**PATIENT TRACKER SYSTEM**

A Report for Evaluation

SUBMITTED BY

**NIKHIL OMAR** (768790)

Under the Supervision of

**Mr. Sabbir Poonawala**



**School of Computing Science and Engineering**

**Greater Noida, Uttar Pradesh**

**April, 2019**

**Dissertation Approval Sheet**

The dissertation entitled **“Patient Tracker System**”submitted by **Nikhil Omar** is approved as partial fulfilment for the award of **Bachelor of Engineering in Computing Science & Engineering** degree by **Galgotias University, Greater Noida.**

**Internal Examiner External Examiner**

**Director**

**Galgotias University,**

**Greater Noida (Uttar Pradesh)**

**Candidate Declaration**

We hereby declare that the work which is being presented in this project entitled Patient Tracker System in partial fulfilment of degree of Bachelor of Engineering in Computer Science & Engineering is an authentic record of our own work carried out under the supervision and guidance of **Sabbir Poonawala**, **Trainer.**

We are 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:**

**Nikhil Omar**

**CERTIFICATE**

**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 our college vice-chancellor Mrs. Renu Luthra for giving us such an amazing opportunity to work 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.

**ABSTRACT**

A comprehensive Patient Tracker System is necessary to drive a thriving medical enterprise such as hospital. It needs to provide a robust framework to assure seamless daily operations at the organizational 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 administrator to be able to record & assimilate data from the patients, doctors, clerk, medicine and patient prescription. Alongside such functionalities, it also allows them to generate as well as update the medical bills of the patients and hence in this way, maintain the database of all the stakeholders of the hospital.

**TABLE OF CONTENTS Page No**

**Dissertation Approval Sheet** ii

**Candidate Declaration** iii

**Certificate** iv

**Acknowledgements** v

**Abstract** vi

**Chapter 1 Introduction**

1.1 Purpose of this document viii

1.2 Project Overview viii

**Chapter 2 Literature Survey**

2.1 Methodology xi

2.2 Technologies and Tools xii

**Chapter 3 Analysis**

3.1 Software Requirements xv

3.2 Hardware Requirements xv

**Chapter 4 Design**

4.1 Diagrams xvii

4.2 Tables xx

**Chapter 5 Conclusion** xxii

**References** xxiii

**CHAPTER 1**

**INTRODUCTION**

**Introduction**

**1.1 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.

**1.2 Project Overview**

**1.2.1 Objectives**

Administrator

Below are the objectives of Administrator Module:

* User registration & credential authentication
* Addition of new entries in Doctor, Patient, Clerk and Medicine
* Updation of stored data in Doctor, Patient, Clerk and Medicine modules
* Generation of bill for the patient on the basis of prescription
* Store and maintain data of each entity in the database
* ID for new Doctor, Clerk, Patient, Medicine, Request are auto-generated by the system and all other entities are mapped to it.

**CHAPTER 2**

**LITERARY SURVEY**

1. **Literary Survey**

**2.1 Methodology**

* SDLC:

In [software engineering](https://en.wikipedia.org/wiki/Software_engineering), a software development process is the process of dividing [software development](https://en.wikipedia.org/wiki/Software_development) work into distinct phases to improve [design](https://en.wikipedia.org/wiki/Software_design), [product management](https://en.wikipedia.org/wiki/Software_product_management), and [project management](https://en.wikipedia.org/wiki/Software_project_management). It is also known as a **software development life cycle**. The methodology may include the pre-definition of specific [deliverables](https://en.wikipedia.org/wiki/Deliverable) and artifacts that are created and completed by a project team to develop or maintain an application.

Most modern development processes can be vaguely described as [**agile**](https://en.wikipedia.org/wiki/Agile_software_development). Other methodologies include [*waterfall*](https://en.wikipedia.org/wiki/Waterfall_model)*,*[*prototyping*](https://en.wikipedia.org/wiki/Software_prototyping)*,*[*iterative and incremental development*](https://en.wikipedia.org/wiki/Iterative_and_incremental_development)*,*[*spiral development*](https://en.wikipedia.org/wiki/Spiral_development)*,*[*rapid application development*](https://en.wikipedia.org/wiki/Rapid_application_development)*, and*[*extreme programming*](https://en.wikipedia.org/wiki/Extreme_programming)*.*

Some people consider a life-cycle "model" a more general term for a category of methodologies and a software development "process" a more specific term to refer to a specific process chosen by a specific organization.

* Agile:

"Agile software development" refers to a group of software development methodologies based on iterative development, where requirements and solutions evolve via collaboration between self-organizing cross-functional teams

Agile software development uses iterative development as a basis but advocates a lighter and more people-centric viewpoint than traditional approaches. Agile processes fundamentally incorporate iteration and the continuous feedback that it provides to successively refine and deliver a software system.

