**Health Care Management system bd**

**by**

**Tanjin Ahamed**

**ID: 171-15-9185**

**AND**

**Md. Nuruzzaman Nirob**

**ID: 171-15-8602**

**AND**

**Golam Rafi Badhon**

**ID: 171-15-8715**

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering.

Supervised By

**Fahad Faisal**

Assistant Professor

Department of CSE

Daffodil International University

Co-Supervised By

**Shah Md. Tanvir Siddiquee**

Assistant Professor

Department of CSE

Daffodil International University

****

**DAFFODIL INTERNATIONAL UNIVERSITY**

**Dhaka, Bangladesh**

**December 2020**

**APPROVAL**

This Project titled “**Population registration system bd**”, submitted by \*Tanjin Ahamed\* and \*Md. Nuruzzaman Nirob\* and \*Golam Rafi Badhon\* to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on \*date\*.

**Board of Examiners**

**(Name)** [**Font-12, Bold**] **Chairman**

**Designation**

Department of CSE [**Font-12**]

Faculty of Science & Information Technology

Daffodil International University

**(Name) Internal Examiner**

**Designation**

Department of CSE

Faculty of Science & Information Technology

Daffodil International University

**(Name) External Examiner**

**Designation**

Department of -------

Jahangirnagar University

**Declaration**

We hereby declare that, this project has been done by us under the supervision of **Fahad Faisal, Assistant Professor, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

**Supervised by:**

**Fahad Faisal**

Assistant Professor

Department of CSE

Daffodil International University

**Co-Supervised by:**

**Shah Md. Tanvir Siddiquee**

Assistant Professor

Department of CSE

Daffodil International University

**Submitted by:**

**(Tanjin Ahamed)**

ID: 171-15-9185

Department of CSE

Daffodil International University

**(Md. Nuruzzaman Nirob)**

ID: 171-15-8602

Department of CSE

Daffodil International University

**(Golam Rafi Badhon)**

ID: 171-15-8

Department of CSE

Daffodil International University

**ACKNOWLEDGEMENT**

First, we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project/internship successfully.

We really grateful and wish our profound our indebtedness to **Fahad Faisal**, **Assistant Professor**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of “*Field name*” to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to -----, -----, andHead**,** Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

**Abstract**

This report is intended as a guide for doctors and patients when considering a medical history of patient to review his/her health history. It's a significant piece of their work process to guarantee they're giving the best consideration and treatment. A patient's clinical history may incorporate insights regarding past infections, diseases running in the family, past determinations, clinical unique, treatments, hypersensitivities, and drug. Truly, this isn't the entire picture yet with the assistance of a point-by-point clinical history, specialists can see health examples of patients over the long run initially. Numerous hospitals depend on paper-based structures for this errand. This web application will resemble in doctor's co-op with simple to utilize adaptable choices. The application is open from anyplace for all specialists or patients in private or at work areas, android or tablets, and so on. It will fundamentally improve the nature of keeping up records and other data identified with specialists or patients or charging and so on. Using web application such as this report, Patients can easily find authentic information of doctor’s schedules and locations. The main key in this web application is improving the overall medical system. Our system will bring a significant change to our health care system.

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **CONTENTS** | **PAGE** |
| Board of examiners | ii |
| Declaration | iii |
| Acknowledgements | iv |
| Abstract | v |
| **CHAPTER** |  |
| **CHAPTER 1: INTRODUCTION & OBJECTIVE** | **1-3** |
| * 1. Introduction   2. Objectives   3. Present System in use   4. Flaws in present system / Need for new system   **CHAPTER 2: PROPOSED SYSTEM & FEASIBILITY STUD**  2.1 Introduction (Proposed System)  2.2 Feasibility Study   * + 1. Technical Feasibility     2. Economic Feasibility     3. Operational Feasibility   2.3Project Category  2.4 Software Engineering Process Model  2.4.1 Requirements specification  2.4.2 Software Design  2.4.3 System Integration  2.4.4 Application Testing  2.4.5 Application Deployment  2.4.6 Implementation & Maintenance  **CHAPTER 3: FLOW CHARTS & DATABASE DESIGN**  3.1 System Flow Chart  3.2 Modules and Modular Charts  3.3 Database & Tables  3.3.1 Data Structure, File Design & Table  3.3.2 Data Flow Diagram  3.3.2.1 0 levels DFD  3.3.2.2 1 levels DFD  3.3.2.3 2 levels DFD  3.3.3 Entity Relationship Diagram  3.3.4 Use Case of the System  3.3.5 Activity Diagram  3.3.5.1 Activity Diagram for Patient  3.3.5.2 Activity Diagram for Doctor  3.3.5.3 Activity Diagram for Admin  3.3.6 UML diagram of the system  3.3.4 User Information & Module  3.3.4.1: Patients information  3.3.4.2: Doctors information  3.3.5 Data Dictionary  3.3.6 Validation Checks  **CHAPTER 4: PROJECT IMPLEMENTATION & TESTING**  4.0 Category & Layout  4.1 Project Design (GUI)  4.1 Login Form:  4.1 Login Form  4.2 Patient Registration  4.3 Patient Profile  4.4 Payment Gateway  4.5 Search Doctors  4.6 Search Patient  4.7 Add Schedule  4.8 Doctor Profile  4.9 Doctor Registration  4.4 Testing  **4.4.1. Unit Testing**  **4.4.2. System integration testing (SIT)**  **CHAPTER 5: CONCLUSION & FUTURE SCOPE**  5.1 Conclusion  5.2 Benefits  **5.3 Technology is everywhere**  5.4 Limitation of the framework  5.5 Future Plans | 2  2  2  3  4-8  4  5 |
|  |
| **APPENDIX** |
|  |  |
| **REFERENCES** | **56-57** |

|  |  |
| --- | --- |
| **LIST OF FIGURES** | |
| **FIGURES** | **PAGE NO** |
| Chapter 3 |  |
| Figure 3.1: System Flow Chart | 10 |
| Figure 3.2: Modules & Modular Flow Charts | 11 |
| Figure 3.3: 0 Level DFD | 13 |
| Figure 3.4: 1st Level DFD | 14 |
| Figure 3.5: 2nd Level DFD | 15 |
| Figure 3.6: Entity Relationship Diagram | 16 |
| Figure 3.7: Use Case of the System | 17 |
| Figure 3.8:Activity Diagram for Patient | 18 |
| Figure 3.9: Activity Diagram for Doctor | 19 |
| Figure 3.10: Activity Diagram for Admin | 20 |
| Figure 3.11: UML diagram of the system | 21 |
| Chapter 4 |  |
| Figure: 4.1 Project Design (GUI) | 28 |
| Figure: 4.1.1 Login Form | 29 |
| Figure: 4.1.2 Patient Registration | 30 |
| Figure: 4.1.3 Patient Portfolio | 31 |
| Figure: 4.1.4 Payment Gateway | 32 |
| Figure: 4.1.5 Search Doctors | 32 |
| Figure: 4.1.6 Search Patient | 33 |
| Figure: 4.1.7 Add Schedule | 33 |
| Figure: 4.1.8 Doctor Profile | 34 |
| Figure: 4.1.9 Doctor Registration | 35 |

**LIST OF TABLES**

**TABLES PAGE NO**

Table 3.1: Patients Addition 14

Table 3.2: Doctors Addition 15

Table 3.3: Report

Table 3.4: Treatment History 16

Table 3.5: Doctor Schedule 16

Table 3.6: Payment 16

Table 4.7: Testing Table 28

# CHAPTER 1

## INTRODUCTION & OBJECTIVES

1.1 Introduction

This is the time of present-day science and innovation. The fundamental reason utilizing modernized framework is to evade manual issues and furthermore documentation stockpiling issue we can't keep up significant stretch information that is the reason we utilized automated framework to defeat all issue identified with college's information putting away and different arias. This electronic application handles online understudy confirmation system.

Population is the principal heritage of every country. Only government, which possesses an information about population, may plan the future of its country in a correct, secure and effective way. A Population Register is a system that stores records of personal information of all citizens and non-citizens residing on the territory of a state that meet the requirement for registration as set out in the relevant legal framework. This task diminishes the time and cost and gives the office to recover understudy all data as indicated by prerequisite. Our project focuses mainly on a particular sector of the total system. Which is health care system. Health care system is a big factor for people as it’s one of the fundamental rights of each human being.

This project works at dealing with dynamic site.

This project will require users to register on the website and obtain a serialID to get the facilities. After getting verified by the authority both the doctors and patients will get a personal dashboard from where they can take the further steps.

The doctors will be able to set their specialization fields and add, remove or edit their availability schedules. The doctors will be able to search the patients and view their profile to check their previous treatment history and test reports. They will also be able to prescribe their patients.

The patients will be able to view the list of specialized doctors and observe the all information such as the doctor’s availability schedule, their fees etc. From there they will be able to book an appointment in the desired date and time according to doctor’s availability and pay the required payments. They will be able to view the previous treatments and generate reports if needed.

1.2 Objectives

A Fully automated system will be developed which will be better than the existing system used by government. It is web-based project that is maintain all the activity related to health care, it stores all the information of patients and doctors. It will show time to time reports of treatment activity of the patients so that the information can be used for further treatments and research purposes.

To insure every local person find him right digital way

* To provide our government an efficient way to use our information and resource.
* To help the authority to provide the right of getting proper treatments when needed.
* To provide the information pharmaceutical companies the information to improve their products and make innovations.
* To provide information to researchers.
* To provide synchronized and centralized information of all database.

**1.3 Present System in use**

The present system consists of static web pages and do not allow dynamic insertions of data. Hence there is a need to create a dynamic website.

* System can be web-based so that everyone can easily interact with system.
* System can provide optimize functionality.
* System can be customizable so that one can update it.
* System can be flexible enough so that it can incorporate different changes time to time.
* All the information related with this project can be documented.
* The most important thing is security. All the data should remain consistent and secure.

