Hazard Analysis
SFWRENG 4G06A

Team #25, RapidCare
Pranav Kalsi
Gurleen Rahi
Inreet Kaur
Moamen Ahmed

Table 1: Revision History

Date	**Developer(s)**	**Change**
18-10-2024	Inreet Kaur	FMEA Table, Critical Assumptions
18-10-2024	Moamen Ahmed	FMEA Table, Safety + Security Reqs.
20-10-2024	Gurleen Rahi	FMEA Table, Safety + Security Reqs.
20-10-2024	Inreet Kaur	System Composition
24-10-2024	Inreet Kaur	Intro, Scope
24-10-2024	Pranav Kalsi	FMEA Table, Safety + Security Reqs, Roadmap, Document Compilation Fix..
25-10-2024	Gurleen Rahi	Reflection.

Contents

1 Introduction

The purpose of this document is to provide a comprehensive hazard analysis for RapidCare, a software application that aims to streamline the healthcare documentation process. According to Nancy Leveson, hazard is a property/condition within the system and its environment that can cause harm or result in loss [1]. To ensure the safety of the system as well as the user, it is critical to identify and mitigate potential hazards.

For the purposes of this document, we will use the Failure Modes and Effect Analysis (FMEA) method for hazard analysis. This document will provide an overview of the scope and purpose of hazard analysis, system boundaries and components, critical assumptions about the system and its environment, and an FMEA table listing the causes and effects of failure along with recommended actions. The document will also list any additional safety and security requirements identified as a result of hazard analysis and a roadmap for implementation.

2 Scope and Purpose of Hazard Analysis

Hazards can arise from various sources such as user input, security issues, system failure or other external factors where the system is deployed. The scope of this document is a hazard within the various system components as well as the environment in which the system will operate.

The purpose of hazard analysis is to proactively identify all potential hazards, the effects and causes of the failure and to develop appropriate mitigation strategies. Since the system will operate in a healthcare setting, it is critical to identify potential hazards. This will ensure the safety, reliability, and security of the system. Moreover, it is essential to protect sensitive information, delays in treatment, other medical errors, and the safety of the system and user.

3 System Boundaries and Components

To identify potential hazards, we first define the system boundaries and break it down into its major components:

- **User Interface:** The user interface is the point of interaction between the users and the system. It is responsible for displaying outputs from the system, such as patient data, medication suggestions, diagnosis predictions etc. The UI plays a crucial role in ensuring a user-friendly and intuitive experience for the users.

  Potential Hazards:

  - Incorrect user input
  - Inadequate feedback when errors occur

  - Incorrect data displayed to the user

- **Data Layer:** The data layer in the system is responsible for managing and processing all data related to patient records, healthcare professionals, health networks, and predictive models for medication and diagnosis. It is divided into the following databases:

  - DB1: Patient, Healthcare Professional, and Network Database: This database stores patient records, healthcare professionals, and healthcare network profiles. This component is responsible for storing, retrieving, updating, and deleting data.
  - DB2: Diagnosis Prediction Database: This database stores the data used by the diagnosis prediction component to suggest potential diagnoses based on analysis of the transcribed data.
  - DB3: Medication Prediction Database: This database holds the data used by the medical prediction component to suggest appropriate medications based on the identified or accepted diagnosis.

  Potential Hazards:

  - Accidental deletion of database entries or the entire database
  - Creation of duplicate records
  - Security breaches
  - Database crashes

- **API Module (OAuth):** This component securely connects different parts of the system using OAuth authentication. It verifies the identity of users and services before allowing them to access protected resources. The module manages the OAuth process, including user authentication, issuing access tokens, and validating these tokens for each request.

  Potential Hazards:

  - Failed connection between components

- **User Authentication:** This component verifies and validates user credentials for secure system access. It implements multi-factor authentication and integrates with the system's cryptographic infrastructure. The component also handles password security and implements measures against unauthorized access attempts.

  Potential Hazards:

  - User cannot log in to the system

- **Account Management:** This component oversees user account lifecycles within the system. It handles account creation, profile updates, and account deletion, ensuring data integrity throughout these processes.

