Deliverable 2: Software Requirements Specification

by

Sohaila Mohammed

Khadija Khalid

Faizan Raza

Ajla Šačić

Prepared for:

Professor Mohamad Kassab

Instructor Dena Ahmed

1. 1. Purpose:

The purpose of developing MedSync is to enable streamlined communication by centralizing the interactions between private doctors and patients into a single web-platform. MedSync strives to revolutionize daily processes and, above all, reduce the reliance on paper-based documentation, strongly prevalent in developing countries; this ensures that an increase in patient load does not affect the quality of healthcare delivery, as is often evident with paper-based documentations systems. By digitizing patients' records, our system improves accessibility and empowers both patients and doctors by providing them with direct access to documents they require in their interactions or daily life. MedSync also aims to enable patients to take a more active role in their healthcare; the centralization of health records and the ease of reaching out to one's private doctors through MedSync enables patients to more effectively manage their medical information and stay aware of their health conditions. In essence, by centralizing communication from the pre-consultation process to the post-consultation one, MedSync provides a platform that simplifies and improves the quality of healthcare delivery, contributing to better doctor and patient experiences.

1. 2. Scope:

The name of the system is MedSync. The system builds on pre-established doctor-patient relationships to allow further communication to be easier; this means the private doctors and their patients already know each other outside the platform, and they use the system to eliminate the inefficacies associated with using disparate platforms in doctor-patient communications, like having to rely on phone calls, emails, SMS, etc. Our system's scope involves users' management of their profile details, appointment scheduling and management, remote consultations, intelligent diagnosis support, and digital prescriptions and medical reports. Our system does not deal with other, more external factors of patient-doctor interactions. For instance, MedSync does not handle financial transactions such

as the billing of patients for a doctor’s services or insurance claim processing. Furthermore, while our system supports the issuing and receiving of medical prescriptions, the dispensing and fulfillment of the medication should be managed externally. Also, our system allows for the exchange of information and documents only between doctors and patients within the system—not with other parties external to the system.It must be noted that our system does not aim to replace hospital management systems; it is more concerned with the business of private doctors and their patients. MedSync also does not aim to establish new connections between private doctors and patients. Through our system, patients will only be able to connect with their personal doctors who had provided them with referral codes, through which a patient can add a doctor to his/her profile to communicate with them.For further clarification, the core functionalities scope is discussed below. It uses MoSCoW prioritization to determine if the functionality is essential for the project and elaborates on both what is in the scope as well as what is out.

Scope Description	
Functionality	**Must (M), Should (S), Could (C), Won't (W)**
Patient Appointment Scheduling	**M** (Essential for facilitating schedule management)
Patient Health Documentation and Clinical Records Management	**M** (Essential to ensure that patients can be given proper treatment, further diagnosis can be obtained, and expert opinion received)
Digital Prescriptions	**M** (Essential for ensuring proper care is given to patients in optimal time)
Remote Consultations	**M** (Important for enabling higher standards of healthcare accessibility)
Intelligent Diagnostic Support	**S** (Useful for easier and quicker diagnosis but needs to be evaluated thoroughly by doctors)
Insurances Processing and Billing	**W** (Not a core functionality of the system which primarily focuses on patient-doctor communication)
Electronic Health Information Exchange	**W** (The system does not intend to interact with other healthcare systems)

Replacement of Hospital Systems	**W** (The system does not wish to replace large system but specifically aims to enable proper healthcare management for private doctors)

The constraints and project exclusions are also discussed in order to properly establish the goals of MedSync. As mentioned previously, the main objective of the system is to streamline the interaction between private doctors and their patients by offering an all-encompassing web-platform that centralizes communication from the pre-consultation process to the post-consultation one. This goes towards achieving the main goals:

- To achieve a high level of both patient and doctor satisfaction by providing a platform that facilitates the access of medical services offered by the private doctor.
- To support the continuous development of healthcare practitioners by developing a reliable system that caters to their professional needs.
- To enhance the quality and timeliness of healthcare by having doctors place their full focus on patient care rather than elaborate administrative tasks

The top-level benefits of the system include:

- enhanced doctor-patient communication
- decreased reliance on paper-based documentation
- improved schedule management
- easier access to health records
- Simplification of more complex systems into one that incorporates only relevant features for the smaller scale

1. 3. 1. Product Perspective:

MedSync stands as an independent platform, distinct from existing healthcare management systems. It is not an extension or modification of an existing product but a novel solution tailored specifically for private medical practitioners and their patients. While it operates independently, MedSync acknowledges its integration with third-party services. MedSync will employ the ZegoCloud SDK for video consultation and the ApiMedic API for the diagnosis process. The data handling, both doctor and patient related, would be done through MongoDB. Although it functions within its own ecosystem, MedSync's integration with these external tools and services enriches its functionality and contributes to a more comprehensive healthcare management experience for both doctors and patients.

Considering an even larger scope, the digital prescriptions and clinical records that are shared through our system could be used by the users as inputs to other systems, which would indirectly interface our system with pharmacy management systems and other healthcare services. MedSync’s development team will also handle authenticating doctors who sign up for the platform; after doctors sign up, they will have to submit an application which the team will review to authorize the doctor to use his/her account. The reviewing of the doctor’s application may have to involve reaching out to the relevant medical authorities or educational institutions to verify the doctor’s credentials, which again will be part of our system’s indirect interactions with external parties.

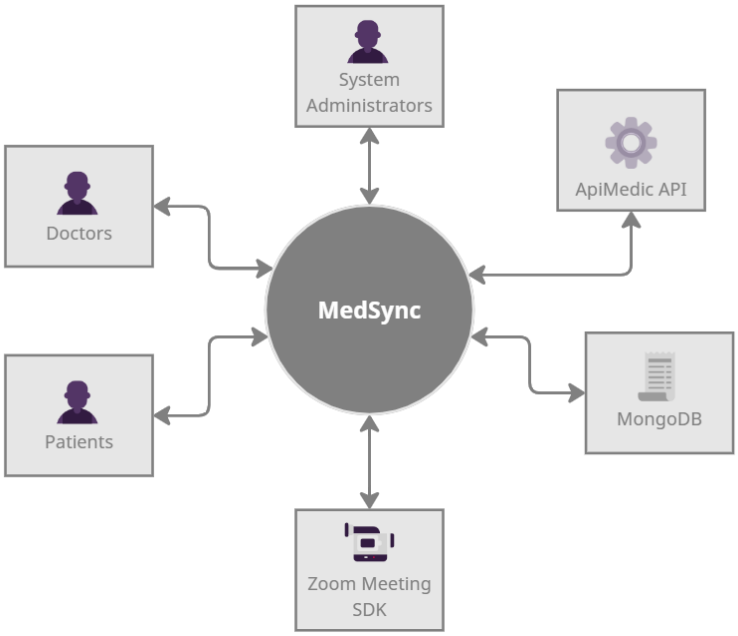


***Figure 1**: MedSync Context diagram*

1. 3. 2. Product Functions:

Provide a summary of the major functions that the software will perform. For example, an SRS for an accounting program may use this part to address customer account maintenance, customer statement and invoice preparation without mentioning the vast amount of detail that each of those functions requires.