**1.4 Flaws in present system / Need for new system**

* Present system is a totally manual system which lacks security and is time consuming. This is not user friendly
* The data is recorded manually, which is error prone and often leads to confusion.
* A lot of file work had to be done for storing information like doctors’ details, patients’ details.
* There may be possibility of delay in managing whole treatment process.
* Also, certain information redundancy may occur then it will become a hurdle to manage.
* Staff member Management entitles all management of activities related to verify doctors and patients, manage scheduled and so on requires a lot of paper work.

**CHAPTER 2**

**PROPOSED SYSTEM & FEASIBILITY STUDY**

**2.1 Introduction (Proposed System)**

This venture is an endeavor to make the undertaking of authority simpler. Eeconomic development in a nation generally relies upon the guidelines of its social infrastructure. healthcare is significant zones of social foundation. Populace is the essential legacy of each nation. Just government, which has a data about populace, may design the eventual fate of its nation in a right, secure and powerful way. A Population Register is a framework that stores records of individual data, all things considered, and non-residents dwelling on the region of an express that meet the necessity for enlistment as set out in the significant legitimate structure. This assignment decreases the time and cost and gives the workplace to recuperate understudy all information as shown by essential. Our task center primarily around a specific area of the absolute framework. Which is medical services framework. Medical care framework is a major factor for individuals as it's one of the basic privileges of every person.

**2.2 Feasibility Study**

An agenda that identifies, describes, and evaluates applicant systems and selects the best structure for the job is called as Feasibility study.

Three key discussion are elaborate in the feasibility analysis:

* + 1. Technical Feasibility
    2. Economic Feasibility
    3. Operational Feasibility

**2.2.1. Technical Feasibility:**

The use of HTML, CSS, Bootstraps, JavaScript and Django makes form design easy and handy. The project can be run on any system with slightest requirements. It cut down data entry failure because of data entry validation, it can be smoothly managed by any user, and it also helps in quicker data updating. Also the project developed in GUI, it is very simple to run. Thus, the project is technically feasible.

**2.2.2. Economic Feasibility:**

Economic benefit analysis is very important in deciding whether the project is economically feasible or not. It is distant from everyone else acceptable to set aside our time and cash. It is onetime expense and does not require proper maintenance. Through cost benefit analysis it was concluded that the project is economically feasible.

**2.2.3. Behavioral Feasibility:**

Behavioral feasibility determines how much labor will go into health care. The project was also calculating to be behaviorally feasible as it is very user-friendly and comparatively use any extra efforts to educate user for its utility and functioning.

**2.3 Project Category**

This is web-based project. This project developed for health care purpose. It provides the better facility for patients and doctors to check all the information related to treatment information and research material.

It serves health care facility and primary objective of this project is raise health care service and inform everyone about the health care system and its facility.

While using this web platform users will get to know the condition of treatment that is delivered in hospital. This project developed for health system, pharmaceutical companies.

**2.4 Software Engineering Process Model**

The waterfall model shows a process, where builder is to follow these phases in order:

* + 1. Requirements specification
    2. Software Design
    3. System Integration
    4. Application Testing
    5. Application Deployment
    6. Implementation & Maintenance

**2.4.1. Requirements Specification:**

A Software Requirements Specification is a complete explanation of the nature of a system to be developed. It consists of a set of use cases that express all the communication the users will have with the software.

We research the requirement and specification provided by patients, doctors and list out all the functional requirement of website that would be implemented from our side. We also advise users some good functionality like generating report.

**2.4.2. Software design:**

Software design is a process of problem solving and planning for a software result. After the output and specifications of software are resolute, software developers will design to develop a plan for a solution.

We have branched the project into small parts and plan how we can design and implement the part as per the user assurance. First, we have planned a database pattern of project, which would help us to go in appropriate flow, we have also designed the DFD (Data flow design) to develop the website.

**2.4.3. System integration:**

System integration is the import together of the fringe subsystems into one framework and guarantee that the subsystems work altogether. In data innovation, frameworks joining is the discard connecting together specific registering frameworks and programming applications truly or practically, to go about as an accommodated entirety. We have accomplished knowledge of all interfaces that would add on our website. It includes interfaces between Modules, Database, Server, and between the other system API (Application program interface), which would work with. For a system to be successfully implemented and used, the elements like DB, files/function must be in place and functioning correctly.

**2.4.4. Application testing:**

Application testing is an examination coordinate to give partners data about the part of the item or administration under test. Programming testing likewise brings a goal, confident perspective on the product to give the business to recognize constantly the dangers of programming usage.

Test approach incorporate, however are not restricted to, the way toward executing a program or application with the plan of discovering programming bugs (mistakes or different imperfections).

We have checked the exactness, fulfillment, consistency, spelling and openness of site. These regions are the primary things decided by the client. Clients should have the most ideal involvement in our site. For program similarity we have tried the site in all programs to ensure the designs and different articles on a site would be shown same. To check these modules, test perusing should be finished. The reason for this test is to discover blemishes in the route of the pages.

**2.4.5. Application Deployment:**

Application arrangement is the entirety of the exercises that make an application framework accessible for use. The overall arrangement measure comprises of a few interrelated exercises with potential changes between them. These exercises can happen at the maker site or at the shopper site or both. Since each product framework is novel, the exact cycles or systems inside every movement can scarcely be characterized. In this manner, "deployment" should be deciphered as an overall cycle that must be redone as indicated by explicit prerequisites or qualities.

After execution and testing of entire site on nearby worker, we have sent the site on fundamental worker to prepare for dispatch. The user has provided their server details along with database details. We transfer (uploaded) files from our local server to the main server through ftp (file transfer protocol), also run the sql file in ajax file to import the database, then we configure the file for database connection to run the system on main server.

**2.4.6. Implementation & Maintenance:**

The Maintenance Phase is the longest phase of the SDLC. In this phase, the product is refreshed to:

- Fulfill the changing patients' need.

- Adapt to oblige change in the outside climate.

- Correct mistakes and oversights beforehand undetected in the testing phase.

- Enhance the proficiency of the product.

**CHAPTER 3**

**FLOW CHARTS & DATABASE DESIGN**

**3.1 System Flow Chart:**

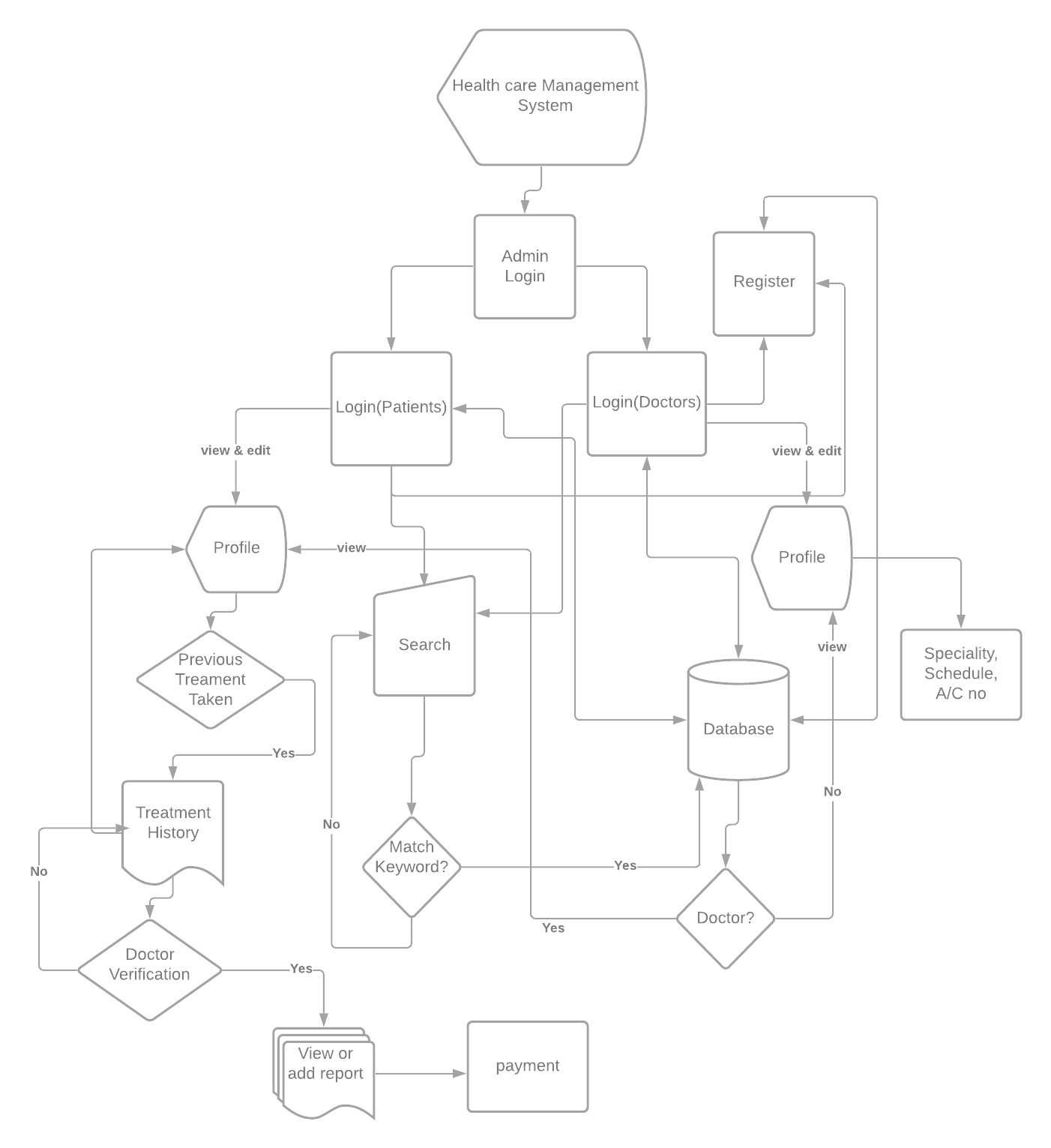
This figure 3.1 is the flow chart of the Health Care Management System by which we can comprehend the working cycle of the application.

