

MINOR PROJECT REPORT
On “Hospital Management System (Care Heaven)”

Submitted for the partial fulfilment of the requirement for Minor project work
of 5th semester of Bachelor of Computer Application (BCA) programme.

Department of Computer Science
NERIM GROUP OF INSTITUTIONS



Padma Nath Sharma Bhawan, Jayanagar, Guwahati-22

Submitted by

Priyakshi Saikia

BCA 5th semester

University Roll No: 22992094, Regd. No: C2200662

Session 2022-2025

Internal Guide

Mr. Dipankar Dutta

Assistant Professor

NERIM Group of Institutions, Guwahati

CERTIFICATE OF APPROVAL

This is to certify that the project work entitled “**Hospital Management System (Care Heaven)**” is carried out by Priyakshi Saikia bearing Examination Roll No: 22992094 , of Dept of Computer Science, NERIM, Guwahati under guidance of Mr. Dipankar Dutta, Asst. Professor, Dept of Computer Science, NERIM, Guwahati has been found satisfactory and is hereby approved as a project work carried out and presented in a manner required for its acceptance in partial fulfilment of course work of 5th semester of 3 year full time BCA under Dibrugarh University, Dibrugarh, Assam.

Internal Examiner

External Examiner

Date:

Date:

Place: NERIM, Guwahati

CERTIFICATE FROM HOD

This is to certify that the project work entitled “**Hospital Management System (Care Heaven)**” is carried out by Priyakshi Saikia bearing University Registration No:C2200662 and Roll Number:22992094, of NERIM and Internal guidance of Mr. Dipankar Dutta, in partial fulfilment of minor project work of 5th semester of 3 years full time BCA course under Dibrugarh University, Dibrugarh, Assam is accepted by the Department of Computer Science, NERIM, Guwahati.

Dr. Tarali Kalita
HOD, Dept of Computer Science
NERIM Group of Institutions
Guwahati, Assam

Date:

CERTIFICATE FROM INTERNAL GUIDE

This is to certify that the project work entitled “**Hospital Management System (Care Heaven)**” is a Bonafide work carried out by Priyakshi Saikia of BCA 5th semester of 3 years full time BCA course under Dibrugarh University at NERIM, Guwahati, bearing University Examination Roll Number: 22992094, under my personal supervision and guidance. The report is found worthy of acceptance for the partial fulfilment of minor project work of BCA 5th semester under 3-year full time BCA program under Dibrugarh University, Dibrugarh, Assam.

All help received has been duly acknowledged and no part of this report has been reproduced for any other degree or diploma.

Mr. Dipankar Dutta
Asst. Professor, Dept of Computer Science
NERIM Group of Institutions
Guwahati, Assam

Date:

ACKNOWLEDGEMENT

At the very inception I would like to thank Dr. Tarali Kalita, HOD, Computer Science Department, NERIM Group of Institutions for his valuable suggestion.

With immense pleasure I acknowledge my inertness to all persons whose support and guidance have helped me in carrying out this project.

Also, I acknowledge most sincerely my heartiest thanks and gratitude to my internal guide Mr. Dipankar Dutta who has guided me throughout the period of this project work.

To those mentioned above and to those who inspired and encouraged me, I would like to express my gratitude once again.

With regards

Priyakshi Saikia

University Roll No: 22992094

BCA 5th semester

DECLARATION

I hereby declare that the project work called “**Hospital Management System (Care Heaven)**” submitted to the Computer Science Department of NERIM, Guwahati under Dibrugarh University is a record of an original work done by me under the guidance of “**Mr. Dipankar Dutta**” and this project is submitted for the partial fulfilment of the degree of Bachelors of Computer Application under Dibrugarh University, Dibrugarh, Assam. The results embodied in this, have not been submitted to any other University or Institute.

With regards

Priyakshi Saikia

University Roll No: 22992094

BCA 5th semester

CONTENT

CHAPTER 1: INTRODUCTION

1.1 BRIEF BACKGROUND OF THE ORGANIZATION

1.2 PROJECT TITLE

1.3 PROJECT DEFINITION

1.4 EXISTING SYSTEM

1.5 PROPOSED SYSTEM

1.6 OBJECTIVES OF THE PROJECT

1.7 HARDWARE AND SOFTWARE USED

CHAPTER 2: SYSTEM ANALYSIS AND DESIGN

2.1 INTRODUCTION

2.2 FEASIBILITY STUDY

2.2.1 SOFTWARE DEVELOPMENT LIFE CYCLE

2.2.2 TECHNICAL FEASIBILITY

2.2.3 BEHAVIOURAL FEASIBILITY

2.2.4 ECONOMICAL FEASIBILITY

2.3 SOFTWARE REQUIREMENTS SPECIFICATIONS (SRS)

2.4 STRUCTURED ANALYSIS

2.4.1 DATA FLOW DIAGRAM (DFD)

2.4.2 ER DIAGRAM

2.5 DATABASE DESIGN

2.5.1 DATA DICTIONARY

2.5.2 TABLE DESIGN

CHAPTER 3: INPUT AND OUTPUT DESIGN

CHAPTER 4: TESTING

4.1 INTRODUCTION

4.2 TYPE OF TESTING

4.2.1 UNIT TESTING

4.2.2 INTEGRATION TESTING

4.2.3 SYSTEM TESTING

4.2.4 VALIDATION TESTING

CHAPTER 5: CONCLUSION AND FUTURE SCOPE OF THE PROJECT

CHAPTER 6: BIBLIOGRAPHY

CHAPTER 1: INTRODUCTION

1.1 Brief Background of The Organisation

NERIM Group of Institutions is a group of colleges established in the year 1992 by NERIM Educational Society with a view to promote specialized professional education in the North Eastern Region of the Country. The Institution is affiliated to Dibrugarh University, Gauhati University and Assam Science and Technology University (ASTU) for its various courses. The Department of Technical Education, Government of Assam initially extended its active support for the establishment of this Institution with a view to promote skill based professional education in this part of the country.

