Independent University, Bangladesh

Department of Computer Science and Engineering
Internship Report
On

Genesis-D Admissions Module

At

Bioforge Health Systems Limited

Submitted By,

Khandaker Sajid Mahmud, ID: 1610125

Submitted to,

Ajmiri Sabrina Khan, Lecturer

Department of Computer Science and Engineering

Submission Date: 20/01/21

Semester: Autumn

Web application Development of “Genesis-D Admissions Module” at Bioforge Health Systems Ltd.

An undergraduate internship report submitted by

Khandaker Sajid Mahmud (ID: 1610125)

has been approved on 27/01/2021

Ajmiri Sabrina Khan

Internship Supervisor & Lecturer

Department of Computer Science and Engineering

School of Engineering & Computer Science

Independent University, Bangladesh

Letter of Submission

20 January 2021
Ajmiri Sabrina Khan
Lecturer,
Department of Computer Science and Engineering,
Independent University, Bangladesh

Subject: Letter of Submission for Internship Report, Autumn 2021

With due honor and respect, I, Khandaker Sajid Mahmud, from Autumn 2021, Section 12, would like to submit my Internship report. This report is written to kindly to inform you that I have completed my internship program and its report. My internship was conducted from 1st November 2020 to 23rd January 2021 at Bioforge Health Systems Limited.

This report is based on my personal experience and the work I did at Bioforge Health Systems Limited during my internship. The primary goal for my internship was to gain first-hand experience in all the different technology related fields of the company which include documentation, software development, research and development and to get acquainted with software development processes and practices with emphasis and priority on understanding how a software is being built rather than what is being built.

Over the period of my internship at Bioforge Health Systems Limited, I found out that I learned and applied a lot of new skills and technologies. The company comprises a small team of software craftsmen who learn, collaborate, and innovate together.

I hope the following report can achieve your approval and is up to the mark.

Sincerely,

____

Khandaker Sajid Mahmud, 1610125

Email: 1610125@iub.edu.bd

Acknowledgements

I would firstly like to thank The Almighty Allah for giving me the endurance and the ability to work hard, for giving me the ability to write this report to and for giving me the chance to be able to do my internship at Bioforge Health Systems Limited. Also, my parents for their unconditional love and support that have sustained, nurtured, and got me ready for this challenge.

I would like to thank my honorable faculty and supervisor Ms. Ajmiri Sabrina Khan, Lecturer, Department of Computer Science & Engineering, Independent University, Bangladesh, for her invaluable guidance, patience, time, constructive criticism and thoughtful advice regarding various aspects of my internship and preparation of this report.

I would like to thank my senior advisors Dr. Dewan AFK Choudhury, Ju-un Nahar Choudhury and all the others who made me feel at home from day one in the company and helped me navigate throughout the projects. I would also like to thank Mr. Mohammad Hossain for his sincere guidance in the project. I am thankful for the continuous guidance and support along with the vast pool of knowledge which was key for the completion of the project.

Lastly, I would like to acknowledge my external supervisor and my mentor Ms. Shama Hoque for supporting, guiding and being patient with me as an Intern for Bioforge Health Systems Limited. Without her extreme energetic support and guidance, I could not finish the project successfully.

Table of Contents

Letter of Submission 3

Acknowledgements 4

Table of Contents 5

List of Figures 7

List of Tables 7

Executive Summary 8

Chapter 1: Introduction 9

- Background of the work 9
- Objectives 10
- Scopes 10

Chapter 2: Literature Review 11

- How the work/project is related to the undergraduate courses 11
- Related works: 12

Chapter 3: Methodology 14

- Web Applications 14
- Waterfall Model Methodology 14
- Development Tools Used 15
- Non-Development Tools Used 19

Chapter 4: Project Management & Financing 21

- Work Breakdown Structure 21
- Gantt Chart 1

Chapter 5: Body of the Project 1

- Description of the work 1
- System analysis 1
- System Design 2
  - Rich Picture 2
  - Use Case Diagram 3
  - Process Flow Diagram 4
  - Entity Relationship Diagram (ERD) 5
  - Functional Requirements 6

Non-Functional Requirements 9
Product Features 10
Input 10
Output 10
Architecture of the System: 10
Chapter 6: Result and Analysis 13
Chapter 7: Project/work as Engineering problem analysis 17
Sustainability of the product/work 17
Social and Environmental effects and analysis 17
Addressing Ethics and ethical issues: 18
Chapter 8: Future Work and Conclusion 19
Future Work 19
Conclusion 20
References 20
Appendix - A (Backend Code Snippets) 21

List of Figures

Figure 2.1: Smart Software Appointment Booking User Interface 12
Figure 2.2: Millenium Heart and General Hospital Admissions System 13
Figure: 3.1: Angular Logo 15
Figure 3.3: Node.js logo 16
Figure 3.4: Express.js logo 17
Figure 3.5: MongoDB logo 17
Figure 3.6: Git logo 18
Figure 3.7: Trello Task Management Board 19
Figure 3.8: Commits in Github 20
Figure 4.1: Gantt Chart with Resource Allocation and Activity wise Time Distribution 1
Figure 5.2: Rich picture for Genesis-D Admissions Module 2
Figure 5.3: Use Case Diagrams for Genesis-D Admissions Module 3
Figure 5.4: Process flow Diagram for Genesis-D Admissions Module 4
Figure 5.5: Entity Relationship Diagram for Genesis-D Admissions Module 5
Figure 5.6: 3-tier Client Server Architecture 11
Figure 5.7: Model - View - Controller (MVC) Pattern 12
Figure 6.1: Screen View of Login Page 13
Figure 6.2: Screen view of Dashboard 14
Figure 6.3 & 6.4: Screen View of Admissions Details 15
Figure 6.5: Screen View of Admissions Summary 15
Figure 6.6: Screen View of ID card generation 16

List of Tables

Table 5.1: Six Element Analysis for Genesis-D Admissions Module 2
Table 5.2: Functional Requirement 1: Log in 6
Table 5.3: Functional Requirement 2: Enter Admissions Details 6
Table 5.4: Functional Requirement 3: Admissions Summary must fetch the inputted values. 7
Table 5.5: Functional Requirement 4: Admissions Summary to be printed in PDF. 7
Table 5.6: Functional Requirement 5: Generate a patient ID card. 8
Table 5.7: Functional Requirement 6: Print patient ID card 8
Table 5.8: Functional Requirement 7: Responsive 8

Executive Summary

In this report, I have written about my experience as an intern at Bioforge Health Systems Limited. Bioforge Health Systems Limited is a biomedical hardware and software company, dedicated to support hospitals and doctors deliver effective and precision healthcare using innovative technology. The company was founded back in 2017 with a handful of engineers and a doctor ready to change the medical industry here in Bangladesh for the better. In 2020, Bioforge and the team became MIT Solvers and recipients of the Bill and Melinda Gates Foundation Award for their ongoing low-cost and portable incubator and phototherapy lights project. As an intern, I was assigned to the software team in the company. During the initial phase of the internship, I was faced with many new technologies which I had never encountered before as I did not have any formal experience in the software industry. Bioforge Health Systems Limited gave me the opportunity to develop myself and introduced me to the software development process where I can use the knowledge, I have gained in the future to improve myself.

During my internship I was introduced to the technologies that are used in the Software Development phases. I was exposed to developing User Interfaces with a popular JavaScript framework -Angular. In this timeline I progressively got familiarized with some of the Software Engineering Process and tools which are involved in taking an application from Inception to Development and finally Production.

After completing all basic learning sessions, I was assigned to work on a company project where my task was to work in a team to develop a web application named Genesis-D Admissions Module and the idea of the application is to provide hospitals with a simple and efficient system to admit their patients and provide an ID card afterwards.

Chapter 1: Introduction

Background of the work

The project "Genesis-D Admissions Module" is being developed as a working part of a hospital management system to specifically take care of patient admission procedures. Hospitals are an essential part of the healthcare structure and thus require an efficient working system to handle all of its incoming patients' data. Information of all the patients entering a hospital needs to be smoothly managed and integrated into the hospital's treatment functionalities for it to run successfully. Inevitably, using a paper-based system, to keep track of this large amount of everyday data would be time-consuming and prone to errors. To mitigate this issue, this project will work on creating an admissions module to take in, store and classify patients' information in an automated manner.

The module is built as a Web application hosted on the user's device to allow for quick access to an admission details form. The patients' information will be stored as a record and simultaneously become linked to a suitable department. Data storage removes the need for duplicate records for each visit of the same patient and helps build on their medical history. Linking the data to the hospital's department allows not only for bed allocation to become fixed on arrival but also for medical personnel to have easy access to patient history and other relevant information.

The system output not only brings patients' details on screen but allows it to be printed for patient identification purposes. The printed output is a form of ID card with a QR code for easy scanning and retrieval of patients' data. A unique identification system for patients has been shown to improve quality of care and management provided by healthcare facilities while also limiting the amount of personal information being shared.

The system aims to replace any paper-based system by offering a more efficient alternative. The module will be primarily operated by medical secretaries, such as administrators or receptionists, at a hospital's reception area or its emergency room where patient intake occurs. Data may get lost or misplaced in transfer to other departments when using paper forms. Medical secretaries have particularly shown higher satisfaction to the use of hospital information systems as compared to other medical staff, making the use of an admissions module as a replacement less challenging [1]. The software also has the potential to be integrated into a developing hospital management system where the patient's data on admission can be updated through diagnosis, testing and treatment up to the patient's discharge. This opens the possibility to automate common hospital administrative and monitoring processes for better benefits to healthcare as a whole.

Objectives

The main objective for the Genesis-D Admissions Module is to fully automate the patient admission procedure from the reception or emergency room, all the way to the patients assigned bed and department.

Other objectives include:

1. To create a system with a user-friendly interface that is simple and efficient.
2. To automate the storage of patient records based on initial data input during the admission process to allow for secure access to data and retrieval of data for analysis and decision making.
3. To provide an ID card to the patient based on initial data input during the admission process for patient identification during the visit or any future visits.
4. Possibility of future integration with other modules of a hospital management system.

Scopes

The project is an alternative solution to the more complex and time-consuming paper-based system of admission procedures. It would save time and be more cost effective in the long term. Information can be shared automatically with the relevant nurses and physicians and with departments for allocation of patient bed and ward. It also facilitates the access to data by hospital personnel at any time. This is done with higher safety and accuracy allowing for better management of the patient. It also provides a physical ID for patient identification and retrieval of data.