Figure 3.1: System Flow Chart

**3.2 Modules and Modular Charts**

The following figure 3.2 shows the modules and modular charts of the admin who will operate the program.

Login

Admin Login

Patients

Doctor

Already User?

Registered user

Yes

Figure 3.2: Modules & Modular Chart

**3.3 Database & Tables**

**3.3.1 Data Structure, File Design & Table**

**Patients:**

The Patients table stores the serial ID, First Name, Last Name, NID, Email, DOB, Division, District, Thana, Union, Ward, Village, Password, Gender, Religion, Name Of Father, Occupation, Name Of Mother, Occupation2, Mobile, Residence Phone, Residence Address, Email, City Country etc.

**Doctors:**

The Doctors table stores the serial ID, First Name, Last Name, NID, Email, DOB, Division, District, Thana, Union, Ward, Village, Password, BMI, Specialists, Account Number, Gender, Religion, Name of Father, Occupation, Name of Mother, Occupation2, Mobile, Residence Phone, Residence Address, Email, City Country etc.

**Doctor Schedule and payment:**

The schedule table stored the day, time, address etc.

The payment table stored payment medium, transaction id, date etc.

**Report:**

The report table stores prescription, disease history, medicine list, report etc.

3.3.2 Data Flow Diagram

A dataflow diagrams shows the utilitarian relationship of qualities registered by a framework, including input esteems, yield esteems and inward information stores. It is a graphical portrayal indicating the progression of information esteems, contains measures, information stream, entertainer items, and information stores. Information Flow Diagram (DFD) is one of the primary instruments used to show framework parts. The parts of DFD's are the framework measures, the information utilized by measures, any outside elements that communicate with the framework and the data streams in the framework.

3.3.2.1: 0 levels DFD

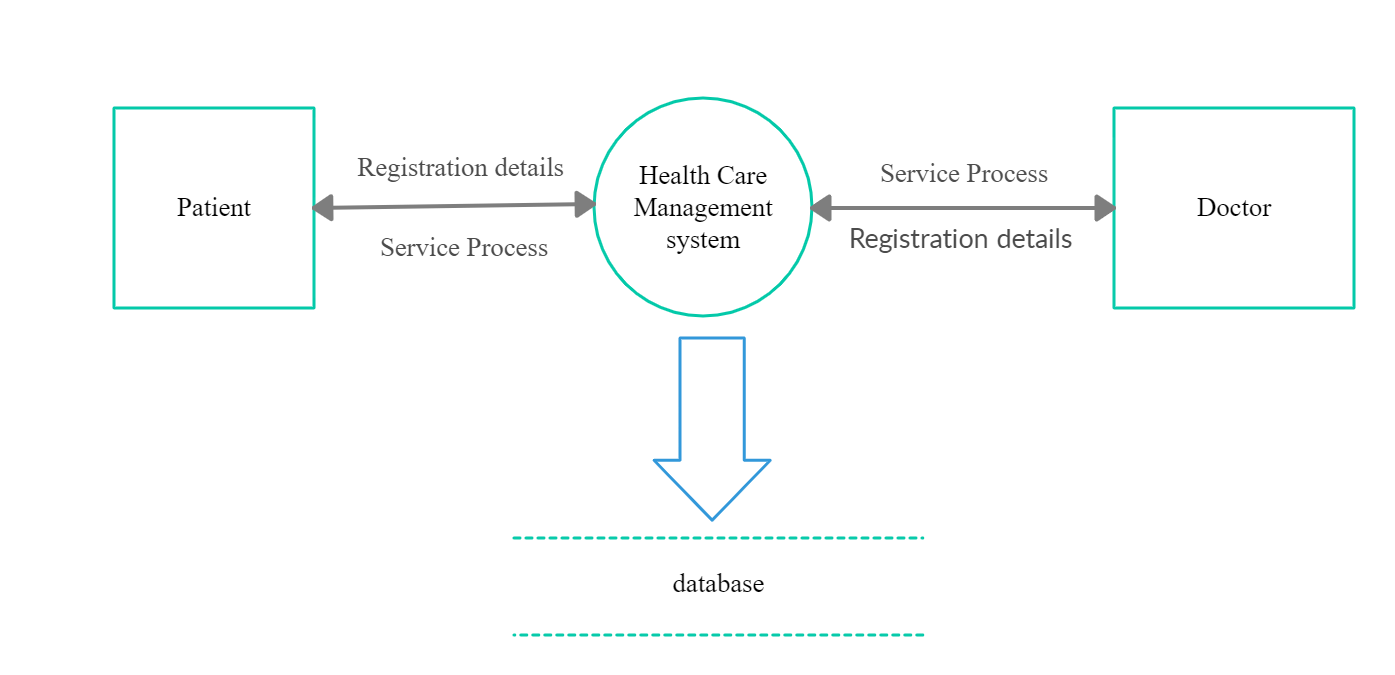


Fig: 3.3

This figure 3.3 shows the 0 levels data flow diagram of Health Care Management System

3.3.2.2: 1st level DFD

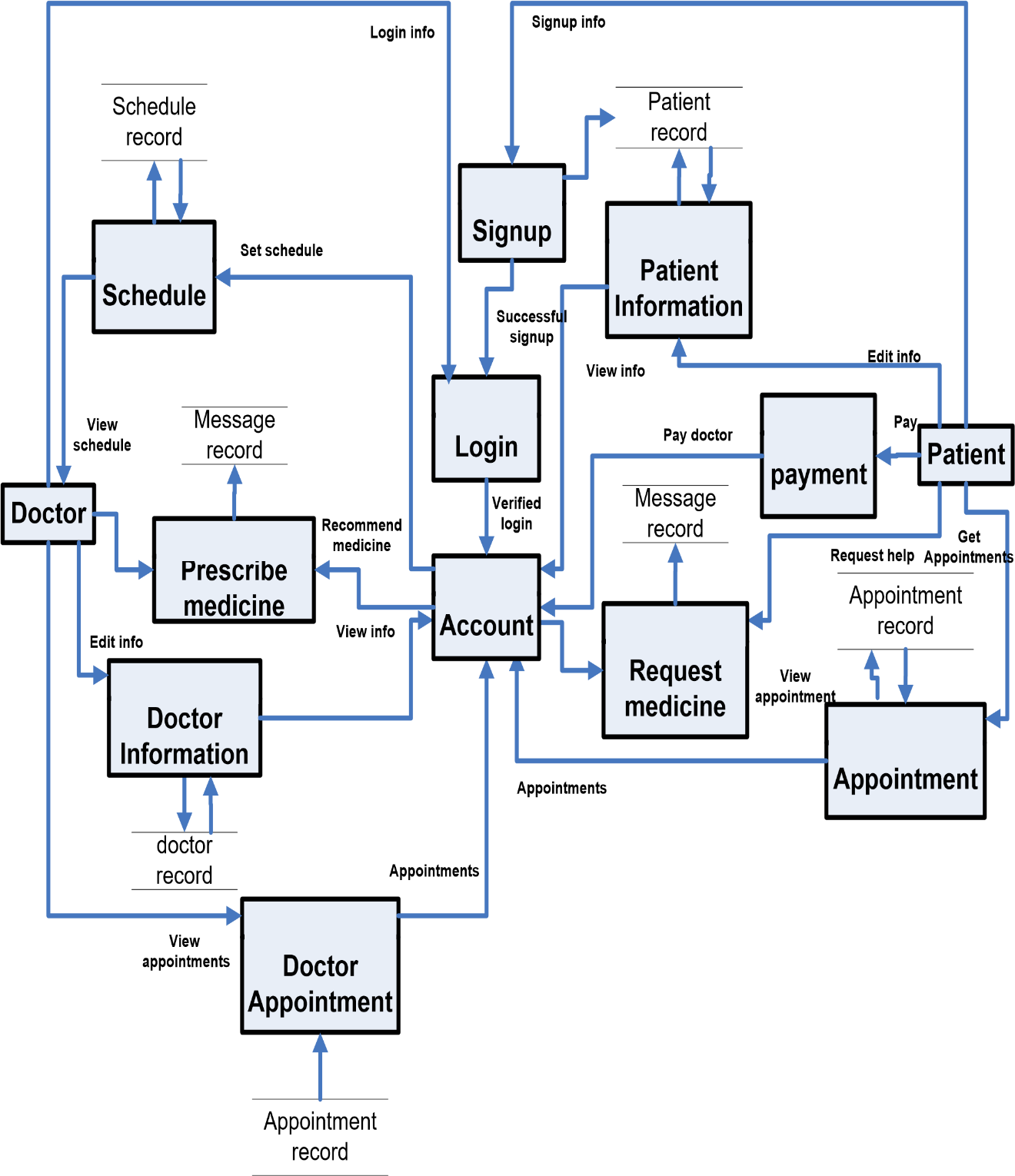


Fig: 3.4

This figure 3.4 shows the 1 levels data flow diagram of Health Care Management System

3.3.2.3: 2nd level DFD:

**Patient**

Patient info

Successfully

registered

Verified

account

patient

record

Verify login

**Doctor**

Update

info

Appointment

record

**Admin**

**Admin**

Sickness

info

Request

appointment

**.**

**Edit**

**/update**

**Account**

**Request**

**medicine**

**Account**

**.**

**Login**

**Account**

**.**

**Fill**

**Signup**

**Form**

**Appointment**

View

info

View

appointment

**Payment**

Pay doctor

Message

record

Fig: 3.5

This figure 3.5 how the 2 levels data flow diagram of Health Care Management System

**­**

**3.3.3 Entity Relationship Diagram**

Entity relationship diagram of patient and doctor is shown below through diagrams 3.5. This is the process to add a student to the database and sort out his activities.

