

Dr.Catalyst EHR (WebIMS)

A PROJECT REPORT

Submitted by

KAUSHAL J JETHAVA

200170107507

In fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER ENGINEERING

VISHWAKARMA GOVERNMENT ENGINEERING COLLEGE

CHANDKHEDA



Gujarat Technological University, Ahmedabad

[April,2023]



VISHWAKARMA GOVERNMENT ENGINEERING COLLEGE
Nr. Visat Three Roads, Sabarmati-Koba Highway, Nigam Nagar, Chandkheda,
Ahmedabad, Gujarat 382424

CERTIFICATE

This is to certify that the internship report submitted along with the project entitled **Dr.Catalyst EHR (WebIMS)** has been carried out by **Kaushal Jethava** under my guidance in partial fulfillment of the degree of Bachelor of Engineering in Computer Engineering, 8th Semester of Gujarat Technological University, Ahmedabad during the academic year 2022-2023.

Internal Guide
Prof. Karan P Bhatt
Assistant Professor

Prof. Mansukh T Savaliya
Head of Department
Associate Professor
Computer
Engineering Department



MEDITAB®

Improving the Business and Delivery of Healthcare

Date: 21 April 2023


To Whomsoever It May Concern

This is to certify that Kaushal Jethava is an intern in our organization since January 23, 2023 until present. He has worked on the "WebIMS" project during his internship period in the organization.

During this period we found him punctual, hardworking and inquisitive.

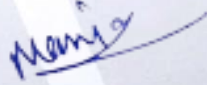
This letter is issued on employee's request and the company is not responsible for any current or future liabilities.

For Meditab Software (India) Pvt. Ltd.


Chaitra Shetty

Human Resources





Meditab Software (I) Pvt. Ltd. • 219/A, 2nd Floor, Kalasagar Shopping Hub,
Opp. Sattadhar Saibaba Temple, Ghatlodia, Ahmedabad - 380061.
Ph.: (079) 40083811 • hrindia@meditab.com • www.meditab.com



VISHWAKARMA GOVERNMENT ENGINEERING COLLEGE
Nr. Visat Three Roads, Sabarmati-Koba Highway, Nigam Nagar, Chandkheda,
Ahmedabad, Gujarat 382424

DECLARATION

We hereby declare that the Internship report submitted along with the Project entitled **Dr.Catalyst EHR (WebIMS)** in partial fulfillment for the degree of Bachelor of Engineering in Computer Engineering to Gujarat Technological University, Ahmedabad, is a bonafide record of original project work carried out by me at Vishwakarma Government Engineering College under the supervision of Prof. Karan P Bhatt and that no part of this report has been directly copied from any students' reports or taken from any other source, without providing due reference.

Name of the Student

Sign of Student

Kaushal J Jethava

Acknowledgement

We, the developers of **Dr.Catalyst EHR (WebIMS)**, with immense pleasure and commitment would like to present the project assignment. The development of this project has given me so many opportunities to think, explore and get to know about the Web Development domain.

Every work that one completes successfully stands on the constant encouragement, good will and support of the people around. We hereby avail this opportunity to express our gratitude to number of people who extended their valuable time, full support and cooperation in developing the project.

We express deep sense of gratitude towards our Head of the CE Department, **Prof. Mansukh T Savaliya** and project guides **Prof. Karan P. Bhatt** (internal guide) and **Mr. Manish Lakhara** (external guide) for the support during the whole session of study and development. It is because of them, that we were prompted to do hard work, adopting new technologies.

Furthermore, we are grateful to all the people who kept supporting me and provided a positive environment to work with. Last but not least, we would like to thank my family members who provide me enormous support during this work directly and indirectly. We would like to thank my friends and colleagues who have been a source of inspiration and motivation that helped me during internship period.

Abstract

Creating a patient demographics module similar to the web ims application to input patient details, working on the front end using HTML, CSS and JS and apis using c# and dot net core . api integration using fetch api in browsers using postgresql as database.

List of Figures

Fig 2.1 JS Event Loop	6
Fig 2.2 Angular Overview	8
Fig 3.1 Agile development model	10
Fig 3.2 Cost and Time Estimation	11
Fig 3.3 Gantt Chart of Backend Development	12
Fig 3.4 Gantt Chart of Frontend Development	13
Fig 5.1 Activity Diagram for patients	21
Fig 5.2 Activity Diagram for admin	22
Fig 5.3 Use Case Diagram	23
Fig 5.4 Class Diagram	24
Fig 5.5 Sequence Diagram	24
Fig 5.6 ER Diagram	25
Fig 6.1 Home page of the app	28
Fig 6.2 Patient Demographics Page	28
Fig 6.3 Patient Contact Section	29
Fig 6.4 Patient Contact Preference Details Section	29
Fig 6.5 Patient Other Contact Details Section	30
Fig 6.6 Facility Page	30
Fig 6.7 Dashboard Page	31
Fig 6.8 Setup Page	31
Fig 6.9 Checked In/ Checked Out Page	32
Fig 6.10 Implementation Page	32

List of Tables

Table 8.1 Tests Performed on the App	37
Table 8.2 Test Suite: 1	39
Table 8.3 Test Suite: 2	40
Table 8.4 Test Suite: 3	42
Table 8.5 Test Suite: 4	43
Table 8.6 Test Suite: 5	44

Abbreviations

EHR	Electronic Health Record
EMO	Electronic Medical Office
JS	JavaScript
TS	TypeScript
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheet
API	Application Programming Interface
SPA	Single Page Application
RCA	Root Cause Analysis
AJAX	Asynchronous JavaScript and XML
HIPAA	Health Insurance Portability and Accountability Act

Table of Contents

Acknowledgement.....	i
Abstract.....	ii
List of Figures.....	iii
List of Tables.....	iv
Abbreviations.....	v
Contents	
Chapter 1: OverviewoftheCompany	1
1.1 History.....	1
1.2 Work done.....	1
1.3 Solutions Provided	1
1.4 Departments at work	2
1.5 Goalsand Achievements	2
1.6Size of the Company	2
Chapter 2: Introductionto Internship	4
2.1 Internship Overview	4
2.2 Purpose	4
2.3 Objective	4
2.4 Technology Overview	5
2.4.1 JavaScript.....	5
2.4.2 TypeScript.....	6
2.4.3 Angular Framework	6
2.5 Internship Flow	8
2.6Scheduling.....	9
Chapter 3: Project Management	10
3.1 Project Planning	10
3.1.1 Project Development Approach And Justification	10

3.1.2 Project Effort And Time, Cost Estimation	11
3.1.3 Roles And Responsibility:	11
3.1.4 Group Dependency	12
3.2 Project Scheduling	12
Chapter 4: System Requirements Study	14
4.1 User Characteristics	14
4.2 Hardware And Software Requirements	15
4.2.1 Development Environment.....	15
4.2.2 Execution Environment.....	16
4.3 Assumptions And Dependencies	17
Chapter 5: System Analysis	18
5.1 Study Of Current System	18
5.2 Problems Of Current System	18
5.3 Functional And Non-Functional Requirements	19
5.3.1 Functional Requirements.....	19
5.3.2 Non-Functional Requirements.....	19
5.4 Feasibility Study	20
5.4.1 Does The System Contribute To The Overall Objectives Of The Organization?	20
5.4.2 Can The System Be Implemented Using The Current Technology And Within The Given Cost And Schedule Constraints?	20
5.4.3 Can The System Be Integrated With Other Systems Which Are Already In Place?	20
5.5 Activity/ Process In New System.....	21
5.6 Features Of New System.....	23
5.7 Usecase Diagram.....	23
5.8 Class Diagram	24

5.9 Sequence Diagram	24
5.10 ER Diagram (System Generated).....	25
5.11 List Main Modules Of New System	25
5.12 Selection Of Hardware & Software And Justification.....	26
Chapter 6: System Design	27
6.1 System Application Design.....	27
6.1.1 Method Pseudo Code.....	27
6.2 Input/Output And System Design	28
6.2.1 Home Page.....	28
6.2.2 Patient Demographics Page	28
6.2.3 Patient Demographics Contact Details Section	29
6.2.4 Patient Demographics Contact Preference Details	29
6.2.5 Patient Other Details.....	30
6.2.6 Facility Page	30
6.2.7 Dashboard Page	31
6.2.8 Setup Page	31
6.2.9 Checked In/ Checked Out Page	32
6.2.10Implementation Page	32
Chapter 7: Implementation Planning	33
7.1 Implementation Environment.....	33
7.2 Program/Module Specification	33
7.3 Security Features	34
7.4 Coding Standards	34
7.4.1 Naming Convention And Standards	34
7.4.2 Naming Convention.....	35
Chapter 8: Testing	37
8.1 Testing Plan.....	37

8.2 Test Strategy.....	38
8.3 Test Suits Design	39
8.3.1 Test Cases.....	39
Chapter 9: Conclusion	45
9.1 Self Analysis Of Project Viabilities	45
9.2 Problem Encountered And Possible Solution	45
9.2.1 Problem Encountered	45
9.2.2 Possible Solution	45
9.3 Summary Of Project Work.....	45
Chapter 10: Limitations And Future Work	46
10.1 Limitations	46
10.2 Future Enhancement	46
References	47
Appendix 1 Plagiarism report	
Appendix 2 Weekly log book	
Appendix 3 Attendance Sheet	
Appendix 4 Feedback From	

1. Overview of the Company

1.1 History

The Patel family founded Meditab in 1998 after experiencing first-hand the frustrations of having an inadequate EHR. Fed up with the tedious software that they were using in their family-owned pharmacy, they were determined to create a system and a company that works with providers, instead of getting in the way.

Soon, IMS was born. By 2003 launched IPS, as well, expanding of tware to pharmacies.

Since then, thanks to the simple desire to make providers' lives easier, Meditab has grown to offer software for more than 40 different medical specialties and has become a pioneer in clinic software.[5]

1.2 Work Done

During the past two decades, MEDITAB have built innovative, specialty-specific software such as AllergyEHR, FertilityEHR, Ophthalmology EHR, CosmetiSuite, and more. They developed solutions that address all aspects of healthcare, from mobile functionality and telemedicine, to an all-new suite of office management tools integrated with your EHR, the electronic Medical Office(EMO).

Healthcare is always changing. That's why customization and adaptability are at the core of IMS. And for more than 20 years, Meditab has always been at the forefront of change and innovation.

1.3 Solutions Provided HER

Our all-in-one multi-specialty EHR, Practice Management and Billing software solution allows doctors to deliver the highest quality of care possible to their patients. Streamline your practice's workflows, optimize financial performance, and coordinate quality care faster with IMS.

Our all-in-one multi-specialty EHR, Practice Management and Billing software solution allows doctors to deliver the highest quality of care possible to their patients. Streamline your practice's workflows, optimize financial performance, and coordinate quality care faster with IMS.

EMO

EMO stands for Electronic Medical Office. It's an entire suite of office management tools integrated with IMS and designed to optimize the medical office. Composed of several practical, time-saving modules, EMO transforms your traditional and manual processes into a digital workflow.

1.4 Departments at work

The different departments who work together to achieve a common goal:

- Human and Resources
- Accounts
- Admin
- Sales and Marketing
- IMS Developer and Quality Analyst
- WebIMS Developer and Quality Analyst
- Cyber Security
- IT Services
- AndroidIMS and Quality Analyst
- Internal PHP

1.5 Goals and Achievements

- ISO9001:2015certified company
- BlackBook 2022 Named Allergy HER the #1 HER for Allergy & Immunology
- Drummond Group 2015Ed Complete HER Certification.
- Intelligent Medical Software Named Gold Health IT Quality Solution by Quest Diagnostics.

