

අයු

PROJECT AYU
SRS

Abstract

The ‘Ayu’ (a patient medical data management system) project's comprehensive functional and non-functional requirements are described in detail in this System Requirements Specification (SRS) document. This extensive paper functions as a guide for the development, maintenance, and assessment of the ‘Ayu’ platform, which seeks to transform healthcare administration via cutting-edge technological innovations.

Numerous needs are covered in the SRS document, such as patients’ data integration into NFC cards, user registration and authentication, patient profile management, telemedicine integration, appointment scheduling, medical record access, security, scalability, and compliance. To guarantee clarity, completeness, and traceability throughout the project life-time, these requirements are painstakingly documented.

This SRS document helps stakeholders, members of the project team including developers, testers, and end users, to have a shared understanding of the functionality, performance, and quality standards of the ‘Ayu’ platform by outlining precise requirements. It acts as a reference manual to direct the system's design, development, testing, deployment, and maintenance, promoting efficient communication and cooperation between all stakeholders.

This document is essential to the ‘Ayu’ project's successful completion because it synchronizes project goals with user requirements, legal requirements, and industry best practices. It provides the framework for creating a strong, intuitive, and scalable healthcare management system that boosts healthcare delivery, improves patient outcomes, and establishes new benchmarks for quality in the healthcare sector.

Table Of Content

1.0 Introduction	1
1.1 Geographical Context	1
1.2 Problem Statement	1
1.3 Objectives	2
1.4 Methodology	3
2.0 Background	4
2.1 Background	4
2.2 Functional Requirements	4
2.3 Non-Functional Requirements	5
2.4 Alternative Design Strategies	6
3.0 Methodology	7
3.1 Introduction	7
3.2 Web Application	7
4.1 Planning and Requirement Analysis	9
4.1.1 Estimated time table	11
4.1.2 Economic	12
4.3 Designing	13
4.4 Development	14
4.5 Testing	14
4.6 Deployment & Maintenance	14
5.0 System Design Methodology	15
6.1 Architecture	16
6.2 Entity Relationship Diagram	17
6.4 Table Model	19
6.5 Use Case Diagram	20
6.6 Natural Language Sentences	21
6.7 Activity Diagram	22
6.8 Data Flow Diagram	30
6.10 Sequence Diagram	33

07. User Interface	34
7.1 Sign In Page	34
7.2 Forgot Password Page	35
7.3 Reset Password Page	36
7.4 Patient's Dashboard Page	37
7.5 Patient's Appointments Page	38
7.6 Patient's Medical Report Page	39
7.7 Patient's Details Page (Doctor's)	40
7.8 Hospital Patient Management Page	41
7.9 Hospital Doctor Management Page	42
7.10 Hospital View Doctor Pop Up	43
7.11 Admin Hospital Manage Page	44
7.12 Admin Health Card Manage Page	44
7.13 Admin Health Card Confirm Pop up	45
08. Evaluation	46
8.1 Introduction	46
8.2 Software Testing Levels	46
8.3 Non-Functional Testing	47
8.4 Test Plan	47
8.5 Test Strategy	48
8.6 Test Cases and Test Results	50
09. Conclusion	53
9.1 Introduction	53
9.2 Problem Encountered	54
10. Lesson Learn	54
12.1 Design Documentation	56
12.2 Use Case Description – User Management	56
12.3 System Documentation	56

1.0 Introduction

1.1 Geographical Context

The goal of the Patient Medical Data Management System ('Ayu') project is to completely transform the way healthcare is managed by offering a complete solution for organizing patient medical records across several hospitals in a particular area. The project was started to solve the difficulties that patients and healthcare professionals have in safely and efficiently accessing, updating, and exchanging medical records.

The project Ayu is designed to service patients and healthcare facilities in a certain geographic area that includes rural, suburban, and urban areas. This geographic context could be a single city, an area, or a network of healthcare facilities spread across several sites. Understanding the target area's distinct healthcare infrastructure, regulatory landscape, and patient demographics depends on knowing the project's precise geographic extent.

The initiative intends to address the many healthcare needs of densely populated populations in urban locations, where numerous hospitals and clinics are located in close proximity to one another. Suburban areas offer a variety of healthcare services to people living in residential areas and business centers, from big hospitals to tiny clinics. The project tackles the issues of healthcare access and connectivity in rural areas, offering vital medical services to marginalized communities in isolated locales.

The project Ayu aims to create a customized solution that answers the unique healthcare demands and difficulties within the target area by taking the local context into account. The project's objectives are to optimize healthcare delivery, improve patient outcomes, and raise the general standard of healthcare services within the specified geographic area through thoughtful implementation and modification.

1.2 Problem Statement

There is currently no effective way for patients to manage and access their medical records across several hospitals in a given area within the present healthcare system. Medical information is frequently difficult for patients to get, update, and share, which causes delays in diagnosis, treatment, and continuity of care.

Disjointed data management systems are a challenge for healthcare professionals, impeding cooperation and patient-centered care delivery. Inaccuracies, inefficiencies, and differences in healthcare outcomes are caused by this fragmentation. In order to overcome these obstacles, the

Patient Medical Data Management System ('Ayu') project intends to create a single, unified platform for the smooth administration of medical records throughout the intended geographic region.

1.3 Objectives

The development of a single platform and NFC card that facilitates the easy maintenance of patient medical records across several institutions in a geographic area is the primary goal of the Patient Medical Data Management System ('Ayu') project.

The following can be identified as the sub objectives of the proposed system.

- **Effective Data Management:** Put in place a centralized database system that enables safe and effective patient medical information retrieval, updating, and sharing for healthcare practitioners.
- **Enhanced Patient Experience:** By giving patients easy access to appointment booking, online services like personal medical data retrieval via the 'Ayu' platform, you can improve the overall patient experience.
- **Better Healthcare Delivery:** By standardizing data management procedures and guaranteeing interoperability across healthcare systems, you may help healthcare practitioners collaborate and communicate with one another.
- **Compliance and Security:** Make sure that all data privacy laws and healthcare standards are followed, and have strong security measures in place to protect patient information.
- **Scalability and Adaptability:** To meet the changing demands of patients and healthcare providers in the region, the 'Ayu' platform should be designed to be both scalable and adaptable.
- **Stakeholder involvement:** To guarantee the project Ayu's success and long-term viability, encourage cooperation and involvement with stakeholders, such as patients, healthcare providers, regulatory agencies, and technological partners.

By accomplishing these goals, the project Ayu hopes to improve patient care, improve healthcare outcomes, and maximize the effectiveness of healthcare delivery while revolutionizing healthcare administration in the targeted geographic area.

1.4 Methodology

In order to guarantee the effective delivery of the solution, the Patient Medical Data Management System ('Ayu') project's development process employs a methodical approach that includes multiple stages, such as analysis, design, implementation, testing, and deployment. The main actions engaged in each stage are delineated in the technique described below:

- **Phase of Analysis:** The project team thoroughly examines the state of healthcare, taking into account the needs and preferences of both patients and healthcare providers, the state of current data management systems, and regulatory and compliance issues.
- **Design Phase:** The project team designs the 'Ayu' platform's architecture and functionality based on the analysis phase's results. This includes specifying the database structure and data flow, establishing the system requirements, and producing mockups of the user interface.
- **Phase of Implementation:** The project team starts working on coding and programming the software in accordance with the design criteria as the 'Ayu' platform development process gets underway. Creating the front-end interfaces, constructing the back-end infrastructure, and integrating essential features like patient account creation, access to medical records, appointment scheduling etc are all part of this phase.
- **Testing Phase:** After the first development is finished, the project Ayu is put through a thorough testing process to find and fix any faults, flaws, or inconsistent functioning. To guarantee the dependability, usability, and security of the platform, a variety of testing techniques are used, such as unit testing, integration testing, and user acceptability testing.
- **Phase of Deployment:** After testing and validation go well, the 'Ayu' platform is set up for pilot testing at a few healthcare facilities in the intended geographic area. Before the platform is fully deployed, user and stakeholder feedback is gathered and integrated into the final version.

To maintain accountability, transparency, and alignment with project objectives, the project team adheres to best practices in project management, communication, and cooperation throughout the development phase. To monitor progress and handle any problems or difficulties that may come up throughout the project, regular meetings, progress reports, and milestone reviews are held.

2.0 Background

2.1 Background

The growing demand for a more effective and easily available way to manage patient medical records gave rise to the Patient Medical Data Management System (Ayu) initiative. Conventional healthcare systems frequently use disjointed, paper-based record-keeping techniques, which creates problems with patient experience, continuity of care, and data accessibility. A system that may improve patient outcomes, foster greater provider collaboration, and streamline medical record management was sought for by stakeholders in the healthcare business, including patients, healthcare providers, and regulatory agencies.

The project Ayu's origin is deeply ingrained in the larger framework of the healthcare industry's digital revolution. The benefits of using technology to solve persistent problems in healthcare administration are becoming more widely acknowledged as a result of technological breakthroughs and the rising digitalization of medical information. This involves the creation of telemedicine platforms, electronic health record (EHR) systems, and other healthcare IT solutions meant to increase effectiveness, accessibility, and treatment quality.

The project Ayu was designed to provide a uniform platform that would make it possible for patients and healthcare professionals to access, update, and share medical records across many hospitals in the target geographic area in a seamless manner in response to these trends and problems. The goal of the project Ayu is to optimize healthcare delivery, improve patient experience, and eventually improve healthcare outcomes within the established context by centralizing patient data.

The background of the project Ayu serves as a reminder of the value of creativity, teamwork, and stakeholder involvement in tackling difficult healthcare issues and promoting constructive change in the healthcare sector. The project Ayu aims to revolutionize healthcare administration and open the door for a more effective, accessible, and patient-centered healthcare system by utilizing technology and taking a patient-centric approach.

2.2 Functional Requirements

The project Ayu's functional requirements specify the features and functionalities that the system must have in order to satisfy users' needs, including those of patients and healthcare practitioners. These specifications spell out the functionality and behavior expectations for the system. The project Ayu's primary functional needs are as follows:

- **User Registration and Authentication:** Users should be able to safely register and create accounts on the system, including patients and healthcare professionals. To guarantee data security, authentication methods like username / password have to be put in place.
- **Patient Profile Management:** Individuals should have the ability to update their personal information and access their medical history.
- **Access to Medical Records:** The system should allow authorized healthcare providers to view patient medical records via patient's health card. During consultations and treatments, this functionality enables healthcare providers to retrieve patient data, such as medical history, diagnosis, treatments, and prescriptions.
- **Data Security and Compliance:** To guarantee the privacy, availability, and integrity of patient information, the system must abide by data security guidelines and healthcare laws. This entails access controls, audit trails to monitor system activity, and encryption of sensitive data.

The project Ayu is built upon these functional requirements, which direct the creation and deployment of an all-inclusive, user-focused healthcare management system.

2.3 Non-Functional Requirements

The project Ayu's non-functional requirements, as opposed to functional requirements, outline the quality attributes, limitations, and standards that the system must meet. These specifications put a strong emphasis on features including scalability, security, usability, and performance. The project Ayu's primary non-functional requirements are as follows:

- **Performance:** When it comes to user engagements like accessing medical records, making and viewing appointments, adding doctor reviews, and getting personalized notifications, the system should be fast and responsive. Even at times of high usage, response times ought to be within reasonable bounds.
- **Dependability:** The system must be dependable and accessible to users at all times. In order to prevent data loss, this entails limiting downtime, guaranteeing data integrity, and putting backup and recovery procedures in place.
- **Usability:** Both patients and healthcare practitioners should find it simple to navigate the system's clear, uniform interface, which should be straightforward to use and intuitive. Usability principles should be the foundation for designing user interfaces in order to reduce user errors and increase user pleasure.

- **Security:** To guard against unauthorized access, alteration, or disclosure of patient data, the system needs to implement strong security measures. This entails putting in place access controls, user authentication, encryption for data transport and storage, and audit trails to monitor system activity.
- **Scalability:** Over time, the system should be able to handle an increase in the number of users, patient records, and transactions. This entails considering scalability when building the system architecture and dynamically allocating resources to accommodate growing workloads.
- **Communication:** The system must be able to work with current healthcare IT standards and systems, enabling easy integration with lab information systems, electronic health record (EHR) systems, and other healthcare applications. Interoperability and data interchange between various healthcare systems and providers are facilitated by this.

The Ayu project's overall quality, usability, and performance are greatly influenced by these non-functional requirements, which make sure that the system satisfies user expectations and follows industry standards and best practices.

2.4 Alternative Design Strategies

There are numerous models for software modeling, including spiral, waterfall, iterative, and agile models.

Only projects with stable requirements may employ the waterfall method. So we will use the agile model for this project.

Until the complete system is realized, the agile process begins with the basic implementation of a subset of the software requirements and gradually improves the evolving versions. New functional capabilities and design adjustments are implemented at every sprint.

Free and open-source software will be used in software development since it will save development costs.

3.0 Methodology

3.1 Introduction

Project Ayu describes the structure and methodical approach that were employed to successfully plan, carry out, and achieve the project's goals. The project team can navigate several phases of the project lifecycle, from early analysis and requirements collection to system deployment and maintenance, with the help of the methodology, which offers an organized framework. It includes the fundamental ideas, procedures, and methods necessary to guarantee the 'Ayu' platform's effective creation and execution.

Setting the stage for the project's execution, the methodology's introduction lists the main goals, guiding principles, and important players. It provides an explanation of the methodology that was selected, along with its fit for the goals, limitations, and scope of the project. In addition, the introduction gives a summary of the important phases, tasks, and outputs associated with the project lifecycle, laying the groundwork for a thorough comprehension of the methodology's use.

For the project team, the methodology introduction acts as a road map, outlining the strategy and standards for project execution. It creates a shared understanding of the project's objectives, schedule, and success criteria between stakeholders. It also highlights how crucial cooperation, communication, and ongoing improvement are to the success of a project.

In summary, the methodology's introduction offers a fundamental structure that facilitates the effective implementation of the project Ayu. It guides the project team through every phase of the development process and guarantees compliance with stakeholder expectations, industry best practices, and project objectives.

3.2 Web Application

Procedure for Creating Web App for the project Ayu:

1. Requirements Analysis and Gathering:
 - Collect requirements from stakeholders through workshops, questionnaires, and interviews for the online and mobile applications.
 - To specify the scope and goals of the applications, analyze user demands, system requirements, and functional specifications.
 - To guarantee alignment with project goals, identify the essential features, user roles, and use cases for the application.

2. Design:

- For web application, create wireframes and prototypes with an emphasis on user interface (UI) and user experience (UX) design.
- Create the screen layouts, navigation flows, and information architecture with the best possible usability and accessibility in mind for the application.
- Establish the visual design components, such as font, color schemes, and branding, to ensure uniformity in the application.

3. Development:

- When creating the application, pick the right technologies and frameworks keeping in mind aspects like scalability, performance.
- Establish the backend infrastructure, data storage methods, and APIs necessary to facilitate data synchronization, storage, and retrieval.
- Create front-end features and components while adhering to best practices for version control, testing, and coding to guarantee maintainability and quality of the code.

4. Testing

- Conduct thorough testing of the web application to find and fix any faults, malfunctions, or inconsistent behavior.
- To guarantee a consistent user experience across various devices, browsers, and screen sizes, do functional, usability, and compatibility testing.
- To boost the testing process, implement automated testing frameworks and tools.

5. Implementation

- Install the web application on web hosting platforms (such as AWS, Azure, Heroku).
- To automate the deployment process and guarantee a seamless distribution of updates and additions, configure deployment pipelines and continuous integration/continuous deployment (CI/CD) processes.
- Track application performance, adoption rates, and user satisfaction levels by keeping an eye on deployment metrics and user feedback.

6. Maintenance and Support

- Update and maintain web applications, taking care of any problems, bugs, or feature requests that users may submit.
- To maintain long-term viability and relevance, update and improve the app frequently in response to user input, technical developments, and shifts in business requirements.

- To promote efficient use of the app and optimize their value, provide users and stakeholders with technical support, documentation, and training materials.

The development team can successfully ensure the delivery of robust, user-friendly, and scalable applications that satisfy the goals and expectations of stakeholders and end users by adhering to this methodology when creating web applications for the project Ayu.

4.0 Software Development Life Cycle (SDLC)

4.1 Planning and Requirement Analysis

To ensure successful project execution, the project Ayu's Software Development Life Cycle (SDLC) planning phase entails defining the project's scope, objectives, schedules, resources, and deliverables. Throughout the development lifecycle, this phase directs decisions and subsequent actions and lays the groundwork for the entire project. Below is a thorough rundown of the Ayu project's SDLC planning phase:

1. Project Start-Up:

- **Describe the Project's Vision:** Clearly state the general aims and objectives of the project Ayu, with a focus on how cutting-edge technological solutions will revolutionize healthcare administration.
- **Determine the Stakeholders:** Determine the roles and duties of the important parties involved in the project, such as the developers, testers, patients, sponsors, and regulatory agencies.
- **Analyze the feasibility:** Determine the project's viability and potential hazards by evaluating its feasibility from a technical, operational, economic, and legal standpoint.

2. Project Planning:

- **Establish Project Scope:** Clearly state what features, functionalities, and deliverables the project Ayu will include in its finished solution. This minimizes scope creep throughout development and helps control expectations.
- **Create a project schedule:** Make a thorough project plan that includes all of the tasks, deadlines, resources, and dependencies. To visualize and measure progress, use project management tools like Gantt charts, agile boards, or Kanban boards.
- **Determine the Risks:** Determine the risks and uncertainties that could affect the project's success, such as obstacles related to technology, a lack of resources, problems with regulatory compliance, or modifications to the project's specifications. Create risk-reduction plans to deal with these hazards early on.

3. Gathering and Analysis of Requirements:

- **Involve Stakeholders:** Work together with stakeholders to collect and evaluate needs for the Ayu platform, encompassing both non-functional (such as security, scalability, and usability) and functional (such as patient medical data management and appointment scheduling) requirements.
- **Set Prioritization for Needs:** Sort needs according to significance, practicality, and effect on project objectives.