The Institution has come into existence with the prime objective of designing and formulating need-based academic and professional programmes independently as well as in collaboration with other leading Management Institutes and Universities for imparting teaching and training to the young educated people in the skills that find ready employment in an era of high career uncertainty. The Institution is committed to contribute significantly towards the various needs of the cross-section of the society by producing trained young professionals in the fields of Management, Computer Science & IT, Education, Social Work, Applied Psychology, Teacher Education, Commerce and Juridical Studies. The Institution is dedicated towards imparting quality teaching and training to the future generation in skills that untie exclusive avenues of opportunities and ready employment amidst high levels of unemployment and career dilemmas.

1.2 PROJECT TITLE

“Hospital Management System (Care Heaven)”

1.3 PROJECT DEFINITION

A Hospital Management System (HMS) is a software platform designed to streamline and automate the operations in a hospital. It acts as a central hub for managing various aspects of hospital administration, patient care, and medical services, ensuring efficiency, accuracy, and timely access to information.

The system facilitates patient registration by storing their information within the platform. Each patient is assigned a unique ID, and the details of both patients and staff are automatically saved. It also features a search function, allowing administrators to look up doctor availability and patient information via the unique ID. Access to the system is secured by a username and password, with only the admin having the authority to add data to the database.

The primary function of the system is to record and store the details of doctors and patients, and retrieve this information as needed. The aim of this project is to digitalise the traditional hospital operations that were previously carried out manually.

1.4 EXISTING SYSTEM

The outcomes of some studied systems reveal that existing hospital management systems frequently depend on a combination of manual processes and outdated methods, resulting in inefficiencies and delays in patient care. These systems often involve cumbersome paperwork, fragmented record-keeping, and disorganized data, which hinder effective management of patient information, appointments, billing, and overall hospital operations. Tasks such as tracking doctor availability or accessing patient records can be time-consuming and challenging. Consequently, these systems struggle to meet the demands of modern healthcare environments efficiently.

1.5 PROPOSED SYSTEM

The proposed system focuses on ensuring efficiency, accuracy, and timely access to information without delay. The system mainly contains three modules the user/patient, the doctor and the admin. The following are some features of the proposed system:

- The proposed system provides registration of patients, assigning unique IDs, and storing their personal and medical details.
- It ensures user authentication which is carried out via a username/email and password, allowing only authorized administrators to add or modify data in the database.
- The system enables fast and easy retrieval of patient and doctor information using a unique ID, ensuring timely access to critical data.

1.6 OBJECTIVES OF THE PROPOSED SYSTEM

- **Streamlined Processes:** Automates important tasks such as patient registration, appointment scheduling, billing and diagnosis, and staff management to make things run more smoothly.
- **Centralized Data Storage:** Combines patient and staff data into one central system, making it easy and quick to access information and reduce redundancy.
- **Reduces Manual Administrative Tasks:** Reduces the need for manual administrative tasks, enabling staff to concentrate more on patient care and enhancing fast coordination among departments.
- **Reduces Manual Paperwork:** The proposed system aims to reduce the manual paperwork regarding diagnosis reports by providing_digital diagnosis reports of the patients.

1.7 HARDWARE AND SOFTWARE USED

HARDWARE SPECIFICATIONS

PROCESSOR: Intel core i3

RAM: 1GB or More

HARD DISK: 20GB or more

KEYBOARD: Standard

MOUSE: Optical

SOFTWARE SPECIFICATION

OPERATING SYSTEM: Windows 8 or better

SERVER: XAMPP

FRONT-END: HTML5, CSS3, JavaScript, Bootstrap

BACK-END: PHP

DATABASE: MySQL

CHAPTER 2: SYSTEM ANALYSIS AND DESIGN

2.1 INTRODUCTION

System design and analysis are essential steps in building great software. The process starts with understanding the problem, identifying what users need, and figuring out the system's goals. This is followed by designing how everything will work, from the overall structure to the smaller components and data flow. These steps help create software that's not just functional but also reliable, scalable, and easy to use, ensuring it meets both user needs and technical requirements.

2.2 FEASIBILITY STUDY

A feasibility study is a critical step in evaluating whether a proposed project is practical and achievable. It assesses various factors such as technical requirements, financial costs, operational challenges, and the project's overall feasibility. By identifying potential risks and constraints early, a feasibility study helps determine if the project is worth pursuing and guides decision-making to ensure its success.

2.2.1 SOFTWARE DEVELOPMENT LIFE CYCLE

The Software Development Life Cycle (SDLC) is a systematic process for designing, developing, testing, and deploying software applications. It ensures high-quality software delivery by providing a structured approach.

“The Hospital Management System (Care Heaven)” project follows a structured Software Development Life Cycle (SDLC) to ensure effective planning, development, and implementation. Below is a detailed explanation of each stage tailored specifically for this project.

1. Planning:

Planning is the initial phase where the objectives of the project are determined, and a clear roadmap is created.

Objective: Establish the foundation for the proposed system.

Activities Performed on the proposed system:

- Gathering initial requirements by studying existing hospital systems and conducting surveys to understand the needs of various users.
- Creating a project plan that defined the scope, objectives, and timelines for the system development.

2. Feasibility Study:

This phase assesses whether the project is practical and achievable. It involves evaluating technical, financial, and operational factors to ensure the system can be built within the constraints of time, and resources.

Objective: Evaluate the viability of the proposed system.

Activities Performed on the proposed system:

- Analysing the technical aspects, ensuring that technologies like PHP, MySQL, and HTML/CSS were suitable for the project.
- Evaluating the hardware, including laptops and computers, to confirm the system could be effectively developed and run on available resources.
- Choosing open-source technologies like PHP, MySQL, HTML/CSS, and JavaScript, which made the project economically feasible by minimizing development costs.

3. System Design:

System design involves creating a blueprint for the software, detailing how the system will look and function. It includes the user interface, the database structure and the system architecture (how different parts of the system will interact).

Objective: Create a blueprint for the proposed system.

Activities Performed on the proposed system:

- Designed the user interface (UI) to be simple and intuitive, making it easy for hospital staff to interact with the system.
- Developed a database schema to manage essential data like patient records, appointments, billing, doctor details, patient details and login details.
- Created system architecture diagrams to plan how the front-end, back-end, and database would interact.

4. Implementation:

The implementation phase where the actual development of the system occurs. The code is written to build the features and functionalities defined in the design phase. It includes creating the front-end (what users interact with) and back-end.

Objective: Build the proposed according to the design specifications.