1.6 Size of the Company

Meditab Software is a growing company, with around 800 employees and more. Company has three locations situated at

- 219/A, Kalasagar Mall, Sattadhar Cross Road Ghatlodiya, near Sai Baba Temple,

Ahmedabad,Gujarat380061

- 8795Folsum,Sacramento,CA
- Skyrise4,Sebucity,Phillipine

2. Introduction to Internship

2.1 Internship Overview

This report is a detailed overview of my internship journey at Meditab Software Inc. During my Internship, I have learned a lot about corporate environment, communication skills and different languages like HTML, CSS, JS, TS and Angular framework.

I have learned to work in a corporate space which not only enriched me professionally but also helped me grow personally as well. As I have had a great opportunity to practically see how software development is done in real life. Moreover, I have tried my level best to make this report meaningful by reflecting my work at the Meditab Software Inc. Also, I have summarized my overall experience, with my learning and challenges faced as an intern.

2.2 Purpose

Being a knowledge seeker, I wanted to gain some experience on how our learning from college can be implemented in a real live problem. As the time is passing, Web Development's popularity is increasing day by day. They are the best alternative to any Operating system based software as it needs installing many dependencies for the end user. This internship has given me a platform to achieve it and helped me in applying technical knowledge in a live project.

2.3 Objective

The main objective of this internship was to develop myself in the area of Front End Web Development. Also I wanted to know what are the different ways and technologies that are applied and used in real life scenarios. Also how a corporate runs and how many departments combine together to achieve a common goal. Another reason was to develop myself in the area of communication.

Overall, this was a great experience to be in and has helped me develop many of my professional and personal set of skills.

2.4 Technology Overview

2.4.1 JavaScript

JavaScript is a scripting language used mostly to make websites interactive. While HTML and CSS are used to provide the blueprint and structure, JS provides dynamic behavior to the site. It is a client side scripting language and not generally used to script servers.

There is a different framework of JS called NodeJS which can be used to script server side machines.

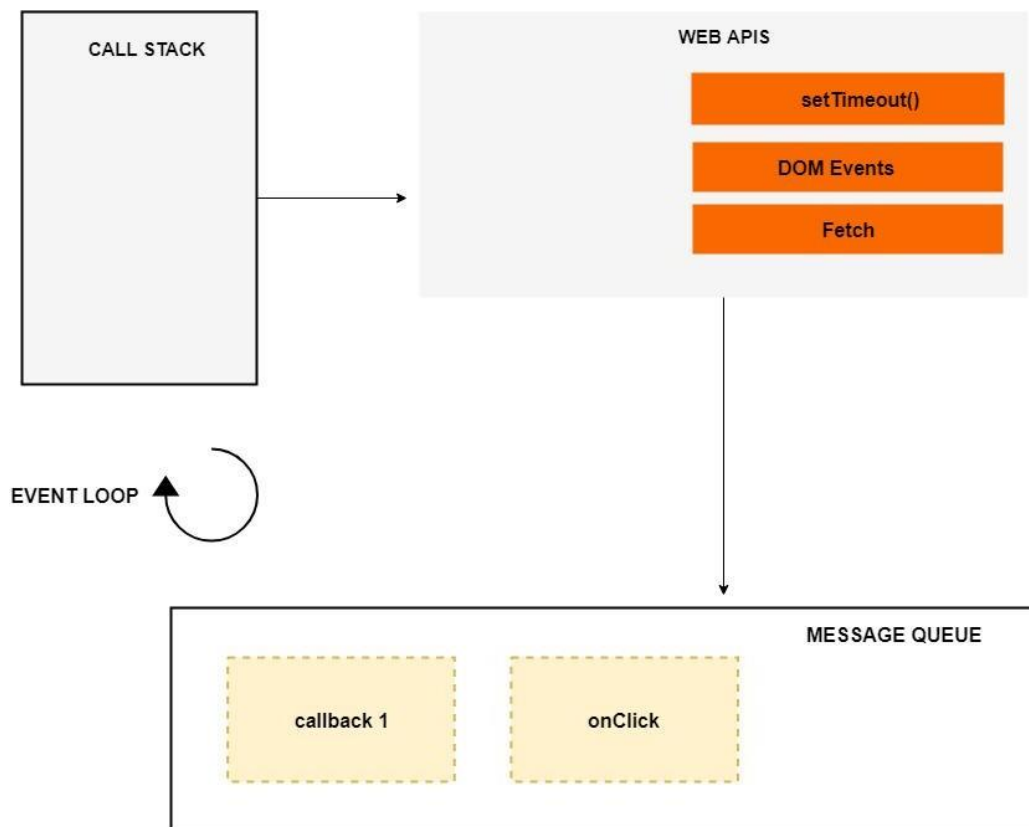
The JS mainly is divided into two parts - synchronous and asynchronous JS. The synchronous JS deals with regular stuff where more than one operations are not to be performed at a time. To say in real world example, it does not deal with fetching data from servers. It can lead to serious delay if we wait for data to come from server and then do other operations.

That is why asynchronous JS came into picture. The basic idea behind async JS is that there are WebAPI offered by our browser, so despite of JS being a single threaded language, when such requests of data are to be handled, we tell those API offered by browser to handle such things and JS compiler can deal with other things while waiting for the data. This is a huge advancement as no one will want to wait for such a time as it is not sure when or even will the data will be delivered.

Look at fig 2.1 for JS event loop and how it is usually handled by API provided by browser.

Browser has two things basically - Call stack and API. Event loop is kind of like a timer or interval where at the end of each loop, async tasks and event tasks are checked.

API provided by the browser keeps track of all the sync process. When a async code or event is encountered, it is sent into API. By doing this the js lint is not always busy. So when the API is done with finishing the async task, it waits for the next event loop and then renders it into the DOM. Call Stack is just a stack which keeps track of all the current executing functions or procedures.^[1]

Fig-2.1JSEvent Loop ^[1]

2.4.2 TypeScript

TypeScript is just like a big brother to JS. It is a superscript of JS. It adds an additional syntax to JS which then helps in tighter integration with the editor. It also helps you in catching errors early. TS is always converted into JS before browser. Browser only deals with JS file, so TS lint will convert code into JS before browser can handle it. TS also adds type referencing to JS which is a huge advantage.

TS is used in Angular framework, as it is more reliable. Usually Angular apps are too large, so using TS makes it easier to debug and keeping track of all the types.

It is developed and maintained by Microsoft, which is a strict syntactical superscript of JS. It is only designed for development of large applications.

2.1.1 Angular Framework

Angular is a FE, Single Page Application (SPA) Framework developed and maintained by Google. First Angular was based on JS, but then the developers rewrote the framework and made it dependent

on TS. Angular 2 and 2+ are strictly based on TS.

Angular includes: A component-based framework for building scalable web applications. A collection of well-integrated libraries that cover a wide variety of features, including routing, forms management, client-server communication, and more. A suite of development tools to help you develop, build, test, and update your code.^[2]

The advantages of Angular are:

- Use modern web platform capabilities to deliver app-like experiences. High performance, offline, and zero-step installation.
- Build native mobile apps with strategies from Cordova, Ionic, or NativeScript.
- Angular turns your templates into code that's highly optimized for today's JavaScript virtual machines, giving you all the benefits of hand-written code with the productivity of a framework.
- Angular apps load quickly with the new Component Router, which delivers automatic code-splitting so users only load code required to render the view they request.
- Command line tools: start building fast, add components and tests, then instantly deploy.
- Create high-performance, complex choreographies and animation timelines with very little code through Angular's intuitive API.

The different features provided by Angular are:

- Components, custom and inbuilt
- Data and Event Binding
- Directives and Pipes
- Services
- Dependency Injection
- Routing
- Observables
- Forms-Reactive and Template
- HttpRequests
- Dynamic Component
- Animations.

The Angular overview is shown in the figure 2.2

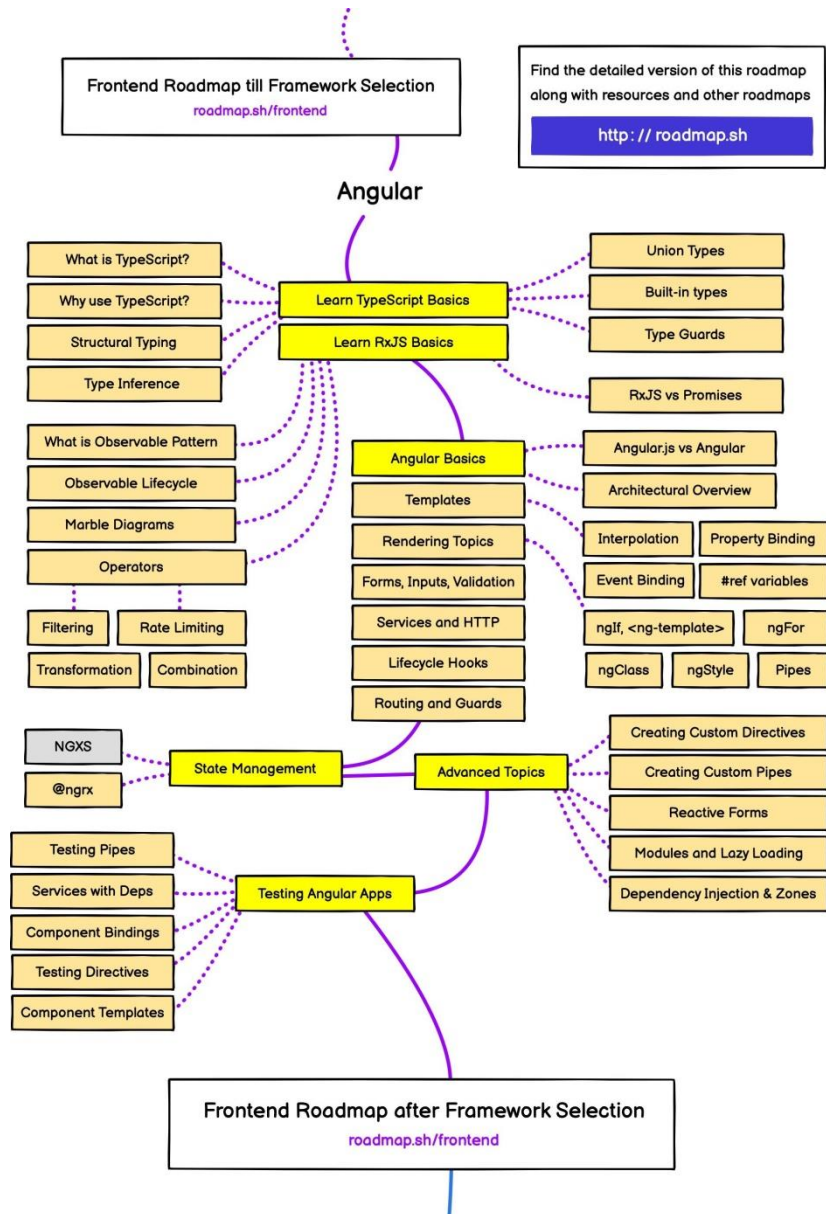


Fig-2.2 Angular Overview^[2]

2.5 Internship Flow

2.5.1 Responsibility and roles

- Reporting to Team Lead about daily work and daily goal.
- Do the tasks assigned for confirmation of profile.
- Understanding already written live project code.

- Finding Root Cause Analysis of the assigned bug.
- Fixing the bug according to the RCA found.

