CHAPTER 1

INTRODUCTION

This project deals with the Corporate Medicare Management. This project is very helpful to both Medicare staff as well as to the public. It is having mainly Administration and Client modules.

The growing quality demand in the hospital sector makes it necessary to exploit the whole potential of stored data efficiently, not only the clinical data, in order to improve diagnoses and treatments, but also on management, in order to minimize costs and improve the care given to the patients.

However, data mining and the overall process, known as Knowledge Discovery from Databases (KDD), is usually an expensive process, especially in the stages of business objectives elicitation, data mining objectives elicitation, and data preparation. This is especially the case each time data mining is applied to a hospital: many meetings have to been held with the direction of the hospital, area coordinators, computer scientists, etc., to establish the objectives, prepare the data, the mining views and for training the users to general DM tools.

1.1 PURPOSE

In Medicare management situations we are dealing with Data Mining objectives such as:

1. To optimize bed occupation.
2. To improve the use of operating theatres, avoiding the cancellation of operations.
3. To know how emergencies affect to the administration of the hospital departments or services (cancellation of operations, etc.)
4. To optimize the allocation of human and material resources towards and shifts.
5. To detect the influence of certain diseases in the hospital’s services.
6. To find clusters of patients.

1.2 DEFINING THE PROBLEM

Defining the problem in a hospital management project involves identifying and clearly stating the specific challenges or issues that the project aims to address. Here are some steps to help define the problem effectively:

1. **Identify the Scope:** Determine the scope of your hospital management project. Are you focusing on a particular department (like emergency services or outpatient care) or looking at hospital-wide issues (like patient flow or resource management)

2. **Gather Information:** Collect relevant data and information about current challenges faced by hospitals. This might include issues such as inefficient patient scheduling, long waiting times, and ineffective communication between departments, high readmission rates, or inadequate resource allocation.

3. **Prioritize Issues:** Once you have a list of potential problems, prioritize them based on their impact on hospital operations, patient care, and overall efficiency. Consider factors like severity, frequency, and feasibility of addressing each issue.

4. **Define Specific Problems:** Narrow down your focus to one or two specific problems that you want your project to tackle. For example, instead of "inefficient patient scheduling," you might refine the problem statement to "reducing wait times for outpatient appointments by optimizing scheduling processes.

5. **Formulate Problem Statements:** Develop clear and concise problem statements that describe the identified issues. A good problem statement should be specific, measurable, achievable, relevant, and time-bound (SMART). For instance, "The current patient discharge process is inefficient, leading to delays and increased hospital costs.

6. **Consider Stakeholder Perspectives:** Understand the perspectives of various stakeholders (hospital administrators, medical staff, and patients) to ensure that the defined problems align with their concerns and priorities.

7. **Validate and Refine:** Validate the identified problems by consulting with key stakeholders and experts in hospital management. Refine the problem statements based on feedback and additional insights gained during this process.

By clearly defining the problem at the outset of your hospital management project, you set a solid foundation for developing effective strategies and solutions to improve hospital operations and patient care.

1.3 Existing System

- Integration of Corporate Medicare centers is very difficult while it is having different branches.
- In most of the cases the database is similar from one hospital to another hospital. In those cases also we can’t easily adapt a new technology in the new hospital.
- It is very difficult to analyze the usage percentage of hospital resources, Bed occupation Ratio, Administration, Laboratory information even in a single center. Then we can expect the complexity while integrating multi multi-specialty Medicare Centers.
- Room Reservations, Doctor Appointment Schedules, Operation Schedules, and Medicine indentation information is very difficult to maintain and share among the different Medicare Centers.
- Lack of generic and unique model we have to implement the same set of data model for every newly established Medicare Center.

1.4 Proposed System

In this project we are trying to implement which parts of a data-mining project for hospital management are equal or highly similar across different hospitals (at least in the same national healthcare system). This allows us to design several data mining modules, which can be portable across several hospitals, thus dramatically reducing the time to implement a data-mining program in a new hospital.

1.5 OBJECTIVES: The objectives of hospital management encompass a range of goals aimed at ensuring efficient operations, excellent patient care, financial sustainability, and overall organizational success. Here are some key objectives of hospital management:

1. **Optimal Patient Care:**
   - Ensure the delivery of high-quality healthcare services that meet patient needs and expectations.
   - Improve patient outcomes, satisfaction, and safety through effective clinical practices and protocols.

2. **Operational Efficiency:**
    - Enhance operational processes to minimize waiting times, reduce inefficiencies, and streamline workflows.
    - Optimize resource utilization, including staff, equipment, and facilities, to improve productivity and cost-effectiveness.
3. **Financial Sustainability:**
    - Maintain financial stability and profitability through effective revenue cycle management, budgeting, and cost control.
    - Implement strategies to maximize reimbursement, reduce expenses, and improve financial performance.
4. **Staff Development and Engagement:**
    - Foster a positive work environment that supports staff well-being, professional growth, and job satisfaction.
    - Provide continuous training and development opportunities to enhance staff competency and performance.
5. **Compliance and Regulatory Adherence**:
    - Ensure adherence to healthcare regulations, standards, and accreditation requirements to maintain legal and ethical practices.
    - Implement robust compliance programs and quality assurance measures to uphold patient safety and regulatory compliance.
6. **Technology Integration**:
    - Adopt and leverage healthcare technologies (like EMR/EHR systems, telemedicine, AI-driven diagnostics) to improve clinical decision-making, patient care, and operational efficiency.
    - Enhance communication and information sharing among healthcare providers, patients, and stakeholders.
7. **Community and Stakeholder Engagement**:
    - Build strong relationships with the community, patients, families, and other stakeholders to foster trust and transparency.
    - Collaborate with healthcare partners, including physicians, insurers, and government agencies, to enhance healthcare delivery and patient outcomes.
8. **Continuous Improvement**:
    - Implement a culture of continuous improvement through data-driven decision-making, performance metrics, and feedback mechanisms.
    - Regularly evaluate processes, outcomes, and patient feedback to identify areas for enhancement and innovation.
9. **Emergency Preparedness and Disaster Management**:
    - Develop and maintain robust emergency preparedness plans to respond effectively to disasters, epidemics, or other emergencies.
    - Ensure readiness to handle crises while maintaining continuity of care and safeguarding patient and staff safety.
10. **Ethical and Patient-Centered Care**:
    - Uphold ethical standards and principles in patient care, decision-making, and organizational practices.
    - Promote patient- centered care that respects patient rights, dignity, and cultural diversity.

1.6 Frontend and Backend

Frontend: - Front-end development is the development of visual and interactive elements of a website that users interact with directly. It's a combination of HTML, CSS and JavaScript, where HTML provides the structure, CSS the styling and layout, and JavaScript the dynamic behavior and interactivity.

HTML: Html stands by hypertext markup language. It is use to create the structure of the webpage.

Key HTML Elements

- **<! DOCTYPE html>**: Declaration of the HTML version being used (HTML5 in this case).
- **<html>**: Root element that wraps all content on the webpage.
- **<head>**: Contains meta-information about the document, such as <title>, <Meta>, <link>, and <style> tags.
- **<meta charset="UTF-8">**: Defines the character encoding for the document (UTF-8 supports most characters worldwide).
- **<title>**: Sets the title of the webpage, displayed in the browser's title bar or tab.
- **<body>**: Contains the visible content of the webpage, such as headings, paragraphs, images, links, etc.
- **<h1> to <h6>**: Headings of different levels (1 being the highest, 6 being the lowest).
- **<p>**: Paragraph element for text content.
- **<img>**: Embeds an image with attributes like src (source URL) and alt (alternative text for accessibility).
- **<a>**: Anchor element for creating hyperlinks with href attribute pointing to the destination URL.

CSS: CSS stands by cascading style sheet. CSS are used to styling of the webpage. There are three types off CSS are used:

1. Inline CSS
2. Internal CSS
3. External CSS

JAVA SCRIPT: JavaScript is not a programming language it is a scripting language. It is mostly used frontend and backend of a webpage. JavaScript is the world's most popular programming language. JavaScript is the programming language of the Web. JavaScript is easy to learn.

Backend: - Back end (computing), the data access layer in software architecture. Back-end CASE. Back-end database, a database accessed indirectly through an external application. Back-end processor, hardware that stores and retrieves data from a database.

1.7 FEASIBILITY STUDY: and opportunities before implementation. Conducting a feasibility study for a hospital management project is crucial to assess its viability, potential challenges here are key components typically included in a feasibility study for a hospital management project:

1. Market Analysis:

- **Demand Assessment:** Evaluate the current and projected demand for healthcare services in the target area or community. Consider demographics, population growth, and healthcare needs.
- **Competitive Landscape:** Analyze existing hospitals and healthcare providers in the area, their services, strengths, weaknesses, and market positioning.
- **Market Trends:** Identify trends in healthcare delivery, technology adoption, patient preferences, and regulatory changes that may impact the project.

2. Technical Feasibility:

- **System Requirements:** Define the technical infrastructure needed for the hospital management system (e.g., hardware, software, and networking).
- **Compatibility:** Ensure compatibility with existing hospital systems (like EMR/EHR, billing systems) and integration capabilities with third-party applications.
- **Scalability:** Assess the ability of the system to scale with the hospital's growth and evolving needs.

3. Operational Feasibility:

- **Workflow Analysis:** Understand current hospital workflows and identify areas for improvement or automation.
- **Staff Training:** Evaluate the readiness of staff to adopt new technologies and processes. Plan for training programs to ensure smooth implementation.
- **Change Management:** Assess organizational readiness and plan strategies to manage resistance to change and ensure acceptance of new systems and processes.

4. Financial Feasibility:

- **Cost-Benefit Analysis:** Estimate the costs associated with implementing the hospital management system (development, implementation, training, maintenance) versus the expected benefits (cost savings, revenue generation, improved efficiency).
- **Revenue Generation:** Evaluate potential revenue streams from enhanced services, increased patient volume, or improved billing and collections.
- **ROI Calculation:** Calculate the return on investment (ROI) and payback period to determine financial feasibility and sustainability.

5. Legal and Regulatory Feasibility:

- **Compliance:** Ensure compliance with healthcare regulations, licensing requirements, data privacy laws (such as HIPAA), and accreditation standards (like JCI or ISO).

- **Risk Management:** Identify legal and regulatory risks associated with the project and develop mitigation strategies.

6. Environmental and Social Feasibility:

- **Environmental Impact:** Assess the environmental impact of the hospital project (e.g., waste management, energy consumption) and ensure compliance with environmental regulations.
- **Community Impact:** Evaluate the project’s social impact on the community, including accessibility, affordability, and healthcare equity considerations.

7. Schedule and Timeline:

- **Project Planning:** Develop a detailed project plan with timelines for each phase of the feasibility study, system development, testing, and deployment.
- **Critical Path Analysis:** Identify critical tasks and dependencies to ensure timely project completion.

8. Conclusion and Recommendations:

- Summarize findings from the feasibility study, including strengths, weaknesses, opportunities, and threats (SWOT analysis).
- Provide recommendations regarding the viability of the hospital management project, potential risks, and strategies for successful implementation.

CHAPTER 2

SYSTEM ANALYSIS & DESIGN

2.1 MODEL USED: - In a hospital management system (HMS) project, several models and methodologies can be utilized during different phases of the project lifecycle to ensure its successful development and deployment. Here are some key models commonly used:

1. **Waterfall Model**: This traditional sequential model is suitable when requirements are well-defined and stable. It progresses through phases such as requirements gathering, design, implementation, testing, deployment, and maintenance in a linear fashion.
2. **Iterative and Incremental Model**: This model involves breaking down the project into smaller parts (iterations) and developing them incrementally. Each iteration includes requirements analysis, design, implementation, and testing. It allows for feedback and adjustments throughout the development process.
3. **Agile Methodologies (e.g., Scrum, Kanban)**: Agile methodologies emphasize flexibility, collaboration, and iterative development. Scrum involves time-boxed iterations (sprints) where features are developed and tested, with regular feedback from stakeholders. Kanban focuses on visualizing the workflow and limiting work in progress.
4. **Unified Modeling Language (UML)**: UML diagrams, such as use case diagrams, class diagrams, sequence diagrams, and activity diagrams, are used to visualize and document different aspects of the HMS. They help in understanding requirements, designing the system architecture, and modeling workflows.
5. **Prototyping Model**: Prototyping involves creating early, simplified versions of the system to gather feedback from users and stakeholders. It helps in refining requirements, designing user interfaces, and validating design decisions before full-scale development.
6. **Data Modeling**: Entity-Relationship Diagrams (ERDs) and Data Flow Diagrams (DFDs) are essential for modeling the data structure and flow within the HMS. ERDs define entities like patients, doctors, and treatments, along with their relationships. DFDs illustrate how data moves through different processes in the system.
7. **Process Modeling**: Process modeling using tools like Business Process Model and Notation (BPMN) helps in defining and optimizing workflows within the HMS, such as patient registration, appointment scheduling, and billing processes.
8. **Decision Support Systems (DSS)**: DSS models and techniques, such as data mining and predictive analytics, can be integrated into the HMS to analyze patient data, optimize resource allocation, and support clinical decision-making.
9. **Security Models**: Security models like Role-Based Access Control (RBAC) and encryption techniques are crucial for protecting sensitive patient information and ensuring compliance with healthcare regulations (e.g., HIPAA).
10. **Quality Assurance Models**: Quality assurance models and techniques, including test planning, test cases, and test automation frameworks, are essential for ensuring the reliability, performance, and usability of the HMS.

