**Project Title: Dental Clinic Management System**

Phase II: User Requirements and Application Specifications

Team Leader name: Enisa Mishka

Submission Deadline: 18.03.2024

**1. Chosen Development Model:**

My project is being developed using the agile methodology because of its built-in adaptability, flexibility, and iterative nature. In the context of a dental clinic project, where requirements might evolve based on patient needs, regulatory changes, or advancements in technology, Agile allows for seamless incorporation of changes throughout the development process. It promotes collaboration between developers and stakeholders, ensuring that the final product meets the evolving needs of the clinic efficiently. Additionally, Agile also makes it possible to provide functional software frequently, which is essential for feature testing and validation in actual clinical settings. Overall things considered, Agile is well adapted to the fast-paced healthcare sector and guarantees the timely delivery of a customized solution for the dental clinic.

**2. User Requirements:**

**a. Stakeholders:**

Software that enables all the services performed by a dental clinic to be automated is an efficient method for a business that operates with a large number of patients and must keep data for each of them. This product that I want to offer for the dental clinic includes not only the internal part that works inside the clinic but all those interested in the services offered. It is designed to meet the requirements of all our stakeholders Dental Clinic Management System following the way this clinic operates.

**Doctors:**

Roles: Treating and diagnosing dental problems in patients is the primary responsibility of dentists. To deliver high-quality care, they need access to patient data, treatment plans, and diagnostic equipment.

Interests: A system that facilitates efficient patient management, allows dentists to operate more efficiently, gives them access to up-to-date patient data, and helps them communicate effectively with patients and other staff members is something that interests them.   
  
**Receptionists:**  
Functions: Receptionists are essential in scheduling appointments, responding to patient questions, processing insurance claims. They are in charge of making sure clinic operations run well and are frequently the initial point of contact for patients.

Interests: Receptionists are interested in a system that simplifies appointment scheduling, automates billing and insurance processes, facilitates effective communication with patients, and provides tools for managing administrative tasks efficiently.

**Patients:**

Roles: Patients are the primary recipients of dental care services provided by the clinic. Their roles include seeking treatment for dental issues, attending appointments, providing accurate medical history and information, and following post-treatment care instructions.

Interests: Patients are looking for a system that allows them to easily schedule appointments, keeps track of appointments in a timely manner, and protects their privacy and the confidentiality of their personal health information.

**b. User Stories:**

* Dentist: As a dentist, I need access to patient medical records, including treatment history and diagnostic images, so that I can provide informed and effective treatment.

Requirement: Develop a secure and centralized database for storing patient records, including treatment history, X-rays, scans, and notes.

Benefit: Dentists can make better treatment decisions, provide personalized care, and improve patient outcomes by having all relevant information readily accessible during appointments.

* End-User: As a patient, I want to be able to schedule appointments online so that I can conveniently choose a time that fits my schedule without having to call the clinic during working hours.

Requirement: Implement an online appointment scheduling system where patients can view available time slots, select an appointment, and receive confirmation.

Benefit: This feature saves time for patients and reduces the workload on clinic staff by automating appointment scheduling, leading to improved patient satisfaction and operational efficiency.

**3. Functional Requirements:**

***a. Brief Description:***

Admin Module: Create employee accounts and view all internal database information.

Doctor Module: View appointments, add patient diagnoses, and access prescriptions and patient lists.

Doctor Assistant Module: Access patient diagnoses and assist in appointment management, including rescheduling and forwarding patient folders to doctors.

Receptionist Module: Manage appointments, forward patient folders to doctors, and create patient profiles.

***b. Acceptance Criteria:***

REQ\_1: The system shall allow the receptionist to add an appointment in the system by filling out the necessary fields.

REQ\_2: The system shall allow the receptionist and the doctor assistant to reschedule an appointment.

REQ\_3: The system should allow the doctor or the doctor assistant to access the folder of the previous prescriptions of each patient.