2.6 Scheduling

The internship was divided into three parts:

1. Completion of tasks assigned for confirmation of work profile
2. Complete the Training of the profile
3. Bug fixing and new feature implementation

3. Project Management

3.1 Project Planning

3.1.1 Project Development Approach and Justification

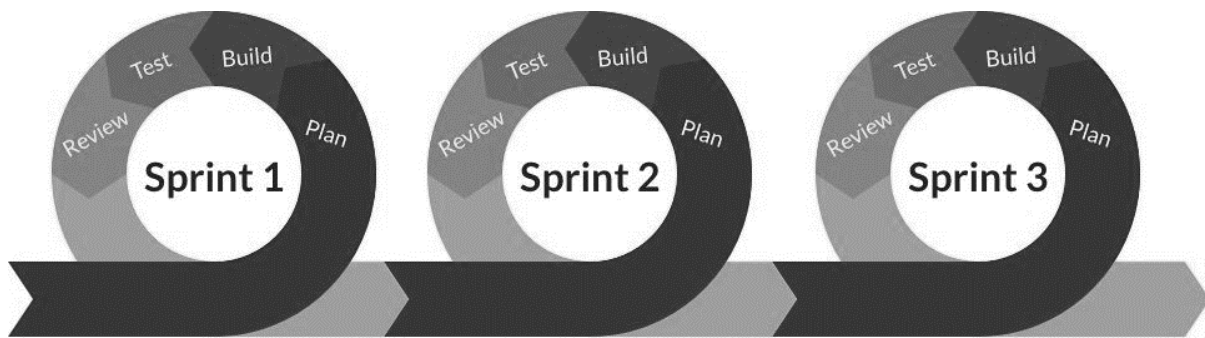


Fig 3.1 Agile development model[5]

Agile models believe that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In Agile, the tasks are divided to time boxes (small time frames) to deliver specific features for a release.

Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.

Following are the Agile Manifesto principles:

- **Individuals and interactions** –In Agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.
- **Working software** –Demo working software is considered the best means of communication with the customers to understand their requirements, instead of just depending on documentation.
- **Customer collaboration** –As the requirements cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirements.
- **Responding to change** –Agile Development is focused on quick responses to change and continuous development.

3.1.2 Project Effort and Time, Cost Estimation

- Regarding the programmer's efforts we spoke to the management of the institution's final affairs, the staff who kept the records in most registers and the accounting officer about their current program, their needs and their expectations for the proposed new system. system.
- Reliable, accurate and secure data was also regarded as hard work outside of the proposed system. Because there was such a record of tracking all activities, performed by medical billing and daily insurance plan.
- This web application started about 5 years ago, Software costs include a small percentage of all computer-based system costs. There are a few factors, considerable, that can affect the final cost of software such as - personal availability, technology, Hardware and Software etc.
- The cost estimates made by our company for the project are also based on the basic metrics collected from past projects and these are used in conjunction with variable rates to improve costs and efforts.

Software Project	a_b	b_b	c_b	d_b	Select
Organic	2.4	1.05	2.5	0.38	<input type="radio"/>
Semi-detached	3.0	1.12	2.5	0.35	<input checked="" type="radio"/>
Embedded	3.6	1.20	2.5	0.32	<input type="radio"/>

Calculate Effort and Duration

Effort (E) = $a_b(KLOC)^{b_b}$ = 12.57 Duration (D) = $c_b(E)^{d_b}$ = 6.06

Reset Data

Fig 3.2 Cost and Time Estimation

3.1.3 Roles and Responsibility

Backend developer: Responsibility of backend developer is to create and update APIs for creating, deleting and updating patient allergies and then test the APIs using postman tool.

Frontend developer: Responsibility in frontend development is to create UI for patient demographics.

3.1.4 Group Dependency

When a developer wants to deploy code on a server, they need to submit their code to a code reviewer, which is a dependency when they want to test the product on servers. Developers are also dependent on requirements which are changing when an agile model is followed. Tester is dependent on the developer.

3.2 Project Scheduling

Gantt Chart:

- It is also known as Bar chart is used exclusively for scheduling purpose. It is a way to manage a project. It is used for scheduling appointments and organizing events for the venue in particular. Budget planning and resource planning. Gantt is a bar chart with each bar representing the function. Bars are drawn against the timeline. The length of time scheduled for work. The Gantt chart in the diagram shows the grey parts of the loosening period which is the last time the work is completed.

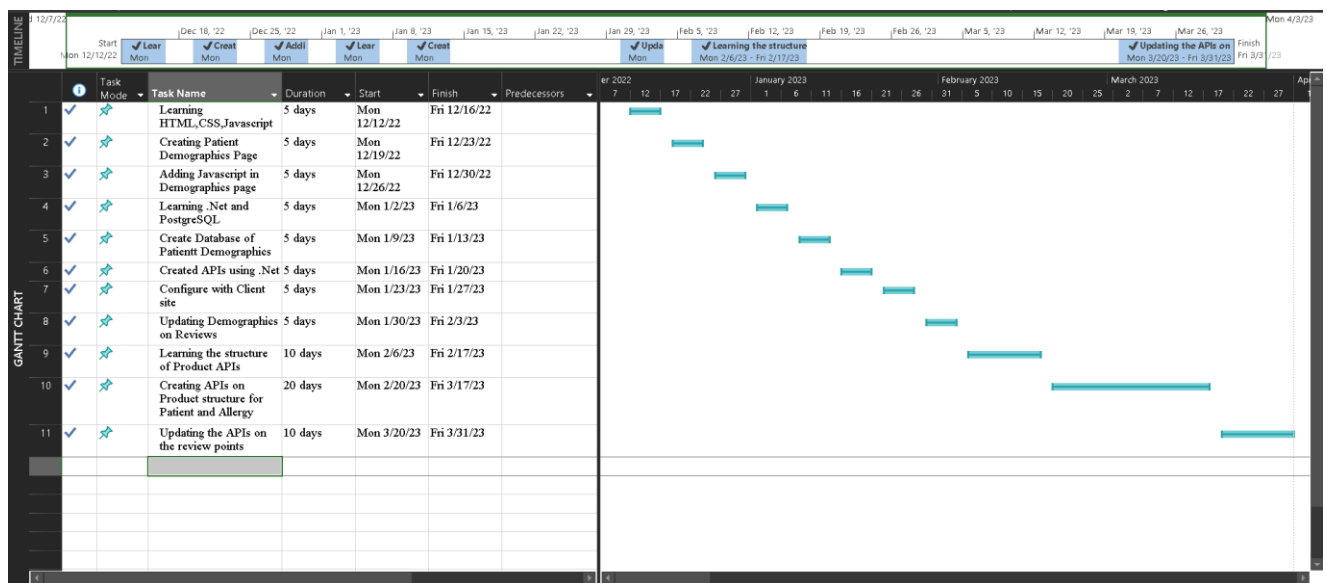


Fig 3.3 Gantt Chart of Backend Development

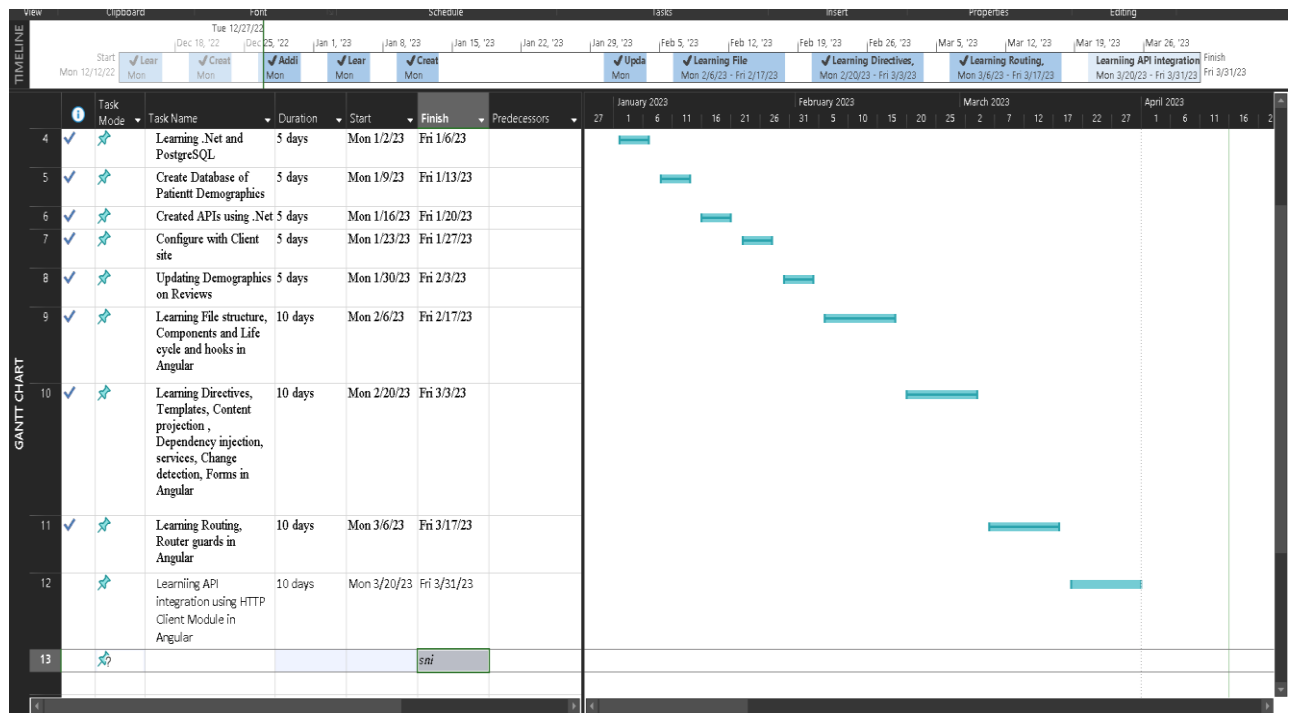


Fig 3.4 Gantt Chart of Frontend Development

4. System Requirements Study

4.1 User Characteristics

I have identified four potential classifications of users of our system.

- Software Developers
- Business Analysts
- Support Team
- Sales Team

1) Software Developers

✓ About

These are the people that take the model generated by the designers and implement it in code. They may also use the system to identify the design of an existing system to maintain it.

✓ Characteristics

- A thorough knowledge of object-oriented design notation.
- Familiarity with common design environments.
- Knowledge about all the screens and forms.
- Information about all processes.
- Can solve any problem of project

2) Business Analysts

✓ About

These are the people that take the model requirements from the clients, represent them to the Developers as a rough design and also implement them in code. They may also use the system to identify the design of an existing system to maintain it.

✓ Characteristics

- Assisting with the business case
- Planning and monitoring
- Eliciting requirements
- Requirements organization
- Translating and simplifying requirements

- Requirements management and communication
- Requirements analyst

3) Support Team

✓ About

These are the people that take the client's error or bug or problem and try to solve it by sending it forward to Developers or Business Analysts.

✓ Characteristics

- Assisting with the clients
- Listen to the clients' problems
- Bug reporting
- Assign someone to fix bug
- Developer assignment

4) Sales Team

✓ About

These are the staff which deals with sales of the product.

