**Online Diagnosis**

**Software Requirements Specification**

Version 6.0

05/24/2023

Sai Maneendra Pingili

Naga Akhil Chaparala

Venkata Bhanu Prakash Reddy Kallam

Haritha Kothalachinta

Mohan Varasiddhi Sai Potti

Submitted in Partial Fulfilment.

of the requirements of

CSIS 44691-02 Graduate Directed Project 1

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| <date> | <Version 1> | <Your Name> | <First Revision> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents Page Number**

1. Introduction 1
   1. Purpose 1
   2. Scope 1
   3. Definitions, Acronyms, and Abbreviations 1
   4. References 1
   5. Overview 2
2. General Description 2
   1. Product Perspective 2
   2. Product Functions 2
   3. User Characteristics 2
   4. General Constraints 3
   5. Assumptions and Dependencies 3
3. Specific Requirements 4
   1. External Interface Requirements 4
      1. User Interfaces 4
      2. Hardware Interfaces 4
      3. Software Interfaces 4
      4. Communications Interface 4
   2. Functional Requirements 4
   3. Use Cases 5
   4. Class/Objects 5
   5. Non-Functional Requirements 5
   6. Inverse Requirements 6
   7. Design Constraints 6
   8. Logical Database Requirements 6
   9. Other Requirements 6
   10. Prototypes (for complete project) 7
   11. Use Case Diagrams 20
4. Design 21
   1. ER Diagram 21
   2. GUI 21
5. Analysis Models 22
   1. Data Flow Diagram 22
   2. Sequence Diagram 23

**1.Introduction**

**1.1 Purpose:**

Patients in need of medical diagnosis go through the rudimentary process of obtaining a doctor’s appointment via phone call, check for doctor's schedule in a specific hospital and then consult the doctor in person. This has become difficult during the days of global pandemics like SARS, MERS & COVID 19 in recent times.

So, to overcome this problem and to make the process of diagnosis easy and user friendly, we have come up with an idea of developing an iOS application using which patients with different kind of symptoms and diseases will be able to search & schedule appointments with any doctor on their own preference. We also provide the functionality to make a live video call/chat to the doctor for a convenient diagnosis. In addition, Patients can obtain the prescription and have the flexibility to procure the medicines at his/her choice of drug store.

**1.2 Scope:**

**The scope of the project includes:**

* We provide the feasibility of scheduling an appointment to the patients with their doctor's preference.
* In this project, we include exclusive features like live chat and video call to give the best consultation experience to the patients with their doctors.
* We would like to add extended features like patient medical history, showing the nearest pharmacy locations, adding information about the specific events conducted by the hospitals, and rating the consulted doctor.

**1.3 Definitions, Acronyms, and Abbreviations**

* Online diagnosis refers to the process of using digital tools, such as websites, apps, or virtual consultations, to determine a medical condition or illness.
* SARS: Severe Acute Respiratory Syndrome
* MERS: Middle East Respiratory Syndrome
* COVID 19: Coronavirus Disease of 2019

**1.4 References**

**[1]** Xcode Official Website – Entire Documentation of iOS Development

Link: [Xcode Developer Tools Apple](https://apps.apple.com/us/app/xcode/id497799835?mt=12)

**[2]** Firebase Official Website – A Google Cloud Storage used for Backend Database.

Link: <https://firebase.google.com/>

**[3]** Apple Developer Tools – This consists of all documentation for required Frameworks.

Link: <https://developer.apple.com/documentation/>

**[4]** Health Care Provider – List of Doctor Specializations based on their experience.

Link: <https://medlineplus.gov/ency/article/001933.htm>

**[5]** Maps Software Development Kit (SDK) - Allows the access the Hospitals near Patient

Link: <https://developers.google.com/maps/documentation/ios-sdk/overview>

**[6]** Chat Bot API – Allows Patient to start conversations with doctor and end up with call too.

Link: <https://www.cometchat.com/chat-sdks-api>

**1.5 Overview**

The idea is to develop an iOS application that simplifies the process of medical diagnosis for patients. The app will allow patients to search and schedule appointments with doctors of their preference and consult with them via live chat or video call. Patients will also be able to obtain prescriptions and purchase medications from their preferred drugstore. The app may include additional features such as patient medical history, nearest pharmacy locations, hospital events, and rating of doctors. Overall, the app aims to provide a user-friendly and convenient way for patients to receive medical consultation, especially during pandemics and other situations that limit in-person visits.

