wikipedia.org/wiki/XUL).

3.1.7. JAVASCRIPT

**JavaScript** often abbreviated as JS, is a [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), [interpreted](https://en.wikipedia.org/wiki/Interpreted_language) [programming language](https://en.wikipedia.org/wiki/Programming_language) that conforms to the [ECMAScript](https://en.wikipedia.org/wiki/ECMAScript) specification. It is a programming language that is characterized as [dynamic](https://en.wikipedia.org/wiki/Dynamic_programming_language), [weakly typed](https://en.wikipedia.org/wiki/Weak_typing), [prototype-based](https://en.wikipedia.org/wiki/Prototype-based_programming) and [multi-paradigm](https://en.wikipedia.org/wiki/Multi-paradigm_programming_language).

Alongside [HTML](https://en.wikipedia.org/wiki/HTML) and [CSS](https://en.wikipedia.org/wiki/CSS), JavaScript is one of the core technologies of the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web). JavaScript enables interactive [web pages](https://en.wikipedia.org/wiki/Web_page) and is an essential part of [web applications](https://en.wikipedia.org/wiki/Web_application). The vast majority of [websites](https://en.wikipedia.org/wiki/Website) use it, and major [web browsers](https://en.wikipedia.org/wiki/Web_browser) have a dedicated [JavaScript engine](https://en.wikipedia.org/wiki/JavaScript_engine) to execute it.

As a multi-paradigm language, JavaScript supports [event-driven](https://en.wikipedia.org/wiki/Event-driven_programming), [functional](https://en.wikipedia.org/wiki/Functional_programming), and [imperative](https://en.wikipedia.org/wiki/Imperative_programming) (including [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) and [prototype-based](https://en.wikipedia.org/wiki/Prototype-based_programming)) [programming styles](https://en.wikipedia.org/wiki/Programming_paradigm). It has [APIs](https://en.wikipedia.org/wiki/Application_programming_interface) for working with text, [arrays](https://en.wikipedia.org/wiki/Array_data_type), dates, [regular expressions](https://en.wikipedia.org/wiki/Regular_expression), and the [DOM](https://en.wikipedia.org/wiki/Document_Object_Model), but the language itself does not include any [I/O](https://en.wikipedia.org/wiki/Input/output), such as [networking](https://en.wikipedia.org/wiki/Computer_network), [storage](https://en.wikipedia.org/wiki/Data_storage), or [graphics](https://en.wikipedia.org/wiki/Computer_graphics) facilities. It relies upon the host environment in which it is embedded to provide these features.

Initially only implemented [client-side](https://en.wikipedia.org/wiki/Client-side) in web browsers, JavaScript engines are now embedded in many other types of host software, including [server-side](https://en.wikipedia.org/wiki/Server-side) in web servers and databases, and in non-web programs such as word processors and [PDF](https://en.wikipedia.org/wiki/Portable_Document_Format) software, and in runtime environments that make JavaScript available for writing mobile and desktop applications, including desktop widgets.

The terms Vanilla JavaScript and Vanilla JS refer to JavaScript not extended by any frameworks or additional libraries. Scripts written in Vanilla JS are plain JavaScript code.

Although there are similarities between JavaScript and [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), including language name, [syntax](https://en.wikipedia.org/wiki/Syntax_(programming_languages)), and respective [standard libraries](https://en.wikipedia.org/wiki/Standard_library), the two languages are distinct and differ greatly in design. JavaScript was influenced by programming languages such as [Self](https://en.wikipedia.org/wiki/Self_(programming_language)) and [Scheme](https://en.wikipedia.org/wiki/Scheme_(programming_language)).