4. Allocation of Resources:

- **Allocate Resources:** Determine the needs and skill levels of the project before assigning human resources, such as developers, testers, designers, project managers, and subject matter experts.
- **Obtain Infrastructure and Tools:** Acquire the infrastructure, software, hardware, and tools needed for the 'Ayu' platform's development, testing, and implementation. Ascertain that every team member has access to the equipment and materials required to carry out their responsibilities successfully.

5. Communication & Cooperation:

- **Create Communication methods:** Provide regular methods and procedures for team members, sponsors, and stakeholders to communicate with one other. To promote communication and cooperation, make use of tools like project management software, video conferencing, email, and instant messaging.
- **Call Meetings with Stakeholders:** Arrange for frequent get-togethers with stakeholders to provide project updates, solicit input, and handle any issues or worries that surface throughout the planning stage.

6. Maintaining documentation:

- **Produce Project Records:** Provide thorough documentation, such as project proposal, requirement specifications, and project plans. Make sure that during the project lifespan, documentation is updated often to reflect changes and advancements.
- **Establish Document regulation:** To regulate project documentation distribution, access restrictions, and versioning, put in place document control methods. Make sure that all team members have access to the most recent versions of project documentation by centralizing document repositories.

7. Planning for Quality Assurance:

- **Define Quality Standards:** Specify requirements for functionality, performance, usability, security, and compliance in order to establish quality standards and criteria for the 'Ayu' platform.

- **Create Testing Plans:** Establish testing frameworks, techniques, and strategies to guarantee comprehensive testing of the Ayu platform at different phases of development. To confirm the dependability and quality of the program, schedule regression, user acceptability testing, system, integration, and unit testing.

8. Project Management and Oversight:

- **Establish Governance Structure:** To guarantee accountability, decision-making authority, and adherence to project policies and procedures, specify the roles and duties of the project governance structure.
- **Establish change control procedures** to efficiently handle modifications to the project's requirements, scope, schedule, and resources. Before putting changes into effect, document change requests, evaluate their implications, and get the go-ahead from pertinent parties.

The project 'Ayu' can create a solid basis for successful project execution by adhering to a methodical and structured approach to project planning within the SDLC. This will allow the development team to produce a high-caliber, user-friendly, and scalable healthcare management system that satisfies stakeholders' needs and expectations.

4.1.1 Estimated time table

Job Roles	Estimated Time
UI/UX engineers	15 / 05 / 2024 to 29 / 05 / 2024
Frontend developers	29 / 05 / 2024 to 16 / 07 / 2024
Backend developers	05 / 06 / 2024 to 20 / 07 / 2024
Full Stack Developer	29 / 05 / 2024 to 20 / 07 / 2024
QA	16 / 07 / 2024 to 24 / 07 / 2024
Estimated Total Time	03 Months

4.1.2 Economic

Economic feasibility analysis is the process of determining whether a new venture is worth the cost and time investment. It is also known as cost-benefit analysis. This kind of analysis takes into consideration the cost of both developing and operating the new venture.

NFC card cost	Rs. 300
NFC Reader cost	Rs. 2000
Profit cost	Rs. -
Tax cost	Rs. -
Total cost project	Rs. -

4.2 Defining Requirements

Collecting, evaluating, and recording the functional and non-functional requirements of the Patient Detail Management System (Ayu) project are the tasks of the Defining Requirements phase of the Software Development Life Cycle (SDLC). This stage is essential for determining the project's scope, identifying the needs of stakeholders, and setting specific development goals. An extensive synopsis of the Defining Requirements stage is provided below:

1. Stakeholder Overview:

- Determine who the main players in the project Ayu are, such as the developers, testers, sponsors, patients, and regulatory agencies.
- Establish each stakeholder's obligations and roles, as well as their expectations for the project.

2. Gaining Requirements:

- Gather needs for the Ayu platform by conducting focus groups, surveys, interviews, and workshops with stakeholders.
- Determine which business processes, user needs, system features, and legal criteria the Ayu system must satisfy.
- To gather thorough needs, use a variety of elicitation techniques, including use case analysis, brainstorming, and prototyping.

3. Analysis of Requirements:

- Examine the collected requirements to find overlaps, inconsistencies, dependencies, and limitations.
- Assign multiple categories to needs, such as non-functional requirements (how the system should operate) and functional requirements (what the system should do).
- Sort needs according to significance, practicality, and effect on project objectives.

4. Requirement Documentation:

- Use cases, functional requirement documents, or requirement specification templates to organize needs into a systematic fashion.
- Each requirement should be precisely defined, together with its dependencies, acceptance criteria, traceability to stakeholder demands, and project goals.
- To make it easier for stakeholders to evaluate, analyze, and validate requirements, organize them into manageable portions or modules.

5. Review and Validation of Requirements:

- To make sure that requirements appropriately represent stakeholders' needs and expectations, conduct reviews and validation sessions with them.
- During the review process, any ambiguities, inconsistencies, or holes should be addressed through conversations, clarifications, and modifications.
- Obtain formal stakeholder consent before moving on to the next stage of the SDLC.

Through a methodical approach to requirement definition within the software development life cycle (SDLC), the project Ayu may guarantee that the final system satisfies stakeholder needs and expectations, is in line with project goals, and establishes a strong basis for successful development and deployment.

4.3 Designing

In this stage, the requirements are translated into a software construction plan.

- **Interface Design:** Create accessible and user-friendly user interfaces. This covers how the screens are arranged and how users move between various interfaces.
- **Database Design:** Create a schema for your database that is capable of effectively handling complicated data and relationships. Ensuring data integrity and safe access is essential for medical records.

- System design: Taking maintenance and scalability into account, choose between a solid, service-oriented architecture or hybrid design based on needs. Given the scalability and reliability requirements of the project, a microservices architecture may be suitable.

4.4 Development

During this stage of development, programmers create code and construct the system in accordance with the design specifications.

- Coding: To create an application, developers utilize tools and programming languages selected in the design stage. To guarantee code quality and maintainability, it is essential to adhere to coding standards and best practices.
- Production, staging, and development environments should be set up. This facilitates the management of the various testing and deployment phases.

4.5 Testing

Thorough testing is necessary to guarantee the system's dependability, security, and usability.

- Unit testing: Verify that each component behaves correctly on its own.
- Verify through integration testing that the various components of the application function as intended.
- System testing: Check that the integrated system as a whole satisfies all requirements by testing it thoroughly.
- User Acceptance Testing (UAT) verifies that the system satisfies their criteria by having actual users evaluate it in a real-world setting.

4.6 Deployment & Maintenance

This entails putting the system in a production setting so that real users can utilize it. The system requires continuous updates and support after deployment.

- First deployment: To get early input, a trial rollout to a small audience is frequently used.
- Complete-scale Deployment: The system is made available to all intended users following the receipt of first feedback and any required modifications.
- Frequent Updates: Include security fixes, new features, and updates.

- Performance Monitoring: Keep an eye on the functioning of the system at all times to address any possible problems early on.
- Feedback Loop: Establish a methodical procedure for consumers to offer input, which is then utilized to enhance and modify the system.

This thorough understanding of the SDLC will guarantee that the system is created in an organized and effective way, resulting in a reliable and user-friendly final product.

5.0 System Design Methodology

There are several system design methodologies such as waterfall incremental, iterative, agile, scrum, extreme programming, etc. Here we have selected the agile methodology due to below reasons;

- Transparency – By using the agile methodology system complexity can be reduced.
- Collaborative Effort - Clients are obligated to actively participate in the requirements definition process, thus reducing unknowns and making the overall process more efficient.
- Project size is medium.
- The clients are available throughout the process.
- Clients' requirements are not stable.

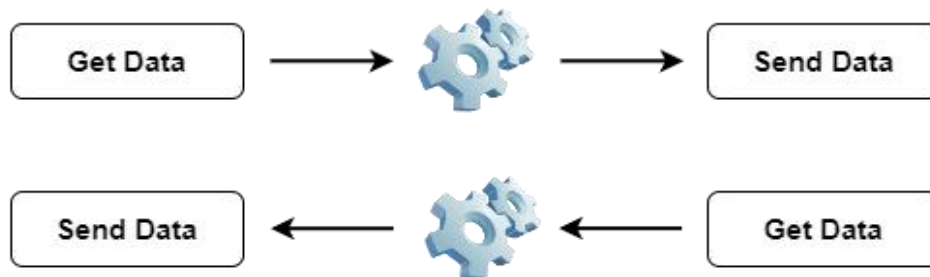
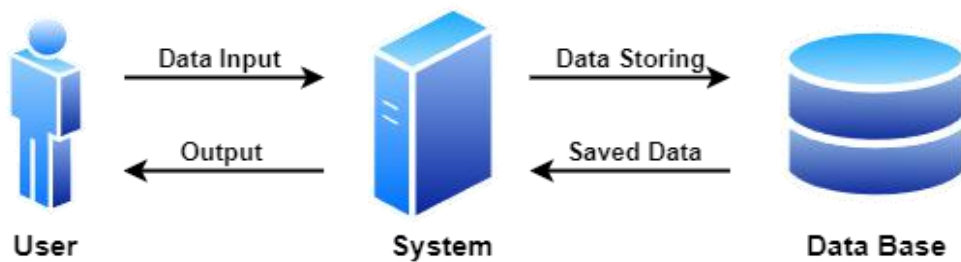
6.0 Design Diagrams

The following diagrams were designed based on the functional requirements.

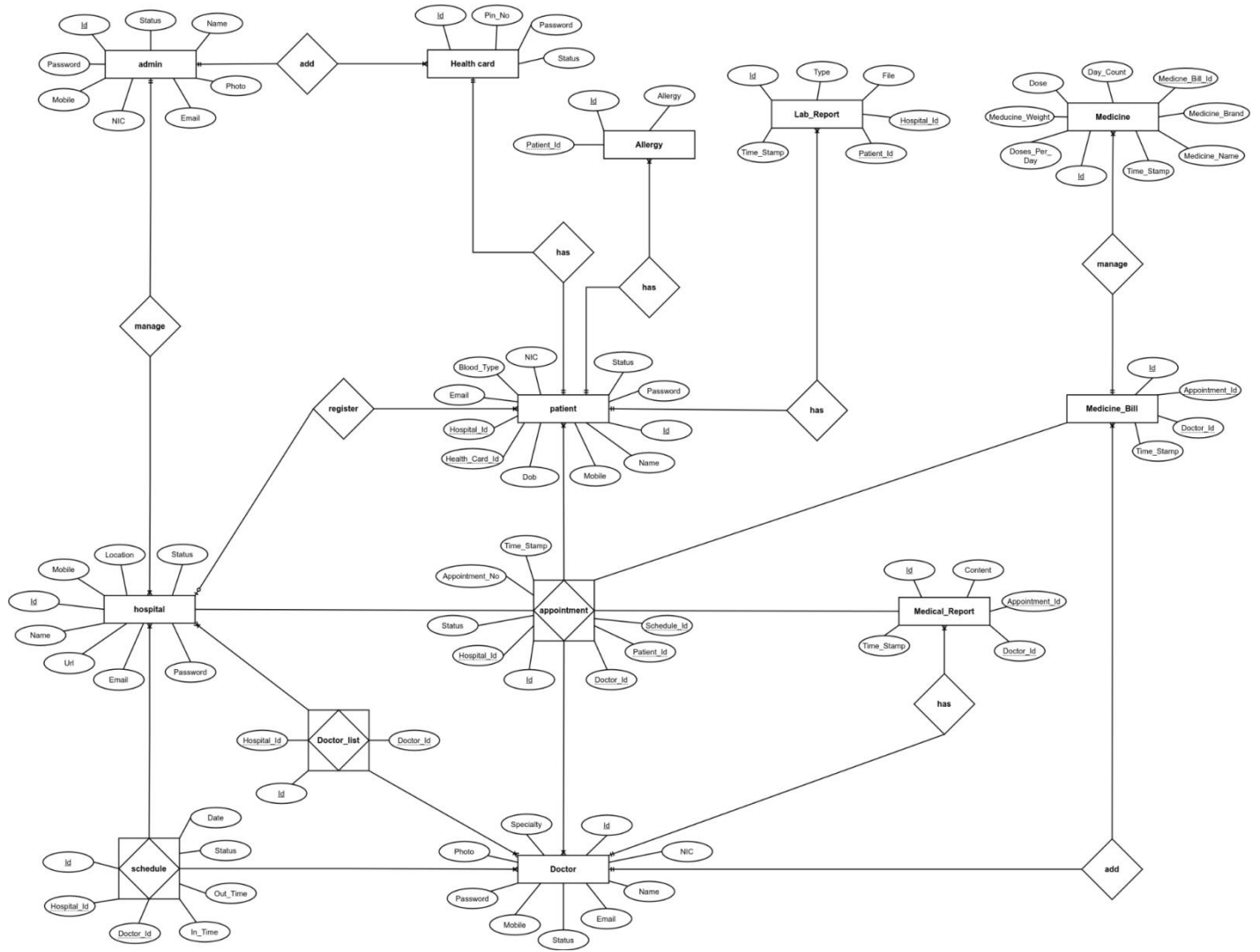
- Use Case Diagram and respective use case descriptions
- Entity Relationship Diagram
- User Interface Design

In this chapter use case diagrams, Entity relationship diagrams, database design, and a few user interfaces are added.

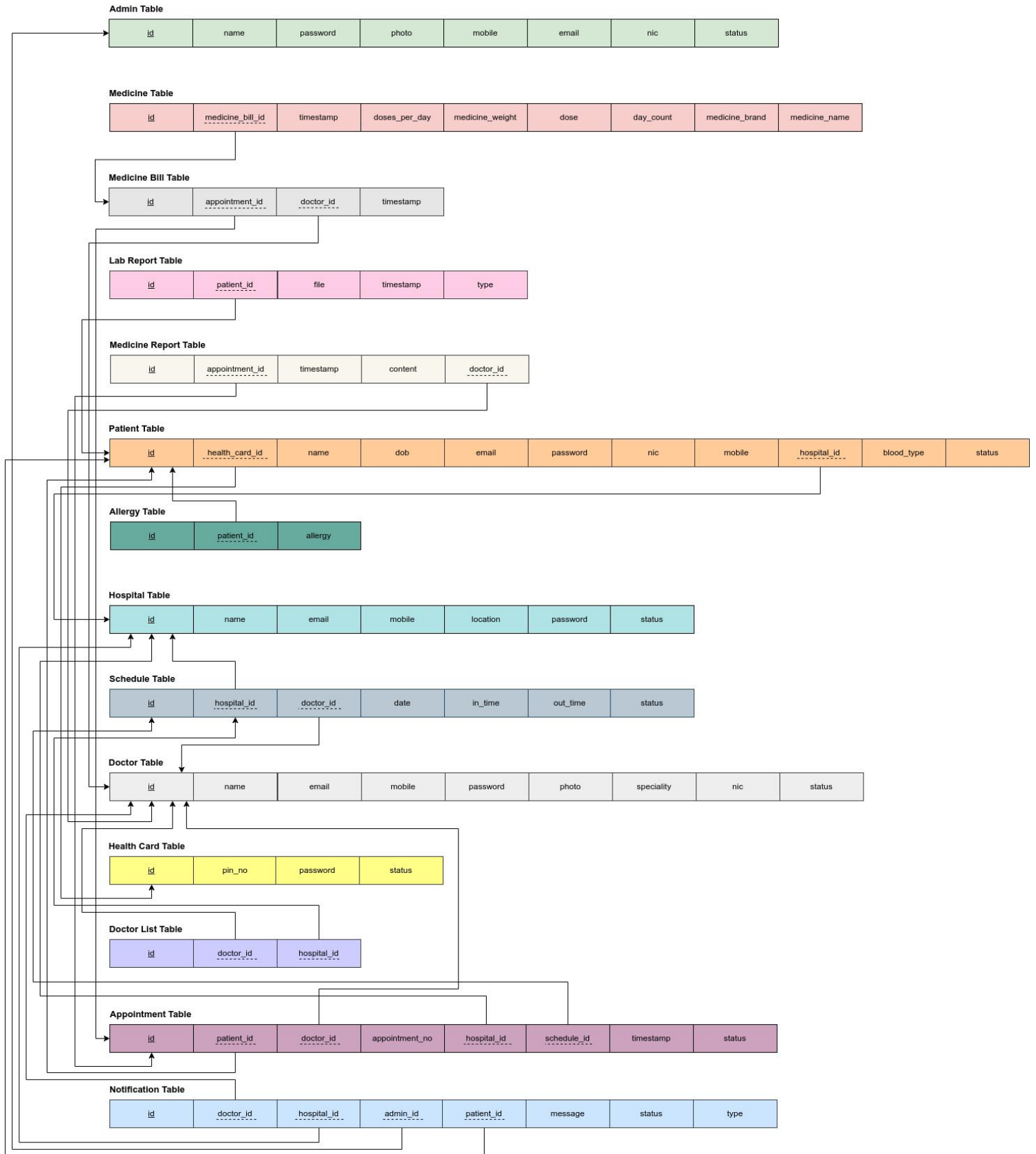
6.1 Architecture



6.2 Entity Relationship Diagram



6.3 Relational Schema



6.4 Table Model

Admin Table

<u>id</u>	name	password	photo	mobile	email	nic	status
-----------	------	----------	-------	--------	-------	-----	--------

Medicine Table

<u>id</u>	<u>medicine_bill_id</u>	timestamp	doses_per_day	medicine_weight	dose	day_count	medicine_brand	medicine_name
-----------	-------------------------	-----------	---------------	-----------------	------	-----------	----------------	---------------

Medicine Bill Table

<u>id</u>	<u>appointment_id</u>	<u>doctor_id</u>	timestamp
-----------	-----------------------	------------------	-----------

Lab Report Table

<u>id</u>	<u>patient_id</u>	file	timestamp	type
-----------	-------------------	------	-----------	------