**2.General Description**

**2.1 Product Perspective**

The iOS app provides a feasible opportunity to the patient to select doctor of their own choice as well as schedule the appointment based on the availability. Our app provides live video/chat functionality for quick diagnosis. Patients can obtain the prescription and procure the medicine from the drug store all by themselves.

**2.2 Product Functions**

* Search and schedule appointments: Patients can search for doctors and schedule appointments according to their preference.
* Live chat and video call: The app provides live chat and video call functionalities for patients to consult with doctors remotely.
* Prescription and medication procurement: Patients can obtain prescriptions and purchase medications from their preferred drugstore.
* Patient medical history: The app may allow patients to store and access their medical history, which can be useful for consultations with doctors.

**2.3 User Characteristics**

* Patients: Users of the application are individuals who need medical diagnosis and treatment.
* Doctors: Healthcare providers who offer consultations to patients.
* Admin: Manages Hospitals, Healthcare providers and patients

**2.4 General Constraints**

* Availability Requirements:
  + The application should always be available to users, with minimal downtime and maintenance windows.
  + The system should be designed to minimize the risk of service interruptions due to hardware failures, software bugs, or other technical issues.
* Scalability Requirements:
  + The application should be designed to handle an increasing number of users and transactions over time.
  + The system architecture should be scalable and allow for easy addition of new features and functionality.
* Maintenance Requirements:
  + The application should be easy to maintain, with an intuitive and user-friendly interface for managing data and performing maintenance tasks.
  + The system should be able to handle updates and patches easily, with minimal disruption to user access.

**2.5 Assumptions and Dependencies**

* Development platform: The app will be developed using the Apple Xcode development platform, which provides the necessary tools and resources for building iOS apps.
* User interface design: The app's user interface (UI) will be designed using Apple's Human Interface Guidelines (HIG) to ensure that it is user-friendly and meets Apple's design standards.
* Live chat and video call functionality: The app's live chat and video call functionality can be built using Apple's WebRTC framework or third-party libraries such as Agora.io or Twilio.
* Database management: The app's database management can be implemented using a database management system such as MySQL or MongoDB.
* Security and privacy: The app will need to adhere to industry-standard security and privacy protocols to protect patient data. This may include data encryption, secure authentication mechanisms, and HIPAA compliance.
* Testing and deployment: The app will need to undergo rigorous testing and quality assurance processes before being deployed to the App Store. This may include functional testing, performance testing, and user acceptance testing.

**3. Specific Requirements**

**3.1. External Interface Requirements**

We will be having external websites like different hospital websites, drug stores.

**3.1.1. User Interfaces**

We are designing a user interface with the help of Swift Language.

**3.1.2. Hardware Interfaces**

1. Operating System (iOS, iPadOS, MacOS)
2. Work only on Apple Products (MacBook)
3. Processor (4-core Processor)

**3.1.3. Software Interfaces**

* XCODE 14.0 or Later
* iOS 9.0 or Later
* Works on iPhone and iPad

**3.1.4. Communications Interface**

The Communication Interface we are using for our project is Appstore.

**3.2. Functional Requirements**

**Admin Functionalities:**

1. Add or remove hospitals.
2. Add or remove doctors.
3. View registered patient’s details.

**Customer side:**

1. Ability to sign up/login into the app.
2. Ability to search for doctor’s after successfully logging in.
3. Ability to view the results from the search list of the app.
4. Ability to book appointments for specific doctors.
5. Ability to connect via live video call/chat with doctors.
6. Ability to obtain prescription from the doctor.
7. Ability to complete consultation payments.

**Doctor side:**

1. Accept or Reject patient’s appointments.
2. Perform chat and extend video calls in emergency.
3. Ability to Send medical prescription to patient after Treatment.

**3.3. Use Cases**

The app can be explained by use cases that demonstrate how various system actors interact and collaborate within different workflows. Examples of such use cases could be:

* Patient searches doctors
* Patient schedules appointment with doctor
* Doctor accepts/rejects appointment request.
* Doctor sends prescription to the Patient.

**3.4. Class/Objects**

The app utilizes various classes and objects that serve as the fundamental building blocks representing different entities and data structures within the system. Examples of such classes/objects may include:

* Admin: Manages the hospitals/doctors & patients with attributes such as name, hospital id, doctor id, patient id and location
* Doctor: Represents a doctor with attributes such as doctor id, name, contact number
* Customer: Represents a patient with attributes such as patient id, patient name, contact number

**3.5 Non-Functional Requirements**

**Performance**

The system must possess the capability to efficiently handle many concurrent users and database operations, ensuring fast response times for user interactions and data retrieval.

