

Requirements Document

Electronic Medical Record

White Coat Clinic (Group 9)

April 11, 2023

Table of Contents

1 Introduction	4
1.1 Purpose	4
1.2 Project Scope	4
1.3 Glossary of Terms	4
1.4 References	5
1.5 Overview	5
2 Overall Description	6
2.1 Product Perspective	6
2.2 Product Features	6
2.3 User Classes and Characteristics	6
2.3.1 Ultimate Access	6
2.3.2 Partial Access	6
2.4 Operating Environment	7
2.5 Design and Implementation Constraints	7
2.6 Assumptions and dependencies	8
3 System Features	8
3.1 Appointment Management	8
3.1.1 Description and Priority	8
3.1.2 Functional Requirements	9
REQ-3.1.2.1: Appropriate users (doctors, nurses, MOAs) shall create appointments from the Schedule page	9
3.2 User Identity Validation	9
3.2.1 Description and Priority	9
3.2.2 Functional Requirements	9
3.3 Patient Profile	10
3.3.1 Description and Priority	10
3.3.2 Functional Requirements	10
3.4 Ordering Lab Tests Portal	11
3.4.1 Functional Requirements	11
3.4.2 Functional Requirements	11
4 External Interface Requirements	11
4.1 User Interfaces	11
4.2 Hardware Interfaces	11
4.3 Software Interfaces	12
4.4 Communications Interfaces	12
5 Other Non Functional Requirements	12

5.1 Performance Requirements	12
5.2 Safety Requirements	13
5.3 Security Requirements	13
5.4 Software Quality Attributes	13
6 Other Requirements	13
Appendix: Issues List	13

Revision History

Name	Date	Reason for Changes	Version
RD 1.0	Feb. 10, 2023	Initial version	1
RD 1.1	Mar. 4, 2023	Requested changes are added	2
RD 1.2	April 4 2023	Second round of changes are added	3

1 Introduction

1.1 Purpose

Medical professionals access patient records on the daily in order to provide the best service to the people they are taking care of. Although the majority of health services have moved on from paper records and files and started utilizing software systems, the system that Team Hungree uses, the Oscar EMR software, is not up to par. With this project, White Coat Clinic would be providing Team Hungree with a brand new system which they can easily navigate, trust in terms of reliability and information security and use to obtain relevant information from integration partners. These will be the three main tenets White Coat Clinic will focus on with this project.

1.2 Project Scope

We will be creating an EMR system that replaces the OSCAR EMR system. The purpose of our software, like OSCAR EMR, is to provide a way for medical professionals to view a patient's medical history, prescriptions, current conditions, and lab work.

Creating an alternative to OSCAR EMR will make it easier for healthcare workers to view and submit vital information about patients that may be vital to ensuring their proper care. The updated user interface will reduce frustration and user error, and the new system architecture will be able to handle more users at a time and support faster global updates.

As this software is meant to be a total replacement for OSCAR EMR, it will include all features of OSCAR that Team Hungree uses. Updating to a newer, more functional EMR system will not only benefit the healthcare workers of Team Hungree, but will also help ensure that patient care remains top-notch.

1.3 Glossary of Terms

Term	Definition
EHR	Electronic Health Record
UI	User Interface
OSCAR EMR	Open Source Clinical Application and Resource
E-pres	Electronic prescription
MOA	Medical Office Assistant
Pentesting	Also known as penetration testing, this is the permitted hacking of an application with the purpose of finding vulnerabilities before

	malicious actors do
--	---------------------

1.4 References

- [1] OWASP, <https://owasp.org/>, [accessed at Feb, 2023]
- [2] Request for Proposal
- [3] <https://dfr.oregon.gov/drugtransparency/data/pages/top-25-most-prescribed.aspx>
- [4] <https://www.thebody.com/article/understanding-lab-results-7>

1.5 Overview

Section 2.0: The Overall Description overview: This section provides an overview of the product perspective of the White Coat Clinic's improved EHR web application. It includes the context and origin of the product as a replacement for an existing system, the main and new features that distinguish it from similar applications, the user classes and characteristics, the operating environment, design and implementation constraints, and assumptions and dependencies.

Section 3.0: The System Features overview: This section outlines the functional requirements of the EHR web application, covering four areas: Appointment Management, User Identity Validation, Patient Profile, and Ordering Lab Test Portal. It provides a comprehensive overview of the features and capabilities available in the application.

Section 4.0: The External Interface Requirements overview: This section defines the requirements for the user interface, hardware interface, software interface, and communication interfaces. It outlines the requirements for each interface and how they interact with the EHR system.

Section 5.0: Other Non-Functional Requirements overview: This section defines the non-functional requirements of the system, including performance requirements, safety requirements, security requirements, and software quality attributes. It provides an overview of the non-functional requirements that must be met to ensure the system is secure and performs as expected.