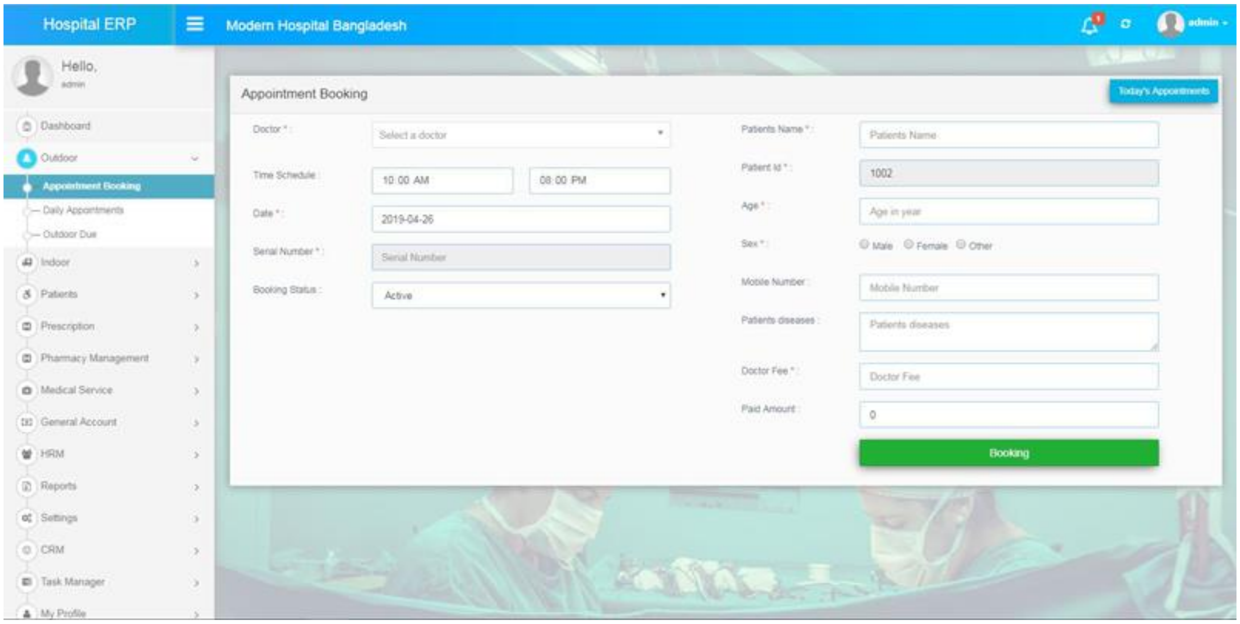
Chapter 2: Literature Review

How the work/project is related to the undergraduate courses

In my last 4 years here at Independent University, Bangladesh almost all my courses have contributed some significant knowledge in the development of the module. Out of which many courses were vitally important and played a huge role. These courses are as per the following in no specific order: -

- **CSE 213, Object Oriented Programming:** It is now a standard practice in the industry to be using object-oriented programming for any type of software application as most of the data represented are in objects. The course teaches on how to write modular based programs which can be recycled and therefore can be used many times for different purposes.
- **CSE 203, Data Structures:** In Data Structures we are taught about Stack, Queue, and Linked List utilizing the C++ programming language. The essential objective of this course is to make students and programmers imagine how unique data structures work. As “Genesis-D Admissions Module” involves many complex data structures, the skills gained from this course made handling them much easier.
- **CSE 309, Web Application and Internet:** This is the course where the development of web applications was taught. It covered very important technologies that are highly in demand in the industry, such as HTML, CSS, JavaScript, jQuery, View Engines (Handlebars and embedded JavaScript), Node.js, Express.js, MongoDB and deployment with Heroku. The tools and technologies learned from this course immensely contributed to the development of “Genesis-D Admissions Module” as it is a web application-built web technology, and it has a backend server which had to be deployed to the cloud server as well.
- **CSE 303, Database Management:** This was the first course which taught on how to design and plan a project. It covered popular planning and strategy practices such as System Development Life Cycle (SDLC), Rich Picture, Requirement Analysis, Entity Relationship Diagram, Business Process Model and Notation Diagram and many more.
- **CSE 307, System Analysis and Design:** This course gives an overview of different SDLCs and how to adopt each one of them to the project.
- **CSE 451, Software Engineering:** This course was particularly important as in this course we were taught how to manage our projects, make timelines, work with our master’s peers to make applications hands on which combined a lot of what we learned from our previous courses.

Related works:



Appointment Booking

Doctor * : Select a doctor

Time Schedule : 10:00 AM 08:00 PM

Date * : 2019-04-26

Serial Number * : Serial Number

Booking Status : Active

Patients Name * : Patients Name

Patient Id * : 1002

Age * : Age in year

Sex * : ○ Male ○ Female ○ Other

Mobile Number : Mobile Number

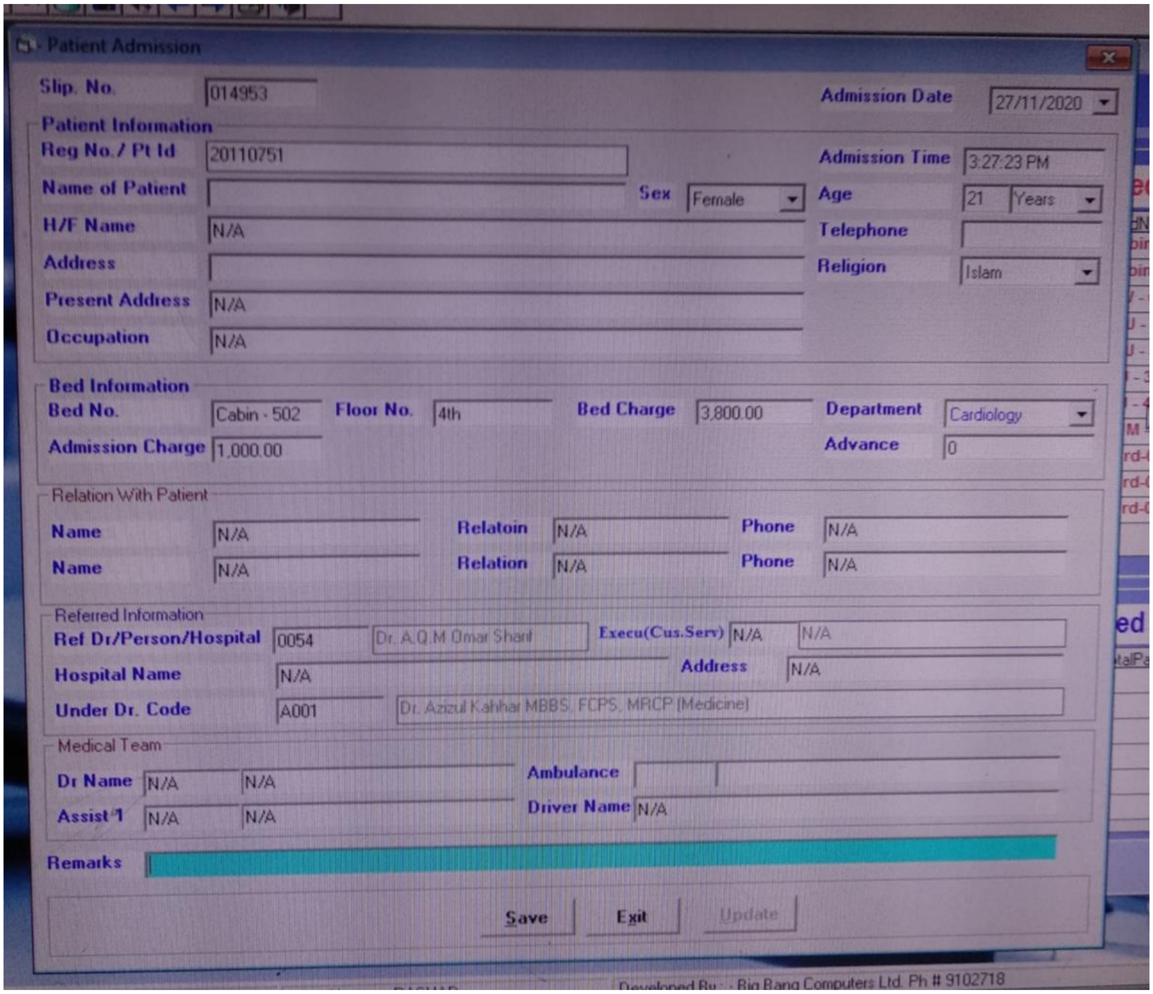
Patients diseases : Patients diseases

Doctor Fee * : Doctor Fee

Paid Amount : 0

Figure 2.1: *Smart Software Appointment Booking User Interface*

1. Smart Software [2] is a healthcare-based software company which develops different kinds of hospital management modules with cloud servers which are all integrated into one hospital management system. The Appointment module here is very similar to Genesis-D Admissions Module as it meets most of the objectives for our project.



Slip. No. 014953

Admission Date 27/11/2020

Patient Information

Reg No./Pt Id 20110751

Admission Time 3:27:23 PM

Name of Patient

Sex Female

Age 21 Years

H/F Name N/A

Telephone

Address

Religion Islam

Present Address N/A

Occupation N/A

Bed Information

Bed No. Cabin - 502

Floor No. 4th

Bed Charge 3,800.00

Department Cardiology

Admission Charge 1,000.00

Advance 0

Relation With Patient

Name N/A

Relatoin N/A

Phone N/A

Name N/A

Relation N/A

Phone N/A

Referred Information

Ref Dr/Person/Hospital 0054 Dr. A.Q.M Omar Sharif

Execu(Cus.Serv) N/A N/A

Hospital Name N/A

Address N/A

Under Dr. Code A001 Dr. Azizul Kahhar MBBS, FCPS, MRCP (Medicine)

Medical Team

Dr Name N/A N/A

Ambulance

Assist 1 N/A N/A

Driver Name N/A

Remarks

Developed By : - Big Bang Computers Ltd. Ph # 9102718

Figure 2.2: *Millenium Heart and General Hospital Admissions System*

2. Millenium Heart and General Hospital [3] situated in Lalmatia, Dhaka is currently using a desktop-based admissions application with a local server which is also a part of an entire hospital management system.

Chapter 3: Methodology

Web Applications

Web applications, sometimes referred to as web apps, are an interactive computer program built with web technologies often with HTML, CSS, and JS, with complete control of manipulation (CRUD) of data that can be stored in a database. CRUD is an acronym it stands for Create, Read, Update, and Delete. Web apps are generally accessed using a web browser such as- Google Chrome, Mozilla Firefox, etc. They usually involve some sort of sign up/log in mechanism. A web application (web app) does not need to be downloaded and is instead accessed through a network.

Waterfall Model Methodology

In software engineering, a software development process is the process of dividing software development work into distinct phases to improve design, product management and project management. It is also known as a system development life cycle (SDLC). We can define SDLC as a framework that describes the activities performed at each stage of a System Development Project. So, it has some basic stages to be followed during the development phase. There are many different SDLC to choose from like

- Waterfall Model
- Prototyping
- Agile
- Spiral Model
- Rapid Application Development
- V-Model
- Incremental
- Evolutionary Model

All methodologies have their own advantages and disadvantages. However, the one that worked great for us was the waterfall method. The waterfall model is a linear project management approach, where stakeholder and customer requirements are gathered at the beginning of the project, and then a sequential project plan is created to accommodate those requirements. The waterfall model is called as such because each phase of the project cascades into the next, following steadily down like a waterfall. [4]

We used the waterfall model methodology because we needed an easy to manage and a rigid model. As is evident when looking into our Gantt chart and WBS mentioned on Chapter 4 as the phases have been completed by the model which are:

- Requirements
- Design
- Implementation

- Verification
- Maintenance

More importantly, we wanted to divide our work into phases with specific deliverables and a review process. The fact that the requirements were reviewed multiple times with the client meant that we had very well-defined requirements which are not likely to change in the near future. There may be additional requirements in the coming iterations however the requirements now have been set in stone.

Development Tools Used

The project “Genesis-D Admissions Module” was developed using many different modern application development tools which consist of a stack of development tools called the MEAN stack. MEAN stack is a JavaScript Stack which is used for easier and rapid deployment of full-stack web applications. MEAN stack consists of 4 technologies namely: MongoDB, Express, Angular and Node.js. The four technologies provide an end-to-end framework for developers to be able to develop web applications.

1. Angular

Figure: 3.1: *Angular Logo*

Angular [5] is a platform and framework for building single-page customer applications utilizing HTML and TypeScript. Angular is written in TypeScript. It implements core and optional functionality as a set of TypeScript libraries that you import into your apps.

2. Node.js

Figure 3.3: *Node.js logo*

Node.js [6] is an open-source development platform for executing JavaScript code server-side. Node is useful for developing applications that require a persistent connection from the browser to the server and is often used for real-time applications such as chat, news feeds and web push notifications. Node.js is intended to run on a dedicated HTTP server and to employ a single thread with one process at a time. Node.js applications are event-based and run asynchronously. Code built on the Node platform does not follow the traditional model of receive, process, send, wait, receive. Instead, Node processes incoming requests in a constant event stack and sends small requests one after the other without waiting for responses. This is a shift away from mainstream models that run larger, more complex processes and run several threads concurrently, with each thread waiting for its appropriate response before moving on. It provides a JavaScript Environment which allows the user to run their code on the server (outside the browser). Node pack manager i.e., npm allows the user to choose from thousands of free packages (node modules) to download.