**Reliability**

To ensure reliability, the system should be available for use during normal business hours with minimal downtime, and it must be able to handle errors in a graceful manner.

**Availability**

High availability is a must, and the system should maintain a high uptime percentage. Any planned maintenance should be communicated to users in advance.

**Security**

The system must implement appropriate security measures to safeguard user data and prevent unauthorized access, including secure user authentication, data encryption, and access control.

**Portability**

For easy deployment on different platforms and environments, the system must be designed and developed in a way that supports different web browsers and devices.

**3.6. Inverse Requirements**

Inverse requirements are negative statements that specify what the system should not do or exhibit. In the case of the online diagnosis, some examples of inverse requirements are:

* Unauthorized access to sensitive data must not be allowed by the system.
* The system must not perform operations that exceed the allocated system resources.
* Conflicting check-in or check-out requests for the same online diagnosis must not be allowed by the system.

**3.7. Design Constraints**

Design constraints are factors that impose restrictions or limitations on the system's design. Here are some examples of design constraints that may apply to the online diagnosis:

* Complying with industry regulations and standards for data security and privacy is a key design constraint for the online diagnosis.
* App is compatible with planned development frameworks & App Kit integrations.
* Scalability to handle future growth and larger data volumes should be a design constraint for the online diagnosis.

**3.8. Logical Database Requirements**

The logical database requirements for the app may involve specifying the structure and organization of the databases used in the system, such as:

* Admin Database:
  + Admin Table: stores information related to working Hospital information including the name, address, sector, official email and establishment details.
* Doctor Database:
  + Doctor table: stores information about doctors including their id, name, official mail, description, field of expertise, contact details and availability hours.
* Patient Database:
  + Patient table: stores information about patients including their id, name, email, dob, gender, blood group, contact details, insurance details and medical history.

**3.9. Other Requirements**

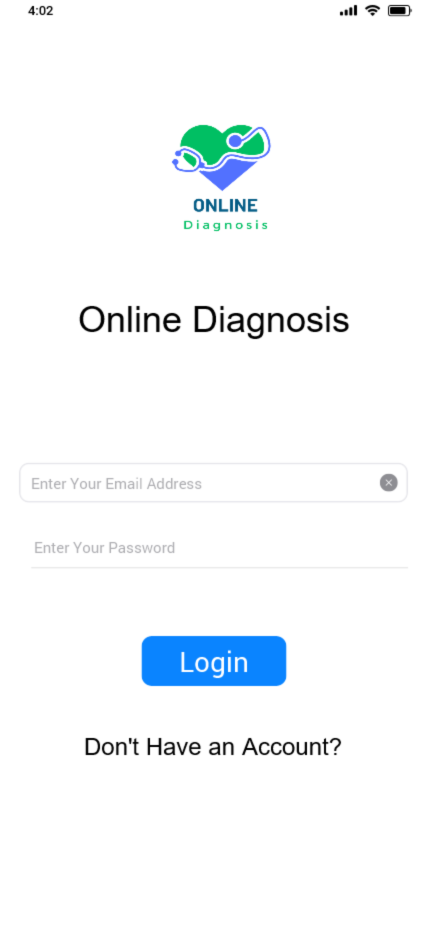
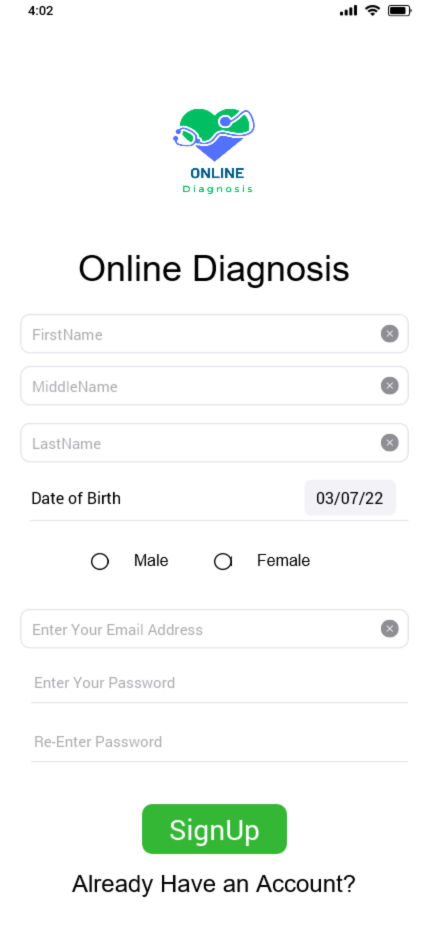
Additional specifications or considerations for the online diagnosis are known as other requirements, which may encompass:

* Ensuring compliance with data backup and recovery procedures to maintain the reliability and accessibility of data.
* Providing support for multi-language and multi-currency functionalities, if necessary.
* Incorporating third-party services or APIs into the system for features such as email notifications or payment gateways.