Appendix: Section 6.0: Appendix: This section includes a list of requirements that have yet to be defined, as well as requirements that have been deleted. It provides a detailed description of each requirement and any related user information.

2 Overall Description

2.1 Product Perspective

The context and origin of the product being specified in this RD is a follow-up system to an older on-premise system. The older system had an overwhelming UI design with too many colours and element choices and outdated security measures for sensitive data. Additionally, the backend was less reliable and hosted on-premises. this new follow-up system is being developed to address these limitations and improve the overall user experience and security of the system

2.2 Product Features

The following list provides a high-level overview of major user features that will be included in our OSCAR EMR replacement software:

- Securely login to the software upon opening, and be required to login again after being inactive for a period of time.
- View an overview of a patient's medical history, which will include the most critical information at the top of the file.
- View all available details of a patient's medical history.
- Update and add new medical information for a patient.
- Create, modify, or view details of scheduled appointments and lab work for a patient.
- Add, remove, or edit a patient's prescriptions, including dosage, date prescribed, and length of prescription.
- Access updated patient information at most 3 hours after it has been added to the system, and view the time that information was last updated.

2.3 User Classes and Characteristics

For this application, there are two important user classes. Both classes will use this system equally as frequently but one class of users will have less access than the other due to the Principle of Least Privilege.

2.3.1 Ultimate Access

The first and most important is the Ultimate Access class. This class will have access to all functionalities as they are entrusted with providing full care to the patients and need all of the available information on the system in order to be able to offer the best service. The users within this class will be clinic and hospital doctors and nurses.

2.3.2 Partial Access

This user class is not granted as much access because they only need specific kinds of information. Medical Office Assistants will be granted access to the appointment scheduling

module and basic appointment information (no details). Care aids will be granted access to view prescriptions and medical condition information. Lab Technicians will be allowed to upload test results to the system.

2.4 Operating Environment

Hardware Platform: The hardware platform of the White Coat Clinic EHR system should include a server and a set of client machines. The server should be able to handle multiple concurrent users and should be equipped with an adequate amount of RAM and storage capacity, as well as an efficient processor. For the client machines, a range of devices should be supported, such as desktop computers, laptops, tablets and mobile phones.

Operating System and Versions: The server should run a Linux operating system, such as Debian and Ubuntu, in order to ensure high performance and stability. The client machines should be able to support the latest versions of Windows, MacOS, iOS and Android operating systems.

Software Components and Applications: The EHR system should be built on a web-based platform, such as Apache or Nginx, and should be written in a programming language such as Python, Node.js, or Java. The system should also include a database, such as MySQL, MongoDB, or PostgreSQL, to store patient and administrative data. Additionally, the system should be able to communicate with other applications, such as lab tests, prescription and appointment systems. The system should also be able to integrate with popular healthcare tools, such as e-pres.

2.5 Design and Implementation Constraints

The development team will be limited in using technologies, tools, and databases. The team will also need to adhere to certain corporate policies such as following the HIPAA regulations to ensure the secure storage and transmission of patient data. Additionally, the team will need to ensure the system is compliant with the laws of Canada and any other relevant jurisdictions. The team will also need to ensure the system is able to be deployed on-premise and allows for the secure storage and transmission of patient data. This will include encrypting patient data at rest and in transit. The team will also be responsible for ensuring the system is able to store data in a secure manner and that backup data is stored in a secure server located in Canada. The team will also need to ensure the system is able to easily export and print forms and records for patients in a timely manner. Additionally, the team will need to ensure the system is able to easily update globally within 3 hours of new information being added. Finally, the team will need to ensure the system is able to quickly and easily access patient information, schedule new appointments, order lab tests, and update patient information accordingly. Additionally, the team will need to ensure the system is able to validate user identities, such as a log-in system with usernames and passwords.

2.6 Assumptions and dependencies

Assumed Factors:

- We are assuming that the application will be developed for web browsers and mobile devices.
- We are assuming that the application will need to be compatible with other third-party software and applications, such as laboratory systems, to ensure data accuracy and reliability.
- We are assuming that the application will be hosted on a secure server with adequate storage and processing power to handle the increased amount of data.
- We are assuming that the user interface will be intuitive and easy to use.
- We are assuming that the application will be able to connect to other third-party systems to access and/or share data.

Dependencies:

- The project has a dependency on third-party software components that will be used in the application.
- The project has a dependency on the availability of secure server hosting.
- The project has a dependency on the user interface being intuitive and easy to use.
- The project has a dependency on the application being compatible with other third-party systems and applications.

3 System Features

3.1 Appointment Management

3.1.1 Description and Priority

This feature will allow medical assistants and/or doctors to be able to schedule appointments with patients. This is of high priority as this is vital to the management of a clinic or hospital.

