Software Requirements Specification

for

Peggasus Booking

Version 1.1.1 approved

Prepared by Santiago Del Rio Obando, Antonio Ortega Guererro, Paulo Da Costa, Morgan Maps, Jason Salazar, Pierce Nance

Cal State LA, Professor Marianne Abalos

12/01/2023

**Table of Contents**

Table of Contents................................................................................................................. pg #2

Revision History................................................................................................................... pg #3

1. Introduction................................................................................................................ pg #4

1.1. Purpose........................................................................................................... pg #4

1.2. Intended Audience and Reading Suggestions................................................ pg #4

1.3. Product Scope................................................................................................ pg #4

1.4. Definitions, Acronyms, and Abbreviations .................................................. pg #4

1.5. References...................................................................................................... pg #5

2. Overall Description.................................................................................................... pg #6

2.1. System Analysis…......................................................................................... pg #6

2.2. Product Perspective........................................................................................... pg #7

2.3. Product Functions........................................................................................... pg #9

2.4. User Classes and Characteristics.................................................................... pg #12

2.5. Operating Environment.................................................................................. pg #12

2.6. Design and Implementation Constraints........................................................ pg #13

2.7. User Documentation...................................................................................... pg #13

2.8. Assumptions and Dependencies.................................................................... pg #13

2.9. Apportioning of Requirements...................................................................... pg #13

3. External Interface Requirements............................................................................... pg #14

3.1. User Interfaces............................................................................................... pg #14

3.2. Hardware Interfaces....................................................................................... pg #16

3.3. Software Interfaces........................................................................................ pg #16

3.4. Communications Interfaces........................................................................... pg #16

4. Requirements Specification....................................................................................... pg #17

4.1. Functional Requirements............................................................................... pg #17

4.2. External Interface Requirements................................................................... pg #20

4.3. Logical Database Requirements.................................................................... pg #22

4.4. Design Constraints......................................................................................... pg #24

5. Other Nonfunctional Requirements........................................................................... pg #25

5.1. Performance Requirements............................................................................ pg #25

5.2. Safety Requirements...................................................................................... pg #25

5.3. Security Requirements................................................................................... pg #25

5.4. Software Quality Attributes........................................................................... pg #25

5.5. Business Rules............................................................................................... pg #26

6. Legal and Ethical Considerations.….......................................................................... pg #27

Appendix A: Glossary........................................................................................................ pg #28

Appendix B: Analysis Models........................................................................................... pg #29

Appendix C: To Be Determined List................................................................................. pg #35

Revision History

| Name | Date | Reason For Changes | Version |
| --- | --- | --- | --- |
| Santiago Del Rio Obando, Antonio Ortega Guerrero,  Paulo Da Costa,  Jason Salazar, Morgan Maps, Pierce Nance | 10/03/23 | Created The Document | 0.0.0 |
| Paulo Da Costa  Jason Salazar  Antonio Ortega | 10/11/23 | Updated Section 4 | 0.0.1 |
| Santiago Del Rio | 10/11/23 | Updated Section 1.5 References | 0.0.1 |
| Paulo Da Costa | 11/9/23 | Updated Section 3.4 - Communication Interfaces  Updated Section 4.1 - Functional Requirements | 0.1.1 |
| Paulo Da Costa Antonio Ortega | 11/30/23 | Final Review | 1.1.1 |

**1. Introduction**

**1.1 Purpose**

The purpose of this document is to outline the requirements for the development of Peggasus Booking. This system aims to streamline the process of scheduling and managing appointments for patients, doctors, and administrative staff. This document will cover everything for the release of version 1.0.0 of the website. It will cover every aspect of the product, from external software interfaces to ethical and legal limitations and restraints.

**1.2 Intended Audience and Reading Suggestions**

Development Team: This document is primarily intended for the development team responsible for creating the Peggasus. It provides a comprehensive overview of the system's requirements, functionalities, and technical specifications.

Project Stakeholders: This includes project managers, business analysts, and other team members involved in the planning and execution of the project. It helps them understand the scope, objectives, and technical aspects of the HAB.

Client Representatives: This document serves as a reference for them to ensure that the project aligns with their expectations and requirements.

**1.3 Product Scope**

Peggasus Booking will facilitate the appointment booking process for both hospitals and patients. It will have features for patients to book, reschedule, and cancel appointments online. Additionally, it will provide a platform for doctors to manage their schedules and view patient information. Administrative staff will have access to monitor and manage appointments, users, and system settings.

**1.4 Definitions, Acronyms, and Abbreviations**

Peggasus Booking: The name of the hospital appointment booking website we are developing

HIPAA: Health Insurance Portability and Accountability Act

EMR: Electronic Medical Record

PHI: Protected Health Information

**1.5 References**

[Product Proposal](https://github.com/csula-cs3337swe/202308Group1-repo/issues/2): https://github.com/csula-cs3337swe/202308Group1-repo/issues/2

[Software Test plan](https://docs.google.com/document/d/1Wnx_Xn9mB8zHTuikZDxcomuHbCti67kefLf2_F_d4Mo/edit?usp=sharing): https://docs.google.com/document/d/1Wnx\_Xn9mB8zHTuikZDxcomuHbCti67kefLf2\_F\_d4Mo/edit?usp=sharing

[HIPAA for Healthcare Privacy Regulations](https://www.hhs.gov/hipaa/index.html): https://www.hhs.gov/hipaa/index.html

[Web Content Accessibility Guidelines (WCAG) International Standard 2.0, 2.1, 2.2](https://www.w3.org/TR/WCAG21/): https://www.w3.org/TR/WCAG21/

[ICT Accessibility 508 Standards and 255 Guidelines](https://www.access-board.gov/ict/): https://www.access-board.gov/ict/

Software Engineering 10th Edition By Ian Sommerville Textbook

[Asp.Net Core MVC Documentation: https://learn.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-8.0](https://learn.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-8.0)

**2. Overall Description**

**2.1 System Analysis**

The primary goals of Peggasus Booking project are as follows:

* Efficient Appointment Management: Provide a user-friendly platform for patients, doctors, and administrative staff to efficiently schedule, manage, and monitor appointments.
* Improved User Experience: Enhance the overall experience of patients by allowing them to easily book, reschedule, and cancel appointments online, reducing the need for manual interventions.
* Streamlined Doctor Scheduling: Enable doctors to have better control over their schedules, allowing them to manage appointments and view patient information conveniently.
* Enhanced Administrative Oversight: Empower administrative staff with the tools to efficiently manage user accounts, appointments, and system settings.
* Integration with Existing Systems: Seamlessly integrate with the hospital's existing Electronic Medical Record (EMR) system for a cohesive patient management experience.

The Technical Hurdles we probably will encounter are as follows:

* Integration with existing EMR System
  + Challenge: Establishing a seamless integration with the existing EMR system to fetch and update patient records securely.
  + Potential Issues: Compatibility issues, data security concerns, and ensuring real-time synchronization.
* Security and Compliance
  + Challenge: Ensuring robust security measures to protect sensitive patient data and compliance with healthcare data privacy regulations (HIPAA).
  + Potential Issues: Data breaches, unauthorized access, compliance violations, and secure transmission of data.
* Scalability and Performance
  + Challenge: Designing the system to handle a potentially large number of concurrent users without compromising performance.
  + Potential Issues: Database optimization, server load balancing, and ensuring responsive UI/UX.
* User Authentication and Authorization:
  + Challenge: Implementing a secure authentication mechanism and role-based access control to ensure that users only have access to appropriate features.
  + Potential Issues: Vulnerabilities in authentication process, unauthorized access, and data exposure.
* Cross-Browser Compatibility and Responsiveness:
  + Challenge: Ensuring that the website functions consistently across various web browsers and devices, including desktops, tablets, and mobile phones.
  + Potential Issues: CSS compatibility, JavaScript compatibility, and responsive design challenges.

