Czech University of Life Sciences Prague

Faculty of Economics and Management

Department of Information Engineering

Master's Thesis

Analysis and Design of an information system for Online Doctor appointment using UML.

Al- kawsar Majumder

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

DIPLOMA THESIS ASSIGNMENT

Al Kawsar Majumder

Informatics

Thesis title

Analysis and Design of an information system for Online Doctor Appointment using UML.

Objectives of thesis

The main goal of this thesis is analysis and design of an information system for online doctor appointment, this information system focuses to helps Bangladeshi people where most of the people live in the rural area, this information system helps them to make an appointment without physical presences. The benefit of the system hassle free health support, reduce time and cost. User Registration, login and other functionalities will represent using UML, this part will be made as a web- based prototype these will be the practical part of work.

Methodology

This work consists theoretical and practical. In theoretical part study of relevant information from multiple reliable sources, such as scholarly publications, scientific journals, and online sites. Gained knowledge will execute on following practical part.

In practical part for implement the system will use UML diagrams such as use case for visualizing user interactions and system functionalities. Class diagrams will use the static structure, representing entities like patients, doctors, and appointments etc. Sequence diagrams and activity diagrams will use for dynamic aspects of the system.

The work concludes with the design of the potential user interface for the information system. The prototype will serve as web-Based application using programming language HTML, CSS, JavaScript.

The proposed extent of the thesis

60-80

Keywords

Information system, analysis and design, UML, Class model, interaction model, state model, prototype, online doctor appointment.

Recommended information sources

https://www.dhakatribune.com/bangladesh/332419/report-68%25-bangladeshis-live-in-villages
https://www.researchgate.net/publication/267706023_Agile_Processes_and_Methodologies_A_Conceptual_Stud
https://www.theknowledgeacademy.com/blog/prototype-model-in-software-engineering/

Expected date of thesis defence

2023/24 WS – PEF

The Diploma Thesis Supervisor

Ing. Martin Pelikán, Ph.D.

Supervising department

Department of Information Engineering

Electronic approval: 26. 3. 2024

Ing. Martin Pelikán, Ph.D.

Head of department

Electronic approval: 28. 3. 2024

doc. Ing. Tomáš Šubrt, Ph.D.

Dean

Prague on 28. 03. 2024

Declaration

I declare that I have worked on my master's thesis titled " Analysis and Design of an Information System for Online Doctor Appointment using UML." by myself and I have used only the sources mentioned at the end of the thesis. As the master’s thesis author, I declare that the thesis does not break any copyrights.

In Prague on 31.03.2024 _____

Acknowledgement

I would like to thank my supervisor **Ing. Martin Pelikán, Ph. D** for his support and positive guidance during the period of research and for making this thesis. I am very much grateful to my supervisor for his support and cooperation in how to make a thesis paper properly. Also, I want to thank my whole family they helped me a lot during my course time. They helped me to achieve my dream. I dedicated this thesis paper to my father. I will always be grateful to them.

Analysis and Design of an information system for online Doctor appointment using UML.

Abstract

This Thesis aims to analysis and design of an information system for online doctor appointment using UML. This system will help to manage online doctor appointments, it focuses on keeping track of appointments and the patients involved. to solve hassle free Health support. Bangladesh is a developing country also 68.34% is a rural area (1), there is not sufficient doctors' facilities, and for this reason, lots of people suffer a lot, sometimes they don't have the option to visit the city area it will be more costly and time-consuming, in this case online doctor appointment booking system can fix the problem. It will be a cost-saving, time-consuming hassle-free, and user-friendly system.

The theoretical section of the thesis covers the basic ideas of background information system design and development, requirement analysis, modeling system, UML (Unified modeling language), and user interface.

In the practical section of the thesis is focus analysis of the patients' requirements and creates a plan for the system based on those requirements. This information system design includes an interaction model, a class model, and a state model.

The final output of the thesis is a system design using UML (Unified modeling language) and a user-interactive prototype of an information system.

Keywords: Information system, analysis and design, UML, Class model, interaction model, state model, prototype, online doctor appointment.

Analýza a návrh informačního systému pro online lékařské jmenování pomocí UML.

Abstrakt

Cílem této práce je analyzovat a navrhnout informační systém pro on-line lékařské schůzky pomocí UML. Tento systém pomůže spravovat on-line lékařské schůzky, zaměřuje se na sledování schůzek a zúčastněných pacientů. Využijte bezplatnou zdravotní péči. Bangladéš je rozvojová země také 68.34% je venkovská oblast (1), není dostatek lékařských zařízení, a z tohoto důvodu mnoho lidí trpí hodně, někdy nemají možnost navštívit městskou oblast to bude dražší a časově náročné, v tomto případě on-line lékař schůzku rezervační systém může vyřešit problém. Bude to nákladově úsporný, časově náročný a uživatelsky přívětivý systém.

Teoretická část práce pokrývá základní myšlenky návrhu a vývoje informačních systémů, analýzy požadavků, modelování systému, UML (Unified Modeling Language) a uživatelského rozhraní.

V praktické části práce se zaměřuje na analýzu požadavků pacientů a vytváří plán pro systém založený na těchto požadavcích. Tento design informačního systému zahrnuje interakční model, třídní model a stavový model.

Konečným výsledkem práce je systémový design pomocí UML (Unified Modeling Language) a uživatelsky interaktivní prototyp informačního systému.

Klíčová slova: Informační systém, analýza a design, UML, třídní model, interakční model, stavový model, prototyp, on-line lékařské schůzky.

Table of content

1. Introduction

Nowadays, everything is going digital, which makes life easier. However, difficulties persist for the people living in Bangladesh's rural, particularly in accessing medical care. The proposed system solution to this issue is to develop an online doctor appointment platform that would make it simple for residents of villages to schedule doctor's visits. With this system, patients will be able to access doctor support from anywhere at any time by using their computers, laptop, tab, smartphones etc.

Achieve benefit from this information system patients need to register the system, after registration they will able to login in the system after that they will able to see doctor list with their specialties, patients can see the available time date for doctor appointment, make appointment, online payment, admin will check doctor availability and will approve the request, patients will get email confirmation and patients will able to see the appointment, they will able to cancel or change the appointment,

The admin can add or remove doctor and can view the user and doctor details. They admin can manage the waiting list.

This Thesis will cover analysis and design of the system using UML (Unified modeling language), of online doctor appointment system, software architectural solution based on software design and analysis. It will show a prototype, before implementation this system and database design will indicate how it will be with online doctor appointment information system.

2. Objectives and Methodology

2.1 Objectives

The main goal of this thesis is analysis and design of an information system for online doctor appointment, this information system focuses to helps Bangladeshi people where most of the people live in the rural area, this information system helps them to make an appointment without physical presences. The benefit of the system hassle free health support, reduce time and cost. User Registration, login and other functionalities will represent using UML, this part will be made as a web- based prototype these will be the practical part of work.

2.2 Methodology

This work consists theoretical and practical. In theoretical part study of relevant information from multiple reliable sources, such as scholarly publications, scientific journals, and online sites. Gained knowledge will execute on following practical part. In practical part for implement the system will use UML diagrams such as use case for visualizing user interactions and system functionalities. Class diagrams will use the static structure, representing entities like patients, doctors, and appointments etc. Sequence diagrams and activity diagrams will use for dynamic aspects of the system. The work concludes with the design of the potential user interface for the information system. The prototype will serve as web-Based application using programming language HTML, CSS, JavaScript.

3. Literature Review

3.1 Introduction

Bangladesh is developing country with large population, A large number of people live in village area, (2) where doctors is not available, difficult to get proper health support, in Bangladesh internet and smart phone is available in everywhere, in this case online doctor appointments system can fix this problem, it will help to village people and doctors both ends,

Currently people are taking appointments physically, it's difficult for village people to visit city area for specialties doctors. Many hospitals adopt computerized appointment system, where system is not efficient and some technical gaps between system and human still there are lots of functionality missing like video conferencing, online payment, view doctors' details etc. still patients need to visit hospital or some clinic. There are no virtual facilities introduced yet.

The main goal of the thesis is an efficient online doctor appointment system to reflect patients' satisfaction and get revenue.

The Online appointment system aims to optimize workflow and minimize waiting times while reducing costs and hassles. It is an active appointment system bridge that links timely access to the services and efficient healthcare system. In Bangladesh hospital handles large number of patients without schedules, poor services, in this case people loss their valuable time and face lots of difficulties

Physically appointment books are time-consuming and limited, especially for small clinics where doctors run their own practices, still they are using paper-based scheduling system. The most important thing is the staff members who handle these appointments, sometimes they become confused and frustrated with the heavy number of patients traffic and the manual process. Web-based appointment systems are faster than paper-based systems. it enables numerous users to access the system simultaneously. (3).

3.1.1 Domain Research

Patients' management:

Time, technology, and the fast-growing population have all contributed to the introduction of the online Doctor appointment system. patients' management system is one of the key factors that has been inspired for develop online doctor appointment system. In the modern healthcare industry, patient services are a crucial component. The purpose of this system is to offer an online doctor appointment system with healthcare support and services that measures optimal performance. (4)

Doctors' facilities:

In Bangladesh, there are 18.2 doctors per 10,000 people in urban regions but just 1.1 doctors per 10,000 people in rural areas. The bulk of the nation's population is served by unqualified or semi-qualified healthcare practitioners, who are more concentrated in rural areas. Trained healthcare providers, especially medical doctors, are mostly located in major cities. In Bangladesh, for instance, the population residing in the four largest cities—Dhaka, Chittagong, Rajshahi, and Khulna—is served by 35% of doctors and 30% of nurses, whereas the population residing in rural areas is served by less than 20% of health workers.

In addition, there has been a significant discrepancy between occupied and authorized posts.10 The Bangladesh Health Facility Survey of 2014 states that 62% of the authorized doctor jobs are filled at the district and sub-district levels; in contrast, the percentage is less than 25% in more rural areas, like the union level (5).

Waiting time:

Long time waiting is very alarming issue for patients using village center in developing countries like Bangladesh. In Bangladesh there are no online doctor appointment system they manage it manually, so everyone visits doctors without appointments, and they wait very long time (6), this is why patients face problem very much. For this problem online appointment system saves waiting time and healthy environment.

Three main categories of research, including primary care clinics, specialty clinics, and surgery appointment scheduling, were identified based on patient health care service requirements (7).

Patients appointment Consideration:

The main aim of the online appointment system is to provide an optimal policy and achieve a positive balance between the performance of doctors and patients. Certain factors, such as patient urgency, punctuality, no-shows, cancellations, and service processes, influence how well an appointment goes. These criteria are used as a baseline while creating a well-thought-out web-based appointment system (8).

Inaccurate arrival:

In Bangladesh most of patients come Without or with appointment an appointment in hospital or clinic, lots of people arrive earlier or some are delay which could be make collision in patients waiting room and create unhealthy environment (9), sometimes doctor arrival inaccurate time, this big reason is transportation and huge traffic in Bangladesh. This is why doctors and patients arrive at an inaccurate time and arise unusual issues (10).

3.2 Information system

An information system comprises people, data, procedures, and technology arranged in a comprehensive framework for gathering, processing, storing, and sharing information for efficient decision-making and smooth coordination within the organization. Transaction processing, management, decision support, and executive information systems are just a few of the several types of information systems that can be categorized. With the use of these technologies, an organization's production and efficiency can be increased through better data management, communication, and teamwork. In order to assist organizations adapt to changing surroundings and make wise decisions, information systems are built to support both operational and strategic tasks (11), (12).

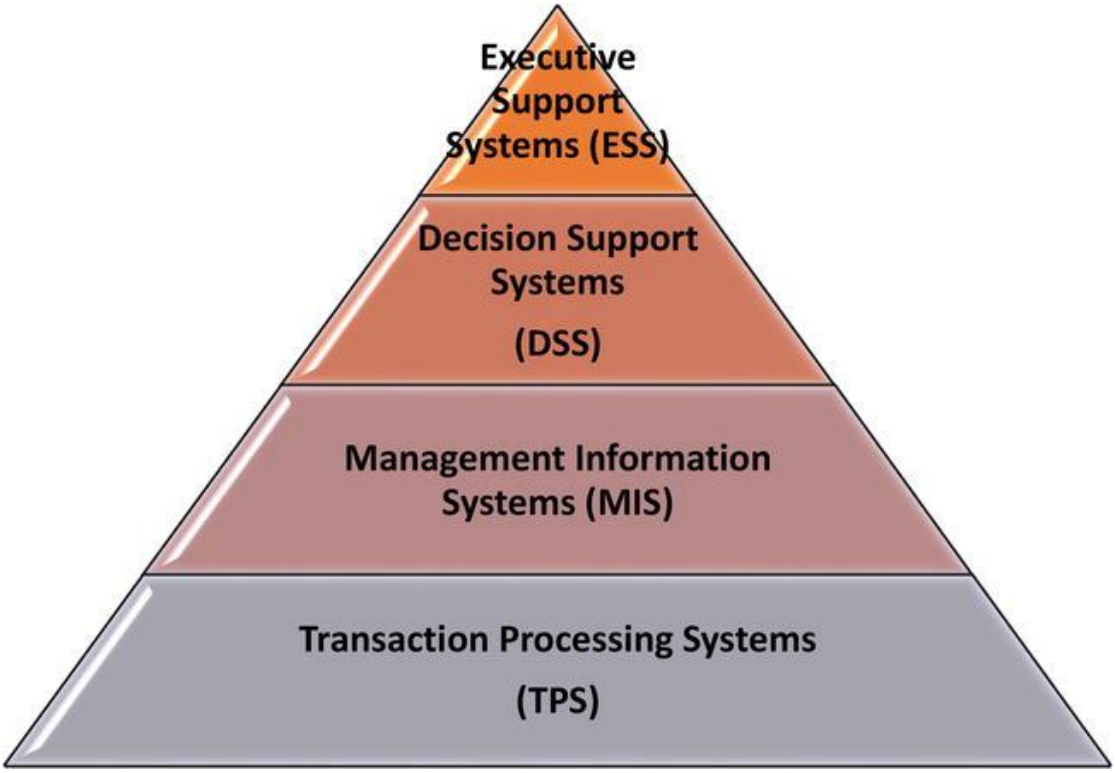

Figure 1: Example of Information system. *(13)*

3.2.1 information system life cycle:

The Information System Life Cycle (ISLC) shows the steps that are taken to create, launch, and keep up with information systems. It is organized in a way that is similar to the Software Development Life Cycle (SDLC), and it includes steps like analyzing needs, designing the system, putting it into action, testing it, deploying it, and keeping it running. The ISLC is very important for making sure that information systems meet the needs of users, work with the goals of the company, and change as needs do. The literature on the Information System Life Cycle stresses how important it is to have a clear and ongoing process, make sure that information systems are in line with business goals, and use feedback loops to make systems work better and be more flexible. In the ISLC, scholars often talk about different methods, taking into account things like the organization's setting, user participation, and new

technologies. Various models exist, such as Waterfall Model, Spiral Model, Agile Model, V-Model, Prototyping Model, incremental Model, (14), (15)

Waterfall model:

The Waterfall Model is a conventional and sequential method for software development that does not include iteration. The process includes many stages such as requirements gathering, system design, implementation, integration, testing, deployment, and maintenance. Each step must be finished before proceeding to the next, and modifications to requirements are typically not allowed once the development process commences. The Waterfall Model is known for its clear and easy-to-follow framework, making it ideal for projects with clearly defined and consistent needs (16), (17).

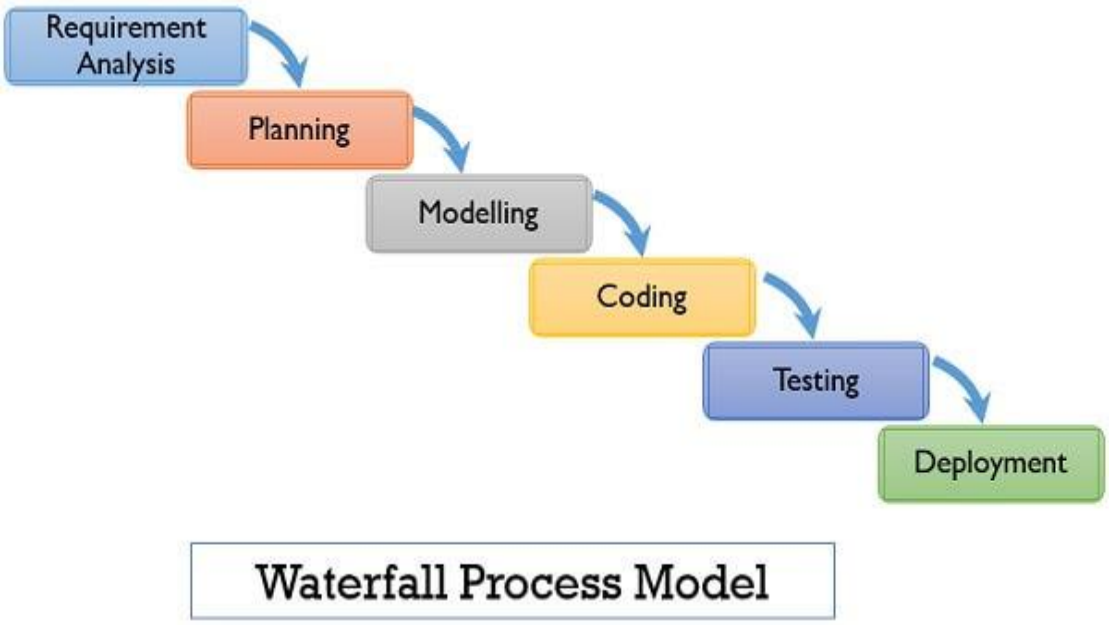

Figure 2:Example of Waterfall model. *(18)*

The Waterfall Model pros:

Clarity: A process that is clear and easy to understand, with clear steps.

Structured Approach: The sequential phases make the growth process organized and structured.
Documentation: There should be a lot of detailed documentation at every step.
Stable Requirements: This method works well for projects with clear and stable requirements.

The Waterfall Model cons:

Inflexibility: Not very flexible, makes it hard to make changes after a phase is over.
Late Feedback: Stakeholders don't get to see the product until it's being tested, so feedback comes in late.
Long Delivery Time: If something is linear, it may take longer to send.
Risk management: Not being able to quickly adjust to new problems or changed needs.
Not Good for Complex Projects: Problems with big, complicated projects or ones whose needs change over time (19).