Activities Performed on the proposed system:

- Developed the front-end using HTML, CSS, and JavaScript to create a responsive user interface.
- Used PHP for the back-end to handle data processing, user authentication, and business logic.
- Integrated key features like patient registration, appointment scheduling, and billing into the system, ensuring they worked seamlessly.

5. Testing:

Testing is the process of verifying that the system works as expected. It includes checking individual features ensuring that different parts of the system work together (integration testing), and getting feedback from actual users to confirm that the system meets their needs.

Objective: Validate the proposed system to ensure it meets requirements and is free of defects.

Activities Performed on the proposed system:

- Conducted functional testing to ensure all features worked as intended.
- Performed integration testing to verify that different modules, such as the admin, doctor, and user dashboards, interacted correctly.
- Addressed any issues or bugs found during testing to improve the system's overall quality and functionality.

6. Deployment:

Deployment involves making the system available for actual use. It includes setting up the production environment, transferring data from older systems, and ensuring all features are fully operational. Training sessions are conducted to help users adapt to the new system.

Objective: Making the proposed system live.

Activities Performed on the proposed system:

- Deployed the proposed system on XAMPP, ensuring all features were fully operational.

7. Maintenance:

Maintenance involves keeping the system functional and up-to-date after it is launched. This includes monitoring its performance, fixing bugs, enhancing features, and applying updates to address security vulnerabilities or improve usability.

Objective: Ensuring proposed system remains functional and up-to-date after deployment.

Activities Performed on the proposed system:

- Monitored the system's performance to identify and resolve any potential issues.
- Took measures to ensure the system remained efficient and continued operating smoothly over time.

2.2.2 TECHNICAL FEASIBILITY

Technical feasibility refers to the assessment of whether the proposed system can be developed and implemented with the current technology and available technical resources.

The technical feasibility of the project was evaluated by assessing the suitability of open-source technologies like PHP, MySQL, HTML/CSS, and JavaScript. These technologies were chosen for their compatibility, reliability, and wide use in web development. The existing hardware, such as personal laptops, was sufficient to handle the development process, ensuring that the system could be built and run effectively without additional infrastructure costs. The selected tools and resources were able to meet the technical requirements of the project, making it technically feasible within the available constraints.

2.2.3 BEHAVIOURAL FEASIBILITY

Behavioural feasibility focuses on assessing whether the proposed system can be accepted and effectively adopted by the users, taking into consideration their behaviour, readiness for change, and potential resistance.

In the case of the proposed system, operational feasibility was closely evaluating the system which was designed with an intuitive interface to ensure ease of use, and specific features, such as doctors adding prescriptions and users receiving them, were incorporated to streamline workflow and enhance user acceptance. The project's success hinged on ensuring that hospital staff could transition smoothly to using the new system.

2.2.4 ECONOMICAL FEASIBILITY

Economic feasibility evaluates whether the project is financially viable, considering the costs of development, implementation, and maintenance in comparison to the potential benefits.

The project was developed using open-source technologies like PHP, MySQL, and HTML/CSS, minimizing costs. The primary investment was time spent on development. With significant benefits in improving hospital management efficiency and processes, the project proved economically feasible, delivering high value with low expenses.

2.3 SOFTWARE REQUIREMENTS SPECIFICATIONS (SRS)

A Software Requirements Specification (SRS) is a vital document in software engineering that outlines the functional and non-functional requirements of a project. It serves as a clear agreement between stakeholders, ensuring alignment and reducing misunderstandings. The SRS guides development, testing, and validation, helping identify challenges early and ensuring the project meets its goals efficiently.

SI No.	Requirement Name	Requirement Description	Priority
SRS001	Admin Log In	This function enables the admin to login to the system by providing a valid username and password	Mandatory requirement
SRS001.1*	Admin Log In	If found valid the system will redirect to the admin dashboard	Mandatory requirement
SRS001.2*	Admin Log In	If found invalid the system will show an alert saying invalid username or password	Mandatory requirement
SRS002	Admin insert's Data	This system allows admin to add doctor data and generates invoices	Mandatory requirement
SRS003	Admin Accepts Requests	This system allows admin to accepts doctors' appointment	Mandatory requirement
SRS004	Admin Updates Data	This system allows admin to update the doctors' data	Mandatory requirement
SRS005	Admin Deletes Data	Admin can delete doctor's data and reject doctors' appointment	Mandatory requirement

SRS for Admin

SRS For Doctors

SI No.	Requirement Name	Requirement Description	Priority
SRS001	Doctor Log In	This function enables doctors to login to the system by providing a valid username and password	Mandatory requirement
SRS001.1*	Doctor Log In	If found valid the system will redirect to the doctor dashboard	Mandatory requirement
SRS001.2*	Doctor Log In	If found invalid the system will show an alert saying invalid username or password	Mandatory requirement
SRS002	Doctor Adds Data	This action allows doctors to add the diagnosis report for respective patients	Mandatory requirement
SRS003	Doctor Updates Data	This action allows doctors to update their existing data	Mandatory requirement
SRS004	Doctor Views Data	This action allows doctors to view their scheduled appointments	Mandatory requirement

SRS For Patients

SI No.	Requirement Name	Requirement Description	Priority
SRS001	Patient Register	This action enables users to register to the system	Mandatory requirement
SRS002	Patient Log In	This function enables patients to login to the system by providing a valid username and password	Mandatory requirement
SRS002.1*	Patient Log In	If found valid the system will redirect to the patient dashboard	Mandatory requirement
SRS002.2*	Patient Log In	If found invalid the system will show an alert saying invalid username or password	Mandatory requirement
SRS003	Patient Update Data	This action enables patient can update their data	Mandatory requirement
SRS004	Patient View Data	This action enables patient can view their data like prescription and invoices	Mandatory requirement

2.4 STRUCTURED ANALYSIS

Structured analysis is a method to design systems by breaking them into smaller components, using tools like Data Flow Diagrams (DFDs) and Entity-Relationship Diagrams (ERDs). It focuses on understanding data flow and system processes, ensuring clarity and reducing complexity.

2.4.1 DATA FLOW DIAGRAM (DFD)

A Data Flow Diagram (DFD) is a visual representation of how data moves within a system, highlighting processes, data stores, and interactions. It helps analyse and model the system's functionality, making it easier to understand data processing and communication at different levels. They are especially useful for identifying inefficiencies and improving system design.