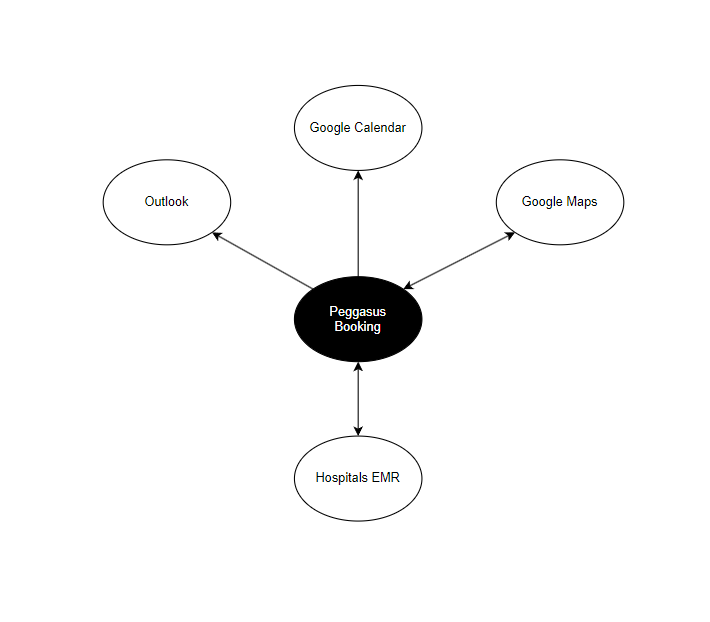
The Potential Solutions:

* Integration with EMR System:
  + Solution: Collaborate closely with EMR system providers for a robust API integration. Implement secure data transmission protocols and conduct thorough testing to ensure data accuracy.
* Security and Compliance:
  + Solution: Employ strong encryption methods for data at rest and in transit. Implement access controls, conduct regular security audits, and train staff on compliance requirements.
* Scalability and Performance:
  + Solution: Utilize cloud-based services to dynamically scale resources. Optimize database queries and employ caching mechanisms. Conduct load testing to identify and address performance bottlenecks.
* User Authentication and Authorization:
  + Solution: Implement industry-standard authentication protocols. Utilize role-based access control to manage user privileges.
* Cross-Browser Compatibility and Responsiveness:
  + Solution: Use responsive design principles and CSS frameworks like Bootstrap. Conduct rigorous testing on different browsers and device to ensure best user experience

**2.2 Product Perspective**

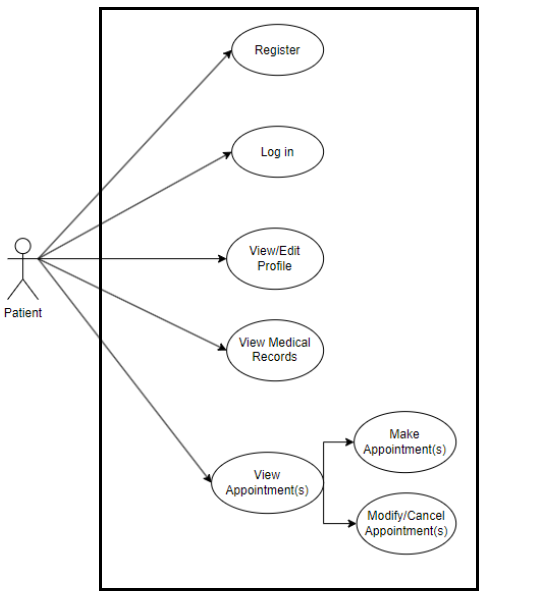
Peggasus Booking is a system that integrates a calendar, maps, email, and the hospital’s EMR in order to become a central hub that will allow patients to book and manage their appointments(i.e cancel or change), and view their medical records. Doctors can manage their schedules, and view and update their patients EMR. Our stretch goal would be to make it the centralized patient portal where they would be able to view lab results and order prescriptions as well.

This product will be similar to other products such as the KAISER PERMANENTE app, but unlike those apps we may not implement a way for patients to be able to order and pay for their prescriptions through our site given the time constraints. In the next page there is a diagram of how our product will fit in the overall hospital system.



**2.3 Product Functions**

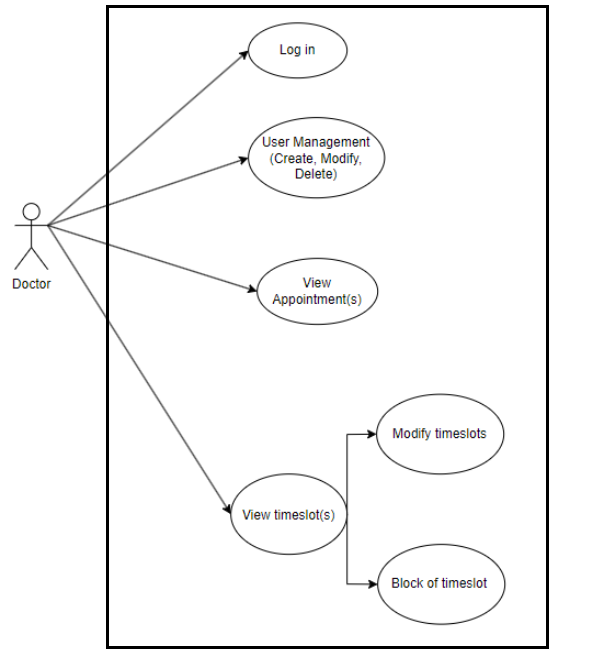
Patient Module Functions Diagram:



Patient Module Functions:

| Register | User can register as a new account, select preferred Healthcare Provider |
| --- | --- |
| Log In | User can log in to website |
| View/Edit Profile | User can view and edit profile |
| View Medical Records | User will be able to view their own Medical Records |
| View Appointment(s) | User will be able to view their appointment(s) |
| Modify/Cancel Appointment(s) | User will be able to modify or cancel their appointment(s) |

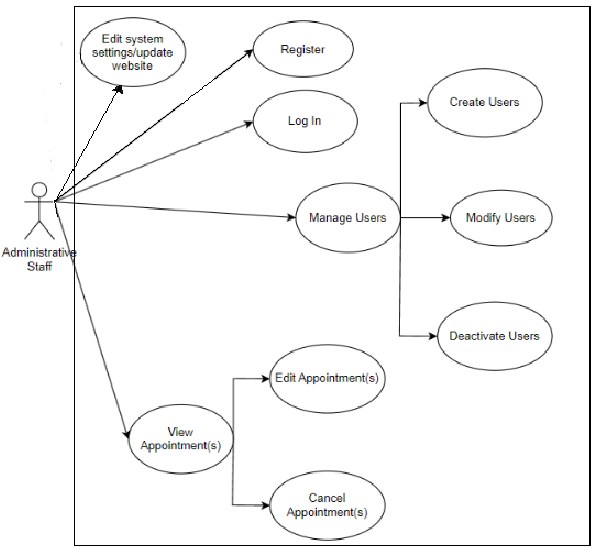
Doctor Module Functions Diagram:



Doctor Module Functions:

| Log In | User can log in to website |
| --- | --- |
| View/Update Patient Medical Records | User will be able to view and make updates to their Patients EMR |
| View Appointment(s) | User will be able to view their appointment(s) |
| View Time Slot(s) | User will be able to view their timeslots |
| Modify Time Slot(s) | User will be able to modify their existing timeslot(s) |
| Block Time Slot(s) | User will be able to block of timeslot(s) from their schedule |

Administrative Module Diagram:

****

Administrative Staff Module Functions:

| Register | User can register to the website |
| --- | --- |
| Log In | User can log in to website |
| Create User | User will be able to create other users(Patients, Doctors, Administrative staff) |
| Modify Users | User will be able to modify other users(Patients, Doctors, Administrative staff) |
| Deactivate Users | User will be able to deactivate other users(Patients, Doctors, Administrative staff) |
| View Appointments | User will be able to view appointments |
| Edit Appointments | User will be able to modify existing appointments |
| Cancel Appointments | User will be able to cancel appointments |
| Edit System Settings | User will have administrative access in order to configure system settings to better meet needs of the hospital |

**2.4 User Classes and Characteristics**

Once released Peggasus Booking will serve as a crucial tool within the hospital digital ecosystem. It will be utilized by patients, doctors, and administrative staff to streamline the appointment booking/management process. Below is an overview of how each user will utilize the product ranked from most important user to least important.