Medicine Report Table

<u>id</u>	<u>appointment_id</u>	timestamp	content	<u>doctor_id</u>
-----------	-----------------------	-----------	---------	------------------

Patient Table

<u>id</u>	<u>health_card_id</u>	name	dob	email	password	nic	mobile	<u>hospital_id</u>	blood_type	status
-----------	-----------------------	------	-----	-------	----------	-----	--------	--------------------	------------	--------

Allergy Table

<u>id</u>	<u>patient_id</u>	allergy
-----------	-------------------	---------

Hospital Table

<u>id</u>	name	email	mobile	location	password	status
-----------	------	-------	--------	----------	----------	--------

Schedule Table

<u>id</u>	<u>hospital_id</u>	<u>doctor_id</u>	date	in_time	out_time	status
-----------	--------------------	------------------	------	---------	----------	--------

Doctor Table

<u>id</u>	name	email	mobile	password	photo	speciality	nic	status
-----------	------	-------	--------	----------	-------	------------	-----	--------

Health Card Table

<u>id</u>	pin_no	password	status
-----------	--------	----------	--------

Doctor List Table

<u>id</u>	<u>doctor_id</u>	<u>hospital_id</u>
-----------	------------------	--------------------

Appointment Table

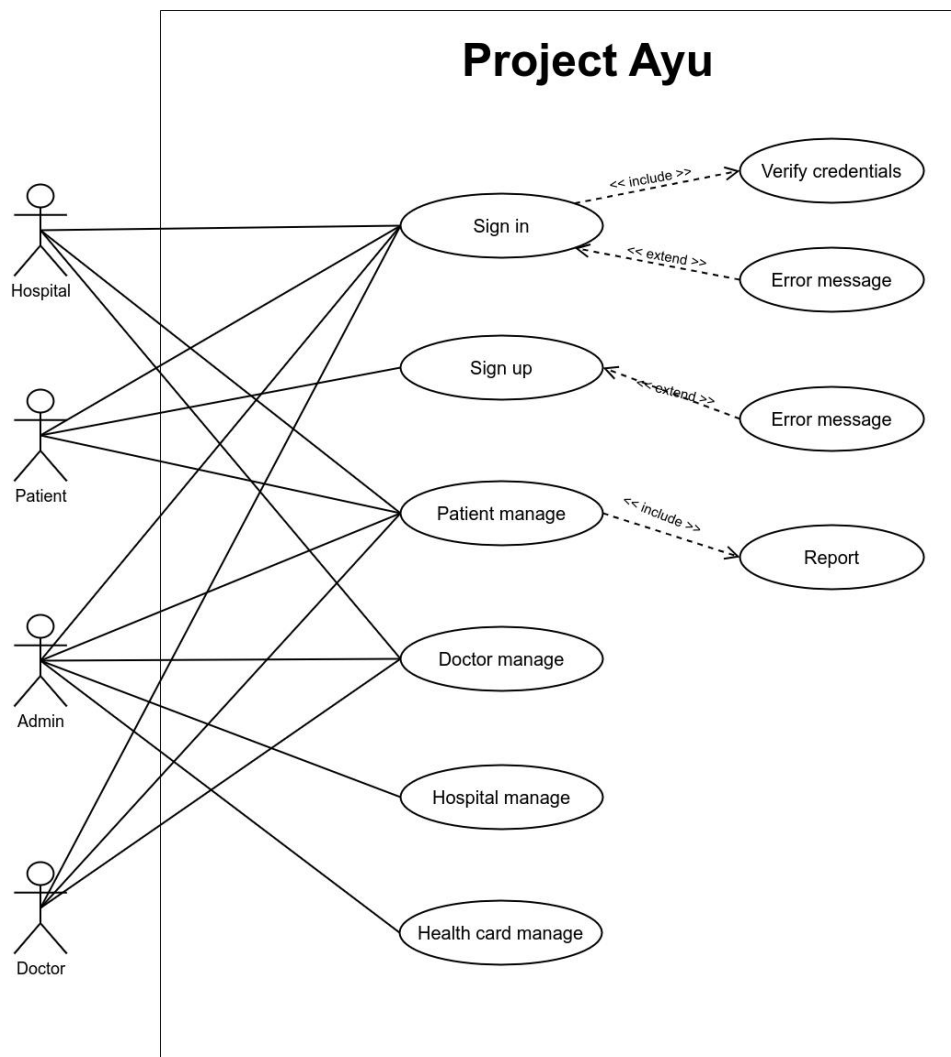
id	patient_id	doctor_id	appointment_no	hospital_id	schedule_id	timestamp	status
----	------------	-----------	----------------	-------------	-------------	-----------	--------

Notification
Table

id	doctor_id	hospital_id	admin_id	patient_id	message	status	type
----	-----------	-------------	----------	------------	---------	--------	------

6.5 Use Case Diagram

The user case diagram is used to demonstrate the user interactions with the system. These highlight scenarios of the systems, goals that the system helps actors to achieve, and the scope of the system.



6.6 Natural Language Sentences

Priority Number	01
Function Name	Hospital Registration
Description	Login to System
Input	Input username and password
Process	Verify relevant credentials
Output	Get Access to the System

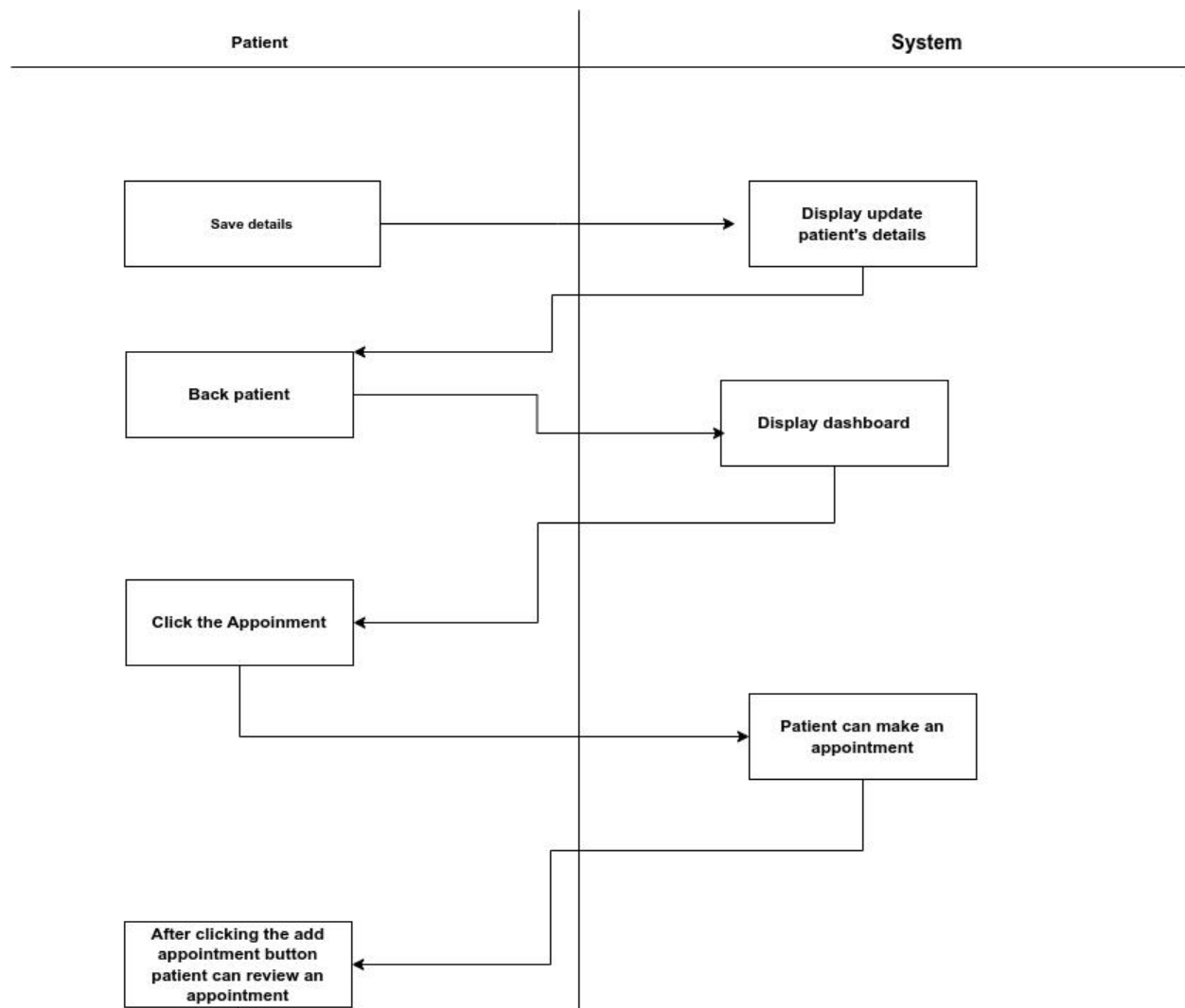
Priority Number	02
Function Name	Doctor Registration
Description	Initially the hospital should register the Doctor
Input	Login to System
Process	Add medical info after scanning patients nfc card
Output	Save the required information, then create a new Patient medical record

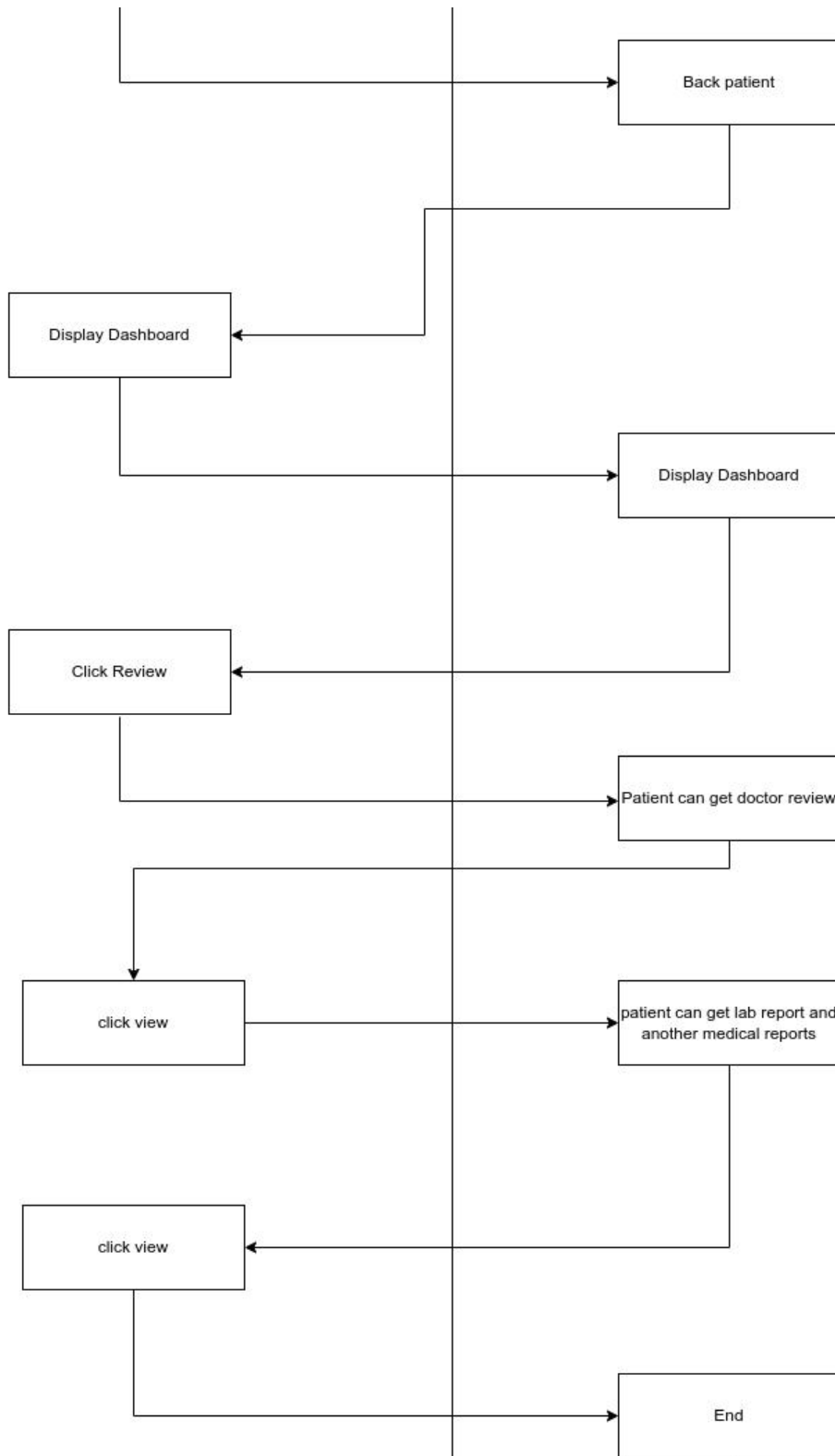
Priority Number	03
Function Name	Patient Registration

Description	Initially the user should register the Patient
Input	Enter the patient's personal details
Process	Application asks for the patient's details
Output	Save the required information, then create a new Patient registration record