Spiral model

The Spiral Model is a risk-driven, iterative way to make software that strategically mixes design and prototyping steps. It includes planning, risk analysis, engineering, testing, and review over and over again. The plan is meant to deal with the uncertainty that comes with development by letting work be done in small steps and project risks being reviewed on a regular basis (20).

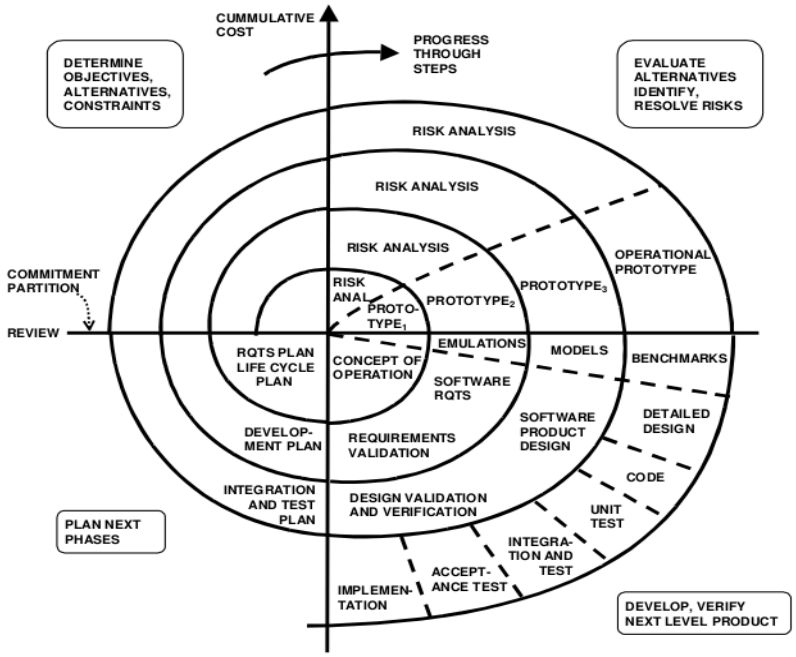

Figure 3: Example of Spiral model. *(21)*

The Spiral Model pros:

Risk Management: The model clearly includes risk assessment and management throughout the development process, which lets problems be found early and fixed.
Flexibility: it can adapt to changes in project scope or needs as they happen.
Customer Involvement: Making samples and iterations often lets customers give feedback and keep working on the project, making sure that the final product meets user needs.
Gradual Development: Software is built up in small, manageable chunks over time. This makes it easier to keep track of work and keep the development process under control.

The Spiral Model cons:

Complexity: The model can be hard to understand and run, especially for smaller tasks with limited resources.

Time-consuming: The Spiral Model may take longer than other straight models, like the Waterfall, because it works in steps.
Cost: The constant tweaking and repetition could make the prices of development go up.

Agile model:

The Agile Model is a way of making software that focuses on being flexible, working together, and getting feedback from customers. To do this, the project is broken up into small, functional pieces that are delivered in short development rounds known as iterations. Agile methods, like Scrum and Kanban, stress flexible planning, ongoing improvement, and small steps of development (22).

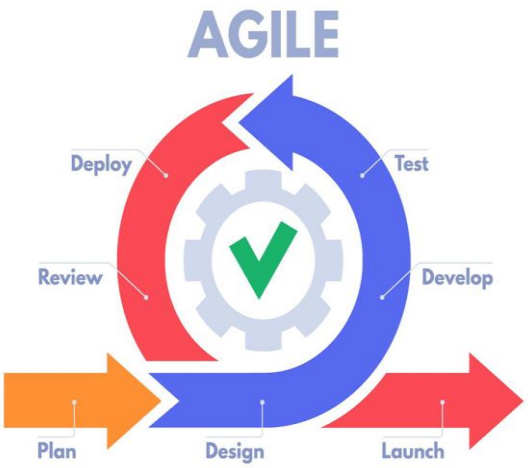

Figure 4: Example of Agile model. *(23)*.

The Agile Model pros:

Flexibility: Agile methodology allows for adaptability to changes in requirements, ensuring that the final product aligns with the developing needs of the customer (24).

Customer Satisfaction: Regular iterations and consumer engagement result in increased customer satisfaction by ensuring the product closely matches user expectations (25).
Iterative Development: Agile methodology facilitates incremental development with frequent releases, leading to faster time-to-market and early achievement of business value.
Continuous Improvement: Agile encourages a culture of learning and adaptation by doing frequent retrospectives that support continuous improvement.

The Agile Model cons:

Lack of Predictability: Agile methodology might make it difficult to accurately forecast the project's schedule and scope.
Resource Intensive: Agile demands active cooperation and regular communication, which can be resource-intensive for development teams and stakeholders.
Documentation Challenges: Prioritizing functional software over extensive documentation might make it difficult to keep project documentation detailed (26).

Prototype Model:

The Prototype Model is an iterative software development approach in which a functional prototype of the system is constructed to enable user interaction and input. This strategy is advantageous when project needs are first unclear. The prototype functions as a tangible depiction of the ultimate product, aiding stakeholders in comprehending and verifying their requirements. The iterative process of creating prototypes and receiving feedback enables the ongoing improvement of requirements and results in the development of a system that better matches user expectations.

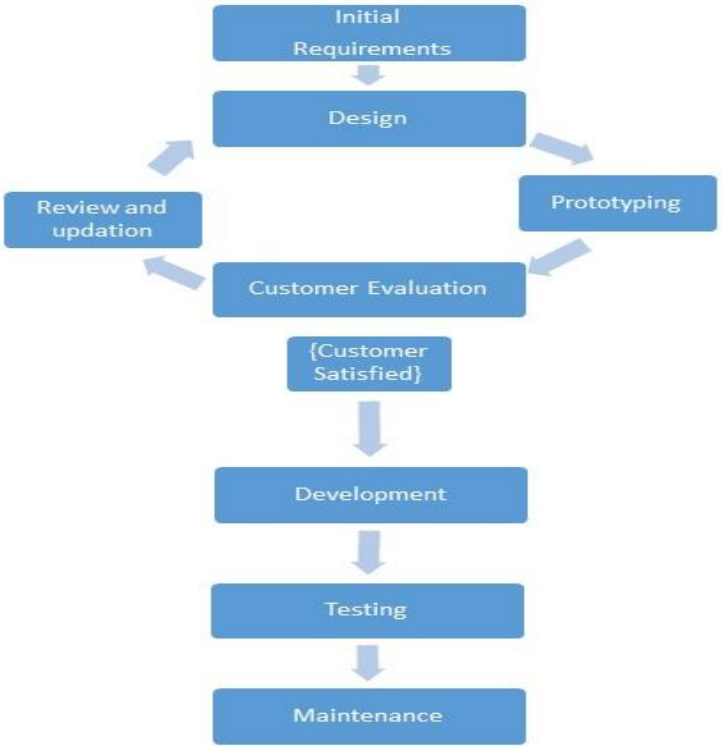

Figure 5: Example of Prototype Model. *(27)*.

Advantages of Prototype Model:

The Prototype Model enhances comprehension of project requirements as developers and clients acquire more accurate insights into the software's functionalities and design, leading to a more polished and efficient solution. (28)

Disadvantages of Prototype Model:

Insufficent Documentation: Because of the focus on making prototypes, there may not be enough detailed paperwork, which could make it harder to maintain and understand the system in the future.

Time-consuming: Making samples and improving them can take a lot of time, especially if they need to be changed more than once, which could make the whole project take longer to finish. (29)

Incremental Model:

The Incremental Model is a software development approach that involves breaking down the overall system into smaller, more manageable components known as increments. It follows an iterative process. Each increment signifies a distinct segment of the ultimate product and is created autonomously. This methodology enables the incremental installation of the system at an early stage of the development process, allowing for testing, feedback, and adjustment. (30).

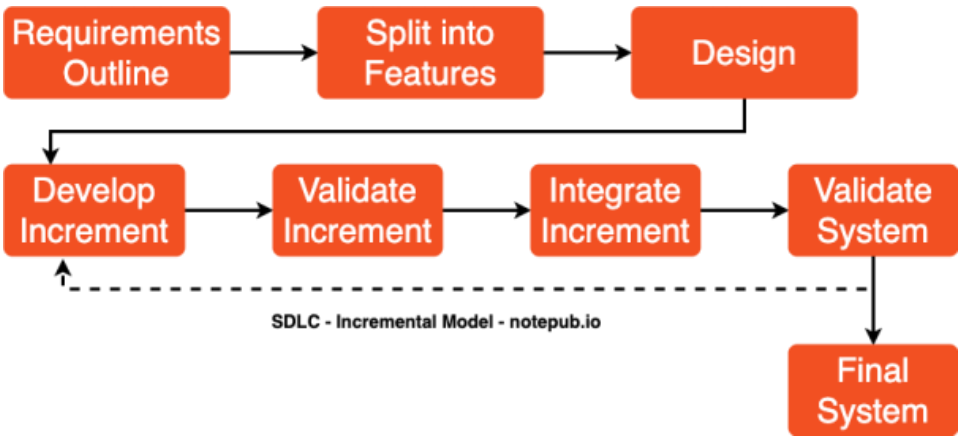

Figure 6: Example of Incremental Model. *(31)*

3.3 Analys of user requirements for information system.

User requirements:

The important step in making information systems is user requirements analysis, which involves finding out what end users want and need, making sure they understand, and documentiaction. To make a system that helps people reach their goals and improves their work processes, it's important to know what those goals are. Usually, this process includes talking directly with people who have a stake in it, like end users, managers, and subject matter experts, to find out what functions, features, and limits they want. The development team learns more about the system's goals, (32) features, and scale by writing down what users want. Some well-known ways to find out what users want are through interviews, polls, and workshops. This phase, which focuses on user-centered design concepts, lays the groundwork for the next phases of system design and development. (33).

Collect user requirements:

The way of collecting user requirements for analysis and information systems is a planned way to understand and doucuments the wants and needs of end users, stakeholders, and the business world as a whole. This step is very important to the success of the system because it sets the stage for future design and growth. To learn more about how users work, what features they need, and what limits they face, technique like interviews, survey, observations, prototyping, brainsorming and workshops are used. Good communication with stakeholders helps find both explicit and implicit needs, making sure that the end system fully meets user requirements. (34).

Interviews:

Interviews are collaborative sessions in which analysts actively communicate with stakeholders, end-users, and subject matter experts to acquire comprehensive insights into their requirements and expectations for the information system. Interviews has a dynamic quality that enables thorough investigation and clarification of requirements. (35).

Survey:

Surveys are tools for collecting structured data that allow for the organized collection of information from a wide range of individuals. Surveys offer a highly scalable method that enables organizations to efficiently gather information from a wide range of consumers. (36).

3.3.1 System requirements:

System requirements are lists of the functions, features, and limits that a system must have in order to meet the needs of its users and other partners. Based on their features and scope, these needs can be put into different groups. This is a list of most common types of system requirements:

Functional Requirements: These are the qualities or functions that the system must have and what it should be able to do. For instance, logging in as a user, handling data,

Non-functional Requirements: These are the qualities that the system must have, rather than the exact actions it must take. Some examples are performance, security, reliability, and usability requirements

3.4 UML (Unified modelling language):

Unified Modelling Language (UML) diagrams are widely used in software engineering as a standard method of representing object-oriented design models.

Because UML diagrams provide visual models, it is easier to determine the needs and scopes of systems and applications. This study attempts to review the literature on the use of UML diagrams in software engineering research in a methodical manner. A thorough examination was carried out between 2000 and 2019—a span of two decades. Out of many papers, 128 were chosen and scrutinized. The primary conclusions demonstrated that class diagrams were the most often used type of UML diagram, and that they were mostly employed for design and modeling purposes (37).

History of UML:

The development of object-oriented languages, like C++, took place in the 1990s. These object-oriented programming languages (OOL) were utilized to create intricate yet captivating systems. Software engineers James Rumbaugh, Ivar Jacobson, and Grady Booch of Rational Software created it in 1994 and 1995. It was not finished until 1996. The three men who created UML, Grady Booch, Ivar Jacobson, and James Rumbaugh, came up with the brilliant notion to create a language that would lessen complexity. Booch's method was flexible to work with throughout the design and creation of objects, Jacobson's method contributed a great way to work on use-cases.It also has an excellent high-level design methodology; Rumbaugh's technique proved helpful when managing delicate systems, David Harel introduced state charts and behavioral models to the UML. OMG (Object Management Group) recognized UML as a standard in 1997. UML has been routinely maintained by the Object Management Group since it was accepted as a standard. UML was made an ISO standard by the International Organization for Standardization in 2005. It is employed in a wide range of sectors for object-oriented model design. The most recent version of UML, 2.5.1, was made available in December 2017 (38) (39).

Characteristics of UML:

- One type of generalized modeling language is Unified Modeling Language.

- It differs from other programming languages used to create software, such Python, C, and C++.
- Strong modeling elements can be created with the help of UML, a visual language.
- It is comparable to analyses and designs that are object-oriented.
- It is widely applicable even beyond the software sector. It is also employed to mirror an organization's workflow (40).

Basic UML Building Blocks:

In UML, there are three different kinds of fundamental structures that are also referred to as building blocks. These are things, relationships, and diagrams.
1.Things: in this structure there are four types of things these are structural things, behavioral, grouping things, annotational things.

- structural things: These stand for both physical and conceptual elements. Seven categories of structural elements exist: Use Case, Active Class, Interface, Collaboration, Class.
- Behavioral: It represents behavior in both space and time. State machines and interactions are the two categories of behavioral entities. Its is dynamic part of UML model.
- Grouping: UML's organizational components. These are boxes that can be used to disassemble models. The Package is the only type of grouping entity.
- Annotational: UML's explanatory components. used to explain, highlight, and comment on any model aspect. One type of annotational item exists.

Relationships

There are 4 types of relations such as dependency, association, generalization, realization.

- **Dependency:** a semantic relationship where modifications to one item (the independent thing) could result in modifications to the other (the dependent thing)
- **Association:** a structural relationship that describes how objects are linked together. Labels indicating the quantity and function of the linkages may also be included. The example below allows for any number of employees (*), each with either 0 or 1 employer.
- **Generalization:** a connection between generalization and specialization. In short, this explains how a parent class (generalization) relates to its subclasses (specializations).
- **Realization:** defines a relationship where a class tells another class what needs to be done. Example: The connection between a class that implements or uses an interface and the interface itself (41).

3.4.1 UML Diagrams:

The Unified Modeling Language (UML) can be used to visualize software and systems through UML diagrams. To comprehend the designs, code architecture, and suggested implementation of complex software systems, software engineers generate UML diagrams. Moreover, business processes and workflows are modeled using UML diagrams (42). Structure diagrams and behavioral diagrams are the two primary categories.

Structure diagrams: The elements of the modeled system are displayed in structure diagrams. Technically speaking, they display various items within a system. Structure diagrams are:

- Class Diagram
- Component Diagram
- Deployment Diagram
- Object Diagram
- Package Diagram
- Profile Diagram

- Composite Structure Diagram

Behavioural diagrams: Behavioural diagrams illustrate the ideal state of a system. They explain how objects work together to form a functional system. Behavioural diagrams are:

- Use Case Diagram
- Activity Diagram
- State Machine Diagram
- Sequence Diagram
- Communication Diagram
- Interaction Overview Diagram
- Timing Diagram.

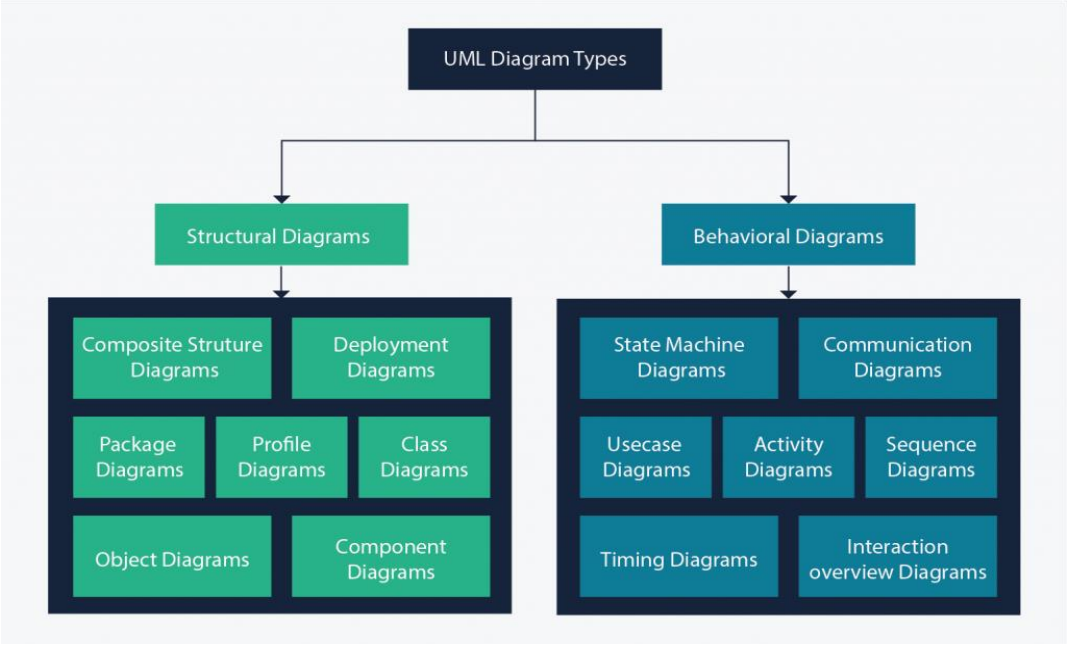

Figure 7: UML Diagram types. *(43)*.

Class diagram:

The fundamental unit of any object-oriented solution is a class diagram. It displays the classes that make up a system, along with their respective operations, properties, and relationships.

Three components make up a class in most modeling systems. Top-level: name; middle: characteristics; bottom-level: operations or methods. Class diagrams are produced by grouping together classes in a complex system with numerous related classes. distinct kinds of arrows indicate distinct relationships between classes (44). UML class diagrams can be drawn using a variety of class diagram notations. The most popular class diagram notations are shown below:

Class:

The main components of a system are represented by classes. It is represented by a rectangle that can have three divisions.

The class name is displayed in the first, and the attributes of the class—that is, the properties of the objects—are displayed in the canter. The class's behavior is represented by the operations listed in the bottom one (45).