2.2 SRS (SYSTEM REQUIREMENTSPECIFICATION):

2.2.1Requirement Analysis:

A requirement is a feature that must be included in the system. Before the actual design and implementation start, getting to know the system to be implemented is of prime importance.

Main emphasis should be on**:**

- Inputs enter into the system.
- Standard Encryption of Input on submit
- The outputs expected from the system.
- The people involved in the working of the system.
- The volume of DATA (INPUT) and the amount of Information (OUTPUT) that will be involved with respect to the system itself, the following facts should be taking into consideration.

The Major process involved:

- The main points of the application.
- The processing rules fort he collected data.
- The exceptions that may be present.
- That checks that should be in place in order to avoid wrong entries.

2.2.2 Software Requirement Specification

OPERATING SYSTEM	:	WIN 98/2000/XP, UNIX/LINUX
DATA BASE	:	ORACLE
SOFTWARE	:	APACHE TOMCAT

FRONT END TOOL	:	DHTML L
SCRIPTING LANGUAGE	:	JAVA SCRIPT
WEB COMPONENTS	:	SERVLETS, JSP
DATA MINING TOOL	:	WEKA

2.2.3Hardware Requirements Specification

PROCESSOR	:	Pentium-IV
PROCESSOR SPEED	:	2.4GHZ
MONITOR	:	COLOR MONITOR
HARD DISK	:	40GB
RAM	:	512MB
MOUSE	:	SCROLLING MOUSE
KEY BOARD	:	MM KEY BOARD

2.2.4 Communication protocols

2.2.4.1 TCP/IP protocol should be installed.

2.2.4.2 Any browser should be installed (Internet explorer 6.0 or Netscape navigator 8.0)

2.2.4.3 HTTP 1.1 should be present on the system.

2.2.4.4 Internet connection should be present in order to access the site.

2.2.4.5 Internal modem or NIC card should be present

2.2.5 Requirement study

The origin of most software systems is in the need of a client, who either wants to automate and existing manual system or desires a new software system. The software system itself is created by the developer finally the completed system will be used by the end user. Thus, there are three major parties interested in a new system: the client, the users, and the developer. The requirements for the system that will satisfy the need of the clients and the concerns of the user have to communicate to the developer.

The problem is that the client usually does not understand software or the software development process, and the developer often does not understand the clients problem and application area. This causes a communication gap between the parties involved in the development project. A basic purpose of software requirement specification is to bridge this communication gap. SRS is the medium through which the client and the user need are accurately specified; indeed SRS forms the basis of software development. A good SRS should satisfy all the parties-something very hard to achieve and involves trade-offs and persuasion.

2.2.6 The Requirement Process:

The main reason of modeling generally focuses on the problem structure, not its external behaviors. Consequently, thing's like user interfaces are rarely modeled, whereas thy frequently from a major components of the SRS.

Similarly performance constraints, design constraints, standards compliance, recovery, etc. are specified clearly in the SRS because the designer must know about there to properly design the system.

To properly satisfy the basic goals, an SRS should have certain properties and should contain diff type of req. A good SRS is [IEE87, IEE94]: complete if everything the software is supposed to and responses to the software to all classes of input data are specified in the SRS.

Correctness and completeness go hand in hand an SRS in unambiguous if and only if every requirement stated has one and only one interpretation, requirements often written in natural language.

An SRS is verifiable if and only if every stated requirement is verifiable. A requirement is verifiable if there exists some cost-effective process that can check whether the final software meets those requirements. An SRS is consistent if there is no requirements that with another.

Writing and SRS is an iterative process. Even when requirement of system are specified they are later modified as the need of the client change. Hence an SRS should be easy to modify. An SRS is traceable if the origin of each of its requirements is clear and if it facilitates the referencing of each requirement in future development [EEE87].

One of the most common problems in requirement specification is when some of the requirements of the client are not specified. This necessitates addition and modifications to the requirements later in the development cycle, which are often expensive to incorporate.

2.2.7 Project Schedule Study phase:

In the study phase we do the preliminary investigation and determine the system requirements. We study the system and collect the data to draw the dataflow diagrams. We follow the methods like questions and observation to find the facts that are involved in the process. This is an important because if the specification study is not done properly then design phase etc will go wrongly.

2.2.8 Design Phase:

In this design phase we design the system making use of study phase and the data flow diagrams. We make use the general access for designing.

We consider the top down approach. In the design phase we determine the entities and their attributes and the relationships between the entities. We do both logical and physical design of the system.

2.2.9 Development Phase:

In the development phase we mostly do the coding part following the design of the system. We follow modular programming for development and after development and after developing each and every module we do the unit testing followed by the integration testing.

2.2.10 Implementation Phase:

The last phase of the project is the implementation phase. Quality assurance is the primary motive in this phase. The quality assurance is the review of software products and related documentation for completeness, correctness, reliability and maintainability. The philosophy behind the testing is it finds errors. The testing strategies are of two types, the code testing and the specifications testing. In the code testing we are examining the logic of the program. On the surface, code testing seems to be ideal methods for testing software, but no tall software errors are uncovered.

2.2.11Feasibility Study

Feasibility is an important phase in software development process. It enables the developers to have an assessment of the product being developed. It refers to the feasibility study of product in terms of outcomes of the product, operational use and technical support required for implementation it.

Feasibility study should be performed on the basis of various criteria and parameters. The various feasibility studies are:

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

Economic Feasibility:

It refers to the benefits or outcomes we are deriving from the product as compared to the total cost we are spending for developing the product.

In the present system, the development of new product greatly enhances the accuracy of the system and reduces the delay in the processing of applications and generating the reports. The errors can be greatly reduced and at the same time providing the great level of security. Here we don't need additional equipment except memory of required capacity. No need for spending money on client for maintenance because the database used is web enabled database.

Operational Feasibility:

It refers to the feasibility of the product to be operational. Some products may work very well at design and implementation but may fail in the real time environment. It includes the study of human required and their technical expertise.

In the present system, the entering the details, updating the details and reports generations are perfect and quick in operations.

Technical Feasibility:

It refers to whether the software that is available in the market fully supports the present application .It studies the pros and cons of using particular software for the development and its feasibility. It also studies the additional time needed to be given to people to make the application work.

In the present system the user interface is user friendly and does not require much expertise and training .It just needs mouse click to do operations and to generate reports. The software that is used for developing is highly suitable for the present applications since the users require fast access to the web pages with a high degree of security. This is achieved through integration of web server and database server in the same environment.

2.3 DATA FLOW DIGRAMS (DFDS)

Data Flow Diagrams (DFDs) are widely used in hospital management system (HMS) projects to depict the flow of data within the system, showing how information moves through different processes and entities. Here's how DFDs can be applied in the context of an HMS project:

Levels of DFDs in HMS Project:

1. **Context Level DFD**:
   - **Purpose**: Provides an overview of the entire hospital management system as a single process.
   - **Components**: Typically includes external entities such as patients, doctors, nurses, administrators, and external systems like laboratories or pharmacies.
   - **Processes**: Represents high-level processes such as patient management, appointment scheduling, billing, and reporting.
   - **Data Flows**: Shows high-level data flows between external entities and the system processes, indicating the main inputs and outputs of the system.
2. **Level 0 DFD**:
   - **Purpose**: Breaks down the context level into more detailed processes and data flows.
   - **Components**: Includes major processes identified in the context level DFD as separate processes.
   - **Processes**: Each process represents a major activity within the system, such as patient registration, treatment scheduling, inventory management, and billing.
   - **Data Stores**: Represents databases or data repositories where information is stored within the system, such as patient records, medical histories, and inventory databases.
   - **Data Flows**: Shows detailed data flows between processes and data stores, indicating how data is passed from one process to another and stored within the system.
3. **Lower-Level DFDs (Level 1 and beyond)**:
   - **Purpose**: Further decomposes processes identified in the level 0 DFD into more detailed sub processes and data flows.
   - **Components**: Continues to break down processes into smaller sub processes as needed for detailed analysis and design.
   - **Processes and Data Flows**: Provides a more granular view of specific activities within each major process, detailing inputs, outputs, and data transformations.

Example Processes and Data Flows in HMS:

- **Patient Registration Process**:
  - **Inputs**: Patient information (name, address, insurance details).
  - **Processes**: Validation of patient data, generation of patient ID, storing information in the patient database.
  - **Outputs**: Confirmation of registration, patient ID.
- **Appointment Scheduling Process**:
  - **Inputs**: Patient details, doctor availability.

- **Processes**: Scheduling algorithm, confirmation with patient and doctor, updating schedules.
- **Outputs**: Scheduled appointment details.

- **Billing Process**:
  - **Inputs**: Services provided (consultation, treatment), patient ID.
  - **Processes**: Calculation of charges, verification with insurance, generation of bill.
  - **Outputs**: Billing statement.
- **Medical Record Management Process**:
  - **Inputs**: Patient ID, medical data (diagnosis, prescriptions).
  - **Processes**: Updating medical records, access control.
  - **Outputs**: Updated medical records.

Benefits of Using DFDs in HMS Projects:

- **Visualization**: DFDs provide a clear and structured visual representation of how data flows through the system, aiding in understanding and communication among stakeholders.
- **Analysis**: Helps in identifying data redundancies, inefficiencies, and opportunities for optimization within processes.
- **Design**: Provides a foundation for system design, facilitating the development of software modules and databases based on identified processes and data flows.
- **Documentation**: Serves as documentation for system requirements, assisting in future maintenance and enhancements of the HMS.

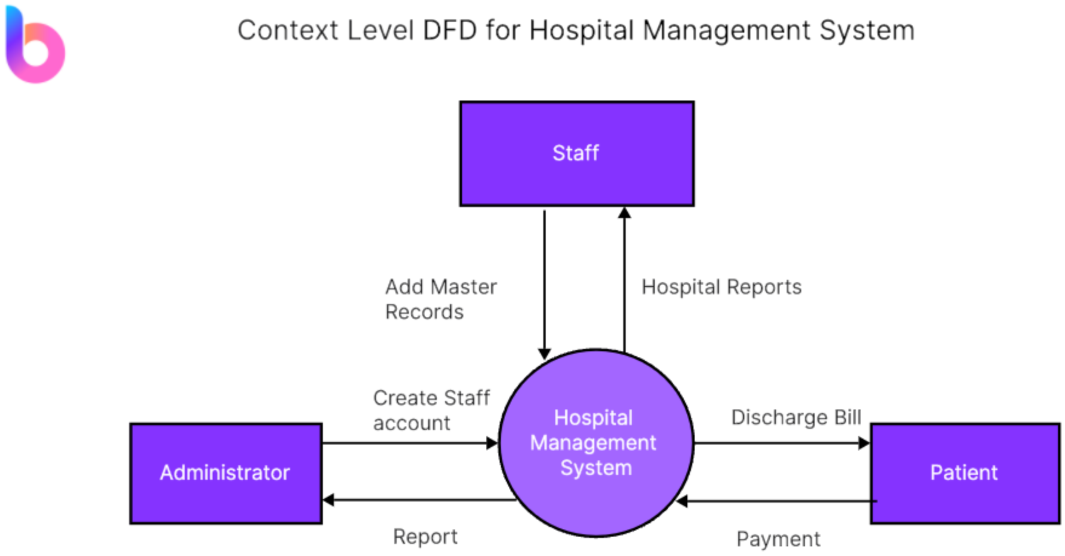


Fig 1 context level DFD for HMS

2.4 Use case Diagrams

Use case Diagrams represent the functionality of the system from a user's point of view. Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behavior of the system from external point of view.

Actors are external entities that interact with the system. Examples of actors include users like administrator, bank customer …etc., or another system like central database

2.4.1Class Diagrams

Class diagrams are widely used to describe the types of objects in a system and their relationships. These model the class structure and contents using elements such as classes, packages and objects. Class diagrams describe three different perspectives when designing a system. Conceptual, specification and implementation.