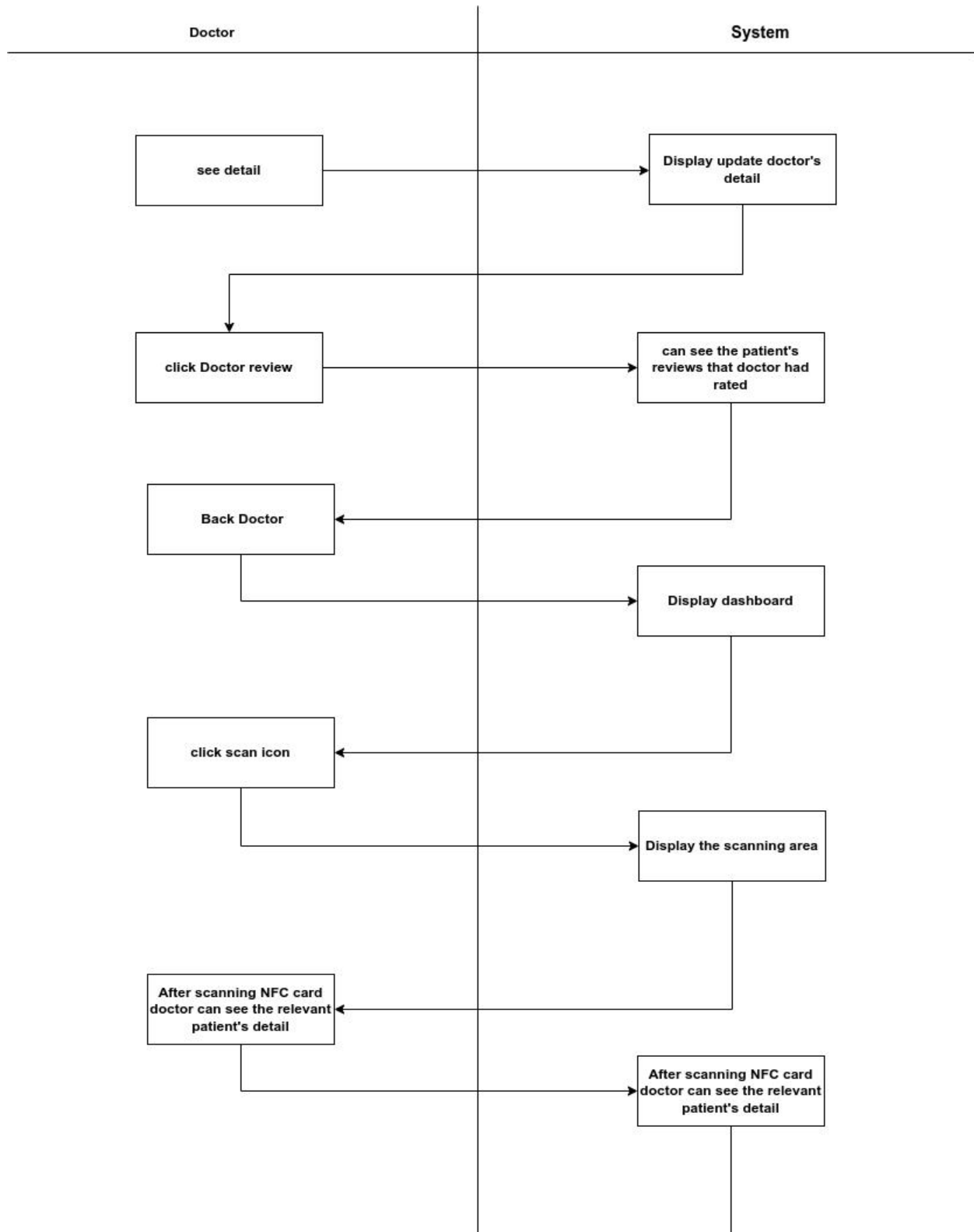
6.7 Activity Diagram

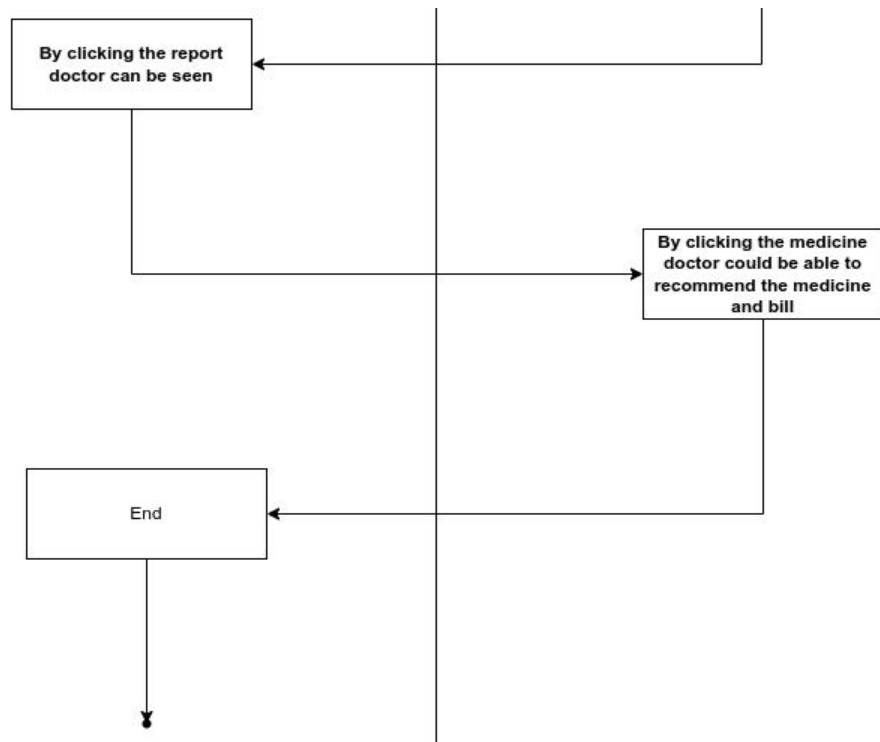
Patient Flow



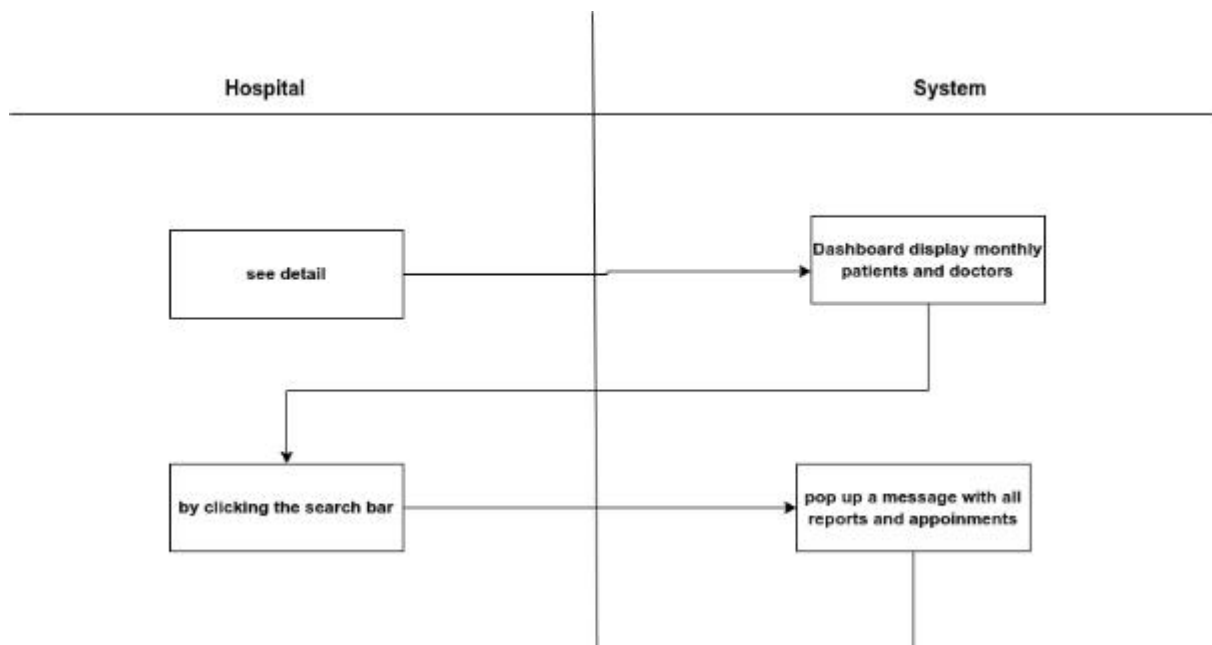


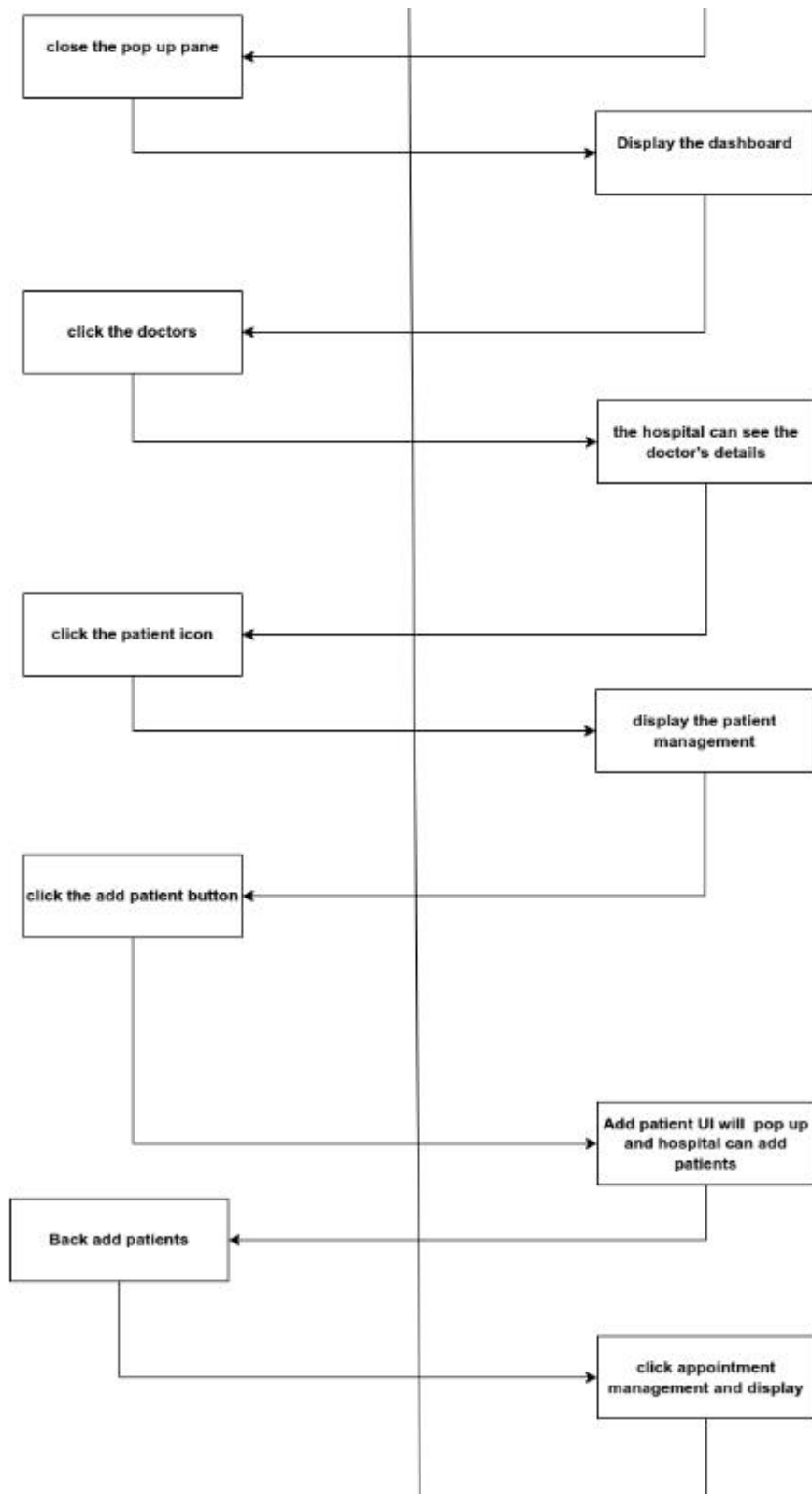
Doctor Flow

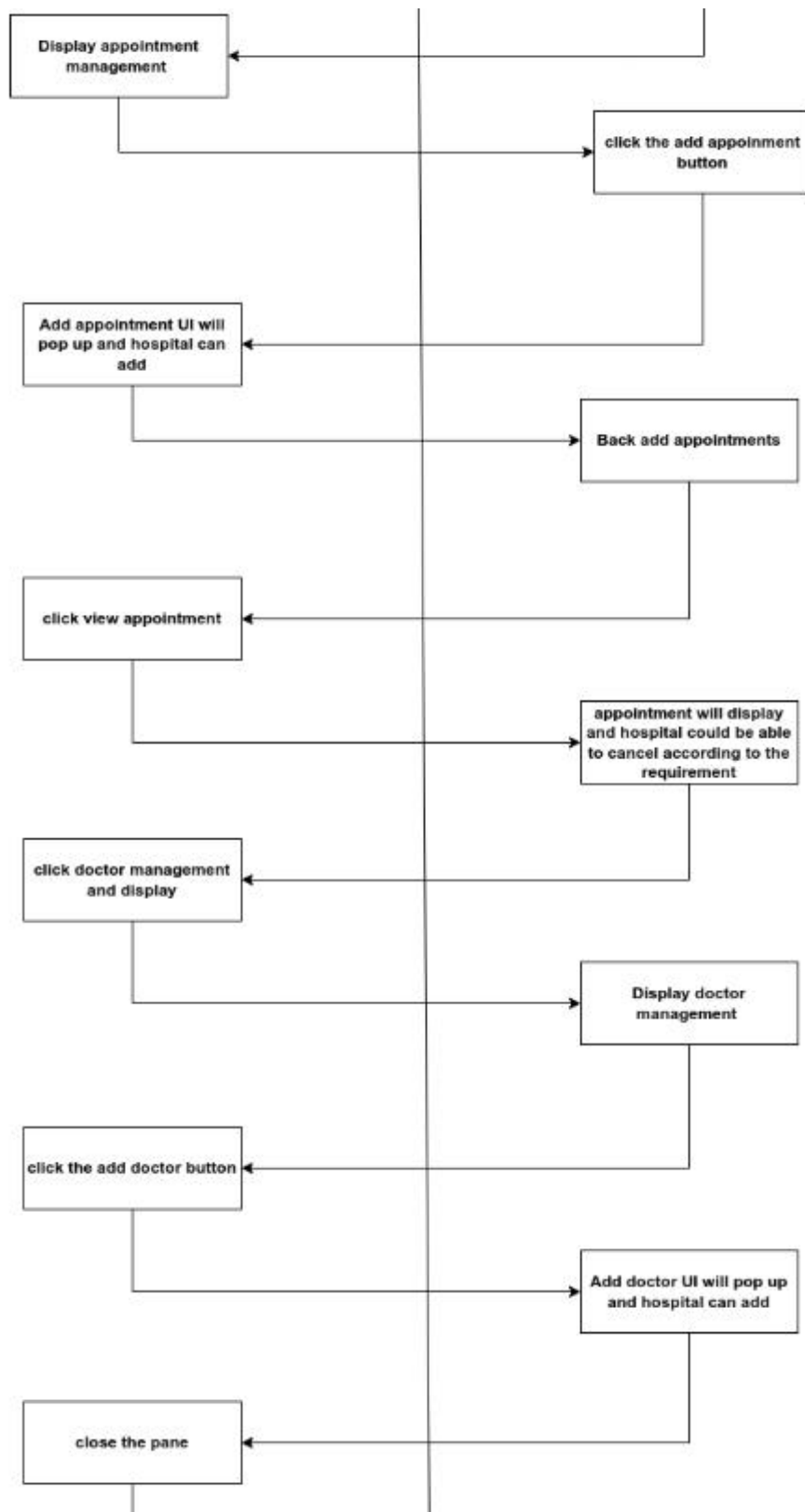


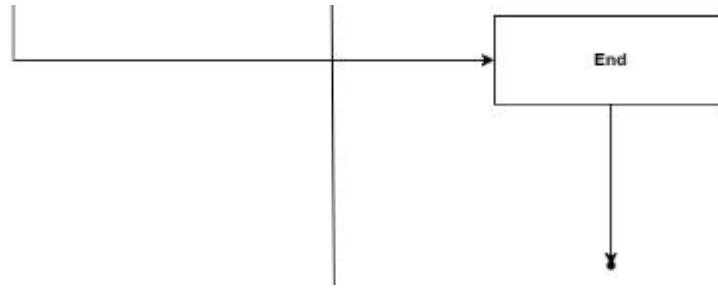


Hospital Flow

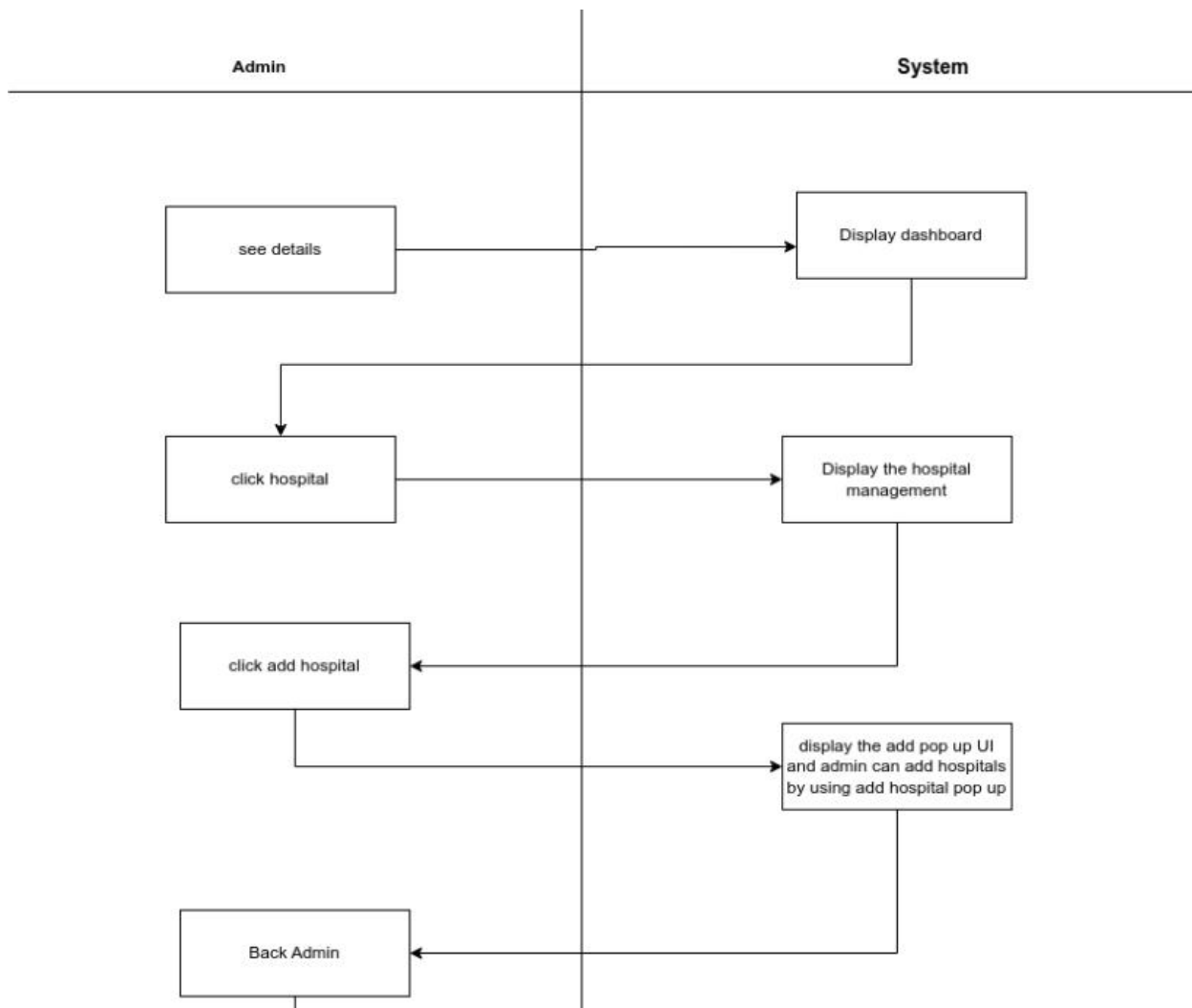


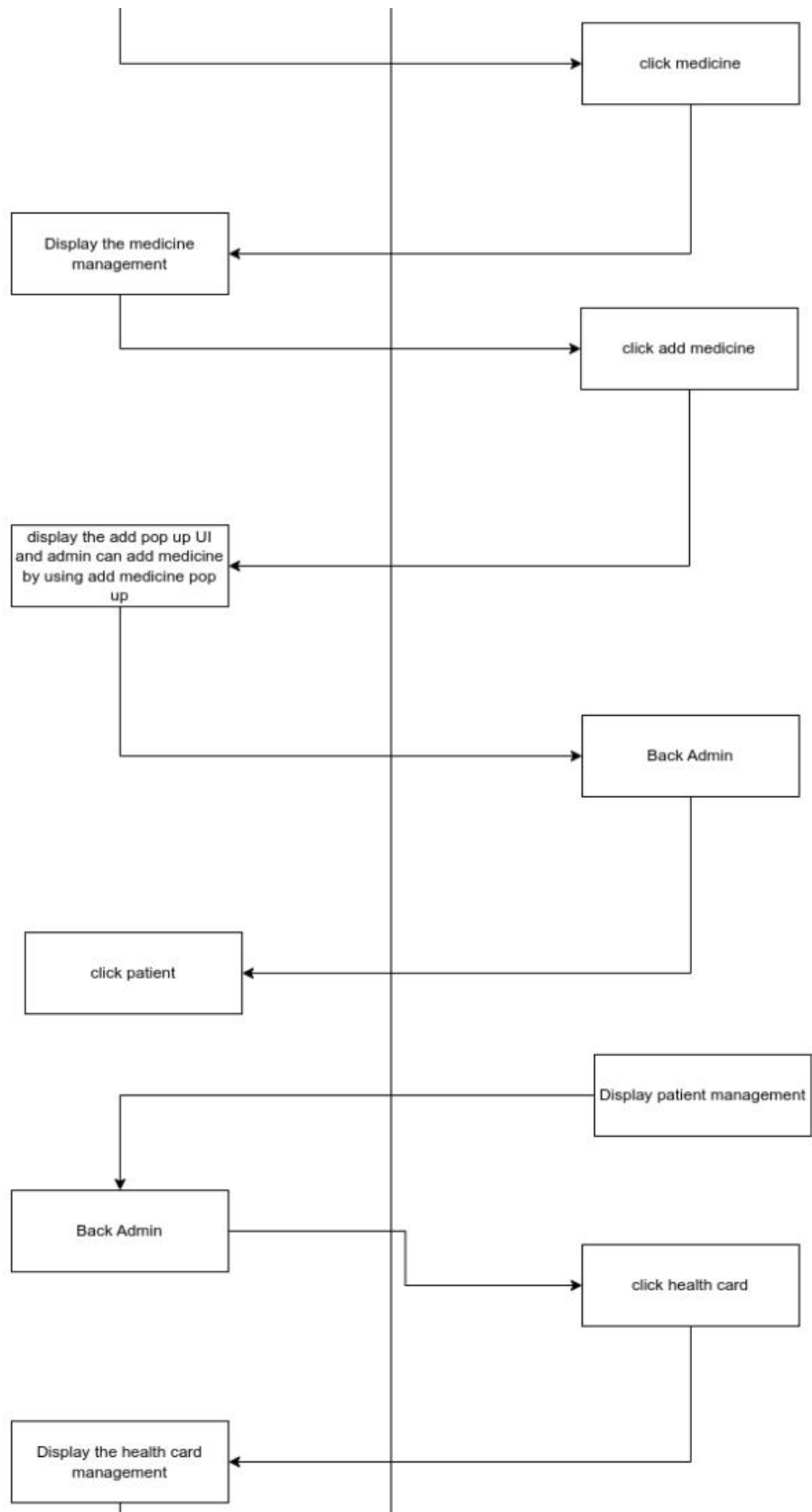


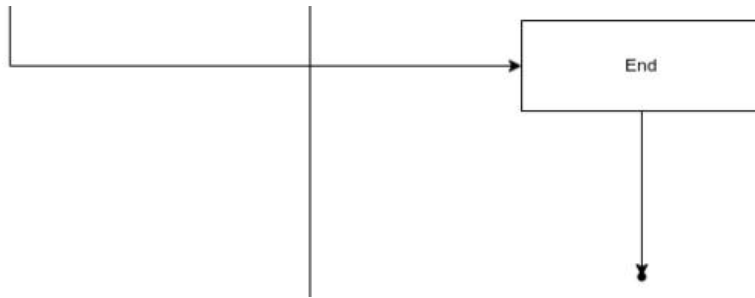




Admin Flow