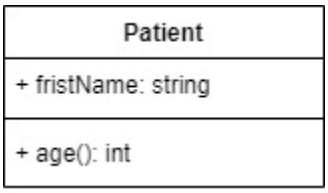

Figure 8: Example of class.

Bidirectional association:

the standard arrangement of two courses. Each class is aware of the other and their mutual relationship. The relationship between these two classes is shown as a straight line (46).

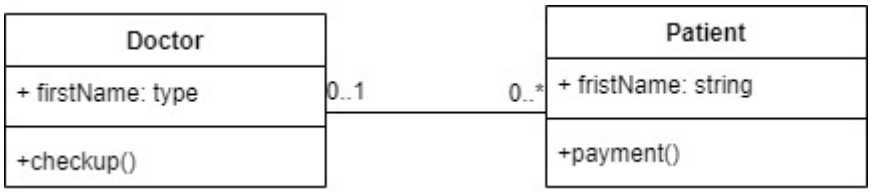

Figure 9: Example Bidirectional association.

Above, the doctor class and patient class are interrelated. At one end of the line, the Doctor takes on the association of "assigneddoctor " with the multiplicity value of 0..1, so when the instance of patient exists, it can either have one instance of Doctor associated with it or no doctor associated with it. In this case, a separate Caravan class with a multiplicity value of 0..* is needed to demonstrate that a patient could have multiple instances of doctor associated with it. Since one doctor instance could have multiple "patients" associations.

Unidirectional association:

A link between two groups that is somewhat less common. One class convers with and is aware of the other. An open arrowhead pointing from the knowing class to the known class is represented by a straight connecting line in unidirectional association modeling.

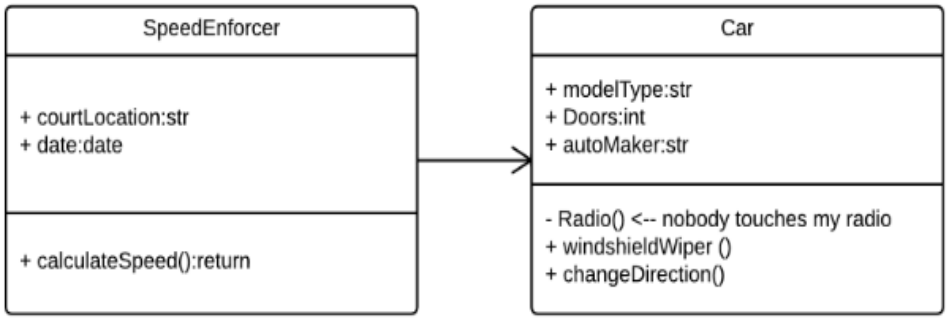

Figure 10: Example of unidirectional association. *(47)*.

Inheritance:

Generalization is the process by which a child or subclass adopts the capabilities of a parent or superclass. It is represented by a closed arrowhead on a straight connected line those points in the direction of the superclass.

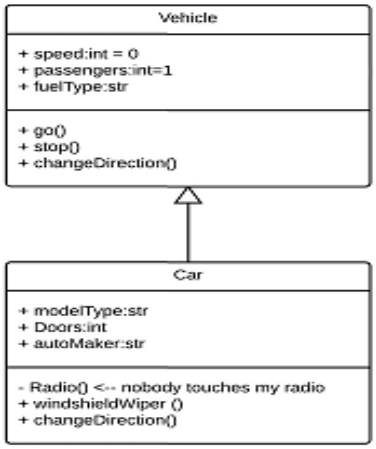

Figure 11: Example of Inheritance. *(48)*

In addition to the specific attributes (model type, number of doors, auto maker) and methods of its own class (Radio(), windshieldWiper(), ac/heat()), the object "Car" would inherit all of the attributes (speed, numbers of passengers, fuel), and methods (go(), stop(), changeDirection()) of the parent class ("Vehicle"). In a class diagram, inheritance is represented by a solid line with a hollow, closed arrow.

State diagram:

A state machine is represented graphically by a state diagram. It displays a behavioral model made up of actions, states, and transitions together with the events that influence them. It is also one of the fourteen Unified Modeling Languages (UML) that are used in software system specification, visualization, construction, and documentation.

There are two categories of UML diagrams: behavioral and structural. While behavioral UML diagrams describe the dynamic changes inside a system, structural

UML diagrams show the static framework of a system or process. One kind of behavioral diagram is a state diagram, in case you hadn't guessed (49).

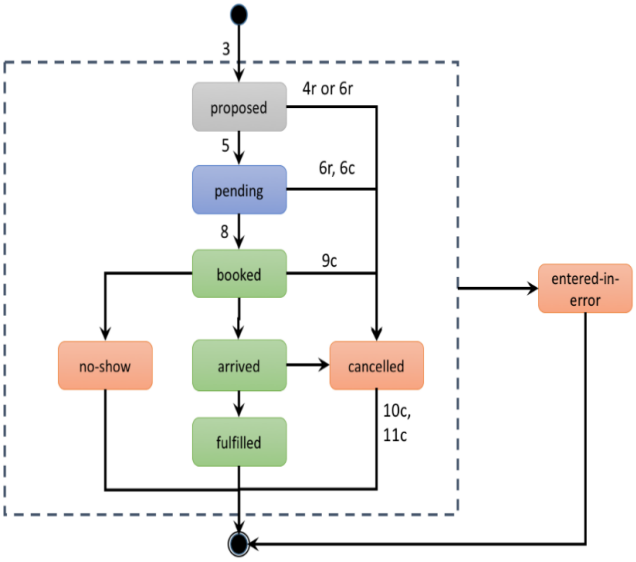

Figure 12: Example of State diagram. *(50)*

State diagram:

A UML dynamic or behavior diagram is called a use case diagram. Use case diagrams employ actors and use cases to model a system's functionality. A use case is a list of tasks, services, and operations that the system must carry out. A "system" in this sense refers to something that is being created or run, like a website. The "actors" are individuals or groups functioning within the system in designated roles (51).

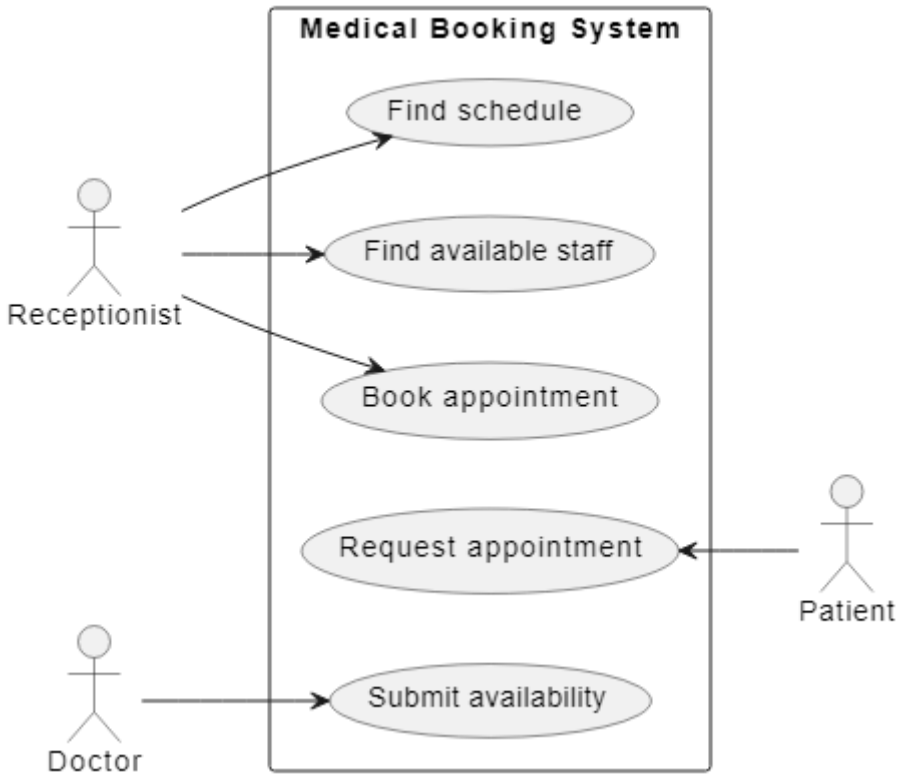

Figure 13: Example of Use case diagram. *(52)*.

Sequence diagram:

UML sequence diagrams are frequently used for both analysis and design purposes. They provide a visual representation of the logic flow within your system, allowing you to validate and document your logic. The most often used UML artifact for dynamic modeling, which focuses on determining the behavior inside your system, is a sequence diagram. The activity, communication, timing, and interaction overview diagrams are examples of other dynamic modeling methodologies. In my view, sequence diagrams are the most crucial design-level models for the creation of contemporary business applications, along with class diagrams and physical data models (53).

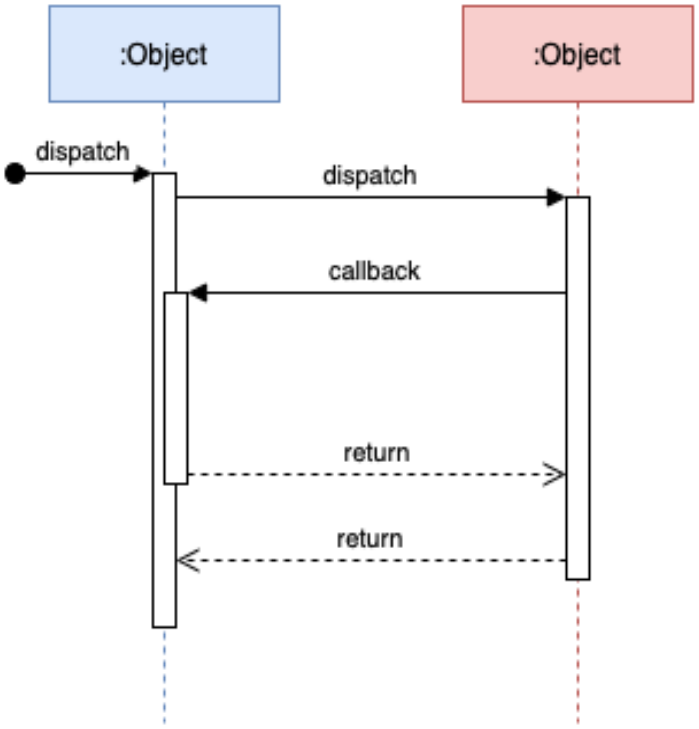

Figure 14: Example of Sequence diagram. *(54)*

Activity diagram:

An activity diagram is a form of flowchart that illustrates how a system or process flows from one activity to another using the Unified Modeling Language (UML). Because it outlines what should occur in the modeled system, it is called a "behavior diagram" and is used to describe the various dynamic features of a system.

Activity diagrams can be used to visualize even the most complicated systems. Because of this, activity diagrams are frequently used in businesses to represent business processes or to explain the phases in a use case diagram. They display each stage in an activity together with the sequence in which they are provided. They can also display how data moves between different operations (55).

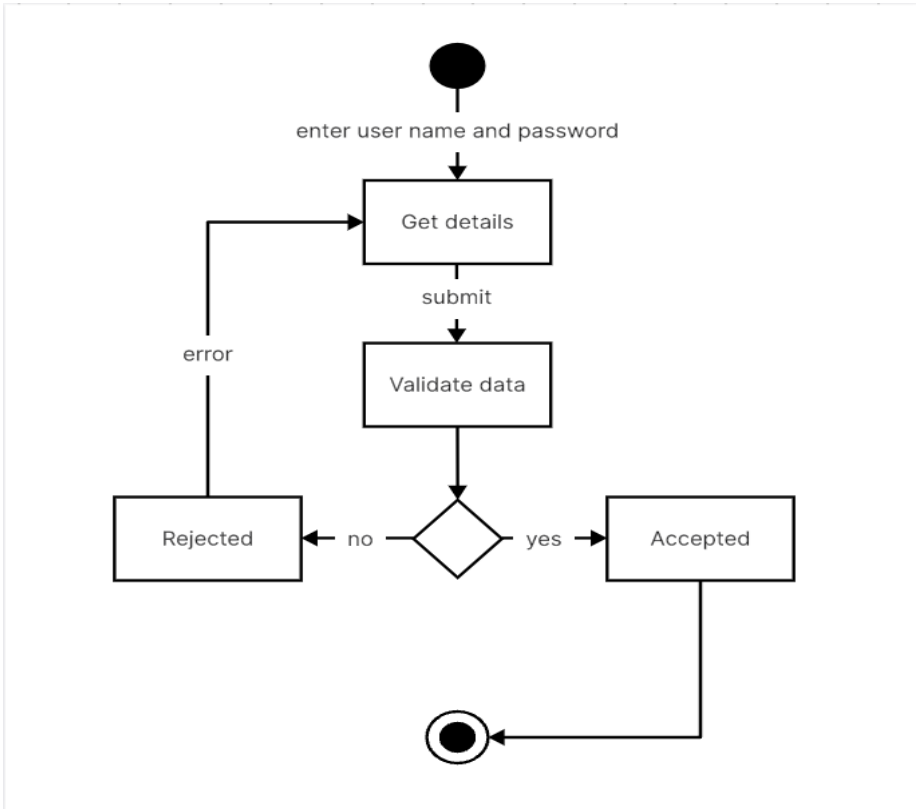

Figure 15: Example of Activity diagram. *(56)*.

Component diagram:

A particular kind of Unified Modeling Language (UML) diagram called a component diagram shows the various components and how they interact to give a visual depiction of a system's high-level structure. In this sense, modular, interchangeable, and executable system components that encapsulate and represent a group of related functionalities are referred to as components. Components are represented as rectangles in a component diagram, while connectors show the connections between them. This image makes it easier to comprehend how the many software components of a system are arranged and interdependent, providing a clearer picture of how various components work together to accomplish the overall operation of the system. System architects and developers can effectively design and convey the structural elements of a software system by using component diagrams (57).

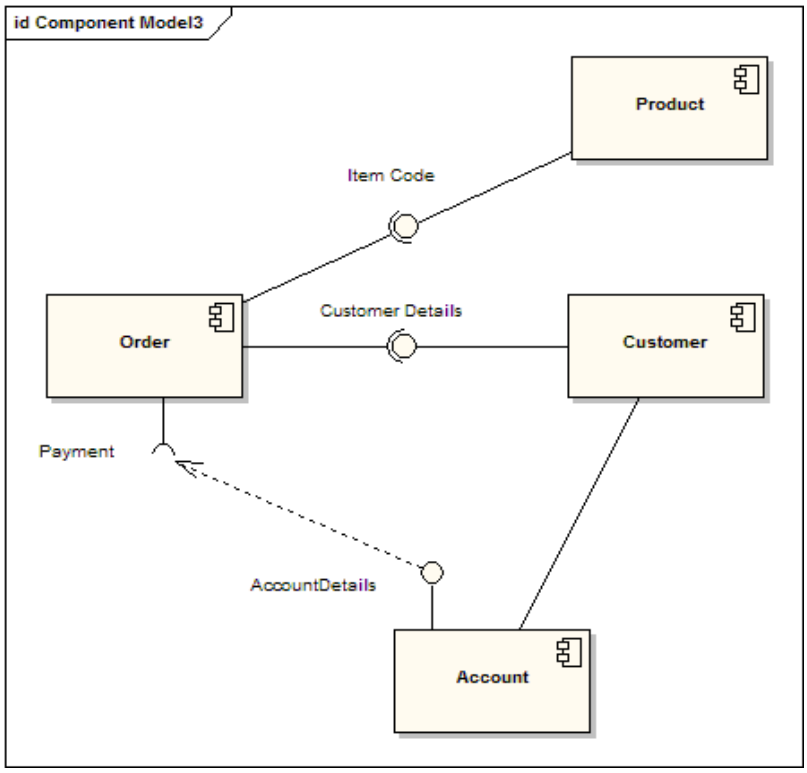

Figure 16: Example of Component Diagram. *(58)*

Deployment diagram:

The physical hardware that the program will be installed on is shown in the deployment diagram. It depicts a system's static deployment viewpoint. It has to do with the nodes and their connections (59).

A deployment diagram describes a feature of the system, it belongs to the structural diagramming family. In this instance, the information created by the software program on hardware components is physically deployed, as shown by the deployment diagram. An artifact is a piece of data produced by the software.

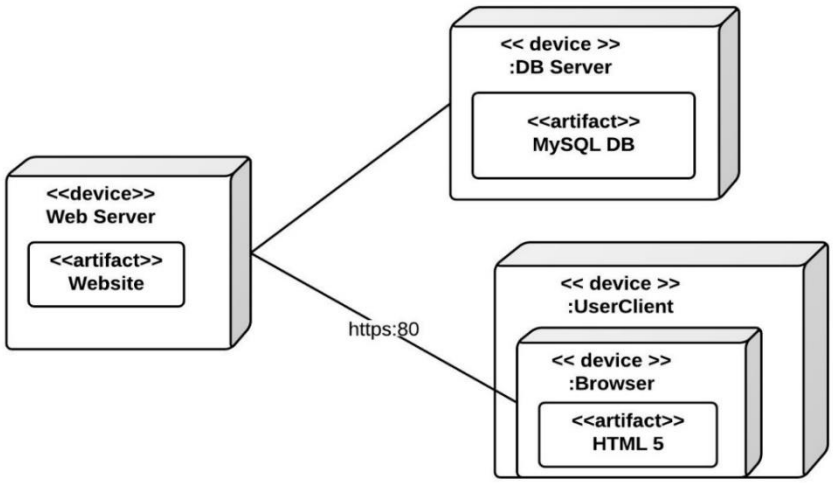

Figure 17: Example of Deployment diagram. *(60)*.

Communication diagram:

In the Unified Modeling Language (UML), a collaboration diagram, sometimes referred to as a communication diagram, is a visual depiction that shows how objects or components interact inside a system. It illustrates the way in which several components work together to accomplish a certain operation or feature. Messages or communication channels are indicated by arrows in a communication diagram, which shows the interactions between objects, which are depicted as rectangles (61). This graphic is useful for simulating a system's dynamic features and aids in stakeholders' comprehension of the information and control flow between various components. Enhancing communication across development teams and visualizing intricate systems are two areas in which it is especially helpful (62).