3.1.8. MOCKITO

**Mockito** is an [open source](https://en.wikipedia.org/wiki/Open_source) [testing framework](https://en.wikipedia.org/wiki/List_of_unit_testing_frameworks) for [Java](https://en.wikipedia.org/wiki/Java_(software_platform)) released under the [MIT License](https://en.wikipedia.org/wiki/MIT_License). The framework allows the creation of [test double](https://en.wikipedia.org/wiki/Test_double) objects ([mock objects](https://en.wikipedia.org/wiki/Mock_object)) in [automated unit tests](https://en.wikipedia.org/wiki/Test_automation) for the purpose of [test-driven development](https://en.wikipedia.org/wiki/Test-driven_development) (TDD) or [behavior-driven development](https://en.wikipedia.org/wiki/Behavior-driven_development) (BDD).

The framework's name and logo are a play on [mojitos](https://en.wikipedia.org/wiki/Mojito), a type of drink.

Features:

**Mockito** allows developers to verify the behavior of the [system under test](https://en.wikipedia.org/wiki/System_under_test) (SUT) without establishing expectations beforehand. One of the criticisms of [mock objects](https://en.wikipedia.org/wiki/Mock_Object) is that there is a tight coupling of the test code to the system under test. Mockito attempts to eliminate the expect-run-verify patternby removing the specification of expectations. Mockito also provides some annotations for reducing [boilerplate code](https://en.wikipedia.org/wiki/Boilerplate_code).

3.1.9. JUNIT

**JUnit** is a [unit testing](https://en.wikipedia.org/wiki/Unit_testing) [framework](https://en.wikipedia.org/wiki/Software_framework) for the [Java programming language](https://en.wikipedia.org/wiki/Java_(programming_language)). JUnit has been important in the development of [test-driven development](https://en.wikipedia.org/wiki/Test-driven_development), and is one of a family of [unit testing](https://en.wikipedia.org/wiki/Unit_testing) frameworks which is collectively known as [xUnit](https://en.wikipedia.org/wiki/XUnit) that originated with [SUnit](https://en.wikipedia.org/wiki/SUnit).

JUnit is linked as a [JAR](https://en.wikipedia.org/wiki/JAR_(file_format)) at compile-time; the framework resides under package junit.framework for JUnit 3.8 and earlier, and under package org.junit for JUnit 4 and later.

A research survey performed in 2013 across 10,000 Java projects hosted on GitHub found that JUnit (in a tie with [slf4j-api](https://en.wikipedia.org/wiki/SLF4J)), was the most commonly included external library. Each library was used by 30.7% of projects.

3.1.10. LOGGER

Logger is a class in the org.apache.log4j package. We have to initialize one Logger object for each Java class. We use Logger’s methods to generate log statements. Log4j provides the factory method to get Logger objects.

The Logger class has some methods that are used to print application status.

We have five methods in the Logger class

* info()
* debug()
* warn()
* fatal()
* error()

**3.2.** **SOFTWARES**

3.2.1 APACHE TOMCAT

**Apache Tomcat**, often referred to as Tomcat Server, is an open-source [Java Servlet Container](https://en.wikipedia.org/wiki/Servlet_container) developed by the [Apache Software Foundation](https://en.wikipedia.org/wiki/Apache_Software_Foundation) (ASF). Tomcat implements several [Java EE](https://en.wikipedia.org/wiki/Java_Platform,_Enterprise_Edition) specifications including [Java Servlet](https://en.wikipedia.org/wiki/Java_Servlet), [JavaServer Pages](https://en.wikipedia.org/wiki/JavaServer_Pages) (JSP), [Java EL](https://en.wikipedia.org/wiki/Unified_Expression_Language), and [WebSocket](https://en.wikipedia.org/wiki/WebSocket), and provides a "pure [Java](https://en.wikipedia.org/wiki/Java_(programming_language))" [HTTP](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) [web server](https://en.wikipedia.org/wiki/Web_server) environment in which [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) code can run.

Tomcat is developed and maintained by an open community of developers under the auspices of the [Apache Software Foundation](https://en.wikipedia.org/wiki/Apache_Software_Foundation), released under the [Apache License](https://en.wikipedia.org/wiki/Apache_License) 2.0 license, and is [open-source software](https://en.wikipedia.org/wiki/Open-source_software).