✓ Characteristics

- Goes to clients to display and sale the product
- Has knowledge about the product

4.2 Hardware And Software Requirements

4.2.1 Development Environment

Software Requirement:

- Angular 11
- .NET
- PostgreSQL
- Visual Studio

- VS Code (1.77.1)
- Postman
- Any compatible browser

Hardware Requirement:

- 8 GB RAM
- 256 GB SSD
- Processor: Intel i5 10th generation
- Mouse: Any Compatible
- Keyboard: Any Compatible
- Display: Any Compatible

4.2.2 Execution Environment

Software Requirement:

- Any Browser

Hardware Requirement:

- RAM – 2GB
- Device with decent internet connectivity
- Mouse: Any Compatible
- Display: Any Compatible

4.3 Assumptions And Dependencies

✓ Assumptions

- This IMS system is Secure.
- Assumptions are made that the system is not responsible for the security if one of the users shares their credentials with somebody else.
- This can lead to breach in the security of the software.

✓ Dependencies

- Software is dependent on the network.
- If the network goes down the working of the system will get affected.
- Dependent on the secure connection. The main disadvantage of having IMS is that IMS systems are vulnerable to hacking, which means sensitive patient data could fall into the wrong hands.

5. System Analysis

5.1 Study of Current System

Traditional way of gathering and storing patient data is very tedious. US health care work flow is completely different from India's health care process. Patient can not directly get medicines and cannot do meeting directly with health care provider. For the check-up or follow up process patient needs to book an appointment and then at the time of appointment at front-desk the patient needs to fill-up various forms, then the get meeting with MA (Medical Assistant) they get the patient vitals and after checking up and allergy check they get meeting with health care provider (Doctor) they give the prescription based on patient reports and then patient gets the medicines.

In this whole process the medical organization have to keep the patient securely according to HIPAA (Health Insurance Portability and Accountability Act) so to store this much amount of data securely and maintain these data is not easy task.

To provide solution for above problem US Government introduce EHR (Electronic Health Record) which store data in electronic form instead manually on paper. And make things easy Meditab Software have come up with IMS software which is Intelligent Medical Software. Which provides healthcare provides to create and manage patient records, medical histories and clinical notes, appointment scheduling, billing management, manage claims, allows healthcare providers to create and manage medical workflow etc.

5.2 Problems of Current System

- **Complexity:** WebIMS is a comprehensive EHR system with many features and capabilities. As a result, it may be complex and require a significant amount of training to use effectively.
- **Customization:** While WebIMS is highly customizable, some users may find that certain features or workflows are not easily customizable to meet their specific needs.
- **Integration:** While WebIMS can integrate with other systems and third-party applications, the integration process may be complex and require additional resources.
- **Cost:** As a comprehensive EHR system, WebIMS may come with a significant cost, including licensing, training, and maintenance fees.
- **Reliability:** As with any software system, there is a risk of downtime or other technical issues that could impact the reliability of WebIMS.

- **Dependence on internet connectivity:** The project relies on a stable internet connection to access and transmit data between Meditab's software and Dr.Catalyst EHR's virtual scribe service. Any disruption in the internet connection may affect the functionality of the system.

5.3 Functional And Non-Functional Requirements

5.3.1 Functional Requirements

- **Patient management:** WebIMS should allow healthcare providers to create, manage, and access patient records, including demographics, medical history, clinical notes, and appointment history.
- **Appointment scheduling:** WebIMS should allow healthcare providers to schedule, manage, and access appointments, including automated appointment reminders.
- **Billing and claims management:** WebIMS should allow healthcare providers to create, manage, and access claims, process payments, and generate reports related to billing and claims management.
- **Clinical workflows:** WebIMS should allow healthcare providers to create, manage, and access clinical workflows, such as order sets and protocols, to improve clinical processes and patient outcomes.
- **Prescription management:** WebIMS should allow healthcare providers to create, manage, and access prescriptions, including electronic prescribing and refill requests.
- **Reporting and analytics:** WebIMS should allow healthcare providers to generate reports and analyse data related to patient care and practice performance.

5.3.2 Non-Functional Requirements

- **Performance:** This includes requirements related to the speed, scalability, and responsiveness of the system.
- **Usability:** This includes requirements related to the ease of use and understandability of the system for the end-users.
- **Availability:** This includes that the features and functionalities are always available to users at any point of time.
- **Maintainability:** This includes requirements related to the ease of maintaining the system, including testing, debugging, and modifying the system.

- **Portability:** This includes requirements related to the ability of the system to be easily transferred to different hardware or software environments.
- **Manageability:** This feature is defined as the ability to control a system efficiently and keep it fully operational.
- **Data integrity:** Data integrity refers to maintaining and assuring data accuracy and consistency over its entire lifecycle.

5.4 Feasibility Study

Feasibility is the study of impact, which happens in the organization by the development of a system. The impact can be either positive or negative. When the positives nominate the negatives, then the system is considered feasible.

5.4.1 Does the system contribute to the overall objectives of the organization?

Yes, it does. Intelligent Medical Software is an important part of the most valuable product of organization Meditab. It will make the process of storing and retrieving the data easier and efficient.

5.4.2 Can the system be implemented using the current technology and within the given cost and schedule constraints?

Development of this application is highly economically feasible. The organization does not need to spend more money for the development of the system. The only thing to be done is making an environment for the development with an effective supervision. If we are doing so, we can attain the maximum usability of the corresponding resources. Even after the development, the organization will not be in a condition to invest more in the organization. Therefore, the system is economically feasible.

5.4.3 Can the system be integrated with other systems which are already in place?

No, the Dr.Catalyst EHR (WebIMS) is web based application which cannot be integrated with other web applications. User can use this with any browser available and on any device so there is no any requirement to integrate the system with other system.