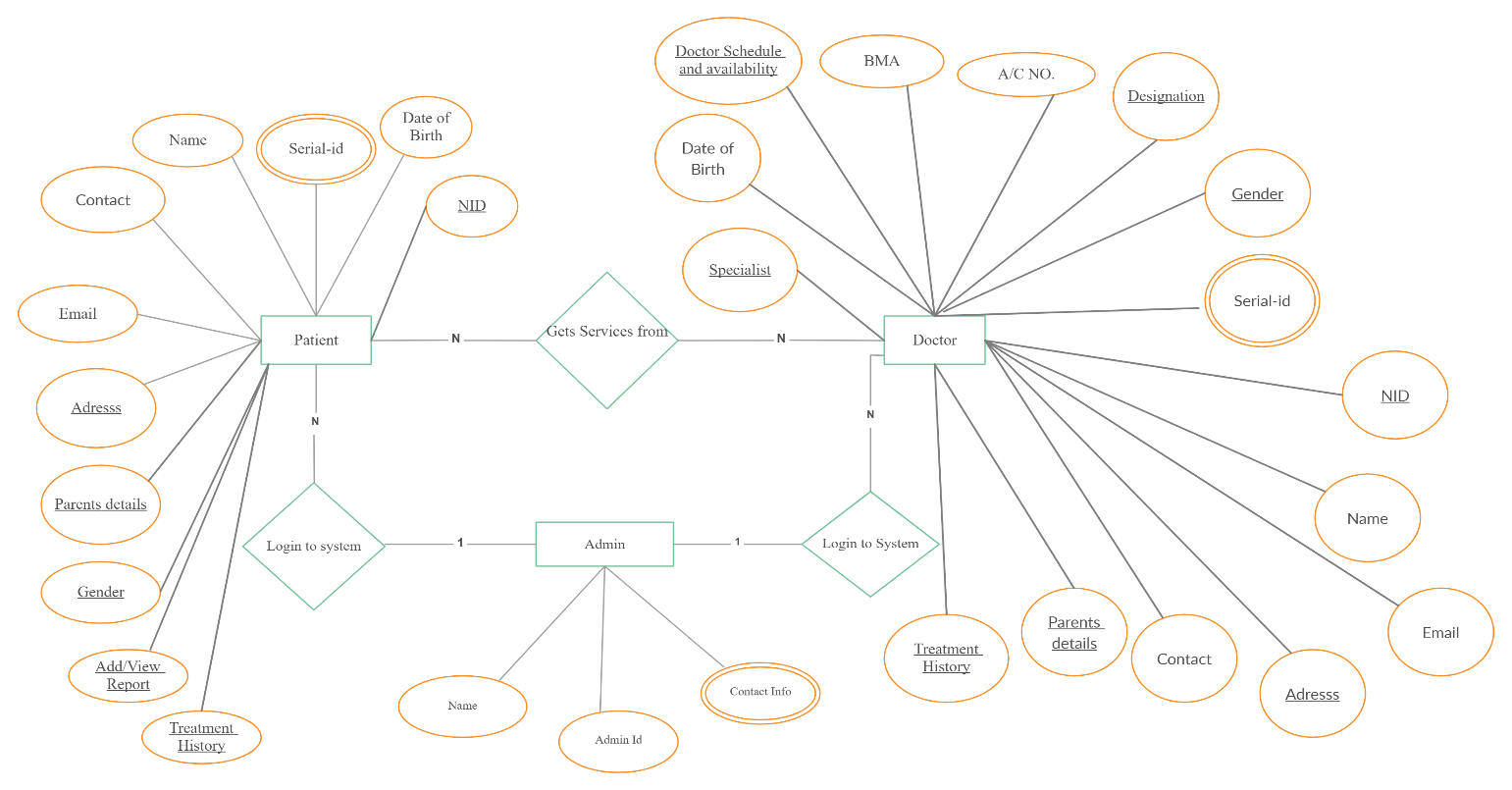
****

Fig: 3.6

This figure 3.6 Entity Relationship Diagram Health Care Management System

**3.3.4 Use Case of the System**

**Patient**

**Doctor**

**System**

**Admin**

**Signup**

**login**

**Addinfo**

**ModifyInfo**

**Get Appointment**

**View Appointment**

**DoctorHelp**

**DeactiveAccount**

**View Appointments**

**PrescribeMedicine**

**AddNewDoctor**

**ViewDoctor**

**DeleteDoctor**

**ViewPatient**

**DeletePatient**

**ViewBooking**

**EditSchedule**

**DeleteSchedule**

**ByName**

**By Specialist**

**By Date**

**login**

**login**

**ViewInfo**

**Addinfo**

**ModifyInfo**

**ViewInfo**

**AddSchedule**

**DeactiveAccount**

Fig: 3.7

This figure 3.7 Entity Relationship Diagram Health Care Management System

**3.3.5 Activity Diagram**

**3.3.5.1 Activity Diagram for Patient**

**Register**

**Login**

**Reg User**

**?**

Yes

No

**Username**

**& password**

**Correct?**

**Add Info**

**Modify Info**

**Appointment**

**Doctor Help**

**Select Doctor**

**Select Date**

**App Success**

**/Reject**

Yes

No

**Logout**

**Select Doctor**

**View Schedule**

**Account**

**Select Pay Option**

Fig: 3.8

This figure 3.8 Activity Diagram for Patient of Health Care Management System

**3.3.5.2** **Activity Diagram for Doctor**

**Register by Admin**

**Login**

**Reg**

**Doctor?**

Yes

No

**Username**

**&**

**password**

**correct**

**?**

**Modify Info**

**Add Schedule**

**Edit**

**/**

**Delete Schedule**

**Select Day**

**Select Time**

**Schedule Added**

Yes

No

**Logout**

**Select Patient**

**Write Comment**

**Send**

**Prescribe Medicine**

**Select Interval**

**Account**

Fig: 3.9

This figure 3.9 Activity Diagram for Patient of Health Care Management System

**3.3.5.3** **Activity Diagram for Admin**

After login into the system what admin can do are shown in figure 3.3.5.3.

**Login**

**Username**

**&**

**password**

**Correct?**

**Add Doctor**

**Doctors**

**Delete Patient**

**Delete Doctor**

Yes

No

**Logout**

**Patients**

**Account**

Fig: 3.10

This figure 3.10 Activity Diagram for Admin of Health Care Management System

**3.3.6 UML diagram of the system:**

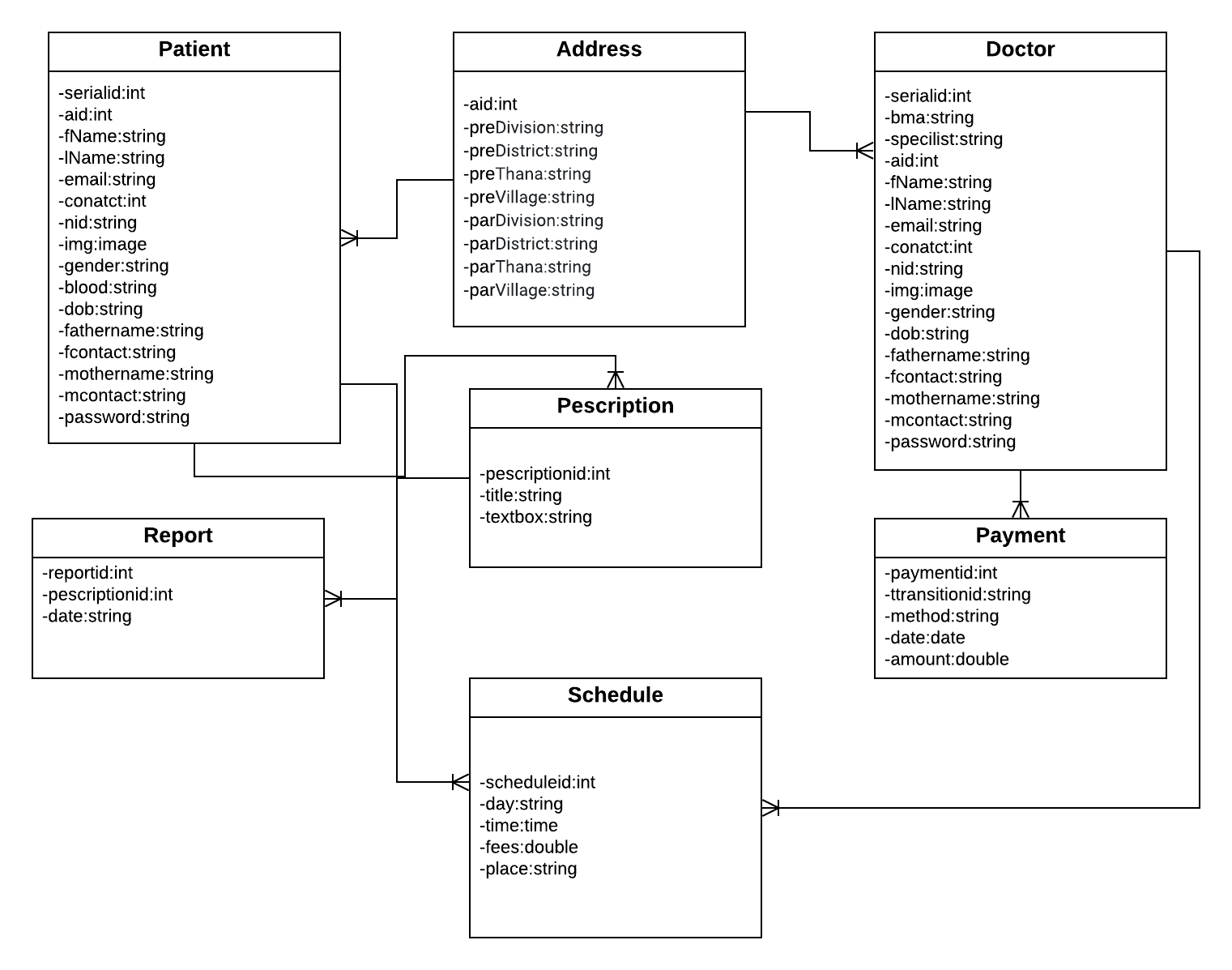
****

Fig: 3.11

This figure 3.11 Activity Diagram for Admin of Health Care Management System

**3.3.4 User Information & Module**

In any Company numerous ventures are going on at a specific time. Each undertaking is separated into different modules relying on the work. Every Module relies on the work. Every module is relying on the assignment it does has gotten different structures, reports and so on.