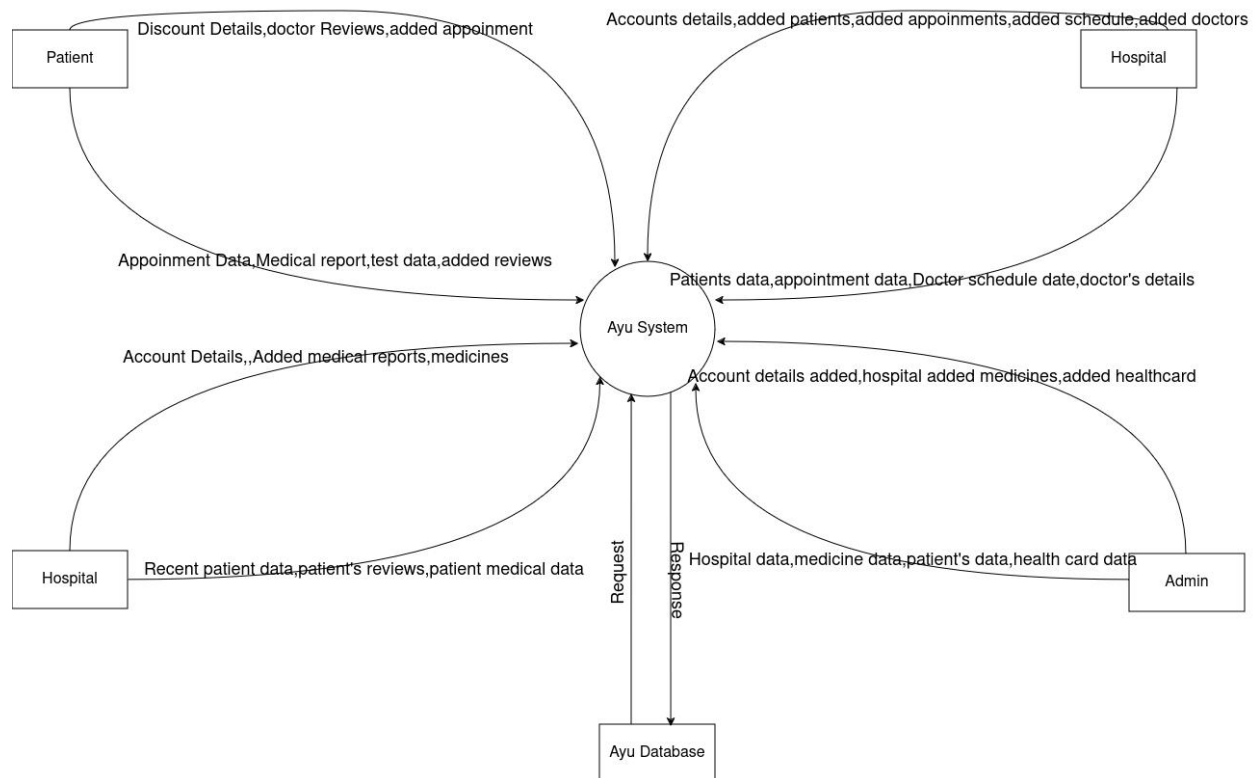




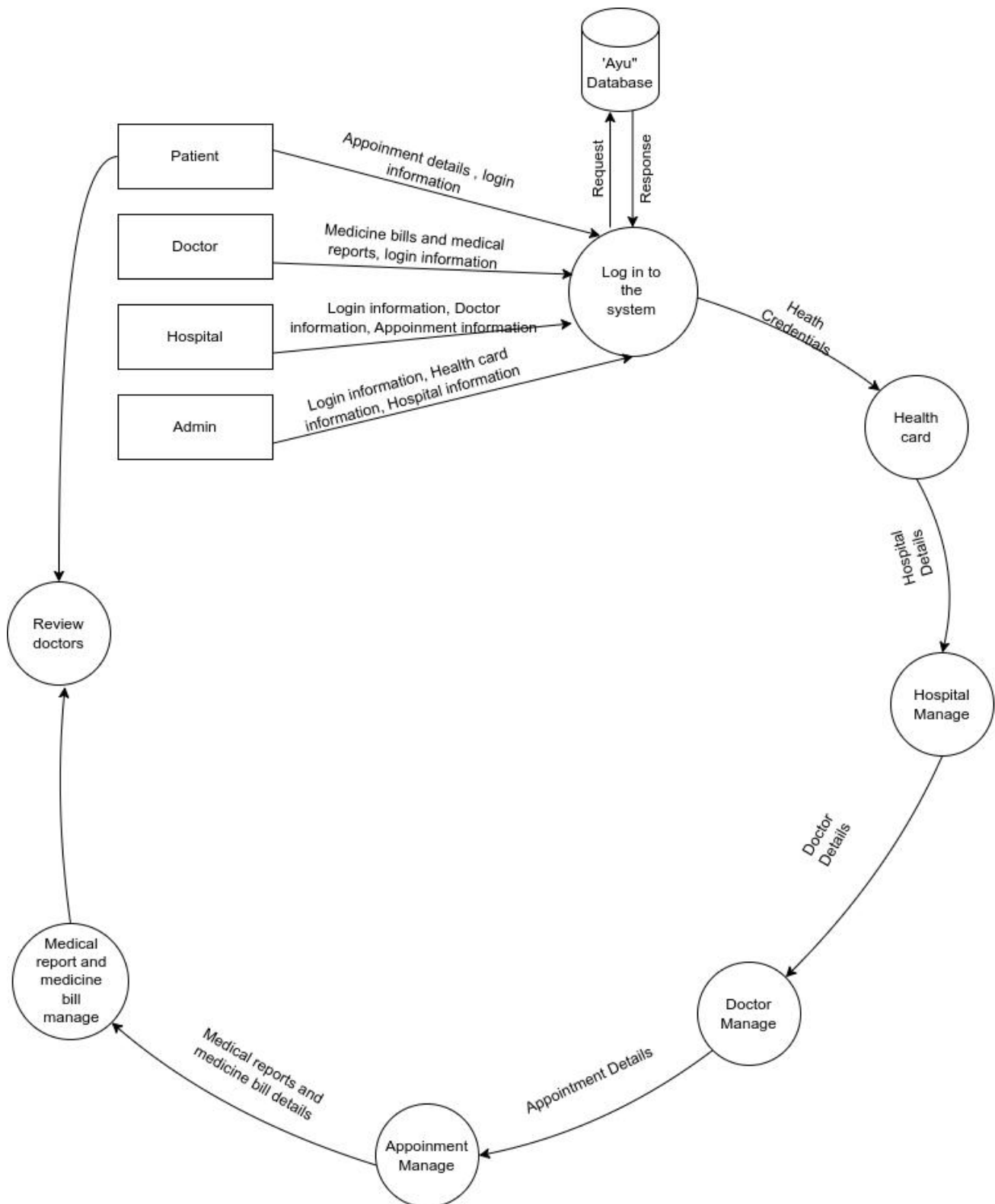


6.8 Data Flow Diagram

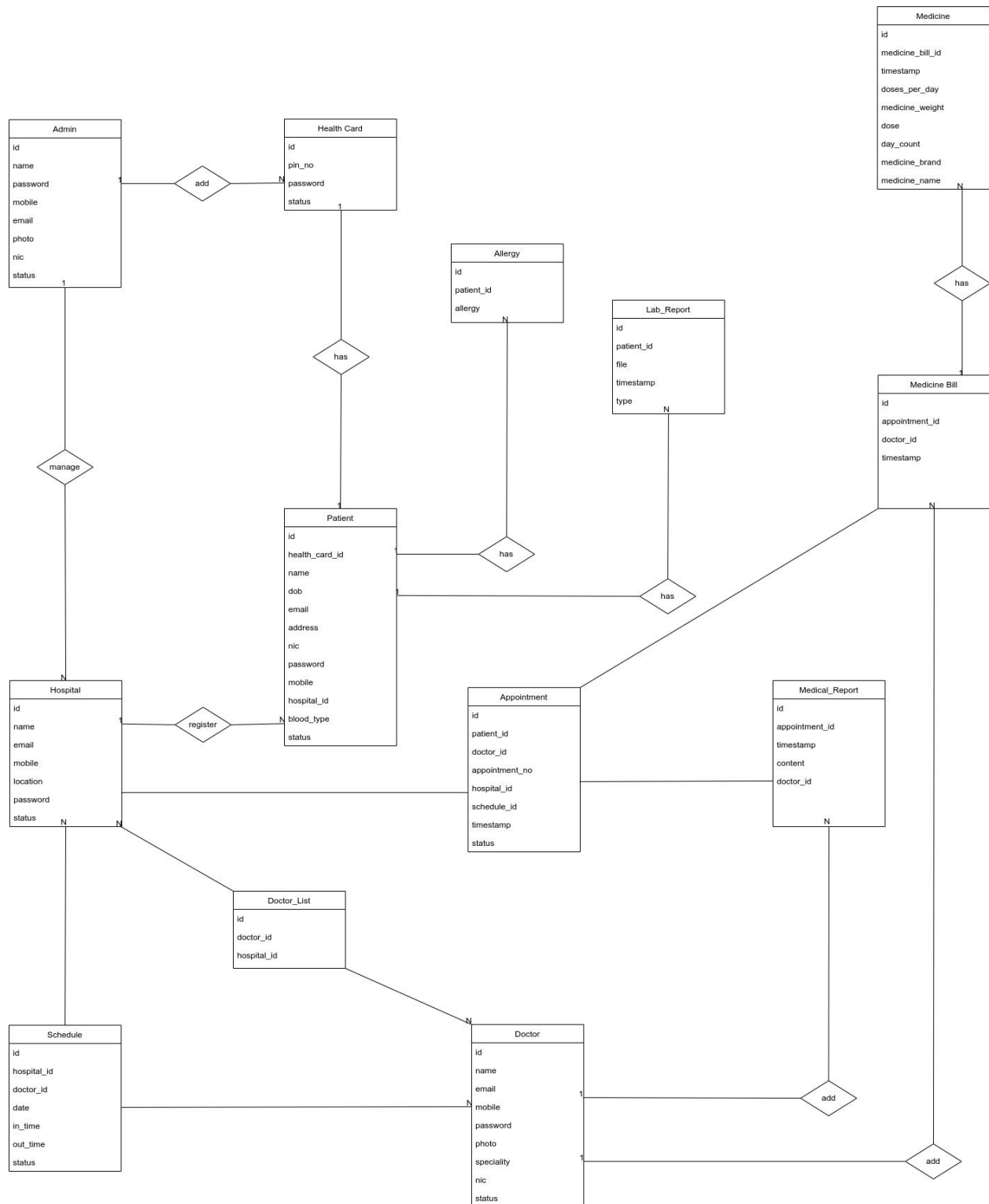
Level 01



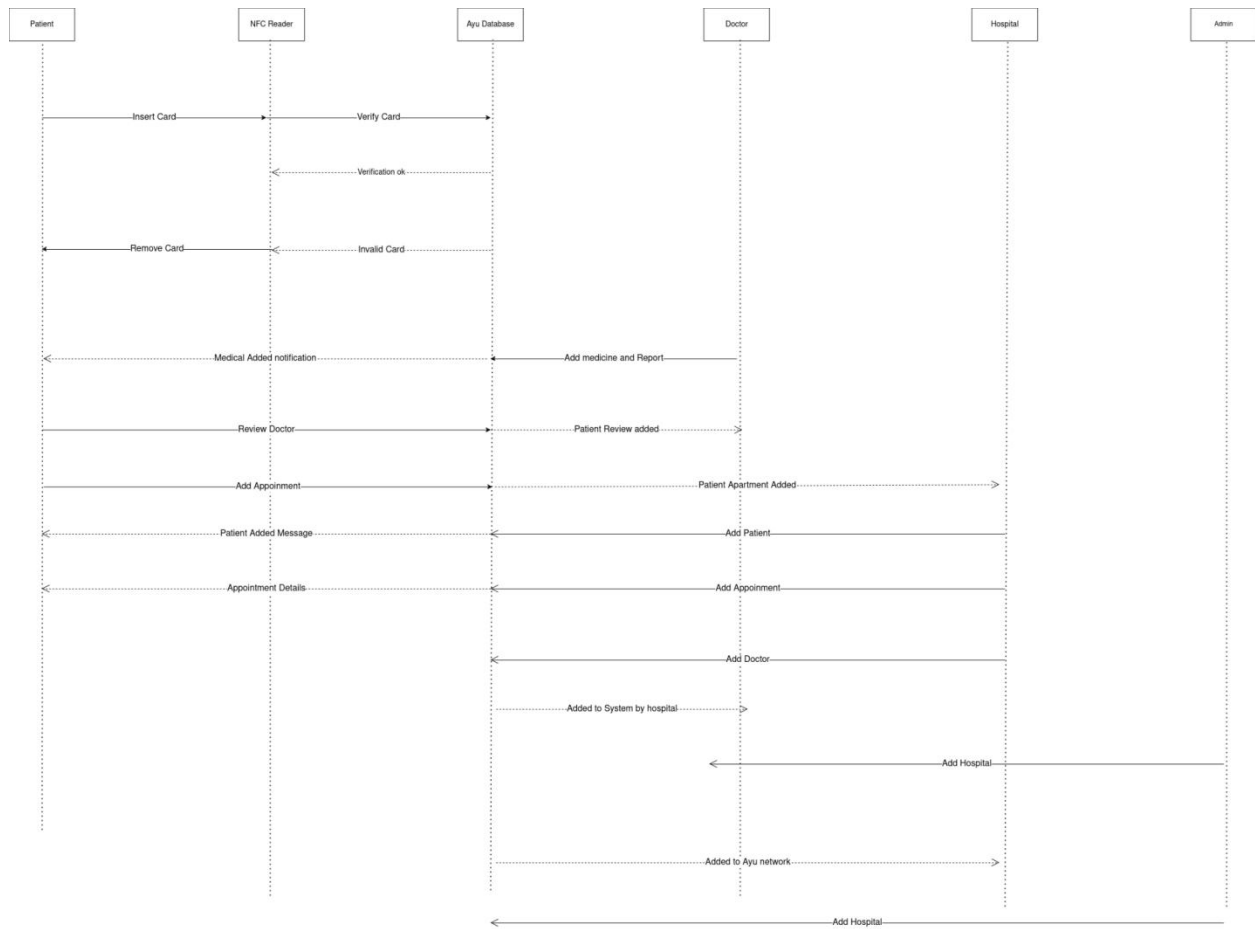
Level 02



6.9 Class Diagram



6.10 Sequence Diagram



07. User Interface

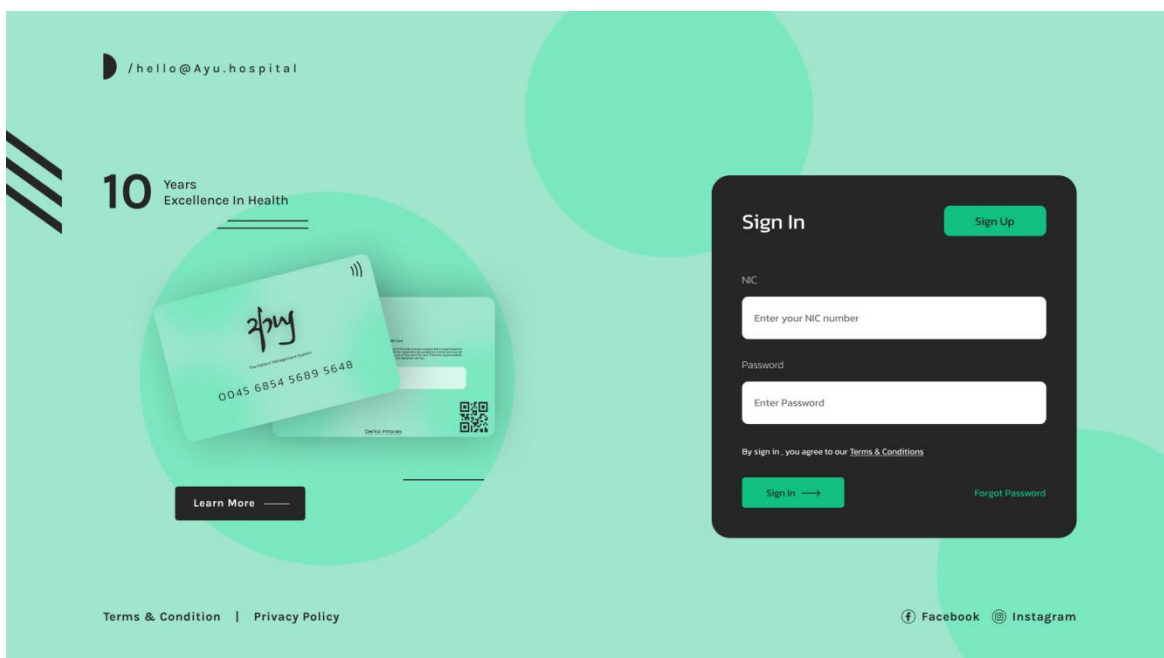
User interfaces are the access points where users interact with designs. Following Are some interface designs specifically designed for this system.