3.1.2 Functional Requirements

REQ-3.1.2.1: Appropriate users (doctors, nurses, Medical Office Assistants (MOAs)) shall create appointments from the Schedule page

REQ-3.1.2.2: Appropriate users (doctors, nurses, MOAs) shall edit appointments from the Schedule page

REQ-3.1.2.3: Appropriate users (doctors, nurses, MOAs) shall delete appointments from the Schedule page

REQ-3.1.2.4: Appropriate users (doctors, nurses, MOAs, care aids) shall search up appointments by PHN from the Schedule page

REQ-3.1.2.5: Appropriate users (doctors, nurses, MOAs) shall search up appointments by patient name from the Schedule page

REQ-3.1.2.6: Appropriate users (doctors, nurses, MOAs) shall add notes to appointments

3.2 User Identity Validation

3.2.1 Description and Priority

The task of validating user identity is one of the highest priorities for the White Coat Clinic EHR system. This task should employ a secure logging-in method, such as tokens, two-factor authentication, or biometric authentication. This method should be easy to use and understand and should provide a secure layer of protection for the patient data stored in the system. Furthermore, the system should be designed with an additional layer of security, such as user access management, to ensure that only authorized users can access the system.

3.2.2 Functional Requirements

REQ-3.2.2.1 Log In (Tokens): The system should provide users with a secure login method that utilizes tokens for authentication. The token should be unique for each user and consist of a combination of letters and numbers that are randomly generated. The token should be valid for a period of time, after which it should expire and require the user to generate a new token to access the system. In collaboration with REQ-3.2.2.1, tokens are generated only when correct credentials are entered and the user provides the accurate one-time code sent to either their email or phone number. Where the code is sent depends on the user's preferences.

REQ-3.2.2.2 Auto Log Out: The system should provide users with an auto-log-out option after 60 minutes of inactivity. This feature should be enabled by default and provide users with the ability to disable it if desired. Additionally, the system should provide users with the ability to manually log out of the system at any time.

REQ-3.2.2.3 Two-Factor Authentication: The system shall provide two-factor authentication for user logins. This shall include the use of a username and password combination with a unique one-time code sent to a user's registered email address or mobile device.

REQ-3.2.2.4 Biometric Authentication: The system shall provide the option of biometric authentication for user logins. This shall include the use of fingerprint and/or facial recognition.

REQ-3.2.2.5 User Access Management: The system shall provide user access management to ensure that only authorized users can access the system. This shall include the ability to set user permissions and roles.

REQ-3.2.2.6 Password Protection: The system shall provide a secure password protection system to ensure that user passwords remain confidential. This shall include the use of strong encryption

and the ability to reset passwords.

3.3 Patient Profile

3.3.1 Description and Priority

The profiles cover basic information besides the medical history of the patient. The task of a high priority.

3.3.2 Functional Requirements

REQ-3.3.2.1 Patients Basic Information and Contact: The system has to support basic profile information about the patient like the name, contact options, address and how long have they been going to the clinic.

REQ-3.3.2.2 Medical History: The profile should cover past appointments in addition to medical history. That includes past prescriptions, blood, organ donations and surgeries.

REQ-3.3.2.3 Looking Up and Adding New Appointments: The system gives the users with the right credentials to add new appointments for the patient's schedule.

3.4 Ordering Lab Tests Portal

3.4.1 Functional Requirements

The system has to support a smooth way for the users with the right credentials to order lab tests from a supported third-party establishment or from inside the clinic if possible. The task is of high priority.

3.4.2 Functional Requirements

REQ-3.4.2.1 Sending Lab Tests requests to Third Party: The medical practitioner can order lab tests from the establishment of choice among the labs connected with the system.

REQ-3.4.2.2 Automatically registering the test results: When the lab of choice finalizes the test report, the report is added to the patient history. The process has to be automated.

4 External Interface Requirements

4.1 User Interfaces

The initial interface of the web application for the EHR system will be a login page where users can insert their credentials. The intended users for the application are doctors, nurses, lab techs, and caretakers. Once completed, doctors and nurses will be shown a list of patients that they have attended and a section for each patient that consists of the patient's medical history, appointments, current medications, vaccination dates, and allergies. Doctors and nurses can schedule appointments for their patients using the schedule appointment section where date and time can be inserted. Likewise, lab techs will be shown a list of patients, ordered by the date of test, to update test results for the patients using the edit and save button. Similarly, caretakers will be shown a list of patients that they are caring for as well as their patients current prescriptions. In addition, there will exist a help section where users can access frequently asked questions by other users.

4.2 Hardware Interfaces

The EHR system will be a web application that can be accessed in any smartphone, tablet, iPad, or computer. All functionalities of the web application can be used using a touch input or the mouse in case users are using a non-touchscreen computer.