5.5 Activity/ Process In New System

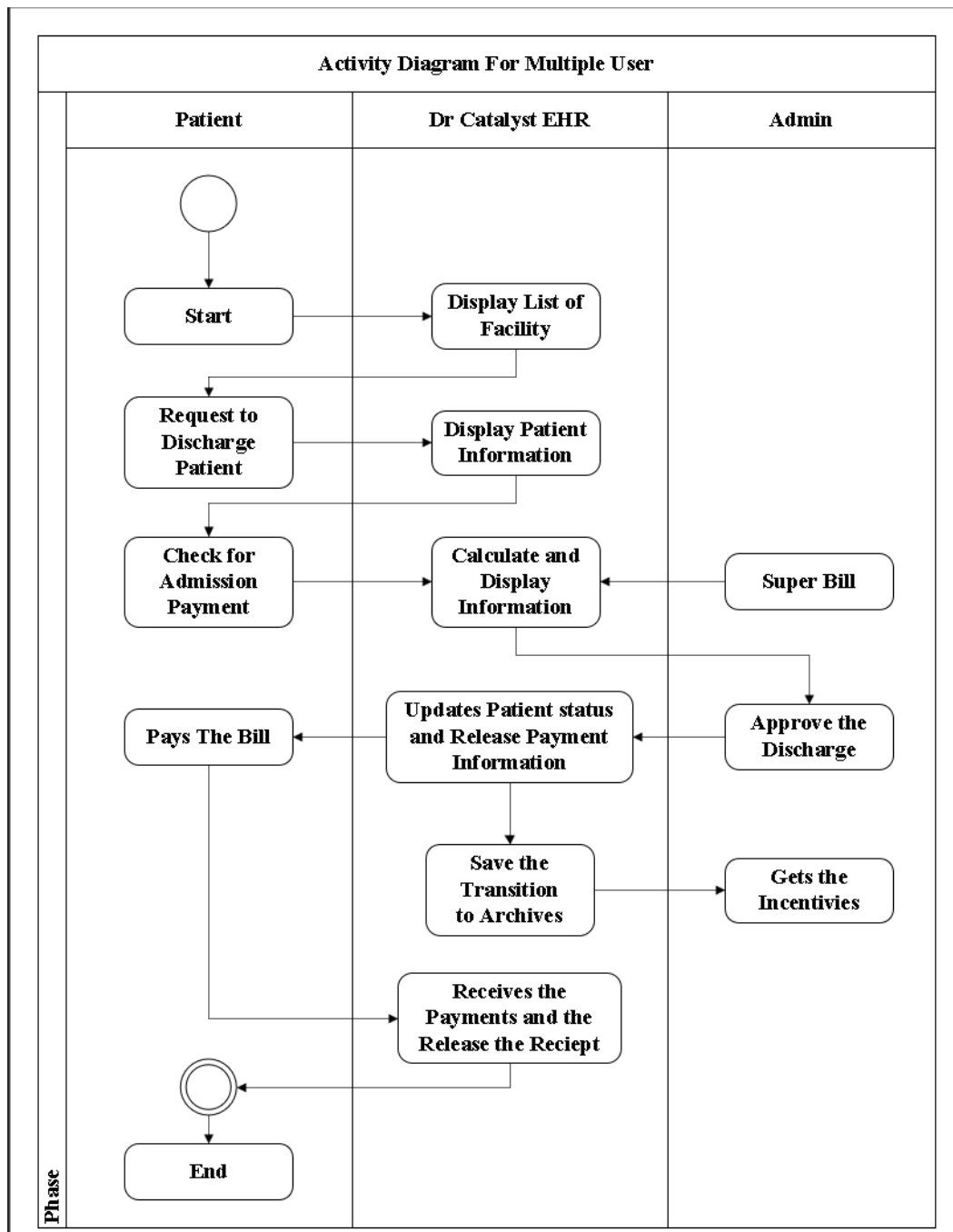


Fig 5.1 Activity Diagram for patients

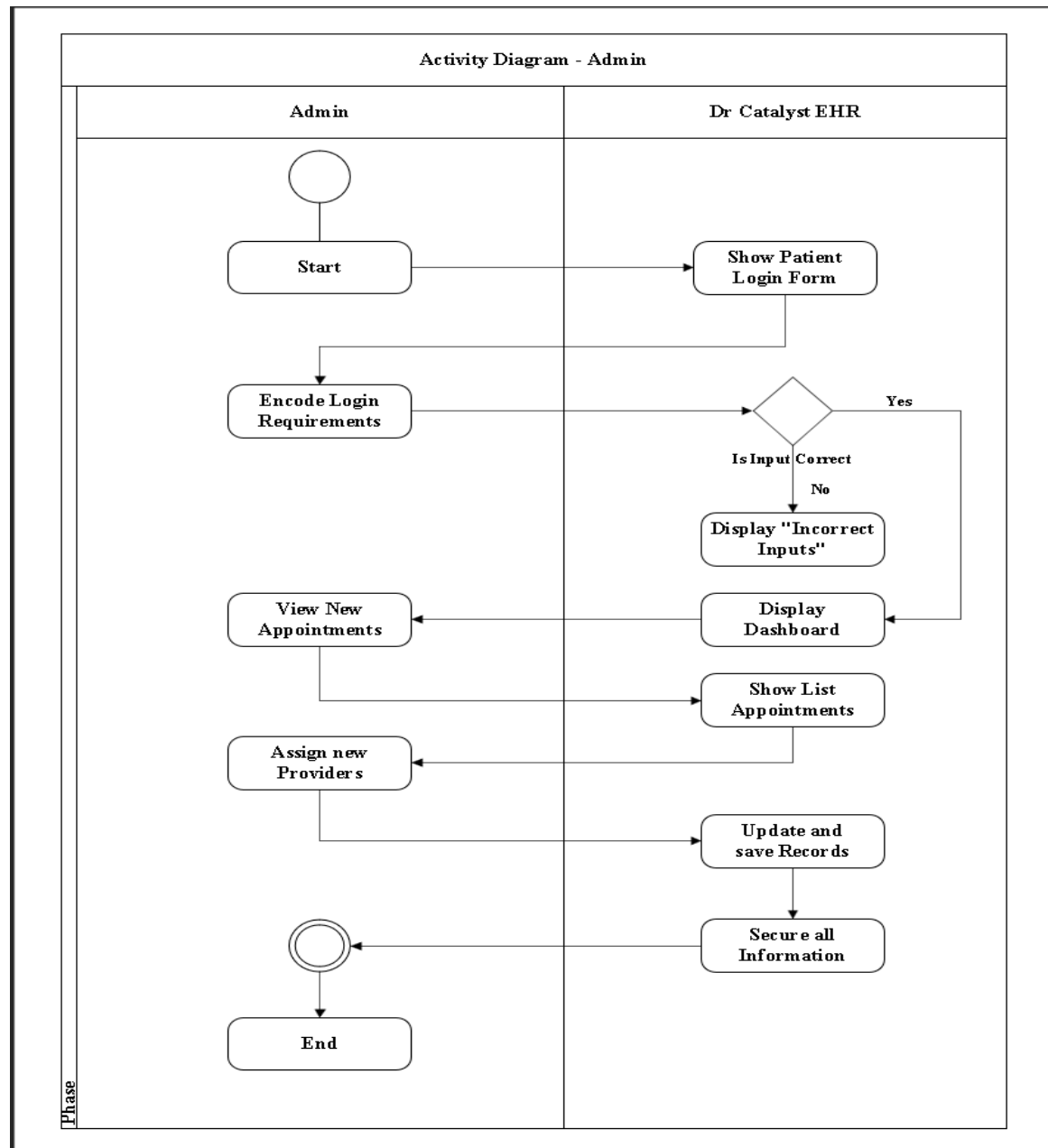


Fig 5.2 Activity Diagram for admin

In the new system, users will enter credentials. It can be done through a simple login portal. After login is done successfully, the main screen will be opened. Users will get options to click on toolbar items or navigate through the menu bar. The client, which needs to open a patient master for entering/editing/deleting the patient details. The client, which needs to enter the insurance details for the selected patient, will open an insurance patient portal from the patient master. As soon as the

user selects the instruction master, the user will be able to enter data regarding the instructions that pop up when the patient and provider performs some action. On clicking on authorization tracking, users can track all the claims for the insurance.

5.6 Features of New System

- Identify and maintain a patient record.
- Manage patient demographics.
- Manage problem lists.
- Manage Patient History.
- Manage claim tracking.
- Instruction master for mentioning the instruction that needs to be followed for some particular processes.

5.7 Use Case Diagram

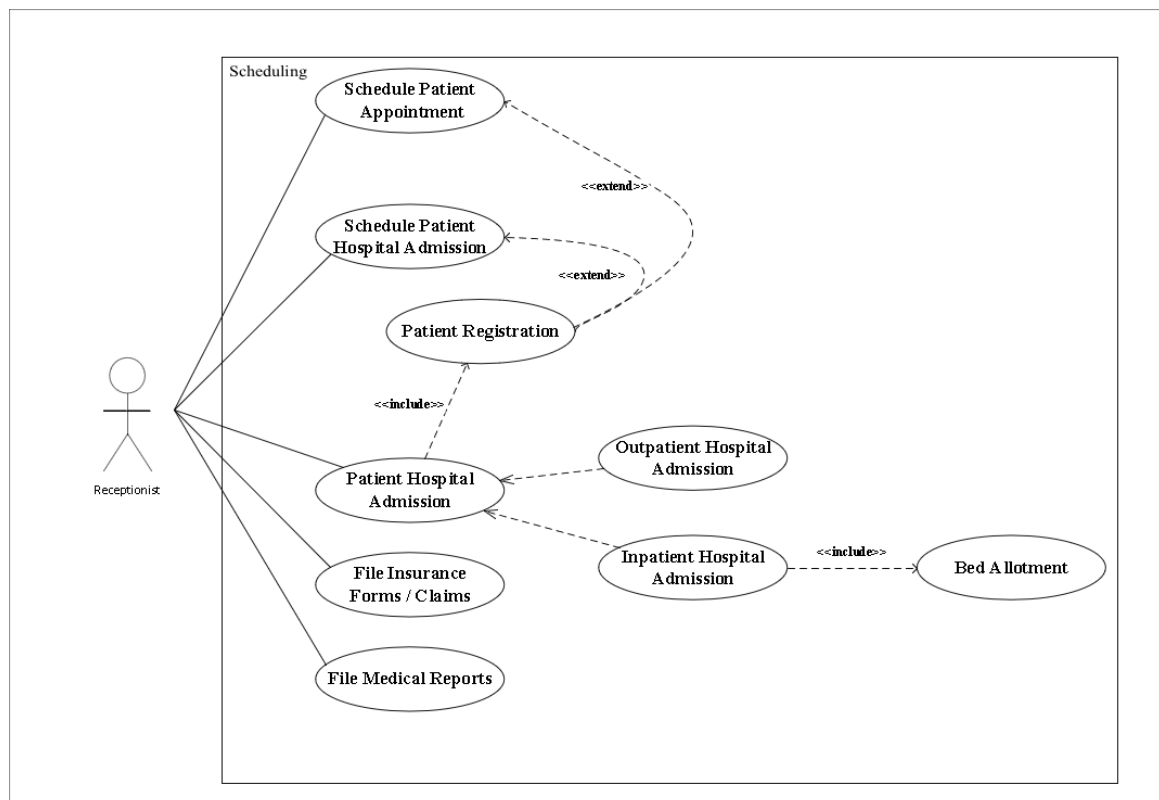


Fig 5.3 Use Case Diagram

5.8 Class Diagram

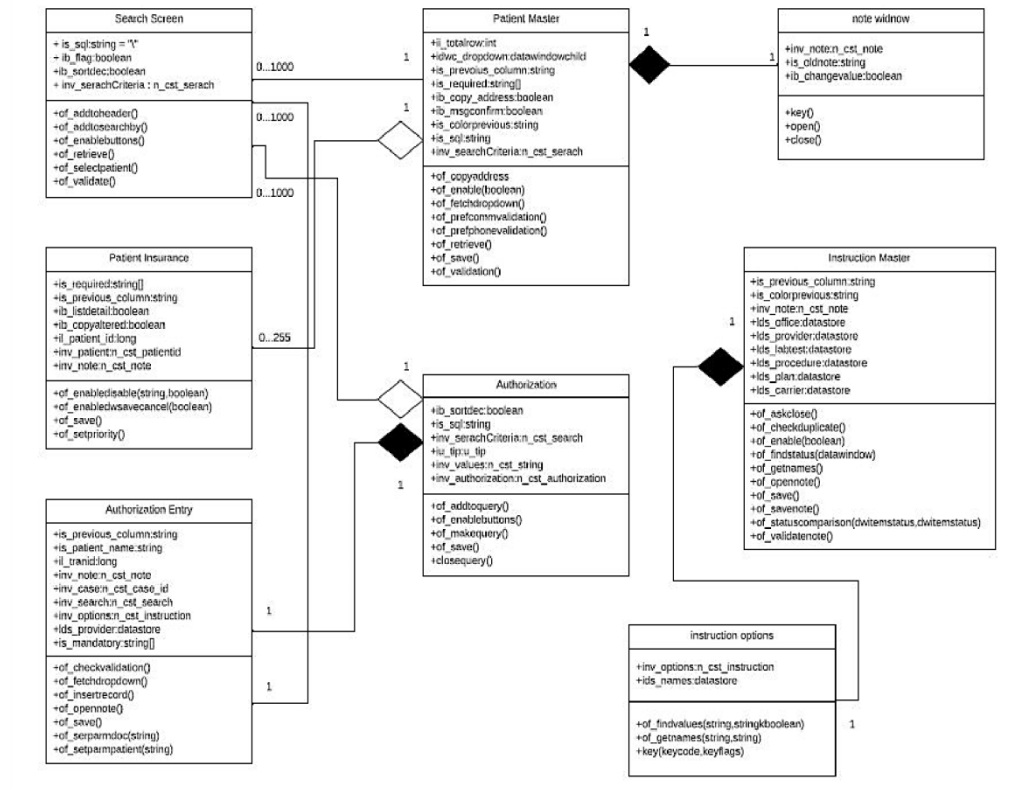


Fig 5.4 Class Diagram[6]

