Hazard Analysis
Software Engineering

Team 14, Reach
Aamina Hussain
David Moroniti
Anika Peer
Deep Raj
Alan Scott

Table 1: Revision History

Date	Developer(s)	Change
October 18, 2023	Aamina Hussain	Added sections 1, 2, and 4
October 19, 2023	David Moroniti	Added trial hazards + SR's
October 19, 2023	Alan Scott	Added additional trial hazards + SR's
October 20, 2023	David Moroniti	Add roadmap
October 20, 2023	Deep Raj	Added additional trial hazards + SR's
October 20, 2023	Anika Peer	Added System Boundaries and Components

Contents

1 Introduction

This document includes a hazard analysis for the web application REACH. REACH will allow users to find clinical trials or research studies they are eligible to participate in. It pulls in information about these studies from existing external public databases. This document will analyze and record any hazards to the system REACH. In this case, a hazard is a property of a system, together with the condition of the environment the system is in, which can cause harm or damage and results in a loss. This definition of hazard is from Nancy Leveson's work.

2 Scope and Purpose of Hazard Analysis

The scope and purpose of this hazard analysis is to identify any system hazards and which components they are related to. This includes analyzing the causes and effects of the hazard and the recommended actions to mitigate the hazard, as well as documenting the resulting safety and security requirements.

3 System Boundaries and Components

The system boundary for REACH consists of the following:

1. The web application itself with all of its front and back end components.
   - The database for user information with secure storage and retrieval.
   - The user authentication subsystem.
   - The trial searching and filtering subsystem.
   - The email generation subsystem.
   - The user notification subsystem.
2. External public databases that the web application pulls information from. While the team has no control over these databases, their data is used by REACH.
3. Devices that the web application is accessed from. This includes mobile devices, tablets, and computers.

As such, the system boundary includes the web application itself, the external public databases, and the devices that the web application is accessed from. Although the physical devices are not part of the system, they are included because, as the environment, they are the means of accessing the web application.

4 Critical Assumptions

One critical assumption that is being made is that the external API will not go down very often, and will not be down for a significant amount of time. The mitigation provided below for the external API crashing is only a temporary one, and would only work under this assumption. Another critical assumption that we are making is that the Google Maps API will not fail to a point that it will impact the user.

5 Failure Mode and Effect Analysis

Definition of Failure - A failure is considered as any event that causes the application to behave in a way that is not intended. For example, someone able to authorize themselves using an account that is not their own would be a failure, since that is an unintended access point into the application.

Table 2: FMEA Analysis

Component	Failure modes	Effects	Causes	Action	SR	Ref.
Trial Fetching/Matching	External API's unavailable	System is unable to search for trials	System failure on the API providers side, scheduled maintenance, and API access method changed	Keep an internal database of trials.	SR-1	HT-1
	Mismatch in trials being recommended	User attempts to sign up for ineligible trial	Not enough/invalid information entered by user	Display a warning/disclaimer with respect to signing up for trials. Display a confidence rating for each matched trial.	SR-2, SR-5	HT-2
	User eligible for "too many" trials	Too many emails being sent to user and it could make it more difficult for a user to find a trial they really like.	Not enough data entered by user.	Inform user if they haven't entered enough data to get a good search.	SR-3, SR-4, SR5	HT-3
Database	Database unavailable	System is unavailable to login users and retrieve users' user data for searching	System failure on hosting side, database maintenance, database accesses updated or cancelled	Constrain users to guest access.	SR-6	HT-4
	Unauthorized user accesses the database	User data, including medical data, is may be comprised. Potential legal consequences in case of data breach	Insuffient database security protocols, leak of access credentials	Bring database online, restrict database access. Notify users in the event of a data breach.	SR-7	HT-5
Login Authentication	User is unable to login	User is unable to access their profile and autofill search parameters.	User forgets password, password authentication is down, database is unavailable	User is given the option to change password if password is incorrect, otherwise the user can still use guest access.	FR-4, FR-5	HT-6
	Unauthorized user logs into a user's account.	Unauthorized user gains access to their medical data and can search on their behalf.	Insufficient password strength, login authentication bug.	Require strong passwords, inform users of login attempts from new locations.	NFR-15, SR-8	HT-7
Email Generation	Email is sent from a different email then the one the user would like to use.	Inconvenience to the user. Potentially disrupting communication between the user and the researcher.	User is signed in with an email that they do not want to use when contacting researchers.	User will have the ability to select what email they would like to use before sending the email. They should also be given the ability to modify the template.	FR-10	HT-8
General	User loses internet connection	User is unable to search for trials. User is unable to update personal data.	User's device lost connection to internet, or ISP outage.	Data is stored locally untill it is uploaded to the database. Application Displays a warning that it is unable to contact the internet.	SR-9, SR-10	HT-9
	Application Crashes	User may lose unsaved data.	Programming error, hardware failure, user device malfunction.	User is informed that unsaved data may have been lost when the application is launched again.	SR-11	HT-10