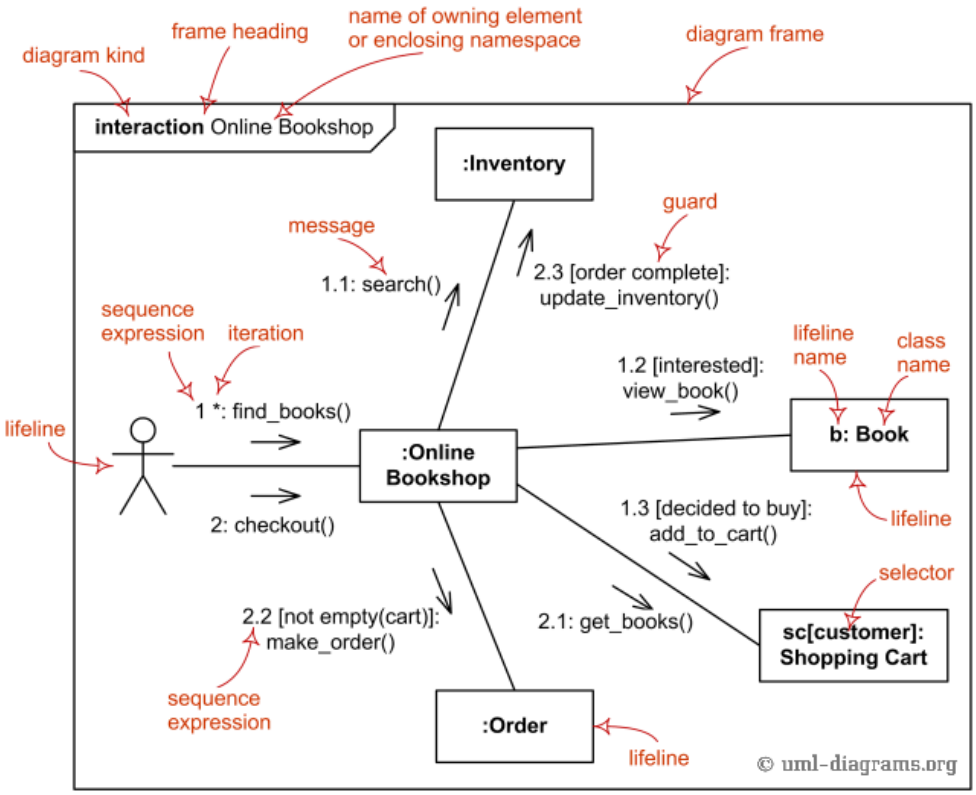

Figure 18: elements of UML communication diagram. *(63)*.

3.5 Programming languages and Database:

HTML:

HTML, also known as Hypertext Markup Language, it uses for the creation and design of web pages. HTML, invented by Tim Berners-Lee in 1991, HTML give structured was to format content on the Internet. HTML use set of markup tags to specify elements like headings, paragraphs, links, images, and other tags. it enables browsers to understand and present the material in the desired manner. HTML is the fundamental framework for web development, providing the necessary structure for web pages and enabling the incorporation of multimedia elements and hyperlinks. With the advancement of technology (64).

CSS:

Cascading Style Sheets (CSS) is an essential technology in web development that improve HTML by offering a method to customize and structure web pages. CSS allow to manipulate the appearance of HTML elements by specifying attributes such as layout, colors, fonts, and spacing. CSS introduced by the W3C in 1996 (65), functions by selecting HTML components and applying style rules to them. CSS plays a vital role in designing visually attractive, adaptable, and user-friendly websites that can be access on different devices. (66).

JavaScript:

JavaScript is a flexible programming language mainly utilized for creating websites, allowing for the implementation of dynamic and interactive elements. JavaScript, which was first introduced by Netscape in 1995, has now become a crucial element of front-end web development (67). Developers can use it to modify the Document Object Model (DOM), which allows for live updates and interactions on web pages. Node.js, a server-side runtime environment, make use of JavaScript to construct server applications that are both scalable and efficient. The reason for its appeal lies in its capacity to improve user experience by including features such as form validation, asynchronous queries, and dynamic content updates. JavaScript is continuously evolving through ongoing standardization initiatives, such as ECMAScript. This evolution brings current language features and capabilities to developers. (68), (69).

PHP:

PHP, short for Hypertext Preprocessor, it is a server-side programming language that is very popular in web development. PHP, initially created by Rasmus Lerdorf in 1994 and subsequently evolved as an open-source initiative, it enables developers to include dynamic content into HTML pages. It excels in managing tasks such as form processing, database transactions, and server-side scripting (70). PHP is processed on the server side, producing HTML output which is

subsequently transmitted to the client's web browser. PHP is highly recognized for its smooth integration with databases such as MySQL and is essential for constructing dynamic and interactive websites (71). PHP has a substantial and engaged community, which has led to ongoing development and the introduction of improved performance and contemporary language features in its latest editions. (72).

3.5.1 Database:

A database is an organized set of data that makes it easy to view, manage, and change. Databases are very important for many types of applications, from simple web apps to complicated business systems, because they store and retrieve information. Usually, tables with rows and sections are used to store data in a structured way (73). MySQL, PostgreSQL, Microsoft SQL Server, and Oracle Database are all well-known database management systems (DBMS). These systems make sure that data management is reliable and scalable by offering features like data integrity, security, and fast searching. To make strong and effective data storage solutions, you need to understand database ideas like normalization and indexing. (74).

MySQL:

MySQL is a widely used open-source RDBMS that has a considerable impact on web development and database management. MySQL, which was initially created by Michael Widenius and David Axmark in 1994, is currently under the ownership of Oracle Corporation. MySQL is renowned for its dependability, capacity to handle large amounts of data, and user-friendly interface. It is extensively utilized in a range of applications such as content management systems, e-commerce platforms, and data-centric websites. The system provides support for SQL (Structured Query Language) to define and manipulate data. Its architecture is designed to efficiently handle huge datasets (75). MySQL is distributed under the GNU General Public License, which contributes to its widespread use in both small-scale projects and enterprise-level applications. (76).

PostgreSQL:

PostgreSQL, which is often just called "Postgres," is a strong open-source relational database management system (RDBMS) that is known for being able to be extended, following standards, and having strong features (77). PostgreSQL was created at the University of California, Berkeley, in the 1980s. It has since become a community-driven project. It works with many advanced data types, indexing, and complex queries, so it can be used for a wide range of tasks, from small projects to big enterprise systems (78). PostgreSQL focuses on following standards, having ACID (Atomicity, Consistency, Isolation, and Durability) traits, and being able to grow by supporting procedural languages such as PL/pgSQL and PL/Python. (79).

3.6 Unser interface (UI):

The visual and interactive components of an application, such as buttons, text, colors, and animations, are created through UI design. It's different from UX design, which focuses on more general aspects of product development, yet it's crucial for a satisfying user experience. Strategic layout decisions, such as where to place the logo and how buttons behave, are part of UI design, which guarantees an appealing user interface (80).

User interfaces (UI) collaborate closely with designers of user experiences (UX) and other design disciplines. Their responsibility is to ensure that each page and action a user takes when interacting with the final product adheres to the overarching concept that UX designers have designed (81).

3.7 User experience (UX):

UX design is concerned with how actual people interact with commonplace goods and services like websites, applications, and even coffee makers (82).

The term "user experience" (UX) describes how a user interacts with a product or service. The process of developing goods or services that offer users meaningful experiences is known as user experience design (UX design). It encompasses a wide range of product development disciplines, such as branding, usability, function, and design (83).

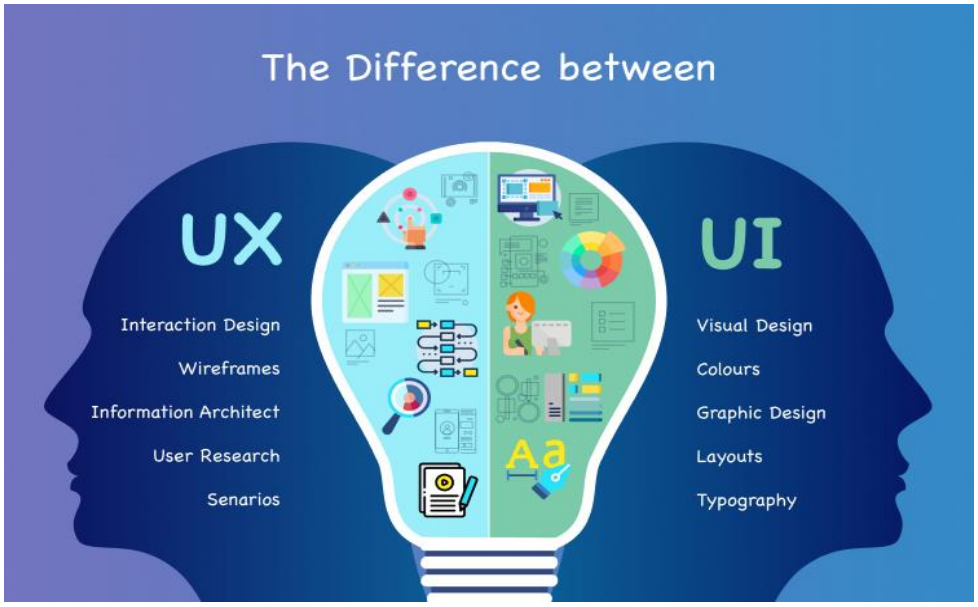

Figure 19: Example of UI and UX. *(84)*.

3.7 User experience (UX):

Usability is the degree to which a system, product, or service may be effectively, efficiently, and satisfactorily utilized by particular users to accomplish particular goals in a particular context of usage.

Any product that has a human-technology or human-machine interface needs to be user-friendly. Whether it be apps or websites (85). Usability is determined by five quality factors:

- Learnability: When consumers first come across the design, how simple is it for them to complete basic tasks?

-
- Efficiency: How fast can users complete tasks after they understand the design?
- Memorability: How easy is it for users to regain proficiency with the design when they come back to it after a period of inactivity?
- Error: How frequently do users make mistakes, how serious are these mistakes, and how quickly can they fix the mistakes?
- Satisfaction: To what extent is the design user-friendly? (86).

Figure 20: Example of Usability. *(87)*

3.8 Wireframe

A wireframe is a simple visual depiction of the design of a webpage or application. Consider it the framework for your design, indicating where buttons, pictures, and text should be placed.

It focuses more on structure and usefulness than on fonts, colors, or actual visuals. It establishes the framework for a seamless user experience (88).

Logical design (Low-Fidelity Wireframe):

A visual depiction of the user interface, a logical design is also known as a wireframe (wire model) or a Lo-Fi (Low Fidelity) model. The most basic kind of

wireframe is a low-fidelity wireframe. This kind of wireframe is made at the initial step of the wireframing process and is frequently created by hand rather than using a computer program. A low-fidelity wireframe is simple and devoid of design components, akin to the framework of a webpage. This kind of wireframe aims to specify the page hierarchy and structure as well as the user flow (89).

Graphical design (High- Fidelity Wireframe):

High-fidelity wireframes are elaborate, minute drawings that provide a far better representation of the final product. High-fidelity prototypes follow high-fidelity wireframes. At this point, the model begins to function and incorporates all the colors, visuals, and interactive features.

In-depth features are covered in high-fidelity wireframes. They provide a thorough preview of the look and feel of a page or application. (90)

These wireframes incorporate information, iconography, and a sense of the intended visual style, rather than just basic forms and placeholders.

3.9 Prototype

A prototype is a preliminary version of software or a product that is created to test an idea or procedure before a full-fledged final version is created.

Using prototypes, you may test the ideas by presenting a preliminary version of the solutions to actual users. It makes it possible for developers and designers to comprehend the needs of the client and gather all input during user testing (91).

4. Practical Part

The information system for the online doctor appointment system will be proposed and the user needs will be analysis as part of the practical part of this thesis.
The requirements are analysis in the first section, which is a collection of requirements gathered from user potential interpolation. requirements are analysis, and the requirements are then modified by the thesis author considering the information gathered from these sources. The final requirement background for creation class model, state model, and interaction model are developed. The indicated UML models' design is covered in the second section, both the interaction and state models. The design of the previously described UML models is covered in the second section. The proposal for the user interface and the development of its prototype for Make a system representation. The technology is specifically designed for patients in mind. This information system will significant solution for the healthcare business in Bangladesh.
For this system, suggestions are made for future development and implementation for the proposed system.

4.1 Basic overview of the information system:

The proposed information system should effectively support all the facilities for the information system. The most important functionality is the registration login of system and View doctors list and make appointments. The information system is primarily intended for
Web-baased applications. Its aim is to reduce the hassle free, make online appointments for doctors without physical presence, make online payment and get prescription. Users will register the system for use and communication to patients and doctors. The system needs to ensure the data integrity and authorization for safety and store user data. There is also needed to make sure its protected by user authentication and modern encryption. The system will provide several additional functions, which are required by the user of the system.

The main benefit of using the system is online communication between patients and doctors, which reduces the cost, time, and transportation hassle.

4.1.1 System requirements analysis:

The system requirements are examined in this chapter. Interviews with potential users served as the method used to gather the system requirements. Since requirements analysis is the foundation for system design as a whole, it must be completed as thoroughly as possible. The prerequisites are given below in order of importance, based on the results of one-on-one interviews. It is therefore extremely roughly formatted. Two distinct tables, one for functional needs and the other for non-functional requirements are the result of the analysis of these requirements. Following the user interviews, the following preliminary needs were developed, which the system ought to support:

- Web-based application, desktop/laptop
- Main page – ability to search doctor list.
- Ability to make appointments.
- Appointment approved. (Admin only).
- Profile for each patient.
- Login phone, password.
- Member registration of the online doctor appointment system.
- Each member's details will be saved in the database.
- Records of user payments.
- Reminder notification.
- Making payments through local system bKash.
- Records of events – appointments, prescription, etc.
- Records of patients' attendance of online doctors.
- Admin will manage all users.
- Manage patients' waiting list (Admin only).

4.1.2 Functional and non- functional requirements list:

Functional requirements:

- The system will allow to registration of a new user.
- The system will allow logging into the system.
- The system will allow to search doctors list.
- The system will allow patients to manage an appointment.
- The system will allow patient to manage payment.
- The system will enable to send notification to patient.
- The system will allow doctor to view appointment list.
- The system will allow doctor to manage patient record.
- The system will allow doctor to give prescription.
- The system will allow admin to manage user.
- The system will allow to admin manage waiting list.
- The system will allow admin to manage Appointment.
- The system will allow admin to manage payment.

FN- 01: The system will allow to registration of new user.

The system enables to users easily sign up and create a new account. It's a simple and user-friendly process for registering as a new user.

FN- 02: The system will allow logging into the system.

The system will allow users an easy-to-use login function that lets users safely enter their username and password to access their accounts.

FN- 03: The system will allow to search doctors list.

The system will allow to see list of doctors, making it easy for patients to choose from the available and specialist doctor.

FN- 04: The system will allow patients to manage an appointment.

The system will enable patients to make appointments, providing a convenient way for them to set up appointment with doctor based on their availability and preferences.

FN- 05: The system will allow patient to manage payment.

The system will allow a payment management feature, users will be able to do transactions securely. This function ensures a convenient and reliable way for users to make payments within the system.

FN- 06: The system will enable to send notification to patient.

The system will allow to send of notifications to patients, ensure appointment time and date. This feature keeps patients informed about their appointments, which improves the user experience.

FN- 07: The system will allow doctor to view appointment list.

The system will allow doctoe to see appointment list, how many done, pending and in process.

FN-08: The system will allow doctor to manage patient record.

System will allow Doctor to see previous record of paitients and able to update patient records.

FN- 09: The system will allow doctor to give prescription.

System will allow doctor to provide medication and prescription.

FN- 10: The system will allow admin to manage user:

The system will allow admin to efficiently manage user accounts, granting them the ability to add, and modify user profiles as needed, ensuring streamlined administrative control and supervision

FN- 11: The system will allow to admin manage waiting list.

The system will allow admin to efficiently handle patients waiting list, like add new patients in waiting lists, delete, update, this feature ensure efficient and fair patients appointments handling.

FN- 12: The system will allow admin to manage Appointment:

System will allow admin to manage appointments.

FN- 13: The system will allow admin to manage payment:

System will allow admin to manage payments.

Non- Functional requirements.

- The system will be available for a mobile application.
- The system will authenticate the system user.
- The system will be Responsive and user-friendly UI.
- The system will be Reliability, Less Weight, Speed.
- In the future, the system will be available via a web browser.
- The system will establish a secure connection using the HTTPS protocol.

NFN-01: The system will be available for a mobile application.

The system will allow users to access through a mobile application. This mobile-friendly approach ensures flexibility and is easy to use for individual user prefer to interact with the system on smartphones or tablets.

NFN-02: The system will authenticate the system user.

The system will ensure user authentication, ensure a secure and verified access process for individuals. This authentication mechanism helps confirm the identity of users, improving overall system security.

NFN-03: The system will be Responsive and user-friendly UI.

The system will allow responsive user- friendly user interface (UI), which can be easily adjusted to fit different screen sizes and devices. This will ensure accessibility and easy to use the system and user experience across all devices such as computer and mobile.

NFN-04: The system will be Reliabile, Less Weight, Speed.

The system will focus on reliability, ensuring consistent and dependable performance. The system will be a lightweight design, it will not put more pressure on users' devices. It will ensure quick and efficient performance for a smoother user experience.

NFN-05: In the future, the system will be available via a web browser.

In the future, the system will allow users to through a web browser. Users will be able to use it directly online communication without any additional software, it will be more accessible and user-friendly.

NFN-06: The system will establish a secure connection using the HTTPS protocol.

The system will establish a secure connection between user and system by using HTTPS protocol. It will help to information exchanged between user devices and the system, it will be encrypted.

4.2 System Design

System design for this work using UML model, such as class model, state model and interaction model for visual representation of the system, All UML model will be implement using Draw.io. tool. After create UML model, prototype will make at end of the practical part and will show the core functionalities of the system, the prototype will create for mobile application.

4.2.1 Class diagram:

The class diagram for the online doctor appointment system has eight essential classes, each representing a unique entity within the system. The "User" class functions as a general superclass for both "Admin" and "Patients," encapsulating shared characteristics and procedures. The classes "Doctors" and "Patients" are subclasses of the class "User," designed to full fill their respective tasks.
The "Appointment" class manage the organization of scheduling information, it connected between patients, doctors, and admin, On the other hand, the "Prescription" class regulates the doctors to prescribed medications and treatments. The "Payment" class encapsulates transaction details, guaranteeing a secure and efficient financial process. Finally, the "Rating" class collects comments from patients.