3. Express.js

Figure 3.4: Express.js logo

Express [7] is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. Rather than writing the code using Node.js and creating loads of Node modules, Express makes it simpler and easier to write the back-end code. Express helps in designing great web applications and APIs. Express supports many middleware which makes the code shorter and easier to write.

4. MongoDB

Figure 3.5: MongoDB logo

MongoDB [8] MongoDB is a cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

5. Git

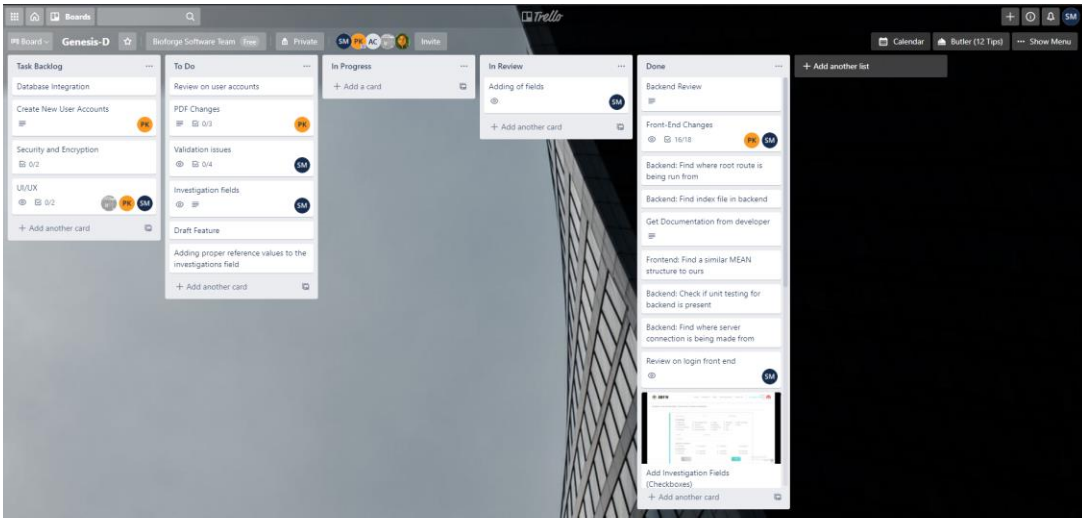
Figure 3.6: *Git logo*

Git [9] is a free, open-source distributed version control system. It is used for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. Its goals include speed, data integrity, and support for distributed, non-linear workflows.
Version control is a system that records changes to a file, or set of files, over time so that specific versions can be recalled later.

Non-Development Tools Used

Besides the development tools, other non-development tools were very critical in the development of the Genesis-D Admissions Module Application. The tools listed below have helped in keeping track of workflow, version control, repository, hosting etc.

1. Trello



Task Backlog

- Database Integration
- Create New User Accounts (PK)
- Security and Encryption 0/2
- UI/UX 0/2 (PK SM)
- + Add another card

To Do

- Review on user accounts
- PDF Changes 0/3 (PK)
- Validation issues 0/4 (SM)
- Investigation fields (SM)
- Draft Feature
- Adding proper reference values to the investigations field
- + Add another card

In Progress

- + Add a card

In Review

- Adding of fields (SM)
- + Add another card

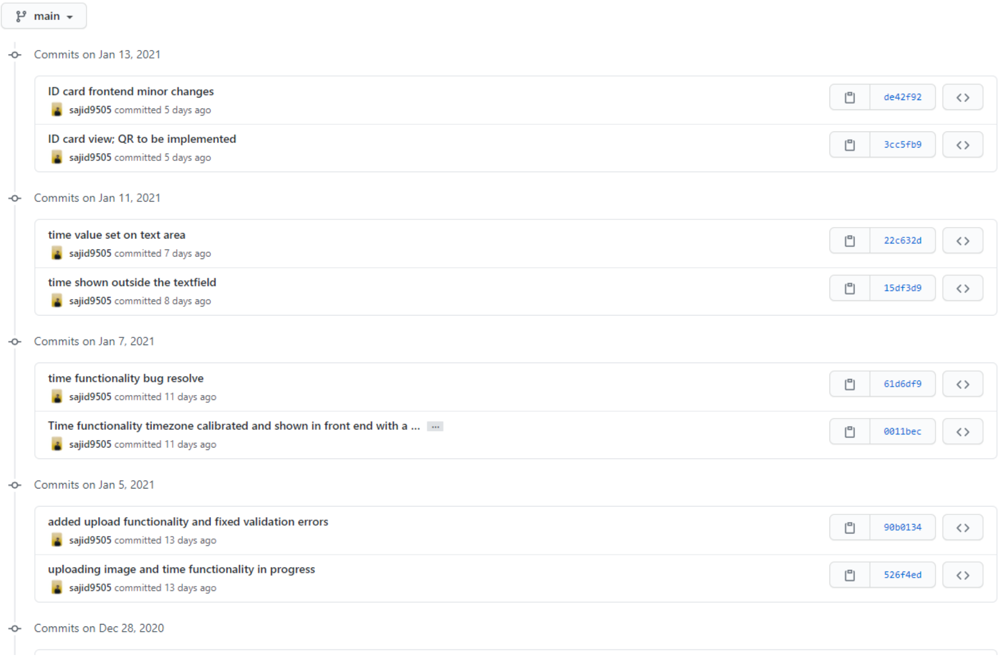
Done

- Backend Review
- Front-End Changes 16/18 (PK SM)
- Backend: Find where root route is being run from
- Backend: Find index file in backend
- Get Documentation from developer
- Frontend: Find a similar MEAN structure to ours
- Backend: Check if unit testing for backend is present
- Backend: Find where server connection is being made from
- Review on login front end (SM)
- Add Investigation Fields (Checkboxes)
- + Add another card

Figure 3.7: Trello Task Management Board

Trello [10] is a web-based Kanban-style list-making application which is a subsidiary of Atlassian. Users can create their task boards with different columns and move the tasks between them. Typically, columns include task statuses such as To Do, In Progress, Done. The tool can be used for personal and business purposes including real estate management, software project management, school bulletin boards, lesson planning, accounting, web design, gaming, and law office case management.

2. Github



Commits on Jan 13, 2021

ID card frontend minor changes
sajid9505 committed 5 days ago
de42f92

ID card view; QR to be implemented
sajid9505 committed 5 days ago
3cc5fb9

Commits on Jan 11, 2021

time value set on text area
sajid9505 committed 7 days ago
22c632d

time shown outside the textfield
sajid9505 committed 8 days ago
15df3d9

Commits on Jan 7, 2021

time functionality bug resolve
sajid9505 committed 11 days ago
61d6df9

Time functionality timezone calibrated and shown in front end with a ... …
sajid9505 committed 11 days ago
0011bec

Commits on Jan 5, 2021

added upload functionality and fixed validation errors
sajid9505 committed 13 days ago
90b0134

uploading image and time functionality in progress
sajid9505 committed 13 days ago
526f4ed

Commits on Dec 28, 2020

Figure 3.8: *Commits in Github*

GitHub, Inc. [11] is an American multinational corporation that provides hosting for software development and version control using Git. It offers the distributed version control and source code management functionality of Git, plus its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project. Headquartered in California, it has been a subsidiary of Microsoft since 2018.

Chapter 4: Project Management & Financing

Work Breakdown Structure

1. Genesis-D Admissions Module

1.1. Project Planning and Brainstorming
1.1.1. Planning of the overview for the project
1.1.2. Initial market and background research
1.1.3. Planning of basic system architecture
1.1.4. Design and development of Process Flow Diagrams
1.1.5. Development of Security Requirements
1.1.6. Development of Quality and Testing Requirements

1.2. Gathering Requirements and making an SRS
1.2.1. Making questionnaire for gathering functional requirements
1.2.2. Identifying Non-functional requirements
1.2.3. Gathering samples for UI UX development (Hospital Visits)
1.2.4. Making an SRS

1.3. Development of UI and Prototype
1.3.1. Development of UI wireframe and prototype
1.3.2. Art works Development
1.3.3. Graphic Asset development
1.3.4. Finalize UI and UX

1.4. Development and Unit testing of all components
1.4.1. Importing of all libraries and setting up environment
1.4.2. Front-End development from UI and UX and testing of each component.
1.4.3. Development of Back-End and unit testing of all features.
1.4.4. Development of Data Models
1.4.5. Development of APIs, Admin and Login components and unit testing of components.
1.4.6. Development and testing of Security components.
1.4.7. Integration and configuration
1.4.8. Bug fixing and issue resolution.
1.4.9. Finalize the development stages and Unit testing.

1.5. Testing
1.5.1. Integration of all components
1.5.2. CI/CD testing
1.5.3. Alpha testing within the company
1.5.4. Beta testing with the client
1.5.5. Gathering feedback and making changes accordingly
1.5.6. Finalize testing features.