2.4.2 Interaction Diagrams

Sequence diagrams and Collaboration diagrams both are called as interaction diagrams. These are two of the five diagrams used in the UML for modeling the dynamic aspects of the systems. An interaction diagrams shows an interaction, consisting of a set of objects and their relationships, including the messages that may be dispatched among them.

2.4.3 Sequence Diagrams

A sequence diagram shows, as parallel vertical lines ("life lines"), different processes or objects that live simultaneously and horizontal arrows, the messages exchanged

Between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

2.4.4 Collaboration Diagrams

A collaboration diagram emphasizes the organization of the objects that participate in an interaction. There are two distinguish features from sequence diagram to represent, first, there is the path to indicate how one object is linked to other and second is sequence number, to indicate the time order of messages by prefix with number. To illustrate the concept of autonomous DDM, we performed an experiment using GMM as the global model to demonstrate how to reach the optimal trade-off between the overall data mining quality and the local source's data granularity levels via self-organization. Instead of assuming that the privacy-control component is passive, we implemented the local data sources with the autonomous property to negotiate with the global broker service regarding which data abstraction level to present. The global broker first requests a data abstraction with coarse granularity from each local source. Then, it actively requests more specific details from those sources on a need-to-know basis so that it can learn the global model in a cost-effective manner.

The global brokering service can send the local sources the global model learned up to a specific moment, for example, and the local sources can then return their bid values computed based on the local data likelihood (defined as the product of the probabilities of generating the data) gained per unit cost by advancing one more level of granularity at the local sources. The global service will ask for more data details from the source with the highest value returned. This protocol continues until the data likelihood stops improving significantly or the computational budget runs out.