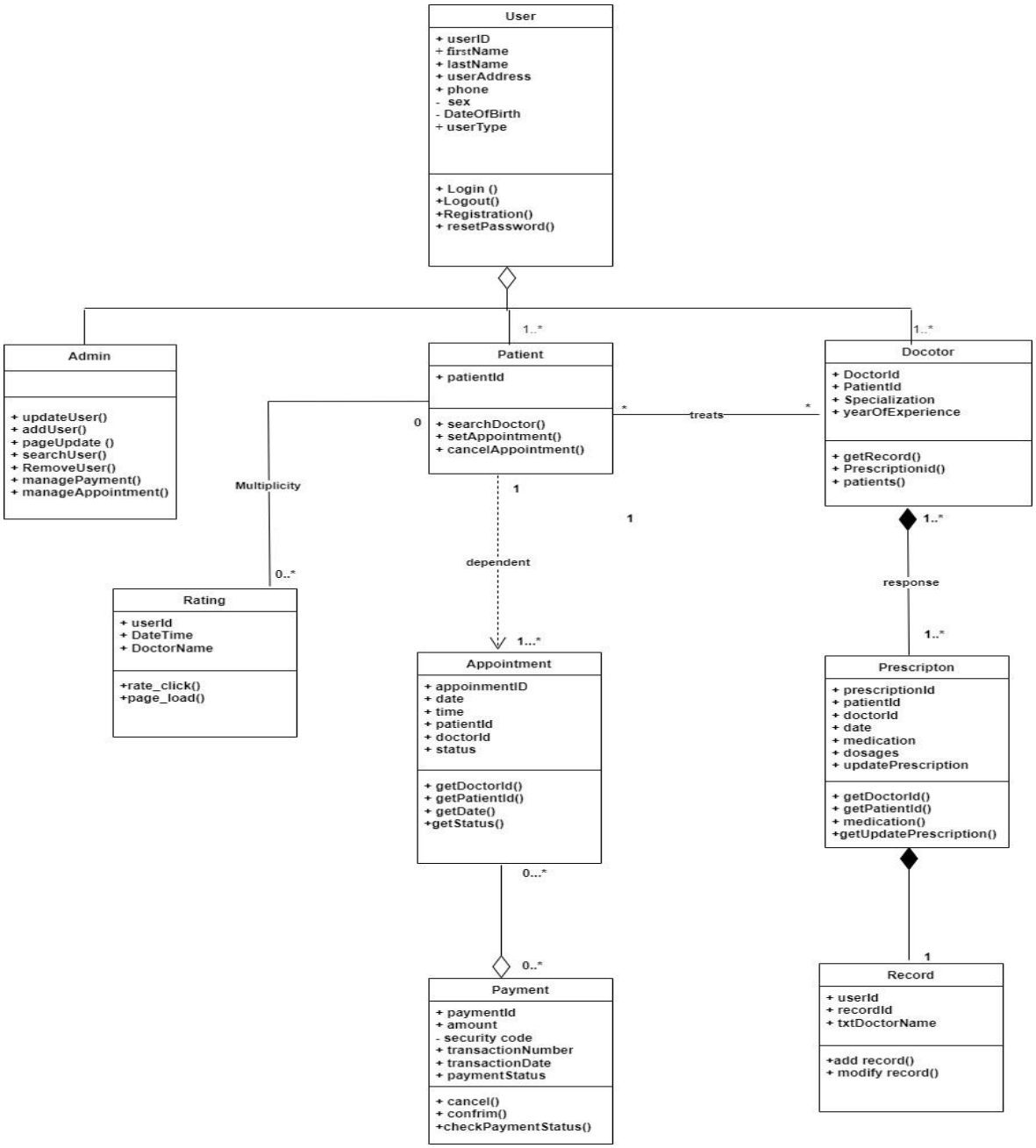

Figure 21: Class diagram.

Data Dictionary:

User Class: This class represents those who are registered or logged into the system as per assigned in the system, authorized user will be able to interact with the system.

Admin Class:

Admin class belongs to the user class and it's a subclass. The admin class inherits all its properties and functions from the user class, while also introducing its own attributes. In contrast to the User class, admin have maximum priority to access on the system, that allows to manage all users and system.

Patient Class:

Patient class belongs to the User class and it's a subclass. The patient's class inherits all its properties and functions from the User class, while also introducing its own attributes. In contrast to the User class, patients can search and choose a doctor, manage appointments, and make payments. Doctor class treats patient class and relation between base two-classes will be many to many.

Doctor Class:

Doctor class belongs to the User class and it's a subclass. The doctor class inherits all its properties and functions from the User class, while also introducing its own attributes. In contrast to the User class, doctors can check patient records, check patients, and give prescription.

Appointment Class:

This class is dependent on patient class, one patient can book one or many appointments. This class managed admin included available date and time.

Payment class:

This class is inherited to appointments class, between classes there are zero to many relationships. Payment will be based doctorId,

Prescription Class:

This class aggregates with doctor class, if doctor class will not exist then this class will be removed from the system, relation between doctor class and prescription is one to many.

Record Class:

This class aggregates with prescription class, if prescription class will not exit then this class will be removed from the system. The doctor will be able to see the previous record and give a prescription.

Rating Class:

Rating will fully depend on patient class, patient class multiplicity with rating class.

4.2.2 State Diagram:

In this practical work state diagram is very important part because it helps to represent the system flow graphically and how this system transitions one state to another, also help to identify potential issues, functionalities. Here I have given two state diagrams, one for registration and login another one for online appointment.

State diagram for User login and registration:

In this diagram, we visualize the states that a user account is exist or not if the user account is already in system, user can login if not, system transitions another state for registration the system. by entering registration details. After a successful registration, the system transitions to the registered state, for the login procedure.

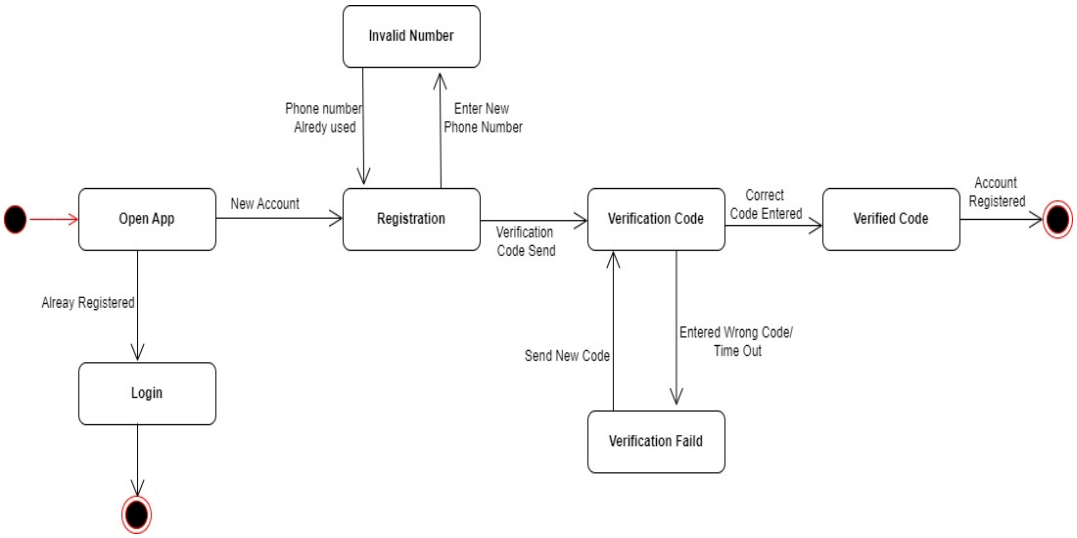

Figure 22 : State Diagram for User Login/Registration.

State diagram for appointment system:

In this diagram start with login page after login state patients able to see doctor list state, then they can select doctor form the list, make appointment, after that make payment state, after payment get confirmation and final state patients have flexibility to change or cancel the appointment.

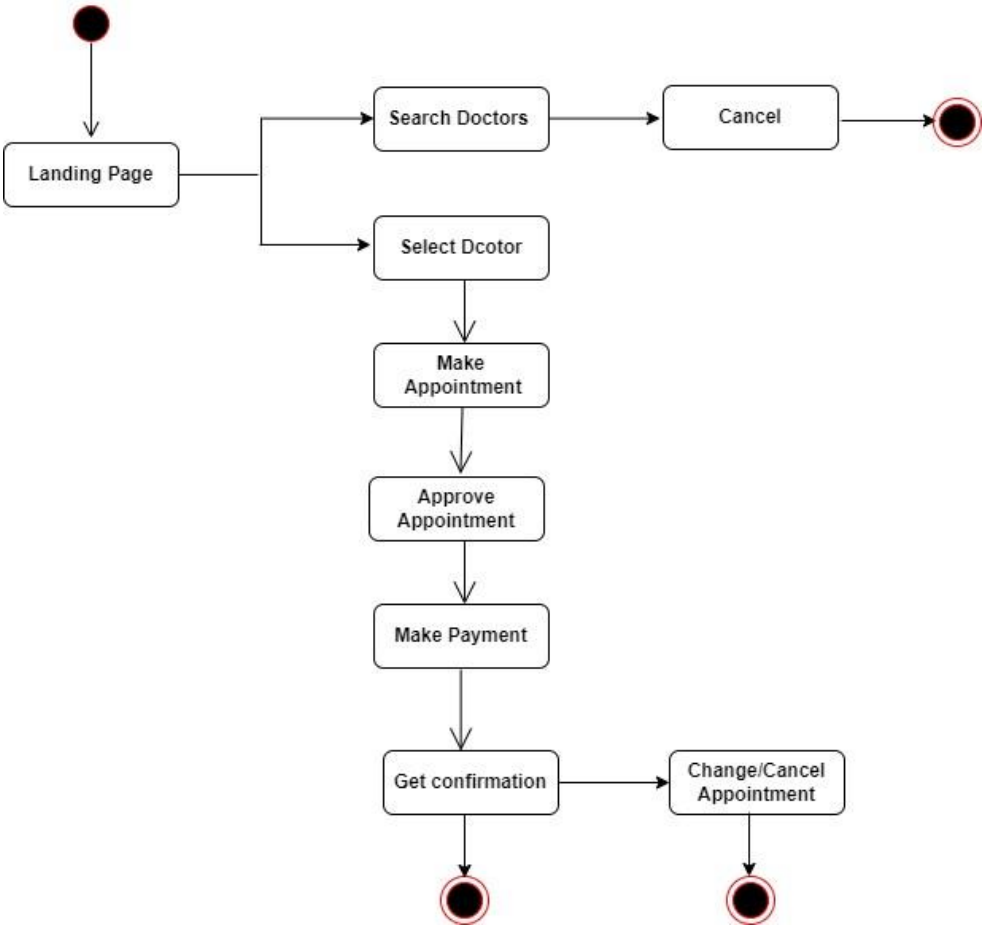

Figure 23: State Diagram for Online Appointment.

4.2.3 Interaction model:

Use case diagram:

The online doctor appointment use case diagram shows the interactions and features of the system. in this system there is three actors: admin, doctor, and patients. The admin, responsible for system management, admin has authority to add or remove doctors, manage appointments, and ensure that the platform works properly. Doctors can log in and view appointment list and confirm or reschedule appointments. Patients can register, able to see available doctors, make

appointments, make payment, and view their medical history. This use case diagram shows the main functionalities and key interactions.

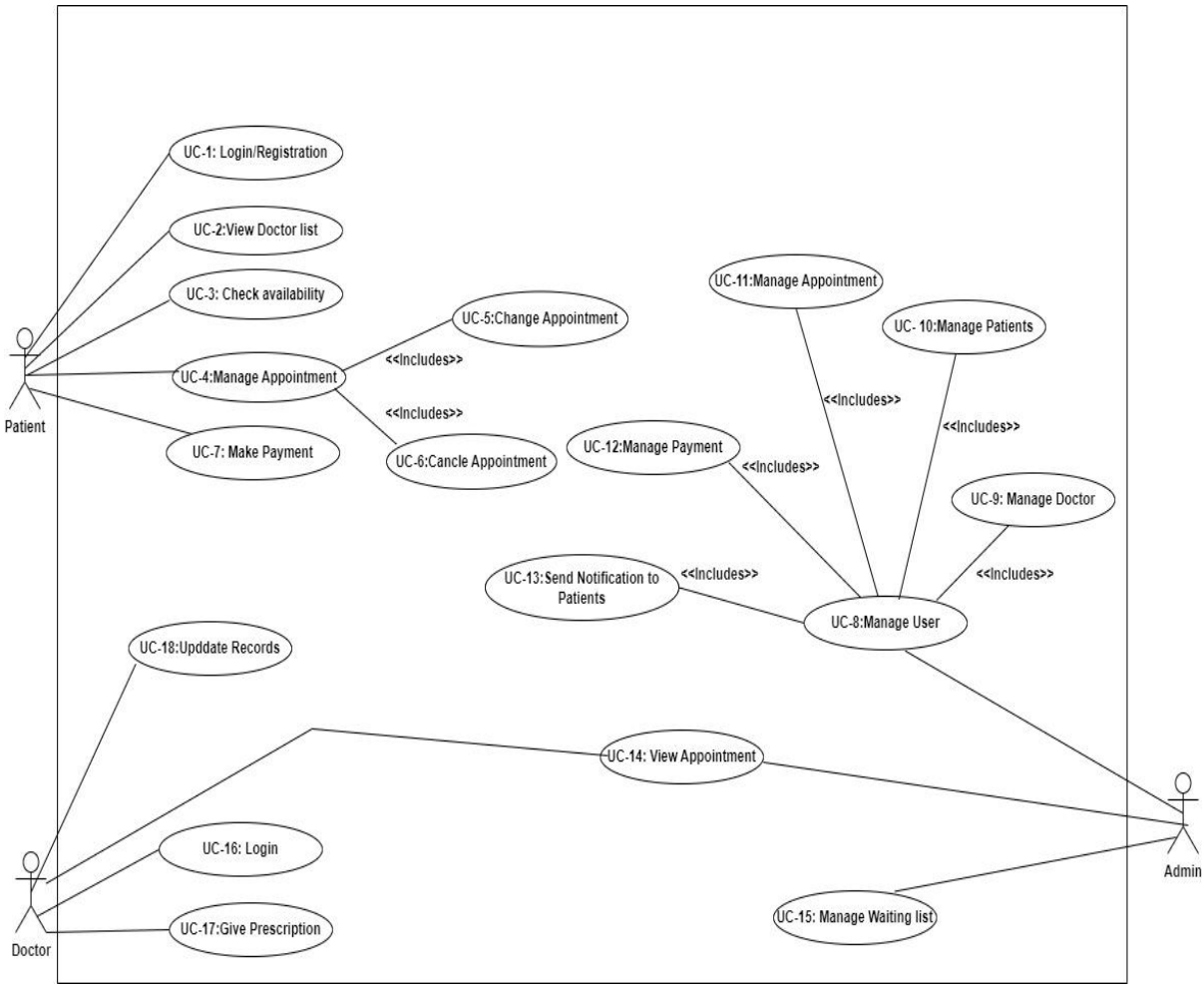

Figure 24: Use case diagram.

Cross check use cases with functional requirements:

The table below shows the execution of at least one functional requirement is appropriate for any use cases. To check the attainment of all the functional requirements this table is used. If the functionality requirement is defined as incorrect it will not match with any use cases which will be then identified as unnecessary.

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13
UC1	X												
UC2			X										
UC3			X										
UC4				X									
UC5				X									
UC6				X									
UC7					X								
UC8										X			
UC9										X			
UC10										X			
UC11												X	
UC12													X
UC13						X							
UC14							X						
UC15											X		
UC16		X											
UC17									X				
UC18								X					

Table- 1: Cross check use cases with functional requirements

Sequential diagram for user Login:

This diagram shows dynamic view of user login in the system between user and the system during the login, it shows step-by-step Interactions. When user start to login request, the diagram shows that the sequential flow of events, such as enter credentials, submit the login details, after that system verifying the information. If credentials are correct, it will show the login is successful and system allow to access the system. otherwise, if enter incorrect user/password the sequence will show how the system handles authentication errors and show the user to re-enter correct credentials.

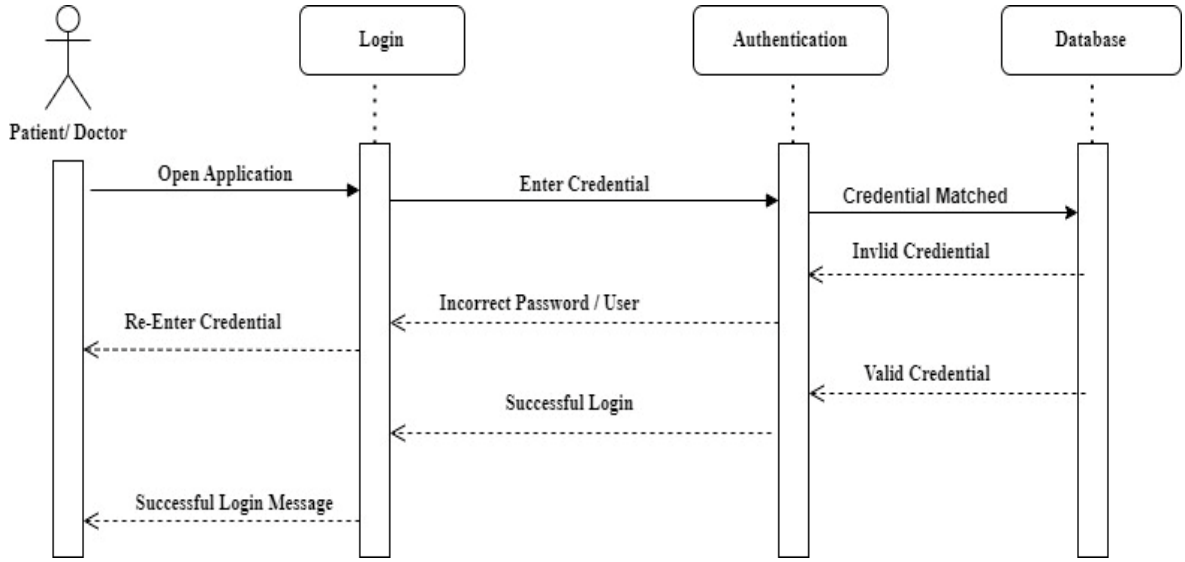

Figure 25: Sequential diagram for User login.