1. Account Maintenance: MedSync will provide capabilities for both the patients and doctors to create, manage, and update their personal and professional profiles within the platform. This includes managing login credentials, personal information, and communication preferences. MedSync will also handle the verification of doctors who sign up on the platform to ensure their authenticity.
2. Appointment Scheduling and Management: MedSync will allow patients to book, cancel, or reschedule appointments with their personal doctors. The

doctors can in turn view, cancel, or reschedule upcoming appointments, as well as adjust their availability.

3. Patient Record Management: The software will enable doctors to access and upload to their patients' medical records. These include any relevant clinical information that the patient has provided or that the doctor wants to share with the patient. Patients will always be able to access their health information to stay informed and updated.
4. Remote Consultations: MedSync will facilitate virtual consultations between doctors and patients through video calls to enhance access to healthcare services without the need for physical presence.
5. Prescription Management: Doctors can issue prescriptions electronically through a designated interface in MedSync. Patients will then receive these prescriptions on their end, view them, and download them as needed.
6. Intelligent Diagnosis Support: MedSync will integrate diagnostic API to offer intelligent diagnostic suggestions to doctors based on the symptoms they entered of the patient. This will aid the doctors in making more informed, quick decisions.

1. 3. 3. User Characteristics:

MedSync is designed for two primary user groups: private doctors and their patients.

1. Private Doctors: MedSync is designed with the understanding that all doctors that will use the platform are adults with professional backgrounds in healthcare and with proficient English skills. They will have to have graduated from medical school or a corresponding institution and will have to have an MD, DO, PsyD, or other relevant qualification based on the services they want to offer on the platform. Their technical experience could range from being tech-savvy to having had minimal technological

interaction. It is expected, however, that doctors who want to offer their services online would have at least some basic technological skills. Nevertheless, the system recognizes that doctors encompass a broad age spectrum, including older doctors that may be less equipped with technological skills. Overall, MedSync must ensure that it provides an interface that simplifies patient management for doctors and enhances care delivery without adding to their workload.

2. Patients: This group has a much wider demographic range. The system specifically targets adult patients (18 years and older), who can understand, read, and speak English and who possess sufficient mental and physical ability to navigate an online platform. Future updates of the system may broaden language options and enhance accessibility features to accommodate a wider range of user abilities.

1. 3. 4 Limitations:

a) regulatory requirements and policies

The system will strive to follow the healthcare requirements that establish the proper use and implementation of an online medical system. The regulations come primarily from the USA, but are generalized across the globe. The following requirements were considered when organizing the system structure and data management:

1. **Health Insurance Portability and Accountability Act (HIPAA)**

   This is the most important federal law that protects sensitive patient information. All healthcare providers within the USA must comply with HIPAA. This is relevant to MedSync, as the system must ensure that patient

information is only accessible to that patient and the relevant doctor, and that these data are not to be shared with third parties without the patient's consent (CDC, 2022).

2. **The Health Informative Technology for Economic and Clinical Health Act**

   This act strengthens HIPAA regulations as well as sets standards for the electronic exchange of patient data (OCR, 2021). Although MedSync does not intend to engage in a large-scale electronic healthcare information exchange, the exchange of information within the private establishment is also vital in order to ensure patient satisfaction.

3. **The Interoperability and Patient Access Rule (CMS-9115-F)**

   This regulation ensures that patients are able to access their information at any point in time (CMS, 2023). This is important in ensuring that patients' access to their own data on MedSync always remains easily possible–no feature should block them from accessing their personal data. As MedSync is a web application, the ability to provide a continuous connection between the patients and doctors also primarily relies on the user's ability to access technology and the internet.

4. **The National Institute of Standards and Technology (NIST) Guidelines**

   This regulation ensures that there are proper cybersecurity measures put in place, not only within EHR systems but also any other system which requires sensitive data management. The compliance with this measure restricts the storing of data to the standards which enable its safekeeping.

5. **Web Content Accessibility Guidelines (WCAG) 2.1**

   Published by the Web Accessibility Initiative of the World Wide Web, these guidelines dictate how to make Web content more accessible to people with

disabilities. The project will mainly focus on the following guidelines (World Wide Web Consortium, 2020):

a. 1.4 - The content shall be distinguishable such as to make it easier for users to see and hear content.

b. 2.4 - The content shall be navigable and the system shall make sure that there exist ways to help users navigate, find content, and determine where they are.

c. 3.2 - The content shall be readable.

d. 3.2 - The content shall be predictable such that pages appear and operate in predictable ways.

e. 3.3 - The content shall provide input assistance such that it helps users avoid and correct mistakes.

b) interfaces to other applications

MedSync facilitates ApiMedic, an API which serves as a medical symptom checker. Its integration within the system could pose compatibility issues and restrict the design choices. Furthermore, since the team does not have contact to the most important stakeholders, the private doctors, there is no way of testing their interactions with the API which would mean that their non-exposure to it prior to the systems publishing would make them less likely to trust the API, significantly increasing the risk of the feature not being used.

d) quality requirements (e.g., reliability)

Time constraint of the project might reduce the reliability of the application on delivery. The proper implementation of an online medical system takes much longer than a few weeks to produce so the reliability of an application developed within a short period of time by a small team might be significantly reduced.

e) safety and security considerations;

MedSync aims to uphold proper cybersecurity measures as exemplified by the regulations and laws above. It primarily relies on the use of well established encryption, hashing and salting techniques for the storing of confidential information within the system. However, the time constraint might significantly reduce the ability of the team to properly test the security features, limiting the system to only the tests that the system developers undertake. Industry standards for security testing such as Static Application Security Testing (SAST) are beyond the scope of the application at this point and although the system can be ensured to withstand injection attacks through limiting database code and restricting database access, the proper evaluation of assembly or source code is not feasible for the providers now.

1. 4. Definitions

a) What is a private doctor?

A private doctor, also called private practitioner or private physician, is a healthcare professional who delivers medical services independently of the public health system. Such physicians could be working for private clinics or hospitals, or they could be catering to patients on a direct-payment basis for a range of personalized healthcare services (Macdonald, 2023). In the context of MedSync, we are referring to the latter when we speak of private doctors.

b) What is a medical prescription?

A prescription is an official order from a licensed medical practitioner or healthcare professional to a pharmacist. It permits the pharmacist to provide a designated medication to a particular patient. This is important because some medications can only be legally acquired through a doctor that prescribes them. (Australian Government Department of Health and Aged Care, 2022)

c) What is a clinical record?