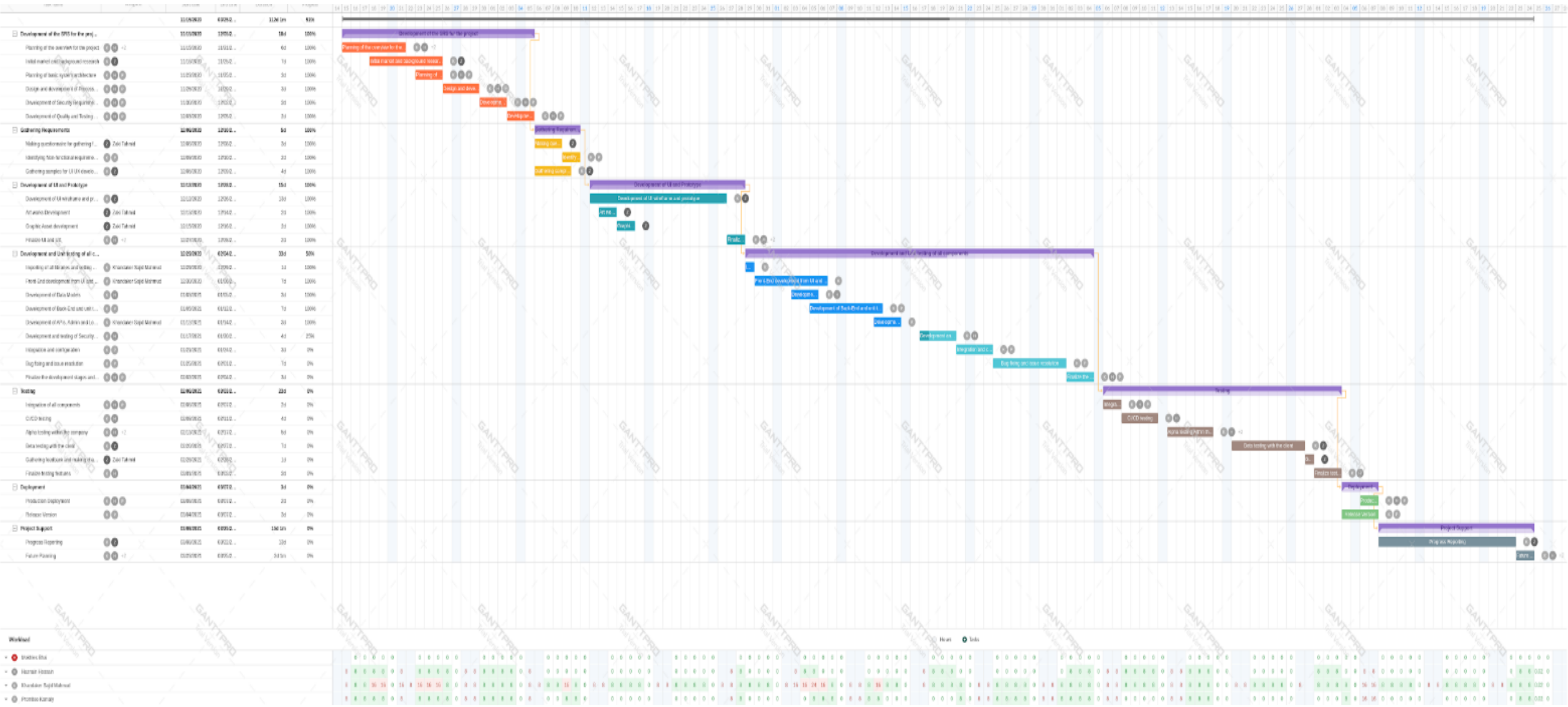
1.6. Deployment

- 1.6.1. Production Deployment
- 1.6.2. Release Version

1.7. Project Support

- 1.7.1. Future Planning
- 1.7.2. Progress Reporting

Gantt Chart



		11/16/2020	[illegible]	[illegible]	[illegible]
Development of the SRS for the proj…		11/15/2020	[illegible]	18d	100%
Planning of the overview for the project	+2	11/15/2020	[illegible]	6d	100%
Initial market and background research		[illegible]	[illegible]	7d	100%
Planning of basic system architecture		[illegible]	[illegible]	3d	100%
Design and development of Process…		[illegible]	[illegible]	3d	100%
Development of Security Requireme…		[illegible]	[illegible]	3d	100%
Development of Quality and Testing…		[illegible]	[illegible]	3d	100%
Gathering Requirements		[illegible]	[illegible]	5d	100%
Making questionnaire for gathering r…	Zaki Tahmid	[illegible]	[illegible]	3d	100%
Identifying Non-functional requireme…		[illegible]	[illegible]	2d	100%
Gathering samples for UI UX develo…		[illegible]	[illegible]	4d	100%
Development of UI and Prototype		[illegible]	[illegible]	15d	100%
Development of UI wireframe and pr…		[illegible]	[illegible]	10d	100%
Art works Development	Zaki Tahmid	[illegible]	[illegible]	2d	100%
Graphic Asset development	Zaki Tahmid	[illegible]	[illegible]	3d	100%
Finalize UI and UX	+2	[illegible]	[illegible]	2d	100%
Development and Unit testing of all c…		[illegible]	[illegible]	30d	50%
Importing of all libraries and testing…	Khandaker Sajid Mahmud	[illegible]	[illegible]	1d	100%
Front End development from UI and…	Khandaker Sajid Mahmud	[illegible]	[illegible]	7d	100%
Development of Data Models		[illegible]	[illegible]	3d	100%
Development of Back-End and unit t…		[illegible]	[illegible]	7d	100%
Development of APIs, Admin and Lo…	Khandaker Sajid Mahmud	[illegible]	[illegible]	3d	100%
Development and testing of Security…		[illegible]	[illegible]	4d	25%
Integration and configuration		[illegible]	[illegible]	3d	0%
Bug fixing and issue resolution		[illegible]	[illegible]	7d	0%
Finalize the development stages and…		[illegible]	[illegible]	3d	0%
Testing		[illegible]	[illegible]	23d	0%
Integration of all components		[illegible]	[illegible]	2d	0%
CI/CD testing		[illegible]	[illegible]	4d	0%
Alpha testing within the company	+2	[illegible]	[illegible]	6d	0%
Beta testing with the client		[illegible]	[illegible]	7d	0%
Gathering feedback and making the…	Zaki Tahmid	[illegible]	[illegible]	1d	0%
Finalize testing features		[illegible]	[illegible]	3d	0%
Deployment		[illegible]	[illegible]	3d	0%
Production Deployment		[illegible]	[illegible]	2d	0%
Release Version		[illegible]	[illegible]	3d	0%
Project Support		[illegible]	[illegible]	[illegible]	0%
Progress Reporting		[illegible]	[illegible]	10d	0%
Future Planning	+2	[illegible]	[illegible]	[illegible]	0%

Figure 4.1: *Gantt Chart with Resource Allocation and Activity wise Time Distribution*

Chapter 5: Body of the Project

Description of the work

Genesis-D Admissions Module is a patient registering web application for hospitals to be able to admit their patients in a much more efficient way than the current methods. The goal of the project is for doctors and hospital staff to be able to interact with an application naturally and not feel burdened to be using a system that they are not familiar with and to be able to retain information in a useful manner for the hospital. The team at Bioforge Health Systems Limited envisions using simple yet effective designs and use cases that may help the hospital processes to be faster and thus be able to treat lives sooner.

System analysis

Six Elements Analysis:

Process	**System Roles**					
	Human	**Non-Computing Hardware**	**Computing Hardware**	**Software**	**Database**	**Comm. & Network**
Log in/ Sign up	User	N/A	Desktop / Smartphone	Web Browser	MongoDB	WAN/LAN
View Dashboard	User	N/A	Desktop / Smartphone	Web Browser	MongoDB	WAN/LAN
Enter Admission Details	User	N/A	Desktop / Smartphone	Web Browser	MongoDB	WAN/LAN
View Admissions Summary	User	N/A	Desktop / Smartphone	Web Browser	MongoDB	WAN/LAN
Print Admissions Summary	User	N/A	Desktop / Smartphone and Printer	Web Browser	MongoDB	WAN/LAN
Generate	User	N/A	Desktop /	Web	MongoDB	WAN/LAN

Patient ID card			Smartphone	Browser		
Print Patient ID card	User	N/A	Desktop / Smartphone and Card printer	Web Browser	MongoDB	WAN/LAN

Table 5.1: *Six Element Analysis for Genesis-D Admissions Module*

System Design

Rich Picture

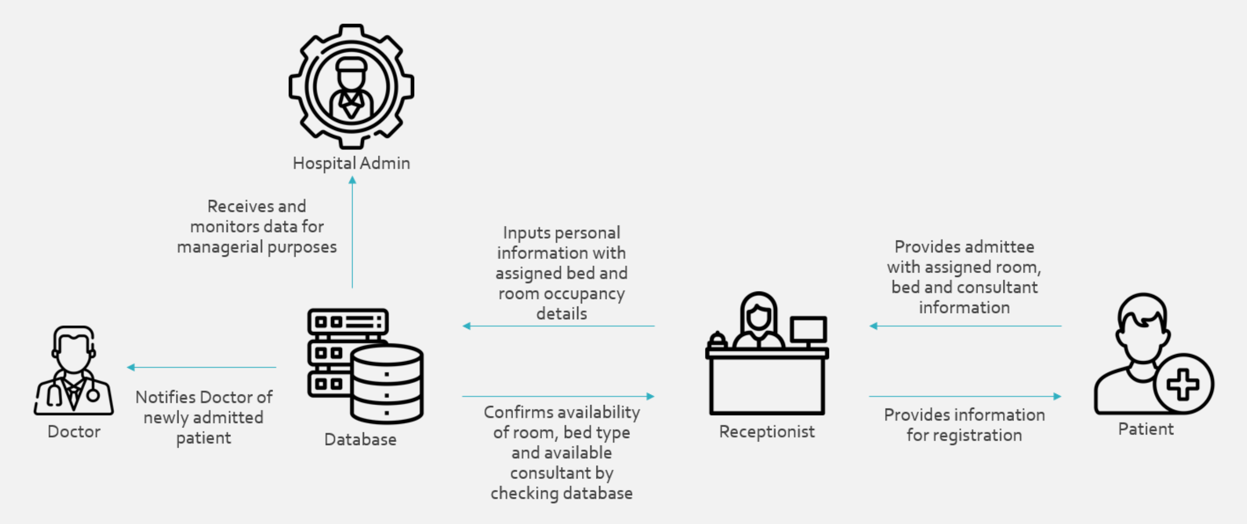

Hospital Admin
Receives and monitors data for managerial purposes
Inputs personal information with assigned bed and room occupancy details
Provides admittee with assigned room, bed and consultant information
Doctor
Notifies Doctor of newly admitted patient
Database
Confirms availability of room, bed type and available consultant by checking database
Receptionist
Provides information for registration
Patient

Figure 5.2: *Rich picture for Genesis-D Admissions Module*

Use Case Diagram

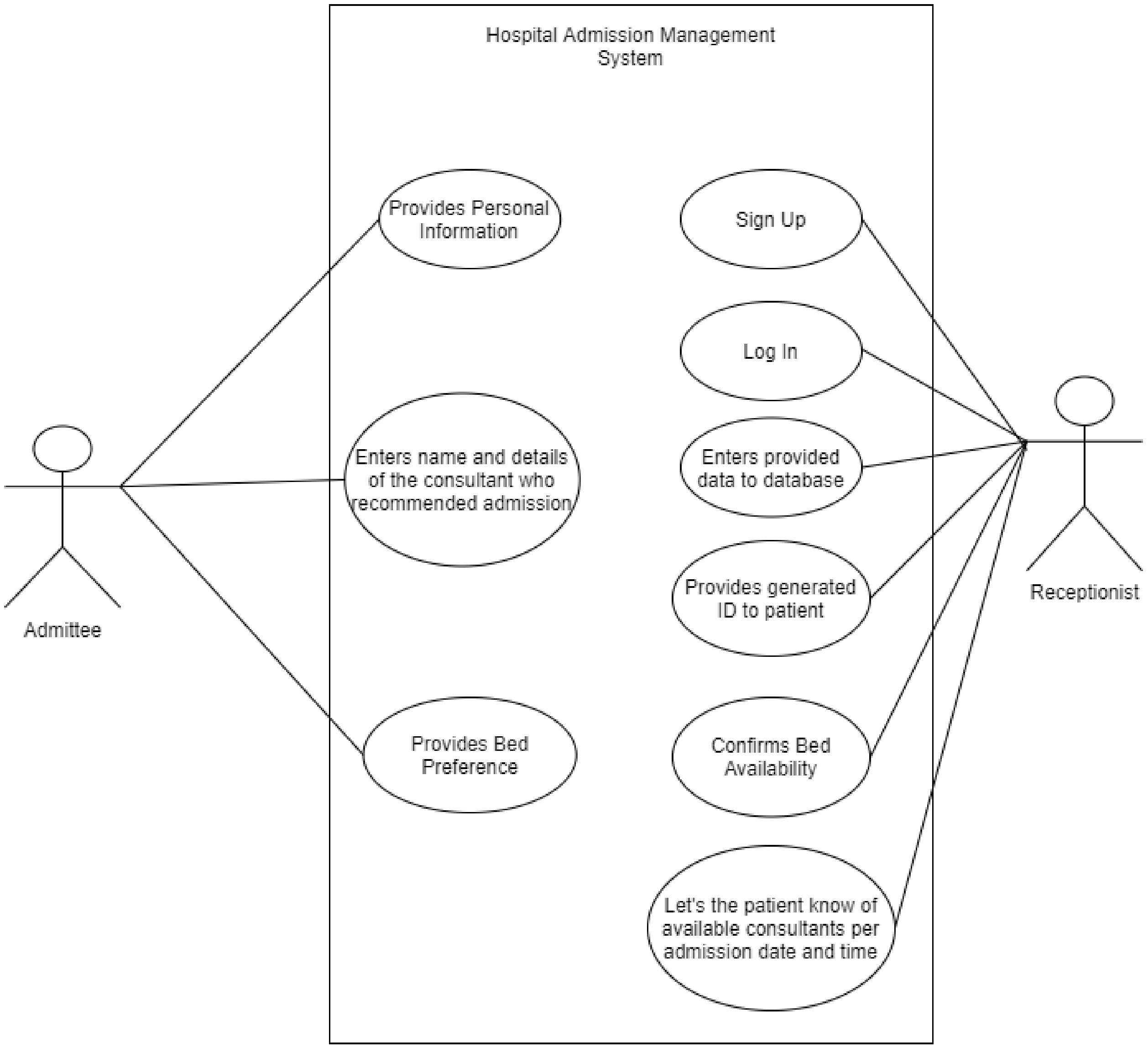

Hospital Admission Management System
Provides Personal Information
Sign Up
Log In
Enters name and details of the consultant who recommended admission
Enters provided data to database
Provides generated ID to patient
Provides Bed Preference
Confirms Bed Availability
Let's the patient know of available consultants per admission date and time
Admittee
Receptionist