Sequential diagram for Appointment system:

In this diagram represent dynamic view of online appointment system, at first user will register the system admin will approve it and change system control based on user(doctor/patients), patients can view the list of available doctors, select doctor, and request an appointment. The admin reviews it and approves the appointment the request, after that patients will get a notification for payment. Once they paid, system send appointment confirmation, the doctor has ability to access and update the patient's records and give treatment. At the end patients has flexibility to change or cancel appointments as needed.

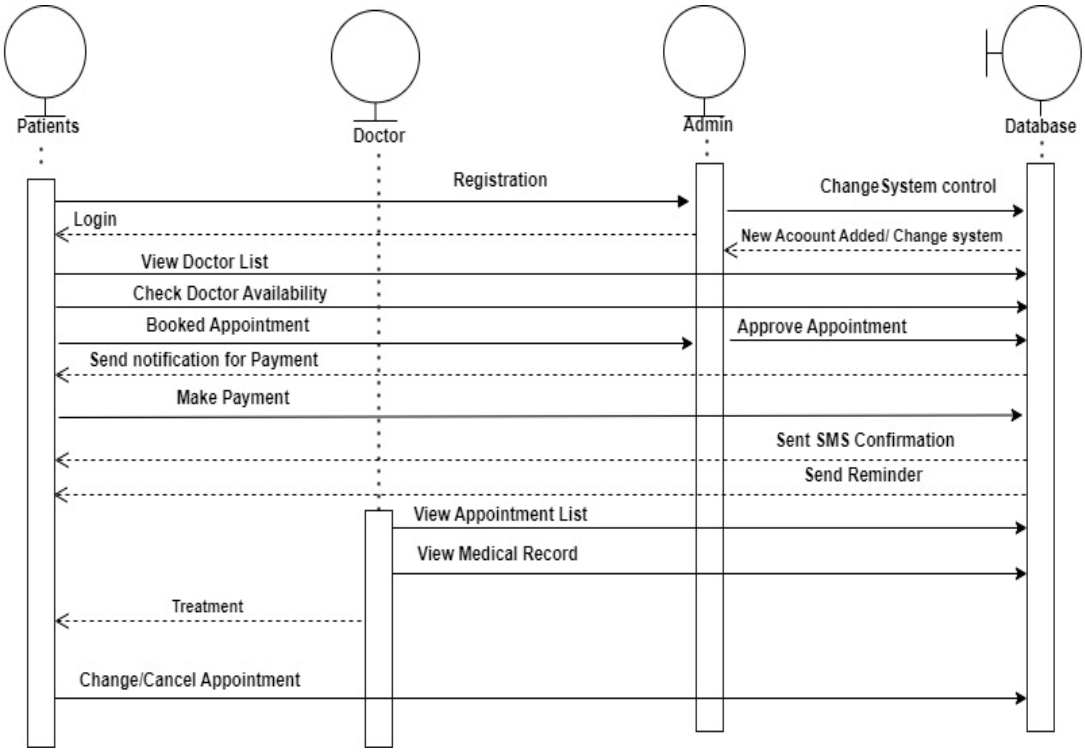

Figure 26: Sequential Diagram for Appointment system.

Activity Diagram:

This diagram represents the overall system activity, after open the system user will see the login page there is options to login if register, if not they can register the system entering their registration details. After login the system patients able to see the doctor list, check their available slot, make appointments, make appointments, get confirmation, they have option to change or cancel appointment and logout the system.

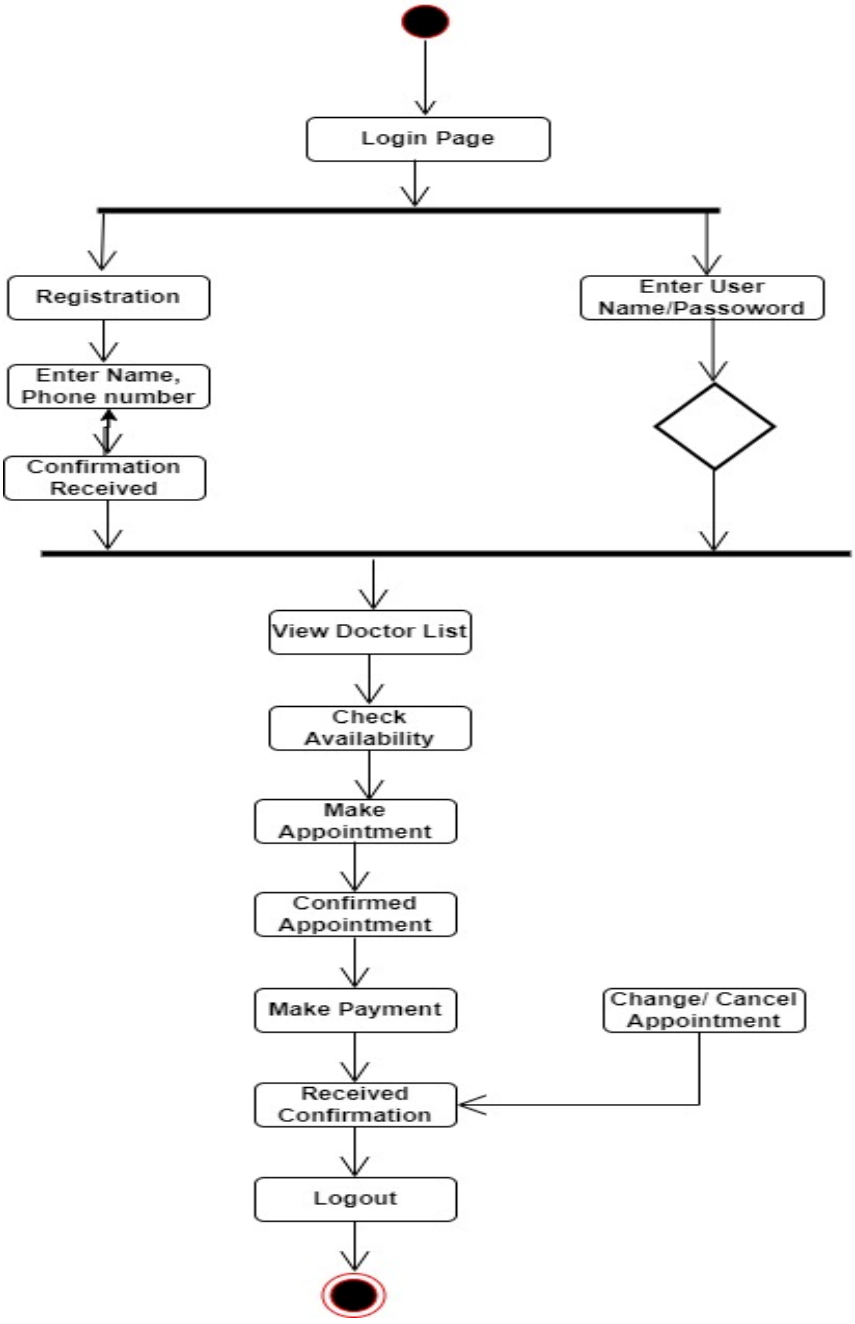

Figure 27: Activity Diagram.

Deployment Diagram:

The Deployment Diagram for the online doctor appointment system establishes a seamless communication flow using TCP/IP and HTTPS protocols. The client-side applications, which are installed on both mobile and desktop platforms, establish a

secure connection to the web server via the TCP/IP protocol. The web server, which utilizes PHP on Apache, enables smooth interactions with clients by handling requests and responses. The entire system securely communicates via the HTTPS protocol, guaranteeing the transfer of encrypted data. The web server securely and efficiently accesses the MySQL database, which is responsible for storing and retrieving information, using the TCP/IP protocol.

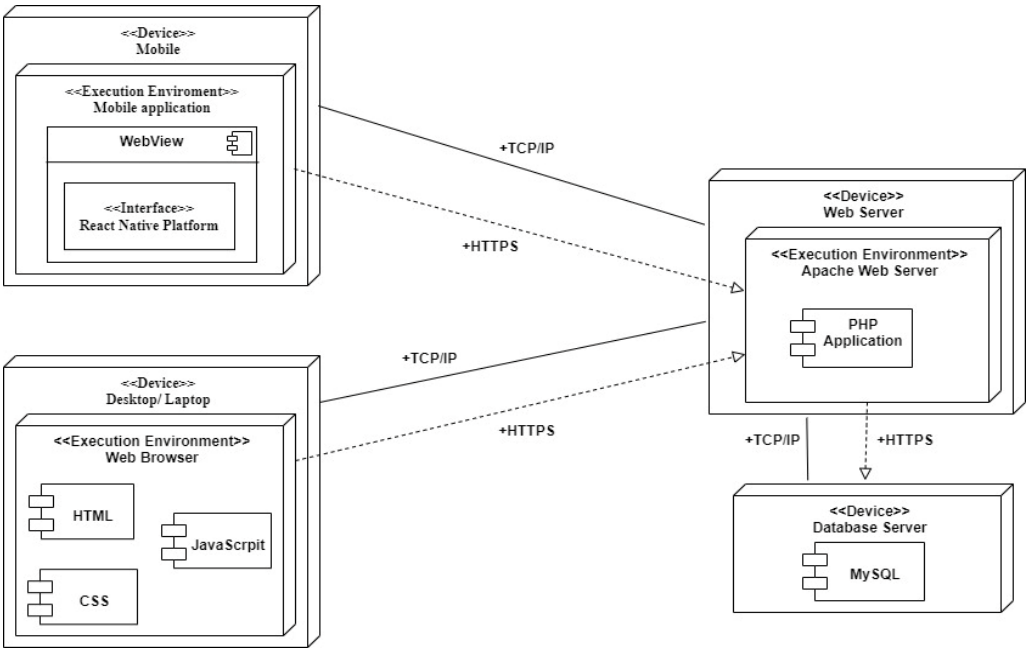

Figure 28: Deployment Diagram.

4.3 System prototype:

Home page of the system:

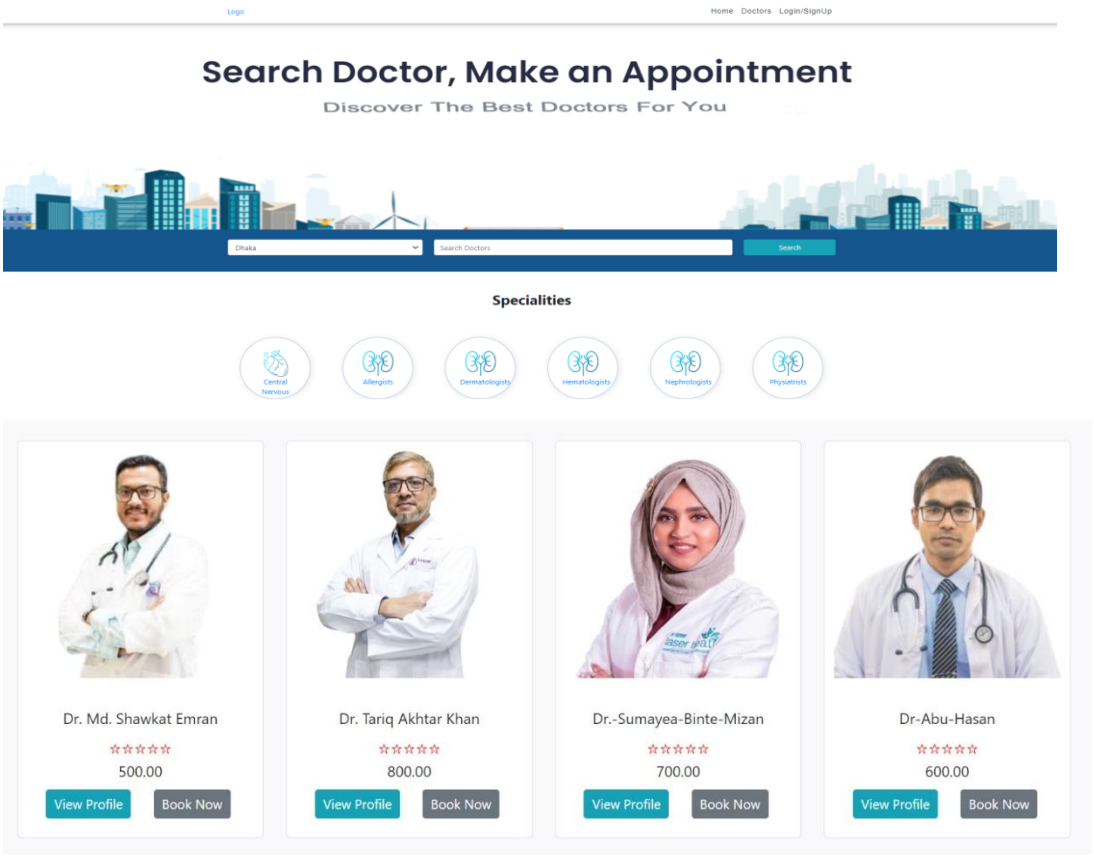

Figure 29: Home page.

Login Login:

To log in the system there are two ways open for the user. user can follow the following steps to get into the system:

1.if they have an email address simply, they will provide their password of the email address and then they will get access to the system.

2. Another way is that they can use their phone number and then they will get an OTP to that number after providing that OTP the patient will get log in to the system.

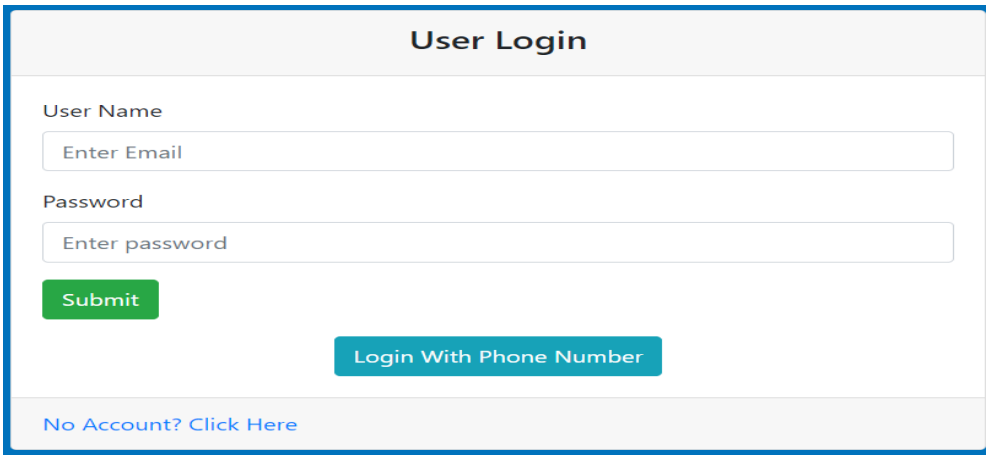

Figure 30: user login using email.

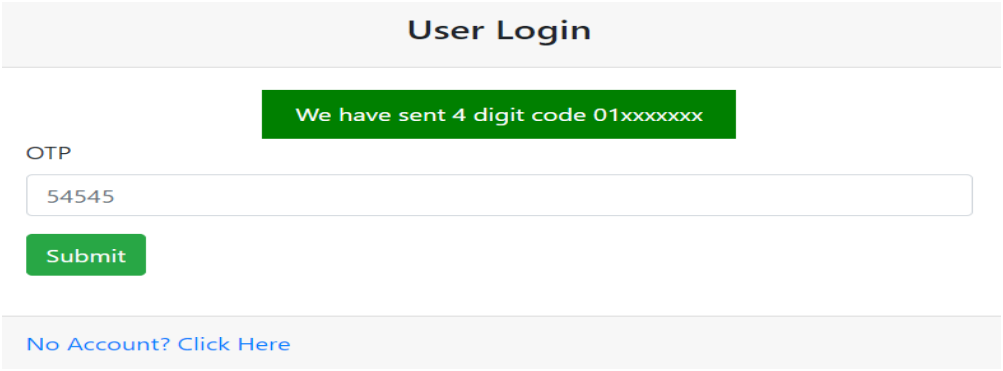

Figure 31: User login using phone.

Registration Page:

If a patient is already registered into the system, they will not do anything here. But if a patient is not registered then a patient will have to provide all those information correctly and after providing all those information the patient will get registered into the system.

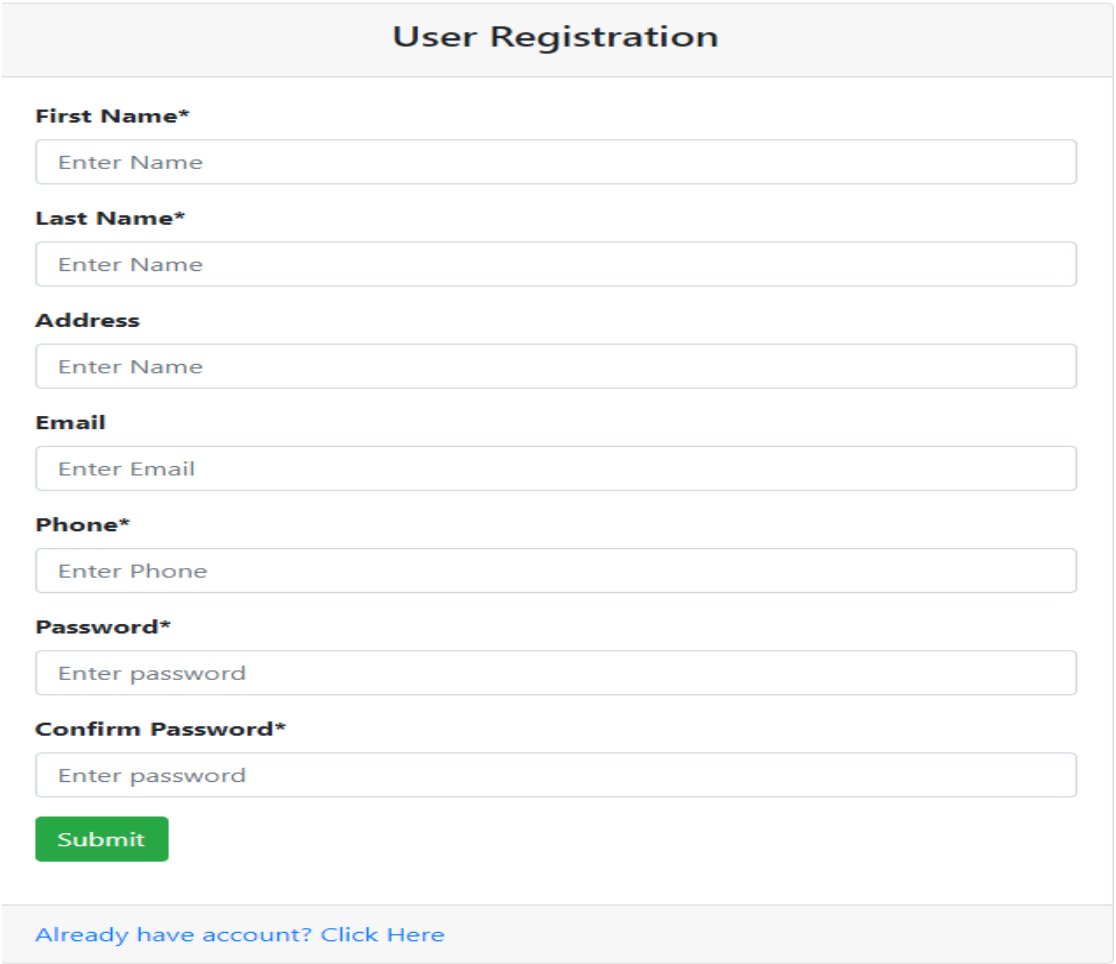

Figure 32: Registration Form.