1. Administrative Staff
   * User Management
   * Oversee Appointments
   * Configure System Settings
2. Doctors
   * Secure Login
   * Manage Schedules
   * Access Patient Information/ Medical History
3. Patients
   * Register and Log In: Patients will create accounts and securely log in to access the system.
   * Book Appointments: Patients can view available appointment slots and book, reschedule, or cancel appointments at their convenience.
   * Access Medical History: Patients can view and update their medical history, including allergies, medications, and previous surgeries.

**2.5 Operating Environment**

* Hardware Platforms:
  + Server: A dedicated web server to host the application and manage database operations.
  + Client Devices: Desktops, laptops, tablets, and mobile phones with internet access for users to access the website.
* Operating System(Server): Windows Server for hosting the web application
* Other Important Components
  + Internet Connectivity: A stable and reliable internet connection for both the server hosting the application and the client devices accessing the website.
  + Firewall and Security Measures: Network security protocols to protect against unauthorized access and potential threats.
  + Compatible Browser: The web application should be compatible with commonly used web browsers like Chrome, Firefox, Safari, and Edge. Though if a user uses an uncommon browser it may impact the experience.

**2.6 Design and Implementation Constraints**

* Regulatory Compliance:
  + The system must comply with healthcare data privacy regulations such as HIPAA in the USA which dictate how patient data is handled, stored, and transmitted.
  + The system would have to comply with internal policies and procedures set by the hospital, potentially influencing design choices and development workflows.
* Integration with Existing Systems:
  + The H.A.B must be compatible with the hospitals’ existing systems, particularly the Electronic Medical Record (EMR) system, which may require specific protocols or APIs.
* Security and Data Protection:
  + Strict security measures must be implemented to protect patient data, potentially imposing constraints on system architecture and data handling practices.

**2.7 User Documentation**

* Directions document guiding patients how they can book appointments themselves through the website.
* The app can have an initial tutorial on how to navigate and use the app with tutorial overlays and tips. All of which can also be revisited in settings.
* Tutorial videos can also be made for the purpose of live demonstration.

**2.8 Assumptions and Dependencies**

* Software design will be ensured to be operable on all operating systems and be compatible with any computer
* Mobile app may also be developed for those wanting to access their information on the phone

**2.9 Apportioning of Requirements**

* Mobile-Website friendly
  + Mobile website UI: We may choose to make a whole new UI in order to make the mobile device experience seamless.
  + Mobile devices for testing: Emulation testing is an option but a proper test can only be conducted on actual devices.
  + Optimize graphics for multiple resolutions: Optimizing graphics can make it so when any graphic is viewed at multiple resolutions, the experience isn’t compromised.
  + Responsive web design: Making a responsive website makes it so the website responds to the users environment and accommodates them making the experience across devices the same.

**3. External Interface Requirements**

The Peggasus Booking App will be accessed with the most common web browsers on typical operating systems. It will display a calendar for viewing/setting availability, booking & canceling appointments. A user dashboard will have options to view & edit information about the user & their schedule.

**3.1 User Interfaces**

* Software components:

o Visual Interactable Calendar

§ Clinic users will be able to set availability, and can add or cancel appointments

§ Patient users will be able to view provider availability, and add or cancel appointments

o User dashboard containing buttons/functions to

§ interact with personal information

§ account settings

§ Help menu

§ Notifications

§ Upcoming appointments

* ADA Compliance:

o Keyboard navigation utilizing standardized keyboard shortcuts

o All information will be detected by screen readers with clear instructions for user input/forms

o Alt text on images, if applicable

o Clear color contrast (at least 4.5:1, 7:1 for text ) & information not reliant on colors or shapes

o Use microdata to markup UI components

o Simple text resizing

o 2D scrolling capability

**3.2 Hardware Interfaces**

The app will be interfaced with PCs and mobile devices:

* Runs the current version of the web browsers specified

o Windows 10/11

o Mac

o Android

o iOS

* Inputs are received through typical operating system navigation (touchscreen, mouse, keyboard).

**3.3 Software Interfaces**

* Database Management System:
  + The Website will use a Database system to store and retrieve patient information, appointments, and system settings
* Web Server:
  + Internet Information Services(10.0) by Microsoft will be used to serve web pages and handle HTTP requests from users accessing the site.
* WindowsServer 2022:
  + Will be the operating environment for hosting and running the web application
* RapidJSON:
  + JSON Parser & Generator for C#. Will be used for data interchange between JS & C#
* Bootstrap 5: CSS framework to build front-end responsiveness

**3.4 Communications Interfaces**

* Supported Browsers (Tentative):

o Chrome 117.0 +

o Firefox 117.0+

o Edge 117.0+

o Opera 102.0

o Safari 16.5

* Network Protocol: HTTPS

o Secure transfer protocol for data security (encryption)

* HTML forms

o Data will be sent to/from web server

o Will use CSRF tokens if feasible

* Data transfer rates: clinic & patient users can load 50mb max file size to accommodate for record files