5.9 Sequence Diagram

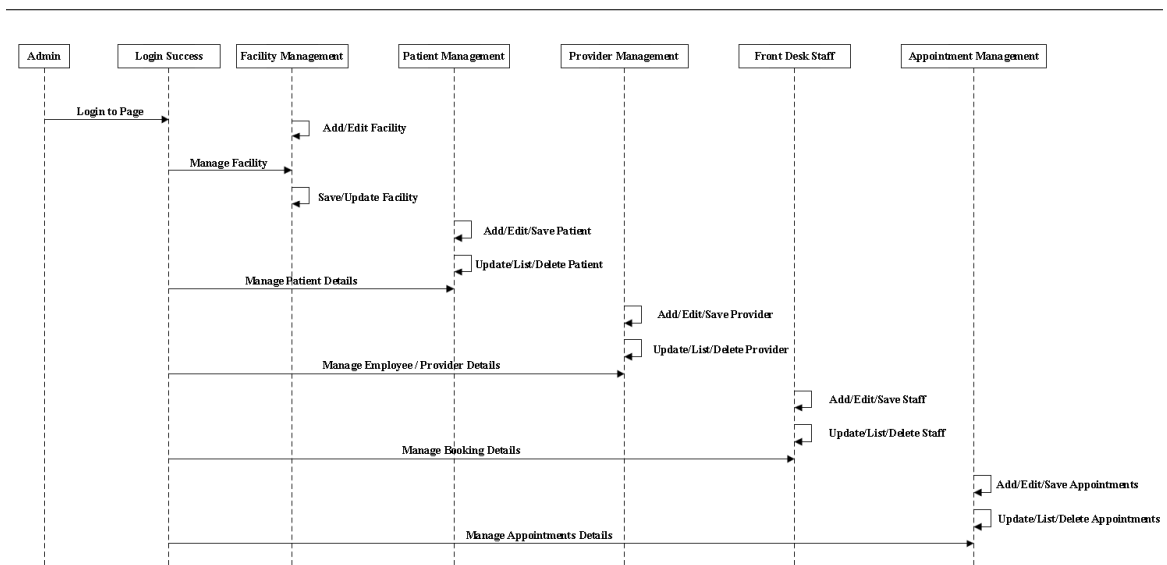


Fig 5.5 Sequence Diagram

5.10 ER Diagram (System Generated)

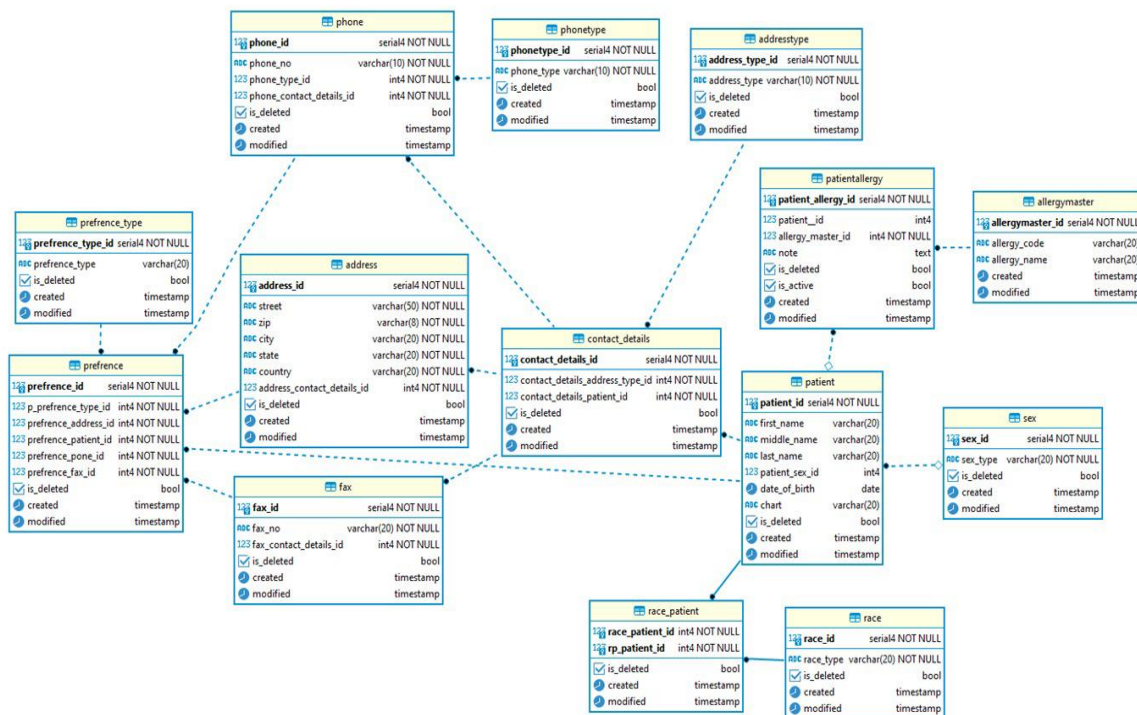


Fig 5.6 ER Diagram

5.11 List Main Modules of New System

- **Patient Management:** This module allows healthcare providers to manage patient demographics, medical history, and clinical notes, including diagnosis, treatment plans, and progress notes.
- **Appointment Scheduling:** This module allows healthcare providers to schedule, reschedule, and cancel patient appointments, as well as send automated appointment reminders to patients.
- **Billing and Claims Management:** This module allows healthcare providers to create, manage, and submit claims, as well as generate billing reports and handle patient billing inquiries.
- **Prescription Management:** This module allows healthcare providers to create and manage electronic prescriptions, handle refill requests, and maintain prescription records.

- **Clinical Workflows:** This module allows healthcare providers to create, manage, and customize clinical workflows, such as order sets and protocols, to improve clinical processes and patient outcomes.
- **Admin:** This module allows admin to create new job titles, manage patient and employee data. It allows admin to create, update and delete employee data.
- **Patient Ledger:** In this module healthcare organization can store and read patient information and it is used to store patient forms and history data.

5.12 Selection of Hardware & Software And Justification

❖ Hardware

- RAM: 4.00GB;
- Processor: Intel Core i5 CPU @ 3.20GHz
- Operating System: Windows 7 or higher version
- System type: 64-bit operating system; Graphics: Intel HD Graphics

❖ Software

- 64-bit operating system.

*Given hardware and software are selected as per the company requirements, guidelines and policies.

6. System Design

6.1 System Application Design

6.1.1 Method Pseudo Code

- Login Module
- Credentials checked with the database for the login process.
- Open Main page
- Main application frame will open after successful login.
- Retrieve Patient List
- Data of the patient is retrieved in the Patient Master Module.
- Data is retrieved from the database on the basis of the specific search keyword.
- Authorization Tracking Module
- The claims generated by the providers or billers are sent to the insurance company.
- The tracking of the claims are handled in this module.
- This consists of sending the claim, changing the status of the claim and submitting them to the insurance company.

6.2 Input/Output And System Design

The interface Will be in the form of web page.

6.2.1 Home Page

This is home page of our website.



Fig 6.1 Home page

6.2.2 Patient Demographics Page

In this page user add patient data and create new patient or update patient data and save them. On the left side there is navigation side bar from which user can select appropriate option.

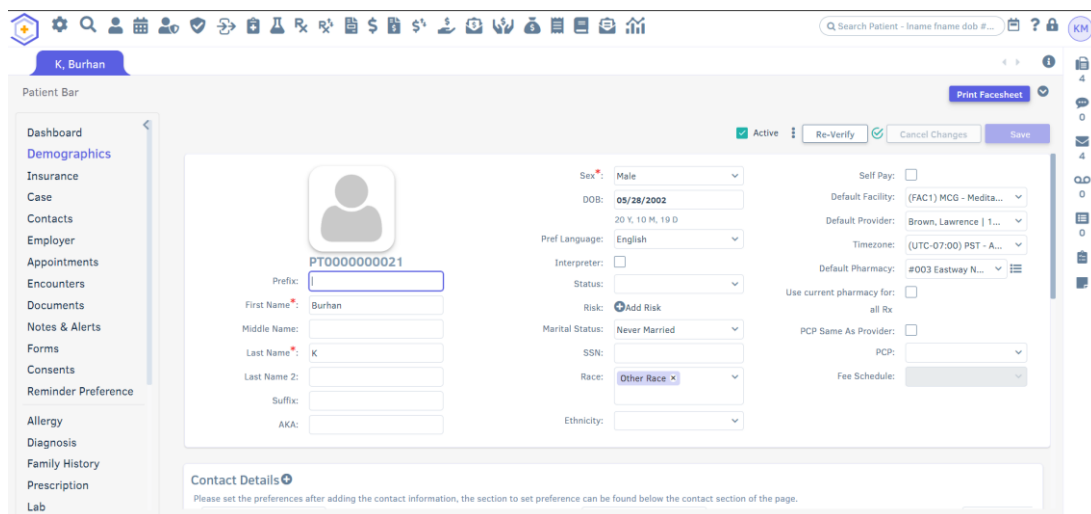


Fig 6.2 Patient Demographics Page

6.2.3 Patient Contact Section

In this section user can add patient contact details like their address or phone numbers and also having the facilities of adding multiple address and contacts.

Fig 6.3 Patient Contact Section

6.2.4 Patient Contact Preference Details Section

In this section user can select patients' preferred address and contacts. All details will be automatically suggested by the saved contact details.

Type	Address	Phone i	Fax	Email i	Website
Appointment Reminder					
Mailing					
Primary					
Statement					

Preferred Comm. Type: ▼ Value: ▼

Fig 5.4 Patient Contact Preference Details Section

6.2.5 Patient Other Contact Details Section

In this section patients' other details can be added by user like employment or first called date or relatives details etc.

The form is titled "Other Details" and contains several input fields organized in three columns:

- Left Column:** Employment (dropdown: Full time employed), First Called Date (text: 02/16/21), Deceased Time (calendar icon), Mother's Last Name (text), Birth Place (text), Religion (dropdown).
- Middle Column:** Gender Identity (dropdown: Male), Referral Source (dropdown), Mother's Maiden (text), Mother Identifier (text), Multiple Birth (checkbox), Driver's License (text).
- Right Column:** Sexual Orientation (dropdown), Deceased (checkbox), Mother's Middle Name (text), Veteran Military Status (dropdown), Nationality (dropdown), Citizenship (dropdown), Note (text area).

Fig 6.5 Patient Other Contact Details Section

6.2.6 Facility Page

In this page all facility of hospital is set from here and all this setup from here.

The screenshot shows a web application interface for managing facilities. The top navigation bar includes "Dashboard", "Implementation", "Facility Type", and "Facility" (selected). The main content area is titled "Facility" and shows details for "(FAC1) Creekside Clinics, LLC".

Facility Details:

- Code:** FAC1
- Name:** Creekside Clinics, LLC
- Facility Type:** Office/Medical Group
- NPI Type:** (dropdown)
- NPI:** (text)
- Timezone:** (UTC-07:00) PST - America/Los_Angeles
- POS:** (11) Office
- WCB Rating Code:** (dropdown)
- Primary Specialty:** (207Q00000X) Family Medicine
- Other Specialties:** (dropdown)
- Federal Tax Id Type:** (dropdown)
- Federal Tax Id:** (text)
- Parent Facility:** (dropdown)
- Service Location:** (checkbox, checked)
- Color:** (color picker)
- Sequence:** 1
- Note:** (text area)
- Child Facilities:** (list)

At the bottom, there is a pagination bar showing "1 - 1 of 1" and a status bar at the very bottom indicating "Waiting for www.facebook.com..."

Fig 6.6 Facility Page

6.2.7 Dashboard Page

In this page all report which is brief of the system is display in the graph form