6 Safety and Security Requirements

SR-1: The system shall periodically store new trials into an internal database, and remove trials that are no longer active.
Rationale: In case of external API failure, there should be some redundancy. Keeping a small

“cache” of active trials can ensure the system is never completely down, due to an external failure.

SR-2: The system shall give users a “confidence rating” when matching trials.
Rationale: It will be nearly impossible for the system to match every single eligible trial perfectly, and the user should know this.

SR-3: The system shall enable the user to put a limit on the number of emails they can receive per day.
Rationale: Some users may only want 1 email per day, and some users may want 10 emails per day. Each user should be able to decide this.

SR-4: The system shall define a pre-set limit number of emails that will be sent to an individual each day.
Rationale: If a user doesn’t set a limit (whether on purpose or by accident), the system could not handle sending thousands of emails to each user every day. A limit for this reason, is necessary.

SR-5: The system shall inform a user if it is likely that they have not entered a sufficient amount of information to get accurate search results or narrow down the search in any way.
Rationale: Some users may not realize the importance of entering sufficient and accurate information. Additionally, some may forget.

SR-6: The system shall restrict users to guest access in the event of a database failure.
Rationale: If the database is down, users will be unable to login or retrieve data for searches. Constraining users to guest access ensures they can still use the system without issue.

SR-7: In the event of an unauthorized user accessing the database, the database should be taken offline and access should be restricted.
Rationale: If an authorized user accesses the database, further access should be limited to developers and maintainers to ensure no further damage can be done until the issue is resolved.

SR-8: The system shall inform the user when their account is accessed from a location for the first time.
Rationale: Informing the user of a new login location will allow them to verify whether an unauthorized user has accessed their account, allowing them to update their password and take further steps if needed.

SR-9: The system shall store all user data locally until it is uploaded to the database.
Rationale: If there is no internet connection data can be saved locally until internet access is available again.

SR-10: The system shall inform the user when the internet is unavailable.
Rationale: Informing the user when the internet is unavailable allows them to check why and

potentially address the issue.

SR-11: The system shall inform the user when the application was closed unexpectedly previously and that unsaved data may have been lost.
Rationale: Informing the user when the application previously closed unexpectedly allows them to check if any changes they made recently to their personal data were saved.

SR-12: The system shall enable the user to select what email they would like to use when sending the email template to a researcher **Rationale:** A user might be signed in to the app under an email that is not the one they would like to use for sending emails, so they should be able to decide what email is used.

7 Roadmap

Requirements implemented during capstone timeline:

- SR-3
- SR-4
- SR-5
- SR-6
- SR-9
- SR-10
- SR-11

The problems which could/would arise from not having the above requirements implemented within the system would likely arise very soon after users begin using the platform, which is why it is necessary for them to be implemented by the time the casptone comes to a close.

Requirements to be implemented post-capstone:

- SR-1
- SR-2
- SR-7
- SR-8

The problems which could/would arise from not having the above requirements implemented within the system would be unlikely to arise soon after users begin using the platform, which is why there is a bit of a buffer post-capstone to get them implemented. Additionally, some of these requirements would greatly increase the scope of the project, taking away from some of the key components that need to be implemented for the capstone project.