* Synchronization: data & files will be stored in Windows Server

* Email Communication

o Emailing system to manage reminder notifications & confirmations

* Calendar API:
  + accessed through HTTP calls
  + Imports event data

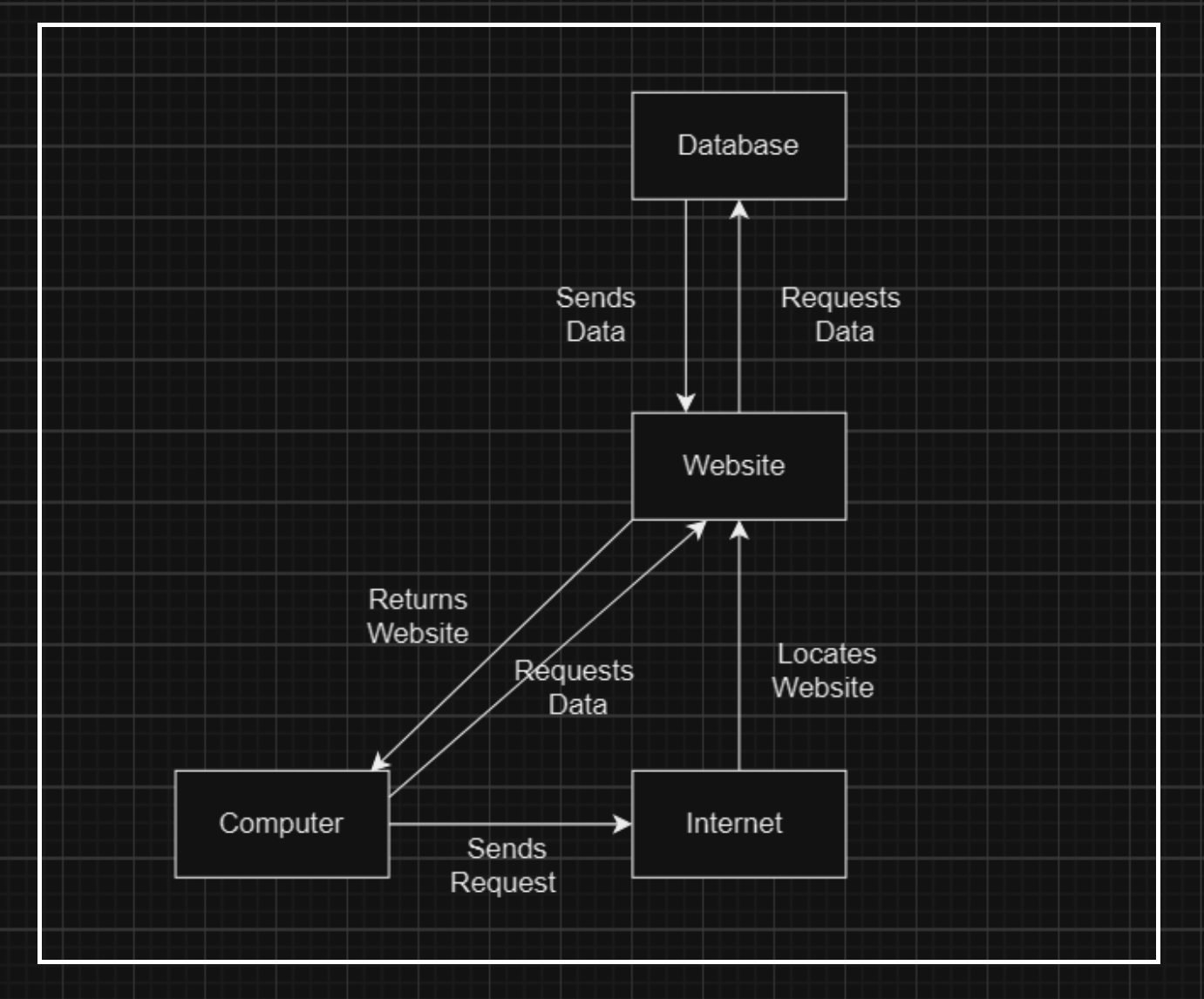
**4. Requirements Specification**

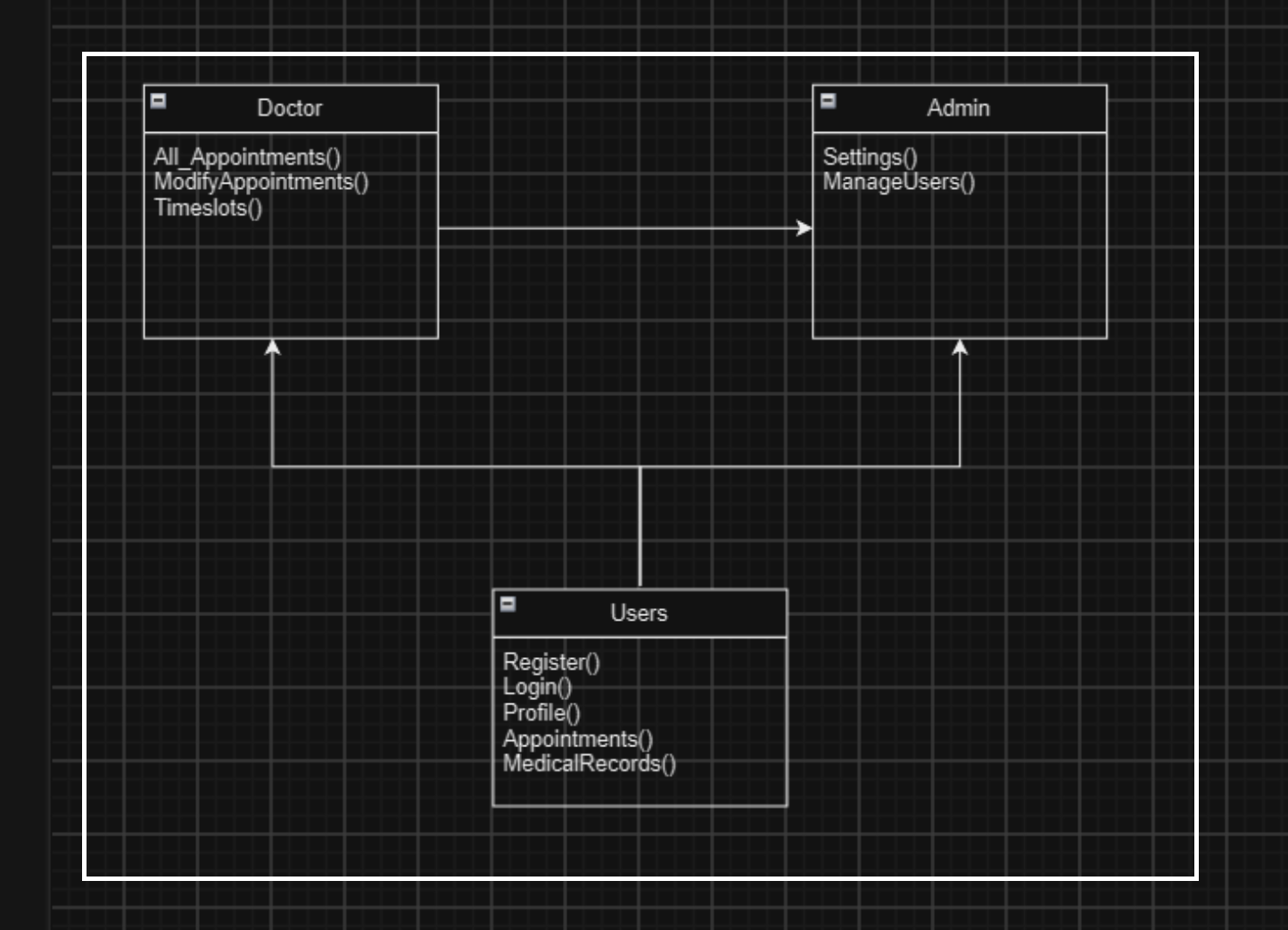
This section provides detailed information about the requirements specification for the Peggasus Online Appointment Booking project.

**4.1 Functional Requirements**

The Peggasus Online Appointment Booking project will include the following requirements:

* User Registration and Login:
  + The system shall allow patients to register for a new account, providing their full name, email address, and a password.
  + The system shall validate the email format during the registration process to ensure it’s a valid email address.
  + The system shall provide a confirmation email to the user for account activation upon successful registration.
  + The system shall allow registered users to log in using their email and password.
  + The system shall maintain a secure login session and log out users automatically after a period of inactivity.
* Profile Management:
  + The system shall allow users to view and edit their profile information, including personal details and contact information.
  + The system shall restrict access to profile editing to the respective user only.
* Medical Records:
  + The system shall enable doctors only to view the patient’s medical records, including allergies, medications, treatments and previous surgeries.
  + The system shall maintain patient records in a secure and confident manner, accessible only by authorized personnel.
* Appointment Management:
  + The system shall allow patients to view available appointment slots based on doctor’s schedules and specialties.
  + The system shall provide a calendar interface for patients to select their desired appointment date and time.
  + The system shall validate appointment slots to prevent double booking or overbooking.
  + The system shall allow patients to request appointment cancellations or rescheduling, and it should update the appointment status accordingly.
  + The system shall allow doctors to view their upcoming appointments.
  + The system shall provide doctors with a user-friendly interface to add new appointment slots to their schedule.
  + The system shall handle appointment conflicts by notifying users and offering alternative time slots.
  + The system shall send email notifications to patients and doctors regarding upcoming appointments, cancellations, or rescheduled appointments.
* Error Handling and Recovery:
  + The system shall display clear error messages for validation errors during registration, login, or appointment scheduling.
  + The system shall log and store error details for debugging and analysis.
  + The system shall recover efficiently from system failures to ensure data consistency and availability.
* Parameter Effects:
  + The system shall sanitize and validate all input parameters to prevent security vulnerabilities.
* Input/Output Sequences:
  + The system shall provide a straightforward and logical sequence for appointment booking, modification, and cancellation.
  + The system shall send confirmation email to the user after an appointment has been successfully booked.
* Formulas for Input to Output Conversion:
  + The system shall calculate and display appointment availability based on doctor’s schedules.
  + The system shall convert patient requests into actual appointments based on available time slots.
  + The system shall calculate and display bill amounts for specific appointments based on predefined rates.