4.3 Software Interfaces

The entire web application will be connected to the database to retain the information that is being updated by the doctors, nurses, and lab techs. Whenever a change is made by the users in the patient information, the change will go through the database and update it to show the most recent information about the patient on the web. For example, when an appointment is booked for the patient on the web, the date and time of the appointment will be stored in the database as a record and shown back on the web. Likewise, when the test result for the patient is updated by the lab techs on the web, the web will communicate with the database to override the current test report with the most updated version and then show the result back on the web application.

4.4 Communications Interfaces

When doctors and nurses update the patients' prescriptions, the EHR system will communicate directly with the third-party service to generate electronic prescriptions. In addition, when lab techs update the patient lab report, the service will generate electronic lab reports that could also be printed.

5 Other Non-Functional Requirements

5.1 Performance Requirements

The following section describes performance requirements for the EMR software and explains why these performance requirements are necessary.

- Global updates at least every 3 hrs - this is important because medical professionals need to have access to updated patient information
- A minimum of 99% uptime - medical professionals need access to patient information around the clock
- Lag on button clicks and similar operations should be < 50ms - the software should not lag to maximize time efficiency

5.2 Safety Requirements

The following section describes safety requirements for the EMR software.

- End-to-end encrypted cloud storage of patient data - patient medical records should be securely stored
- Secure authentication process - unauthorized parties should not access private patient data
- Encrypted backups of data are created once every 3 hours - cannot risk losing medical records
- Servers have DOS defense - the software and patient data should be accessible at all times

5.3 Security Requirements

The following section describes security requirements for the EMR software.

- Every piece of code pushed into production will undergo static code analysis with SonarCloud in order to catch vulnerabilities before they are released
- The login functionality will require multi-factor authentication via email or text message
- Every unsuccessful login will be logged and assessed by the security team
- The user is allowed 3 incorrect attempts. If they reach that limit, they will not be allowed to log in until they reset their password. They can request to change their password by clicking the button that will pop up underneath the credential fields and they shall provide the email address that is associated with their account
- Sensitive data will be in a separate database from other data such as appointment schedules
- Sensitive data will be encrypted in transit and rest

- Trusted and current encryption algorithms will be used such as the Advanced Encryption Standard
- User profiles ensure access control
- Servers will be hosted within Canada
- The application will undergo pentesting from a third party every 6 months in order to ensure vulnerabilities are known
- Internal pentesting will occur every month using OWASP ZAP [1]
- Vulnerabilities will be addressed upon discovery by the security team who will assess their priority. High priority and high severity vulnerabilities will be resolved first
- Text fields will be sanitized and will only allow white-listed input patterns. The patterns are decided on by the security team in order to rule out and scripts and queries that may be inserted
- HttpOnly Cookies will be used to prevent XSS
- Application updates will be automated and will not interfere with the system functionality while it is being updated

5.4 Software Quality Attributes

The following section describes additional attributes that the software is required to have. These attributes are important to either the customers or the developers.

The software should be easily adaptable so that healthcare professionals can tailor their user experience to suit how they prefer to run their clinic without extensive intervention from developers. Adaptability should be prioritized over ease of learning, because the software is intended to function similarly to other software that is currently in use/on the market.

Additionally, the software should not require frequent updates to operate properly, and outdated versions of the software should run smoothly. This ensures that the vital medical software will be usable for the foreseeable future. Longevity should be prioritized over being up-to-date.

Finally, developers do need to make updates to the software, it should be easy to modify and expand. The program should not be made up of “legacy code” that is difficult to update and ages quickly. This will keep development and maintenance costs low, and ensure that the program will last a long time and remain working properly.

6 Other Requirements

On April 04 2023, we have received 4 changes from the clients and they were as follows:

1. Clean the interface to make it more organized
2. By default, make the sidebar visible even when the hamburger menu is not clicked.
3. Make the schedule editable.
4. Fix the bug where navigation to medical history through the button on the home page was not working.

The changes received from our clients were discussed by our team and came to the following conclusion.

1. Our team believed that the first change of cleaning the interface did not need to be changed since it seemed to us that the interface was as clean as possible. This change was feasible but because of the opinion and experience of using other medical software help us conclude that implementing this change was not essential.
2. In addition, the second change request by our clients was disregarded because after discussion with the team, it was concluded that in most softwares the sidebar is never visible since it makes the interface more cluttered.
3. Furthermore, the change requested to make the schedule editable was very essential to our application but we were not able to perform this change within the given timeframe so it seemed that it was not a feasible change given the timeframe and tough exam schedules for all the teammates.
4. Our team was able to fix the bug mentioned by our clients where the navigation for the medical history page from the home page was not working. This bug was fixed since it was feasible and essential.

Appendix: Issues List

The following is a list of pending issues that have not yet been resolved:

- Must work later on the third change requested by our clients where they asked us to make the schedule editable.