**Web** **Application**:

It has also added user- as well as system-based web applications enhancement to add support for deployment across the variety of environments. It also tries to manage sessions as well as applications across the network.

Tomcat is building additional components. A number of additional components may be used with Apache Tomcat. These components may be built by users should they need them or they can be downloaded from one of the mirrors.

**Features**:

Tomcat 7.x implements the Servlet 3.0 and JSP 2.2 specifications. It requires Java version 1.6, although previous versions have run on Java 1.1 through 1.5. Versions 5 through 6 saw improvements in [garbage collection](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)), JSP parsing, performance and scalability. Native wrappers, known as "Tomcat Native", are available for [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) and Unix for platform integration.

Tomcat 8.x implements the Servlet 3.1 and JSP 2.3 Specifications. Apache Tomcat 8.5.x is intended to replace 8.0.x and includes new features pulled forward from Tomcat 9.0.x. The minimum Java version and implemented specification versions remain unchanged.

**History**:

Tomcat started off as a servlet [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation_(computing)) by [James Duncan Davidson](https://en.wikipedia.org/wiki/James_Duncan_Davidson), a software architect at Sun Microsystems. He later helped make the project [open source](https://en.wikipedia.org/wiki/Open-source_software) and played a key role in its donation by Sun Microsystems to the Apache Software Foundation. The [Apache Ant](https://en.wikipedia.org/wiki/Apache_Ant) software build automation tool was developed as a side-effect of the creation of Tomcat as an open source project.

Davidson had initially hoped that the project would become open sourced and, since many open source projects had [O'Reilly](https://en.wikipedia.org/wiki/O%27Reilly_Media) books associated with them featuring an animal on the cover, he wanted to name the project after an animal. He came up with [Tomcat](https://en.wikipedia.org/wiki/Cat) since he reasoned the animal represented something that could fend for itself. Although the tomcat was already in use for another O'Reilly title, his wish to see an animal cover eventually came true when O'Reilly published their Tomcat book with a [snow leopard](https://en.wikipedia.org/wiki/Snow_leopard) on the cover in 2003.

3.2.2. ORACLE

Oracle Database 11g Express Edition (Oracle Database XE) is an entry-level, small-footprint database based on the Oracle Database 11g Release 2 code base.  It's free to develop, deploy, and distribute; fast to download; and simple to administer.

Oracle Database XE is a great starter database for:

* **Developers** working on Node.js, Python, PHP, Java, .NET, XML, and Open Source applications.
* **DBAs** who need a free, starter database for training and deployment.
* **Independent Software Vendors (ISVs) and hardware vendors** who want a starter database to distribute free of charge.
* **Educational institutions and students** who need a free database for their curriculum.

With Oracle Database XE, you can now develop and deploy applications with a powerful, proven, industry-leading infrastructure, and then upgrade when necessary without costly and complex migrations.

Oracle Database XE can be installed on any size host machine with any number of CPUs (one database per machine), but XE will store up to 11GB of user data, use up to 1GB of memory, and use one CPU on the host machine.

3.2.3. ECLIPSE

Eclipse is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) used in [computer programming](https://en.wikipedia.org/wiki/Computer_programming), and is the most widely used Java IDE. It contains a base [workspace](https://en.wikipedia.org/wiki/Workspace) and an extensible [plug-in](https://en.wikipedia.org/wiki/Plug-in_(computing)) system for customizing the environment. Eclipse is written mostly in [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) and its primary use is for developing Java applications, but it may also be used to develop applications in other programming languages via plug-ins, including Ada, ABAP, C, C++, C#, Clojure, COBOL, D, Erlang, Fortran, Groovy, Haskell, JavaScript, Julia, Lasso, Lua, NATURAL, Perl, PHP, Prolog, Phython, R, Ruby (including Ruby on Rails framework), Rust, Scala and Scheme. It can also be used to develop documents with [LaTeX](https://en.wikipedia.org/wiki/LaTeX) (via a TeXlipse plug-in) and packages for the software [Mathematica](https://en.wikipedia.org/wiki/Mathematica). Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++, and Eclipse PDT for PHP, among others.