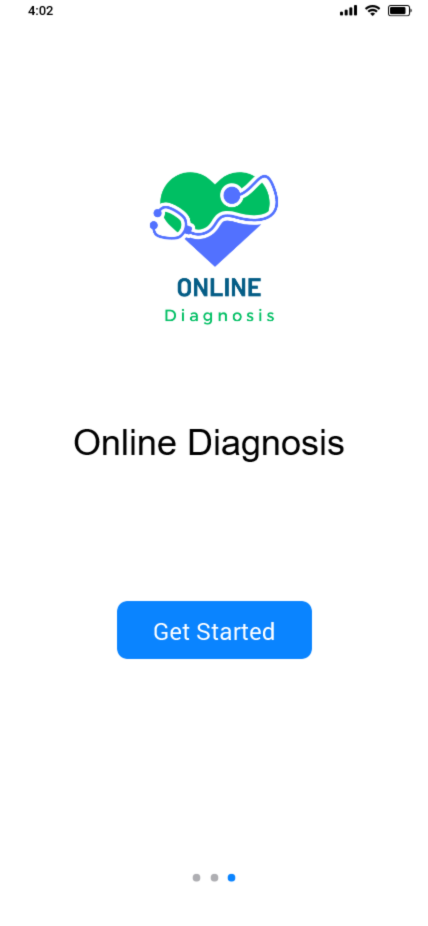
**3.10 Prototypes (for complete project)**

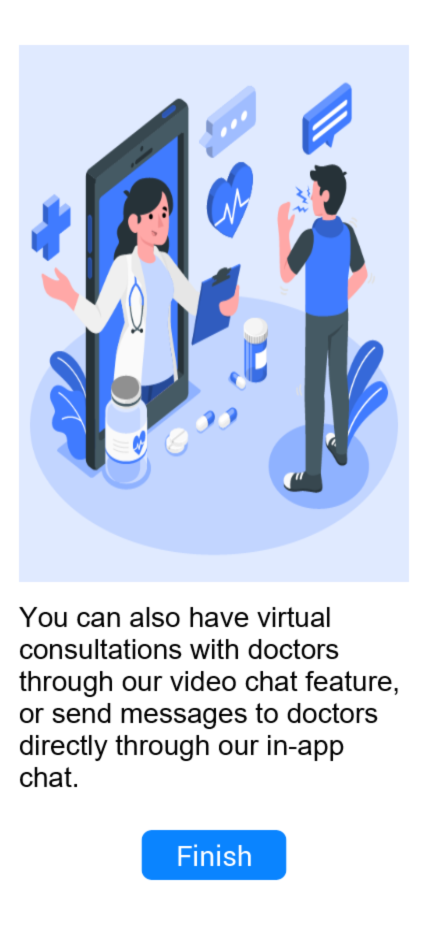
**Launch Screen**  **Login & Signup Screen**

**Login Screen** **Signup Screen**

**Welcome Screen**  **Find a doctor near by**



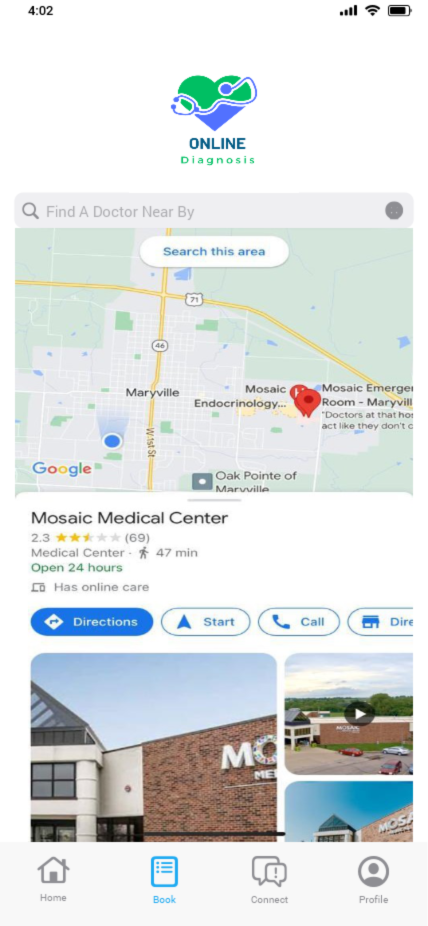
**Doctor Virtual Consulting**

**A screenshot of a phone

Description automatically generated with medium confidence A picture containing text, screenshot, font, diagram