For all the user interfaces have been designed as a combination of green, white, and black colors. All the form formats have been designed with more usability and all minimum required details have been added as fields. Most of the data has been loaded by drop-down, then it will be easy for the user to select.

Also, all the mandatory fields are marked. For all the validation breaks proper error messages will be promoted. Also With The header navigation part, the user can navigate to any page at any time. For the SRS Process, we used draw.io software to build wireframes to build prototypes. For UI/UX design we used Figma as our design software.

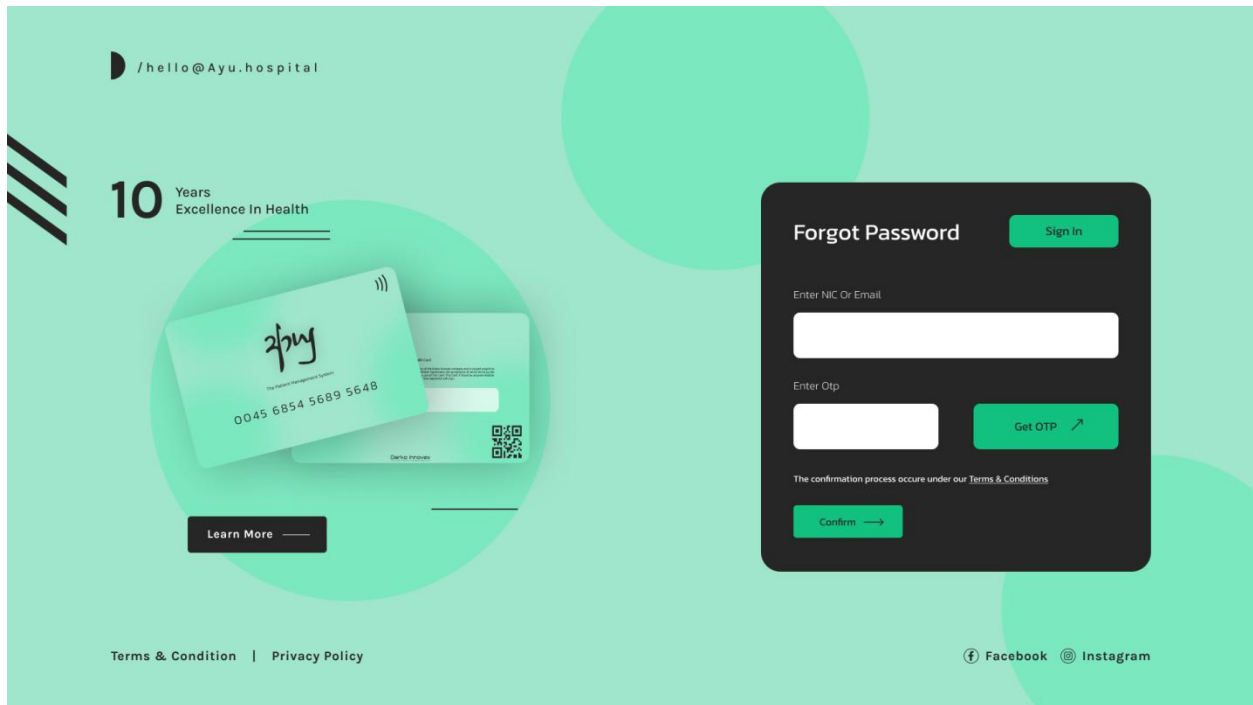
7.1 Sign In Page

In the sign in page, users have to input NIC or email and the password. After that, software checks whether it is the admin, patient, hospital, and doctor because there are separate dashboards for each of them. For Invalid user login details, proper validation messages will be prompted.



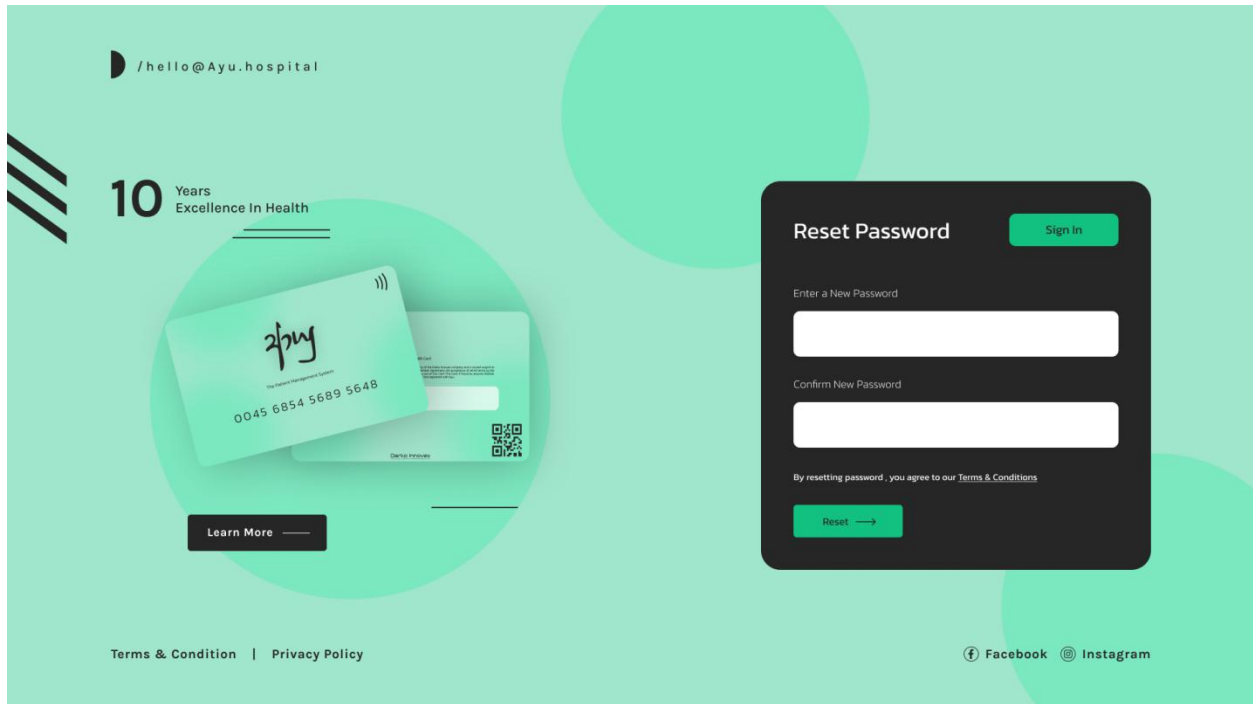
7.2 Forgot Password Page

The "Forgot Password" page is a feature on websites or applications that allows users who have forgotten their password to reset it and regain access to their account.



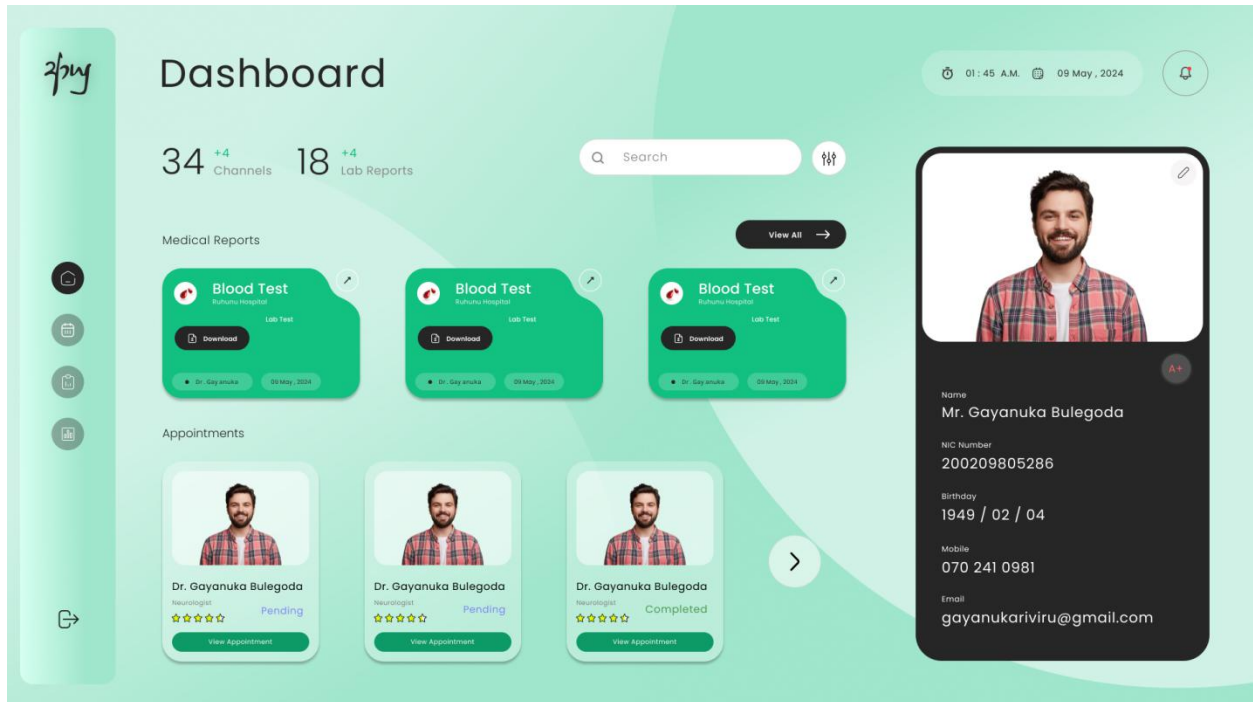
7.3 Reset Password Page

The "Reset Password" page is a crucial feature on websites or applications that enables users to securely update their password.



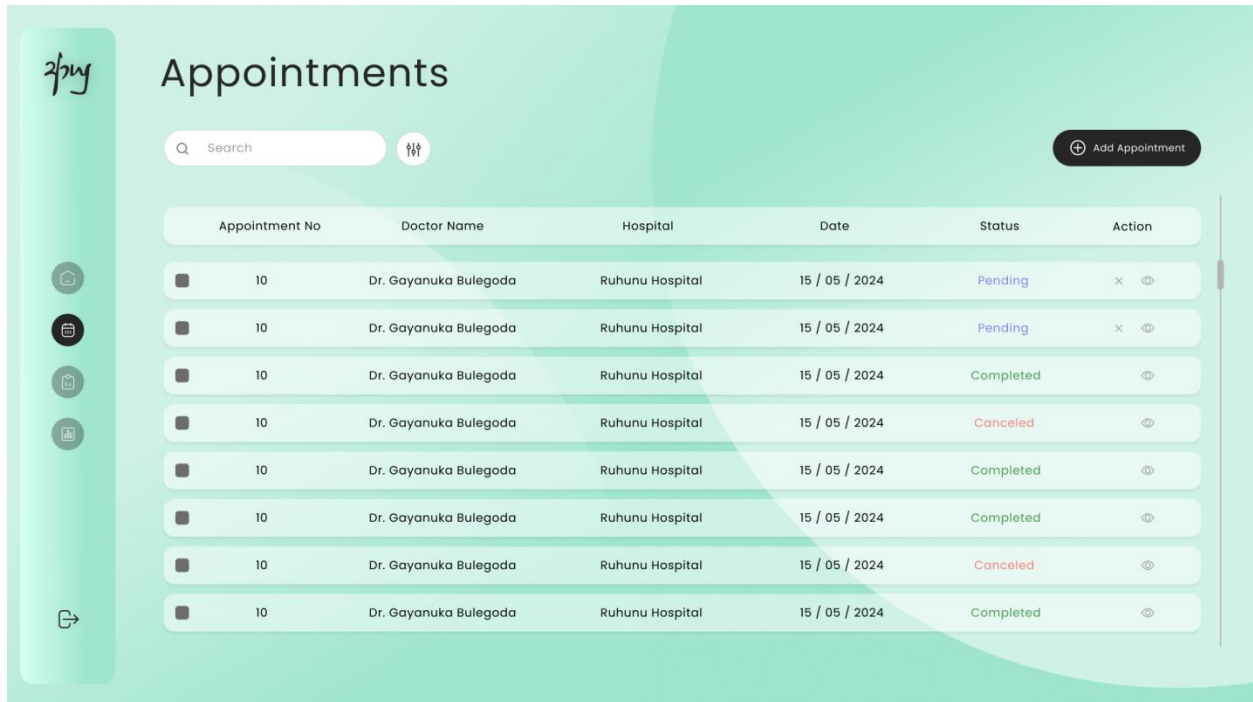
7.4 Patient's Dashboard Page

After sign up processing, the main page will be displayed. The patient's dashboard consists of the following options.



7.5 Patient's Appointments Page

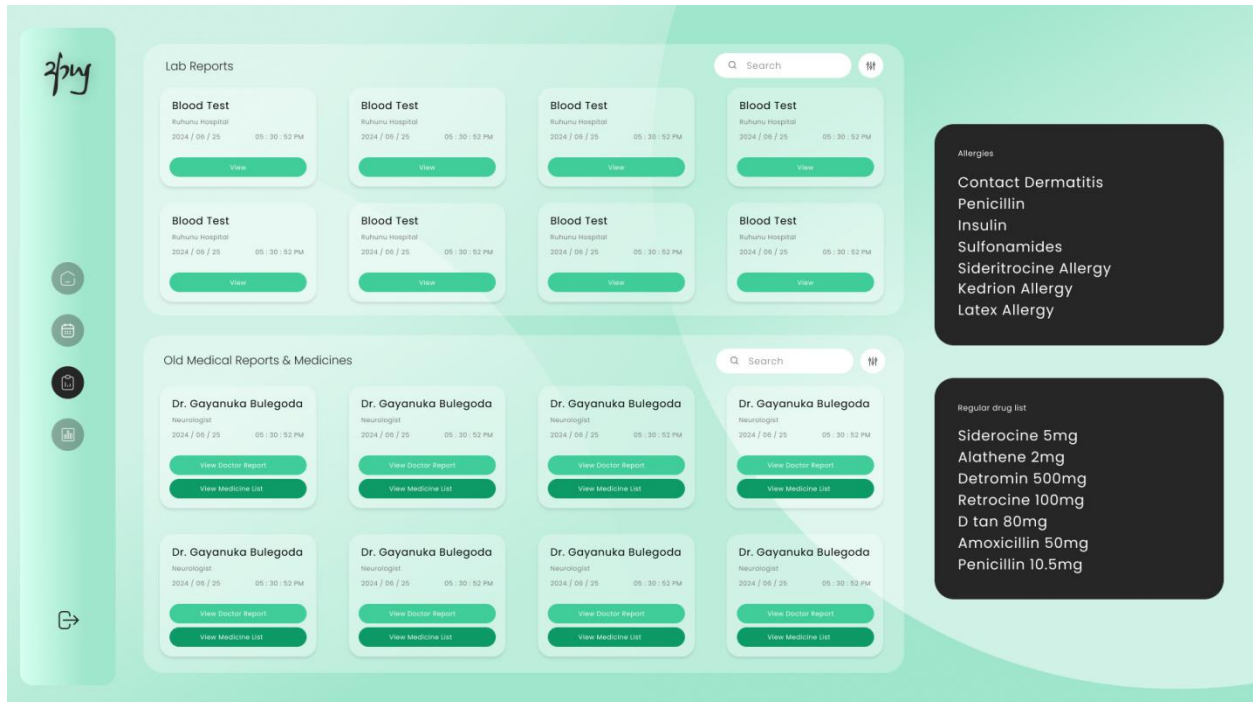
The patient's appointment page consists of the appointment number, doctor name, hospital, date and status.