The initial [codebase](https://en.wikipedia.org/wiki/Codebase) originated from [IBM VisualAge](https://en.wikipedia.org/wiki/IBM_VisualAge). The Eclipse [software development kit](https://en.wikipedia.org/wiki/Software_development_kit) (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules. Since the introduction of the [OSGi](https://en.wikipedia.org/wiki/OSGi) implementation ([Equinox](https://en.wikipedia.org/wiki/Equinox_(OSGi))) in version 3 of Eclipse, plug-ins can be plugged-stopped dynamically and are termed (OSGI) bundles.

Eclipse [software development kit](https://en.wikipedia.org/wiki/Software_development_kit) (SDK) is [free and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software), released under the terms of the [Eclipse Public License](https://en.wikipedia.org/wiki/Eclipse_Public_License), although it is incompatible with the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License).[[10]](https://en.wikipedia.org/wiki/Eclipse_(software)#cite_note-10) It was one of the first IDEs to run under [GNU Classpath](https://en.wikipedia.org/wiki/GNU_Classpath) and it runs without problems under [IcedTea](https://en.wikipedia.org/wiki/IcedTea).

**CHAPTER-4**

**PROJECT OVERVIEW**

4.1. PROBLEM INTRODUCTON

**Lack of Immediate Retrievals:**

The information is very difficult to retrieve and to find particular information like- E.g. to find out about the patient’s history, the user has to go through various registers. This results in inconvenience and wastage of time.

**Lack of Immediate Information Storage:**

The information generated by various transactions takes time and efforts to be stored at the right place.

**Lack of Prompt Updating:**

Various changes to information like patient details or immunization details of patient are difficult to make as paper work is involved.

**Preparation of Accurate and Prompt Reports:**

This becomes a difficult task as information is difficult to collect from various registers.

4.2. BUSINESS CASE

A comprehensive healthcare management is necessary to drive a thriving enterprise. Hospital management needs to provide a robust framework to assure seamless daily operations at the enterprise level, which integrates the data from different departments within the hospital.

The solution developed will address the objective in a holistic manner and will have all the features and functionalities which shall let the hospital be able to record & assimilate data from the patients, doctors, lab diagnosis and correlate the data with the patient’s visit history. Parameter based search algorithm & transfer of patient records are scalable features.

4.3. PROJECT SCOPE

* Administrator fills up all the details of the patient as well as the physician in the system. His login is mandatory to view, modify or delete any data. If we have a new administrator then he will have to register himself into the system with all his details filled including his first name, last name, age, gender, password and email id.
* Patient is registered in the hospital by the administrator and his complete information including his first name, last name, age, gender, contact, email, address, city state and zip code is stored in the database. Whenever a new patient comes up his information is stored freshly.
* The same way physician is registered in the hospital by the administrator and his complete information including his first name, last name, age, gender, contact, email, address, city, state, zip code, degree, specialty and work hours are stored in the database. Whenever a new physician comes up in the hospital his information is stored freshly.
* Diagnosis details of the patients are stored in the system, so as if in any case he/she misplaces the report, the hospital has the record for his/her further treatment. This is the complete report of the patient’s health condition.
* Patient history can be viewed; this includes all the previous treatments the patient has undergone.

All the work is done manually by the administrator. Patient and the doctors don’t need to remember the treatment patient has taken and the previous records, everything is kept safe in the system to prevent it from loss.