Clinical records consist of various documents related to patient care, created by healthcare professionals. These can include electronic notes, laboratory results, imaging records, consent forms, and the like. (*What Are Clinical Records*, n.d.) Other general terms used to refer to clinical records in this documentation are “medical reports” or “medical records”.

d) What is a medical consultation?

A medical consultation is a professional meeting between a patient and a physician or healthcare provider, aimed at diagnosing health issues, discussing symptoms, and developing a treatment plan if needed. In modern times, this encompasses online communications as well. (Poston, 2023)

3. 1. Functions:

1. Account Management

1.1. The system shall enable a user to create an account on the platform.

1.2. The system shall activate the account of a doctor only through instructions issued by the development team.

1.3. The system shall provide doctors with referral codes that they can share with their patients.

1.4. The system shall save all users’ information in a database.

1.5. The system shall hash and salt all users’ passwords.

1.6. The system shall enable users to view their profile information.

1.7. The system shall allow users to modify their profile information.

2. Appointment Management

2.1. The system shall allow patients to schedule new appointments only with the doctors connected to their profile.

2.2. The system shall allow users to reschedule their appointments.

2.3. The system shall enable users to cancel their appointments.

2.4. The system shall allow users to view past and upcoming appointments.

3. Medical Records Access

3.1. The system shall allow patients to view past and recent medical reports accompanied with their dates.

3.2. The system shall allow patients to download any of their medical reports.

3.3. The system shall allow doctors to view only their patients' medical records

3.4. The system shall allow doctors to update their patients' medical records

4. Prescription Management

4.1. The system shall allow doctors to issue prescriptions for their patients through the platform.

4.2. The system shall allow users to view their past and recent prescriptions accompanied with their dates and issuing doctors.

4.3. The system shall allow users to download prescriptions issued in the past 6 months only.

5. Communication

5.1. The system shall enable a patient to establish a connection with his/her doctor on the platform using referral codes received from the doctor.

5.1.1. The system shall enable doctors to confirm a patient's connection request.

5.2. The system shall offer an encrypted video call option for remote consultations through the platform.

6. Lab and Radiology Reports

6.1. The system shall allow laboratory and radiology staff to upload patient tests and images

6.2. The system shall allow doctors to view the test results and images.

6.2.1. The system shall allow users to view their lab results and radiology reports, with the approval of their doctor.

7. Intelligent Diagnostic Support

7.1. The system shall enable doctors to input patient symptoms into a dedicated interface for ApiMedic.

3. 2. Performance Requirements:

Performance requirements are a crucial aspect of MedSync which deals with important aspects of human life. It is important that it ensures the capabilities of the system to deliver optimal user experience, establish a baseline for the efficiency, and support the workload without compromise. To discuss these requirements, we will define both static ones which relate to the system's capacity and architecture as well as dynamic ones which relate to system operations and user interactions with the system.

3. 2. 1. Static Performance Requirements:

1. System Capacity

Generally, a general practitioner or a physician deals with about 2,500 patients in their work (Weber, 2019). Although this number is not high to begin with, it is important to note that the implementation of the system will be made in a short time frame, so its scale will be on the smaller side as we aim to strike a

balance and not overwhelm the newly developed system. Hence, the system should support 50 patients for each private doctor on the platform. Further, the system should support up to 100 doctors on the platform. This cap is set in order to properly deliver the functionalities offered and test the system without overburdening the system developers.

2. Data storage

The system should be able to store medical records, patients' and doctors' profiles, appointment histories, and prescriptions within a data storage solution implemented by the developers. Even though the system is not supporting a comparatively huge number of patients to begin with, it is important to note that the demand for storage will still be high as the data piles up quickly. At the present point, the system should reserve at least 1GB of space for storing each patient's information.

3. Security Infrastructure

The system should uphold the aforementioned data protection regulations in order to properly establish the doctor-patient communication.

3. 2. 2. Dynamic Performance Requirements:

1. User Load

The system should be able to support all of the users who are registered onto the system. At a single time, the system should be able to uphold at least 550 active users. Although they might rarely be all active, it is important to consider the extreme cases (perhaps during flu season or epidemics/pandemics) while developing the system.

2. Response times

It is important to also consider critical transactions such as appointment scheduling or intelligent diagnosis support, which should have quick response

times. Regarding appointment booking, the system should process a new appointment booking in at most 15 minutes. Additionally, the system should process an appointment modification in at most 5 minutes. As for interactions with ApiMedic, the system should display the diagnostic suggestions from ApiMedic in less than 20 seconds from the point of request submission. It is also worth highlighting that during peak times for any activity on the platform, the system should process requests in at most double the time as when the load is normal.

3. 3. Usability Requirements:

Through domain analysis of similar apps and prototyping our own, we discovered various usability requirements for multiple scenarios in the MedSync system.

In the account maintenance scenario, the need to first verify a doctor's account before he/she can use it is crucial to ensure that doctors who use the platform are legit. Through domain analysis, we were inspired to adopt similar easy methods through which a doctor can verify his/her credentials, to enhance user satisfaction and streamline the account management process. Therefore, the system shall provide an application method for doctor accounts through which doctors can submit their request to offer their services on the platform. Additionally, the system shall provide a doctor with instructions on how he/she can verify his/her credentials to have his/her account activated.

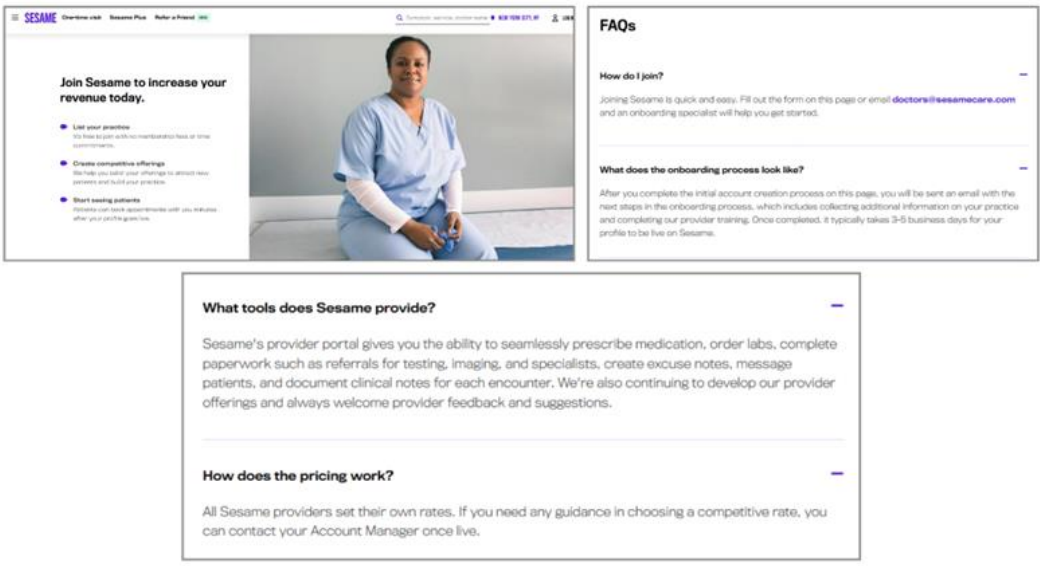


***Figure 2**: Domain analysis through Sesame platform*

***Figure 3**: Domain analysis through HealthTap platform*

Similarly, in appointment scheduling and management, prototyping highlighted to us the importance of clearly indicating to the users a doctor's availability if they want to book an appointment. To facilitate that, the system shall automatically update a doctor's availability for every appointment booked with him/her.