Appointment No	Doctor Name	Hospital	Date	Status	Action
10	Dr. Gayanuka Bulegoda	Ruhunu Hospital	15 / 05 / 2024	Pending	× 👁
10	Dr. Gayanuka Bulegoda	Ruhunu Hospital	15 / 05 / 2024	Pending	× 👁
10	Dr. Gayanuka Bulegoda	Ruhunu Hospital	15 / 05 / 2024	Completed	👁
10	Dr. Gayanuka Bulegoda	Ruhunu Hospital	15 / 05 / 2024	Canceled	👁
10	Dr. Gayanuka Bulegoda	Ruhunu Hospital	15 / 05 / 2024	Completed	👁
10	Dr. Gayanuka Bulegoda	Ruhunu Hospital	15 / 05 / 2024	Completed	👁
10	Dr. Gayanuka Bulegoda	Ruhunu Hospital	15 / 05 / 2024	Canceled	👁
10	Dr. Gayanuka Bulegoda	Ruhunu Hospital	15 / 05 / 2024	Completed	👁

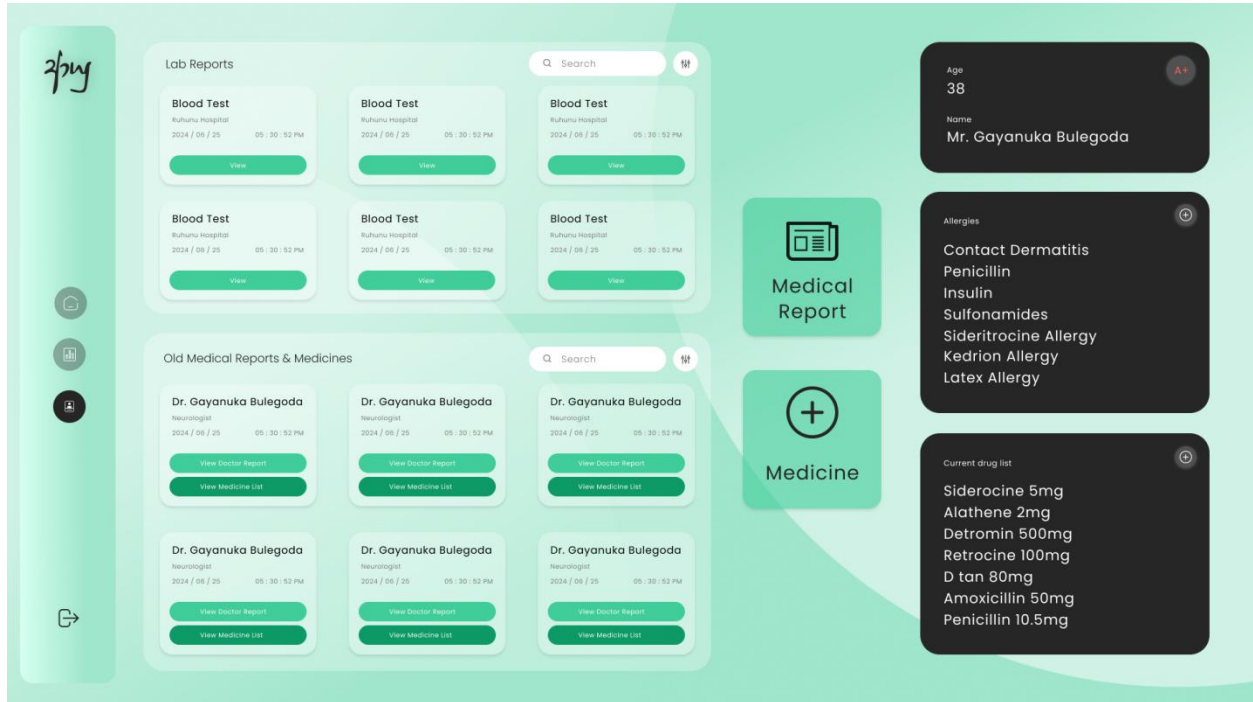
7.6 Patient's Medical Report Page

The patient's medical report page consists of lab reports, old medical reports, Medicines bills, allergies, and current medicine list.



7.7 Patient's Details Page (Doctor's)

The patient's details page displays all the medical details of a patient.



7.8 Hospital Patient Management Page

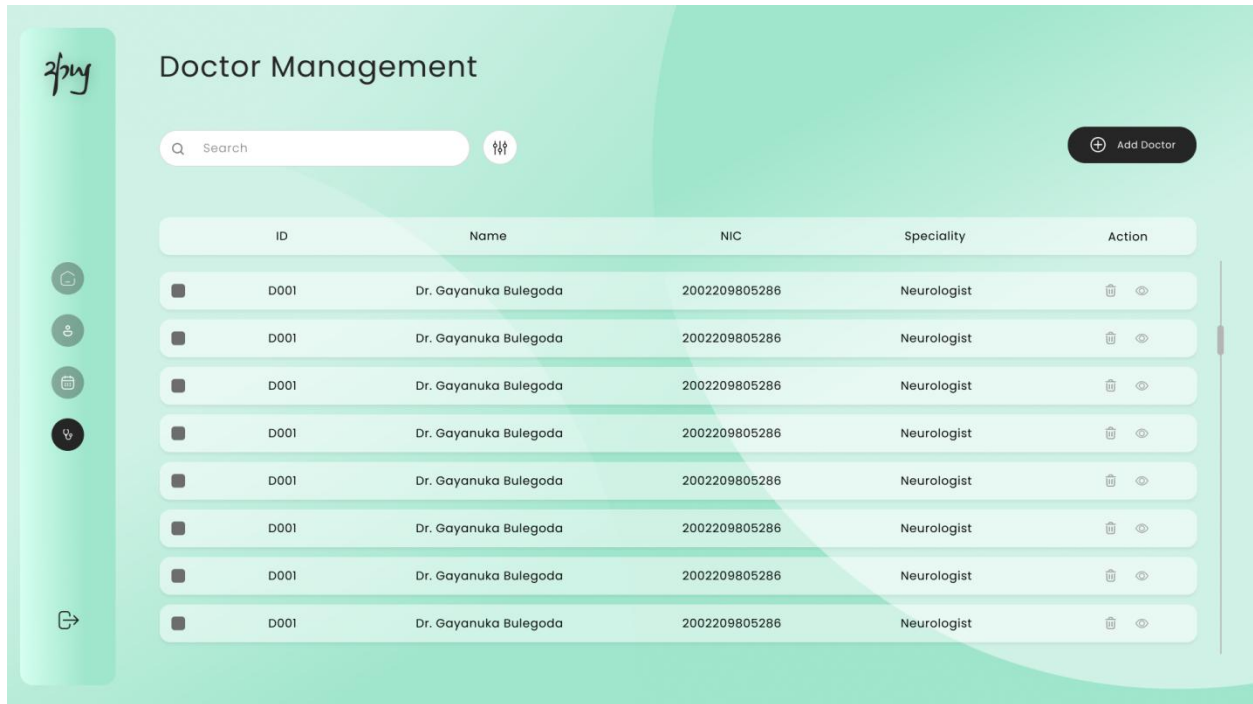
Patient Management

Search

















ID	Name	NIC	Index . No	Date	Time
P001	Mr. Gayanuka Bulegoda	2002209805286	1682 8512 6254 8563	2024 / 05 / 10	10 : 20
P001	Mr. Gayanuka Bulegoda	2002209805286	1682 8512 6254 8563	2024 / 05 / 10	10 : 20
P001	Mr. Gayanuka Bulegoda	2002209805286	1682 8512 6254 8563	2024 / 05 / 10	10 : 20
P001	Mr. Gayanuka Bulegoda	2002209805286	1682 8512 6254 8563	2024 / 05 / 10	10 : 20
P001	Mr. Gayanuka Bulegoda	2002209805286	1682 8512 6254 8563	2024 / 05 / 10	10 : 20
P001	Mr. Gayanuka Bulegoda	2002209805286	1682 8512 6254 8563	2024 / 05 / 10	10 : 20
P001	Mr. Gayanuka Bulegoda	2002209805286	1682 8512 6254 8563	2024 / 05 / 10	10 : 20
P001	Mr. Gayanuka Bulegoda	2002209805286	1682 8512 6254 8563	2024 / 05 / 10	10 : 20

7.9 Hospital Doctor Management Page

The Hospital Doctor Management Page displays the doctors who are currently working in the hospital.



The screenshot shows the 'Doctor Management' page. It includes a sidebar with icons for home, doctor, appointment, and user management. The main content area has a title 'Doctor Management', a search bar, and an 'Add Doctor' button. Below these is a table with columns for ID, Name, NIC, Speciality, and Action. The table contains 8 rows of data, all for Dr. Gayanuka Bulegoda, a Neurologist, with NIC 2002209805286.

ID	Name	NIC	Speciality	Action
D001	Dr. Gayanuka Bulegoda	2002209805286	Neurologist	 
D001	Dr. Gayanuka Bulegoda	2002209805286	Neurologist	 
D001	Dr. Gayanuka Bulegoda	2002209805286	Neurologist	 
D001	Dr. Gayanuka Bulegoda	2002209805286	Neurologist	 
D001	Dr. Gayanuka Bulegoda	2002209805286	Neurologist	 
D001	Dr. Gayanuka Bulegoda	2002209805286	Neurologist	 
D001	Dr. Gayanuka Bulegoda	2002209805286	Neurologist	 
D001	Dr. Gayanuka Bulegoda	2002209805286	Neurologist	 

7.10 Hospital View Doctor Pop Up

Doctor view pop up displays the relevant doctor detail.

View Doctor

← Back

Name

Dr. Gayanuka Bulegoda

Speciality


Cardiologist

Email

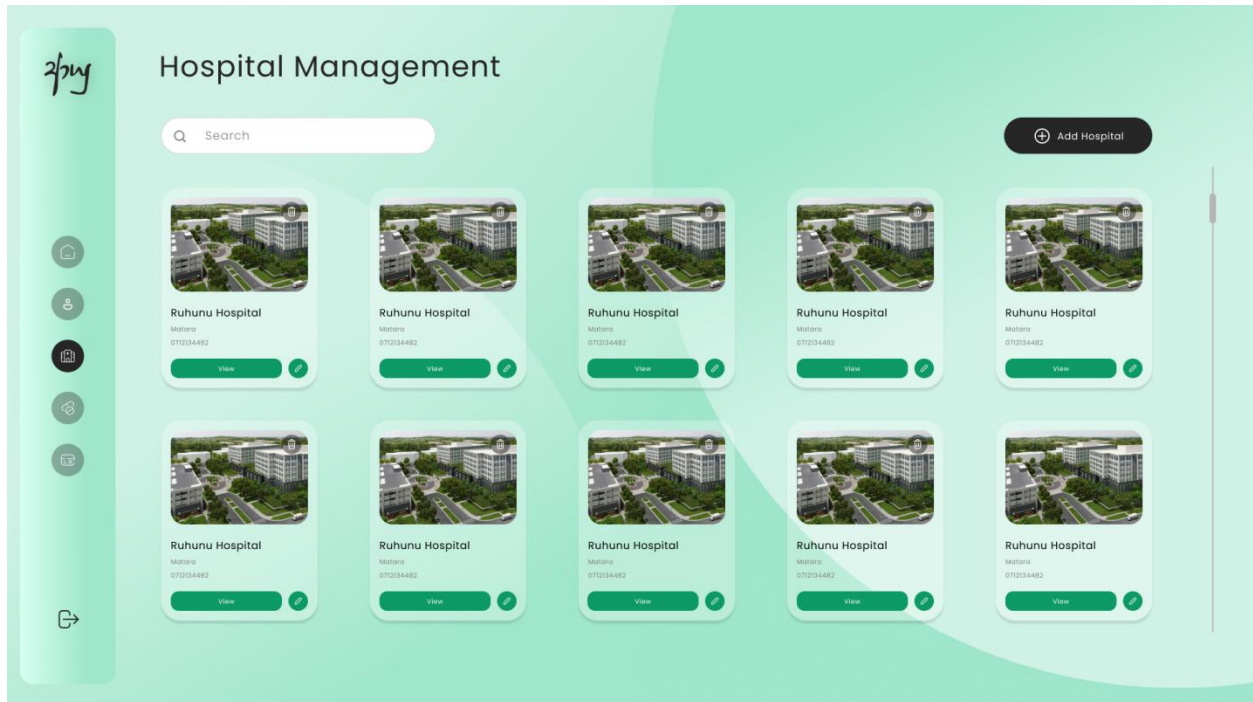
grbulegoda@gmail.com

Phone Number

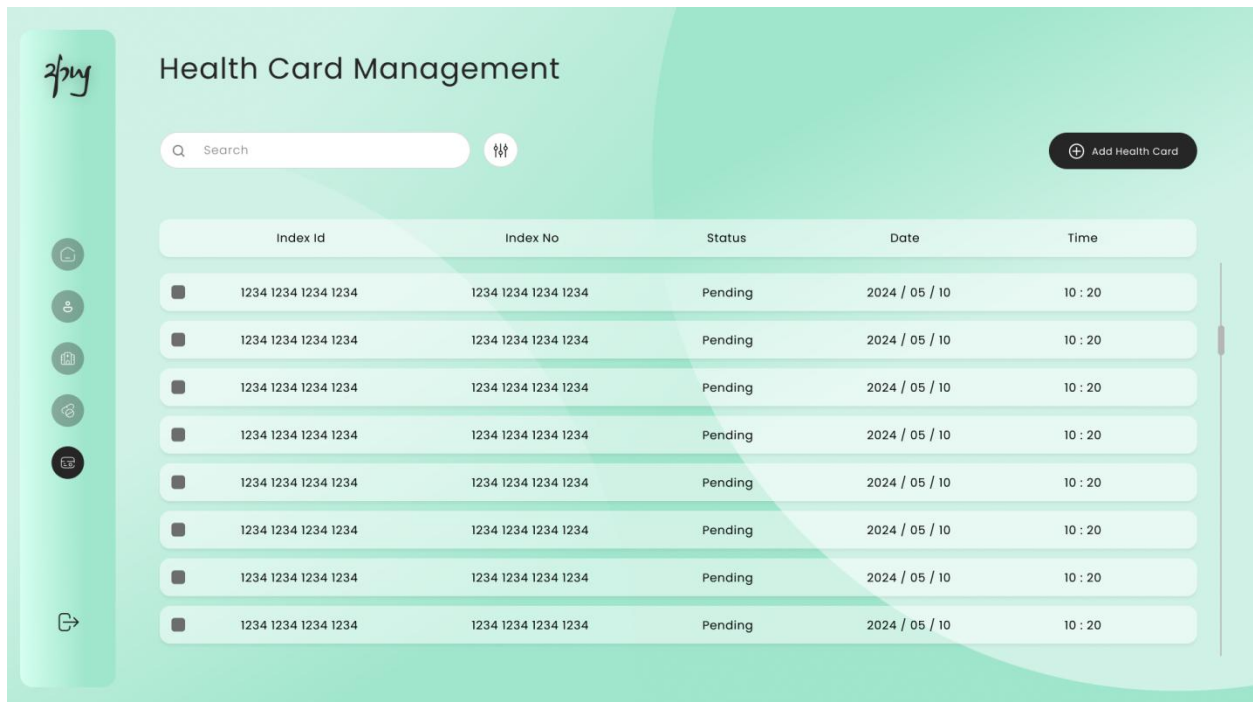
0701234567



7.11 Admin Hospital Manage Page




7.12 Admin Health Card Manage Page



7.13 Admin Health Card Confirm Pop up

Add Card

 Back

No	Index Number
00001	1652 5888 7845 8965
00002	1652 5888 7845 8965
00003	1652 5888 7845 8965
00004	1652 5888 7845 8965
00005	1652 5888 7845 8965
00006	1652 5888 7845 8965
00007	1652 5888 7845 8965
00008	1652 5888 7845 8965

Save

08. Evaluation

8.1 Introduction

This chapter includes software testing and user evaluation details. Software testing means verification and validation of the application under test (AUT). It helps to identify the missing requirements, gaps, and errors that make conflict with the actual requirements.

Main levels of Software testing are functional and nonfunctional. User evaluation helps to know how well users can learn a product and use a product to achieve their goals. It also shows how satisfied users are with the process.

8.2 Software Testing Levels

Functional Testing Functional testing is used to verify/validate that each and every function of the software operates in conformance with the requirement specification.

1. Unit Testing

These kinds of testing helps to test each module separately. It checks whether the individual components are fulfilling the functionalities or not.

2. Integration Testing

In this testing phase, individual software modules are combined and tested as a group to make sure the system is ready for system testing.

3. System testing

System testing will be performed on a complete integrated system. It tests the overall functionalities of the system and involves load testing, performance testing, security testing, etc.

4. Acceptance Testing

This is the software testing process where a system is tested for acceptability. The main target of this test type is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery or not. Acceptance Testing is normally done by the end-user or the customer.

8.3 Non-Functional Testing

Non-functional testing is a type of testing to check non-functional facts (performance, usability, reliability, etc.) of software.

1. Performance Testing

This is a testing type executed to determine the system parameters (users) in terms of reactivity and solidity under various workloads.

2. Usability Testing

This is a testing type executed to expose usability defects.

8.4 Test Plan

The test plan is designed to describe the scope, objectives, resources, and schedule of a software testing effort for this application.

8.5 Test Strategy

Scope of Testing

❖ **Features to be tested – (Functional & Non -Functional)**

Features	Phase
Hospital Registration	phase 01
Patient Registration	phase 01
Doctor Registration	phase 01
Patient Data Management	phase 02
Hospital Data Management	phase 03
Doctor Data Management	phase 03
Medicine Data Management	phase 04

❖ **Features not to be tested**

- Hardware Interfaces
- Database Logical
- Communication Interfaces

❖ **Test Type**

In the project, there are 6 types of testing that should be conducted.

1. Unit Testing
2. Integration Testing
3. System testing
4. Acceptance Testing
5. Mobile Testing
6. Performance Testing

8.6 Test Cases and Test Results

Test Case ID	Test Scenario	Test Case	Expected Results
A01	Sign in to the system with invalid data	Sign in with empty username and password	Prompt error messages
		Sign in with correct password and empty username	Prompt error messages
		Sign in with correct username and empty password	Prompt error messages
		Sign in with incorrect username and password	Prompt error messages
A02	Sign in to the system with valid data	Sign in with correct username and correct password	Successfully Sign in to the system

❖ User Sign in

Table A.01 Test Cases and Results – User Login

Test Case ID	Test Scenario	Test Case	Expected Results
A03	Submit Hospital details with invalid data	Remain all the fields empty	Prompt error messages mandatory fields
A04	Submit Hospital details with valid data	Filled all the fields correctly	Successfully registered the Hospital
A05	Validations	Enter all the text field maximum length with characters numeric & special characters	Successfully registered the Hospital

❖ Hospital Registration.