****

****

**4.2 External Interface Requirements**

The Peggasus Online Appointment Booking project will contain the following formats for external interfaces:

* User Interfaces:
  + Name of the Item: Registration Page
  + Description of Purpose: Allows new users (patients, doctors, and administrative staff) to create accounts and provide essential personal information.
  + Source of Input: Users (new registrants)
  + Accuracy/Tolerance: The system shall validate the accuracy of the provided information.
  + Timing: Available during user registration.
  + Relationships: It connects to the user’s database and registration process.
  + Screen-Format/Organization: User-friendly web-based registration form.
  + Data Format: Form fields for name, email, password, and additional details.
  + Name of the Item: Appointment Booking Interface
  + Description of Purpose: Allows patients to view available time slots, select an appointment date and time, and request appointments.
  + Source of Input: Patients, doctor schedules.
  + Valid Range: Time slots based on doctor schedules.
  + Accuracy/Tolerance: The system shall validate appointment availability and prevent conflicts.
  + Units of Measure: Date and time.
  + Timing: Available when patients request appointments.
  + Relationships: Links with the doctor’s schedules and patient database.
  + Screen Format/Organization: Calendar-based interface.
  + Data Format: Displays date and time slots for selection.
  + Name of Item: Profile Management
  + Description of Purpose: Allows users to view and edit their profile information.
  + Source of Input: Registered users.
  + Accuracy/Tolerance: The system shall ensure users only access and modify their own profiles.
  + Timing: Accessible to users after login.
  + Relationships: Connects to the users database.
  + Screen Format/Organization: User-specific profile page.
  + Data Format: Displays and edits user data such as name, and contact details.
* Hardware Interfaces:
  + Name of Item: Server
  + Description of Purpose: Hosts the online appointment booking system, manages the application logic, and communicates the database.
  + Source of Input: Software application, user requests and database.
  + Accuracy/Tolerance: Must be operational and responsive to user requests.
  + Timing: Always active.
  + Relationships: Links to the database server for data retrieval and storage.
* Software Interfaces:
  + Name of Item: Database Management System
  + Description of Purpose: Stores and retrieves user data, appointment details, and system settings.
  + Source of Input: Application database queries.
  + Accuracy/Tolerance: The system should provide efficient data storage and retrieval.
  + Timing: Responds to queries and updates as needed in less than one second.
  + Relationships: Connected to the server and user actions.
  + Data Format: Utilizes database queries for data interaction.
* Communications Interfaces:
  + Name of Item: Email System
  + Description of Purpose: Sends notification emails to users for appointment confirmations, reminders, and other communication.
  + Source of Input: Automated system notifications via user actions.
  + Accuracy/Tolerance: Emails should be delivered to the correct recipients.
  + Timing: Emails triggered by system events or user input.
  + Relationships: Integrates with the system to send email notifications.
  + Data Format: Email content and format.
  + Name of Item: Calendar API
  + Description of Purpose: Allows synchronization of appointments details with Calendar.
  + Source of Input: Application event data.
  + Accuracy/Tolerance: Synchronizes appointments accurately with Calendar
  + Timing: Synchronizes appointments with Calendar real-time upon creation or modification.
  + Relationships: Links to the application for appointment data transfer.
  + Data Format: Calendar-compatible event data.
  + Name of Item: Internet Connectivity:
  + Description of Purpose: Ensures the system’s ability to communicate with the user devices and external services via the internet.
  + Source of Input: Network connection.
  + Accuracy/Tolerance: Requires a stable internet connection for system operation.
  + Timing: Constant as it is required for all user interactions
  + Relationships: Core for all online functionality.

**4.3 Logical Database Requirements**

This section specifies the logical requirements for any information that is to be placed into a database.

**1. Database**

**1.1 Patients**

The system shall require inputs for:

* firstName
* lastName
* gender
* ssn
* insurance
* height
* weight
* bloodType
* dateOfBirth

The system shall assign each patient:

* patientID

**1.2 Doctors**

The system shall require inputs for:

* firstName
* lastName
* specialty
* gender
* phone
* email

The system shall assign each doctor:

* doctorID

**1.3 Appointments**

The system shall require inputs for:

* startTime
* endTime
* appStatusID

The system shall assign each appointment:

* appID

**1.4 Hospitals**

The system shall require inputs for:

* name
* phone
* address
* state
* zipCode

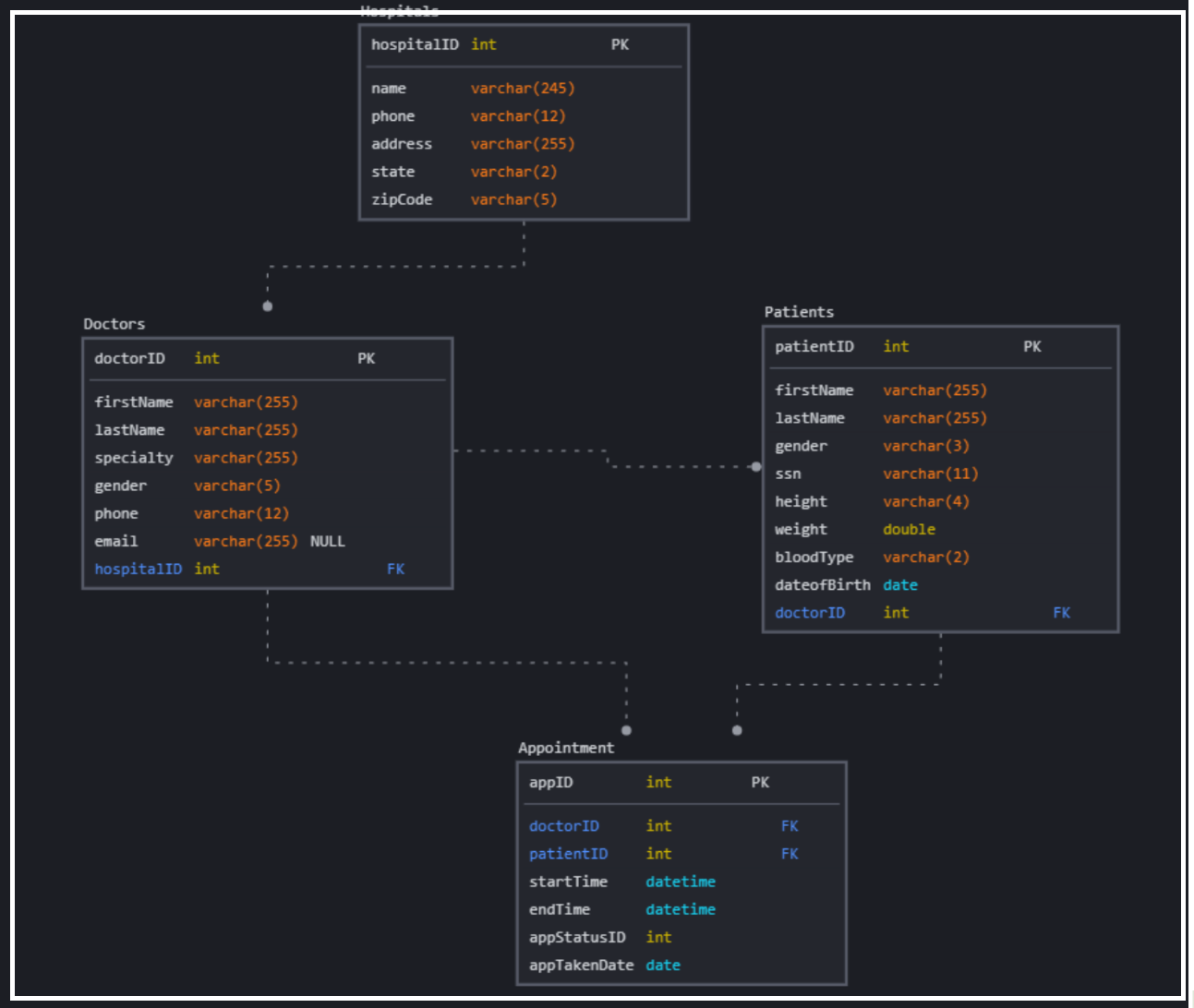
The system shall assign the hospital:

* hospitalID

For now we are working with just one hospital but if we were to introduce different hospitals to the system in the future then each would carry their own hospital ID.

**2. Data Retention**

**5.1** Data should be kept indefinitely until patient or administration deletes account information from the database



**4.4 Design Constraints**

This section outlines the design constraints that can be imposed by other standards and hardware limitations.

1.1. The system should consider the user device screen and size. We want the website to adapt to the size of the screen of the user.

* + Computer Monitor
  + Mobile Device
  + Tablet

1.2 The system should consider internet connection. Many people don’t have the best internet connection so we would want to optimize the website for fast loading and consider offering lower-resolution versions of images and videos

1.3 The system shall consider accessibility. We want to accommodate all users and take into consideration any users with disabilities.

1.4 They system may have a print styleSheet. Some users may want to print out information so we should consider a legible and readable printout.(Out of scope)

**5. Other Nonfunctional Requirements**

**5.1 Performance Requirements**

Once released Peggasus Booking will handle the following:

* Support 100-200 concurrent working terminals
* Support 2,500-3,000 users logged in at the same in between patients, doctors, office assistants, receptionists, system administrators and vendors from various locations
* The platform will store the following information:
  + Patient records, including personal information and medical history
  + doctor profiles, including specialties and availability schedules
  + Appointment details, such as date, time and location
  + Able to handle 3,000-5,000 transaction records, including billing and payment processing 97% of the time in less than a second
  + The system will handle current data processing and allow for future scalability as the amount of data grows

**5.2 Safety Requirements**

* Regular backups
  + We want to prevent any loss of personal data so we will devise to regularly have backups to store the most up to date information
* Account Recovery
  + Software will have a method of recovering your account in cases such as password is forgotten so users don’t lose their account

**5.3 Security Requirements**

* User login encryption
  + Secure login to prevent anyone from gaining access to your account
* Personal information security
  + Patients should only have access to their own personal information
  + Only people with proper authorization should be able to access database information

**5.4 Software Quality Attributes**

* Reliability
  + Software should be able to create and store appointments for all patients with no issues
* Ease of use
  + A simple software that doesn't overwhelm users
* Secure
  + Software will have preventative measures in order to protect user personal information

**5.5 Business Rules**

* Administrators are able to update settings on the website and have control over patient accounts.
* Patients should mainly take control of appointments with the liberty to create, change, or cancel appointments
* Administration will also have access to these functions in order to personally help patients having trouble navigating the appointment system

**6. Legal and Ethical Considerations**

The Main legal consideration for our project will be to make sure it complies with data privacy regulations set by HIPAA. There are four four specific HIPAA storage requirements that covered entities and business associates must meet. These include:

* Ensuring the confidentiality, integrity, and availability of all e-PHI through encryption, password protection, and other protection measures.
* Identifying and protecting against reasonably anticipated threats through regular monitoring and risk analysis.
* Protecting against reasonably anticipated impermissible uses or disclosures with safeguards such as IT security protocols, IAM, restricting physical access, and regular audits of internal processes.
* Ensuring compliance by the workforce through regular training and adherence to rules set by HIPAA enforcement officers.

**Appendix A: Glossary**

For this report, we used three primary different abbreviations. They are as follows:

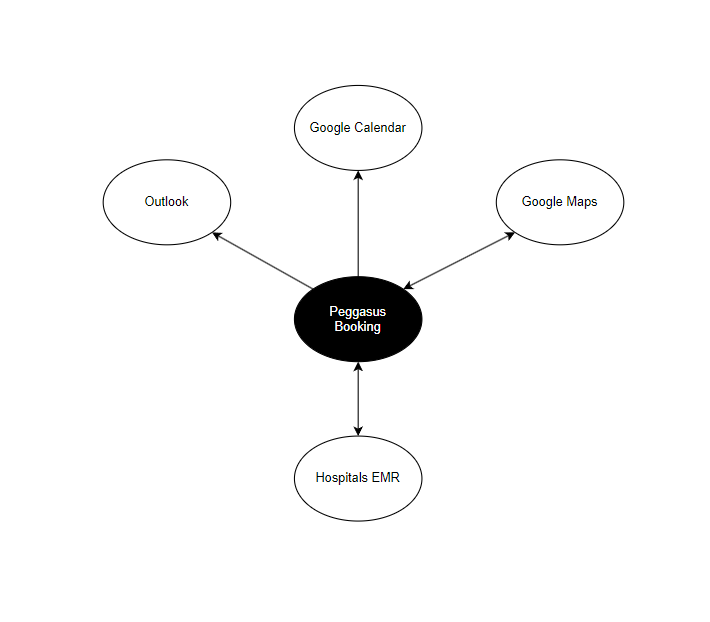
HIPAA: Health Insurance Portability and Accountability Act, which is a federal law that requires the creation of national standards to protect sensitive health information from being disclosed.

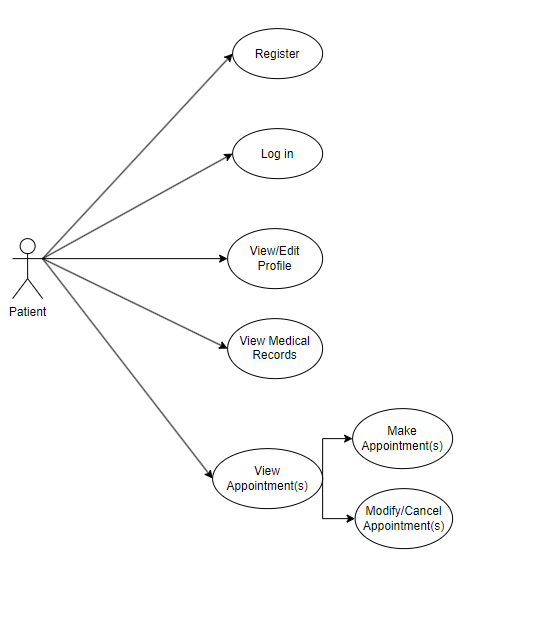
EMR: Electronic Medical Record. The EMR is a digital version of the paper charts in the clinician’s office. An EMR contains the medical and treatment history of the patients in one practice.