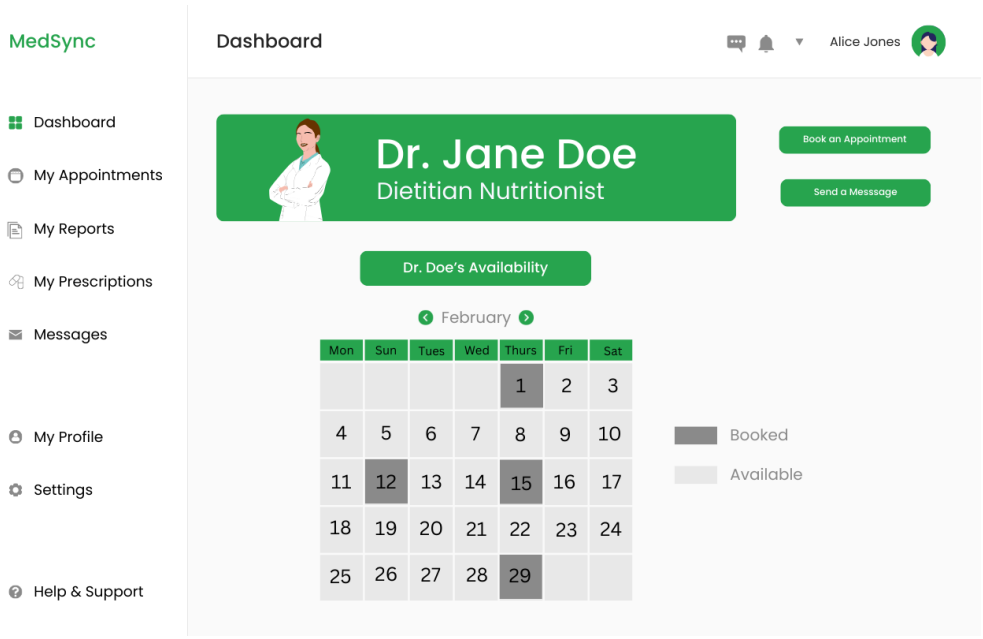


***Figure 4**: Prototype of a doctor's profile from a patient's perspective in Medsync*

For intelligent diagnostic support, prototyping helped us conclude that access to ApiMedic should be restricted to doctors only. This decision was reached as a safety measure because patients should not be diagnosing themselves; instead, results from ApiMedic are better off considered by professionals who will better know how to interpret them. The system shall provide access to ApiMedic's services to doctors only.

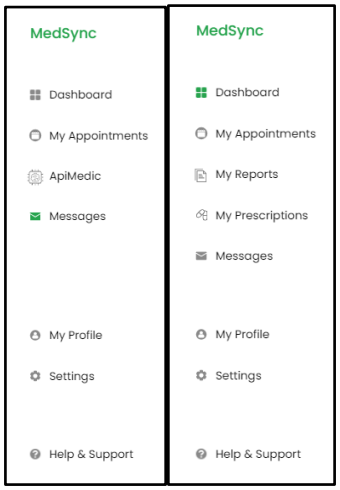


***Figure 5**: Prototype of a doctor's side menu (left) and a patient's (right) on the platform*

Again through prototyping, we recognized the importance of ensuring our users are able to have their queries answered or concerns addressed. Thus, the system shall feature a dedicated section that answers the most common questions users are expected to ask. Additionally, the system shall incorporate a mechanism for collecting user feedback to capture their satisfaction level.

3. 4. Interface Requirements:

3. 4. 1. User Interface

The system will integrate a user interface for a web based application. To do that, the team will comply with certain UI requirements and standards. The first one is consistency with existing EHR systems such as SimplePractice, which is an EHR system primarily intended for mental health practices but an application which provides important features, for example, patient scheduling. The second one is color scheme and typography which is going to comply with usability standards such as WCAG which is mentioned in the regulatory requirements and policies section. The use of standard typography and WCAG guidelines will distinguish the different sections and common actions such as scheduling, prescription and profile management. The third requirement is responsiveness which will ensure that the web application can be accessed from both the phone and the desktop. Though the primary use of the system sees itself in the development of a web application for the desktop computer, it is important to make the UI adapt to the other EHR system standards. The fourth standard is the error message display which will display clear non-technical error messages when the user does not comply with the system requirements. This can be invalid or missing input which will prompt alerts to correct the mistakes. Last requirement is the standardized UI which requires that it involve features such as the help option for accessibility.

3. 4. 2. Hardware interfaces

The system shall primarily be developed for a web interface. The team shall ensure responsiveness given that it has proper time to implement it.

3. 4. 3. Software interfaces

The system uses an external API, ApiMedic, for easier diagnosis of patients. This API requires that the doctor has an API key which is provided by creating an account on the API website. The API offers 100 searches per month for free for any of the features it offers such as the diagnosis feature MedSync will utilize. The user will send a JSON file with symptom codes, patient age and their sex, and receive back a JSON file containing the diagnosis, model confidence and area of expertise for the diagnosis care. The JSON file will be presented in a suitable manner for the user.

3. 4. 4. Communication interfaces

The message formatting utilizes HTML and CSS as well as client-side JavaScript for validation. The system needs to ensure that asynchronous data submission does not hinder its performance which can be mitigated by proper setup of callbacks or implementation of synchronous processes. The protocols used are TCP/IP for basic communication and HTTP(S) for web services.

3. 5. Logical Database Requirements:

3. 5. 1. Frequency of use:

- Appointment Management:
    - The system shall maintain a high-frequency read and write capability for appointment schedules; it shall accommodate frequent operations during both appointment creation and updates. This functionality is critical to ensure real-time scheduling and management.