REQ\_4: The system should allow the doctor to add new prescriptions and notes for the patients.

REQ\_5: The system should allow the doctor to add new prescriptions and notes for the patients.

REQ\_6: The system shall allow the new user to change the password immediately after s/he has accessed the account.

REQ\_7: The system provides the user with to his/her personal information.

REQ\_8: The system shall allow the user to modify the email, phone number and the address fields in the system, if there have been any changes.

REQ\_9: The system should allow the receptionist to create new patient profiles.

REQ\_10: The system confirms the creation of the profile after all the required fields: name, surname, phone number, email, birthday, occupation, medical history (previous medication, surgeries, diagnosis); are filled with information provided by the patient.

REQ\_11: The system shall allow the doctor or the doctor assistant to have viewing access of patients’ profiles.

**4. Non-Functional Requirements:**

***a. Brief Description:***

**Product Requirements:**

UI requirements:

* There will be a navigation bar on the top of the homepage with a login button on the right side. The login button will redirect to a simple login page.
* The login page will be minimalistic with only two fields to be filled: employee ID / patient ID and password. There will also be a submit button as well as a “Forgot password” hyperlink.
* There will also be a “book appointment” button for the guest users. It will redirect the guest user to a page prompting the user to enter the full name and choose an available date and time from the system for the appointment. Mobile number, email address and comments on extra inquiries can also be added optionally.
* The patient dashboard will contain: Profile menu - here he can view and edit personal information about him. Health records menu - here he can view in a chronological way the different examinations he did in the clinic.
* The doctor dashboard will contain the following menus: Profile menu - here he can view and edit personal information about him. Patients menu - here the doctor can view a list of all patients of the clinic and can click a button to access more information about any one of them. Health records menu - here can view a list of health records he has created so far. He can also edit them. o Appointments menu - here he can view all the appointments he has for the day and possibly future and past appointments.

Usability: It is important for the software to be easy to learn, access, and effectively use it. The users will receive a read-me file and a manual explaining how to use every function of the system. In order to improve user experience and ease of use, alert and help messages will appear in the event that incorrect credentials are entered or if there is a problem with the log-in process.

Availability: The system will be available for use 24/7, especially for patients who want access at any time. During business hours, the programme is expected to operate 99.9% of the time on average.

Maintenance: This software product will be maintained by developers to support new features required by customers or to change various types of system functionalities in response to the customer needs.

Performance: Due to the online-based nature of this product, its functionality is strongly dependent on the quality of the internet connection, the number of active users that are accessing the web simultaneously or server hardware performance.

***b. Acceptance Criteria:***

Usability: When performing actions like deletions or sensitive updates, the system will always display a message that will ask the confirmation of the user to proceed with the asked operation.