Figure 5.3: *Use Case Diagrams for Genesis-D Admissions Module*

Process Flow Diagram

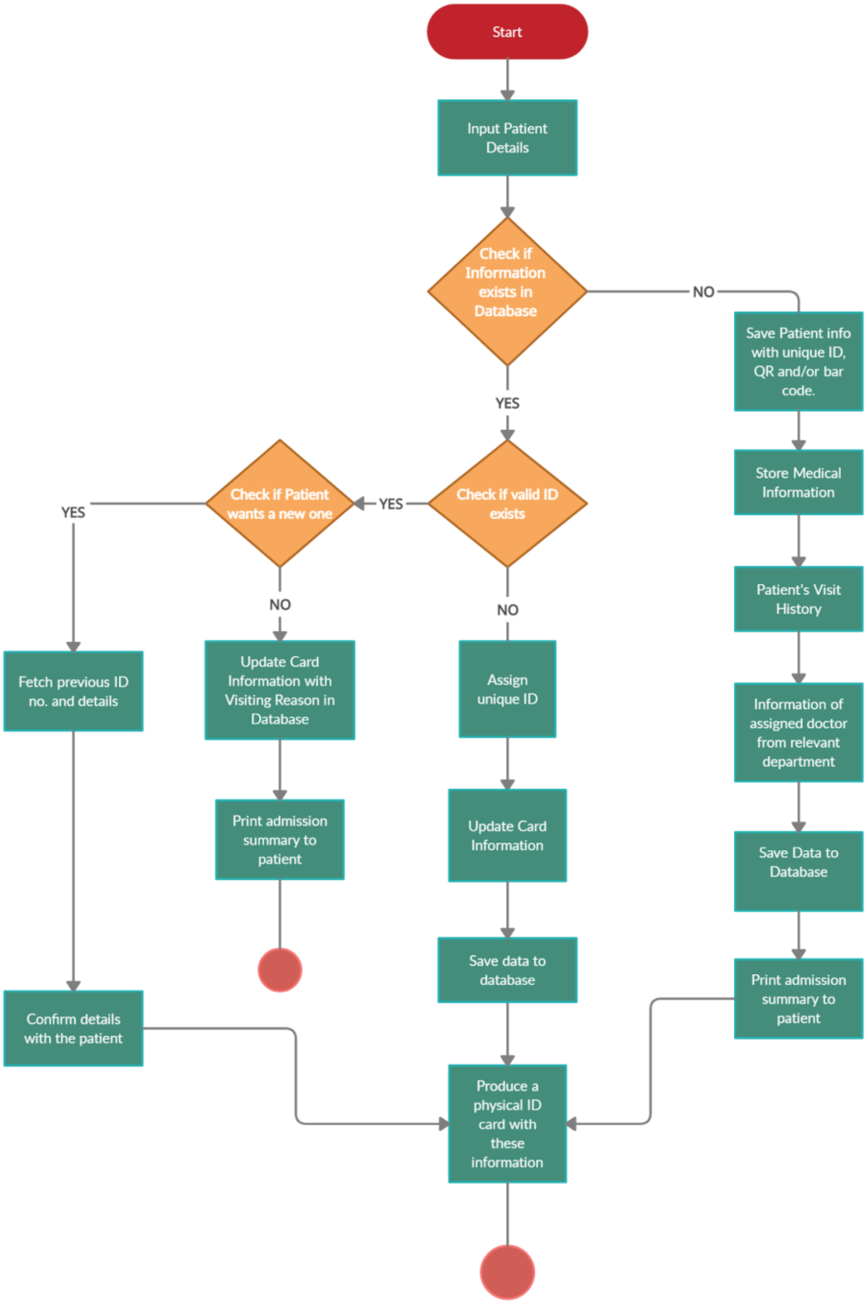
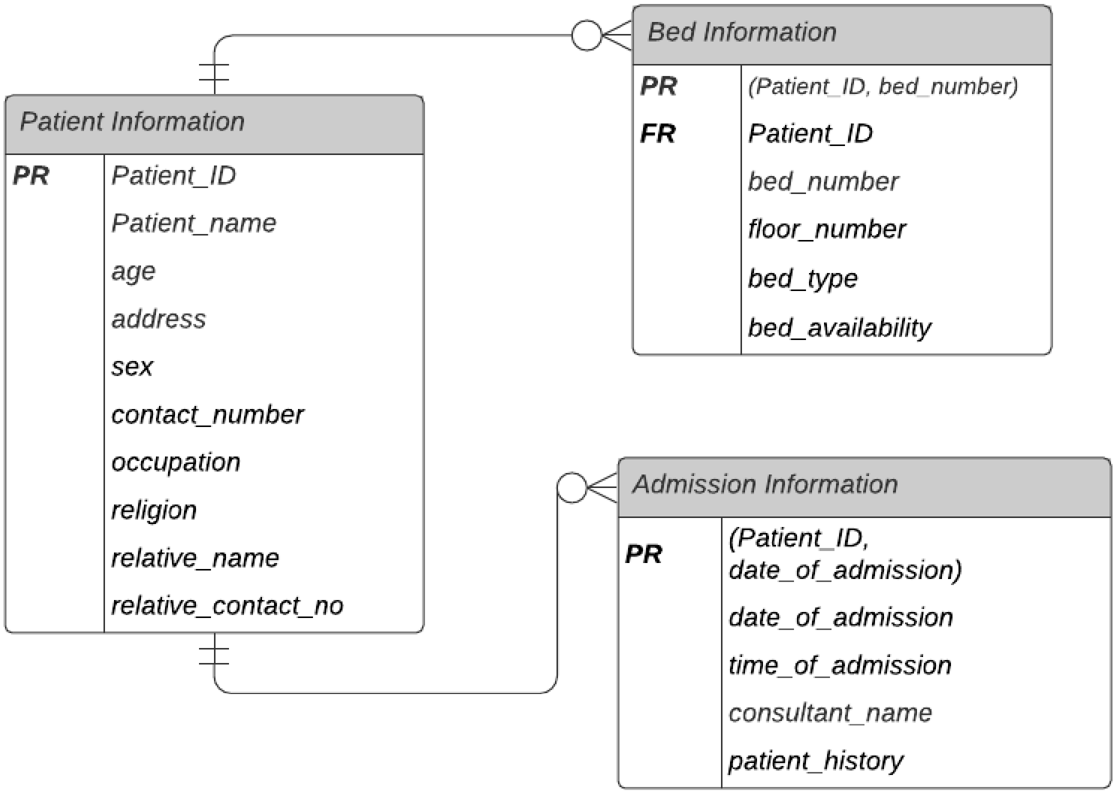

Start
Input Patient Details
Check if Information exists in Database
NO
Save Patient info with unique ID, QR and/or bar code.
Store Medical Information
Patient's Visit History
Information of assigned doctor from relevant department
Save Data to Database
Print admission summary to patient
YES
Check if valid ID exists
YES
Check if Patient wants a new one
YES
NO
NO
Fetch previous ID no. and details
Update Card Information with Visiting Reason in Database
Assign unique ID
Print admission summary to patient
Update Card Information
Save data to database
Confirm details with the patient
Produce a physical ID card with these information

__Figure 5.4:__ Process flow Diagram for Genesis-D Admissions Module

Entity Relationship Diagram (ERD)



Bed Information	
PR	*(Patient_ID, bed_number)*
FR	*Patient_ID*
	bed_number
	floor_number
	bed_type
	bed_availability

Patient Information	
PR	*Patient_ID*
	Patient_name
	age
	address
	sex
	contact_number
	occupation
	religion
	relative_name
	relative_contact_no

Admission Information	
PR	*(Patient_ID,* *date_of_admission)*
	date_of_admission
	time_of_admission
	consultant_name
	patient_history

Figure 5.5: *Entity Relationship Diagram for Genesis-D Admissions Module*

Functional Requirements

Function: User must be able to log in		
Input: Username and password	**Process:** Call an API	**Output:** User will be logged in
Precondition: User must be connected to the internet		
Postcondition: User will be redirected to dashboard page		

Table 5.2: *Functional Requirement 1: Log in*

Function: Enter Admissions Details		
Input: Patient Details, Bed Details, Relative information, Patient History	**Process:** Call an API to server to save the information to the database	**Output:** Admission details will be created and added to the database
Precondition: User must be connected to the internet		

Table 5.3: *Functional Requirement 2: Enter Admissions Details*

Function: Admissions Summary must fetch the inputted values		
Input: N/A	**Process:** Call an API to server to fetch the information from the database	**Output:** Admission details will be shown on the Admissions Summary Page
Precondition: User must be connected to the internet		

Table 5.4: *Functional Requirement 3: Admissions Summary must fetch the inputted values.*

Function: Admissions Summary to be printed in PDF		
Input: N/A	**Process:** Call a PDF generation function to download the information in PDF	**Output:** Admission details will be printed onto the local device
Precondition: User must be connected to the internet		

Table 5.5: *Functional Requirement 4: Admissions Summary to be printed in PDF.*

Function: Generate a patient ID card		
Input: N/A	**Process:** Call an API to server to fetch the information from the database	**Output:** Relevant data to be shown on the ID card view of the ID card generation page
Precondition: User must be connected to the internet		

__Table 5.6:__ Functional Requirement 5: Generate a patient ID card.

Function: Print patient ID card		
Input: N/A	**Process:** Call a jpeg/png generation function to download the information in jpeg/png	**Output:** Image of the ID card will be printed onto the local device
Precondition: User must be connected to the internet		

__Table 5.7:__ Functional Requirement 6: Print patient ID card

Function: Must be responsive across all types of screen sizes		
Input: N/A	**Process:** Application must be developed in a common development environment	**Output:** Application can be viewed by using any sort of devices
Precondition: User must have a device with an internet Connection		
Postcondition: Everyone can use this application from any devices with different screen sizes		

__Table 5.8:__ Functional Requirement 7: Responsive

Non-Functional Requirements

Non-functional requirements are briefly described of the project are listed and described below:

- **Performance:** It represents the performance of the system which is required to exhibit and to meet the needs of users. Performance describes the acceptable throughput rate and acceptable response time. This application should provide a smooth experience for the user and should have no input lag if the device has a certain minimum hardware specification.

- **Information:** represents the information that is pertinent to the users in terms of content, timeliness, accuracy, and format. Information is about the necessary inputs and outputs and how it will be managed, types of the required data to be stored, how currently the information will be saved into the system, how the interfaces of external systems will work, etc.

- **Security & Control:** Security and administrations are always a concern for any system. All information on the server side and client side is secured. Only the application administrators and developers have access to core code of the application to be able to directly manipulate any sort of information. In this project, node.js and express.js have been used for backend technology, which have various layers of security, where security requirements for this system have been taken care of. Control requirements represent the environment in which the system must operate, as well as the type and degree of security that must be provided. Access to the system or information must be controlled with the privacy requirements.

- **Efficiency:** This represents the system's ability to produce outputs with minimal waste. We have tried to eliminate duplicate steps in the processes and to use the resources in an efficient way. Keeping our code non repetitive by using reusable code and components is how we achieved efficiency.

- **Service:** represent needs to make the system reliable, flexible, and expandable. It is deals with:

    - Who will use the system and where they are located?
    - How many types of users will be in this system?
    - The appropriate human factors.
    - What training materials will be included in the system?
    - Reliability/availability requirements
    - How the system will be distributed
    - What types documentation is required?

- **Extensibility and Maintainability Requirements:** There is one standard user interface designed for the look and feel of the application. The application can be expanded to accommodate many further modules without making any changes to any existing modules. The application is created in such a way that the developers can easily maintain both the server and client sides.