4.4. MODULES

The entire project mainly consists of modules:

* Admin Module
* Patient Module
* Physician Module
* Diagnosis Module
* Patient History Module

**Admin Module**

* Register Patients.
* Register Physicians.
* View Patient History.
* View Diagnosis Details.
* Edit Patient Record.
* Edit Physician Record.
* Enter Diagnosis Details after every Treatment.
* Carefully save all the data into the system.

**Patient Module**

* Detailed patient Information.
* From here we can view and edit patient records.

**Physician Module**

* Detailed physician Information.
* From here we can view and edit physician records.

**Diagnosis Module**

* Contains patient Diagnosis Details.
* Diagnosis details can be viewed or edited from here.
* Detailed information for any future need.

**Patient History Module**

* Contains patient record.
* All the previous visits with his/her diagnosis details will be shown.

4.5. REQUIREMENT ENGINEERING

Systematic requirements analysis is also known as requirements engineering. It is sometimes referred to loosely by names such as requirements gathering, requirements capture, or requirements specification. The term requirements analysis can also be applied specifically to the analysis proper, as opposed to elicitation or documentation of the requirements, for instance. Requirements engineering can be divided into discrete chronological steps:

* Requirements elicitation,
* Requirements analysis and negotiation,
* Requirements specification,
* System modeling,
* Requirements validation,
* Requirements management.

Systems engineering and software engineering that is concerned with determining the goals, functions, and constraints of hardware and software systems. In some life cycle models, the requirement engineering process begins with a feasibility study activity, which leads to a feasibility report. If the feasibility study suggests that the product should be developed, then requirement analysis can begin. The requirement engineering provides the appropriate mechanism for Requirement engineering according to Laplante (2007) is "a sub discipline of understanding what customer wants.

**CHAPTER-5**

**BUSINESS REQUIREMENTS**

5.1. HIGH LEVEL BUSINESS REQUIREMENTS

|  |  |  |  |
| --- | --- | --- | --- |
| S. No. | Business Requirement ID | Short Description | Description in Detail |
| 1. | Req\_1 | Admin Registration | Ability of the system to procure the fundamental details of the admin. |
| 2. | Req\_2 | Admin Authentication | Ability of the system to authenticate the credentials of the registered Admin. |
| 3. | Req\_3 | Create Patient Records | Ability of the system to ADD new Patient. Update the patient records according to the need of the patient in case any detail is filled incorrectly. |
| 4. | Req\_4 | Create Physician Records | Ability of the system to ADD new Physician. Update the physician records according to the need of the physician in case any detail is filled incorrectly. |
| 5. | Req\_5 | Search Physician | Ability of the system to set amenities for the create contract. |
| 6. | Req\_6 | Patient Diagnosis Details | Ability of the system to UPDATE the patient Diagnosis details. |
| S. No. | **Business Requirement ID** | **Short Description** | **Description in Detail** |
| 7. | Req\_7 | View Patient History | Ability of the system to view complete Patient History. |
| 8. | Req\_8 | Admin Log Off | Ability of the system to enable Admin to Log Off. |

5.2. DETAILED BUSINESS REQUIREMENTS

**5.2.1**. **Functional Requirements**

The Requirements in this document are prioritized as follows:

|  |  |  |
| --- | --- | --- |
| **Value** | **Rating** | **Description** |
| 1 | Critical | This requirement is critical to the success of the project. The project will not be possible without this requirement. |
| 2 | High | This requirement is high priority, but the project can be implemented at a bare minimum without this requirement. |
| 3 | Medium | This requirement is somewhat important, as it provides some value but the project can proceed without it. |
| 4 | Low | This is a low priority requirement, or a “nice to have” feature, if time and cost allow it. |
| 5 | Future | This requirement is out of scope for this project, and has been included here for a possible future release. |