Search Doctor:

Here the patient will get all the information about their required doctors. They will be able to choose their doctor and their time slots as per their own problems and convenience.

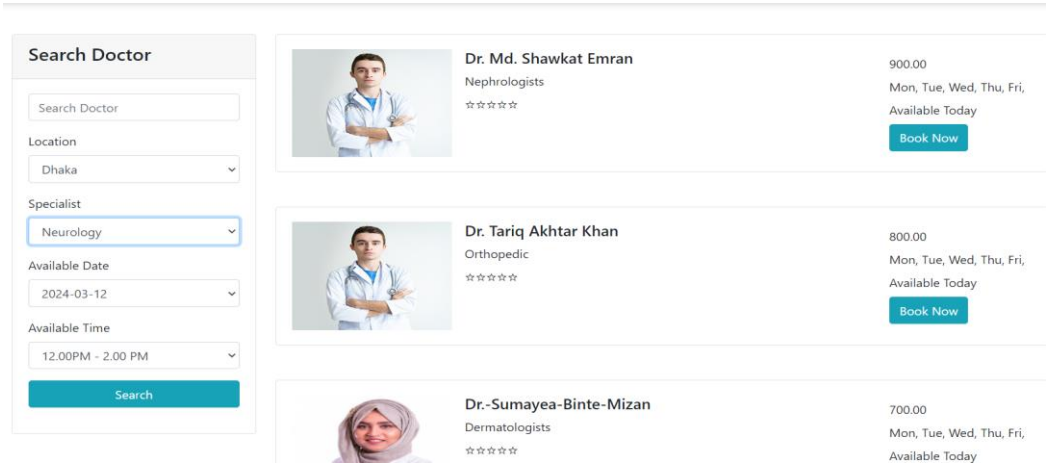

Figure 33: Search Doctor.

Payment:

After confirming an appointment the patient will have to pay the exact amount through bkash app and then their payment will be done .

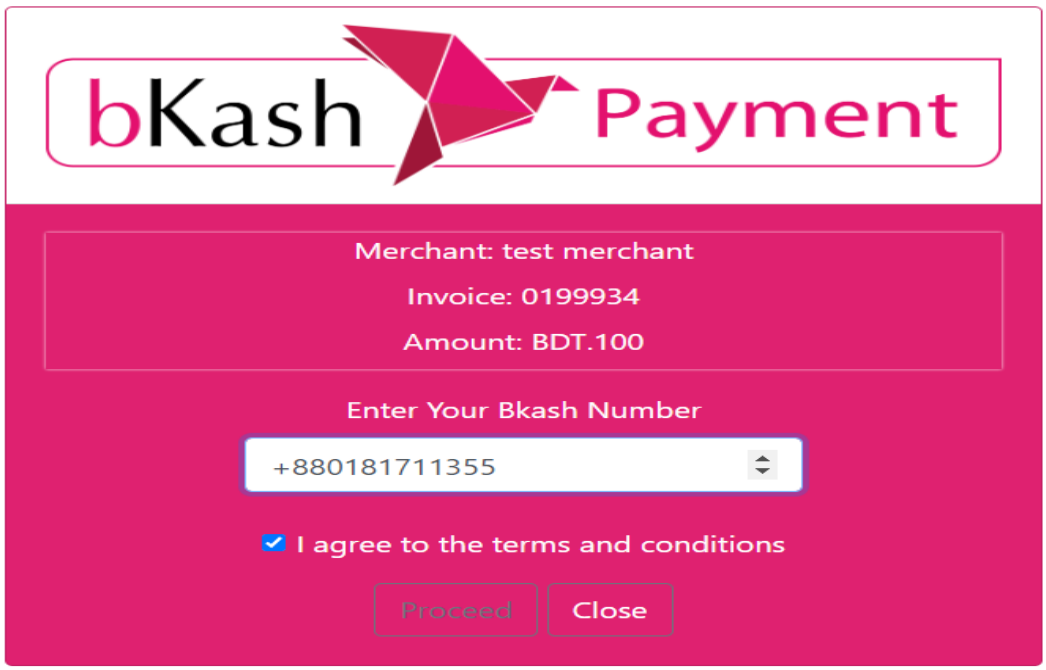

Figure 34:Payment Gateway.

Prescription:

Then the doctor will prescribe medicines according to the problems of the patients and the doctor will also tell the patients for how long they have to take the medicines and how much dose for one medicine they have to consume.

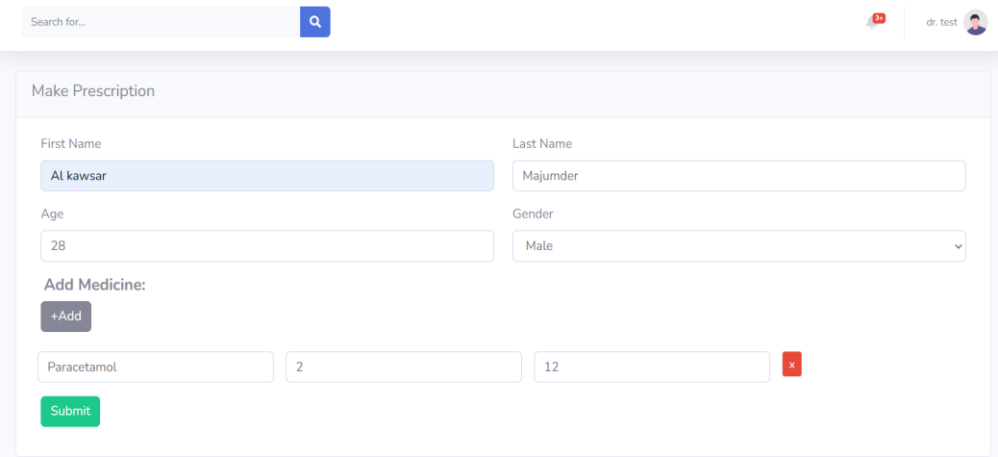

Figure 35: Make prescription.

Appointment list view from doctor end:

Doctors will be able to see the state of patient's appointment left to be prescribe and how many patients the doctors have already prescribed. Moreover, the doctors will also get to know about the patients who cancelled their appointment.

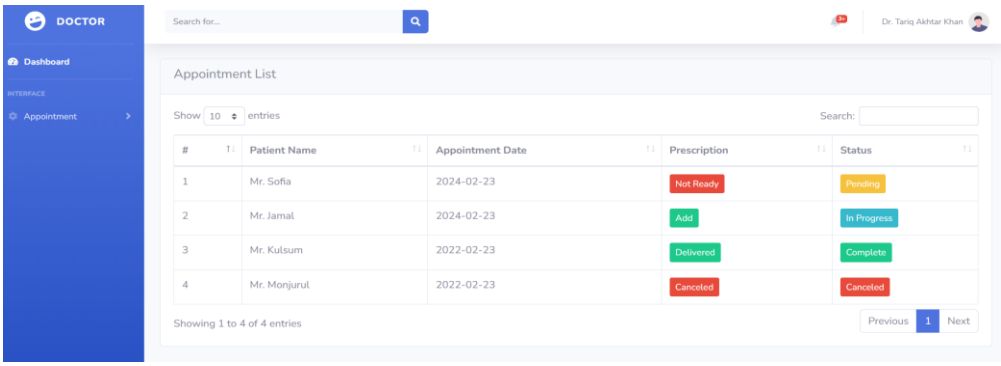

Figure 36: Appointment view by doctor.

Patients manage by Admin:

Admins will be able to manage all the patients. They will be able to add and remove patients.

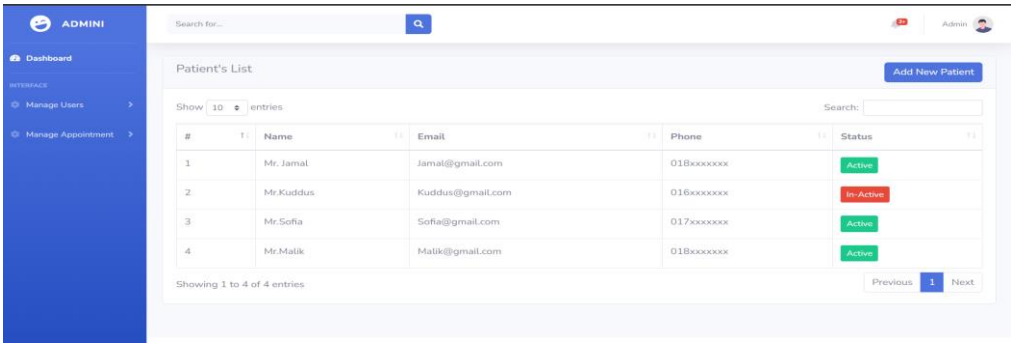

Figure37: Manage Patients.

Manage doctor by Admin:

Admin will be able to manage the list of doctors.

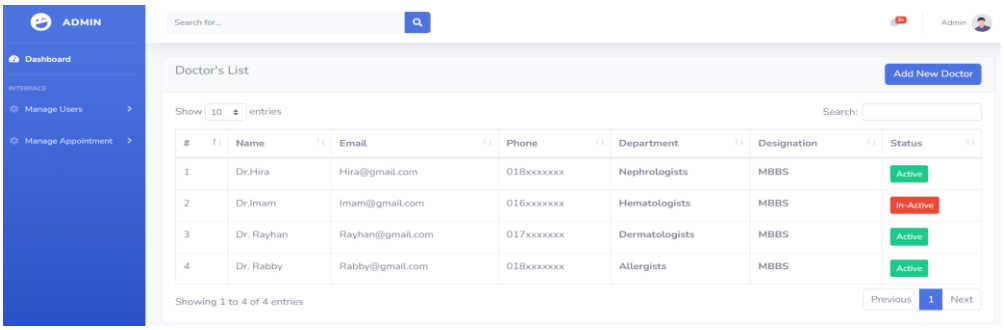

Figure 38: Manage Doctors.

Manage Appointment by Admin:

Appointments are managed by admins. They will be able to see the updated status of appointments.

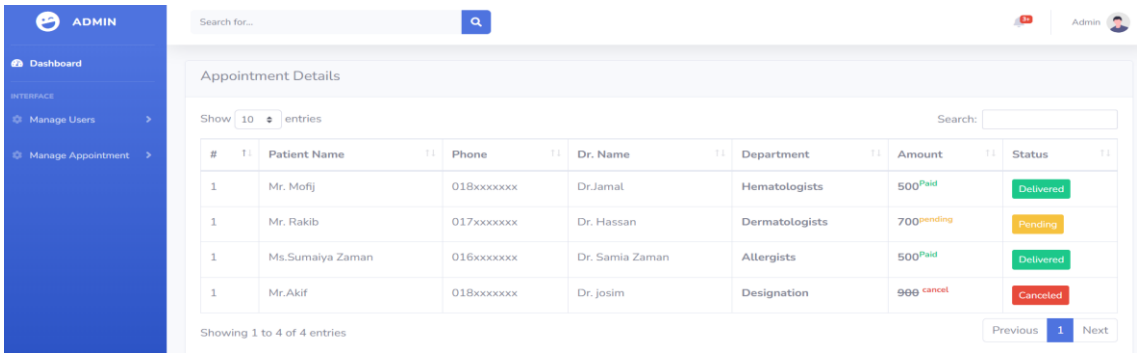

Figure 39: Manage appointment.

4.3 information system testing:

The process of information system testing involves the verification of the implementation of all patients' requirements. Each functional requirement should be allocated with at least one Use Case. The Use Case is use for help to identify all the functionality of the system. Through the process of validation utilizing use case mapping, it was determined that each functional requirement is accompanied by at least one use case. Therefore, upon complete implementation, the system will be able to successfully pass testing and thereby it will satisfy all the specified functional requirements of the system.

The verification of the complete implementation of all non-functional requirements may only be achieved when the system has been fully built. On the contrary, by examining and validating deployment diagram partial verification can be also conducted. At present, the proposed system fully satisfies the entire requirement for the online doctor appointment system.

5. Results and Discussion

As part of this thesis, this system is implemented as prototype, which is not implemented as a fully functional system. Therefore, evaluation of the quality of the implemented system it is not possible by this, to complete the standardized methodologies are used in information system for evaluating the quality. Moreover, non-functional requirements for the system can only be measured after the system's actual implementation. Furthermore, to evaluate all pre-determined requirements this system is designed for both functional and non-functional user requirements. At first, the class diagram was created to analyse including all the user's requirements and to describe the structure of the system. A data dictionary described the class model. After that, state diagram was created to represent the system flow graphically and how this system transitions one state to another state, it also helps to identify potential issues, functionalities. The use case diagram is used to build interaction between user and system. It also helps to check features of the system. This subset was selected to serve as a list of crucial features that had to be included in the initial stages of development. This is generally the functionality for a frequent user with the lowest allocated integrity in the system. Correlated with the scenario's subset were selected for use cases and all other necessary diagrams. Later, the user interface design and interactive prototype were accepted. In this prototype, for the frequent user it is possible to implement all the defined use cases subsets of functionality. Moreover, a prototype was built to have impact in the system as a web- based application. By using mapping use cases for functional requirement verification of the fulfilment of all defined requirements was carried out. Therefore, the created prototype was used to test the requirements that fell into the mentioned subset. Additionally, the system fulfils the requirements for some non-fictional needs, and it also permits for future expansion, object- oriented design ensures easy extensibility. The result of the work is a proposal, the information system has the precondition to simplify book doctor appointment without physically presence. To work with this thesis, I had my own witnessed situation, during the time I lived in Bangladesh, the rural area people of Bangladesh were

facing huge problem in this sector. They are still facing the same issues in this sector. This information system can be a best solution for the rural of Bangladesh. Further development of the system will rely on mobile based application and dynamic web-based application.

6. Conclusion

The main goal of this thesis is to have an effective analysis and proper design of an information system in UML, the support for all the management of the online doctor appointment system will be provided by this, specifically the Bangladeshi rural people will be much benefited with this system they will be able to get health support without any kind of inconvenience situation .The theoretical part of this work relies on the study of professional information sources .For the theoretical starting points the topic dealt with is elaborated in this thesis .This thesis wraps up many sectors all by itself such as the ones with software engineering, information system design ,requirement analysis, UML, system modelling and user interface design .At first , the opinion for software was explained in information engineering . Afterwards, information system was created with the issue of division of information systems, their life cycle, and methodologies. In the first phase of this information system, it analysed the details of development and collection of user requirements. Then it was described with the principles of the UML modelling language as well as with the system modelling. Several terms from the field of creating user interfaces were demonstrated with a few basic rules for their creation at the end of the theoretical part. To have a great grasp of the practical part of the work theoretical starting point was necessary, also the knowledge and skills gained in the theoretical part is implemented in practical part. At the same time one of the sub goals of the work was satisfied. Moreover, in the practical part of the work, the analysis and design of the information system was eventually carried out. The collection and subsequent analysis of users were also accomplished with it. Regarding the analysis requirement in UML modelling language a design was

created. In the proposal different models were introduced in the proposed system such as the class model, state model and interaction model. Creation of these models were a part of partial goal of the work. The next step was taken to create logical design for the single screen of the system. Moreover, a fully functional interactive prototype was created which was established with logical design this was another sub- goal of this work. Furthermore, the most recent situation and recently used way out within the framework were explained for the selected online doctor appointment system. As a result, it was compared with the proposed solution of this system. Finally, a set of recommendations was made for the future development of the system. Based on the analysis conducted, it can be concluded that all the objectives outlined in the assignment of this thesis have been successfully achieved.

7. References

1. *Dhaka Tribune.* [Online] 28 11 2023. https://www.dhakatribune.com/bangladesh/332419/report-68%25-bangladeshis-live-in-villages.

2. Macrotrends. [Online] 2024. https://www.macrotrends.net/countries/BGD/bangladesh/rural-population.

3. Madhab Chandra Das, Muhammad Zakaria, Feng Cheng, Junfang Xu. PubMed Central. [Online] 09 September 2021. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8466760/.

4. *mproving Safety with Information Technology.* David W. Bates, M.D, Atul A. Gawande, M.D., M.P.H. s.l. : New England Journal of Medicine, 2003, Vols. 348(25), . 2526–2534.

5. Taufique Joarder, Lal B. Rawal, Syed Masud Ahmed, Aftab Uddin, Timothy G. Evans. PubMed Certral. [Online] 07 September 2018. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6186485/.

6. Tawsia Tajmim. *The Business Standard.* [Online] 08 April 2022. https://www.tbsnews.net/bangladesh/health/when-doctors-have-no-time-patientseven-private-chambers-399878.

7. Diwakar Gupta, Brian Denton. *Taylor & Francis Online.* [Online] 17 Julay 2008. https://www.tandfonline.com/doi/abs/10.1080/07408170802165880.

8. Tugba Cayirli, Emre A Veral. ResearchGate. [Online] January 2009. https://www.researchgate.net/publication/229881171_Outpatient_scheduling_in_health_care_A_review_of_literature.

9. Athula Wijewickrama, Soemon Takakuwa. IEEE Xplore. [Online] 2008. https://ieeexplore.ieee.org/abstract/document/4736238.