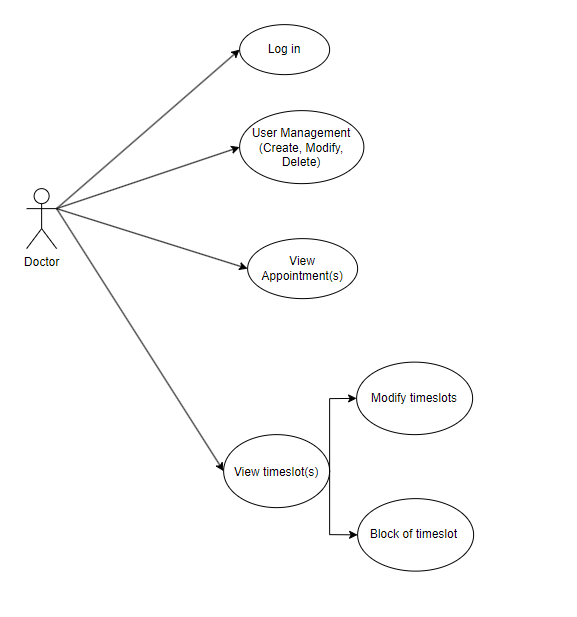
PHI: Protected Health Information. The PHI is any information in the medical record or designated record set that can be used to identify an individual that was created, used, or disclosed in the course of providing a health care service such as diagnosis or treatment.

**Appendix B: Analysis Models**

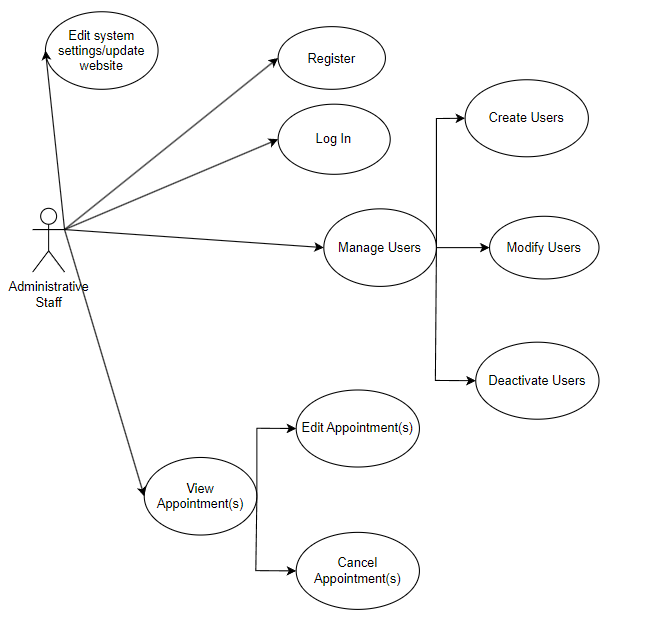
2.2 Product Perspective integrates calendar, maps, email, and the hospital’s EMR in order to become a central hub, as follows:



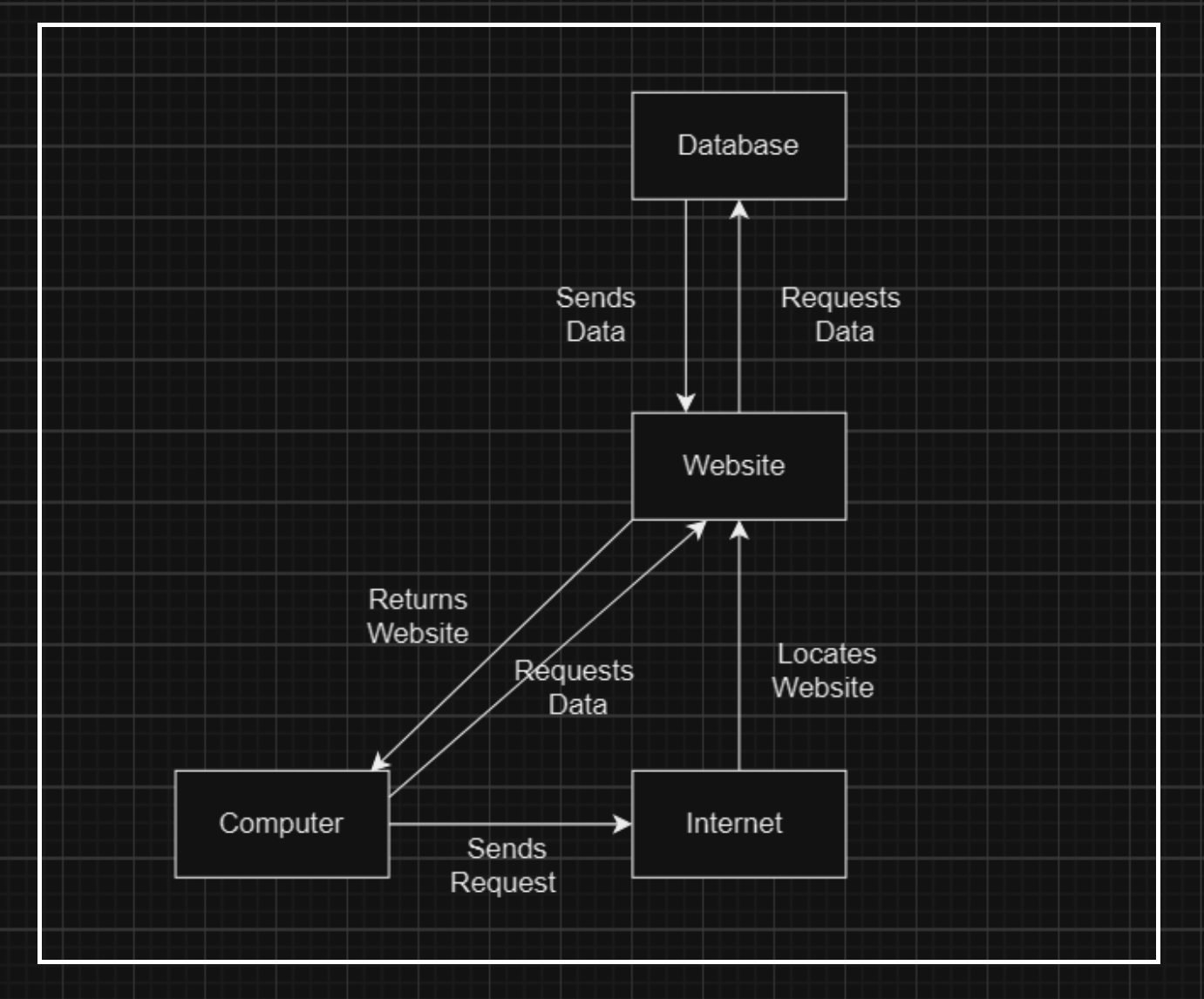
Patient Module Functions Diagram:

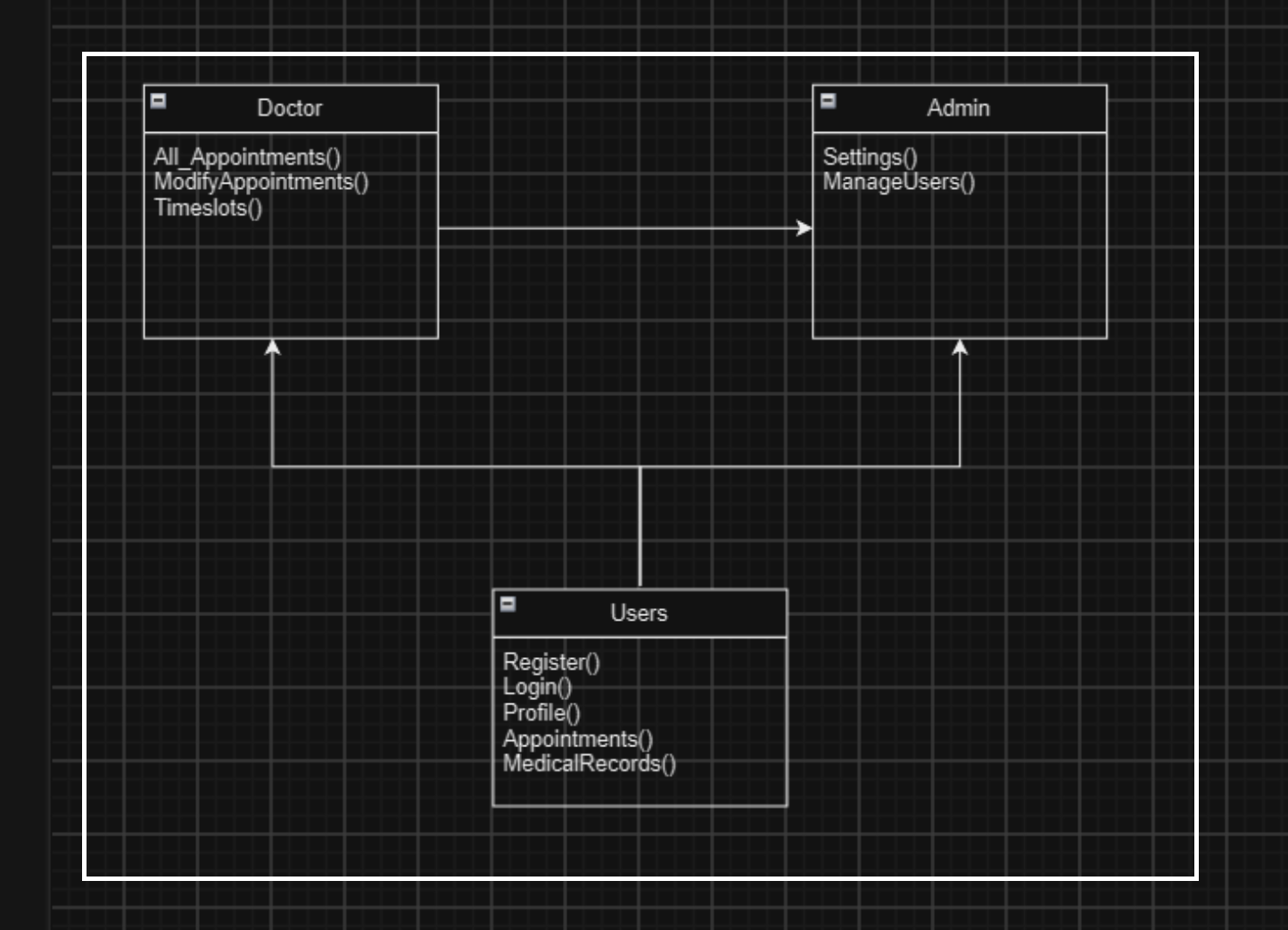
Doctor Module Functions Diagram:

Administrative Module Diagram:

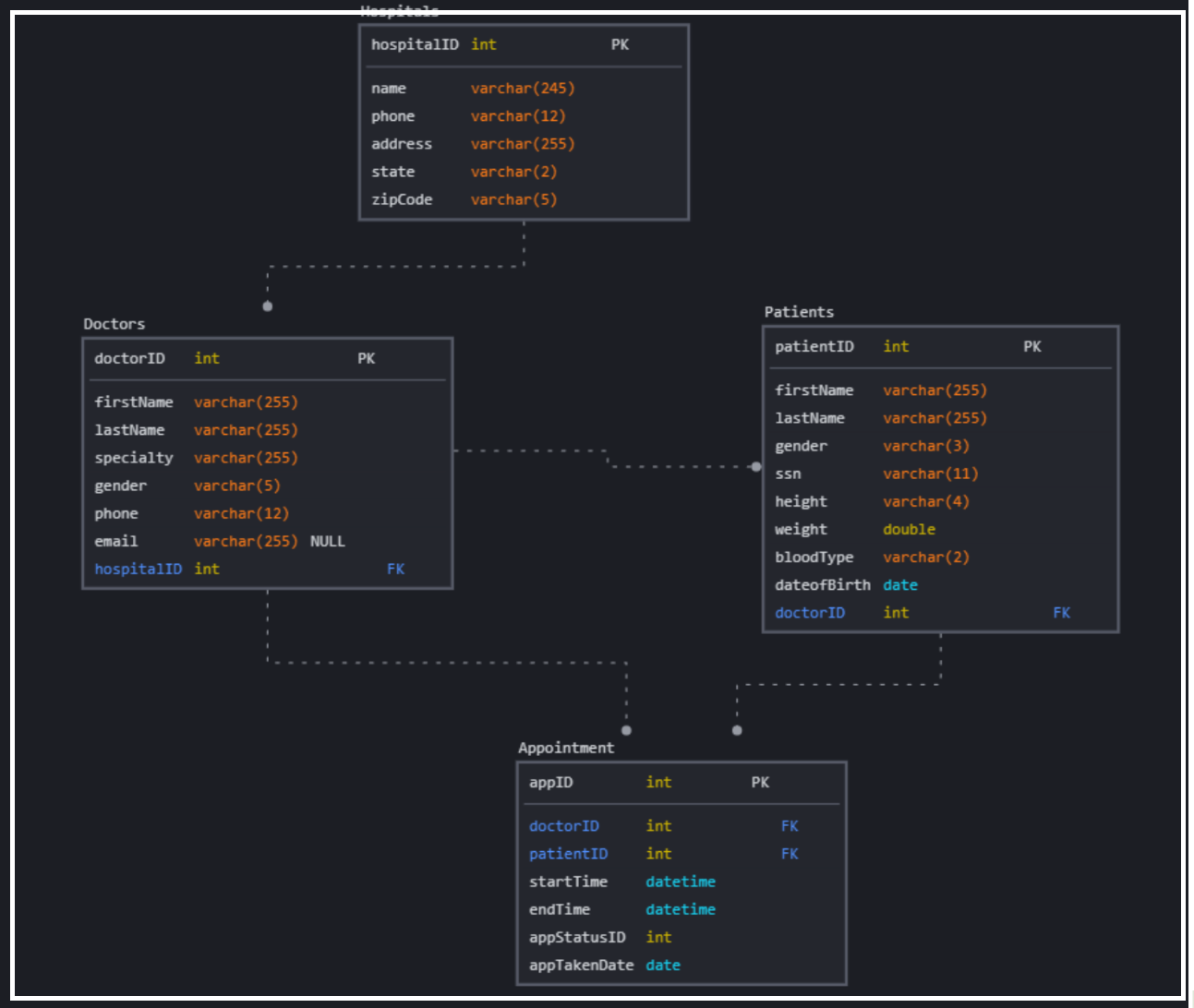
****

4.1 Functional Requirement Charts provides data sequence and classes for registration, appointment management and broader access to doctors:

****

****

4.3 The Logical Database Requirement chart specifies the logical requirements for any information that is to be placed into a database:



**Appendix C: To Be Determined List**

All TBD references in the Software Requirements Specification have been diligently reviewed and addressed during the course of the project development. As of 11/2023, there are no outstanding TBD items remaining in the SRS. The project team has successfully resolved all uncertainties, and the specifications are considered complete and finalized.