Use case Diagram for Admin

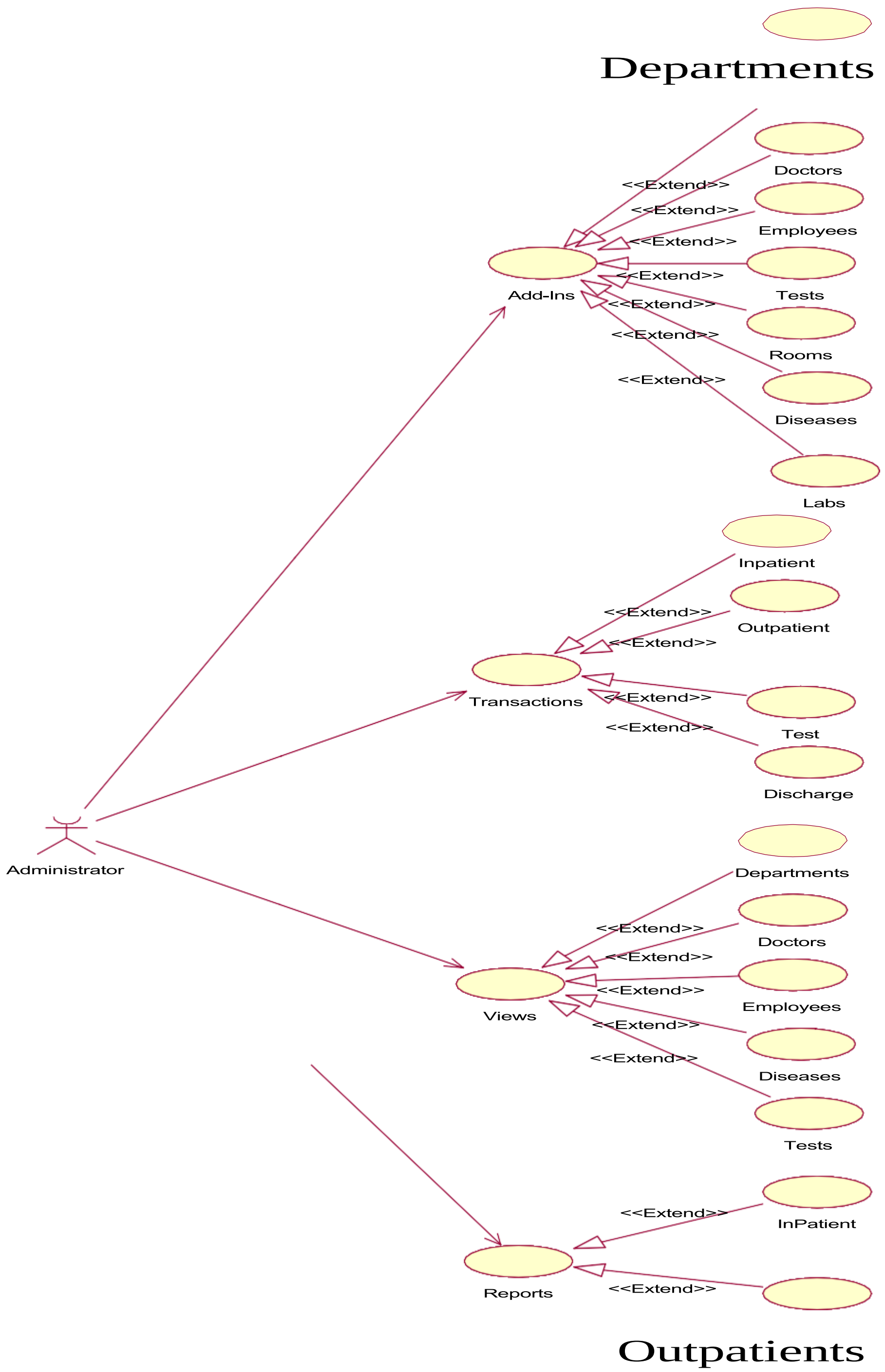


Figure 4: Use case for admin

Use case Diagram for Patient

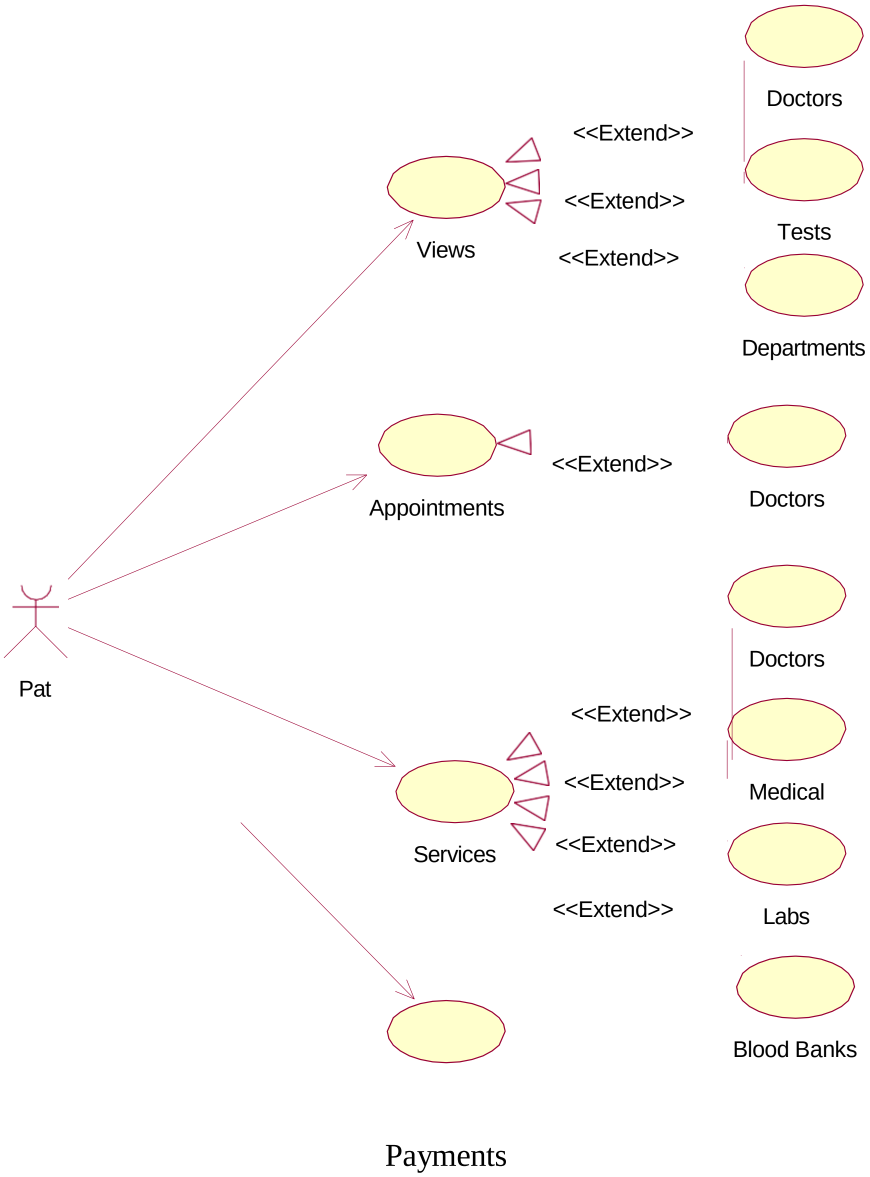


Figure 5: Use case for Patient

Use case Diagram for Patient

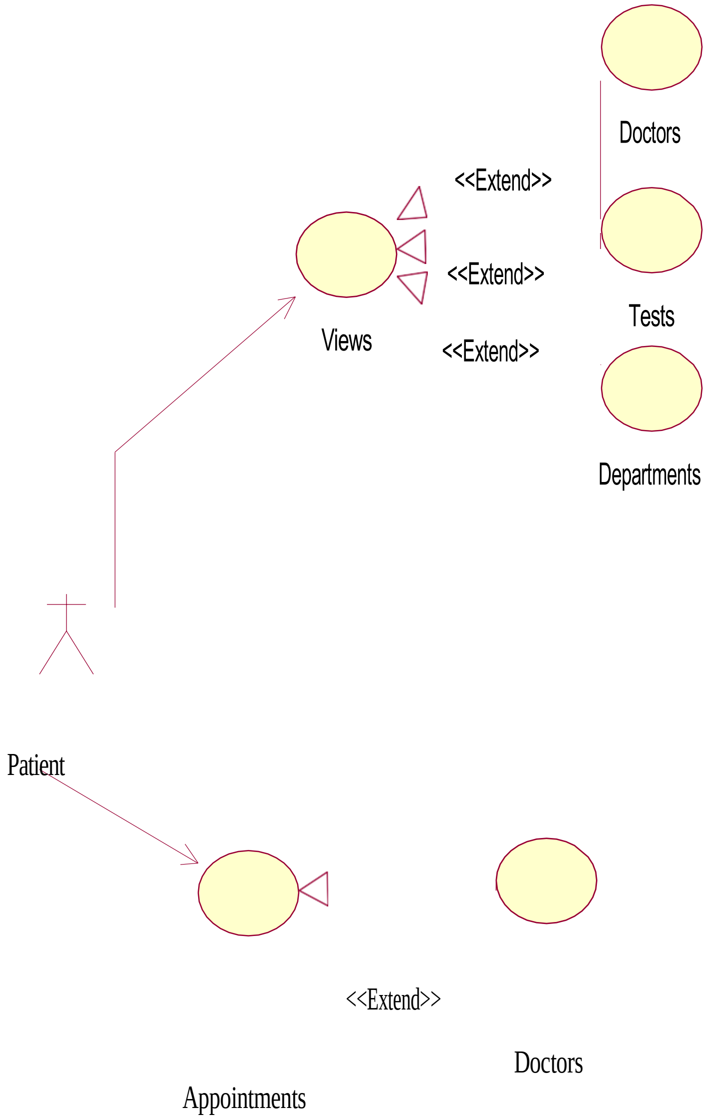


Figure 6: Use case for Public

Class Diagram for Medicare

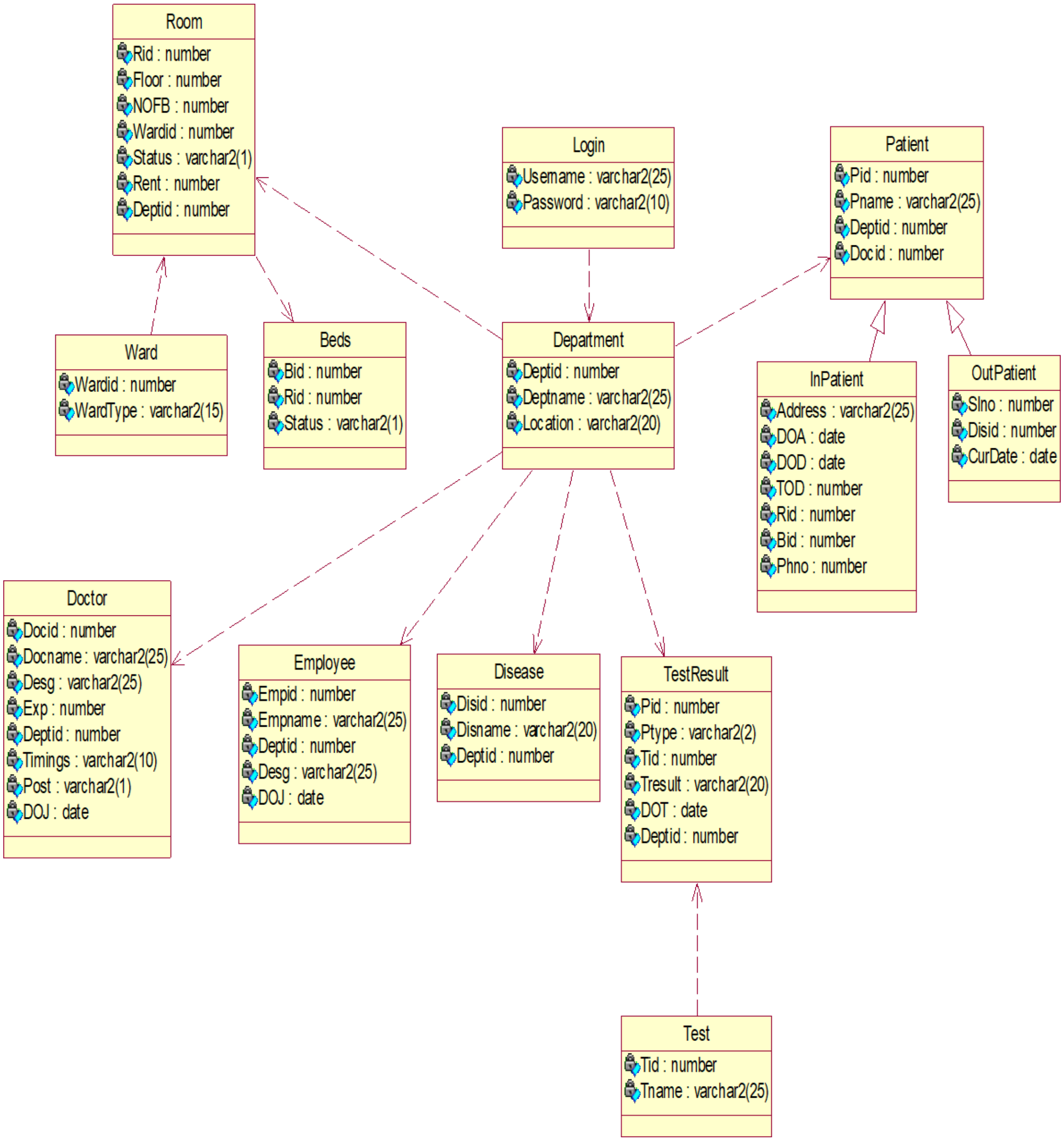


Figure 7: Class Diagram for Medicare

Sequence Diagram for patient

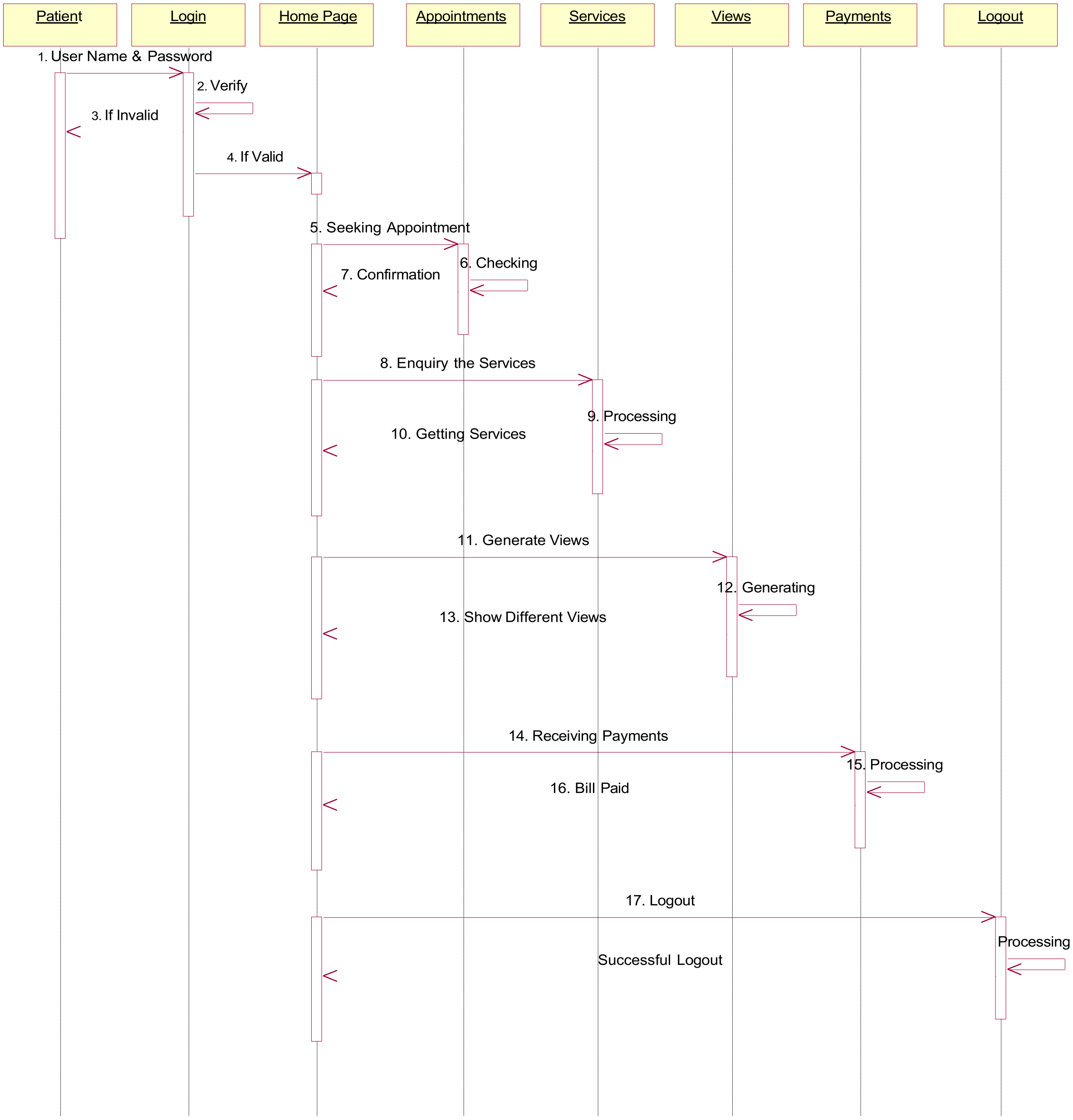


Figure: 8 Sequence Diagram for patient

Sequence Diagram for public

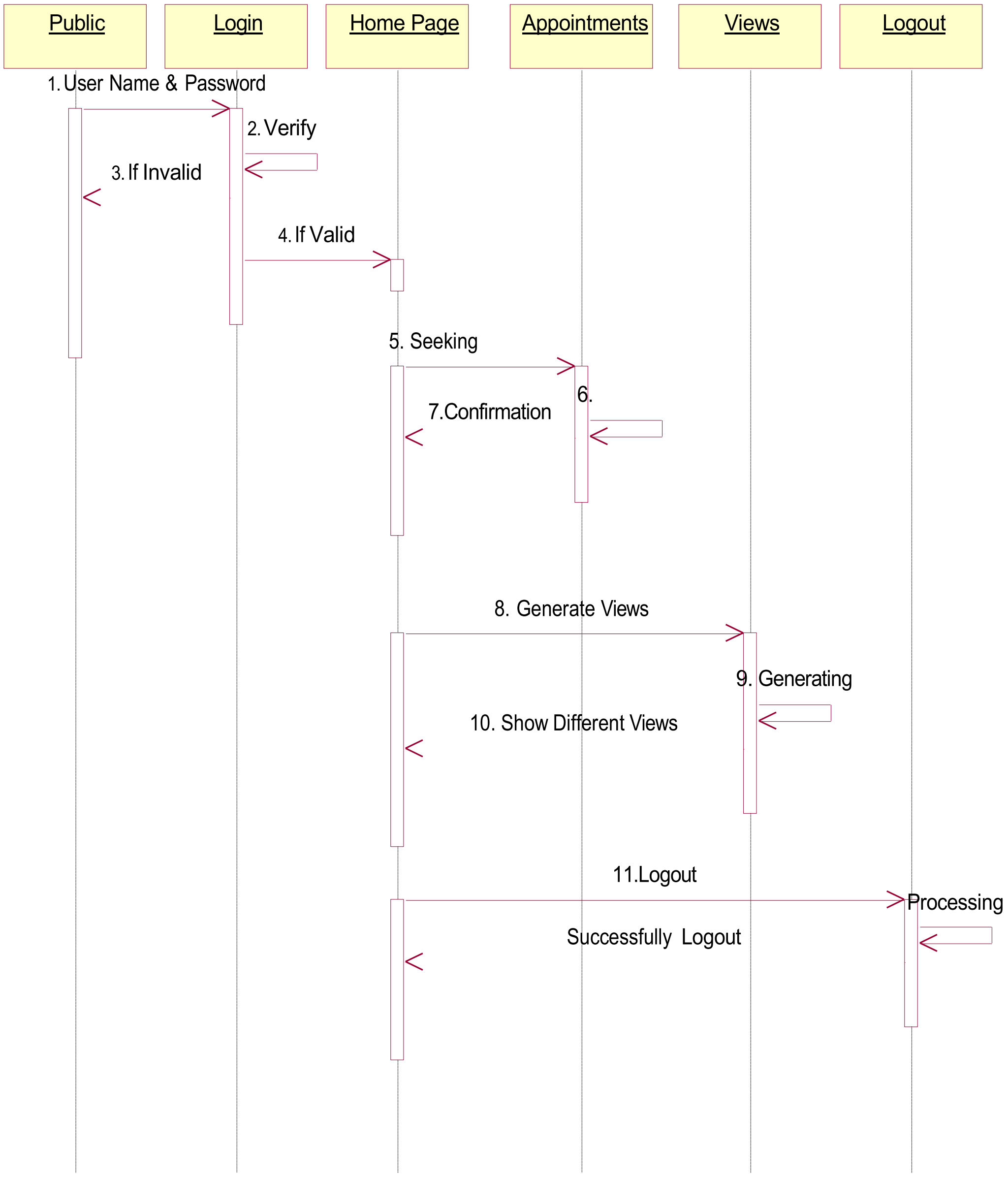


Figure: 9 Sequence Diagram for public

Collaboration Diagram for Admin

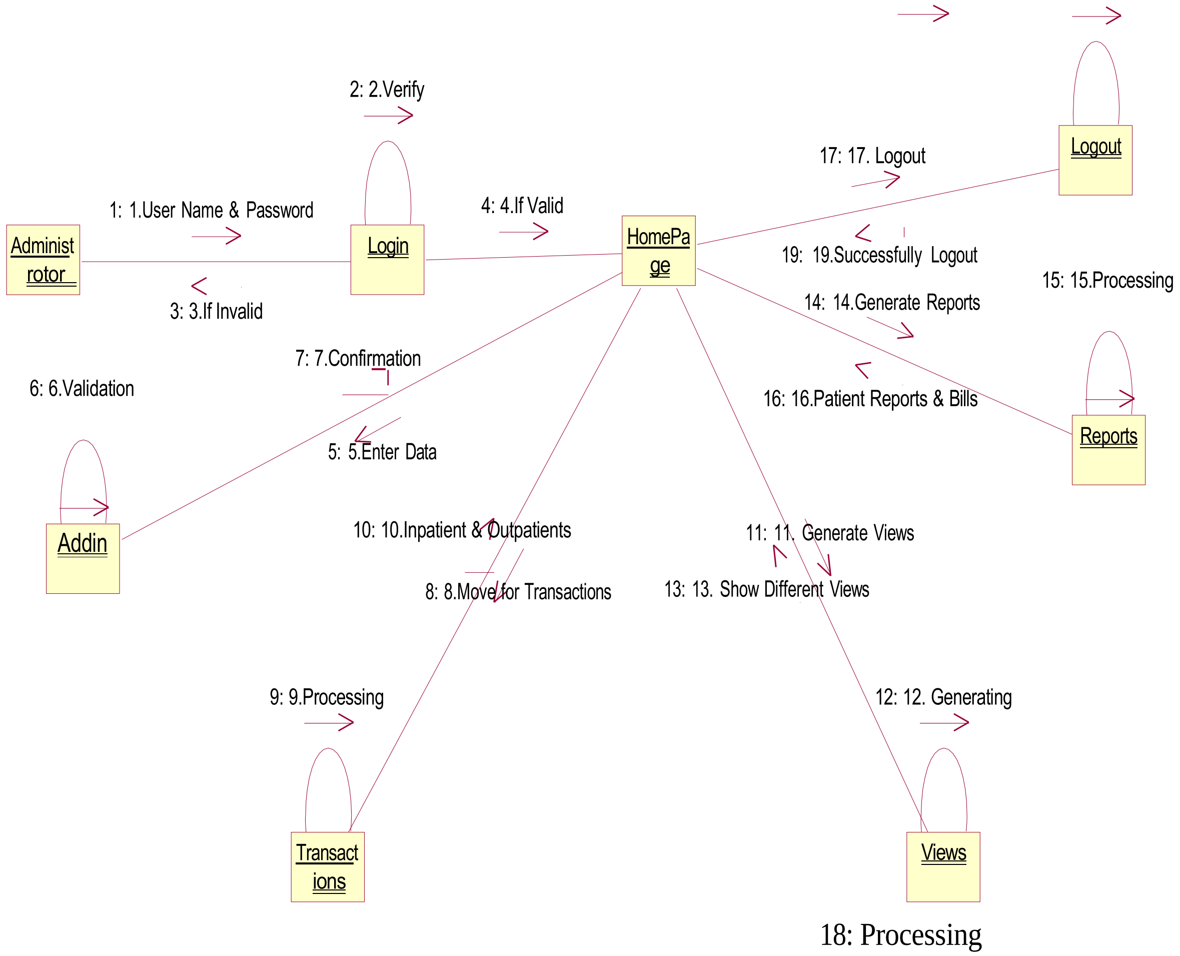


Figure: 10 Collaboration Diagram for Admin

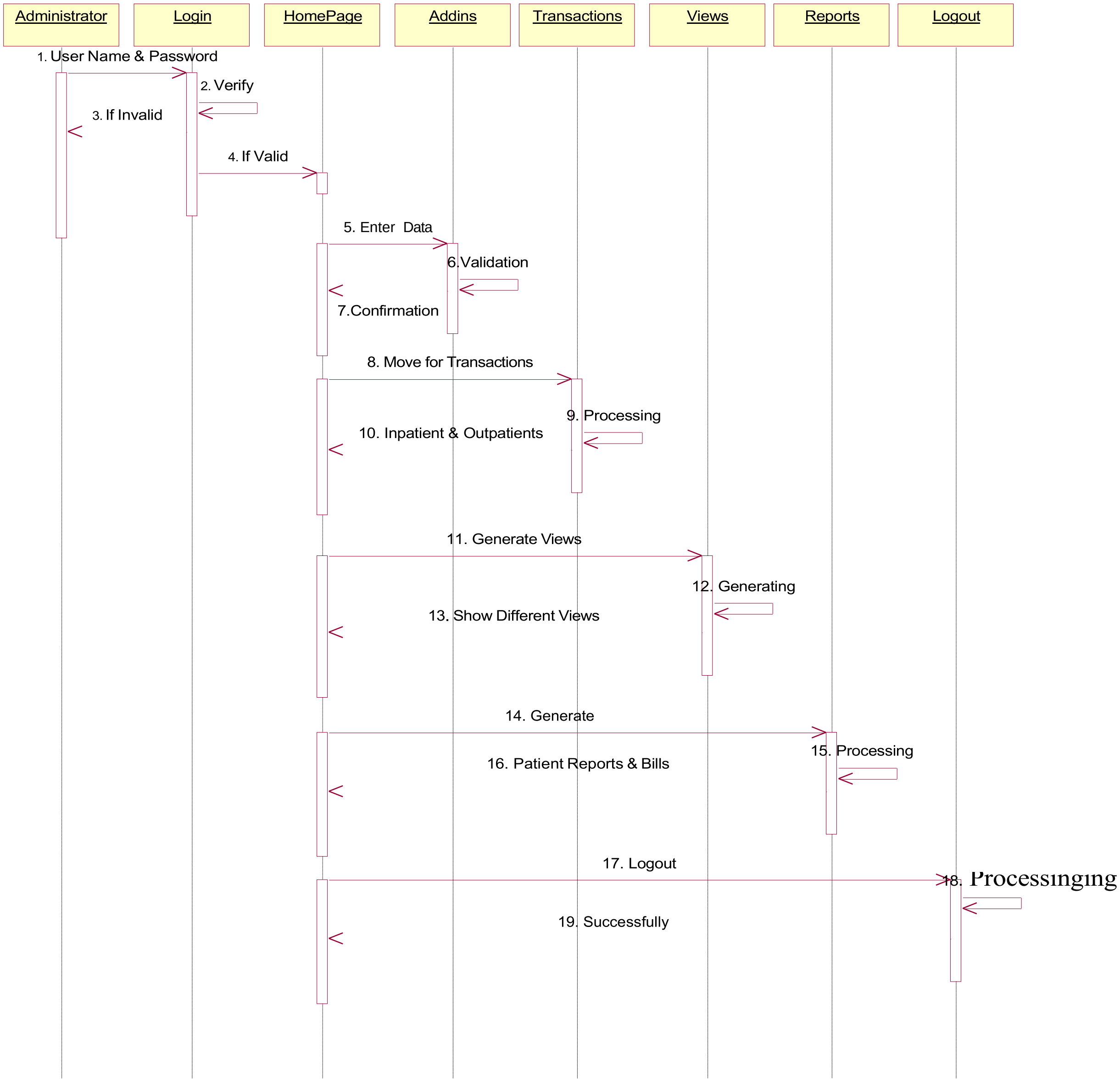


Fig 11 Collaboration Diagram for Patient

Collaboration Diagram for public

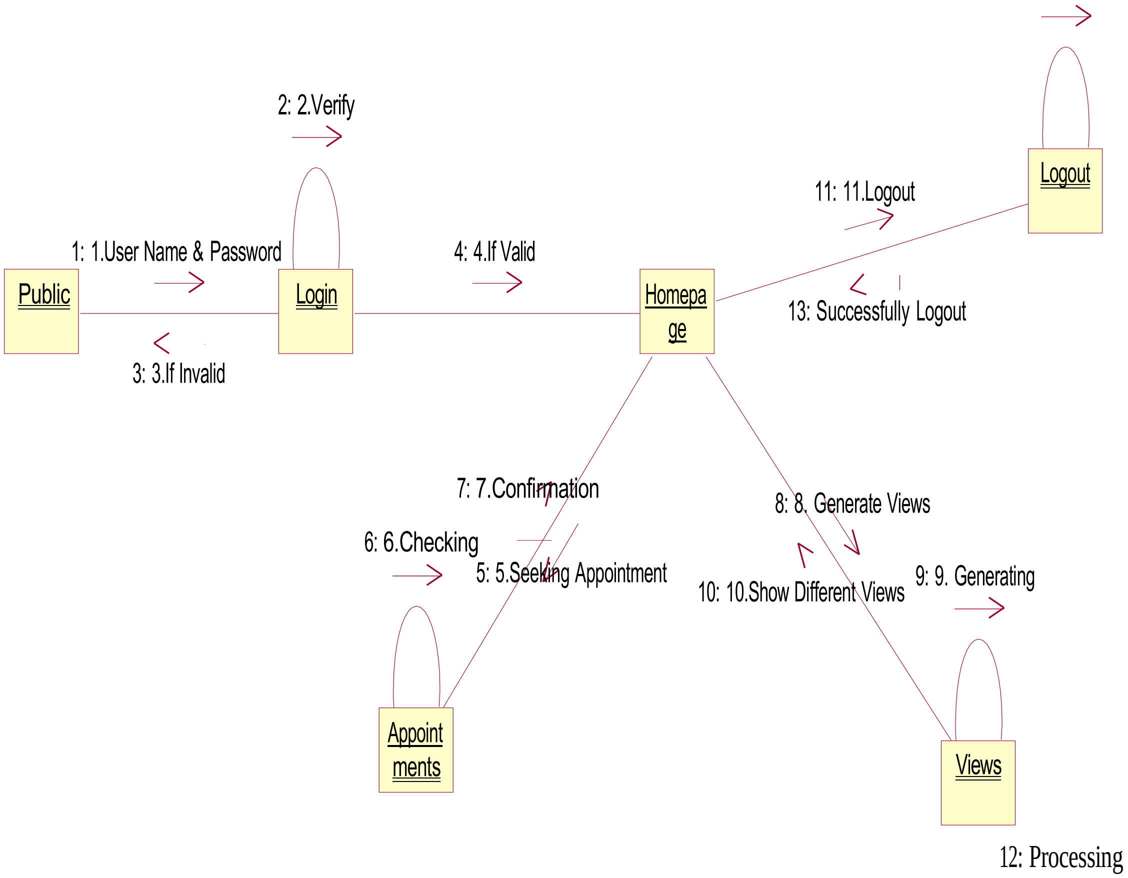


Figure: 12 Collaboration Diagram for public

State chart Diagram for Admin

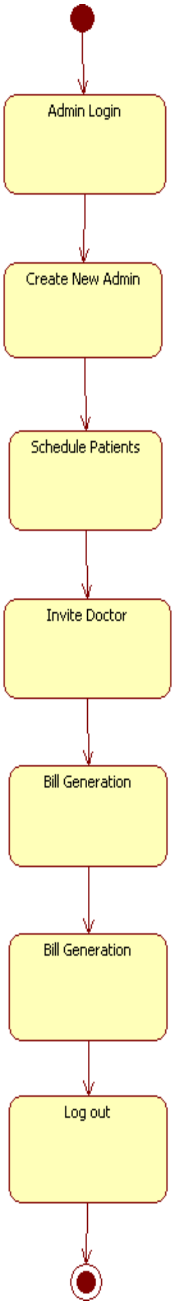


Figure 13: State chart Diagram for Admin

State Chart Diagram for Patient

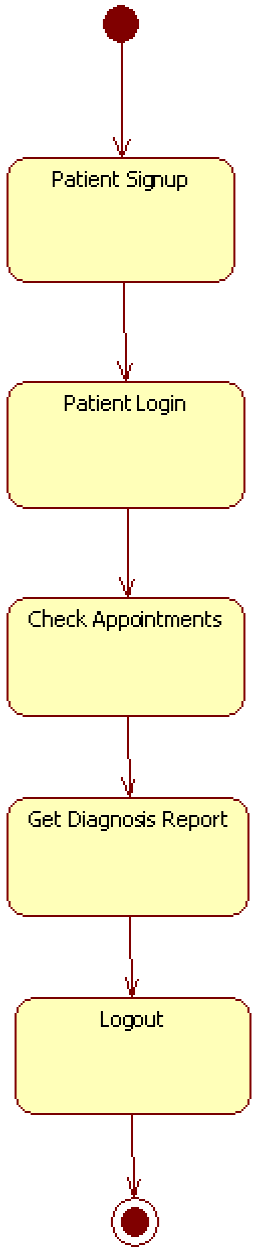


Figure 14 State Chart Diagram for Patient

State Chart Diagram for Doctor

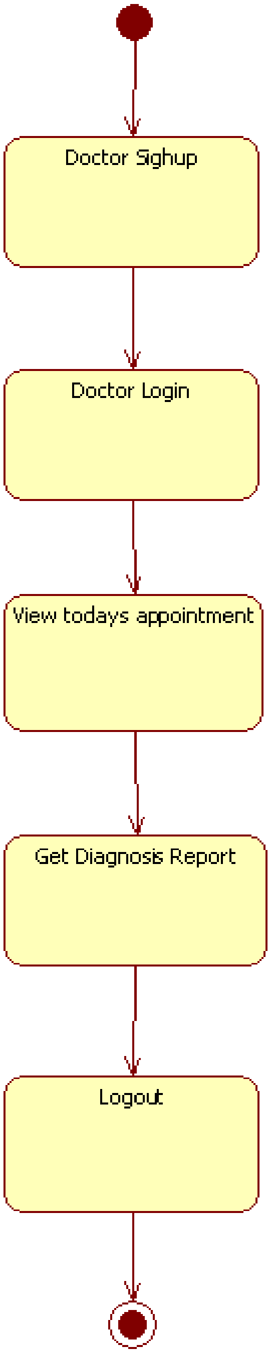


Figure 15 State Chart Diagram for Doctor

CHAPTER 3

WORKING OF PROJECT

3.1 INTRODUCTION TO PROJECT:

The project Hospital Management system includes registration of patients, storing their details into the system, and also computerized billing in the pharmacy, and labs. The software has the facility to give a unique id for every patient and stores the details of every patient and the staff automatically. It includes a search facility to know the current status of each room. Users can search the availability of a doctor and the details of a patient using the id. The Hospital Management System can be entered using a username and password. It is accessible either by an administrator or receptionist. Only they can add data into the database. The data can be retrieved easily. The interface is very user-friendly. The data are well protected for personal use and make the data processing very fast. Hospital Management System is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals. Hospital Management System is designed for multispecialty hospitals, to cover a wide range of hospital administration and management processes. It is an integrated end-to-end Hospital Management System that provides relevant information across the hospital to support effective decision-making for patient care, hospital administration, and critical financial accounting, in a seamless flow. Hospital Management System is a software product suite designed to improve the quality and management of hospital management in the areas of clinical process analysis and activity-based costing. Hospital Management System enables you to develop your organization and improve its effectiveness and quality of work. Managing the key processes efficiently is critical to the success of the hospital helps you manage your processes.