**3.3.4.1: Patients information:**

* **Patient Addition:** Basically it contain whole information of patients regarding to patients registration form, in this form required information like name, email, gender, serialid , phone number, date of birth, and password of patients are required mandatory field.
* **Doctor Schedule:** Through the dashboard patients can see the specialist list, their schedule, address, contact, amount of fees.
* **Profile View:** Through this segment patients can see their previous disease history, medicine list etc.
* **Test and Report:** Through this section patients able to upload test report and also see all.

**3.3.4.2: Doctor Information:**

* **Doctor Addition:** Basically, it contains whole information of doctors regarding to doctor’s registration form, in this form required information like name, email, gender, serialid, phone number, bank account, BMI, specialist, date of birth, and password of patients are required mandatory field.
* **Patients Schedule:** Through the dashboard doctors can create the schedule list, address, contact, amount of fees.
* **Profile View:** Through this segment doctors can search patients by serial id and add prescription or see previous disease history list by generating OTP etc.
* **Test and Report:** Through this section doctors able to comments test report and prescribed medicine.

**3.3.5 Data Dictionary**

Table 3.1: Patients Addition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Sr. No*** | ***Field name*** | ***Data type*** | ***Constraints*** | ***Description*** |
| 01 | SerialID | Int | Primary Key | Store user ID |
| 02 | First Name | CharField(24) | Not null | Store fist name |
| 03 | Last Name | CharField (24) | Not null | Store last name |
| 04 | Email | EmailField(24) | Not null | Store email |
| 05 | Contact | Integer(24) | Not null | Store contact |
| 06 | Sex | CharField (24) | Not null | Store sex |
| 07 | DOB | Datetime | Not null | Store date of birth |
| 08 | NID | Integer (10) | Not null | Store nationality |
| 09 | Division | CharField (24) | Not null | Store division |
| 10 | District | CharField (24) | Not null | Store district |
| 11 | Thana | CharField (24) | Not null | Store thana |
| 12 | Union | CharField (24) | Not null | Store union |
| 13 | Ward | CharField (24) | Not null | Store ward |
| 14 | Father Name | CharField (24) | Not null | Store father name |
| 15 | Father contact | Integer(24) | Not null | Store father contact |
| 16 | Mother Name | CharField (24) | Not null | Store mother name |
| 17 | Mother contact | Integer(24) | Not null | Store mother contact |

**Doctor Addition:**

Table 3.2: Doctor Addition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Sr. No*** | ***Field name*** | ***Data type*** | ***Constraints*** | ***Description*** |
| 01 | SerialID | Int | Primary Key | Store user ID |
| 02 | First Name | CharField(24) | Not null | Store fist name |
| 03 | Last Name | CharField (24) | Not null | Store last name |
| 04 | Email | EmailField(24) | Not null | Store email |
| 05 | Contact | Integer(24) | Not null | Store contact |
| 06 | Sex | CharField (24) | Not null | Store sex |
| 07 | DOB | Datetime | Not null | Store date of birth |
| 08 | NID | Integer (10) | Not null | Store nationality |
| 09 | Division | CharField (24) | Not null | Store division |
| 10 | District | CharField (24) | Not null | Store district |
| 11 | Thana | CharField (24) | Not null | Store thana |
| 12 | Union | CharField (24) | Not null | Store union |
| 13 | Ward | CharField (24) | Not null | Store ward |
| 14 | Father Name | CharField (24) | Not null | Store father name |
| 15 | Father contact | Integer(24) | Not null | Store father contact |
| 16 | Mother Name | CharField (24) | Not null | Store mother name |
| 17 | Mother contact | Integer(24) | Not null | Store mother contact |
| 18 | BMA | Integer(24) | Not null | Store BMI |
| 19 | Specialists | CharField (24) | Not null | Store specialists |
| 20 | Bank Account | Integer(24) | Not null | Store bank account |

**Report:**

Table 3.3: Report

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Sr. No*** | ***Field name*** | ***Data type*** | ***Constraints*** | ***Description*** |
| 01 | Report name | CharField(24) | Primary Key | Sotre report name |
| 02 | result | CharField(24) | Not null | Store result |
| 03 | Date | Datetime | Not null | Store date |

**Treatment History:**

Table 3.4: Treatment History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Sr. No*** | ***Field name*** | ***Data type*** | ***Constraints*** | ***Description*** |
| 01 | Report name | CharField(24) | Primary Key | Sotre report name |
| 02 | result | CharField(24) | Not null | Store result |
| 03 | Date | Datetime | Not null | Store date |
| 04 | Medicine list | CharField(MAX) | Not null | Store medicine list |

**Doctor Schedule:**

Table 3.5: Doctor Schedule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Sr. No*** | ***Field name*** | ***Data type*** | ***Constraints*** | ***Description*** |
| 01 | Day | CharField(24) | Primary Key | Sotre day |
| 02 | Time | Time(24) | Not null | Store time |
| 03 | Address | CharField(50) | Not null | Store address |

**Payment:**

Table 3.6: Payment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Sr. No*** | ***Field name*** | ***Data type*** | ***Constraints*** | ***Description*** |
| 01 | Account Number | CharField(24) | Primary Key | Store account number |
| 02 | Payment system | CharField(24) | Not null | Store payment system |
| 03 | Transaction id | CharField(24) | Not null | Store transaction id |

**3.3.6 Validation Checks**

Under validation we have provided certain constraints and primary keys to few fields of the tables of the database used in application. This approval made at information base level is recorded underneath:

* **Required Field Validation**: We utilize the necessary field to fill the data mandatory in the undertaking without this approval the information won't be submitted in the task.
* **Not null**: Not invalid requirement is utilized limit field to have invalid qualities. Scarcely any fields in our information base are obligatory to fill.
* **Numeric only**: Numeric just requirements limit field to have numeric qualities as it were. Else it disregards the standard.
* **Character only**: It confines the field to acknowledge just character esteem.
* **Date:** The legitimate date with substantial configuration should be enter in the given textbox.
* **Email:** The @ Symbol is required in this field otherwise it will not work properly.

**Chapter 4**

**PROJECT IMPLEMENTATION & TESTING**

Names: Based on requirement, we picked out a temporary Name. Which is “Hello DIU.” While a name that uses words that summarize the app or services are good for university administrator, which help us in deciding the name and getting it registered.

Hosting: We have to take a space on server for our files so the website would be access through internet.

**4.0 Category & Layout**

The message to get across to the watcher rapidly and without any problem. We have set aside the effort to figure out what the principal topic is or message is to be, at that point separate that topic or message into classes. Which will help manage us through this cycle and help decide a web design that heaps.

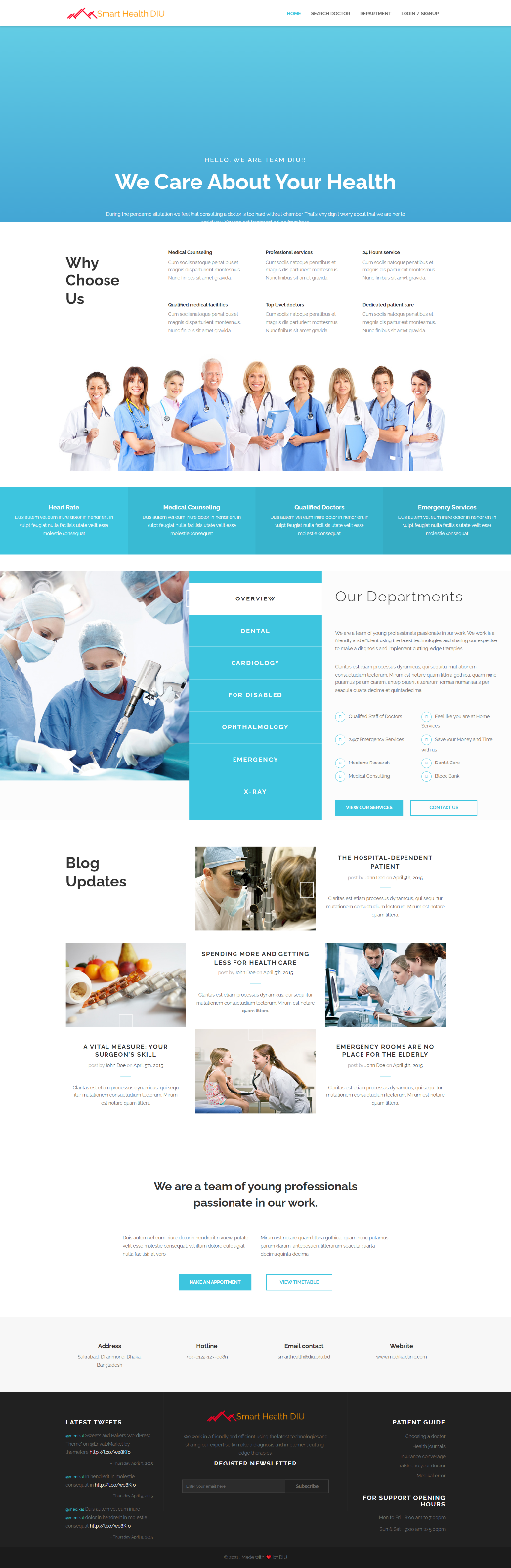
**4.1 Project Design (GUI)**

Figure:4.1 Project Design (GUI)

