

Hazard Analysis Software Engineering

Team #22, TeleHealth Insights

Mitchell Weingust

Parisha Nizam

Promish Kandel

Jasmine Sun-Hu

Table 1: Revision History

Date	Developer(s)	Change
October 24 2024	Jasmine Sun-Hu	Added Sections 1,2,3
October 25 2024	Jasmine Sun-Hu	Added Drafts of Section 4, Reflection
...

Contents

1	Introduction	1
2	Scope and Purpose of Hazard Analysis	1
3	System Boundaries and Components	1
4	Critical Assumptions	2
5	Failure Mode and Effect Analysis	2
6	Safety and Security Requirements	3
7	Roadmap	3

1 Introduction

This document contains the hazard analysis for Telehealth Insights, a project for a website that will help parents administer language tests at home for bilingual children with speech difficulties. A hazard is defined as a property or condition in a system that when combined with a condition in the environment has the potential to harm or damage to the system. A hazard is not limited to safety, it can also be related to system security, user sensitivity and unexpected human or technology interactions. The purpose of this document is to identify any hazards to the project, and develop newsafety and security requirements from a Failure Mode and Effect Analysis.

2 Scope and Purpose of Hazard Analysis

The hazard analysis focuses on identifying, evaluating, and mitigating any hazards that could negatively impact the speech language assessment platform. This includes both technical and user interaction hazards in particular since the project will include handling sensitive patient data. The analysis will cover many aspects of the system such as data handling, software stability and user sensitivity.

The purpose of the hazard analysis is to identify any risks that could affect data privacy and security, system reliability, data collection accuracy, and compliance with relevant standards. A hazard analysis minimizes these risks, as the loss from unaddressed hazards could involve patient safety, data breaches, and legal or financial consequences for the assessment organization.

3 System Boundaries and Components

The system referred to throughout the document consists of several major components:

1. **User Interface (UI):** The front-end platform where users interact with the system. It allows users to navigate through the assessment, accepts inputs, and displays results.
2. **Backend Server:** The back-end platform handles data collection and processing, business logic, and communication between all system components.
 - **Authentication:** Manages the login and access control mechanisms.
3. **Assessment Recording**
 - **Video Recording Module:** Responsible for capturing and transmitting video data during an assessment session.
 - **Audio Recording Module:** Responsible for capturing and transmitting audio data during an assessment session.

4. Assessment Analysis

- **Video Analysis Model:** Processes and analyzes the video recording for disturbances and other behaviours against assessment instructions.
- **Audio Analysis Model:** Processes and analyzes the audio recording for disturbances and other behaviours against assessment instructions.

5. **Database:** A centralized storage for all assessment results, recordings, analysis results, user data, and any other data as necessary.

The system boundary for this project includes the entire platform, consisting of the user interface, backend server, assessment recording and analysis components, and the database. Components such as the user's device (e.g. computer or tablet used for the assessment), and any third-party services used are external to the system and outside the control of the capstone team and will not be directly considered in the hazard analysis. The connections to external components however will be considered within the system boundary and may be included in the hazard analysis.

4 Critical Assumptions

[These assumptions that are made about the software or system. You should minimize the number of assumptions that remove potential hazards. For instance, you could assume a part will never fail, but it is generally better to include this potential failure mode. —SS] This is just my brainstorm, feel free to change the formatting or add/remove constraints -Jasmine

1. Users have reliable internet access while using the web application.
2. The user's device is compatible with the web application and has the necessary capabilities and system requirements to run the assessment session.
3. Users will not share their password with anyone. (maybe remove? it's not about the software or the system directly -JS)
4. Any third-party services (e.g. cloud storage or hosting) used by the backend server are reliable and secure according to industry standards.
5. When a user is taking the assessment they will have a functional microphone and camera that meet the minimum requirements for recording assessment sessions.

5 Failure Mode and Effect Analysis

[Include your FMEA table here. This is the most important part of this document. —SS] [The safety requirements in the table do not have to have the prefix

SR. The most important thing is to show traceability to your SRS. You might trace to requirements you have already written, or you might need to add new requirements. —SS] [If no safety requirement can be devised, other mitigation strategies can be entered in the table, including strategies involving providing additional documentation, and/or test cases. —SS]

6 Safety and Security Requirements

[Newly discovered requirements. These should also be added to the SRS. (A rationale design process how and why to fake it.) —SS]

7 Roadmap

[Which safety requirements will be implemented as part of the capstone timeline? Which requirements will be implemented in the future? —SS]

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?

The structured format and clear guidelines from both the document outline and lecture slides for writing the hazard analysis, as well as the student examples from previous years were helpful and provided a clear idea of what was expected from us for this deliverable.

2. What pain points did you experience during this deliverable, and how did you resolve them?

One of our main challenges was defining the system components and boundaries clearly, especially in deciding which external components to include or exclude in the hazard analysis. The team considered including third-party services and user devices as part of the boundary, but after further discussion, we decided to limit our scope to only include components the team can have control over. The reasoning for this is because including external components would have added complexities out of our control, such as third-party security protocols and user device management, which could vary and introduce risks that are outside our scope to address. This approach still lets us focus on designing reliable interactions with external systems, but not need to address the external components themselves.

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?
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?

1. Data Security Risks: Risks such as unauthorized access, data breaches, and data leaks are critical because of the increased reliance on digital storage of sensitive information. A data breach could result in severe legal and financial repercussions, especially in healthcare applications where patient confidentiality is extremely important.
2. User Experience (UX) Risks: Things like confusing navigation or unresponsive interfaces can lead to user frustration, decreased usage, and even refusal to use the software. This is particularly important for non-technical users such as parents and children in a telehealth setting.