3.2 TECHNOLOGY USED IN THE PROJECT:

3.2.1 JAVA SCRIPT

JavaScript is a script-based programming language which was developed by Netscape Communication Corporation. JavaScript was originally called Live Script and renamed as JavaScript to indicate its relationship with Java.

JavaScript supports the development of both client and server components of Web- based applications. On the client side, it can be used to write programs that are executed by a Web browser within the context of a Web page. On the server side, it can be used to write web server programs that can process information submitted by a Web browser and then updates the browser's accordingly.

Even though **JavaScript** supports both client and server Web programming, we prefer JavaScript at Client side programming since most of the browsers supports it.

JavaScript is almost as easy to learn as HTML, and JavaScript statements can be included in HTML documents by enclosing the statements between a pair of scripting tags <SCRIPT>...</SCRIPT>.

```
<SCRIPT LANGUAGE="JavaScript">

JavaScript statements

</SCRIPT>
```

Here are a few things we can do with JavaScript:

- Validate the contents of a form and make calculations.
- Add scrolling or changing messages to the Browser's status line.
- Animate images or rotate images that change when we move the mouse over them.
- Detect the browser in use and display different content for different browsers.
- Detect installed plug-ins and notify the user if a plug-in is required.

We can do much more with JavaScript, including creating entire application.

3.2.2 HTML:-

Hypertext Markup Language (HTML), the languages of the World Wide Web (WWW), allows users to produces Web pages that include text, graphics and pointer to other Web pages (Hyperlinks). **HTML** is not a programming language but it is an application of ISO Standard 8879, **SGML (Standard Generalized Markup Language),** but specialized to hypertext and adapted to the Web. The idea behind Hypertext is that instead of reading text in rigid linear structure, we can easily jump from one point to another point. We can navigate through the information based on our interest and preferences. A markup language is simply a series of elements, each delimited with special characters that define how text or other items enclosed within the elements should be displayed. Hyperlinks are underlined or emphasized works that load to other documents or some portions of the same documents. **HTML** can be used to display any type of document on the host computer, which can be geographically at a different location. It is a versatile language and can be used on any platform or desktop. **HTML** provides tags (special codes) to make the document look attractive. HTML tags are not case-sensitive. Using graphics, fonts, different sizes, color, etc., can enhance the presentation of the document. Anything that is not tag is part of the document itself.

Basic HTML Tags

<! --- -->	specifies comments
<A>…………</A>	Creates hypertext links
<B>…………. </B>	Formats text as bold
<BIG>…………</BIG>	Formats text in large font
<BODY>……. </BODY>	Contains all tags and text in the HTML document
<CENTER>…. </CENTER>	Creates text
<DD>…………..</DD>	Definitions of a term
<DL>……………</DL>	Creates definition list
<FONT>……. </FONT>	Formats text with a particular font
<FORM>…..</FORM>	Encloses a fill-out form
<FRAME>…. </FRAME>	Defines a particular frame in a set of frames

<H>………..</H>	Creates headings of different levels
<HEAD>……..</HEAD>	Contains tags that specify information about a document
<HR>……..</HR>	Creates a horizontal rule
<HTML>……</HTML>	Contains all other HTML tags
<META>……</META>	Provides meta-information about a document
<SCRIPT>……</SCRIPT>	Contains client-side or server-side script
<TABLE>……</TABLE>	Creates a table
<TD>……</TD>	Indicates table data in a table
<TR>……</TR>	Designates a table row
<TH>……</TH>	Creates a heading in a table.

Advantages

- A HTML document is small and hence easy to send over the net. It is small because it does not include formatted information.
- HTML is platform independent and HTML tags are not case-sensitive.

3.2.3 CSS -: CSS stands for Cascading Style Sheets. CSS describes how HTML elements are to be displayed on screen, paper, or in other media.

CSS saves a lot of work. It can control the layout of multiple web pages all at once External stylesheets are stored in CSS files

CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

CSS Example

```
 body {
  background-color: lightblue;
}

h1 {
  color: white;
  text-align: center;
}

p {
 font-family: verdana;
 font-size: 20px;
}
```

3.2.4 Bootstrap- : Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites. It solves many problems which we had once, one of which is the cross-browser compatibility issue. Nowadays, the websites are perfect for all the browsers (IE, Firefox, and Chrome) and for all sizes of screens (Desktop, Tablets, Phablets, and Phones). All thanks to Bootstrap developers -Mark Otto and Jacob Thornton of Twitter, though it was later declared to be an open-source project. Bootstrap has evolved many versions and every time when we want to use this framework we can select the version which we want to use.

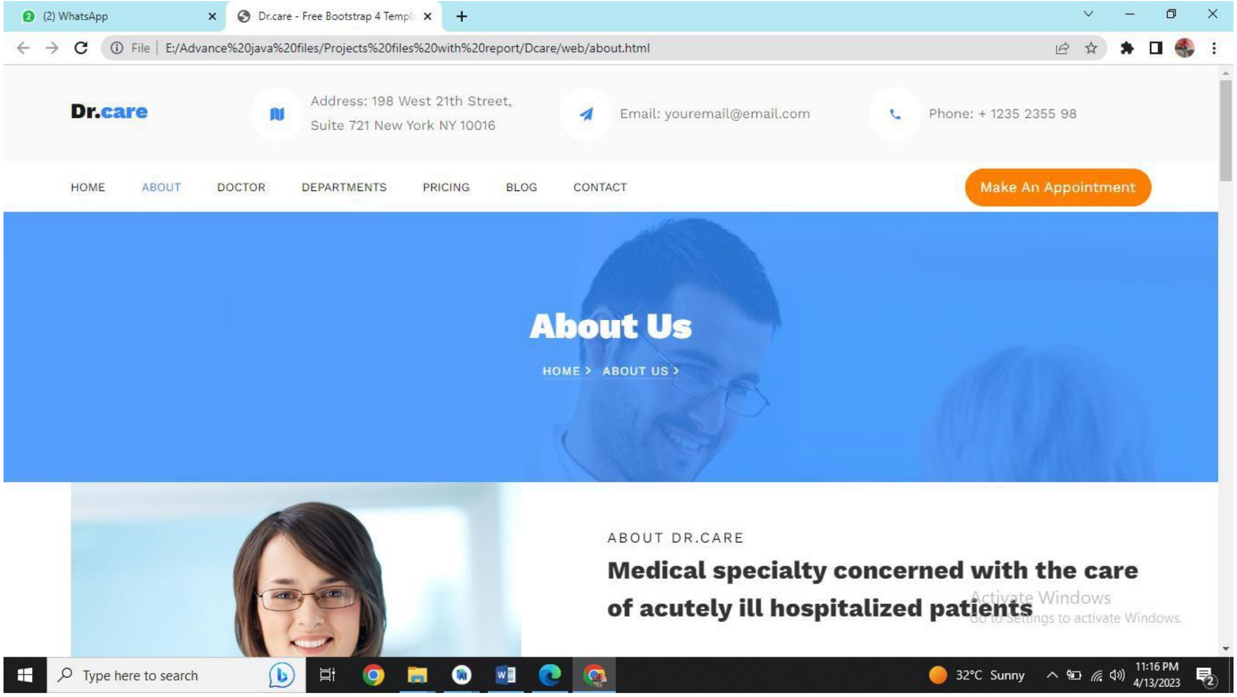
3.2.4 MYSQL: MySQL is a relational database management system

- MySQL is open-source
- MySQL is free
- MySQL is ideal for both small and large applications
- MySQL is very fast, reliable, scalable, and easy to use
- MySQL is cross-platform
- MySQL is compliant with the ANSI SQL standard
- MySQL was first released in 1995
- MySQL is developed, distributed, and supported by Oracle Corporation

CHAPTER 4

RESULT AND DISCUSSION

About page



Address: 198 West 21th Street, Suite 721 New York NY 10016

Email: youremail@email.com

Phone: + 1235 2355 98

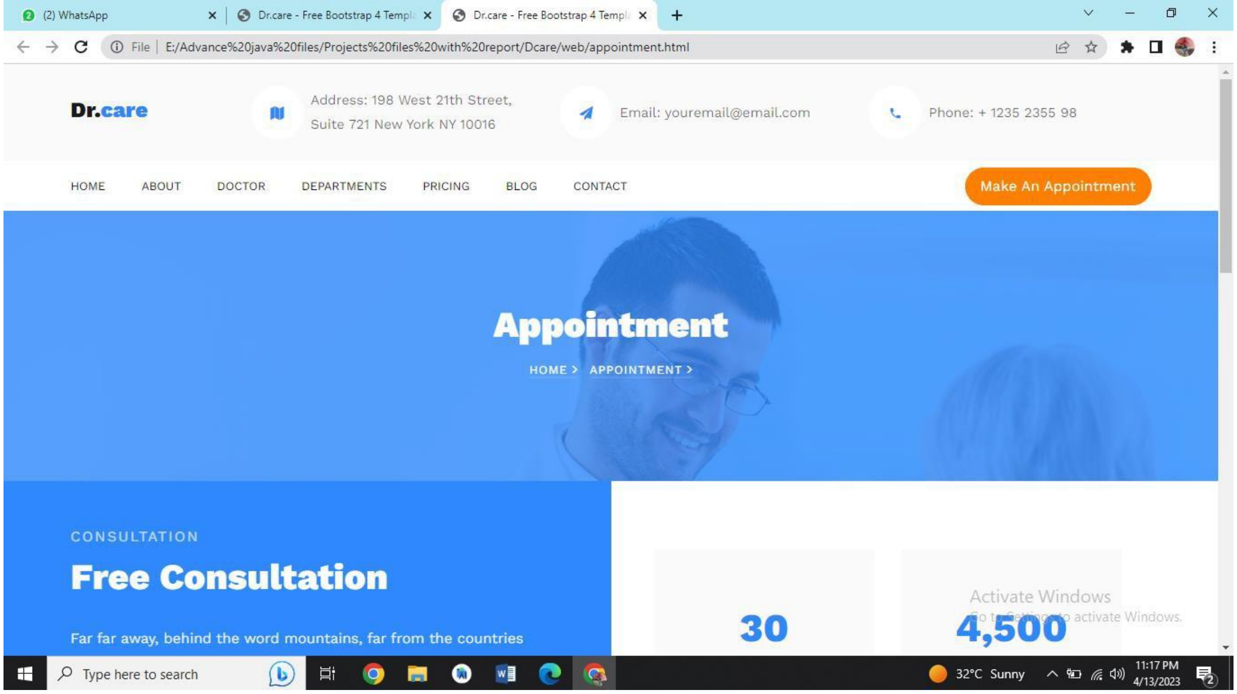
About Us

ABOUT DR.CARE

Medical specialty concerned with the care of acutely ill hospitalized patients

Fig. 4.1 About Us Page

Appointment page



E:/Advance%20java%20files/Projects%20files%20with%20report/Dcare/web/appointment.html