**4.1.1 Login Form:**

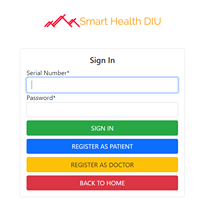
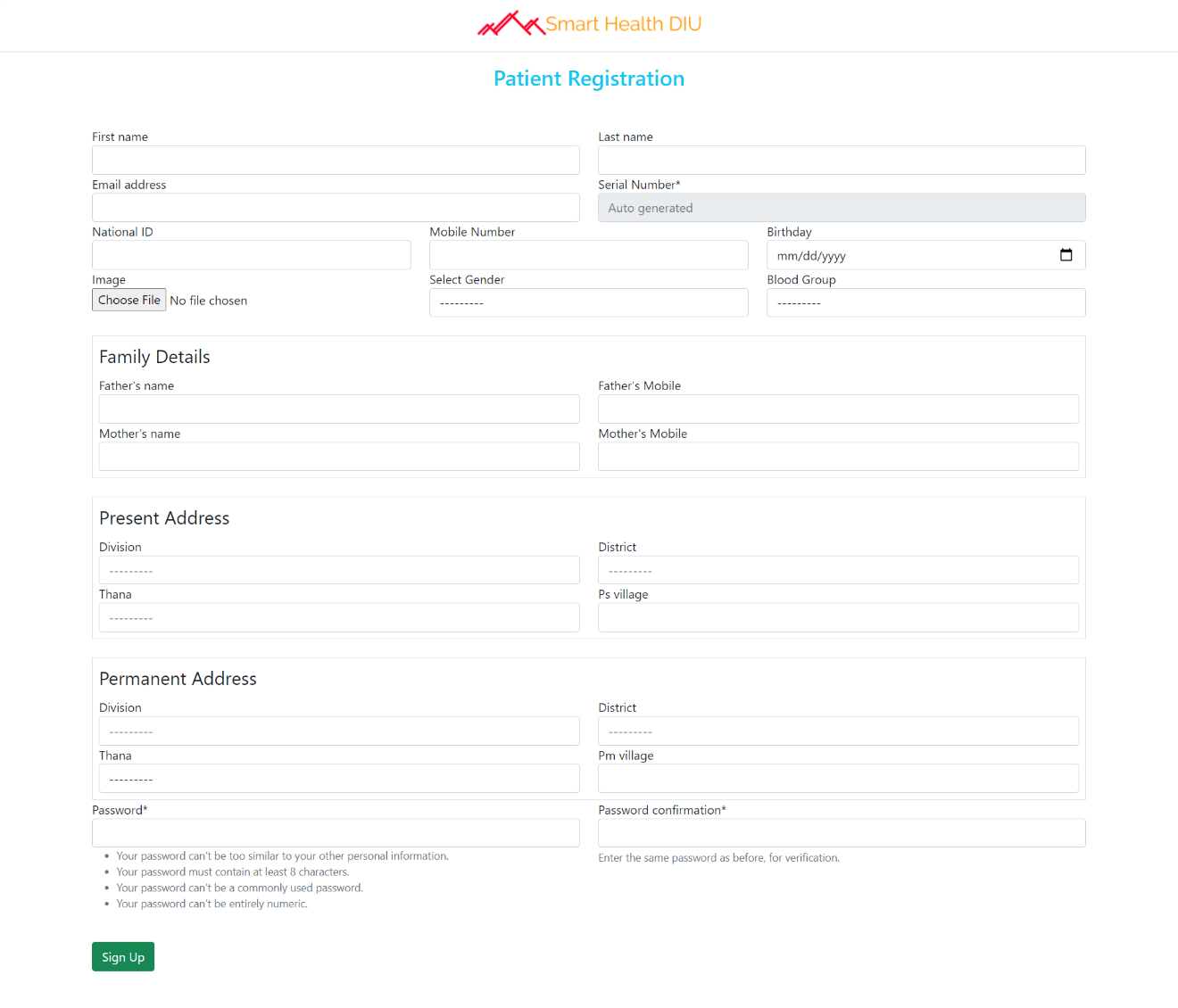
****