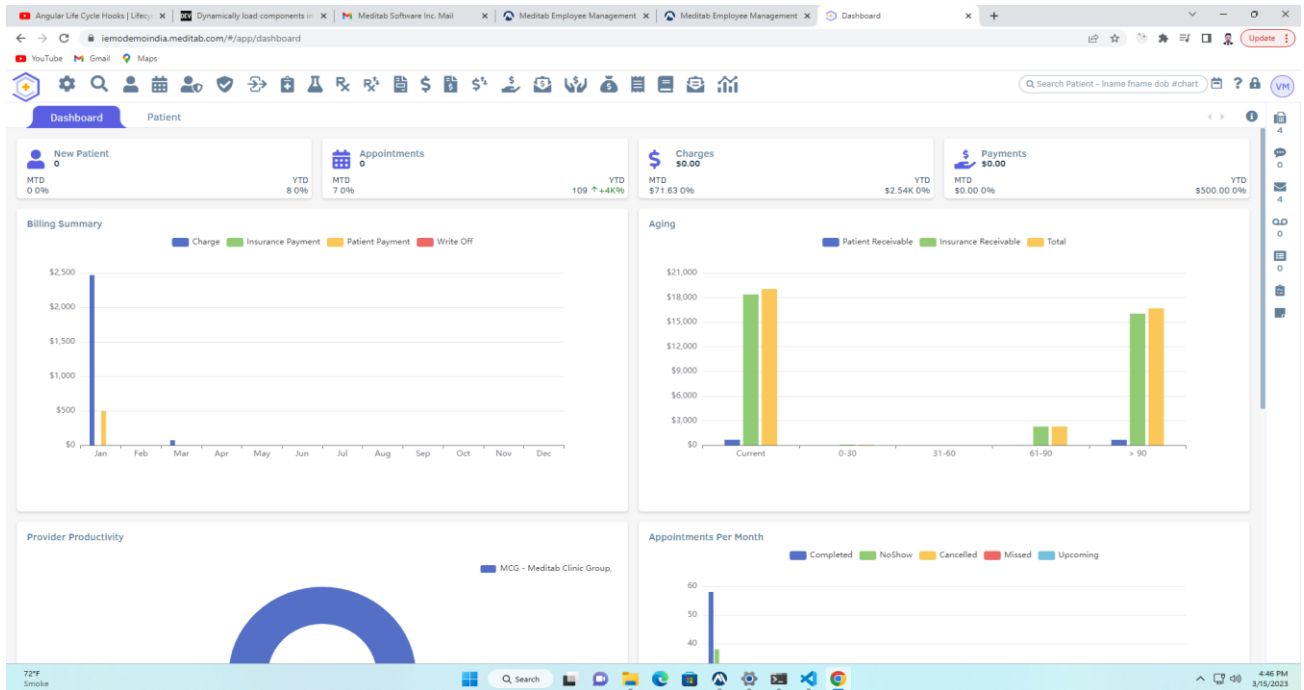


Fig 6.7 Dashboard Page

6.2.8 Setup Page

In this page all setup for website is available so we can setup website

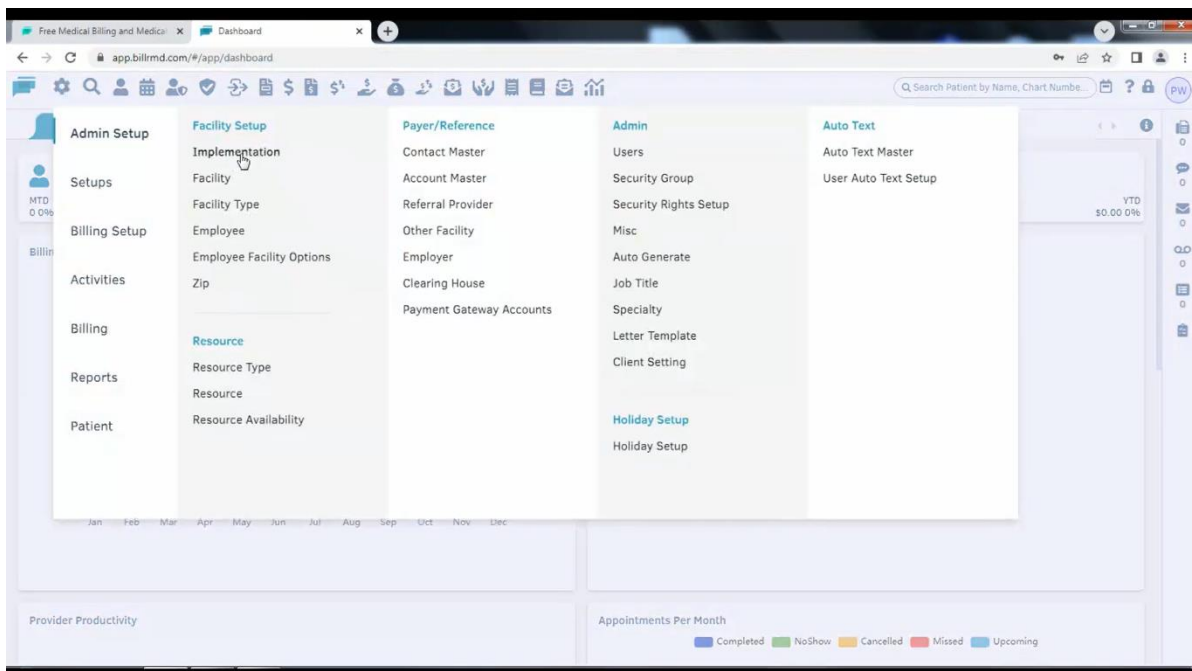


Fig 6.8 Setup Page

6.2.9 Checked In/ Checked Out Page

In this page use to know when check in patient or check out patient

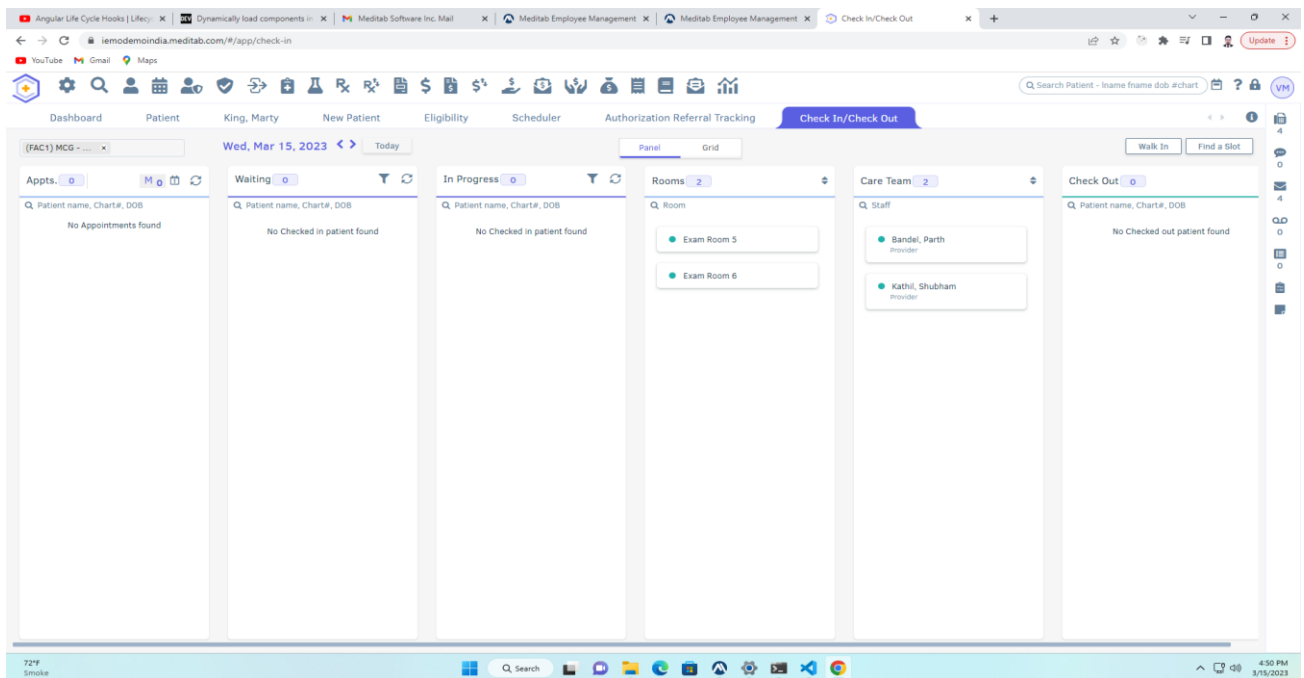


Fig 6.9 Checked In/ Checked Out Page

6.2.10 Implementation Page

In this page all implementation of the whole project is visible and we check direct from here.

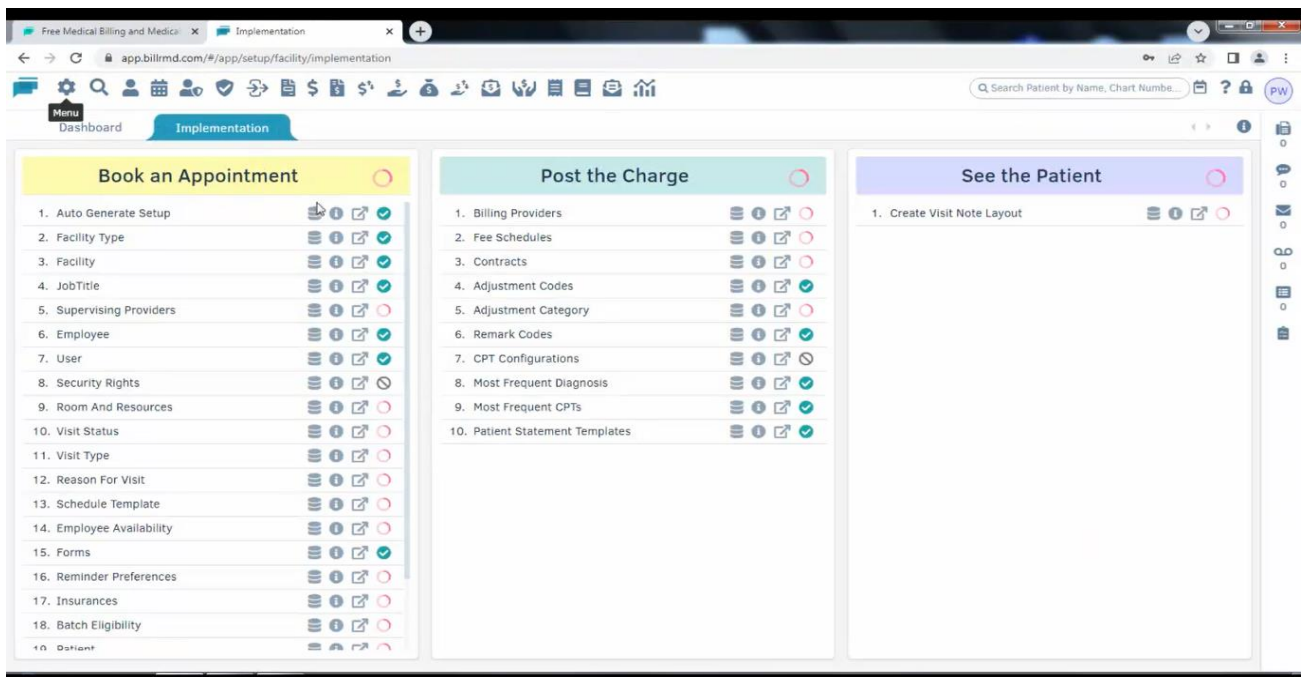


Fig 6.10 Implementation Page

7. Implementation Planning

Planning is an essential aspect of any successful project. However, it can be difficult to turn goals and strategy into tangible action, and, therefore, projects of all sizes and across all industries have a high tendency to fail. Implementation planning can reduce this chance of failure by helping turn strategy into action.

7.1 Implementation Environment

The application is a single server multiple client application.

Multi-user vs. Single-user: Single user applications are the application where it is useful to only one user at a time. While in Multi user given application is used by many users at the same time and thus web application is used by many users at the same time. Our system is a multi-user system as we can have many healthcare practitioners can use application at same time.

GUI vs. Non-GUI: Non-GUI application uses command Prompt for input and output while GUI application has graphics form to interface and other graphics property for various I/O operations and are easy to use Our System is a GUI based and thus easy and effective to use therefore users can easily give commands and can get specified results.

7.2 Program/Module Specification

1. **Primary Design Phase:** The system is designed at block level. These all blocks are created on the basis of study analysis done in the problem identification phase. It will be minimizing the information flow between blocks by different blocks are created for different function. Thus, all activities which require more interaction are kept on one block.
2. **Secondary Design Phase:** In the secondary phase the detailed design of each block is performed.

The general tasks involved in the design process are the following:

1. Create various blockchain processes for the entire system.
2. Create small, integrated and usable modules for each block.
3. Create multi-database frameworks.
4. Specify program details to achieve the performance you want.
5. Create input method, and system results.
6. Create design texts.
7. System updates.

7.3 Security Features

- Login credential validations.
- Patient Master Fields validation.
- Search filters validations.
- Security features regarding the Authorization Tracking.

7.4 Coding Standards

To develop reliable and maintainable applications, you must follow coding standards and best practices. The naming conventions, coding standards and best practices described in this document are compiled from our own experience and by referring to various guidelines. There are several standards that exist in the programming industry. None of them are wrong or bad and you may follow any of them. What is more important is, selecting one standard approach and ensuring that everyone is following it.

In this phase of software development, the design is related to a system converted into a machine-readable code that can be compiled and executed. Although the coding phase does not affect the structure of the system, it has a great impact on the internal structure of the module, which affects the testability, under the stability of the system.

7.4.1 Naming Convention and Standards

Your peer programmers have to understand the code you produce. A coding standard acts as the blueprint for all the team to decipher the code. Simplicity and clarity achieved by consistent coding

saves you from common mistakes. If you revise your code after some time, then it becomes easy to understand that code. It's industry standard to follow a standard to be more quality in software.