The following are the symbols used in a Data Flow Diagram (DFD) to represent data are:

External Entity



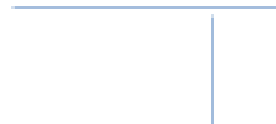
Process



Data Flow



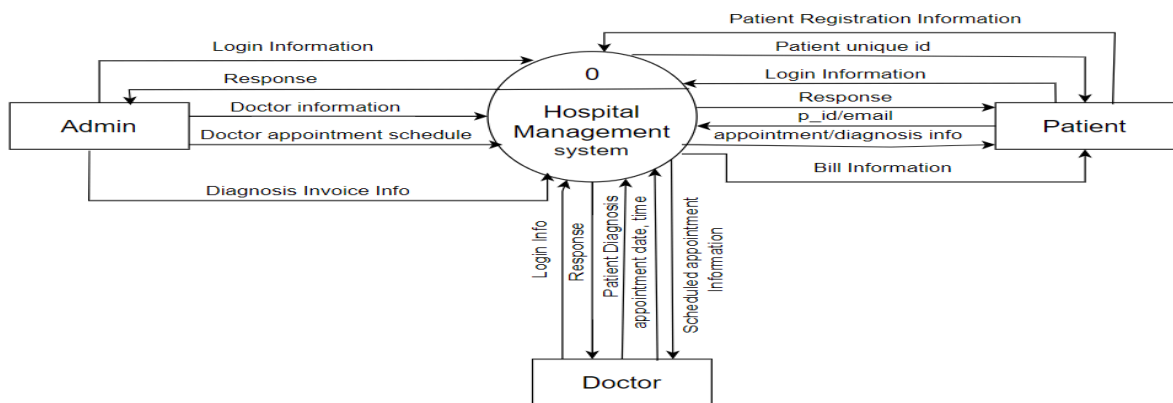
Data Storage



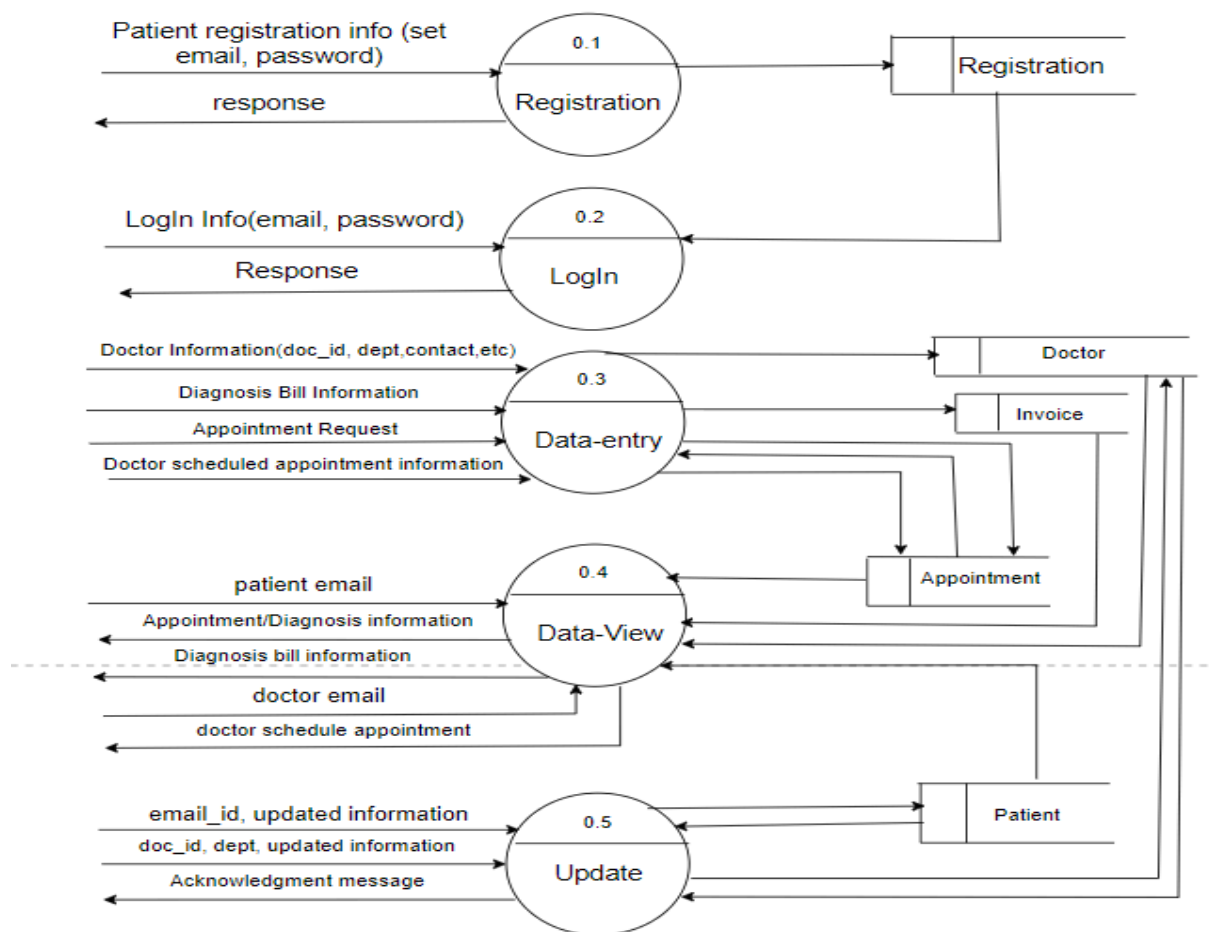
A report or output from the system

0 LEVEL DFD:





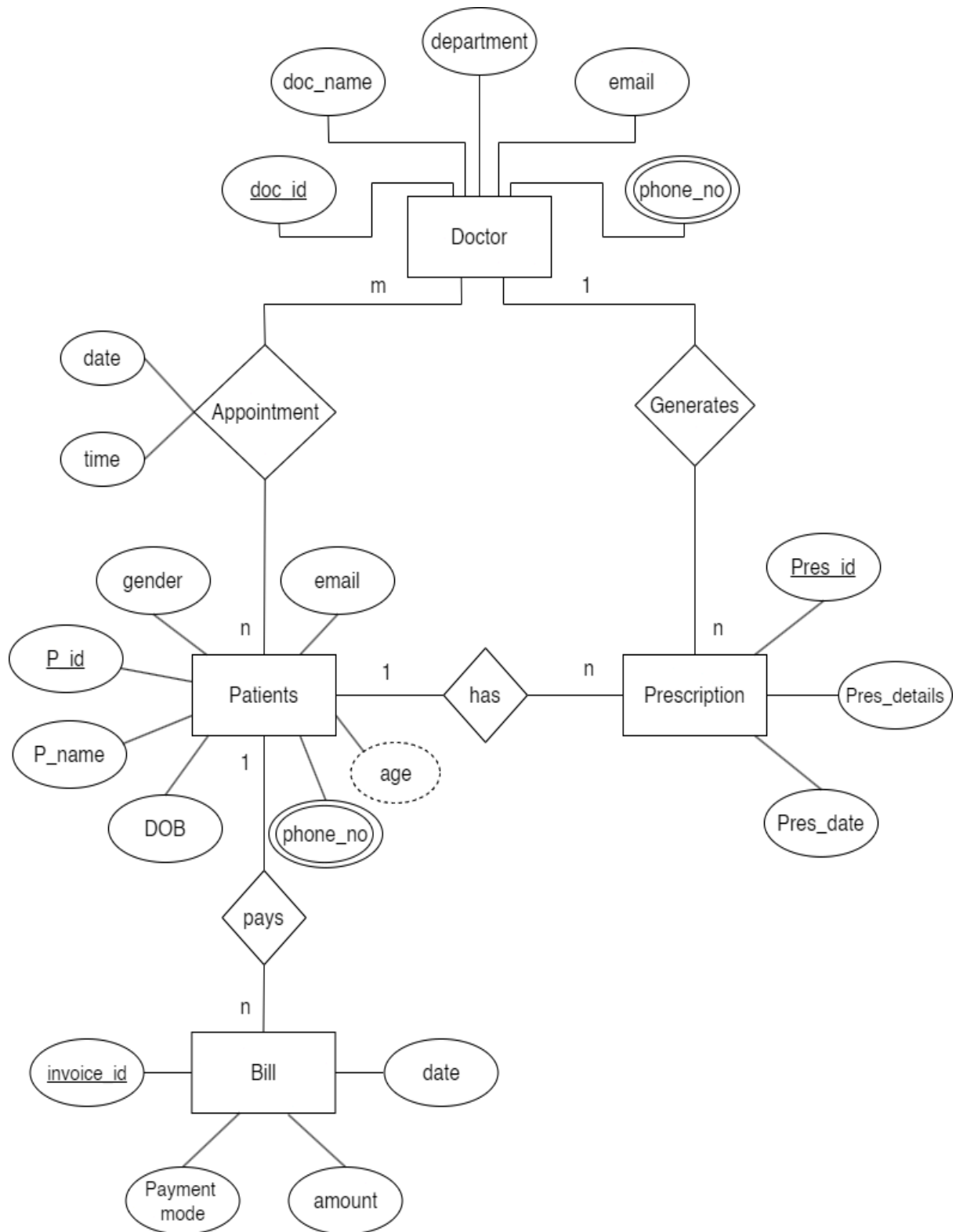
1st LEVEL DFD:



2nd LEVEL DFD:

- **Entities:** An entity represents an object in the real world with independent existence, either physical or conceptual.
- **Attributes:** Attributes are the properties that describe an entity, and their values are stored in the database.
- **Relationships:** Entities are often related to each other through relationships. A binary relationship involves two entities, while a ternary involves three.
- **Cardinality Ratio:** The cardinality ratio defines the number of instances an entity can participate in within a relationship. It can be 1:1, 1: N, or M: N for binary relationships.

ER diagram:



2.5 DATABASE DESIGN

2.5.1 DATA DICTIONARY

A data dictionary is a centralized repository that describes the structure, attributes, and relationships of data elements within a database, including tables, fields, data types, and constraints.

SL No	Field Name	Datatype	Size	Description
1	appointment_id	int	11	Unique identifier for each appointment.
2	name	varchar	100	Name of the patient booking the appointment.
3	email	varchar	100	Patient's email address.
4	department	varchar	100	Department for the consultation (e.g., Cardiology).
5	doctor_id	int	100	Links to the assigned doctor (Foreign Key -> doctors(id))
6	appointment_date	datetime	-	Date and time of the appointment.
7	status	enum	-	Current status of the appointment (e.g., pending, accepted, rejected)