Figure:4.1.1 Login Form

**4.1.2 Patient Registration:**

****Figure:4.1.2 Patient Registration

**4.1.3 Patient Profile**

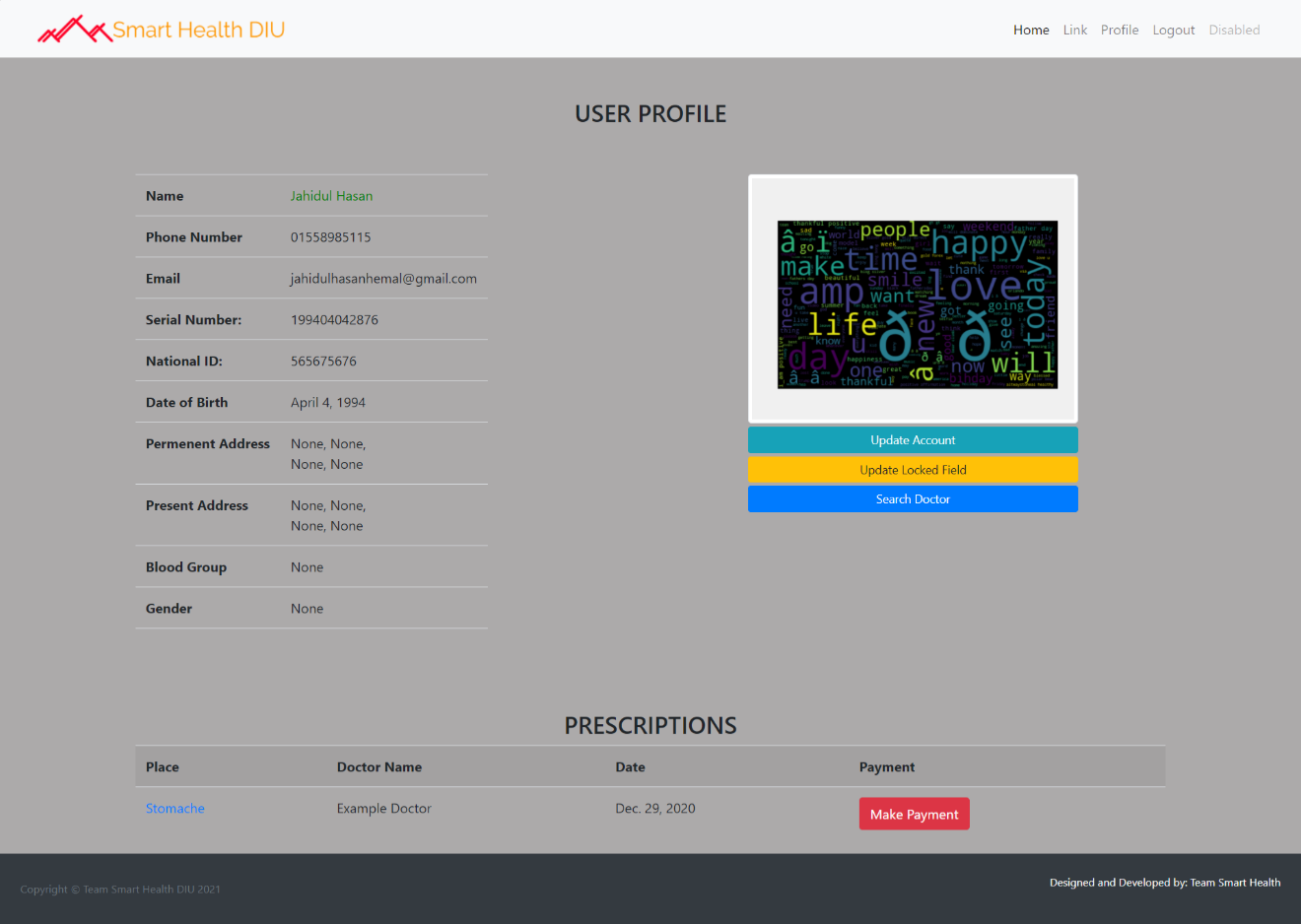
****

Figure:4.1.3 Patient Portfolio

**4.1.4 Payment Gateway:**

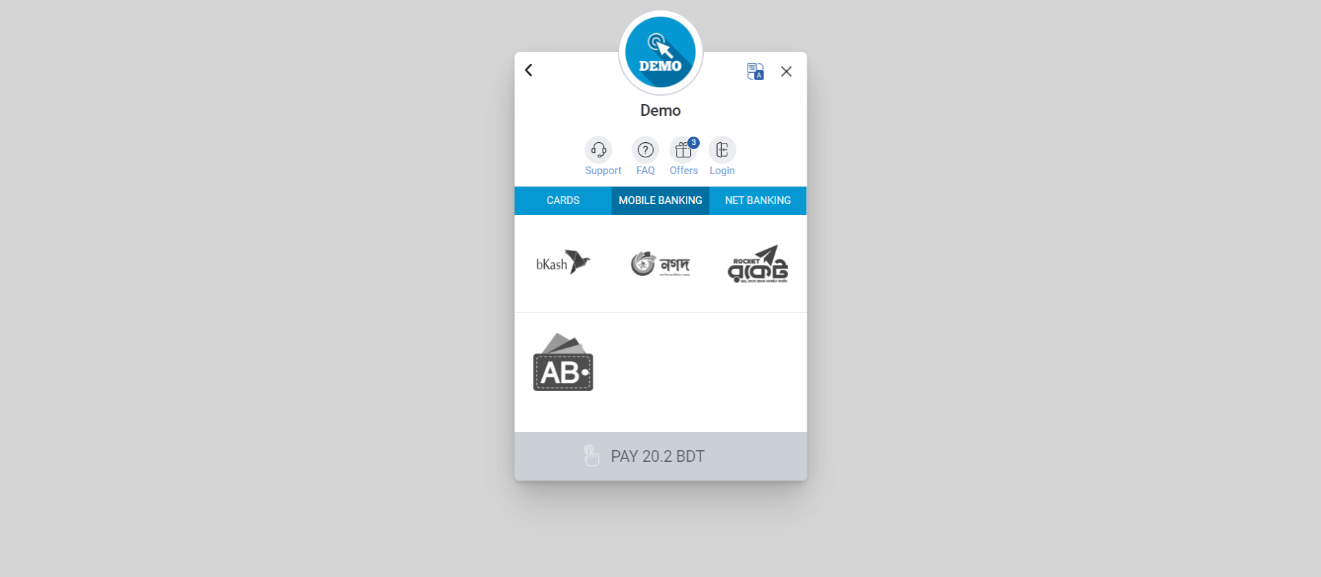
****

Figure:4.1.4 Payment Gateway

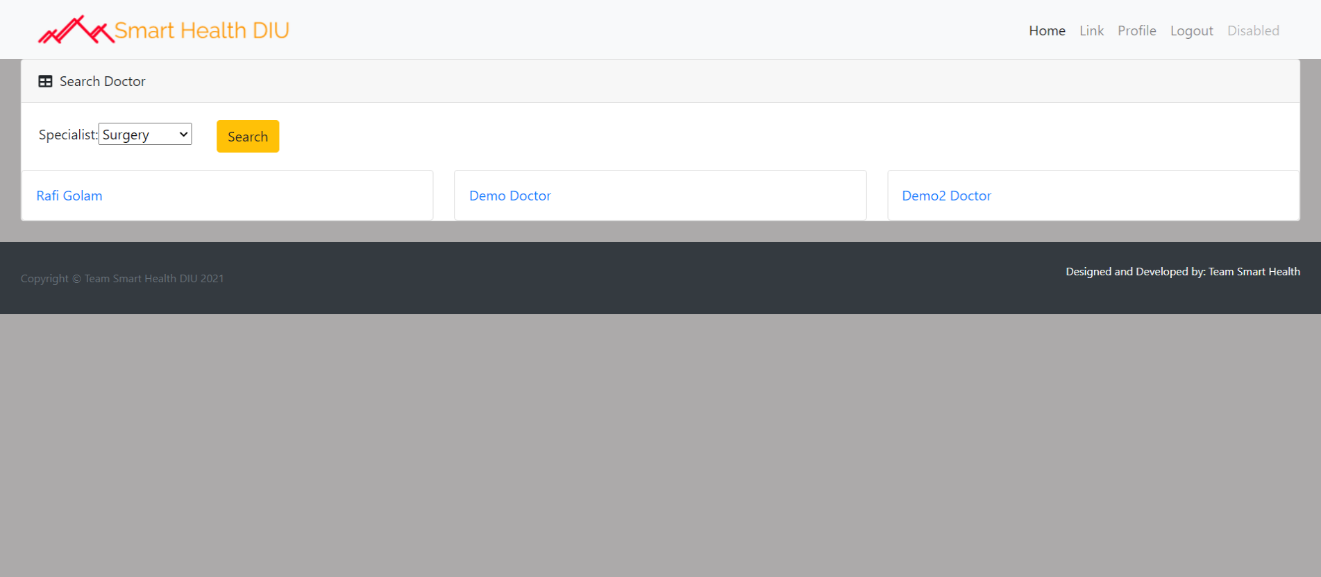
**4.1.5 Search Doctors:**

Figure:4.1.5 Search Doctors

**4.1.6 Search Patient:**

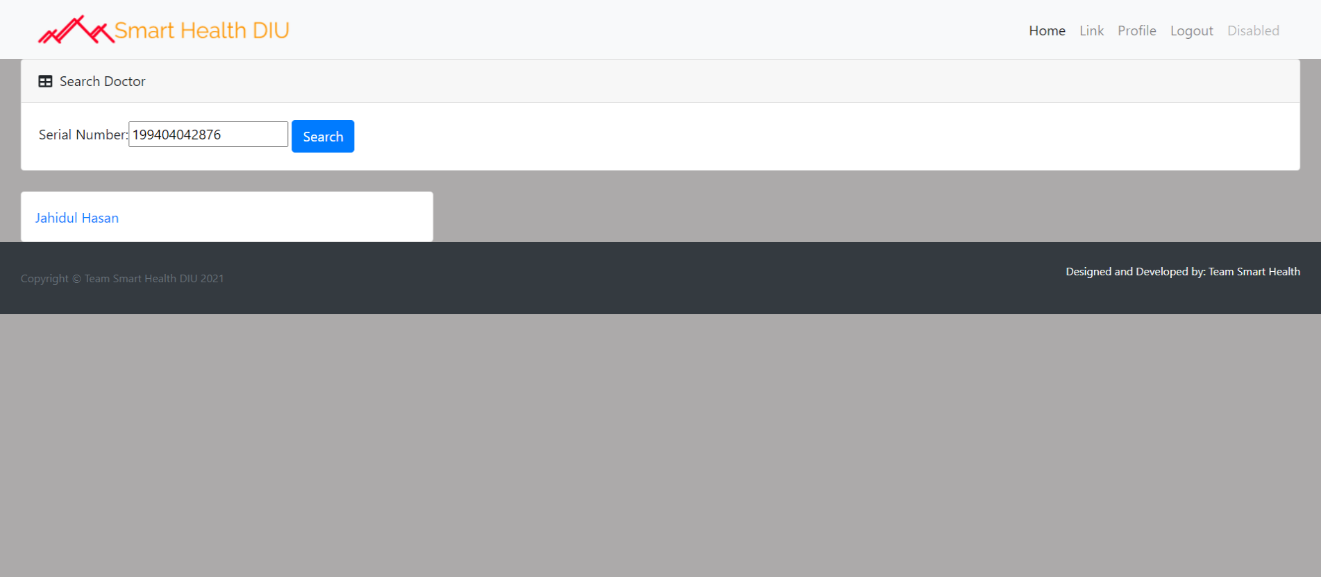
****

Figure:4.1.6 Search Patient

**4.1.7 Add Schedule:**

****

Figure:4.1.7 Add Schedule

**4.1.8 Doctor Profile:**

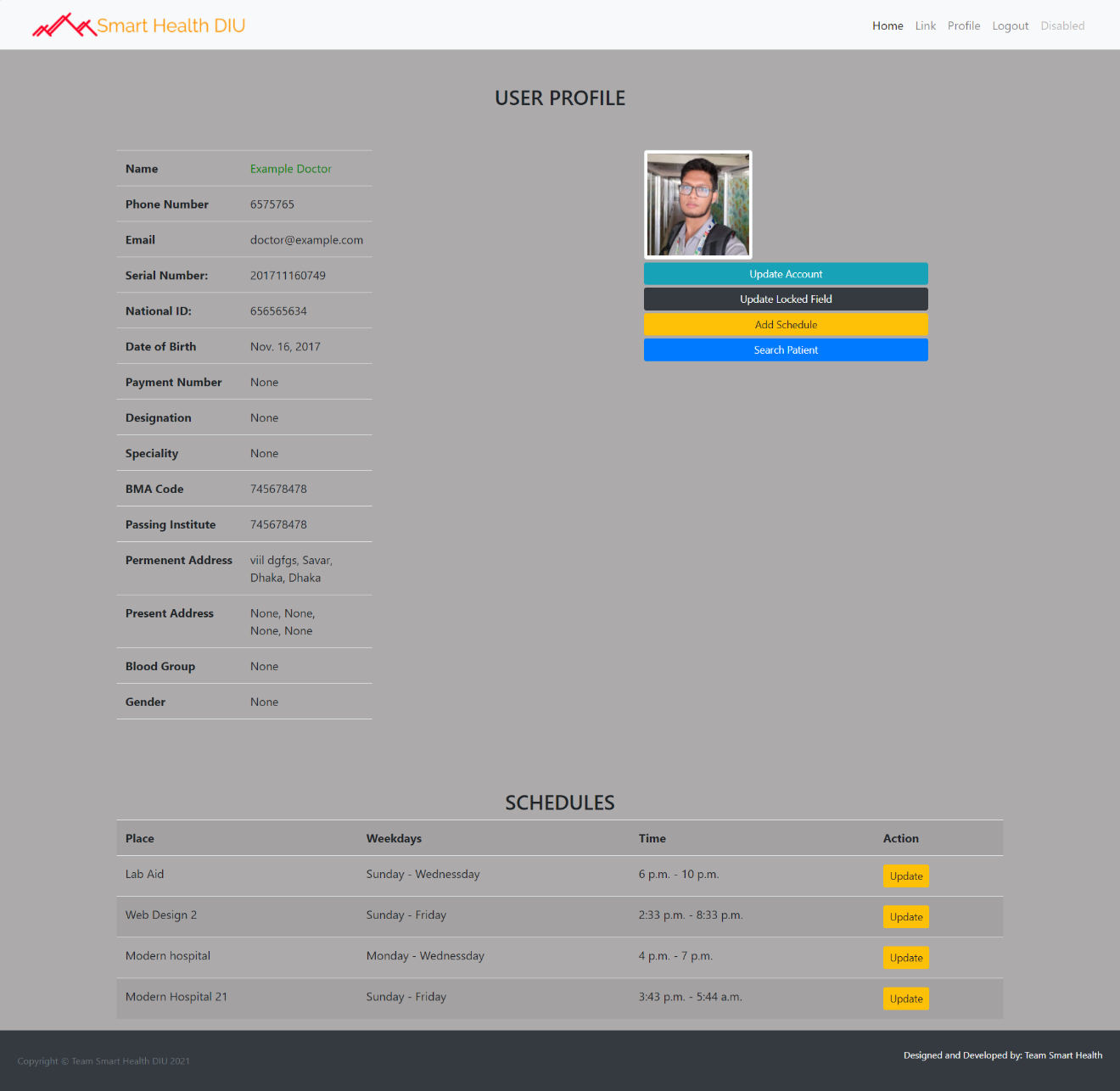
****

Figure:4.1.8 Doctor Profile

**4.1.9 Doctor Registration:**

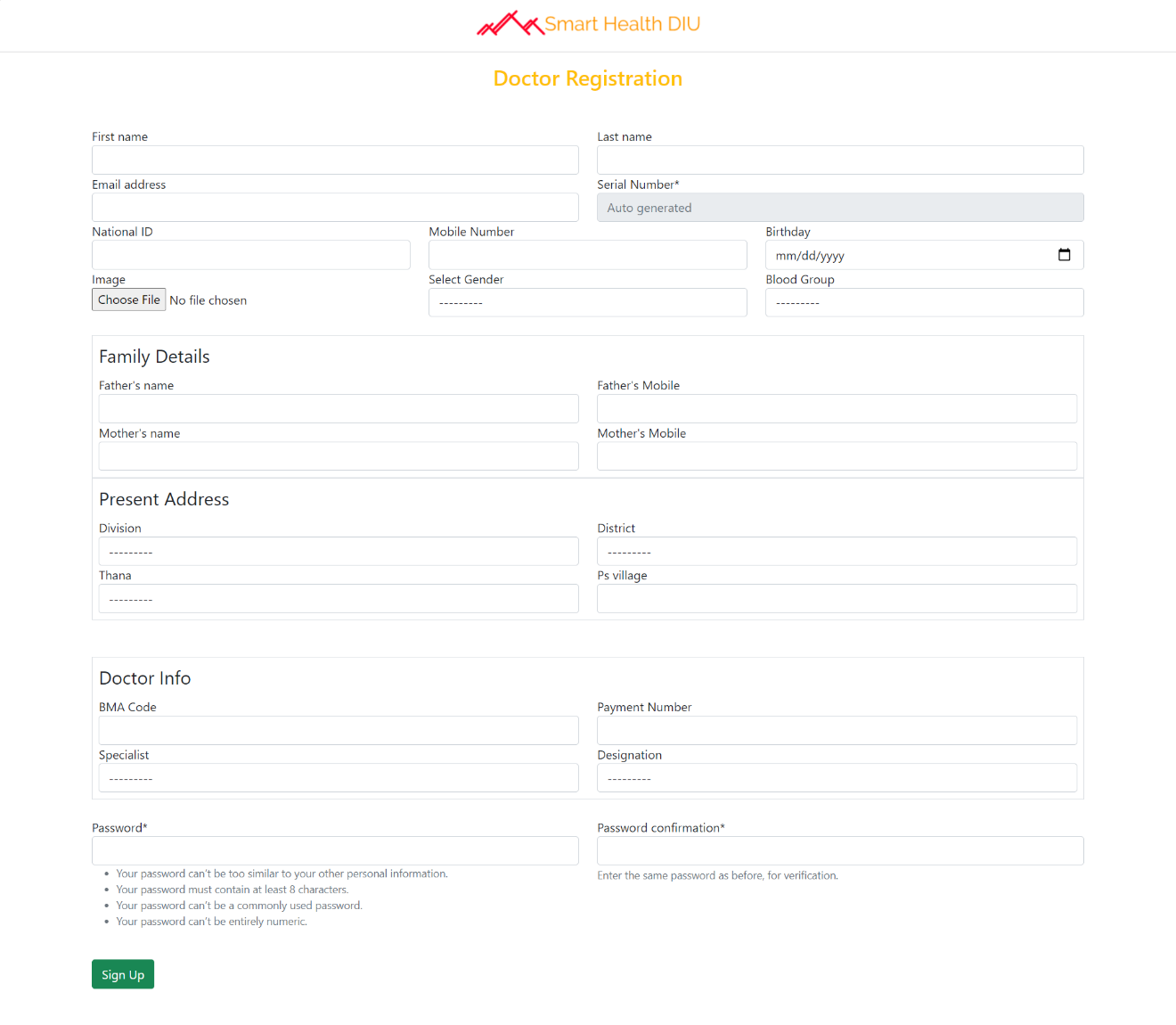
****

Figure:4.1.9 Doctor Registration

**4.2 Testing**

There are different sorts of web application testing without which we can't state that the total framework is appropriately working. Probably the main web testing has been referenced beneath:

**4.2.1. Unit Testing:**

Unit testing occurs at the advancement level. At the point when an engineer constructs a bit of code that conveys a bunch of usefulness, they should test it to ensure it works and that it conveys the necessary usefulness. A designer tests by running the code in their own current circumstance. A bit of code ought to never go into a framework mix climate until it has been unit tried.

**4.2.2. System integration testing (SIT):**

A framework mix climate is a test climate where code is put to guarantee the Application in general cooperates.

Normally there's more than one designer assembling an application or site. Every one unit tests their individual capacities and pages, and one a customary premise, their code is conveyed into the SIT climate and tried together. This guarantees one designer's code doesn't break the others. Normally experiments and test contents are created dependent on the useful prerequisites and tried here.

It gives a more incorporated perspective on the application. This is likewise the climate that gives a reflection of the creation climate. Most applications live with different applications underway. This is the primary opportunity to guarantee that the new application/site doesn't break and isn't broken by different destinations or applications in a similar climate.

**Testing Table**

Table 4.7**:** Testing table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case Title** | **Description** | **Expected Outcome** | **Result** |
| 1 | Successful startup of the application and the login menu being present to take the user to login prompt | The menu is clicked | The login prompt should come up | Passed |
| 2 | Select the user type | Select the appropriate user type i.e. administrator, teacher and student | Ask for username and password | Passed |
| 3 | Registration of Admin | Click on the login button. | The appropriate admin panel will appear | Passed |
| 4 | Update user information | Click edit button | The update page should come with user information | Passed |
| 5 | Delete User | Click delete button | Message should come “whether you want to delete user” and after clicking yes user should be deleted | Passed |

**CHAPTER 5**

**CONCLUSION & FUTURE SCOPE**

**5.1 Conclusion**

The conclusion of “health care management system” is to construct such dynamic website which will provide information about the relative health care institute activities such as treatment history, treatment schedule, payment, search specialist, upload report, medicine list.