7.4.2 Naming Convention

a) model field Naming Conventions (pascal case)

- Example: patientid = PatientId
- firstname = FirstName
- lastname = LastName
- middlename = MiddleName
- dob = DOB
- createdby = CreatedBy

b) parameter naming conventions (Camle case)

- objPatientBORequest_
- returnResponse
- isShowCount

c) Comment

commenting for each API services

- In event and function there should be comment on top as below :-

```
/// <summary>
```

```
/// This method returns the patient record by the firstname, lastname, sex and dateofbirth  
according to the pagenumber and pagesize and orderby sorting
```

```
/// </summary>
```

```
/// <example>
```

```
/// For example:
```

```
/// /api/Patient/GetList?PageNumber=1&PageSize=5
```

```
/// results in records of patient having paginayion of pagesize of 5
```


records and 1 page number

```
/// </example>

/// <param name="patient_obj"></param>

/// <param name="PageNumber">1</param>

/// <param name="PageSize">10</param>

/// <returns>

/// [{ "patient_id": 1,

      "chartnumber": "CHART001",

      "first_name": "Test",

      "last_name": "User",

      "middle_name": "testUser",

      "sex_id": 1,

      "sex": "MALE",

      "dob": "18/05/2001 12:00:00 AM",

      "isdeleted": false,

      "created_on": "0001-01-01T00:00:00",

      "modified_on": "0001-01-01T00:00:00",

      "allergyname": "MILK",

      "note": "note1",

      "orderby": "Patients.patient_id",

      "patientAllergyId": 0,

      "allergyMasterId": 0}]/// </returns>
```

CHAPTER 8: TESTING

8.1 TESTING PLAN

The test plan describes the test environment and the required test resources. It also provides measurable goals by which management can gauge testing. Furthermore, it facilitates communications within the test team, between the test team and the development team, and between the test team and management. It tells what to test.

The main test plan in our application includes:

For the Testing plan, we worked on Security rights testing particular screens. In that We have to check the user rights of VIEW, CREATE, UPDATE and DELETE and also, we have done developer testing for CRUD.[5]

Table 8.1 Tests Performed on the App

Test Case ID	Test Scenario	Test Data	Pass/Fail
T01	Check CRUD rights on Patient Details	Test data from the dataset	Pass
T02	Fill Form and obtain dynamic object with its data	Enter the data Manually	Pass
T03	Validation on input field works or not	leave some inputs empty to see validation	Pass
T04	Get patient data from backend and repopulate the data in Form	Test data object from backend GET by ID api call	Pass
T05	Update patient data	data obtained by get method mentioned in T04	Pass

T06	Relocate page when new patient is added	Enter the data Manually	Pass
T07	Relocate to home page when patient data not found or when patient data is deleted	Try to search patient form URL by?id=,or deleting patient	Pass
T08	Remove Changes on existing patient data	Data from get method mentioned in T04	Pass

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. Testing is the process of executing a program with the intent of finding errors.

Characteristics of a Good Test:

- Tests are likely to catch bugs
- No redundancy
- Not too simple or too complex

8.2 TEST STRATAGY

Testing is the process of analysing a software item to detect the differences between existing and required conditions and to evaluate the features of the software item. The Test Strategy decides how I are going to approach the testing in terms of people, tools, procedures and support. It tells how to test. Testability is one of the primary design goals for Training Management.

Each type of test has a distinct purpose. They are the following:

Unit tests: Unit tests are written by developers and run under a unit testing framework. Unit tests isolate and verify discrete units of program logic. They isolate the logic by replacing

dependencies on the run-time environment, such as API call, with test-provided substitutes. Isolation allows unit tests to run quickly, and Developers can run unit tests frequently.

Integration tests: Integration tests differ from unit tests in that the code under test is not isolated. Developers or testers write integration tests. They run in a unit-testing framework.

Acceptance tests: Acceptance tests consist of multiple steps that represent realistic usage scenarios of the application as a whole. These tests verify that an application meets the needs of the intended users. Their scope includes usability, functional correctness, and performance. Generally, test engineers create these tests.

8.3 TEST SUITS DESIGN

8.3.1 Test Cases:

Following are the test cases for which the application goes under the test check before it is deployed.

Test Suite: 1

Test suite detail: Checking login module as shown in Table 8.2:

Table 8.2 Test Suite: 1

Test Case ID	Function Name	Test Case	Expected Result	Actual Result	Pass/Fail
1	Login Credentials	Login with wrong credentials	Should show alert message	Alert box is showing	Pass
2	Login Credentials	Login with right credentials	Should show success message	Success message showing	Pass

3	Login Credentials	One of the loginfailed	Next step should not be enabled	Error displayed	Pass
4	Login Credentials	Success in both login	Next step should be enabled	Redirected to home page	Pass
5	Login Credentials	While entering the user id according to that user the office must be auto selected in the below drop down	Selection of user's default office on pressing tab	As Expected	Pass
6	Login Credentials	On entering the system as the login credentials must pop-up a another login screen that is only valid for the organization members	Pop-up of another login screen.	As Expected	Pass

Test Suite: 2

Test suite detail: Checking client grid and selection of clients:

Table 8.3 Test Suite: 2

Test Case ID	Function Name	Test Case	Expected Result	Actual Result	Pass/Fail
1	Open Patient Master	On click of patient icon on the icon toolbar	Should Open Patient Master	Opening Patient Master module	Pass
2	Open Patient Master	Search existing client	Should Open Patient Master	Opening Patient master module	Pass
3	Open Patient Master	Setup>>Patient on the menu bar	Should Open Patient Master	As Expected	Pass
4	Patient Master GUI Verification	Visibility, alignment, spelling, tab sequence, clickable fields, editable fields works properly or not	All the icons must be properly visible and align with no spelling error as well as with proper tab sequence and proper functioning of all the fields.	As Expected	Pass
5	Patient Master Default Values	On Addition of a new patient in the patient master.	Active and SOF checkbox must be checked. Office must be selected. Date must be in this 00/00/0000 format.	As Expected	Pass

6	Patient Master Required Values	On Addition of a new patient in the patient master.	Proper email validation, and according to that only related field must activate. Phone number mustn't be kept half filled	As Expected	New
7	Patient Master Validation checking	On Addition of a new patient in the patient master.	Proper email validation, and according to that only related field must activate. Phone number mustn't be kept half filled	As Expected	Pass
8	Patient Master CRUD Operations	On Opening the patient master	Must successfully perform all the crud operations.	As Expected	Pass

Test Suite: 3

Test suite detail: Checking project grid and selection of projects:

Table 8.4 Test Suite: 3

Test Case ID	Function Name	Test Case	Expected Result	Actual Result	Pass/Fail
1	Open Authorization Tracking Screen	Clicking the icon on the icon bar	Should open authorization Tracking screen	As Expected	Pass
2	Open Authorization Tracking Screen	Activities >> Authorization Tracking	Should open authorization tracking screen	As Expected	Pass
3	Authorization tracing CRUD operation	On opening the Authorization tracking screen	Must successfully perform all the crud operations	All CRUD APIs working fine	Pass
4	Authorization tracking filter bar	On opening the authorization tracking	All the drop-down must filter properly and must retrieve apt results	As Expected	Pass
5	Authorization tracking Report Generation	On clicking the print/fax	If not selected any row then must prompt for selecting at least one row for generation of report	As Expected	Pass

6	Authorization GUI verification	Check the Visibility, alignment, spelling, tab sequence, clickable fields, editable fields works properly or not	All the icons must be properly visible and align with no spelling error as well as with proper tab sequence and proper functioning of all the fields.	As Expected	Pass
---	--------------------------------------	--	--	-------------	------

Test Suite: 4

Test suite detail: Checking Copy Module:

Table 8.5 Test Suite: 4

Test Case ID	Function Name	Test Case	Expected Result	Actual Result	Pass/Fail
1	Open Instruction Window	Setup>> Other>> Instruction	Should open the instruction window	As Expected	Pass
2	Instruction GUI verification	Check the Visibility, alignment, spelling, tab sequence, clickable fields, editable fields works properly or not	All the icons must be properly visible and align with no spelling error as well as with proper tab sequence and proper Functioning of all the fields.	As Expected	Pass

3	Instruction CRUD operation	On opening the instruction window	Must successfully perform all the crud operations	As Expected	Pass
4	Instruction window required fields and drop- downs	On opening the instruction window	All the drop-down must filter properly and without filling the required fields it should not be saved.	As Expected	Pass
5	Instruction window default values.	On opening the instruction window	All the drop-down must be default selected to “All” except the insurance field and Active must be checked.	As Expected	Pass

Test Suite: 5

Test suite detail: Checking Authorization Tracking window

Table 8.6 Test Suite: 5

Test Case ID	Function Name	Test Case	Expected Result	Actual Result	Pass/Fail
1	Open Authorization Tracking Screen	Clicking the icon on the icon bar	Should open authorization Tracking screen	As Expected	Pass

2	Open Authorization Tracking Screen	Activities >> Authorization Tracking	Should open authorization tracking screen	As Expected	Pass
3	Authorization tracing CRUD operation	On opening the Authorization tracking screen	Must successfully perform all the crud operations	As Expected	Pass
4	Authorization tracking filter bar	On opening the authorization tracking	All the drop-down must filter properly and must retrieve apt results	As Expected	Pass
5	Authorization tracking Report Generation	On clicking the print/fax	If not selected any row then must prompt for selecting at least one row for generation of report	As Expected	Pass
6	Authorization GUI verification	Check the Visibility, alignment, spelling, tab sequence, clickable fields, editable fields works properly or not	All the icons must be properly visible and align with no spelling error as well as with proper tab sequence and proper functioning of all the fields.	As Expected	Pass

9. Conclusion

9.1 Self Analysis of Project Viabilities

- Fetching a particular data while searching from the database will take time.
- At a time, multiple users can login and use the system easily.
- Deletion of data is not allowed as it might be used by some other user, so it turned inactive.

9.2 Problem Encountered And Possible Solution

9.2.1 Problem Encountered

In Patient Master, for some clinics the choice for required fields were different while for others were different. So which fields to make required so that issue can be solved.

9.2.2 Possible Solution

For the above problem, I made a setup where the client can appropriately choose his/her own required fields and set it from the back end without changing the code.

9.3 Summary of Project Work

Dr.Catalyst EHR is Web based software. It makes the life of providers (doctors) much easier by converting file-based systems to software-based systems. This is done by registering the entire patient in the IMS software and managing all the patient related information in the patient master module. Medical Insurance is mandatory in the US. To manage the patient insurance our system uses the patient insurance module to keep the records of the patient insurance and help to track all the claims. Thereafter, as the number of patients is huge, a search screen module is available in our software. This makes it easier for the user to search the patients, insurances, providers, etc. Instruction master is used to provide the facility of jotting down predefined instructions for a particular action. Authorization tracking module is used to track the claims that need to be sent to the insurance company.

References

1. <https://blog.bitsrc.io/understanding-asynchronous-javascript-the-event-loop>
“Medium.” *Medium*, blog.bitsrc.io/understanding-asynchronous-javascript-the-event-loop.
Accessed 3 May 2023.
2. <https://roadmap.sh/angular>
Ahmed, Kamran. “Angular Developer Roadmap: Learn to Become a Angular Developer.”
Roadmap.sh, roadmap.sh/angular. Accessed 3 May 2023.
3. <https://angular.io/featureshttps://angular.io/guide/what-is-angular>
“Angular.” *Angular.io*, angular.io/featuresangular.io/guide/what-is-angular. Accessed 3 May 2023.
4. <https://front.com/blog/email-etiquette-rules-in-the-workplace>
“16 Workplace Email Etiquette Rules for Communicating with Co-Workers and Customers | Front.” *Front.com*, front.com/blog/email-etiquette-rules-in-the-workplace.
5. <https://www.edureka.co/blog/bugs-in-software-testing/>
“Bugs in Software Testing – What, Where and How.” *Edureka*, 5 Aug. 2019,
www.edureka.co/blog/bugs-in-software-testing/.
6. <https://www.advancedmd.com/emr-ehr-software/>
“EHR Software (Electronic Medical Records) | AdvancedMD.” *AdvancedMD*, 2015,
www.advancedmd.com/emr-ehr-software/.
7. <https://www.cms.gov/Medicare/E-Health>