Description automatically generated**

**Patient Home Screen Patient Booking Screen**

** A screenshot of a medical center

Description automatically generated with medium confidence**

**Locate Hospital Screen Doctor List Screen**

**A picture containing text, screenshot, font, design

Description automatically generated A screenshot of a appointment confirmation

Description automatically generated with low confidence**

**Appointment Screen Confirmation Screen**

**A screenshot of a phone

Description automatically generated with medium confidence A screenshot of a chat

Description automatically generated with medium confidence**

**Virtual Connect Screen Chat & Video Call Screen**

**A screenshot of a login form

Description automatically generated with low confidence A screenshot of a login screen

Description automatically generated with low confidence**

**Profile Screen Change Password Screen**

**A screenshot of a phone number

Description automatically generated with low confidence A screenshot of a application

Description automatically generated with low confidence**

**Medical History Screen Appointment History Screen**

**A screenshot of a appointment

Description automatically generated with medium confidence A screenshot of a phone

Description automatically generated with low confidence**

**Detail History Screen Doctor Profile Screen**

**A screenshot of a phone application

Description automatically generated with low confidence A screenshot of a phone

Description automatically generated with medium confidence**

**Doctor Home Screen Doctor Appointment List Screen**

**A screenshot of a phone

Description automatically generated with medium confidence A screenshot of a chat

Description automatically generated with medium confidence**

**Doctor Connect Screen Doctor Chat & Video Call Screen**

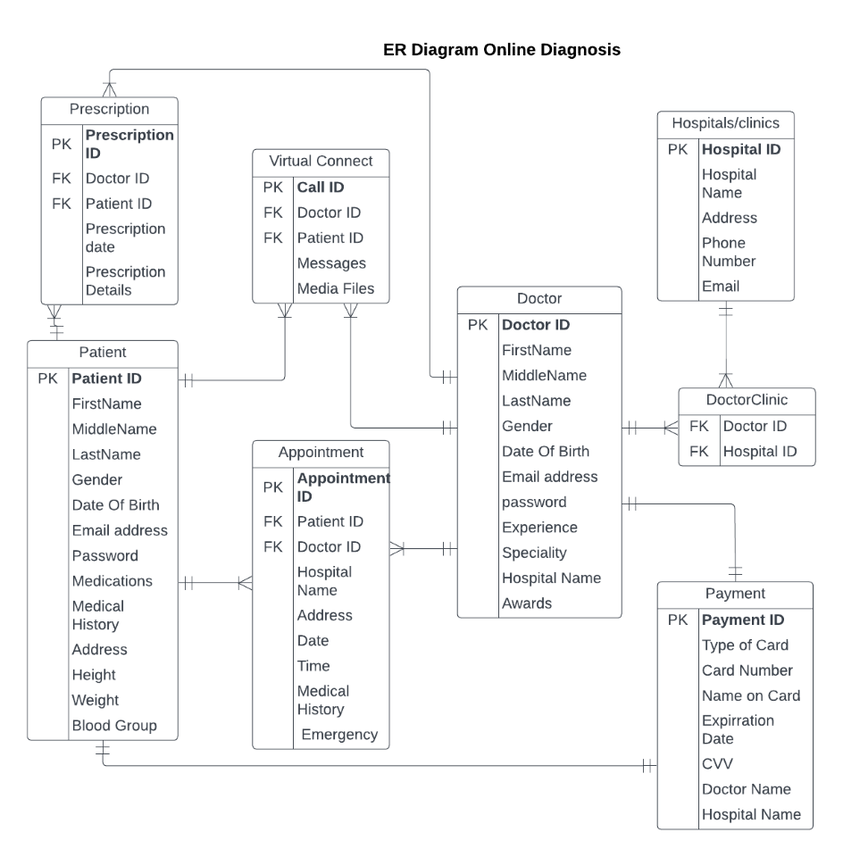
**3.11 Use Case Diagrams**

A picture containing text, screenshot, diagram, font

Description automatically generated

**4. Design**

* 1. **ER Diagram**



* 1. **GUI**

Refer to section 3.10 - Prototypes

1. **Analysis Models**
   1. **Data Flow Diagram**

A picture containing text, diagram, technical drawing, plan

Description automatically generated

* 1. **Sequence Diagram**

**A picture containing text, diagram, parallel, screenshot

Description automatically generated**