Towards the end, we might want to state that the objective, which was at first set up, was accomplished to a decent degree. The venture caused us to understand the importance of creating programming for customer, where the sole point is to learn.

During this task, the genuine significance for following all rule of framework examination and configuration occurred to us. We felt the need of experiencing the few phases.

As we done the underlying examination, presently we can say that this application conceivable to make. Be that as it may, as undertaking will advance there may some adjustment in usefulness of the task.

**5.2 Benefits:**

The advantages of health management system serve everyone involved in the health care process: the administrators who oversees the verification both doctors and patients.

* This application ensures the one of the most important fundamental right which is right to health security.
* This application provides us important information, which could be used researches purpose.
* This application provides specialist doctors lists.
* This application provides user treatment history.
* This project will help to gather information of new born and people all classes.

**5.3 Technology is everywhere**

Health is witnessing unparalleled growth and patients; doctors are increasingly using desktops and mobile devices such as smartphones and tablets to access the virtual platform. There has been a transition in health care administration and the easy-to-use technology solution is doing wonders to face the challenges and improve the way health care authority are managed.

**5.4 Limitation of the framework:**

There are a few constraints of the framework. They are,

* Full system security is impossible.
* Payment framework isn't fully secure.
* It isn't completely automated.
* Admin need to erase unnecessary records physically.

**5.5 Future Plans:**

Likely plans are,

* Make the framework completely mechanized.
* Implement security framework for the venture.
* Make the framework more adaptable for installment and other exchange.
* Develop mobile applications for the framework.

# 

# References

1. Incremental model, http://istqbexamcertification.com/what-is-incremental-model- advantages-disadvantages-and-when-to-use-it/, retrieved on November 20,2020.
2. <https://getbootstrap.com/>, retrieved on November 20,2020.
3. Software design description, http://en.wikipedia.org/wiki/Software\_design\_description, retrieved on November 20,2020.
4. Software design description,

http://www.google.com.bd/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&u act=8&ved=0CCgQFjAB&url=http%3A%2F%2Fwww.cs.bilkent.edu.tr%2F~cagatay%2 Fcs413%2Fsample\_group13\_sdd.doc&ei=ygUFVcrVGceVuQStiYKYDA&usg=AFQjC NFBMb\_j-3tQIIGZ\_dL7z7aw5P6jmw&bvm=bv.88198703,d.c2E,Retrieved on November 20,2020.

1. Traceability, http://en.wikipedia.org/wiki/Traceability, retrieved on November 20,2020.
2. Maintenance, http://en.wikipedia.org/wiki/Software\_maintenance, retrieved on November 20,2020.
3. Maintenance, http://agile.csc.ncsu.edu/SEMaterials/MaintenanceRefactoring.pdf, retrieved on November 25,2020.
4. <https://docs.djangoproject.com/en/3.1/> November 25,2020.
5. <https://stackoverflow.com/> November 25,2020.