Detailed Description Table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Req.** | **Rationale Categorization** | **Business Requirement** | **Priority** | **BR Traced to Business Requirement / Use case ID** | **Impacted Stakeholders** |
| 1.1. | Admin  Registration | When the Admin clicks on the registration link, it should re-direct to registration form. | Critical | Req\_1 |  |
| **Req.** | **Rationale Categorization** | **Business Requirement** | **Priority** | **BR Traced to Business Requirement / Use case ID** | **Impacted Stakeholders** |
| 1.2. | Admin  Registration | Admin needs to fill some of the basic attributes/fields as  mentioned below in requirement: First Name, Last Name,  Age, Gender, Contact Number, Vendor Id,  Password. | Critical | Req\_1 | Please refer to  Table 7.1. |
| 1.3. | Admin  Registration | Clicking ‘Submit’ should validate the datatype constraints for each field. | Critical | Req\_1 |  |
| 1.4. | Admin  Registration | Admin failing to provide information on the mandatory  fields be provided with an alert message – ‘Please update the  highlighted mandatory field(s).’ Highlight the missed out field in red. | Medium | Req\_1 |  |
| 1.5. | Admin  Registration | Post-successful field level validation, save the information in the database. | Critical | Req\_1 |  |
| 1.6. | Admin  Registration | Upon saving the information in the  database, display  the message ‘Your details are submitted successfully’. | Medium | Req\_1 |  |
| **Req.** | **Rationale Categorization** | **Business Requirement** | **Priority** | **BR Traced to Business Requirement / Use case ID** | **Impacted Stakeholders** |
| 2.1. | Credential Authentication | A registered user – is able to click ‘Login’ link, after keying in  ‘Admin ID’ & ‘Password’ field and get his credentials  authenticated with the existing database entry. | Critical | Req\_2 |  |
| 3.1. | Create Patient records | On clicking “Create Patient Record”, it should redirect Admin to the Patient Record  Creation Page. | Critical | Req\_3 |  |
| 3.2. | Create Patient records | Admin should be able to update the mandatory fields of the Patient. | Critical | Req\_3 | Please refer to  Table 7.2. |
| 3.3. | Create Patient records | Clicking ‘Submit’ should validate the datatype constraints for each field. | Critical | Req\_3 |  |
| 3.4. | Create Patient records | Admin failing to provide information on the mandatory fields be provided with an alert message – ‘Please update the highlighted mandatory field(s).’ Also, highlight the missed out field in red. | Critical | Req\_3 |  |
| 3.5. | Create Patient records | Post-successful field level validation, save the information in the database. | Critical | Req\_3 |  |
| **Req.** | **Rationale Categorization** | **Business Requirement** | **Priority** | **BR Traced to Business Requirement / Use case ID** | **Impacted Stakeholders** |
| 3.6. | Create Patient records | Upon saving the information in the database, display the message ‘Your details are submitted successfully’. | Critical | Req\_3 |  |
| 4.1. | Create Physician Records | On clicking “Create Physician Record”, it should redirect Admin to the Physician Record Creation Page. | Critical | Req\_4 |  |
| 4.2. | Create  Physician  Records | Admin should be able to update the mandatory fields of the Physician. | Critical | Req\_4 | Please refer to  Table 7.3. |
| 4.3. | Create  Physician  Records | Clicking ‘Submit’ should validate the datatype constraints for each field. | Critical | Req\_4 |  |
| 4.4. | Create  Physician  Records | Admin failing to provide information on the mandatory fields be provided with an alert message – ‘Please update the highlighted mandatory field(s).’ Also, highlight the missed out field in red. | Critical | Req\_4 |  |
| 4.5. | Create  Physician  Records | Post-successful  field level validation, save the information in the database. | Critical | Req\_4 |  |
| **Req.** | **Rationale Categorization** | **Business Requirement** | **Priority** | **BR Traced to Business Requirement / Use case ID** | **Impacted Stakeholders** |
| 4.6. | Create  Physician  Records | Upon saving the information in the database, display the message ‘Your details are submitted successfully’. | Critical | Req\_4 |  |
| 5.1. | Search Physician | When Admin clicks the link “Search  Physician”, it should display all the Physicians present in the Database. | Critical | Req\_5 |  |
| 5.2. | Search Physician | When Admin clicks the link “Search  Physician”, and with specific search criteria, then it should display the list of Physicians satisfying the search criteria. | Critical | Req\_5 |  |
| 6.1. | Patient  Diagnosis  Details | When Admin clicks  “Add Patient Diagnosis Details” link, Admin should be redirected to creation page. | Critical | Req\_6 |  |
| 6.2. | Patient  Diagnosis Details | Admin should be able to update the mandatory fields of the Patient. | Critical | Req\_6 | Please refer to  Table 7.4. |
| 6.3. | Patient  Diagnosis Details | Clicking ‘Submit’ should validate the datatype constraints for each field. | Critical | Req\_6 |  |
| **Req.** | **Rationale Categorization** | **Business Requirement** | **Priority** | **BR Traced to Business Requirement / Use case ID** | **Impacted Stakeholders** |
| 6.4. | Patient  Diagnosis Details | Admin failing to provide information on the mandatory fields be provided with an alert message – ‘Please update the highlighted mandatory field(s).’ Also, highlight the missed out field in red. | Critical | Req\_6 |  |
| 6.5. | Patient  Diagnosis Details | Post-successful  field level validation, save the information in the database. | Critical | Req\_6 |  |
| 6.6. | Patient  Diagnosis Details | Upon saving the information in the database, display the message ‘Your details are submitted successfully’. | Critical | Req\_6 |  |
| 7.1. | View Patient  History | When Admin clicks  “View Patient  History” link, the system should list all the available patient ID and patient name. | Critical | Req\_7 |  |
| 7.2. | View Patient  History | On clicking the  Patient ID, it should display the Patient History. | Critical | Req\_7 |  |
| 8.1. | Admin Log off | Option to log off from the system. | Critical | Req\_8 |  |