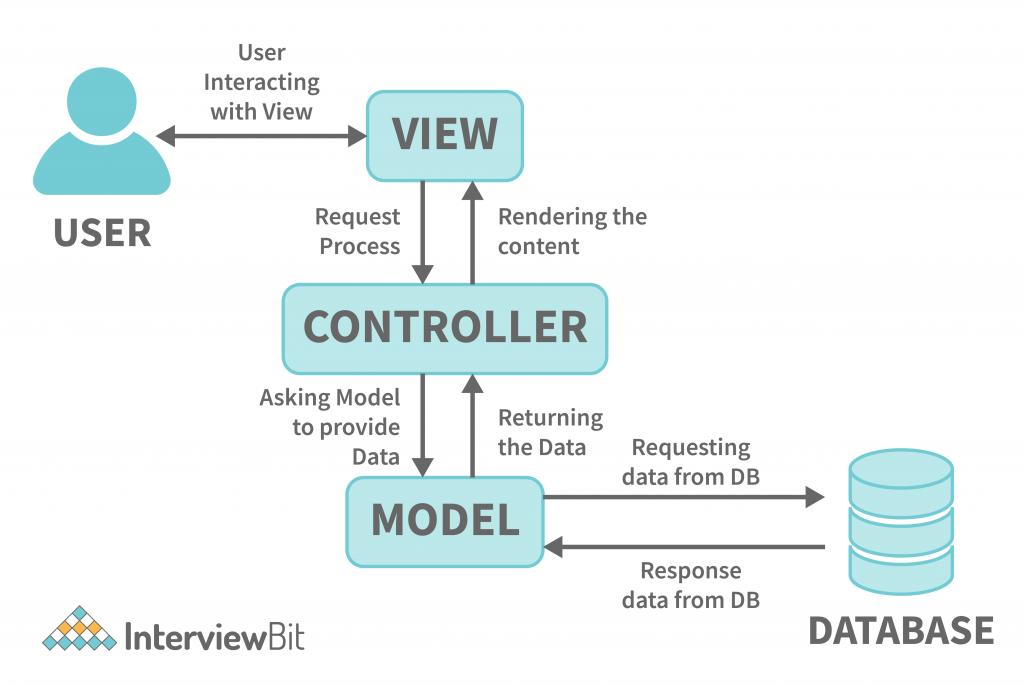
Availability: Since the programme ought to be accessible at all times, Sunday midnight is chosen as the time for updates, modifications, and error corrections. It is expected to be offline for a maximum 1 hour.

Maintenance: If there is a problem with the requested service, an error message will appear if the data is not available or the service is not available at this time. The message may give a solution or may direct the user to the main page.

Performance: All pages and functionalities should load quickly, ensuring a smooth user experience.

**5. Application Specifications:**

a. Architecture:



*MVC Architecture:* MVC is a design pattern commonly used in web development to separate the concerns of an application into three main components: Model, View, and Controller.

* Model: Represents the data and business logic of the application.
* View: Represents the presentation layer of the application, responsible for displaying data to the user.
* Controller: Acts as an intermediary between the Model and the View, handling user input and updating the Model accordingly.

*Client-Server Architecture:* In a client-server architecture, the application is divided into two parts: the client-side and the server-side. The client is typically a web browser or a client application, while the server handles requests, performs computations, and manages data storage.