Product Features

Input

The input of the system has been defined on Chapter 5: Body of the Project; ***Functional Requirements*** *and we can see the view of the input on Chapter 6: Result Analysis.*

Output

The output of the system has been defined on Chapter 5: Body of the Project; ***Functional Requirements*** *and we can see the view of the input on Chapter 6: Result Analysis.*

Architecture of the System:

Software architecture is what characterizes and structures a solution that meets specialized and operational necessities. Software architecture upgrades credits including a progression of choices, for example, security, execution, and sensibility. It depicts the association and cooperation of software segments. There are many types of architecture that are used among them. The **client server architecture** is one of them which was used for the "Genesis-D Admissions Module" Web application. To make the UI code easier to maintain and test **MVC (Model View Controller)** pattern was used.

Client Server Architecture

Client–server model is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. Often clients and servers communicate over a computer network on separate hardware, but both client and server may reside in the same system. A server host runs one or more server programs, which share their resources with clients. A client does not share any of its resources, but it requests content or service from a server. Clients, therefore, initiate communication sessions with servers, which await incoming requests. Examples of computer applications that use the client-server model are email, network printing, and the World Wide Web. [12]

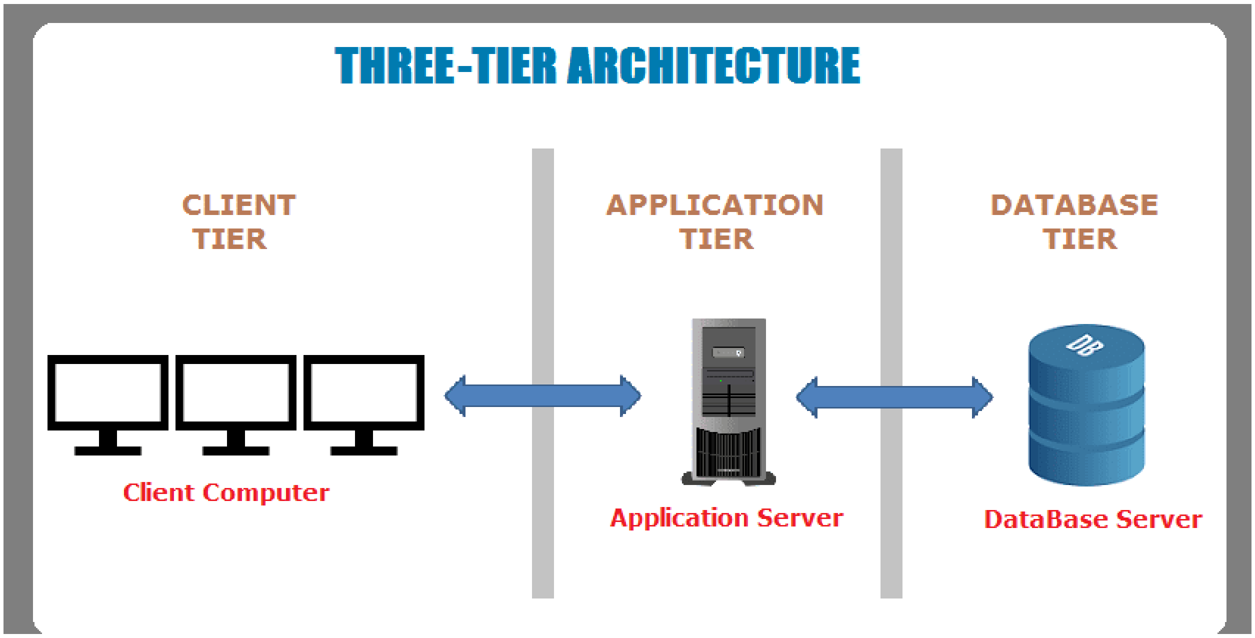
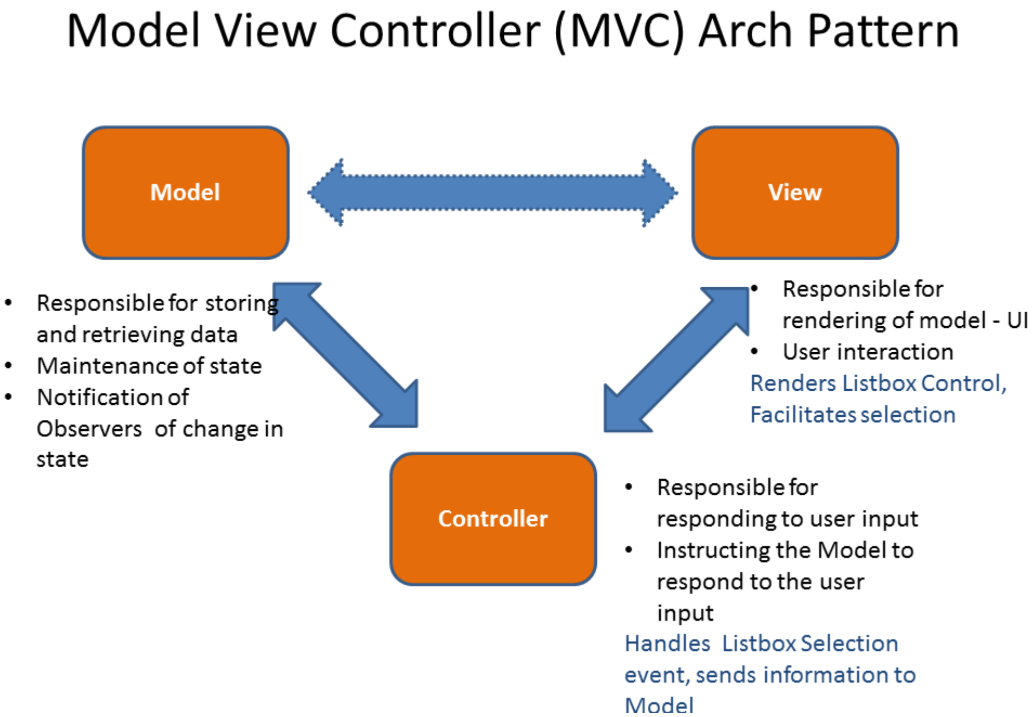


Figure 5.6: 3-tier Client Server Architecture

In “Genesis-D Admissions Module”, each client computer accessing their browsers (Google Chrome, Mozilla Firefox, Safari, etc.) is a client that will have access to the application via an application server. Upon interacting with the application to perform CRUD activities, the server accesses the Database server (Atlas MongoDB) and updates the data accordingly.

MVC (Model View Controller) pattern

MVC is a design pattern for structuring user interface code. It is a pattern followed in the application code of the project. When the MVC pattern is used a larger portion of the UI code can be unit tested. MVC architecture is triangular: the view sends updates to the controller, the controller updates the model, and the view gets updated directly from the model.



- Responsible for storing and retrieving data
- Maintenance of state
- Notification of Observers of change in state

- Responsible for rendering of model - UI
- User interaction

Renders Listbox Control, Facilitates selection

- Responsible for responding to user input
- Instructing the Model to respond to the user input

Handles Listbox Selection event, sends information to Model

Figure 5.7: *Model - View - Controller (MVC) Pattern*

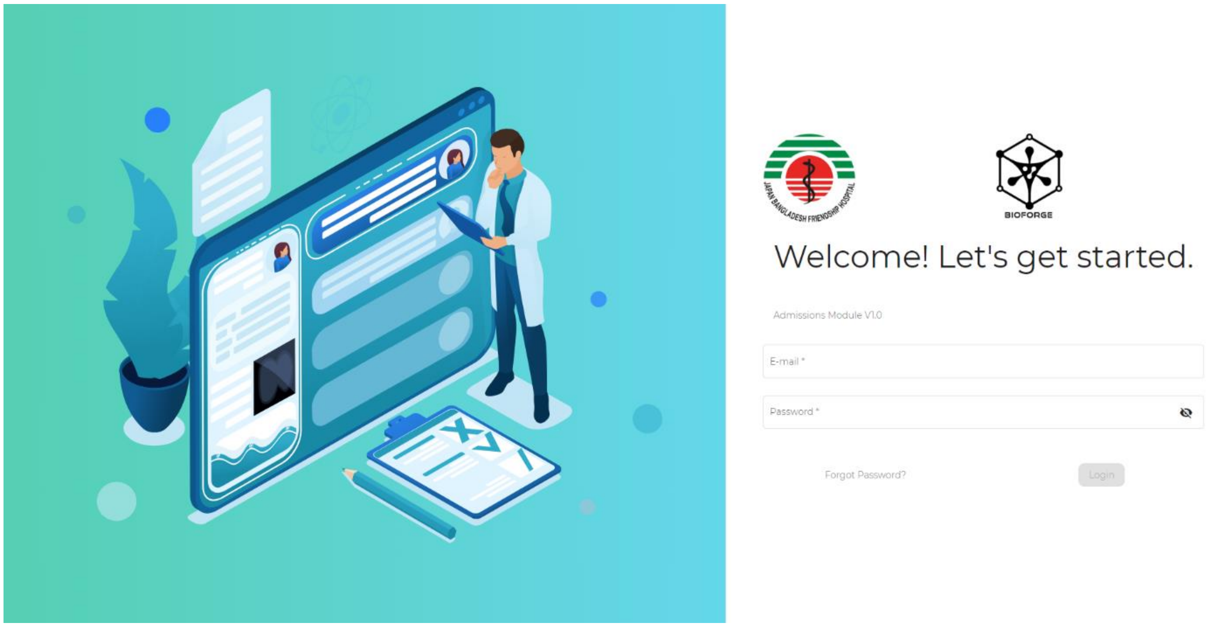
In “Genesis-D Admissions Module”, the MVC architectural pattern was implemented. Angular was used in the View (Client side), the server contains routes and controllers, and the mongoose models contain the business logic in the model part.

Chapter 6: Result and Analysis

As previously mentioned in Chapter 3, the application Genesis-D Admissions Module, is a full stack application developed using a full JavaScript stack called the MEAN stack, consisting of: MongoDB, Express, Angular and Node.js. The stack is designed to make the development process smoother and easier. The four technologies provide an end-to-end framework for developers to be able to develop web applications.

This chapter contains screenshots of the application after development.

- Login Page: This is the landing page for users to log in from



Welcome! Let's get started.

Admissions Module V1.0

E-mail *

Password *

Figure 6.1: *Screen View of Login Page*

- Dashboard: This is where users will be able to choose with which modules, they are going to be using