**CHAPTER-6**

**ANALYSIS**

6.1. HARDWARE REQUIREMENTS

* Desktop PC with 8GB RAM.

6.2. SOFTWARE REQUIREMENTS

* Eclipse IDE for Java EE Developers (Oxygen)
* Maven 3.6.0
* Tomcat 8.0
* Oracle 11g express version

**CHAPTER-7**

**TABLES**

7.1. ADMIN TABLE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** **Name** | **Field Type** | **Data Type** | **Mandatory** | **Possible Values** |
| Admin ID | Text(10) | Alphabetic | Yes | System Generated |
| First Name | Text(50) | Alphabetic | Yes |  |
| Last Name | Text(50) | Alphabetic | Yes |  |
| Age | Numeric(2) | Numeric | Yes |  |
| Gender | Drop Down | NA |  | Male,  Female |
| DOB | Text(10) | Alphanumeric | Yes |  |
| Contact Number | Text(10) | Numeric | Yes |  |
| Alt Contact Number | Text(10) | Numeric | No |  |
| Email ID | Text(50) | Alphanumeric | Yes |  |

7.2. PATIENT TABLE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** **Name** | **Field Type** | **Data Type** | **Mandatory** | **Possible Values** |
| Patient ID | Text(10) | Alphanumeric | Yes | System Generated |
| First Name | Text(50) | Alphabetic | Yes |  |
| Last Name | Text(50) | Alphabetic | Yes |  |
| Age | Numeric(2) | Numeric | Yes |  |
| Gender | Drop Down | NA |  | Male,  Female |
| DOB | Text(10) | Alphanumeric | Yes |  |
| Contact Number | Text(10) | Numeric | Yes |  |
| Alt Contact Number | Text(10) | Numeric | No |  |
| Email ID | Text(50) | Alphanumeric | Yes |  |
| Password | Text(15) | Alphanumeric | Yes |  |
| Address Line 1 | Text(100) | Alphanumeric | Yes |  |
| Address Line 1 | Text(100) | Alphabetic | Yes |  |
| City | Text(50) | Alphabetic | Yes |  |
| State | Text(50) | Alphabetic | Yes |  |
| Zip Code | Text(10) | Numeric | Yes |  |