8	id	Int	11	Unique identifier for diagnosis, doctor, or patient in their respective table
9	patient_name	varchar	100	Full name of the patient receiving.
10	amount	int	11	Total amount billed for the appointment.
11	payment_mode	text	-	Payment method used (e.g., cash, online, UPI)
12	doctor_name	varchar	100	Name of the doctor associated with the appointment or billing.

2.5.2 TABLE DESIGN

Register table:

Column Name	Data Type	Constraints	Description
Id	int(11)	Primary Key, AUTO_INCREMENT	Unique identifier for the user.
username	varchar(100)	NOT NULL	Username of the user.
email	varchar(100)	NOT NULL	Email address of the user.
Password	varchar(100)	NOT NULL	Encrypted password for authentication.
User-type	enum	NOT NULL	Role of the user (admin, patient, doctor).

Patient Table:

Column Name	Data Type	Constraints	Description
-------------	-----------	-------------	-------------

id	int(11)	Primary Key, AUTO INCREMENT	Unique identifier for the patient.
patient_name	varchar(100)	NOT NULL	Full name of the patient.
email_id	varchar(100)	UNIQUE, NOT NULL	Email address of the patient.
phone_no	varchar(15)	NOT NULL	Contact number of the patient.
dob	date	-	Date of birth of the patient.
age	int(11)	NOT NULL	Age of the patient.
address	Varchar(100)	NOT NULL	Address of the patients

For Doctors:

Column Name	Data Type	Constraints	Description
id	int(11)	Primary Key, AUTO INCREMENT	Unique identifier for the doctor.
Doctor_name	varchar(100)	NOT NULL	Full name of the doctor.
department	varchar(100)	NOT NULL	Medical department of the doctor.
phone_no	int(11)	NOT NULL	Contact number of the doctor.
username	varchar(50)	NOT NULL	Username for login.
email	varchar(50)	NOT NULL	Email address of the doctor.
password	varchar(100)	NOT NULL	Encrypted password for authentication.