- Medical Record Access:
    - The system shall support frequent reads and occasional writes to access and update patient medical records. This feature is vital for patients to view their health history and doctors to provide accurate and up-to-date information during consultations.
- Prescription Management:
    - The system shall facilitate highly frequent reads and occasional writes for digital prescriptions. This functionality is essential for both patients and doctors to access and update prescription information.
- Intelligent Diagnostic Support:
    - ApiMedic, being a crucial component, will be accessed with medium frequency by authorized doctors for diagnostic support. The system shall support the frequent reads and writes expected during consultations, helping doctors make informed decisions.
- Communication Features:
    - Video call logs, supporting communication, will be subject to frequent reads during communication sessions. This ensures a seamless communication experience for both patients and doctors.

3. 5. 2. Accessing Capabilities

- Account Management:
    - The system shall enable all users to modify their personal profile information. Further, the system shall only allow connected users to view each other’s profiles.
- Appointment Management:

- The system manages appointment schedules with a focus on providing authorized access to patients and doctors. The system shall enable all users to see their appointments; it shall enable patients to book appointments with connected doctors; it shall enable all users to modify appointment-related data. This careful approach maintains the integrity of the scheduling process, contributing to the effective management of appointments.

- Medical Record Access:
  - Access to detailed medical records is facilitated with precision, allowing patients and doctors the necessary information based on their authorization levels. The system shall only allow a patient and all his connected doctors to view his/her medical records. Additionally, the system shall only allow a patient and all his connected doctors to update his/her medical records. Unconnected doctors and other patients cannot see nor modify that patient's medical records. This tailored access approach prioritizes privacy and confidentiality, contributing to effective medical record access.

- Intelligent Diagnostic Support:
  - The system shall restrict access to ApiMedic, allowing only authorized doctors (those whose accounts have been activated by system administrators) to utilize intelligent diagnostic support. This measure prevents unauthorized use, ensuring that only qualified medical professionals can access diagnostic recommendations. This controlled access contributes to maintaining a professional oversight of diagnostic features.

- Prescription Management:
  - Access to digital prescriptions is carefully authorized based on user roles. Doctors, with the necessary privileges, can create and issue prescriptions, while patients have access limited to viewing and downloading their

prescriptions. This controlled access approach adds a layer of protection to prescription-related information, contributing to effective prescription management.

- Communication Features:
    - Access to video call functionalities is provided for all users, but a video call instance will only be accessible to the patient and doctor involved in the consultation. This discreet access approach ensures that sensitive discussions remain private, contributing to effective communication within the system.

3. 5. 3. Data Entities and Their Relationships

- Account Management:
    - Entities: Patient, Doctor, Profile
    - Relationships: The system shall establish relationships where each user (patient or doctor) is associated with his/her respective profile.
- Appointment Management:
    - Entities: Appointment, Patient, Doctor
    - Relationships: The system shall establish relationships where an appointment is associated with a specific patient and a designated doctor.
- Medical Record Access:
    - Entities: Medical Record, Patient, Doctor
    - Relationships: The system shall define relationships where a medical record is owned by a patient and contributed to by the doctor.
- Intelligent Diagnostic Support:
    - Entities: ApiMedic, Doctor

    - Relationships: The system shall establish relationships where symptom checking requests into ApiMedic are linked to a specific doctor for diagnostic suggestions.

- Prescription Management:
    - Entities: Prescription, Patient, Doctor, Medication
    - Relationships: The system shall define relationships where a prescription is created by a doctor, issued to a patient, and contains specific medications.
- Communication Features:
    - Entities: Video Call Log, Patient, Doctor
    - Relationships: The system shall establish relationships where video call logs are associated with specific patients and doctors, ensuring accurate tracking of communication.

3. 5. 4. Data Retention Requirements

- Account Management:
    - The system shall retain profile data connected to a user's account for as long as the account exists. This ensures the continued availability and access of users' information.
- Appointment Management:
    - The system shall retain appointment data connected to an account for as long as the account exists. This ensures the availability of past appointment information for analysis.
- Medical Record Access:

    - The system shall retain medical records connected to an account for as long as the account exists. This supports the continuity of care and ensures the availability of historical health data.

- Intelligent Diagnostic Support:
    - The system shall retain diagnostic support results for as long as the request session lasts with ApiMedic. This ensures doctors get the diagnosis support needed without overburdening the system with inquiry results.

- Prescription Management:
    - The system shall retain prescriptions connected to a patient's account for as long as his/her account exists. This ensures the availability of past prescription information for reference and compliance.

- Communication Features:
    - The system shall retain video call logs connected to an account for as long as the account exists. This ensures transparency and accountability in communication records.

3. 6. Design Constraints:

The system is primarily intended to be a web application which means its use is limited to systems such as Windows, macOS and Android and iPhone smartphones. To rely less heavily on the quality of internet connection, the system shall employ image reduction such that resolution of images and other UI objects does not hinder the access to the web page. This is also to stay compliant with the WCAG. Most of the other requirements and limitations are not directly related to standards compliance but time as mentioned before. It is also important to note the budget constraint. Considering that the team is not working on any budget, the implementation of the system relied on free software and open-source APIs that can be used. The ApiMedic provides 100 free queries per month which is suitable for the project's needs but not real life applications. Furthermore, the use of free software entails a significant constraint on the degree of freedom within the system itself. The dependency on external factors which are beyond the team's reach does not allow complete flexibility with and freedom of design of the system. Additionally, the system does not require special hardware restrictions to run; it simply needs any device that is capable of opening websites. As such, there are no relevant hardware limitations.

3. 7. Software System Attributes:

We shall designate the software system attributes into five categories, design and software architecture, runtime, system, user and non-runtime quality attributes.

3. 7. 1. Design and Software Architecture

This category contains attributes such as maintainability, reusability and correctness.

Maintainability refers to the ease with which the product can be maintained, its parts replaced or updated as well as how it responds to the emergence of new technologies. To ensure that the system can navigate these challenges, the team will ensure proper documentation of the system and make its implementation understandable such that any new changes are easy to implement and also understand.

Reusability means that the components of the system, such as code can be reused within the same or different system. The implementation of MedSync which is constrained to an object oriented design entails that the team will ensure that the system features can be reused both within the system and beyond it, i.e., in other systems. The cost of system implementation also decreases with the reusability of components, but considering that the team does not have a budget, the reusability mainly implies the latter option – its use beyond the current implementation.

Correctness refers to the ability of the software to act with respect to its purpose. To ensure the correctness of code and its implementation, the team will follow a code review checklist as well as run tests which will be mentioned later.

3. 7. 2. Runtime

Runtime qualities are qualities which ensure the ability of the system to do the work it is intended to do. They include 'ilities' such as reliability, interoperability, scalability, and security.

Reliability includes a few sub-categories such as availability, fault tolerance and recoverability and is a quality which ensures that systems behave as expected. Although it might be beyond the team's abilities since the deployment of the website on a server is expected to be maintained by the people behind the server, the team shall ensure that any failure within the system is quickly resolved and the web application recovered.