Dr.care

Address: 198 West 21th Street, Suite 721 New York NY 10016

Email: youremail@email.com

Phone: + 1235 2355 98

Appointment

CONSULTATION

Free Consultation

Far far away, behind the word mountains, far from the countries

Fig 4.2 Appointment page

Blog Page

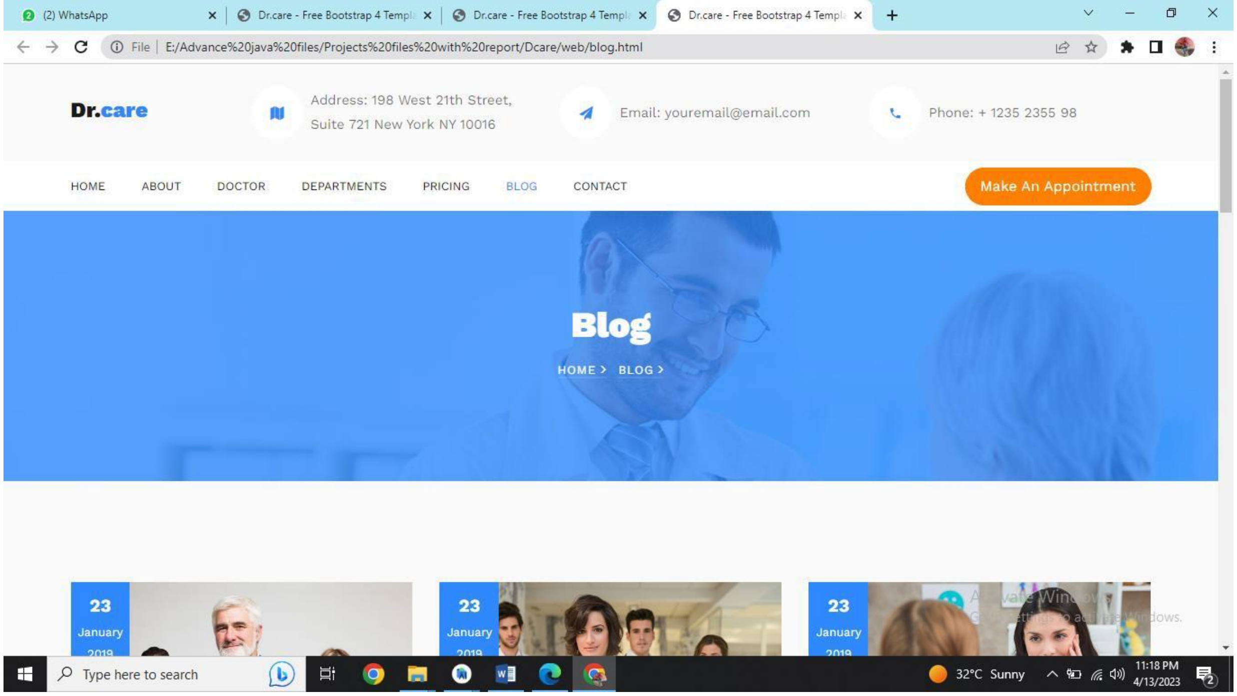
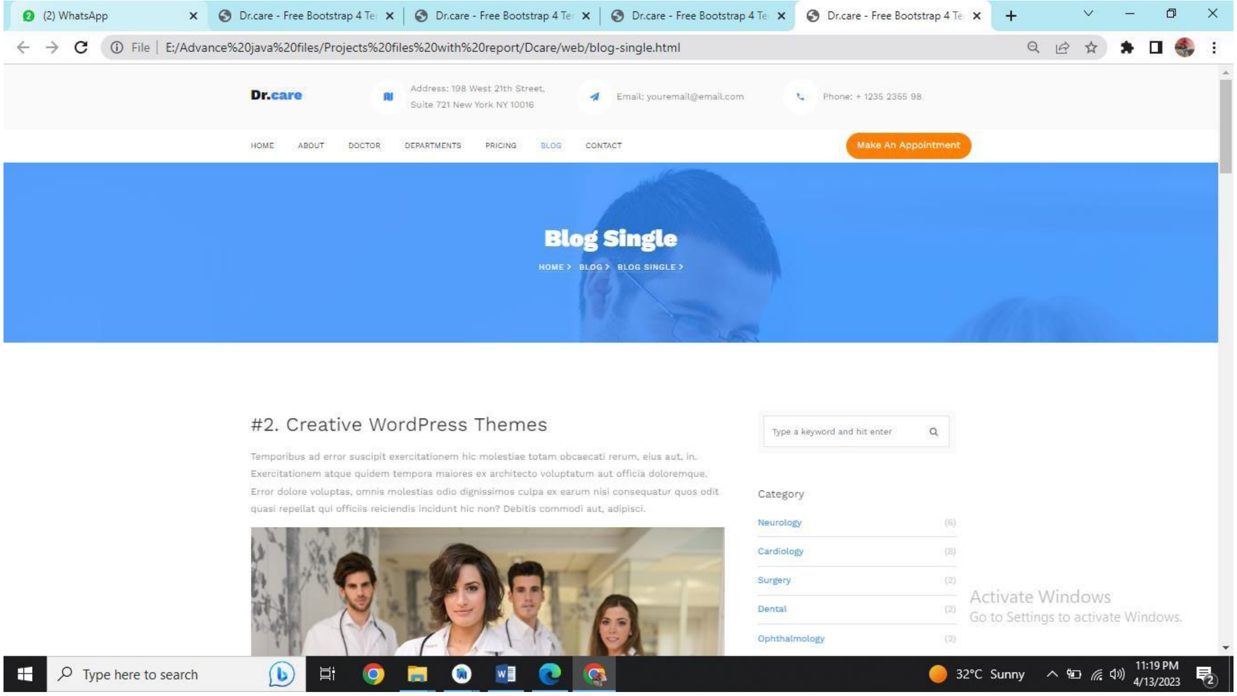

(2) WhatsApp
Dr.care - Free Bootstrap 4 Templ
Dr.care - Free Bootstrap 4 Templ
Dr.care - Free Bootstrap 4 Templ
File
E:/Advance%20java%20files/Projects%20files%20with%20report/Dcare/web/blog.html
Dr.care
Address: 198 West 21th Street, Suite 721 New York NY 10016
Email: youremail@email.com
Phone: + 1235 2355 98
HOME
ABOUT
DOCTOR
DEPARTMENTS
PRICING
BLOG
CONTACT
Make An Appointment
Blog
HOME > BLOG >
23 January 2019
23 January 2019
23 January 2019
Type here to search
32°C Sunny
11:18 PM
4/13/2023

Fig 4.3 Blog Page

Blog- single page

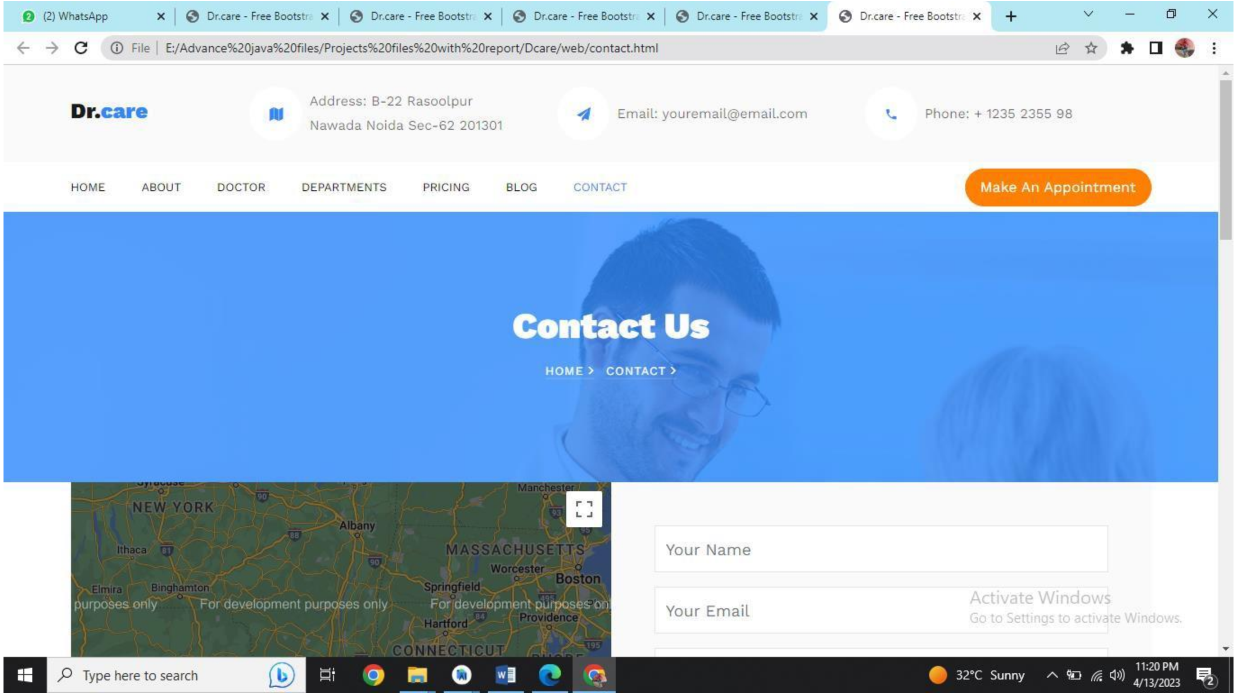

(2) WhatsApp
Dr.care - Free Bootstrap 4 Te
Dr.care - Free Bootstrap 4 Te
Dr.care - Free Bootstrap 4 Te
Dr.care - Free Bootstrap 4 Te
File
E:/Advance%20java%20files/Projects%20files%20with%20report/Dcare/web/blog-single.html
Dr.care
Address: 198 West 21th Street,
Suite 721 New York NY 10016
Email: youremail@email.com
Phone: + 1235 2355 98
HOME
ABOUT
DOCTOR
DEPARTMENTS
PRICING
BLOG
CONTACT
Make An Appointment
Blog Single
HOME > BLOG > BLOG SINGLE >
Type a keyword and hit enter
Category
Neurology (6)
Cardiology (8)
Surgery (2)
Dental (2)
Ophthalmology (2)
Activate Windows
Go to Settings to activate Windows.
Type here to search
32°C Sunny
11:19 PM
4/13/2023

#2. Creative WordPress Themes

Temporibus ad error suscipit exercitationem hic molestiae totam obcaecati rerum, eius aut, in. Exercitationem atque quidem tempora maiores ex architecto voluptatum aut officia doloremque. Error dolore voluptas, omnis molestias odio dignissimos culpa ex earum nisi consequatur quos odit quasi repellat qui officiis reiciendis incidunt hic non? Debitis commodi aut, adipisci.