There are many agile methodologies, including:

* [Dynamic systems development method](https://en.wikipedia.org/wiki/Dynamic_systems_development_method) (DSDM)
* [Kanban](https://en.wikipedia.org/wiki/Kanban_(development))
* [Scrum](https://en.wikipedia.org/wiki/Scrum_(development))
* Client–server model:

**Client–server model** is a [distributed application](https://en.wikipedia.org/wiki/Distributed_application) structure that partitions tasks or workloads between the providers of a resource or service, called [servers](https://en.wikipedia.org/wiki/Server_(computing)), and service requesters, called [clients](https://en.wikipedia.org/wiki/Client_(computing)).[[1]](https://en.wikipedia.org/wiki/Client%E2%80%93server_model#cite_note-1) Often clients and servers communicate over a [computer network](https://en.wikipedia.org/wiki/Computer_network) on separate hardware, but both client and server may reside in the same system. A server [host](https://en.wikipedia.org/wiki/Host_(network)) runs one or more server programs which share their resources with clients. A client does not share any of its resources, but requests a server's content or service function. Clients therefore initiate communication sessions with servers which await incoming requests.

**2.2Technology and Tools**

**Front End:**

**Java (HTML, CSS, JavaScript)**

**HTML:**

* **Hypertext Mark-up Language** (**HTML**) is the standard [markup language](https://en.wikipedia.org/wiki/Markup_language" \o "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.

**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" \o "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" \o "Color), and [fonts](https://en.wikipedia.org/wiki/Typeface).[[3]](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-3) 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.

**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.

**Middleware:**

**Java (Java Servlet, JDBC)**

**Java Servlet:**

* A Java servlet processes or stores a [Java class](https://en.wikipedia.org/wiki/Java_class) in [Java EE](https://en.wikipedia.org/wiki/Java_EE) that conforms to the Java Servlet API,,a standard for implementing Java classes that respond to requests. Servlets could in principle communicate over any [client–server](https://en.wikipedia.org/wiki/Client%E2%80%93server_model) protocol, but they are most often used with the [HTTP](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol).
* Thus "servlet" is often used as shorthand for "HTTP servlet". Thus, a [software developer](https://en.wikipedia.org/wiki/Software_developer) may use a servlet to add [dynamic content](https://en.wikipedia.org/wiki/Dynamic_web_page) to a [web server](https://en.wikipedia.org/wiki/Web_server) using the [Java platform](https://en.wikipedia.org/wiki/Java_platform). The generated content is commonly [HTML](https://en.wikipedia.org/wiki/HTML), but may be other data such as [XML](https://en.wikipedia.org/wiki/XML) and more commonly, JSON. Servlets can maintain [state](https://en.wikipedia.org/wiki/State_(computer_science)) in [session](https://en.wikipedia.org/wiki/Session_(computer_science)) variables across many server transactions by using [HTTP cookies](https://en.wikipedia.org/wiki/HTTP_cookie), or [URL mapping](https://en.wikipedia.org/wiki/URL_mapping).

**JDBC:**

**Java Database Connectivity** (**JDBC**) is an [application programming interface](https://en.wikipedia.org/wiki/Application_programming_interface) (API) for the programming language [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), which defines how a client may access a [database](https://en.wikipedia.org/wiki/Database). It is a Java-based data access technology used for Java database connectivity. It is part of the [Java Standard Edition](https://en.wikipedia.org/wiki/Java_Standard_Edition) platform, from [Oracle Corporation](https://en.wikipedia.org/wiki/Oracle_Corporation). It provides methods to query and update data in a database, and is oriented towards [relational databases](https://en.wikipedia.org/wiki/Relational_database). A JDBC-to-[ODBC](https://en.wikipedia.org/wiki/ODBC) bridge enables connections to any ODBC-accessible data source in the [Java virtual machine](https://en.wikipedia.org/wiki/Java_virtual_machine) (JVM) host environment.

**Backend:** *{can run on any database}*

**Oracle/SQL Server**

**MySQL:**

* **MySQL**  is an [open source](https://en.wikipedia.org/wiki/Open-source_software) [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS). "[SQL](https://en.wikipedia.org/wiki/SQL)", is abbreviation for [Structured Query Language](https://en.wikipedia.org/wiki/Structured_Query_Language).
* MySQL is [free and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software) under the terms of the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License), and is also available under a variety of [proprietary](https://en.wikipedia.org/wiki/Proprietary_software) licenses. MySQL was owned and sponsored by the [Swedish](https://en.wikipedia.org/wiki/Sweden) company [MySQL AB](https://en.wikipedia.org/wiki/MySQL_AB), which was bought by Sun Microsystems (now [Oracle Corporation](https://en.wikipedia.org/wiki/Oracle_Corporation)).