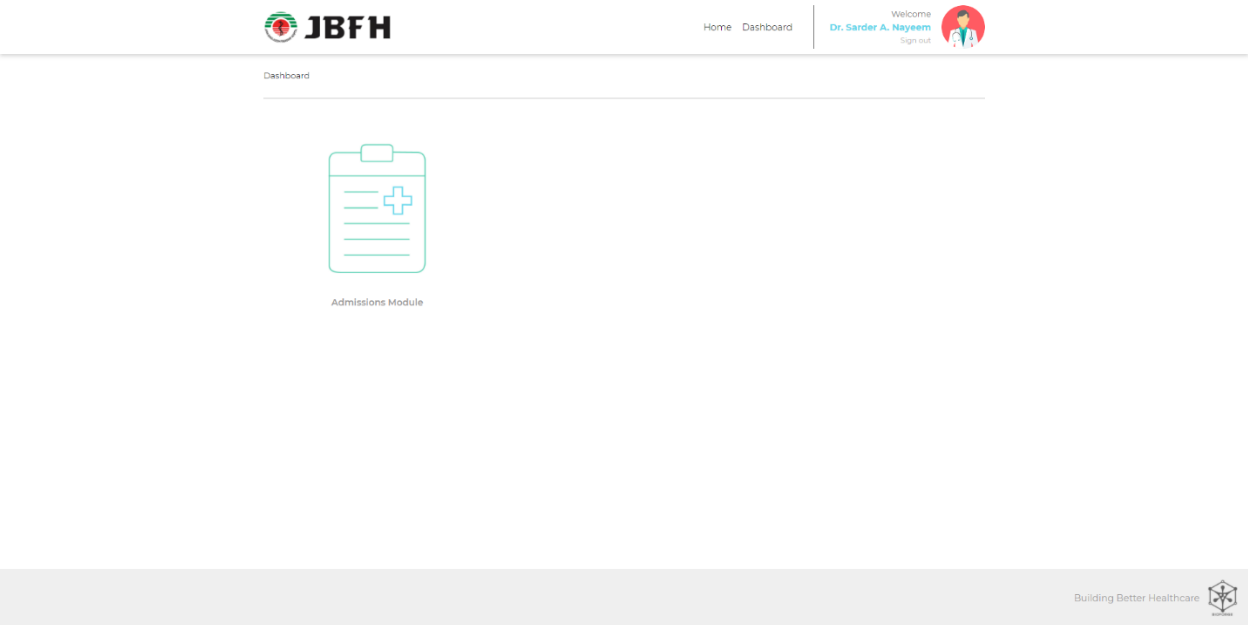
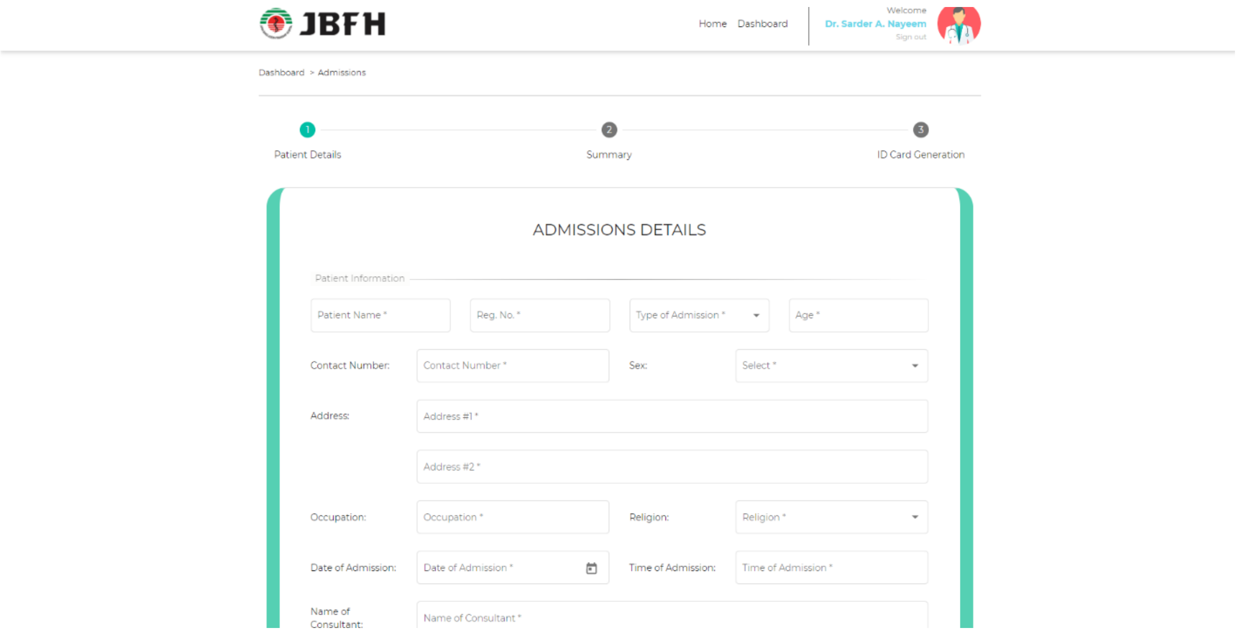
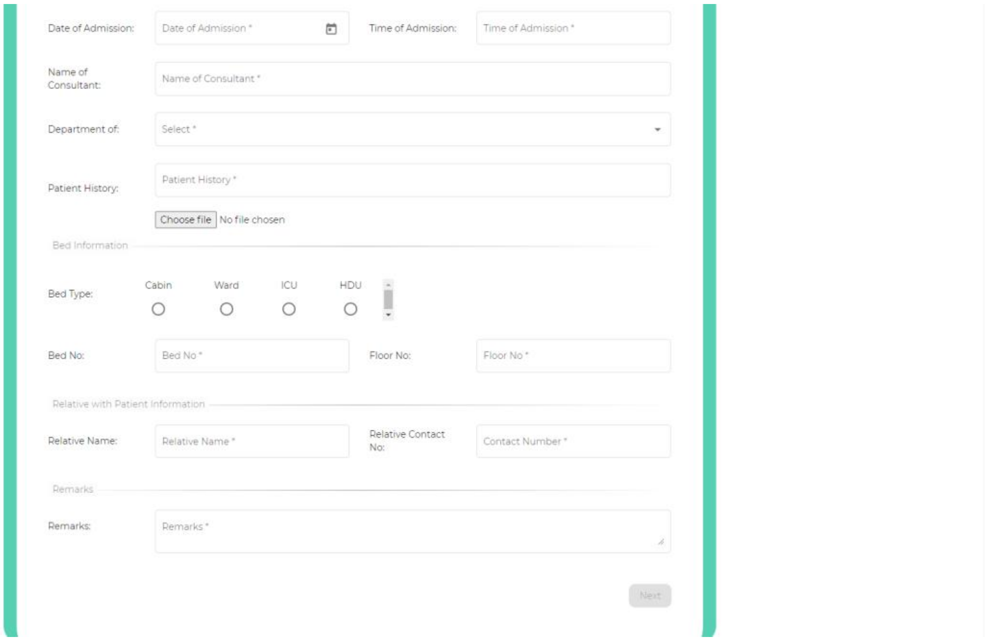


Figure 6.2: *Screen view of Dashboard*

- Admissions Details: This is the first page of the admissions details.





Date of Admission: Date of Admission *
Time of Admission: Time of Admission *
Name of Consultant: Name of Consultant *
Department of: Select *
Patient History: Patient History *
Choose file No file chosen

Bed Information

Bed Type: Cabin ○ Ward ○ ICU ○ HDU ○
Bed No: Bed No *
Floor No: Floor No *

Relative with Patient Information

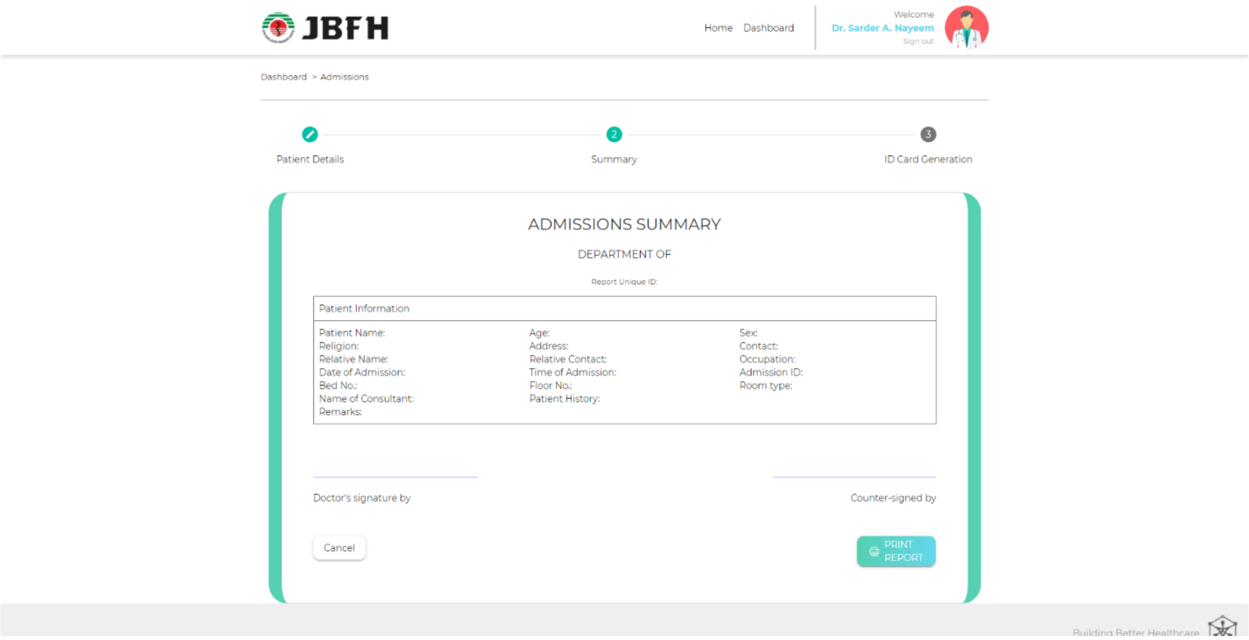
Relative Name: Relative Name *
Relative Contact No: Contact Number *

Remarks

Remarks: Remarks *

Figure 6.3 & 6.4: *Screen View of Admissions Details*

- Admissions Summary: This is where all the inputted values will be shown as an output and the user will be able to download it as a PDF.



ADMISSIONS SUMMARY

DEPARTMENT OF

Report Unique ID:

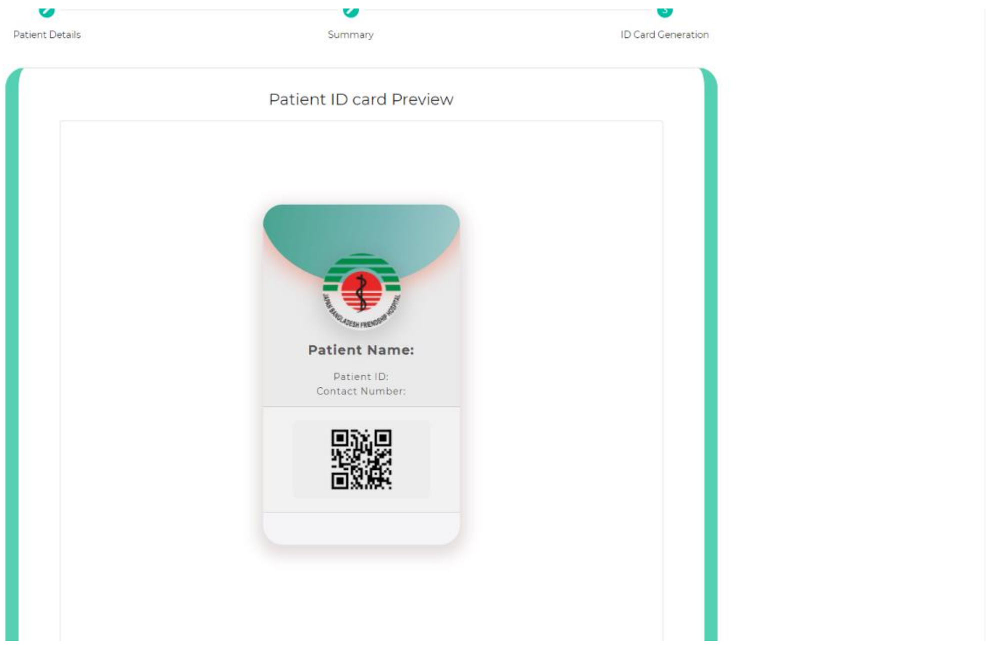
Patient Information		
Patient Name: Religion: Relative Name: Date of Admission: Bed No.: Name of Consultant: Remarks:	Age: Address: Relative Contact: Time of Admission: Floor No.: Patient History:	Sex: Contact: Occupation: Admission ID: Room type:

Doctor's signature by

Counter-signed by

Figure 6.5: *Screen View of Admissions Summary*

- ID card generation: This is where the details get printed into a card view to be printed later and given to the patient.



Patient ID card Preview

Patient Name:

Patient ID:
Contact Number:

Figure 6.6: *Screen View of ID card generation*

Chapter 7: Project/work as Engineering problem analysis

Sustainability of the product/work

Sustainability of the product refers to its ability to be maintained and updated. In the modern world, every application being released needs to be maintained and continuously updated for its user base.

A product can be sustainable in three main categories:

- **Community Sustainability:** This type of support comes in many forms such as downloading and installing the application, using the application, subscribing to paid services, feedback and referring to other people etc. After the deployment and official release of the application Genesis-D Admissions Module, it is believed that it will have a strong user base since the target audience for the application is specifically narrow and concise. We can expect the users to refer to other hospitals regarding the application and thus growing the user base. With a growing user base, it will also grow a community and hence it can be said that it is Sustainable in terms of Community.