Fig 4.4 Blog- Single Page

Contact us page



Dr.care

Address: B-22 Rasoolpur
Nawada Noida Sec-62 201301

Email: youremail@email.com

Phone: + 1235 2355 98

HOME ABOUT DOCTOR DEPARTMENTS PRICING BLOG CONTACT

Make An Appointment

Contact Us

HOME > CONTACT >

Your Name

Your Email

Fig 4.5 Contact us page

Department page

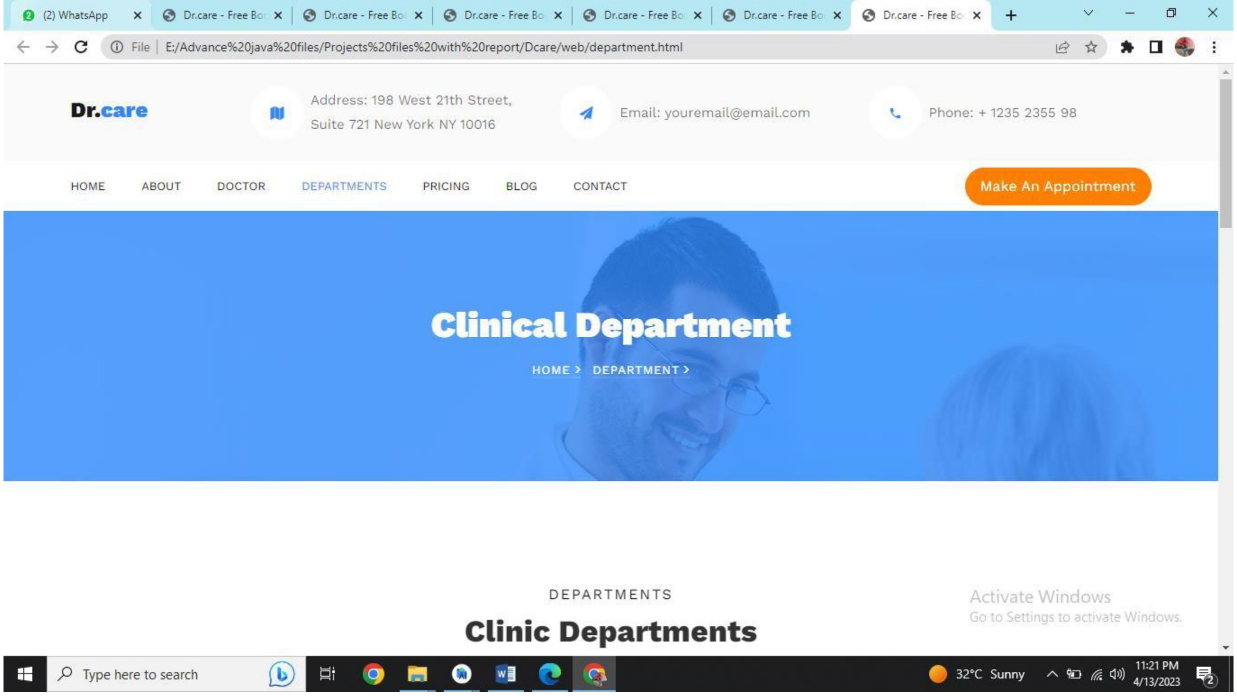


Fig 4.6 Clinical Departments

Doctors

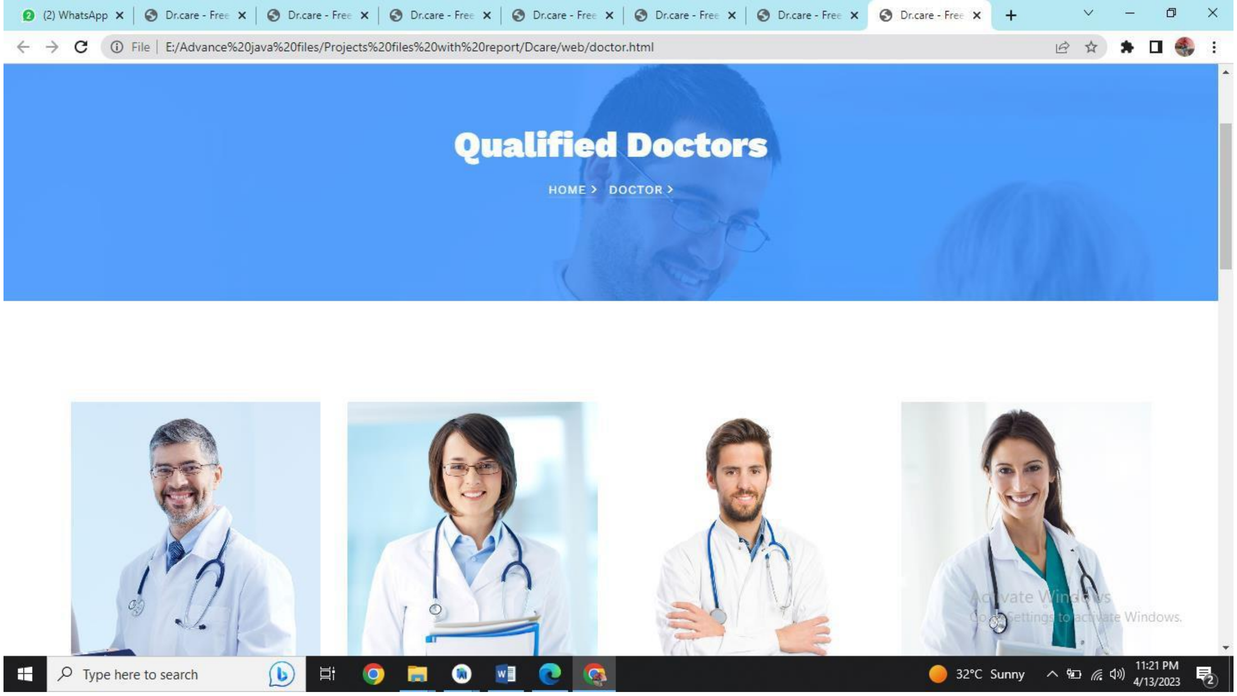
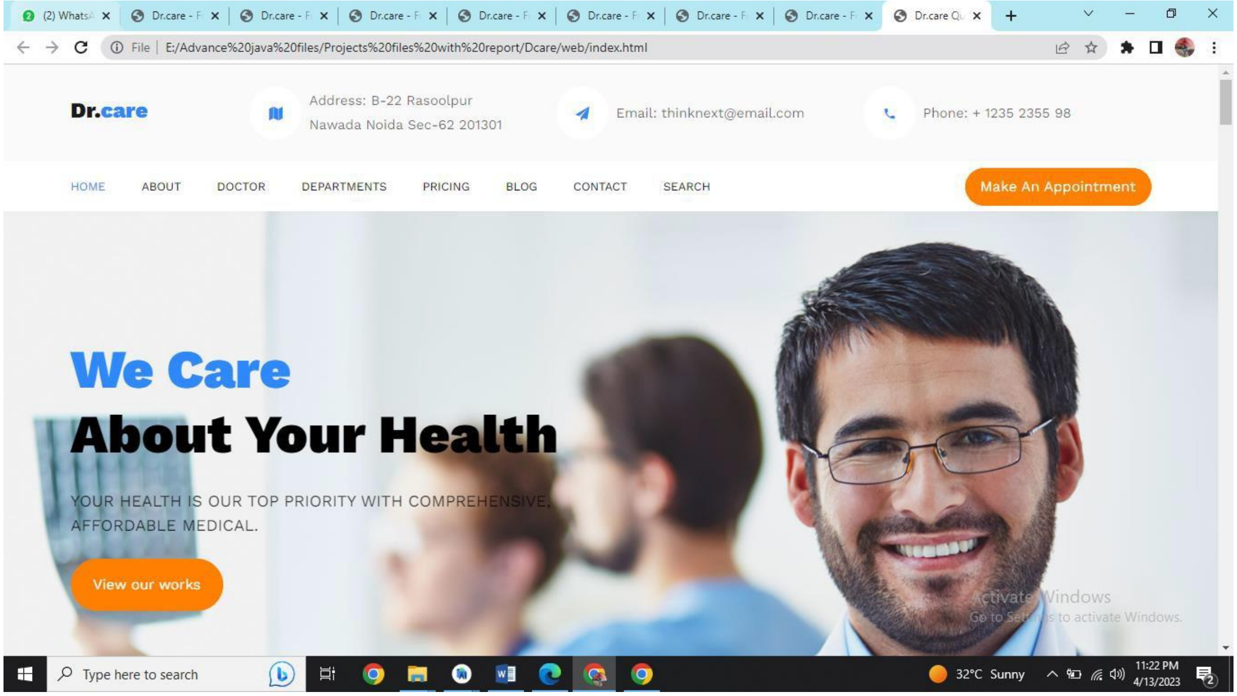


Fig 4.7 Doctors

Index Page



Address: B-22 Rasoolpur
Nawada Noida Sec-62 201301

Email: thinknext@email.com

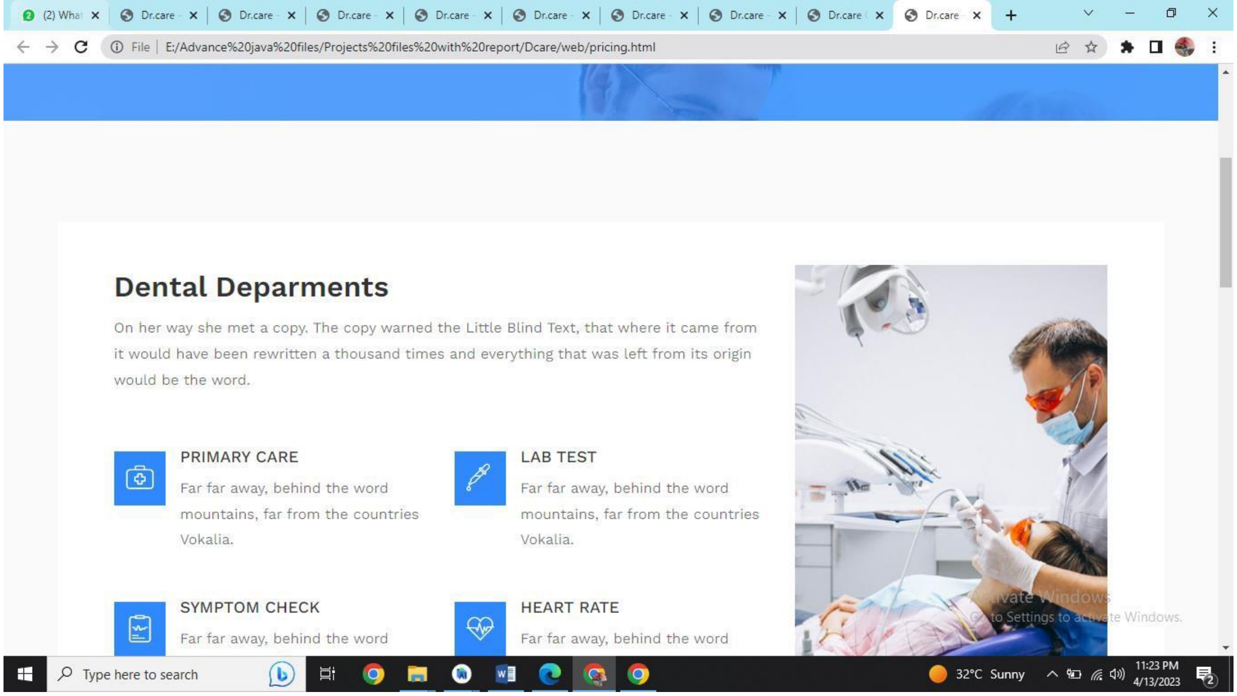
Phone: + 1235 2355 98

We Care
About Your Health

YOUR HEALTH IS OUR TOP PRIORITY WITH COMPREHENSIVE, AFFORDABLE MEDICAL.

Fig 4.8 Index Page

Pricing


(2) What
Dr.care
File
E:/Advance%20java%20files/Projects%20files%20with%20report/Dcare/web/pricing.html
Type here to search
32°C Sunny
11:23 PM
4/13/2023
Activate Windows
Go to Settings to activate Windows.

Dental Deparments

On her way she met a copy. The copy warned the Little Blind Text, that where it came from it would have been rewritten a thousand times and everything that was left from its origin would be the word.

PRIMARY CARE

Far far away, behind the word mountains, far from the countries Vokalia.

LAB TEST

Far far away, behind the word mountains, far from the countries Vokalia.

SYMPTOM CHECK

Far far away, behind the word

HEART RATE

Far far away, behind the word

Fig 4.9 Pricing

CHAPTER 5

CONCLUSION & FUTURE SCOPE

5.1 CONCLUSION

The growing quality demand in the hospital sector makes it necessary to exploit the whole potential of stored data efficiently, not only the clinical data, in order to improve diagnoses and treatments, but also on management, in order to minimize costs and improve the care given to the patients.

In this sense, Data Mining (DM) can contribute with important benefits to the health sector, as a fundamental tool to analyze the data gathered by hospital information systems (HIS) and obtain models and patterns which can improve patient assistance and a better use of resources and pharmaceutical expense.

5.2 FUTURE SCOPE: The future scope of Hospital Management Systems (HMS) is vast and continually evolving, driven by advancements in technology, changing healthcare needs, and increasing regulatory requirements. Here are several key areas where the future of HMS is likely to expand:

1. **Integration with IOT (Internet of Things):**
   - HMS can leverage IOT to monitor patients in real-time, track medical equipment, manage inventories automatically, and even control environmental factors like temperature and humidity in hospital rooms.
2. **Artificial Intelligence (AI) and Machine Learning (ML):**
   - AI and ML can enhance diagnostics, predict patient outcomes, optimize resource allocation, and automate administrative tasks like scheduling and billing. AI-powered chatbots can also provide immediate responses to patient inquiries.
3. **Telemedicine and Remote Patient Monitoring:**
   - With the rise of telemedicine, HMS will increasingly support remote consultations, virtual health monitoring, and electronic health records (EHR) accessible from anywhere, improving access to healthcare and patient convenience.
4. **Block chain Technology:**
   - Block chain can secure sensitive patient data, streamline billing processes, and enhance interoperability between different healthcare providers while maintaining data integrity and privacy.
5. **Personalized Medicine:**

- HMS can facilitate personalized treatment plans based on genetic information, patient history, and real-time health data, enabling tailored healthcare solutions and improving patient outcomes.

6. **Data Analytics and Business Intelligence:**
    - Advanced analytics tools integrated into HMS can provide insights into patient trends, operational efficiencies, and financial performance, enabling data-driven decision-making and proactive healthcare management.
7. **Enhanced Patient Engagement:**
    - Mobile apps and patient portals integrated with HMS can empower patients with access to their health records, appointment scheduling, medication reminders, and educational resources, promoting active participation in their healthcare.
8. **Regulatory Compliance and Security:**
    - As healthcare regulations evolve (e.g., GDPR, HIPAA), future HMS will need robust security measures and compliance features to protect patient data and adhere to legal requirements.
9. **Collaborative Healthcare Ecosystems:**
    - HMS will increasingly facilitate collaboration among healthcare providers, insurers, pharmacies, and other stakeholders, promoting seamless care coordination and interoperability.
10. **Virtual Reality (VR) and Augmented Reality (AR):**
    - These technologies can be used for medical training, surgical simulations, pain management, and therapeutic interventions, enhancing medical education and patient care within the HMS framework.

REFERENCES:-

1. https://www.w3schools.com
2. https://hi.wikipedia.org
3. https://www.geeksforgeeks.org
4. https://en.wikipedia.org
5. https://developer.mozilla.org
6. https://www.javascript.com
7. https://www.geeksforgeeks.org
8. https://www.codecademy.com
9. https://www.javatpoint.com
10. https://getbootstrap.com