For Appointments:

Column Name	Data Type	Constraints	Description
appointment_id	int(11)	Primary Key, AUTO INCREMENT	Unique identifier for the appointment.
department	varchar(100)	NOT NULL	Medical department of the doctor.
Appointment_date	datetime	NOT NULL	Scheduled date and time of the appointment
doctor_name	varchar(100)	NOT NULL	Name of the assigned doctor.
status	enum	Default: pending	Status of the appointment (pending, accepted, rejected).

For Prescription:

For Invoice:

Column Name	Data Type	Constraints	Description
Id	int(11)	Primary Key, AUTO_INCREMENT	Unique identifier for the diagnosis.
appointment_id	int(11)	Foreign Key -> appointments(appointment_id)	Linked appointment.
patient_name	varchar(255)	NOT NULL	Name of the diagnosed patient.
medicines	text	NOT NULL	Prescribed Medicines
health_status	text	NOT NULL	Current health status of the patient.
description	text	NOT NULL	Additional notes or comments.
status	enum	NOT NULL	Status of the diagnosis (Pending, Completed).

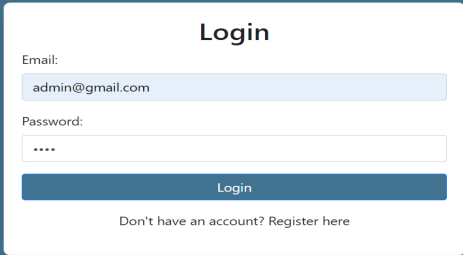
Column Name	Data Type	Constraints	Description
id	int(11)	Primary Key, AUTO_INCREMENT	Unique identifier for the invoice.
appointment_id	int(11)	Foreign Key -> appointments(appointment_id)	Related appointment ID.
amount	int(100)	NOT NULL	Amount billed.

payment_mode	text	NOT NULL	Mode of payment (e.g., Cash, UPI, Online)
--------------	------	----------	---

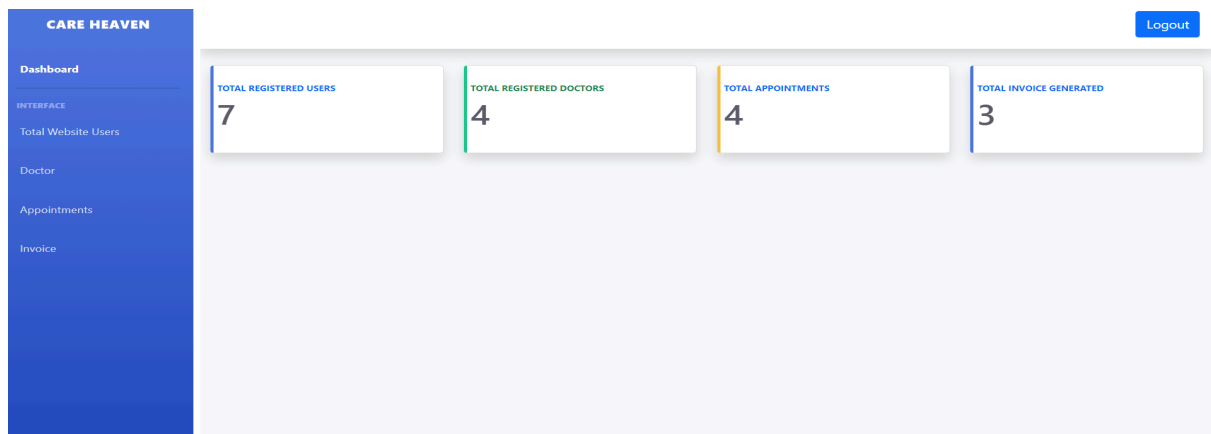
CHAPTER 3: INPUT AND OUTPUT DESIGN

For Admin:

When admin logs in using valid username and password, the admin is redirected to the admin dashboard.



The image shows a login form titled "Login" centered on a dark blue background. The form has two input fields: "Email:" with the value "admin@gmail.com" and "Password:" with masked characters "****". Below the fields is a blue "Login" button. At the bottom of the form, there is a link that says "Don't have an account? Register here".



The admin can update, delete and add doctors in the doctors' section.

CARE HEAVEN

Dashboard

INTERFACE

Total Website Users

Doctor

Appointments

Invoice

Doctor Profile

Add Doctor

ID	Name	Department	Phone Number	Username	Email	Password	Edit	Delete
1	Dr.Priyakshi Saikia	Cardiology	998789112	Priyakshi11	priyakshi@gmail.com	0000	EDIT	DELETE
2	Dr. Sneha Newar	Nephrology	2147483647	Sneha	sneha@gmail.com	123	EDIT	DELETE
3	Dr. PriyaTiwari	Neurology	2147483647	priya	priya@gmail.com	priya11	EDIT	DELETE
4	Dr. Chayanika Devi	Orthopedics	889900765	chayanika	chayanika@gmail.com	1111	EDIT	DELETE

Admin can accept or reject doctor appointments and generate invoice.