7.3. PHYSICIAN TABLE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Data Type** | **Mandatory** | **Possible Values** |
| Physician ID | Text(10) | Alphanumeric | Yes | System Generated |
| First Name | Text(50) | Alphabetic | Yes |  |
| Last Name | Text(50) | Alphabetic | Yes |  |
| **Field Name** | **Field Type** | **Data Type** | **Mandatory** | **Possible Values** |
| Age | Numeric(2) | Numeric | Yes |  |
| Gender | Drop Down | NA |  | Male,  Female |
| DOB | Text(10) | Alphanumeric | No |  |
| Contact Number | Text(10) | Numeric | Yes |  |
| Alt Contact Number | Text(10) | Numeric | No |  |
| Email ID | Text(50) | Alphanumeric | Yes |  |
| Password | Text(15) | Alphanumeric | Yes |  |
| Address Line 1 | Text(100) | Alphanumeric | Yes |  |
| Address Line 2 | Text(100) | Alphabetic | No |  |
| City | Text(50) | Alphabetic | Yes |  |
| State | Text(50) | Alphabetic | Yes |  |
| Zip Code | Text(10) | Numeric | Yes |  |
| Degree | Text(50) | Alphabetic | Yes |  |
| Specialty | Text(50) | Alphabetic | Yes |  |
| Work hours | Drop down | NA | Yes |  |
| Hospital/Clinic  Name | Text(100) | Alphanumeric | Yes |  |

7.4. DIAGNOSIS DETAILS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Data Type** | **Mandatory** | **Possible Values** |
| Report ID | Numeric(10) | Numeric | Yes | System Generated |
| Patient ID | Text(10) | Alphanumeric | Yes | System Generated |
| Physician ID | Text(10) | Alphanumeric | Yes | System Generated |
| **Field Name** | **Field Type** | **Data Type** | **Mandatory** | **Possible Values** |
| Service date | Text(10) | Alphanumeric | Yes |  |
| Test  Result date | Text(10) | Alphanumeric | Yes |  |
| Diag 1 –  Actual  Value | Numeric(10) | Numeric | Yes |  |
| Diag 1 –  Normal  Range | Numeric(10) | Numeric | Yes |  |
| Diag 2 –  Actual  Value | Numeric(10) | Numeric | No |  |
| Diag 2 –  Normal  Range | Numeric(10) | Numeric | No |  |
| Diag 3 –  Actual  Value | Numeric(10) | Numeric | No |  |
| Diag 3 –  Normal  Range | Numeric(10) | Numeric | No |  |
| Diag 4 –  Actual  Value | Numeric(10) | Numeric | No |  |
| Diag 4 –  Normal  Range | Numeric(10) | Numeric | No |  |
| Diag 5 –  Actual  Value | Numeric(10) | Numeric | No |  |
| Diag 5 –  Normal  Range | Numeric(10) | Numeric | No |  |
| Diag 6 –  Actual  Value | Numeric(10) | Numeric | No |  |
| **Field Name** | **Field Type** | **Data Type** | **Mandatory** | **Possible Values** |
| Diag 6 –  Normal  Range | Numeric(10) | Numeric | No |  |
| Physician  Comments | Text(300) | Alphabetic | Yes |  |
| Other info | Text(300) | Alphabetic | No |  |

**CHAPTER-8**

**DESIGN**

**CONCLUSION:**

* Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system which is more user friendly……….
* We can improve the efficiency of the system, thus overcome the drawbacks of the existing system

**REFERENCES:**

* www.google.co.in(Internet Search Engine)
* www.youtube.com (internet website)
* www.stackoverflow.com
* www.java2s.com