10. Yasin Ali, Muhammad rafay Anis Ur Rehman,Meng Kheang Sorn,Raja Danish Akbar Khan. ResearchGate. [Online] January 2023. https://www.researchgate.net/publication/370562909_Traffic_Problems_in_Dhaka_City_Causes_Effects_and_Solutions_Case_Study_to_Develop_a_Business_Model#:~:text=The%20lack%20of%20public%20transportation,air%20pollution%20and%20economic%20costs..

11. Kenneth C. Laudon and Jane P. Laudon. pearson. [Online] 2020. https://www.pearson.com/nl/en_NL/higher-education/subject-catalogue/information-systems/Laudon-management-information-systems-digital-firm-16e.html.
12. George Marakas, James O'Brien. 2018.
13. Zemmouchi-Ghomari, Leila. *intechopen.* [Online] 15 July 2021. https://www.intechopen.com/chapters/76540.
14. Laudon, K. C., & Laudon, J. P. s.l. : Pearson, 2020.
15. Stair, R., & Reynolds, G. s.l. : Cengage Learning, 2018.
16. Royce, W. W. s.l. : IEEE WESCON, 1970.
17. Sommerville, I. s.l. : Addison-Wesley, 2011.
18. *binaryterms.* [Online] 2024. https://binaryterms.com/waterfall-process-model.html.
19. Pressman, R. S. s.l. : McGraw-Hill Education, 2014.
20. Boehm, B. W. 1988.
21. Dmitry Gurendo. *xbsoftware.* [Online] 26 October 2015. https://xbsoftware.com/blog/software-development-life-cycle-spiral-model/.
22. Cockburn, A., Highsmith, J., & Meyers, T. . s.l. : IEEE Computer Society, 2001.
23. Benjamin Hermitte. *wimi.* [Online] 4 October 2023. https://www.wimi-teamwork.com/blog/how-to-use-the-agile-method-in-remote-work/.
24. Beck, K., Beedle, M., van Bennekum, A., et al. agilemanifesto. [Online] 2001. https://agilemanifesto.org/.
25. Highsmith, J. s.l. : Addison-Wesley, 2002.
26. Schwaber, K., & Sutherland, J. scrum.org. [Online] 2017. https://www.scrum.org/.
27. Sharma, Monika. *includehelp.* [Online] 05 April 2023. https://www.includehelp.com/basics/the-prototyping-model-software-engineering.aspx.
28. Roberts, Sienna. theknowledgeacademy. [Online] 21 September 2023. https://www.theknowledgeacademy.com/blog/prototype-model-in-software-engineering/.
29. Pressman, R. S. s.l. : McGraw-Hill Education, 2014.
30. Boehm, B. W. s.l. : ACM SIGSOFT Software Engineering Notes, 1988.
31. *Notepub.* [Online] 2 April 2021. https://notepub.io/notes/software-engineering/software-development-life-cycle/sdlc-incremental-model/.
32. Wiegers, K. E., & Beatty, R. *Software Requirements.* s.l. : Microsoft Press., 2013.

33. Pressman, R. S. *Software Engineering.* s.l. : McGraw-Hill Education, 2014.
34. Kotonya, G., & Sommerville, I. *Requirements Engineering:*. 1998. John Wiley & Sons.
35. Goguen, J. A., & Linde, C. *Techniques for Requirements Elicitation.* s.l. : ACM SIGSOFT, 1993. 39–48.
36. Dennis, A., Wixom, B. H., & Tegarden, D. *Systems Analysis and Design.* s.l. : Wiley, 2015.
37. Hatice Koç, Ali Mert Erdoğan, Yousef Barjakly, Serhat Peker. MDPI. [Online] 10 March 2021. https://www.mdpi.com/2504-3900/74/1/13.
38. Walker, Alyssa. guru99. [Online] 29 December 2023. https://www.guru99.com/uml-diagrams.html.
39. Studys Ection. [Online] April 2021. https://studysection.com/blog/history-of-unified-modeling-language-uml/.
40. *javatpoint.* [Online] 2024. https://www.javatpoint.com/uml.
41. *uah.* [Online] 2024. https://www.cs.uah.edu/~rcoleman/Common/SoftwareEng/UML.html.
42. miro. [Online] 2024. https://miro.com/diagramming/what-is-a-uml-diagram/.
43. *Creately.* [Online] 28 September 2022. https://creately.com/blog/diagrams/uml-diagram-types-examples/.
44. *creately.* [Online] 28 September 2022. https://creately.com/blog/diagrams/uml-diagram-types-examples/.
45. *Creately.* [Online] 5 January 2023. https://creately.com/blog/software-teams/class-diagram-tutorial/.
46. Lucidchart. [Online] 2024. https://www.lucidchart.com/pages/uml-class-diagram.
47. *Lucidchart.* [Online] 2024. https://www.lucidchart.com/pages/uml-class-diagram.
48. *lucidchart.* [Online] 2024. https://www.lucidchart.com/pages/uml-class-diagram.
49. Guthrie, Georgina. Nulab. [Online] 10 March 2022. https://nulab.com/learn/software-development/a-simple-guide-to-drawing-your-first-state-diagram-with-examples/.
50. *FHIR.* [Online] 2024. https://fhir.org/guides/argonaut/scheduling/state-diagram.html.
51. *Smartdraw.* [Online] 2024. https://www.smartdraw.com/use-case-diagram/.
52. Rafael J. BARROS. *hackmd.* [Online] 29 September 2021. https://hackmd.io/@profraf/r18d49CXF.

53. Ambysoft Inc. *agilemodeling.* [Online] 2022. https://agilemodeling.com/artifacts/sequencediagram.htm.
54. *drawio.* [Online] 16 May 2022. https://www.drawio.com/blog/sequence-diagrams.
55. *MindManager.* [Online] 2024. https://www.mindmanager.com/en/features/activity-diagram/.
56. Harry. *Boardmix.* [Online] 17 Feb 2023. https://boardmix.com/tips/uml-activity-diagram/.
57. Fowler, Martin. *Amazon.* [Online] 2004. https://www.amazon.com/UML-Distilled-Standard-Modeling-Language/dp/0321193687.
58. *sparxsystems.* [Online] 2024. https://sparxsystems.com/resources/tutorials/uml2/component-diagram.html.
59. JavatPoint. [Online] 2021. https://www.javatpoint.com/uml-deployment-diagram.
60. *Lucidchart.* [Online] 2024. https://www.lucidchart.com/pages/uml-deployment-diagram.
61. Object Management Group (OMG). omg.org. [Online] 2017. https://www.omg.org/spec/UML/2.5/.
62. Ambler, S. W. Agile Modeling. [Online] 2004. http://agilemodeling.com/artifacts/communicationDiagram.htm.
63. Fakhroutdinov, Kirill. *uml-diagrams.* [Online] https://www.uml-diagrams.org/communication-diagrams.html.
64. Berners-Lee, T. *W3C.* [Online] 1991. https://www.w3.org/History/19921103-hypertext/hypertext/WWW/MarkUp/Tags.html.
65. *W3C.* [Online] 1996. https://www.w3.org/Style/CSS/.
66. Meyer, E. A. *CSS.* s.l. : O'Reilly Media, 2007.
67. mozilla. [Online] 2024. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide.
68. Flanagan, D. *The Definitive Guide.* s.l. : O'Reilly Media, 2011.
69. Crockford, D. *The Good Parts.* s.l. : O'Reilly Media, 2008.
70. Lerdorf, R., Tatroe, K., & MacIntyre, P. *Programming PHP.* s.l. : O'Reilly Media, 2006.
71. Williams, R., & Lane, S. . *Professional PHP7.* s.l. : Wiley, 2016.

72. *php.* [Online] 2024. https://www.php.net/.
73. Date, C. J. *An Introduction to Database Systems.* s.l. : Addison-Wesley, 2003.
74. Silberschatz, A., Korth, H. F., & Sudarshan, S. *Database System Concepts.* s.l. : McGraw-Hill, 2010.
75. Ullman, L., & Widom, J. . *A First Course in Database Systems.* s.l. : Pearson, 2019.
76. *mysql.* [Online] 2024. https://www.mysql.com/.
77. Momjian, B. *Introduction and Concepts.* s.l. : Addison-Wesley, 2001.
78. Riggs, S. . *PostgreSQL Administration Essentials.* s.l. : Packt Publishing, 2016.
79. *PostgreSQL.* [Online] 2024. https://www.postgresql.org/.
80. Fitzgerald, Anna. HubSpot. [Online] 07 February 2024. https://blog.hubspot.com/website/ui-design.
81. Blackman, Katie. Toptal. [Online] 2 December 2020. https://www.toptal.com/designers/ui/job-description.
82. WHITE, CAROLINE. Careerfoundry. [Online] 30 NOVEMBER 2023. https://careerfoundry.com/en/blog/ux-design/what-does-a-ux-designer-actually-do/.
83. Columbia Engineering. [Online] 2020. https://bootcamp.cvn.columbia.edu/blog/what-is-ux-design/.
84. Verma, Ankita. *linkedin.* [Online] 20 September 2022. https://www.linkedin.com/pulse/ux-vs-ui-design-ankita-verma/.
85. Usability.de. [Online] 2024. https://www.usability.de/en/usability-user-experience.html.
86. Nielsen, Jakob. Nielsen Norman Group logoNielsen Norman Group. [Online] 3 January 2012. https://www.nngroup.com/articles/usability-101-introduction-to-usability/.
87. Hierso, Severine. *McGaw.* [Online] 30 August 2022. https://mcgaw.io/blog/product-usability-metrics/#gs.504k04.
88. G., Angie. thedesignership. [Online] 29 September 2023. https://www.thedesignership.com/blog/low-vs-high-fidelity-wireframes.
89. Hanak, Alec. digitalsilk. [Online] 18 December 2023. https://www.digitalsilk.com/digital-trends/what-is-a-wireframe/.
90. G., Angie. thedesignership. [Online] 29 September 2023. https://www.thedesignership.com/blog/low-vs-high-fidelity-wireframes.

91. Lenczewska, Żaneta. pagepro. [Online] 25 March 2022. https://pagepro.co/blog/what-is-a-software-prototype-and-how-to-build-one/.
92. Sheetal Sharma, Darothi Sarkar,Divya Gupta. ResearchGte. [Online] 2012. https://www.researchgate.net/publication/267706023_Agile_Processes_and_Methodologies_A_Conceptual_Study.
93. David Cohen, Mikael Lindvall,Patrícia Costa. ReseachGate. [Online] 2004. https://www.researchgate.net/publication/222704289_An_Introduction_to_Agile_Methods.
94. Chandana. *simplilearn.* [Online] 27 June 2023. https://www.simplilearn.com/agile-manifesto-and-its-twelve-principles-article.
95. Gosia Podżorska. *white boards.* [Online] 30 09 2022. https://whiteboards.io/blog/types-of-agile-methodologies/.
96. Madhab Chandra Das, Muhammad Zakaria, Feng Cheng, Junfang Xu. PubMed Central. [Online] 09 September 2021. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8466760/.
97. C Dennis O'Hare,John Corlett. GesearchGate. [Online] March 2004. https://www.researchgate.net/publication/7390529_The_outcomes_of_open-access_scheduling.
98. ResearchGate. [Online] 2003. https://www.researchgate.net/publication/229881171_Outpatient_scheduling_in_health_care_A_review_of_literature.
99. Semantic Scholar. [Online] 01 March 2003. https://www.semanticscholar.org/paper/Scheduling-doctors%27-appointments%3A-optimal-and-Robinson-Chen/ac97a6644879e24f8b6d0c52361cbec5d89e1b6c.
100. Springer Link. [Online] 01 Auguest 1985. https://link.springer.com/article/10.1057/jors.1985.128.
101. Science Direct. [Online] November 1979. https://www.sciencedirect.com/science/article/abs/pii/0377221779902455.
102. Bailey, N. Semantic Scholar. [Online] 01 July 1952. https://www.semanticscholar.org/paper/A-Study-of-Queues-and-Appointment-Systems-in-with-Bailey/bb47559e105f890395f05e0056ef7d2ec069e317.

103. Kum Khiong Yang, Mun Ling Lau, Ser Aik Quek. Online Library. [Online] 07 December 1998. https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291520-6750%28199804%2945%3A3%3C313%3A%3AAID-NAV5%3E3.0.CO%3B2-A.
104. lucidchart. [Online] 2024. https://www.lucidchart.com/pages/uml-class-diagram.
105. FHIR. [Online] 2024. https://fhir.org/guides/argonaut/scheduling/state-diagram.html.
106. Lekan, Akinode John. researchGate. [Online] December 2017. https://www.researchgate.net/figure/Use-case-diagram-for-Patient_fig1_332864696.
107. draw.io. drawio. [Online] 16 May 2022. https://www.drawio.com/blog/sequence-diagrams.
108. Harry. Boardmix. [Online] 17 Feb 2023. https://boardmix.com/tips/uml-activity-diagram/.
109. Lucidchart. [Online] 2024. https://www.lucidchart.com/pages/uml-deployment-diagram.
110. Verma, Ankita. linkedin. [Online] 20 September 2022. https://www.linkedin.com/pulse/ux-vs-ui-design-ankita-verma/.
111. Hierso, Severine. McGaw. [Online] 30 August 2022. https://mcgaw.io/blog/product-usability-metrics/#gs.504k04.
112. Fakhroutdinov, Kirill. uml-diagrams. [Online] 2009. https://www.uml-diagrams.org/communication-diagrams.html.
113. *sparxsystems.* [Online] 2024. https://sparxsystems.com/resources/tutorials/uml2/component-diagram.html.
114. *Lucidchart.* [Online] 2024. https://www.lucidchart.com/pages/uml-class-diagram.
115. *Creately.* [Online] 28 September 2022. https://creately.com/blog/diagrams/uml-diagram-types-examples/.
116. Schwaber, K., & Sutherland, J. Scrumguides. [Online] 2017. https://scrumguides.org/scrum-guide.html.
117. Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Kern, J. agilemanifesto. [Online] 2001. http://agilemanifesto.org/.
118. softwaretestinghelp. [Online] 23 June 2023. https://www.softwaretestinghelp.com/agile-manifesto/.

119. Saigon Technology. [Online] 05 NOV 2018. https://saigontechnology.com/blog/agile-development-process-in-software-outsourcing.
120. productplan. [Online] 2024. https://www.productplan.com/glossary/agile-principles/.
121. pm-partners. [Online] 23 June 2021. https://www.pm-partners.com.au/insights/the-agile-journey-a-scrum-overview/.
122. *Ideas.* [Online] 1999. https://ideas.repec.org/a/wly/navres/v46y1999i5p549-559.html.
123. Laudon, Kenneth C. Laudon and Jane P. pearson. [Online] 2020. https://www.pearson.com/nl/en_NL/higher-education/subject-catalogue/information-systems/Laudon-management-information-systems-digital-firm-16e.html.
124. Kenneth C. Laudon and Jane P. Laudon. 2020.
125. Sharda, Ramesh, Delen, Dursun and Turban, Efraim. s.l. : Pearson, 2014.
126. DT Bourgeois, JL Smith, S Wang, J Mortati. 2014.
127. Gray, J., & Reuter, A. s.l. : Morgan Kaufmann, 1993.
128. *binaryterms.* [Online] 2024. https://binaryterms.com/waterfall-process-model.html.
129. Leila Zemmouchi-Ghomari. *intechopen.* [Online] 15 July 2021. https://www.intechopen.com/chapters/76540 .
130. —. *intechopen.* [Online] 15 July 2021. https://www.intechopen.com/chapters/76540 .
131. Dmitry Gurendo. *xbsoftware.* [Online] 26 October 2015. https://xbsoftware.com/blog/software-development-life-cycle-spiral-model/.
132. Benjamin Hermitte. wimi. [Online] 4 October 2023. https://www.wimi-teamwork.com/blog/how-to-use-the-agile-method-in-remote-work/.
133. geeksforgeeks. [Online] 2024. https://www.geeksforgeeks.org/advantages-and-disadvantages-of-prototype-model/.
134. Sharma, Monika. includehelp. [Online] 05 April 2023. https://www.includehelp.com/basics/the-prototyping-model-software-engineering.aspx.
135. Notepub. Notepub. [Online] 2 April 2021. https://notepub.io/notes/software-engineering/software-development-life-cycle/sdlc-incremental-model/.
136. Rafael J. BARROS. *hackmd.* [Online] 29 September 2021. https://hackmd.io/@profraf/r18d49CXF.

8. List of pictures, abbreviations

8.1 List of pictures

8.2 List of abbreviations

UML- Unified modeling language

HTML-Hypertext Markup Language

CSS- Cascading Style Sheets

PHP- Hypertext Preprocessor

HTTPS-Hypertext transfer protocol secure

TCP- Transmission Control Protocol

OOP-Object-oriented programming

OMG- Object Management Group

UI- User Interface

UX- User Experience

Appendix

Use case	Login
Description	Login to the system
Actors	Doctor, Admin, patient
Pre- Condition	Admin/doctor need to have valid user ID and password.
Basic flow	• Actors identify itself admin or doctor. • Enter id and password. • System will check entered id and password. • If entered id password are valid system will allow to give access and display the main page.
Alternative flow	If entered id and password wrong, system will deny access to the system.

Use case Description for doctor and admin.

Use case Description for register patient.

Use case	New patient register
Description	New patients register to the system
Actor	Patients
Pre- Condition	Patient needs to login the system
Basic flow	• For register new patients need to enter all personal details. • Confirm registration massage will be display in the system.
Alternative flow	If enter wrong information system will show warning

Use case Description for manage appointments.

Use case	Appointment manages
Description	Patient Appointment manage
Actor	Admin and patient
Pre- Condition	Actors need to logon to the system first
Basic flow	• Patients need to make an appointment. • Patients can reschedule before make payment. • After payment patient will get confirmation via email/phone.
Alternative flow	If patient enter wrong details, system will show warning.

Use case Description for patient account details.

Use case	Patient account details
Description	To access patients' system logs
Actor	Admin
Pre- Condition	Admin needs to login to the system
Basic flow	• Admin will allow to see all details about patient system logs. • Admin will allow to modify patients' information if needed.
Alternative flow	If entered different kind of details, system will display warning message.

Use case description for rating.

Use case	Rating
Description	To post doctor rating
Actor	patient
Pre- Condition	Patients need to login to the system and finish the Appointment.
Basic flow	• Patients need to login the system. • Patients need to confirm the appointment and payment. • Consultant with doctor. • Review the doctor service and give rating.
Alternative flow	If patient does not post, system will prompt

Source code of this system given below Github link:

https://github.com/Alkawsar28/online_doctor_appointment_system