CARE HEAVEN

Dashboard

INTERFACE

Total Website Users

Doctor

Appointments

Invoice

Pending Requests

Name	Email	Phone	DOB	Age	Address	Department	Doctor	Appointment Date	Action
patient4	patient4@gmail.com	1234567891	1976-06-22	48	Goalpara	Orthopedics	4	2025-01-24 00:00:00	Accept Reject

Accepted Appointments

Name	Email	Phone	DOB	Age	Address	Department	Doctor	Appointment Date	Invoice
patient1	patient1@gmail.com	0976437543	1990-11-13	34	Guwahati	Cardiology	1	2024-11-26 00:00:00	View Invoice
patient2	patient2@gmail.com	9883438781	2000-01-23	24	Guwahati	Nephrology	2	2024-11-22 00:00:00	View Invoice
patient3	patient3@gmail.com	1234567891	2004-11-04	20	Guwahati	Cardiology	1	2024-11-28 00:00:00	View Invoice

Admin can generate and view patient invoices

CARE HEAVEN

Dashboard

INTERFACE

Total Website Users

Doctor

Appointments

Invoice

Invoices

Invoice ID	Appointment ID	Doctor Name	Patient Name	Amount	Mode Of Payment	View Invoice
1	3	Dr. Priyakshi Saikia	patient1	500	Paid through Cash	View Invoice
2	4	Dr. Sneha Newar	patient2	600	paid through online	View Invoice
3	5	Dr. Priyakshi Saikia	patient3	500	paid through UPI	View Invoice

patient4	patient4@gmail.com	1234567892	1976-06-22	48	Goalpara	Orthopedics	4	2025-01-23 00:00:00	Generate Invoice
----------	--------------------	------------	------------	----	----------	-------------	---	---------------------	------------------

For Doctor:

When admin logs in using valid username and password, the admin is redirected to the doctor dashboard.

Login

Email:

priyakshi@gmail.com

Password:

....

Login

Don't have an account? Register here

Doctor Menu

Dashboard

Appointments

Upload Prescription

Edit Information

Logout

Doctor Dashboard

Good Evening, Priyakshi11

Department: Cardiology

Doctor ID: 1

Doctor email: priyakshi@gmail.com

Doctor Phone Number: 998789112

Total Appointments: 2

Doctors can view their scheduled appointments

Doctor Dashboard

Appointments

Name	Appointment Date	Status	Action
patient1	2024-11-26 00:00:00	accepted	<div>Add Prescription</div>
patient3	2024-11-28 00:00:00	accepted	<div>Add Prescription</div>

Doctors can add patient diagnosis report

Name

patient1

2

patient3

2

Add Prescription

Patient Name

patient1

Medicines

Health Status

Diagnosis

Submit Prescription

Action

Add Prescription

Add Prescription

Doctors can edit their existing data.

Doctor Dashboard

Edit Information

Username

Priyakshi11

Department

Cardiology

Phone Number

998789112

Email

priyakshi@gmail.com

Update Information

For Patient:

When patient logs in using valid username and password they are redirected to their patient dashboard.

Login

Email:

patient1@gmail.com

Password:

User Type:

Patient

Login

Don't have an account? Register here

In the dashboard the patient can view their appointment status, their prescription and their invoices.

Patient Menu

My Appointments

Edit Data

Logout

Patient Dashboard

Good Evening, patient1@gmail.com

My Appointments

Name	Email	Phone	Appointment Date	Department	Doctor	Status	Action
patient1	patient1@gmail.com	0976437543	2024-11-26 00:00:00	Cardiology	Dr. Priyakshi Saikia	accepted	<div>View Invoice</div> <div>View Prescription</div>

The Patient can edit their existing information

Patient Dashboard
Good Evening, patient1@gmail.com
Update Your Information

Name

Email

DOB

Address

CHAPTER 4: TESTING

4.1 INTRODUCTION

Testing in software engineering is the process of evaluating a software application to ensure it meets specified requirements and functions as intended. It plays a critical role in identifying defects, verifying functionality, and ensuring the quality and reliability of the software. By systematically executing test cases, testing helps to detect errors early, reduce development costs, and enhance user satisfaction.

Types of Testing

1. **Unit Testing:** Tests individual components.
2. **Integration Testing:** Verifies interactions between modules.
3. **System Testing:** Evaluates the complete system.
4. **Acceptance Testing:** Validates readiness for deployment.

4.2.1 UNIT TESTING

Unit testing is a software testing technique that focuses on verifying the functionality of individual components or modules of a program in isolation. It ensures that each unit of the

code, such as a function, method, or class, performs as expected. Typically performed during the development phase, unit testing allows developers to identify and fix bugs early, reducing the cost of debugging and improving code quality. By providing controlled inputs and validating the outputs, it ensures the reliability of the smallest building blocks of the software and forms the foundation for higher-level testing like integration and system testing.

For Admin

SL. No	Process	Test Case	Action
SRS001	Admin Login	If admin tries to login using empty username and password	The input fields will display a "required" message.
SRS001.1	Admin Login	If admin tries to login using invalid username or password	There will be an alert saying "Invalid Username or Password"
SRS002	Insert Doctors' Data	When Admin adds doctor data	The process shows a message "Doctor data is successfully added to the database"
SRS003	Update Doctors Data	When admin updates existing doctor data	The process shows a message saying "Doctor data is updates"
SRS004	Delete Doctors Data	When admin deletes existing doctor data	The process shows a message saying "Doctor data is deleted"
SRS005	Insert Invoice Data	When admin generates invoices	There will be an alert saying "Invoice is generated successfully"
SRS006	Admin accepts doctor appointments	When admin accepts doctor appointment requests	Appointment status is updated to "Accepted"
SRS007	Admin rejects doctor appointments	When admin rejects doctor appointment requests	Appointment status is updated to "Rejected"

For Doctor

SL. No	Process	Test Case	Action
SRS001	Doctor Login	If registered doctor tries to login with a valid username and password.	Doctor is redirected to the doctor dashboard.
SRS001.1	Doctor Login	If registered doctor tries to login using invalid username or password.	Process displays an alert: "Invalid username or password."
SRS002	Doctors add prescription	When registered doctor adds prescription of the patients	Prescription is successfully added to the patient's record.
SRS003	Doctors update existing data	When registered doctor updates their existing data (e.g., diagnosis reports, profile information).	Doctor's data is updated successfully.
SRS004	Doctors view their data	When registered doctor views their scheduled appointments.	Scheduled appointments are displayed correctly.

For Patients

SL. No	Process	Test Case	Action
--------	---------	-----------	--------

SRS001	Patient register	Verify that users can successfully register to the system.	Patient is registered and directed to the login page.
SRS002.1	Patient Login	If registered patient logs in with a valid username and password.	Patient is redirected to the patient dashboard.
SRS002.2	Patient Login	If registered patient tries to login using an invalid username or password	Process displays an alert: "Invalid username or password."
SRS003	Update patients' data	When registered patient updates their data (e.g., personal information, contact details).	Patient's data is updated successfully.
SRS004	View Data	When registered patient views their data, including prescriptions and invoices	Patient's data (prescription, invoices) is displayed correctly.