  Potential Hazards:

- Account cannot be created, updated, or deleted

- **Report Generating Module:** The report generating module is responsible for generating accurate reports using the transcribed data from transcription module. The important aspects of medical data (such as the symptoms, illness history, etc.) is extracted from the data which is then compiled into the report using this module.

  Potential Hazards:

  - Data inaccuracy in reports

- **Transcription Module:** The transcription module is responsible for converting audio data from the conversation to written text. The converted written text is used thereafter used by the report generation module to generate the report of the patient.

  Potential Hazards:

  - Incorrect transcription from audio to text

- **Prediction Modules** This module is responsible for all of the predictions that will occur based on the medical notes.

  - PR1: Diagnosis Prediction: The prediction module is responsible for using the clinical notes to provide some diagnosis predictions. The goal of this module is to associate symptoms with common diagnoses. This allows for a lateral view of possible patient diagnoses.
  - PR2: Medicine Prediction: Once a diagnosis is selected through the model or manual entry, the corresponding frequently used medication will also be provided.

  Potential Hazards:

  - Wrong diagnosis family is predicted
  - Processing non-medical related inputs

4 Critical Assumptions

The following assumptions are made regarding both the software and hardware components of the system:

- **Stable Network Connection:** It is assumed that the network connection between the client and server will be stable. If the connection is unstable, it could cause interruptions to the process, which results in significant issues in the system's performance.

- **Reliable Hardware:** It is assumed that there won't be any major hardware failures. Although hardware problems are rare, they could severely affect system availability and accuracy, especially in critical healthcare environment.
- **Well Intentioned Inputs:** We will assume that the user will enter input with good intention and will not be attempting to break the system or overload the system with the requests.
- **User Expertise:** We will assume that users possess the domain expertise, such as medical terminologies, which is required to effecitvely interact with the system and validating the report generated by the software.

5 Failure Mode and Effect Analysis

The following FMEA table lists all potential hazards related to identified system components above along with the failure mode, effects and causes of failure, detection, and recommended actions. It also provides tracibility to associated requirements.

Comp.	**Design Function**	**Failure mode**	**Effects of failure**	**Causes of failure**	**Detection**	**Recommended action**	**Req.**	**Ref.**
User Interface	Allow user to provide data through input for processing	System accepts incorrect data	Incorrect data stored in the database; Incorrect diagnosis; Inaccurate data may lead to medical errors	Lack of input validation; Insufficient feedback mechanism	User reports; Record validation checks	Improve UI design for discoverability and use appropriate signifiers for various data fields. Display soft feedback to guide user input i.e. implement input masks, field-level validation, and page-level validation to prevent the system from saving any invalid data. Implement constraints on input data fields.	NFR1; NFR2; IR2	H1.1

	Display error messages and provide feedback	System provides inadequate feedback when errors occur	Users are unaware of the current system state; Unresolved issues; Inaccurate data stored in a database	Insufficient feedback mechanism	Error logs; User reports; Record validation checks	Provide clear and actionable error messages when an error occurs. Use language familiar to the user for easy interpretation. Provide steps to recover from the error state.	NFR1; NFR2; IR2	H1.2
	Display correct data to the user	System displays incorrect data to the user	Incorrect medical decisions; Compromise patient safety	Data processing errors; System bugs	User reports; Error logs; Record validation checks	Ensure user input is accurately interpreted and stored by the system. Add data verification steps to ensure the system retrieves the correct data to display.	IR4	H1.3

Data Layer	Manage and store data in a secure manner	Accidental deletion of database entries or the entire database	Permanent loss of critical data	User errors; Lack of validation checks	User reports; Failure to retrieve or access a data instance or database; Real-time database monitoring; Automated alerts for data discrepancies	Display appropriate feedback before confirming the deletion. Implement role-based access control for deletion action. Implement automatic data backup and recovery system.	FR5; FR9; FR2	H2.1
		Creation of duplicate records	Incorrect output displayed to the user; Medical errors	Lack of validation on user input	Record validation checks; Real-time database monitoring; Automated alerts for data discrepancies	Implement validation checks for user input. Implement validation checks before storing a new entry. Regular data integrity checks.	IR4	H2.2