Interoperability is the ability of the system to communicate with those outside it, such as databases. It is ensured by proper implementation of the communication procedures within the code. To maintain interoperability the team will ensure that standard procedures are followed when enabling the communication between the system and that which is outside it such as using standard packages for enabling the web application to store database information as well as retrieve it rather than implementing new methods which might not be compatible with the standard approach.

Scalability refers to the ease at which the system can adapt to an increase in the demand without affecting the performance. To ensure this, the system will be developed beyond the currently required load, i.e., it will be implemented for a higher number of users than what's currently required by its documentation.

Security is the ability of the system to safeguard its information against unauthorized personnel. Although the system cannot ensure all aspects of security due to time constraints, the safeguarding of personal information such as passwords will be ensured through hashing and salting. The ability of users to only access their own information will be prioritized in the implementation of the system. However, testing for security breaches is outside of the scope of the project.

3. 7. 3. System

System quality attributes include testability and supportability.

Testability is the level of easy quality checks that can be done on the system to ensure any defects which might hinder its performance are removed. The testing of the system will be eased through the implementation by sequential testing of components as well as running the test mentioned later.

Supportability is the degree to which the system is able to provide information for its faults. To ensure that the system can provide enough information for any of its faults, the team will employ logging which will record any issues not present at the high level of the system.

3. 7. 4. User

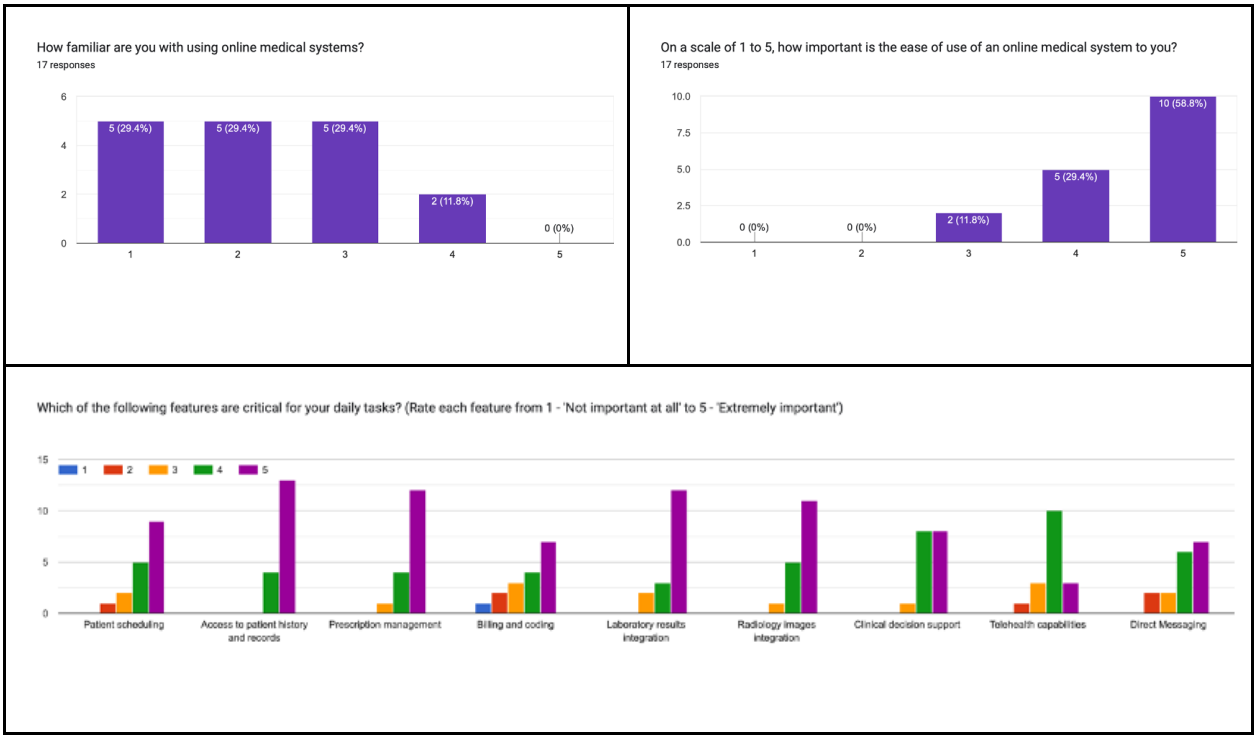
The most important user quality attribute is the usability which ensures that the user has an easy time while using the system. This mainly refers to the design and accessibility which the team will ensure to maintain up to standard by following the WCAG guidelines mentioned above.