4.2.2 INTEGRATION TESTING

Integration testing is a software testing process where individual modules are combined and tested as a group to ensure they work together correctly. It focuses on detecting interface errors, data flow issues, and communication problems between integrated components. Conducted after unit testing and before system testing, integration testing ensures that the software modules function cohesively and reliably, contributing to the overall system quality.

Integration ID	Process	Test	Action
INT001	Admin Login	If registered admin tries to login with correct username and password	Admin is redirected to the dashboard upon successful login.
INT002	Doctor Data Management by Admin	If admin adds, updates, and deletes doctor data.	Success messages for adding, updating, and deleting doctor data are displayed.
INT003	Invoice Management by Admin	If admin can generate invoices.	Process displays: "Invoice is generated successfully."
INT004	Admin manages Appointment Requests	If admin accepts or rejects doctor appointment requests	Appointment status updates to "Accepted" or "Rejected."
INT005	Doctor Login	If doctor tries to login with valid username and password	Doctor is redirected to the doctor dashboard upon successful login.
INT006	Doctor adds Prescription	If doctors add prescriptions for patients.	Prescription data is successfully added to the patient record.
INT007	Patient Registration and Login	Patient registration and login functionality.	Upon successful login, the patient is redirected to the dashboard. If the patient is not registered, they must complete the registration process before logging in.
INT008	End-to-End Workflow (Admin-Doctor)	Admin-added doctor data is visible and accessible by the doctor after login.	Doctor can log in and view their data added by the admin and their scheduled appointments.
INT009	End-to-End Workflow (Admin-Patient)	Admin-generated invoices are visible to patients after login.	Patient can view the invoice in their dashboard.
INT010	End-to-End Workflow (Doctor-Patient)	Doctors' prescriptions are visible to patients after login.	Patients can view the prescription added by the doctor.

4.2.3 SYSTEM TESTING

System testing ensures that the entire integrated system functions as expected and meets the specified requirements. For the "Hospital Management System (Care Heaven)," system testing covers the following key aspects:

1. Admin Login:
 - Verify that the admin can log in with the correct username and password.
 - Ensure the admin is redirected to the dashboard upon successful login.
2. Doctor Data Management by Admin:
 - Verify that the admin can add, update, and delete doctor data.
 - Ensure appropriate success messages are displayed for these actions.
3. Invoice Management by Admin:
 - Ensure the admin can generate invoices successfully.
 - Verify the system displays the message: "Invoice is generated successfully."
4. Admin Managing Appointment Requests:
 - Verify that the admin can accept or reject doctor appointment requests.
 - Ensure the appointment status updates to "Accepted" or "Rejected."
5. Doctor Login:
 - Ensure doctors can log in with valid credentials.
 - Verify that doctors are redirected to their dashboard upon successful login.
6. Doctor Adding Prescriptions:
 - Verify that doctors can add prescriptions for patients.
 - Ensure that prescriptions are successfully added to the patient's record.
7. Patient Registration and Login:
 - Ensure patients can register successfully and log in.
 - If not registered, ensure patients are prompted to complete the registration process before logging in.

8. End-to-End Workflow (Admin-Doctor):

- Verify that admin-added doctor data is visible and accessible to doctors after login.
- Ensure doctors can view their data and scheduled appointments.

9. End-to-End Workflow (Admin-Patient):

- Verify that admin-generated invoices are visible to patients after login.
- Ensure patients can view their invoices in their dashboard.

10. End-to-End Workflow (Doctor-Patient):

- Verify that prescriptions added by doctors are visible to patients.
- Ensure patients can view prescriptions after logging in.

4.2.4 VALIDATION TESTING

Validation testing ensures that the software system meets the specified requirements and fulfils user needs. It verifies that the system works as intended in real-world scenarios, delivering the expected outcomes. This testing typically includes user acceptance testing (UAT) and system testing, where feedback from users ensures the software aligns with their expectations and business goals.

The “Hospital Management System (Care Heaven)” is validated through validation testing where healthcare professionals test the system to ensure it meets their needs, including features like patient management, doctor scheduling, and invoicing. System and integration testing are conducted to verify that all components, such as admin data management, doctor prescriptions, and patient records, work together seamlessly. The system is also checked for compliance with data privacy and security standards. This ensures that proposed system is reliable, user-friendly, and ready for hospital deployment.

CHAPTER 5: CONCLUSION AND FUTURE SCOPE OF THE PROJECT

Conclusion

The “Hospital Management System (Care Heaven)” is designed to automate and streamline hospital operations, enhancing efficiency and data accuracy across various functions such as patient registration, doctor management, and billing. The system integrates key features like secure login, data management, appointment scheduling, and prescription handling, making it a comprehensive tool for hospital staff and administrators. Through the use of open-source technologies and a structured development approach, the system has been successfully implemented to address common issues in traditional manual hospital management, such as data redundancy, delays, and inefficiencies. With its user-friendly interface and centralized data storage, the proposed system aims to significantly improve hospital management processes, reduce administrative workload, and enhance patient care.

Future Scope

The “Hospital Management System (Care Heaven)” can be further developed and enhanced in several ways. Future updates may include:

1. **Mobile Application:** A mobile version for easy access by patients and doctors.
2. **Advanced Analytics:** Data analytics for better decision-making and resource optimization.
3. **External System Integration:** Connecting with external systems like lab and pharmacy management.
4. **Telemedicine:** Enabling virtual consultations for improved patient access.
5. **AI Integration:** Incorporating AI for predictive patient care and automated scheduling.

These enhancements would further streamline hospital operations and improve overall healthcare delivery.

CHAPTER 6: BIBLIOGRAPHY

BOOKS REFERRED:

- Fundamentals of Software Engineering, Fourth Edition by Rajib Mall
- Fundamentals of Database System, 5th Edition, Pearson by Elmasri, R., Navathe, S.B.

WEBSITES:

- <https://www.youtube.com/watch?v=1SnPKhCdlsU&t=7323s>
- <https://www.w3schools.com/>
- <https://www.youtube.com/watch?v=IqrBr-KT5qk&list=PLR5xnAxI5K55BhYlWb7CqiL7KGSIEOlXO>
- <https://app.diagrams.net/>