		Security breaches	Unauthorized access to sensitive data; Regulatory and compliance issues	Improper authentication and encryption	Security audits; Access logs	Implement strong authentication protocols. Encrypt sensitive data using standard encryption protocols. Ensure compliance with PIPEDA and regulatory standards.	NFR6; NFR8; FR7	H2.3
	Retrieve and store data in real-time.	Database crashes	Inability to access stored data; Inability to store new data	Server overload; System failure	Error messages; Monitoring system performance	Implement failover systems. Implement automatic backups. Implement scalable server infrastructure.	NFR4; NFR5	H2.4
General	Provide continuous access to the system	App closes unexpectedly	Unsaved progress is lost; Delayed medical access to patients	Loss of power or internet; Software failure	User reports; System logs	Implement automatic data backups and recovery system.	NFR4	H3

API Module (OAuth)	Securely connect components with OAuth	Failed connection between components	Inability to authenticate users; Disrupting services	OAuth misconfiguration	Monitor connection status; Failed authentication attempts	Monitor network status. Verify OAuth configuration. Implement regular security audits. Implement fallback systems for component failures.	AC??	H4
User Authentication	Verify user credentials	User cannot log in to the system	User cannot access any system data or functions	Invalid credentials; Database failure	Failed login attempts trigger security alerts	Reset credentials. Verify database connectivity	IR1	H5
Account Management	Manage user accounts (create, update, delete)	Account cannot be created, updated, or deleted	User unable to register, update info, or remove account	Database failure; Validation errors	Log account creation, update, and deletion attempts	Check database integrity. Implement Validation checks for inputs.	AC2	H6

Report Generating Module	Generate medical reports from transcribed data	System produces inaccurate reports; Inputs wrong data in various data fields; Inconsistent formatting of the data.	Inaccurate diagnosis and medicine predictions; Inaccurate data stored in databases; Difficulty to interpret data due to formatting issues.	Misclassification of the transcribed data	User reports; Error logs; Real-time database monitoring	Implement validation checks when the system classifies data for different data fields. Validate data for formatting issues. Prompt the user to validate the data. Allow the user to edit the data manually.	FR11; NFR3; IR6	H7
Transcription Module	Convert audio data from the conversation to written text	System provides incorrect transcription	Inaccurate diagnosis; Medical errors	Background noise disruption; Misinterpretation of the pronounced words	User reports	Use models with high transcription accuracy. Prompt user to review the transcribed data. Allow user to validate and edit transcribed data.	FR11; NFR3; IR7	H8
Prediction Module	Use medical notes to predict diagnosis and medicine required.	Incorrect due to biased prediction	Healthcare professional may be misled.	Poorly trained model; Biased data.	Use validation and cross-validation to evaluate the models.	Use healthcare professional evaluation and train systematically.	IR3; NFR6	H9.1

	Use medical notes to predict diagnosis and medicine required	Processing non-medical related inputs	Healthcare professional may be misled.	Inputs for model are not appro-priately filtered.	Add filters to the model pipeline to ensure data inputted is useful data.	Add filters to check for quantitative inputs.	IR5; NFR6	H9.2

Table 2: **Failure Mode and Effect Analysis of the System**

6 Safety and Security Requirements

6.1 Access Requirements

AC1: The API Module (OAuth) must allow only authenticated users access to system resources. Failed authentication attempts must be logged, and repeated failed login attempts from a single user must trigger an automatic security response, temporarily locking the account and notifying the security team.
Rationale: Authentication is fundamental to system security as it ensures that only verified users can access sensitive resources. Logging failed attempts is crucial for detecting potential security breaches and maintaining an audit trail for security investigations. Implementing an automatic response to repeated failed login attempts helps mitigate the risk of brute-force attacks and ensures proactive security measures.

AC2: Only authorized personnel can create, update, or delete user accounts. Unauthorized actions should be blocked, logged, and reviewed to detect potential threats.
Rationale: Restricting user management operations to authorized personnel prevents unauthorized account creation and modification, which could lead to security breaches. Logging and reviewing blocked attempts provide accountability, help identify potential internal threats, and ensure quick responses to compromised credentials or suspicious activities.