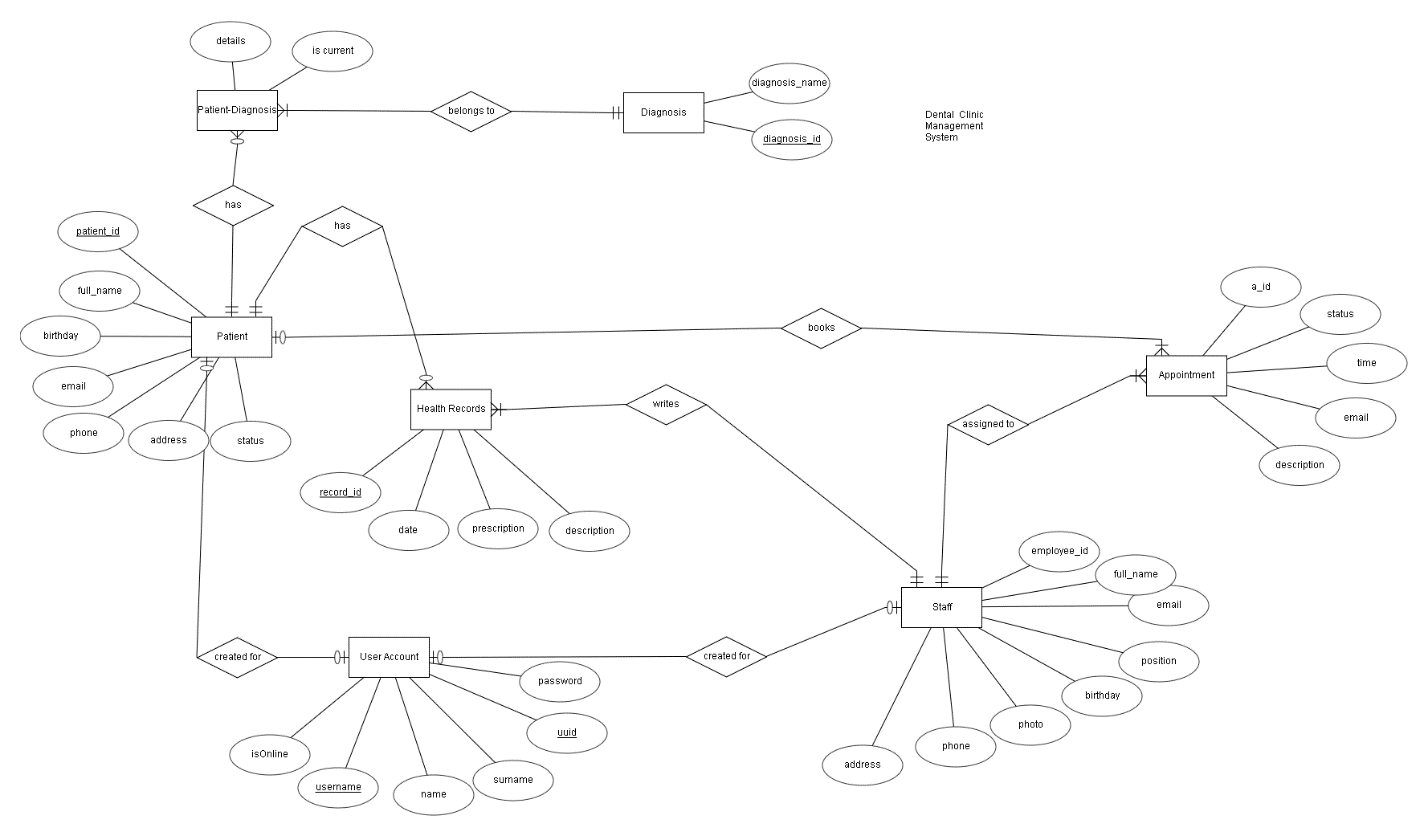
* Client: The client interacts with the user and sends requests to the server for data or actions.
* Server: The server processes requests from clients, executes application logic, retrieves data from databases, and sends responses back to clients.

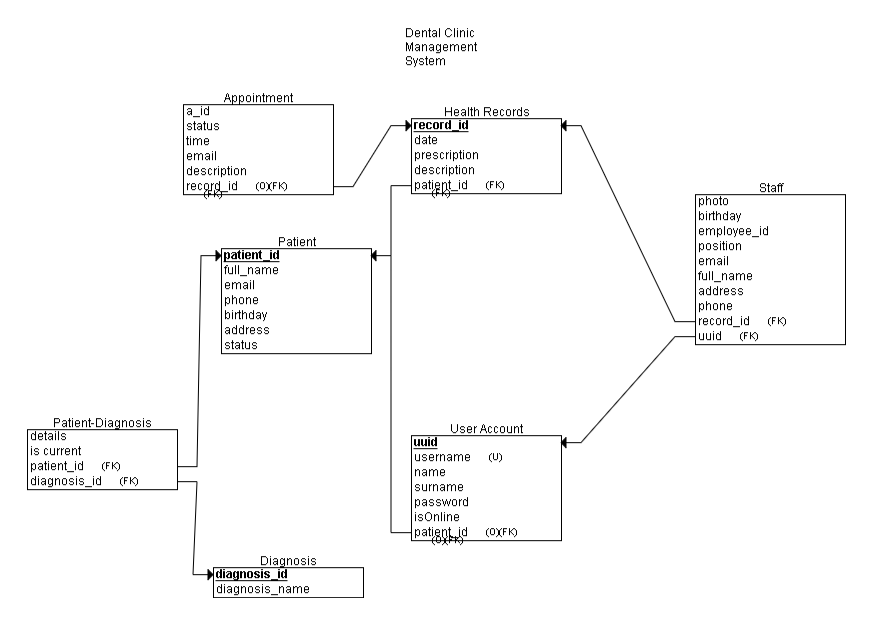
Provide an overview of the chosen architecture:

Include high-level diagrams or descriptions of system components and their interactions.

b. Database Model:

ER Diagram of Dental Clinic Management System





c. Technologies Used:

**Front End**

The programing languages that are going to be used for the client-side web development will be:

• HTML

• CSS

• JavaScript

HTML is a markup language used for structuring and presenting content on the World Wide Web. It is used to create the structure of the front-end.