- **Financial Sustainability:** This refers to how the application's running cost will be maintained after it has been released and whether it will generate enough revenue as acceptable profit. An application's running cost includes - server cost, database storage cost, third party API cost, etc. The initial release of Genesis-D Admissions Module will have a small development fee to use but as the user base grows there are plans to introduce new premium services which will eventually be used to generate revenue.

- **Organizational Sustainability:** It relates to how the organization will continue to operate after the release of the application. After the release of an application, usually the organization maintains the application via its current team, an extended team or by a fresh new team. Also, organizations update their project by adding newer features to it and organization may pivot to other projects, expand the teams, create new teams, etc. Genesis-D Admissions Module has many more features planned for the future to be worked on and released. Since the application has further plans, the project will be maintained and updated after its release as well and release premium services to it. In conclusion, it can be said that the project is Organizationally Sustainable.

Social and Environmental effects and analysis

Social Effect:

Genesis-D Admissions Module aims to get more hospitals interested in an efficient and data-oriented admissions process which develops an ease to use virtual applications where they can speed up the paperwork process and focus more on the treatment side of things.

Environmental effect:

The hospitals here in Bangladesh still have a long way to go in regard to getting with the time and using technology for their advantage. As future iterations of the module start rolling out the company hopes to build an eco-system where all applications merge to create a one system which the hospital staff and doctors will be using without any hesitation thus furthering the growth inside the medical industry here in Bangladesh.

Addressing Ethics and ethical issues:

Ethics and Ethical issues

In the world of smartphones with so much data collection, hacking, cybercrime, etc. there are some unspoken rules and ethics guidelines that need to be followed when working on creating and releasing an application. The developers of Genesis-D Admissions Module believe that the application does not breach any code of conduct of application release and development since they all have been taken into serious concern. Some of them are:

- Collecting only relevant User data: The app does collect user data, but those are strictly and only relevant for the app. The only data that is being collected are the number of patients being admitted and hospital relevant queries.
- Not Sharing or Selling any User data: Even though the data collected may not be of any privacy concern for most users, the app does not let any service, any application or any third party have access to the data collected.
- Data Storage Security: Only the lead developer and the owner of Genesis-D Admissions Module has access to the server and the database. Since they are hosted in the cloud and can only be accessed via lead developer's and the owner's login credentials; the data stored can be deemed as safe and secure.
- Proper use of third-party services and API: Genesis-D Admissions Module does not violate any rules of the third-party services or the APIs that have been used in its development.
- No Discrimination or Favoritism: Genesis-D Admissions Module does not discriminate of any kind based on race, sexuality, gender, religious beliefs, color, language, political or other opinion, national or social origin, property, birth, or other status.
- Clear Promotion: Genesis-D Admissions Module only intends to promote the company that created it, itself, and people's health. Other than what has been mentioned, Genesis-D Admissions Module has no intention of promoting anything or anybody else.

Chapter 8: Future Work and Conclusion

Challenges Faced:

In the midst of my internship here with Bioforge Health Systems Limited there were a few obstacles which I had to face and overcome:

- **Worldwide Pandemic of Covid19:** Even though almost a whole year had gone by for the pandemic, the norms were still very disruptive during daily procedures. Due to the lockdown almost, every member had been working from home and therefore many aspects lacked communication in the beginning however with routine changes and different protocols being set like regular meetings, pair programming and issue reporting, we got into a new normal to work in with which we picked up our productivity gradually.

- **Adapting to the Team:** It usually becomes very difficult for me to join into a new team and be able to gel with them right away however the environment here and the people made that transition very smooth and eventually became a dynamic group of people.

- **Debugging:** As a programmer who has always been solving problems from scratch, having other applications being handed over and expected to solve existing issues was new to me. It was a learning experience which may have taken me quite some time however my mentors and peers have helped and guided me through being able to learn quickly and surely.

Future Work

The project Genesis-D Admissions Module will have a few more iterations in the future which may include features like:

- Graphical representations of patients being admitted.
- Integration with other Modules

Conclusion

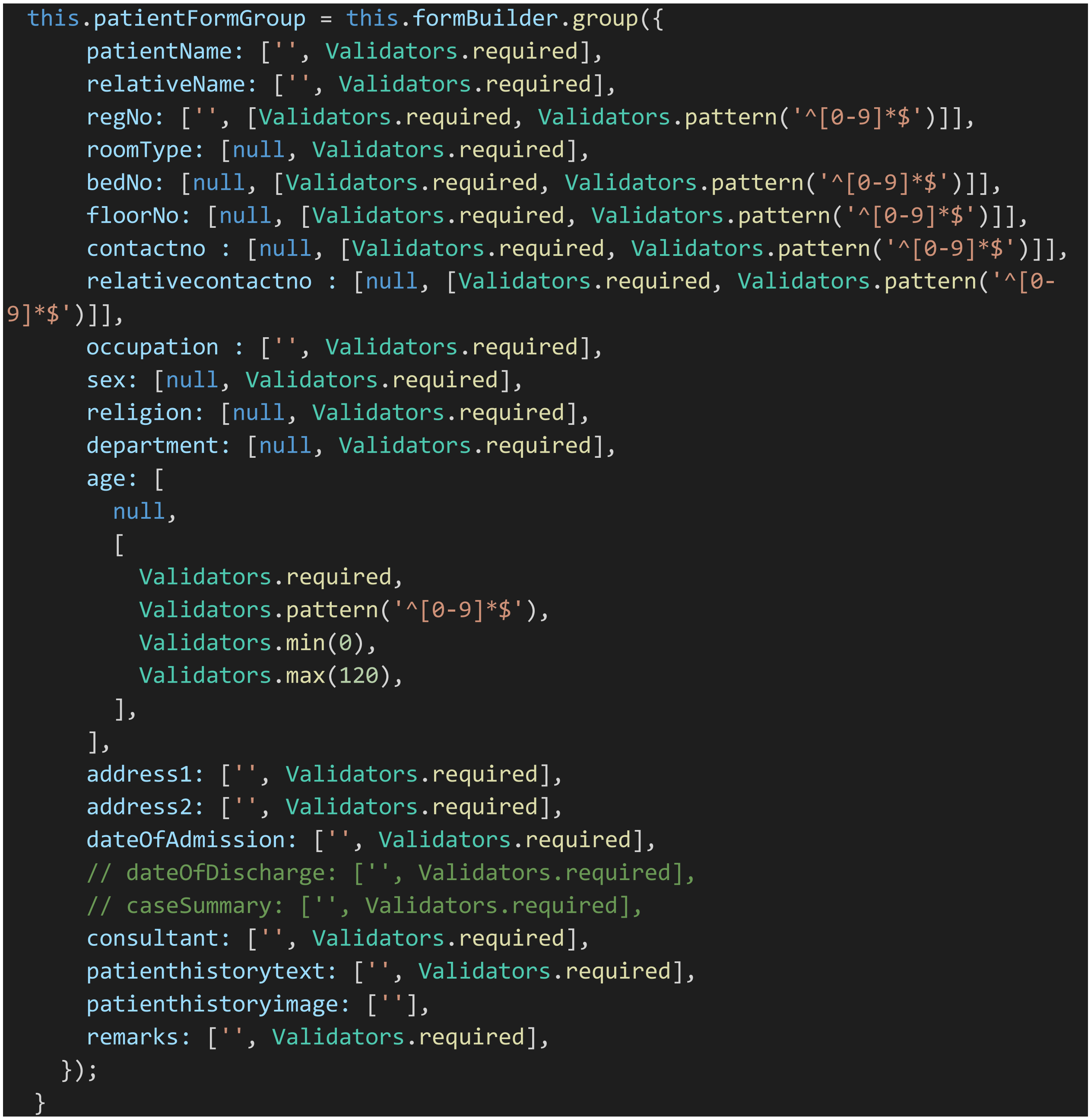
It was a wonderful experience working with the Bioforge family as an intern. During the internship period I have learnt and applied a great deal of knowledge from my university courses and experience here. I was introduced to new cutting-edge technologies like Angular, Node.js and Express.js. I have learned a lot about developing different kinds of applications also about various development styles. I was pushed to adapt to changes rapidly and come up with logical solutions. During my project, I cooperated with my mentors and seniors to solve the challenges faced. Despite their workload, my supervisors were always there to answer any queries and help me settle nicely. This internship opportunity has paved the way to investigate the development environment and marketplace. I would like to appreciate once again everyone who has made my life as an intern such a great experience.

References

1. Lærum, Hallvard, et al. “Use of and Attitudes to a Hospital Information System by Medical Secretaries, Nurses and Physicians Deprived of the Paper-Based Medical Record: A Case Report.” BMC Medical Informatics and Decision Making, vol. 4, no. 1, Oct. 2004, p. 18. Springer Link, doi:10.1186/1472-6947-4-18.
2. “Best Hospital Management System in Bangladesh - Smart Software Ltd.” Best Hospital Management System in Bangladesh - Smart Software Ltd., https://www.smartsoftware.com.bd/hospital-management. Accessed 20 Jan. 2021.
3. Millennium Heart & General Hospital Ltd. http://millenniumhospitalbd.com/. Accessed 20 Jan. 2021.
4. “Waterfall Model: The Ultimate Guide to Waterfall Methodology.” ProjectManager.Com, https://www.projectmanager.com/waterfall-methodology. Accessed 20 Jan. 2021.
5. Angular. https://angular.io/. Accessed 20 Jan. 2021.
6. Node.js. “Node.Js.” Node.Js, https://nodejs.org/en/. Accessed 20 Jan. 2021.
7. Express - Node.Js Web Application Framework. https://expressjs.com/. Accessed 20 Jan. 2021.
8. “The Most Popular Database for Modern Apps.” MongoDB, https://www.mongodb.com/1. Accessed 20 Jan. 2021.
9. Git. https://git-scm.com/. Accessed 20 Jan. 2021.
10. Trello. https://trello.com. Accessed 20 Jan. 2021.
11. “GitHub: Where the World Builds Software.” GitHub, https://github.com/. Accessed 20 Jan. 2021.
12. “Client–Server Model.” Wikipedia, 19 Jan. 2021. Wikipedia, https://en.wikipedia.org/w/index.php?title=Client%E2%80%93server_model&oldid=1001393187 .

Appendix - A (Backend Code Snippets)

Validation code for each field



```
  this.patientFormGroup = this.formBuilder.group({
      patientName: ['', Validators.required],
      relativeName: ['', Validators.required],
      regNo: ['', [Validators.required, Validators.pattern('^[0-9]*$')]],
      roomType: [null, Validators.required],
      bedNo: [null, [Validators.required, Validators.pattern('^[0-9]*$')]],
      floorNo: [null, [Validators.required, Validators.pattern('^[0-9]*$')]],
      contactno : [null, [Validators.required, Validators.pattern('^[0-9]*$')]],
      relativecontactno : [null, [Validators.required, Validators.pattern('^[0-
9]*$')]],
      occupation : ['', Validators.required],
      sex: [null, Validators.required],
      religion: [null, Validators.required],
      department: [null, Validators.required],
      age: [
        null,
        [
          Validators.required,
          Validators.pattern('^[0-9]*$'),
          Validators.min(0),
          Validators.max(120),
        ],
      ],
      address1: ['', Validators.required],
      address2: ['', Validators.required],
      dateOfAdmission: ['', Validators.required],
      // dateOfDischarge: ['', Validators.required],
      // caseSummary: ['', Validators.required],
      consultant: ['', Validators.required],
      patienthistorytext: ['', Validators.required],
      patienthistoryimage: [''],
      remarks: ['', Validators.required],
    });
  }
```

Downloading Admissions Summary

```
import { ReportPDFService } from '../../common/report-pdf.service';

downloadReport(): void {

      this.reportPDFService.generateReportPDF(this.report, this.comorbidities);
}
```