**CHAPTER 3**

**ANALYSIS**

**Analysis**

* 1. **Software Requirements**
* Operating System: Linux OS, Windows 7/8/10
* IDE: Eclipse IDE for Java EE Developers (Oxygen)
* Server: MySQL Workbench Server 6.2, Tomcat 8.5
* RDBMS: MySQL
* Environment: JDK 1.6, 1.7, 1.8 for Java 6, 7, 8 configured on the workstation

**3.2 Hardware Requirements**

* Processor: 1.7GHz Intel Core2Duo or above
* RAM: 4 GB
* Hard Disk: 100 GB-1 TB
* Network Adaptor

**CHAPTER 4**

**DESIGN**

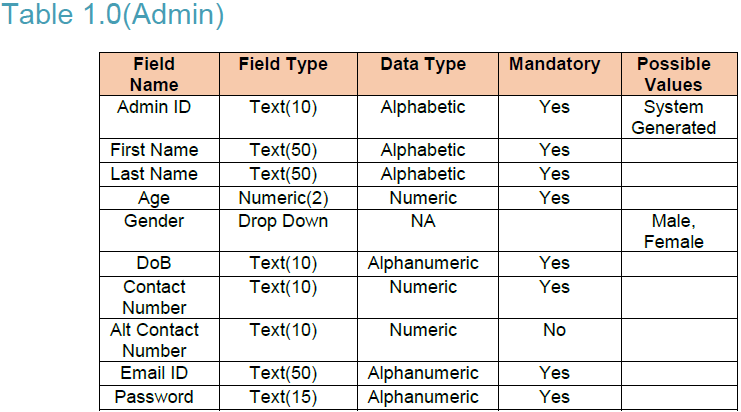
**Design**

* 1. **Diagram**

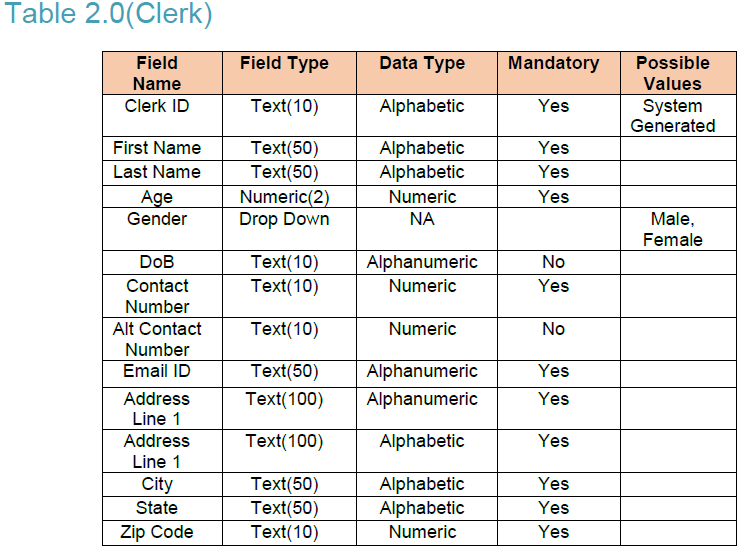


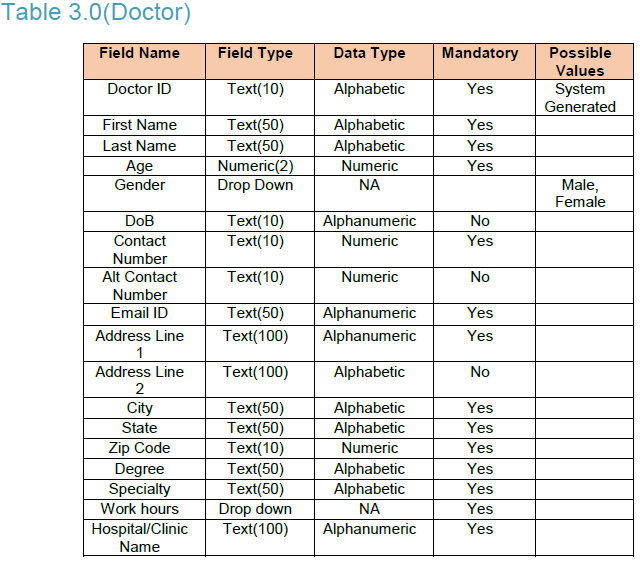
* 1. **Tables**

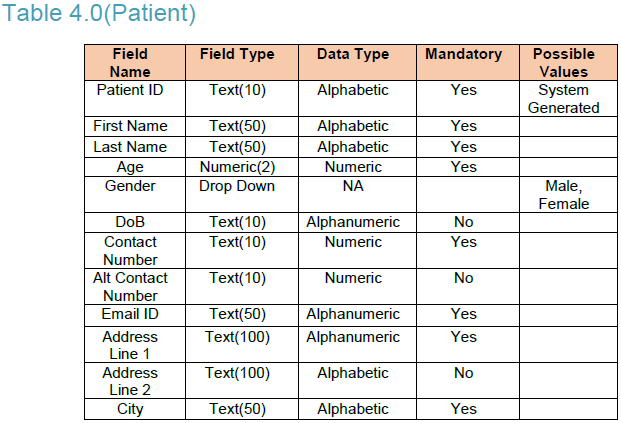
(Admin Table)