CSS is a styling language, used for designing the pages, layouts, fonts, colors.

JavaScript is a high level, dynamic, and interpreted programming language that makes the pages more interactive.

Libraries:

1) Bootstrap - the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites.

2) jQuery – A JavaScript library that simplifies the JavaScript usage with respect to front end tasks. I will have utilized the following features from the jQuery library:

* HTML/DOM manipulation
* CSS manipulation
* AJAX (Asynchronous JavaScript and XML) - Make HTTP requests to the server without reloading the page, receive and work with data coming from the server. JSON objects are used to exchange data with the server.

Back End

The programing languages that are going to be used for the server-side web development will be:

• PHP

• MySQL

PHP is a general-purpose scripting language that is particularly well-suited for web development. It is quick, adaptable, and practical. The most widely used PHP development environment (not intended for production) is called XAMPP. The Apache HTTP Server is utilized with XAMPP. Apache is a cross-platform HTTP web server that is used all over the world to distribute web content. The user receives the requested files, pictures, and other documents from Apache's remote server.

d. User Interface Design:

**The doctor dashboard will contain the following menus:**

* Profile menu - here s/he can view and edit personal information about him/her.
* Patients menu - here the doctor can view a list of all patients of the clinic and can click a button to access and edit personal and medical history information about any one of them.
* Create a record menu - here the doctor can create a health record for the patient based on the examination. He can specify the patient, some comments for the record, the diagnosis and the treatment prescribed.
* Appointments menu - here he can view all the appointments he has for the day and possibly future and past appointments.

**The doctor’s assistant dashboard:**

* Profile menu - here s/he can view and edit personal information about her/him.
* Patients list menu - here he can view a list of all patients of the clinic and can click a button to access more information about any one of them.
* Appointments menu - here he can view all the appointments for the day. Assistant can schedule new, reschedule or cancel appointments.
* Doctors’ schedules – has access to agendas to create appointments. Software Requirements the following menus:
* Profile menu - here s/he can view and edit personal information about her/him.
* Patients list menu - here he can view a list of all patients of the clinic and can click a button to access more information about any one of them, expect of their medical history. He can also edit the phone and email information of a user.
* Create a patient profile - here he can create a new patient profile. He will add the required information.
* Appointments menu - here he can view all the appointments for the day. Receptionist can schedule new, reschedule or cancel appointments.
* Doctors’ schedules – has access to agendas to create appointments.

**The admin dashboard:**

* will contain all the menus and functionalities of every other user mentioned above. He can also add, remove or modify other users.

e. Security Measures:

The main protocol used to transfer data between a web browser and a website is HTTPS connections protocol, which is the secure variant of HTTP that my system uses to enable access. To improve data transfer security, HTTPS is encrypted.   
• When storing user credentials in the database, encryption is a must.   
It is also necessary for the password input to become encrypted during validation. Storing user credentials in the database, encryption is a must.

It is also necessary that the password input be encrypted during authentication.

The PHP function password\_hash() is used to encrypt passwords and because it uses a powerful hashing algorithm it prevents attacks.

Authentication:

* Users must provide a username and password to authenticate themselves.
* Sessions are managed using PHP sessions stored on the server, preventing sensitive information from being stored on the user's computer.
* Each user is assigned a unique session identifier (SID) to link them with their data on the server. Sessions are destroyed upon logout to prevent unauthorized access.