6.2 Integrity Requirements

IR1: User credentials must remain intact during authentication. Failed login attempts should not affect the system's functionality or stored data.
Rationale: Maintaining the integrity of user credentials during authentication is essential to prevent unauthorized access and data corruption. The system must remain stable and secure regardless of authentication failures to ensure continuous service availability and protect stored credentials from potential tampering or corruption.

IR2: The system should provide real-time error detection based on validation checks and provide feedback to users.
Rationale: To prevent user errors and incorrect output, it is vital to check the integrity of user input. Moreover, in the event of an error, the system should communicate its current state, how the input has been interpreted, and any related errors to the user.

IR3: The system should provide confidence score on any prediction. Additionally, after each model training session validation accuracy score must be outputted.
Rationale: To understand the bias in any data and in the model. Using validation scores for training and test sets the model can be evaluated

to detect underfitting and overfitting. Additionally, adding model confidence scores will allow the healthcare professionals to see how confident the model is with the scores.

IR4: The system should provide duplicate record detection for the record in various databases of the system.
Rationale: To prevent confusion and medical errors resulting from duplicate entries, the system should validate and flag potential duplicate records before they are created.

IR5: The system should filter out irrelevant parameters before feeding the data into the model.
Rationale: Medical data may contain a lot of side notes and information which is not critical for diagnosis and medicine prediction. That data should be filters out and not be passed into the model as an input. This is because the model will be expecting only a certain set of parameters. Any irrelevant parameters must be left out.

IR6: The system should have proper validation when classifying the data for report generation.
Rationale: It is essential to generate correct reports with consistent formatting to avoid any future medical and diagnosis errors.

IR7: The system should be able to filter the background noise to produce accuracy transcribed data.
Rationale: It is essential to produce accurate data to avoid any inaccurate records, potential medical errors and incorrect diagnosis.

6.3 Privacy Requirements

Covered in SRS

6.4 Audit Requirements

Covered in SRS

6.5 Immunity Requirements

Covered in SRS

7 Roadmap

After the hazard analysis we identified a lot of new safety and security requirements. In terms of the scope of the capstone project, in terms of these requirements the team will aim to deliver all of the mentioned requirements in this document including the mentioned access and integrity requirements. To

add in our feedback we will first prioritize critical components and incrementally integrate changes by assigning the tickets to the respective developers.

IR5 and IR3 may be simplified. For the input check, it may be reduced to a simple filter, and the validation scores maybe to simplified to just attempting to mitigate model bias. The roadmap will re-assessed towards the end of the project to see if features can be augmented to complete the requirements.

8 References

1 N. G. Leveson and J. P. Thomas, "STPA handbook (MIT-Stamp-001)," STPA Handbook, https://psas.scripts.mit.edu/home/get_file.php?name=STPA_handbook.pdf (accessed Oct. 9, 2024).

Appendix — Reflection

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

1. What went well while writing this deliverable?
   This document has let us build more on the critical hazards and mitigation strategies to overcome them. While going through the outline of this document, we were able to provide the system components as well as their failure modes of the system.

2. What pain points did you experience during this deliverable, and how did you resolve them?
   Every team project has challenges that must be solved in order to move forward successfully. We had to create a donation plan to guarantee smooth operations. To ensure that all of our plans are in line with the system, we need to plan the system components that complement our project. In order to contribute to the document and evaluate each other's work as effectively as possible, we also needed to set up a schedule.

3. Which of your listed risks had your team thought of before this deliverable, and which did you think of while doing this deliverable? For the latter ones (ones you thought of while doing the Hazard Analysis), how did they come about?
   Our team had some knowledge of the system's components and possible software hazards prior to this deliverable. However, we also came up with some safety requirements, including integrity requirements, while working on this deliverable. Integrity requirements have listed a lot of detection modules that are essential for authentication purposes.

4. Other than the risk of physical harm (some projects may not have any appreciable risks of this form), list at least 2 other types of risk in software products. Why are they important to consider?
   The other two categories of risks are security risks, like cyber-attacks, and functional risks, like performance problems. They are crucial since a malfunctioning system could cause the automation process to be delayed, which would defeat the goal of the project. Additionally, phishing and

hacking may result in the loss of patient data, which would affect data privacy.