****

(Clerk Table)

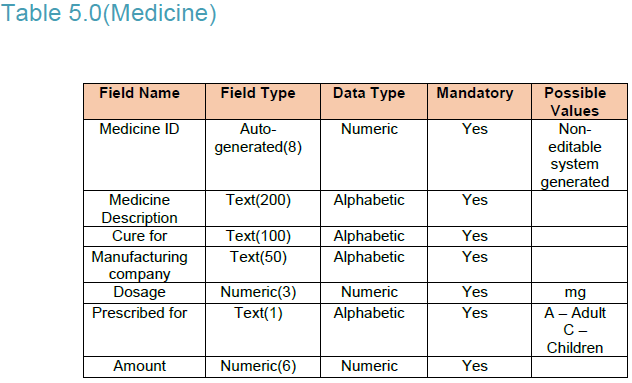


(Doctor Table)

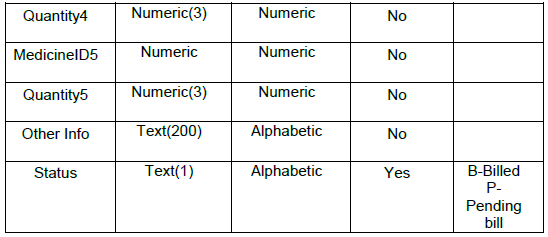
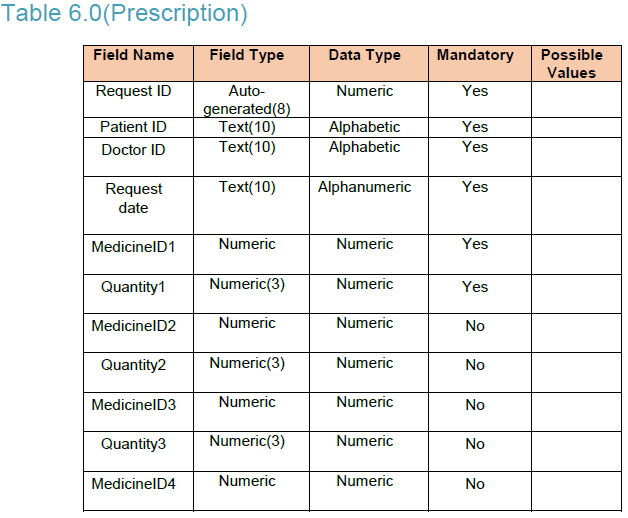
(Patient Table)



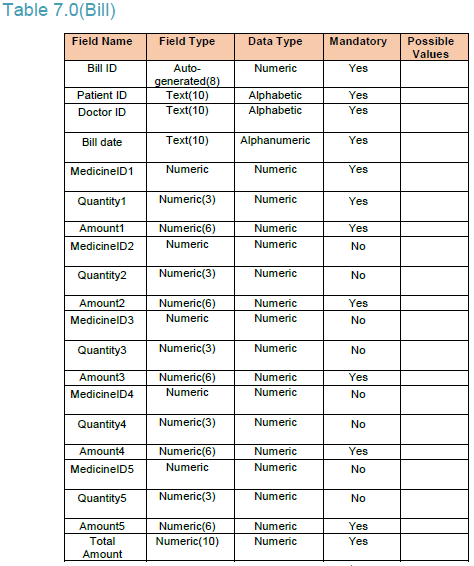
(Medicine table)



(Prescription Table)

****

(Bill Table)



**CHAPTER 5**

**CONCLUSION**

**Conclusion**

The patient Tracker management system guided us through the proper architecture of a management system, making us learn the structured levels of development. Being new to the software, a little problem was faced while writing the complete code from scratch however learning the new technologies in order to build the project has leveled up the making process of management system. As a limitation of project, there could be a few more functionalities added to enhance the system and similarly a better architecture can be used to make it smoother. But whatever is made, it has surely cleared our basics and taught us a lot. Concluding to it, we look forward to enhance the features of the system as well as expanding the possible opportunities that come its way.

**REFERENCES**

* w3schools.com
* docs.oracle.com
* stackoverflow.com
* css-tricks.com
* [www.techonthenet.com](http://www.techonthenet.com)
* [www.tutorialspoint.com](http://www.tutorialspoint.com)
* beginnersbook.com