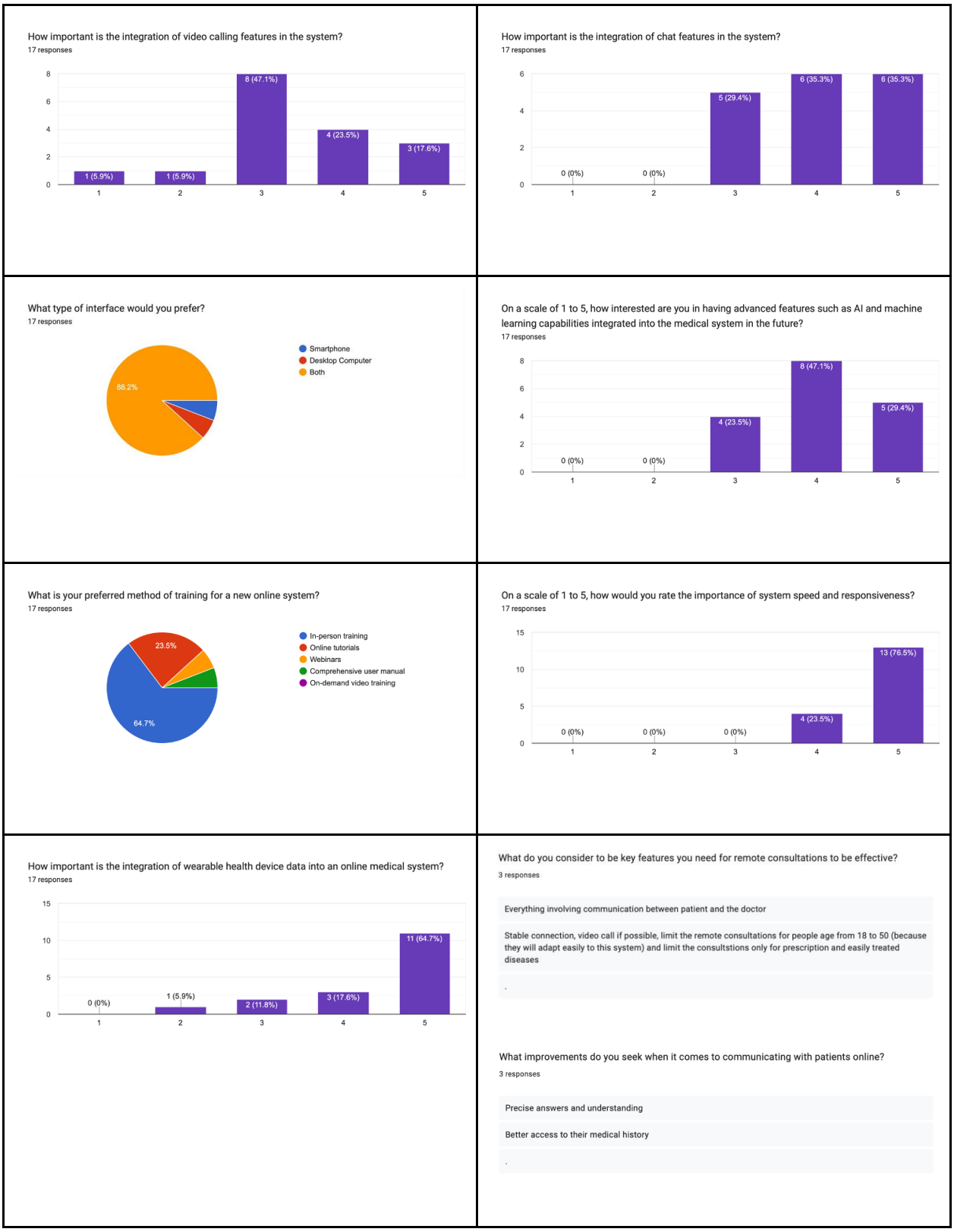
3. 7. 5. Non-runtime

Non-runtime quality attributes are attributes which cannot be observed as the system executes. These include primarily portability as well as reusability which was mentioned prior. Portability ensures that the system can migrate from one environment or hardware to another. Since the system is a web application, the team will ensure that the system is responsive and that different web browsers are supported.

3. 8. Additional Information:

The team did conduct a very small questionnaire within Medical Faculty University of Sarajevo to get a better look at the requirements future doctors want when using an online medical system they are not familiar with. The result of the poll is below and any scales present on the poll reflect 1 for no familiarity with the system or importance of a feature and 5 for extreme familiarity or importance of a feature. The poll received 17 responses, 16 from the Medical Faculty University of Sarajevo and one from Istanbul University-Cerrahpaşa.





***Figures 6-12**: Questionnaire answers for EHR requirements*

4. Verification:

- Code Review: As we're working behind the scenes, we will ensure that we are reviewing our code periodically to make sure that we have used the most effective methods in the implementation of features (like avoiding unnecessary looping, using too much memory, etc) and to spot any security vulnerabilities that could be caused by our design choices.

- Functional Testing: We will develop test cases on each functional requirement listed in the SRS. For each software function, we will compare it with its associated requirement to ensure its output aligns with what the end user anticipates. This will involve supplying test inputs to the functions, recording the outputs produced, and confirming that these actual outputs match the anticipated results. (*What Is Functional Testing?,* n.d.)

- API Testing: For features relying on third-party services like ApiMedic, we will conduct API tests to verify that the API interactions are correctly implemented and that data flows seamlessly between MedSync and these services. To do this, we will test the APIs with extreme inputs and observe the correctness of the outputs. If we ensure the success of communication under these challenging scenarios, we can be assured that the API will function as needed under regular usage conditions. (Inflectra, 2023)

- Load Testing: Through a load testing tool, we will be able to simulate high-traffic conditions on the website to ensure it maintains its functionality and performance. We will be able to observe if the system gets significantly slower or lags, helping us identify any potential bottlenecks or scalability issues. (*What Is Load Testing?,* n.d.)

- Beta Testing: Releasing a "beta" version of the web application will serve as a critical phase where we have many "real" users engage with the software in a live environment. This will help us identify any bugs or usability challenges reported by the users before the "official" launch of the platform. (*Beta Test*, n.d.)

5. 1. Assumptions and dependencies:

5. 1. 1. Assumptions

5.1.1.1. Operating System Compatibility:

a) Assumption: The system assumes deployment on popular servers running operating systems such as Linux (Ubuntu 20.04 LTS) or Windows Server 2019.
b) Rationale: Certain functionalities, such as server configurations and security features, may be OS-specific. The system assumes compatibility with the chosen operating environment for optimal performance and reliability.

5.1.1.2 Internet Connectivity:

a) Assumption: The system assumes a stable and reliable internet connection with a minimum bandwidth of 5 Mbps.
b) Rationale: MedSync relies on real-time communication, scheduling, and external API integrations. A constant, high-speed internet connection is essential to ensure smooth and uninterrupted operation.

5.1.1.3 User Cooperation:

a) Assumption: Users are assumed to follow proper input procedures, undergo mandatory training on system usage, and not engage in intentional misuse.
b) Rationale: The security and integrity of patient data depend on users adhering to prescribed usage patterns, maintaining the confidentiality of sensitive information, and undergoing necessary training to use the system effectively.

5.1.1.4 Availability of Doctors and their Medical Facilities:

a) Assumption: The system assumes that affiliated doctors maintain their operational status.
b) Rationale: MedSync relies on appointments and interactions with doctors and their medical facilities. The availability and functionality of these facilities during standard hours are assumed for proper system operation.

5.1.1.5 Timely Data Entry:

a) Assumption: Medical professionals will enter data, including appointments and prescriptions, within 24 hours of the occurrence.
b) Rationale: The accuracy and reliability of patient records and treatment plans depend on the prompt entry of data by healthcare providers. Timely data entry ensures up-to-date information for better patient care.

5. 1. 2. Dependencies

5. 1. 2. 1. External APIs and Services:

- Dependency: The system depends on external APIs for functionalities such as video consultations (ZegoCloud SDK) and intelligent diagnostic support (ApiMedic).
- Rationale: Changes or disruptions in these external services, such as updates to APIs or service outages, can directly impact the corresponding features in MedSync.

5. 1. 2. 3. User Training and Education:

- Dependency: The successful adoption of MedSync depends on users receiving mandatory training sessions and educational materials about the system's functionalities.

- Rationale: Users need to understand how to effectively use the system to maximize its benefits, minimize errors, and adhere to best practices in healthcare information management.

5. 1. 2. 4. Collaboration with Doctors and their Medical Facilities:

- Dependency: The system's effectiveness relies on ongoing collaboration with doctors and their medical facilities for accurate scheduling, availability information, and real-time updates.
- Rationale: Accurate and up-to-date information from doctors and their medical facilities ensures the system's alignment with real-world healthcare scenarios and enhances the overall user experience.

5. 1. 2. 5. Stakeholder Engagement:

- Dependency: Continuous collaboration with stakeholders, including healthcare professionals, system administrators, and patient representatives, is crucial for ongoing feedback, improvements, and system evolution.
- Rationale: Stakeholder involvement ensures that the system remains aligned with evolving healthcare needs, industry standards, and user expectations.