Table A.02 Test Cases and Results – Hospital Registration

Test Case ID	Test Scenario	Test Case	Expected Results
A06	Submit Patient details with invalid data	Remain all the fields empty	Prompt error messages mandatory fields
A07	Submit Patient details with valid data	Filled all the fields correctly	Successfully registered the Patient
A08	Validations	Enter all the text field maximum length with characters numeric & special characters	Successfully registered the Patient
		NIC in invalid format.	Error message prompt

❖ Patient Registration.

Table A.03 Test Cases and Results – Patient Registration

Test Case ID	Test Scenario	Test Case	Expected Results
A9	Submit Doctor details with invalid data	Remain all the fields empty	Prompt error messages mandatory fields
A10	Submit Doctor details with valid data	Filled all the fields correctly	Successfully registered the Doctor
A11	Validations	Enter all the text field maximum length with characters numeric & special characters	Successfully registered the Doctor
		NIC in invalid format.	Error message prompt

❖ Doctor Registration.

Table A.04 Test Cases and Results – Doctor Registration

09.Conclusion

9.1 Introduction

This portion chews over the objectives achieved and the future enhancements of the developed system. In Hospitals, the primary responsible for delivering quality hospital related services and providing the patient community with modern infrastructure and facilities.

9.2 Problem Encountered

1. Requirements were not Clear and Stable.

Although the client has a clear idea about the manual process of their work, they hadn't any idea how this process can be converted to an automatic system. Since the client does not have any technical knowledge or experience, it was very hard to finalize what the system does according to the client's needs. So, it took time to finalize the requirement with the client and changed many times as well.

2. Lack of Development Knowledge and Experience.

Due to a lack of development knowledge of computers, it was very difficult when implementing the system. To overcome this we need to refer to many tutorials, video tutorials, E books, etc. Since some technologies used for implementations are very new to us and We had a great experience and improved knowledge as well.

10. Lesson Learn

The knowledge added throughout the project was actually valuable. Starting from the requirement gatherings to the end of development this process gave us incomparable experience in many ways.

This project gave as a chance to test and implement the most important theories and technologies learned throughout the GDSE Program.

It also facilitated learning of very interesting new and updated technologies (React, Spring Boot, Arduino etc.) in order to improve the system performance. Moreover, special efforts were taken to learn the frameworks.

Furthermore ,working on the project encourages us to improve our technical skills as well as intellectual skills by collaborating with many individuals from collecting teams.

We also gained good knowledge in project management including time and resource management and the report writing knowledge, which was acquired by writing this dissertation, will be very important for our future career.

11. Future Work

According to the suggestions and comments gained by the client through the acceptance testing phase some functional and non-functional requirements could be added to improve the system. The following functions could be added to improve the effectiveness and functionality of the system.

- Registration of Hospital
- Registration of patient
- Doctor Registration
- Manage Patient Data
- Manage Hospital Data
- Manage Doctor Data
- Manage medicine Data
- Manage NFC Card
- Ayu's future enhancements, including telemedicine, AI-driven health predictions, pharmacy integration, doctor recommendations, and a patient feedback mechanism, will attract a wide user base. Investing in Ayu offers significant advantages and good profit potential due to its growing reach and community impact.

12. Appendix

12.1 Design Documentation

Refer to the following use case diagram descriptions to gain a clear idea about the System which was designed for the Patient's medical records .

12.2 Use Case Description – User Management

Use case	Patient detail Management System
Actors	Hospital staff,Patient, Admin,Doctor
Description	
Simplify the get Patients medical reports through the NFC Card	
Normal Cause	
1. get medical reports through the NFC Card	
Alternative Causes	
❖ If duplicate user details are entered, a relevant error message will be displayed.	
Post conditions	

12.3 System Documentation

This system documentation provides the guidelines to implement the back-end and the respective front-end in a relevant server PC.

Although the following Requirements should be available with the server to host both the front-end and the back-end services

❖ **Developing end**

- MySQL server - Database connectivity and management.
- Figma and diagram.net - Logo and other designs such as User-Interfaces.
- Visual Studio Code - For front-end development.
- IntelliJ IDEA- IDE for Java development.
- JDK 17- Java is fast, secure, and reliable. From laptops to data centers, game consoles to scientific super computers, cell phones to the Internet.

❖ **Client End.**

- Run react application on web browser.

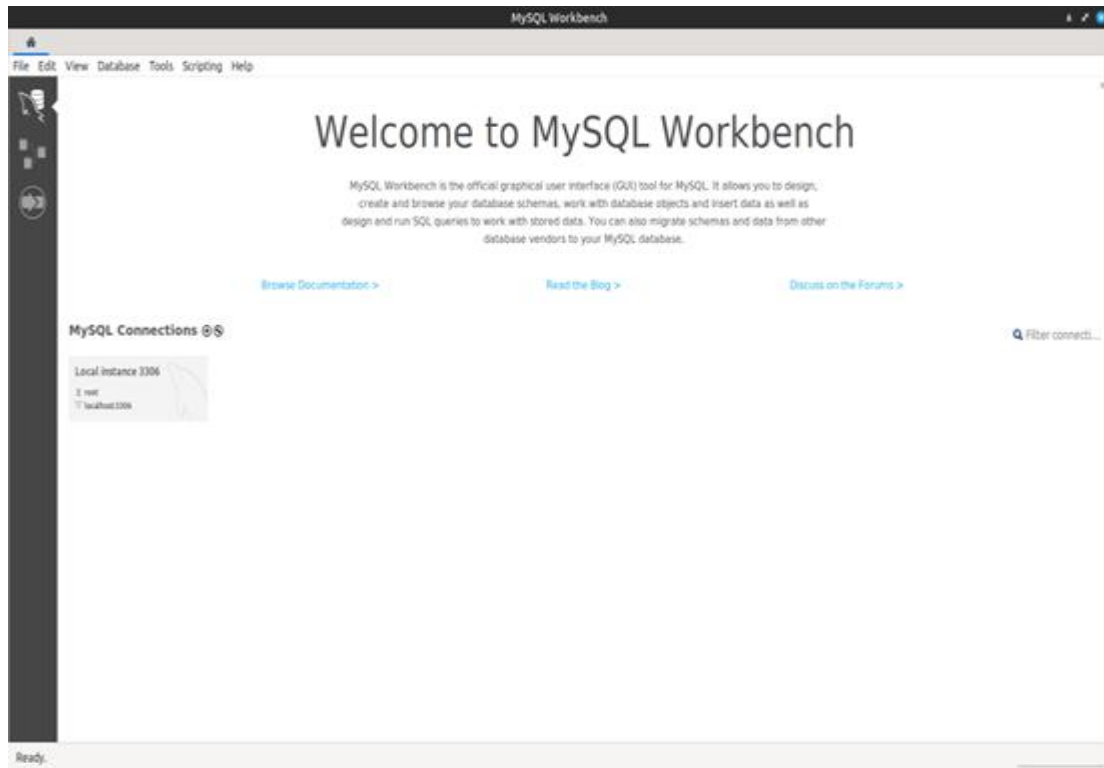
Below described the process of configuring each item in the deployment environment.

❖ **Configuring Deployment Environment: DB Setup - MySQL**

MySQL is an open-source relational database management system (RDBMS). A relational database organizes data into one or more data tables in which data may be related to each other; these relations help structure the data.

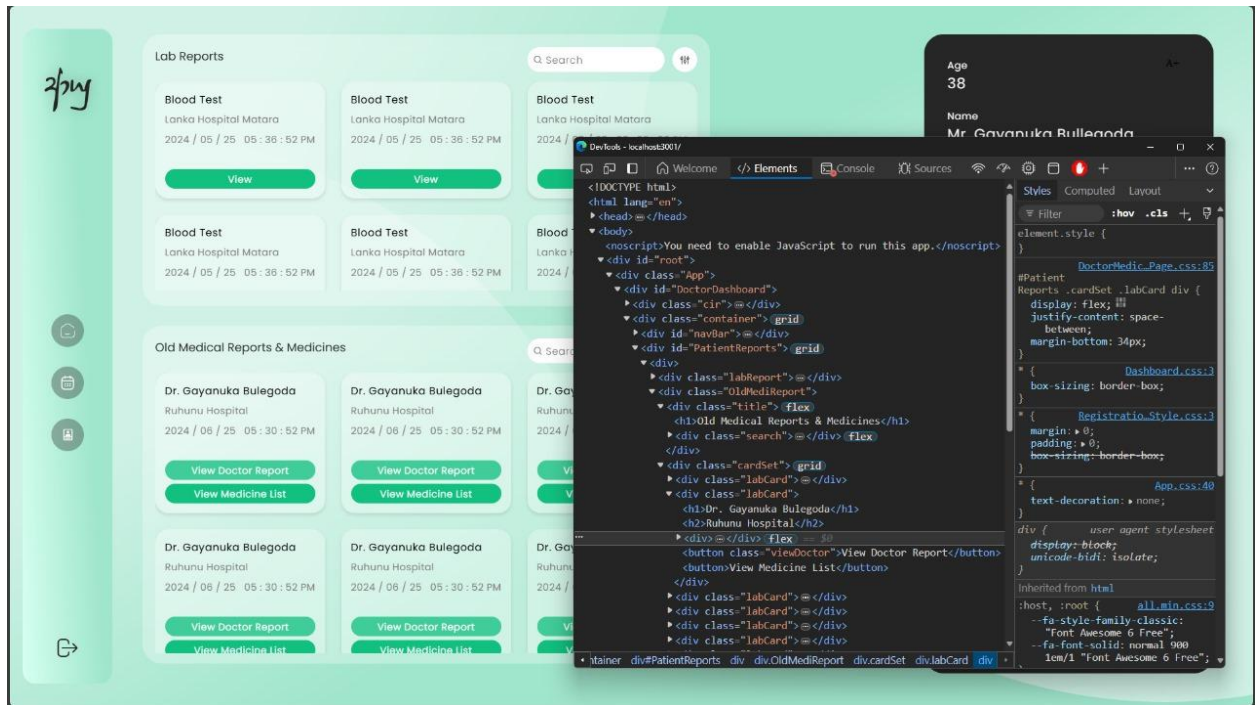
SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database.

Step 1: Navigate to <https://www.mysql.com/downloads/> and download the MySQL Enterprise Edition, when downloading get the zip file rather than msi setup



❖ **Configuring Deployment Environment: React**

React is one of the most popular front-end libraries for designing websites. It helps define form, function, and style, which are all necessary skills for web developers.



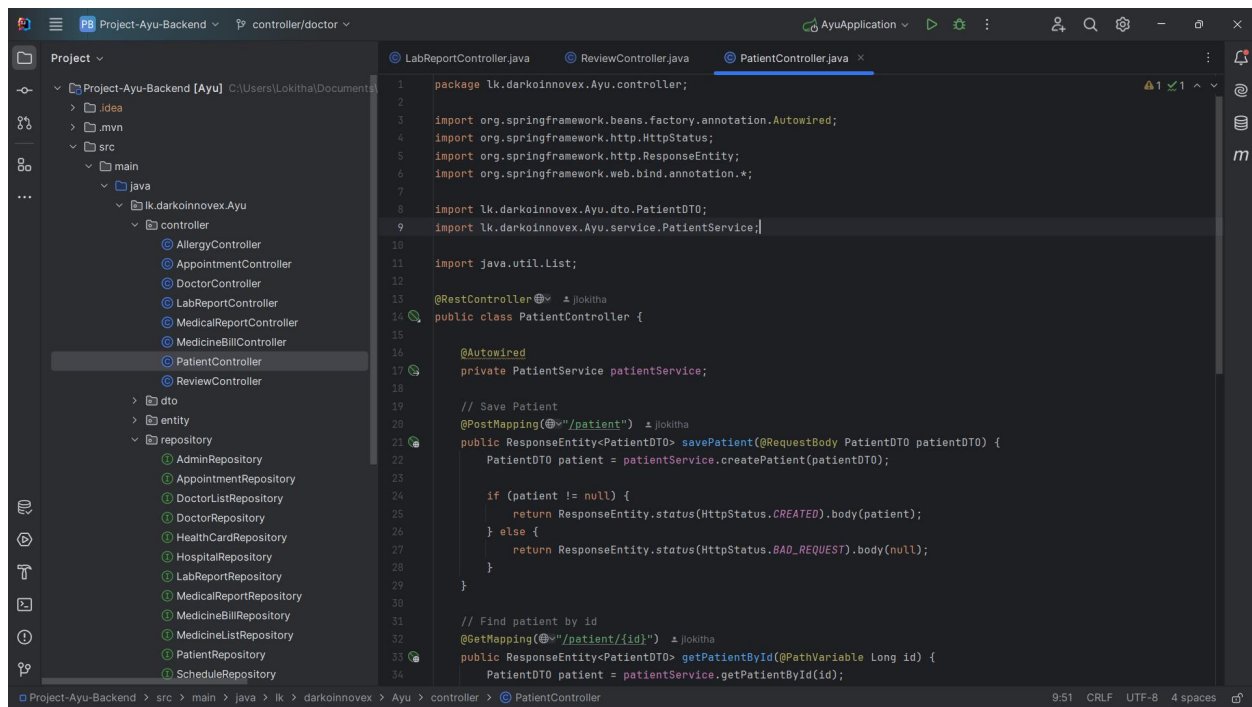
❖ Configuring Deployment Environment: Coding - IntelliJ IDEA

IntelliJ IDEA is an integrated development environment (IDE) written in Java for developing computer software written in Java, Kotlin, Groovy, and other JVM-based languages.

It is developed by JetBrains (formerly known as IntelliJ) and is available as an Apache 2 Licensed community edition, and in a proprietary commercial edition. Both can be used for commercial development.

Step 1: Navigate to

<https://www.jetbrains.com/idea/download/#section=windows>



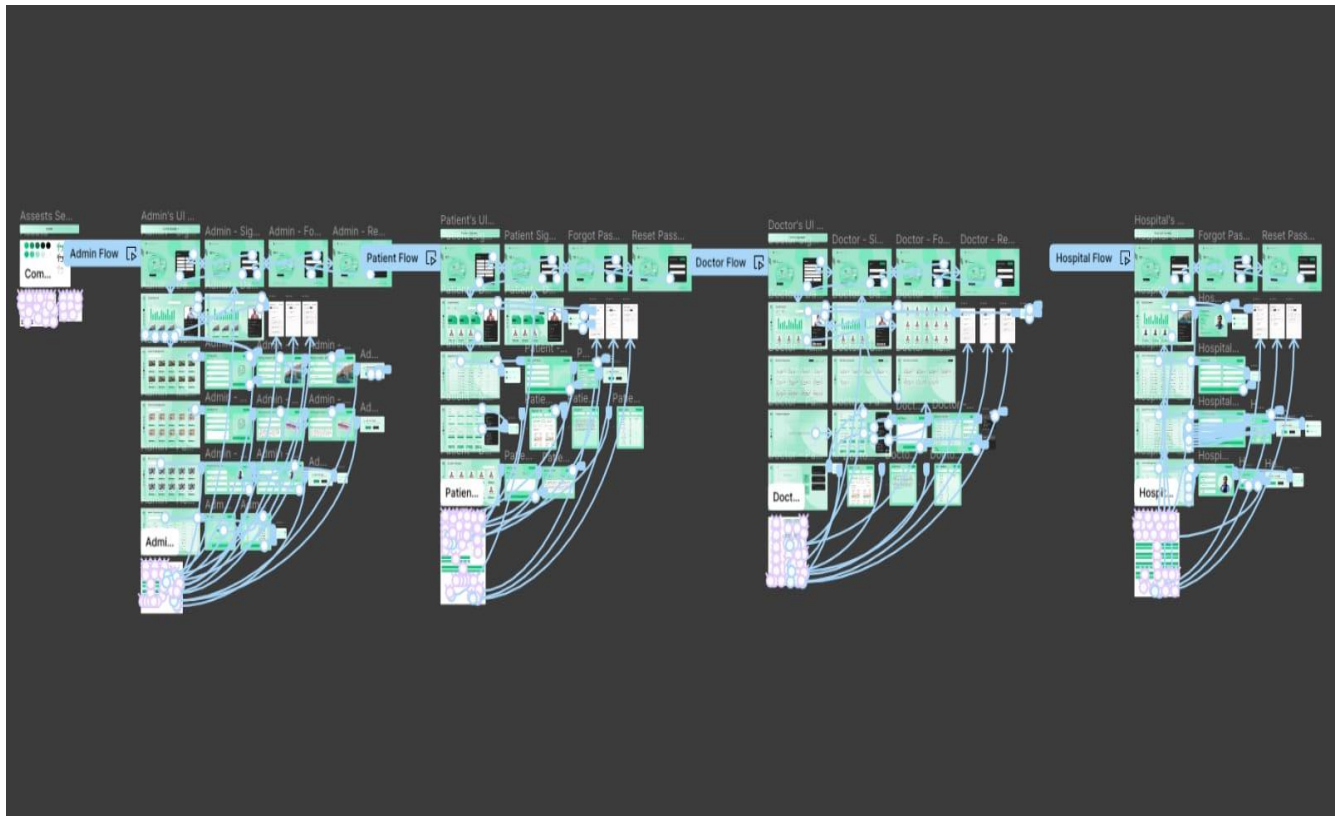
❖ Configuring Deployment Environment: UI Design - Figma

Figma is a collaborative web application for interface design, with additional offline features enabled by desktop applications for macOS and Windows. The feature set of Figma focuses on user interface and user experience design, with an emphasis on real-time collaboration, utilizing a variety of vector graphics editors and prototyping tools. The Figma

mobile app for Android and iOS allows viewing and interacting with Figma prototypes in real time on mobile and tablet devices.

Step 1: Navigate to

<https://www.figma.com/downloads/>



❖ **Configuring Deployment Environment: Wire Frame - Diagram.net**

diagrams.net is a free and open source cross-platform graph drawing software developed in HTML5 and JavaScript. Its interface can be used to create diagrams such as flowcharts, wireframes, UML diagrams, organizational charts, and network diagrams.

Step 1: Navigate to <https://github.com/jgraph/drawio-desktop/releases/tag/v20.8.10>