5. 2. Acronyms and abbreviations:

- **API** - application programming interface
- **DO** - Doctor of Osteopathic Medicine
- **EHR** - Electronic Health Record
- **MD** - Medicinae Doctor (Doctor of Medicine)
- **MoSCoW** - must-have, should-have, could-have, and won't-have or will not have right now
- **PsyD** - Doctor of Psychology

- **SDK** - software development kit
- **UI** - user interface
- **USA** - United States of America

6. 1. Appendix:

Addressing the feedback of our project proposal:

Firstly, thank you for bringing these points to our attention.

1. Stakeholders: What about negative stakeholders?

We have identified the following stakeholders as entities that will potentially be negatively impacted by MedSync:

- Hospitals [LOW]: Hospitals might view digital solutions to healthcare services, like MedSync, as threats to the traditional medical establishments. Particularly, hospital-employed doctors might perceive MedSync as a competitive threat that can impact their job security.
- Traditional Reception Staff [LOW]: The adoption of MedSync may automate certain administrative tasks, potentially reducing the workload for traditional reception staff. Their roles in appointment scheduling and record-keeping could be diminished.
- Paper-Based Record Management Services [LOW]: Services relying on paper-based record management may experience a decline in demand as MedSync shifts the paradigm towards digital record-keeping, potentially impacting their business negatively.
- “Technophobes” [LOW]: Generally speaking, the implementation of more technology-driven solutions in medical interactions will not be perceived positively by those who dislike new technologies or avoid using them.

2. Time constraints: What are your plans in case the time is not enough or if you encounter an unexpected setback?

In response to time constraints, our strategy will include breaking down the project into smaller, manageable segments that can be completed in stages. This way, we can have functional parts of the project completed at each stage. We will also prioritize core features of the platform that target requirements of a “high” level to ensure they are developed first. We also hold two meetings weekly to ensure that all members are kept on track with the workflow of the project and that we are all aware if there are any obstacles in the development of the platform. If faced with unexpected setbacks, we will reassess our feature list, deprioritizing or postponing non-critical features to focus on delivering a minimum viable product (MVP) within the deadline.

3. Carefully consider all the factors related to integration with external entities, using APIs/SDKS, or databases and study if this is actually feasible within the context of your project. Also, study any associated issues/risks and how they would be mitigated before proceeding with the rest of the project.

Regarding integration with external entities, we have indeed considered the feasibility of this integration and determined that it is the most practical decision as it allows us to implement features into our platform that would otherwise need much more time than the timeline allotted to the development of the project. The ApiMedic API provides a free plan that would be suitable for our project, and they also provide tutorials that will help us easily implement the integration (priaid-eHealth, 2018). The same thing applies to the ZegoCloud SDK, as they provide free plans and tutorials to follow (ZegoCloud, 2022); it is essential for our platform to facilitate a video call feature between the doctors and the patients, but we would not be able to do this in the limited time frame without relying on external integration. As for databases, we only need MongoDB for database management to handle our own platform’s data, which is essential to manage the users on the platform and would be impractical to go about otherwise. Granted, we do acknowledge the

potential for compatibility issues by our reliance on third-party services. To mitigate these risks, we will perform thorough testing of these integrations early in the development cycle to ensure that we recognize any obstacles early on. If such issues arise in the early test, we will look into alternative APIs or SDKs. In the worst case scenario, we will have to resort to implementing the features reliant on third-party services ourselves, making sure that we simplify them enough to be able to replicate them.

4. Constraints: Elaborate on specific strategies for addressing these constraints.

The time management and external integration constraints have been addressed above. As for technical constraints, our team members are already self-studying to familiarize themselves with React and MongoDB so that by the time we start coding, we would all be on the same track. We have also established a clear communication channel through WhatsApp so we can easily reach out to each other with any forms of questions. Through our regular meetings, we also synchronize our efforts and address any teamwork challenges, helping us successfully collaborate and work together as a group. Testing constraints will be managed by simulating environments where we simply have regular users test the platform as “doctors” or “patients”. We plan to do this by releasing a “beta” version of the web application where we will ask our friends and acquaintances to use the platform and report any bugs or usability challenges they may have faced.

6. 2. References:

Australian Government Department of Health and Aged Care. (2022, November 15). About prescriptions. https://www.health.gov.au/topics/medicines/about-prescriptions

Beta Test. (n.d.). Www.productplan.com. Retrieved February 29, 2024, from https://www.productplan.com/glossary/beta-test

CDC (Centers for Disease Control and Prevention). (2022, June 27). Health Insurance Portability and accountability act of 1996 (HIPAA). Centers for Disease Control and Prevention. https://www.cdc.gov/phlp/publications/topic/hipaa.html

CMS, (Centers for Medicare & Medicaid Service). (2023). CMS interoperability and Patient Access Final Rule (CMS-9115-F). CMS.gov. https://www.cms.gov/priorities/key-initiatives/burden-reduction/policies-and-regulations/cms-interoperability-and-patient-access-final-rule-cms-9115-f

Inflectra. (2023, November 18). What is API Testing? https://www.inflectra.com/Ideas/Topic/API-Testing.aspx

Macdonald, E. (2023, June 6). What is a Private doctor. WriteUpp Blog. https://www.writeupp.com/blog/glossary/private-doctor

OCR, Office for Civil Rights (2021, June 28). Hitech Act Enforcement Interim Final Rule. HHS.gov. https://www.hhs.gov/hipaa/for-professionals/special-topics/hitech-act-enforcement-interim-final-rule/index.html

Poston, L. (2023, May 31). Understanding Medical Consultations. TelegraMD. https://telegramd.com/what-is-a-medical-consultation/

priaid-eHealth. (2018, December 31). priaid-eHealth/symptomchecker. GitHub. https://github.com/priaid-eHealth/symptomchecker

What are clinical records. (n.d.). Www.medicalprotection.org. https://www.medicalprotection.org/ireland/booklets/medical-records-in-ireland-an-mps-guide/what-are-clinical-records

What is Functional Testing? Types & Examples. (n.d.). OpenText. https://www.opentext.com/what-is/functional-testing

What is Load Testing? How it works. (n.d.). OpenText. https://www.opentext.com/what-is/load-testing

Weber, D. (2019, February 11). *American Association for Physician Leadership | AAPL*. American Association for Physician Leadership - Inspiring Change. Together. https://www.physicianleaders.org/articles/how-many-patients-can-primary-care-physician-treat

World Wide Web Consortium, (W3C). (2020). Web content accessibility guidelines (WCAG) 2.2. W3C. https://www.w3.org/TR/WCAG22/#abstract

ZegoCloud SDK. (2022). ZegoCloud Documentation. https://www.zegocloud.com/docs