Hazard Analysis

Table 1: Revision History

Date	**Developer(s)**	**Change**
Date1	Name(s)	Description of changes
Date2	Name(s)	Description of changes
...	...	...

Contents

1 Introduction

The following document contains an overview of the hazards highlighted in the Large-Group Eye-Tracking Capstone Project. For the purposes of this document, a hazard is (based on the work of Nancy Leveson) defined as any aspect or property of this project which causes harm, damage or loss in the environment the system inhabits. This document identifies key hazards involved, and uses the Failure Modes and Effects Analysis (FMEA) method to analyze them and their respective impacts on the system.

2 Scope and Purpose of Hazard Analysis

The purpose of this Hazard Analysis is identifying system properties which may cause harm to stakeholders. In order to narrow the scope of this assessment, the following potential losses have been highlighted:

- Privacy: unauthorized access, re-identification, misuse of gaze data
- Participant discomfort
- Data inaccuracy: invalid findings and conclusions
- Loss of stakeholder value: instructors receiving inaccurate or unusable real-time data
- Disrupting live classroom activities

3 System Boundaries and Components

The proposed system is a learning platform that integrates large-group eye tracking with classroom activities, allowing instructors to view aggregated gaze information in real time and after class. To perform a meaningful hazard analysis, the system is divided into the following components: **Data Collection (Eye Tracking)**, **Supplementary Data (Student Survey)**, **Data Analysis (Mapping Gaze Data)**, and **Dashboards (Instructor Visualization Interfaces)**.

4 Critical Assumptions

This analysis assumes that the eye-tracking hardware functions reliably and that hardware-level failures are out of scope. It also assumes that the network and server infrastructure are stable and secure, with standard IT reliability already in place.

5 Failure Mode and Effect Analysis

Table 2: Failure Mode and Effect Analysis (FMEA)

Design Function	**Failure Modes**	**Effects of Failure**	**Causes of Failure**	**Detection & Recommended Action**	**SR**
Display UI	Poor UI design and Visualization Errors	Incorrect conclusions, misinterpreted data, distracting instructors in real-time dashboard	Unclear visualization, overcomplex layouts, poor labeling, Software bugs	Conduct usability testing; iterative UI design reviews; gather instructor feedback	Dashboard must undergo user-centered design testing before deployment
Display real-time dashboard feedback	Real-time dashboard lag	Delayed feedback, ineffective responses	Network latency, computation overload, inefficient refresh rate	Optimize data pipeline; limit visual update rate; provide timestamp indicators	Dashboard must display latency indicator and ensure refresh rate 2s delay
Manage participant data securely	Privacy and consent violations if linked to identifiable gaze records	Breach of confidentiality; ethical non-compliance	Improper anonymization; insecure linkage between datasets	De-identify all participant data; IRB/ethics protocol compliance	No personal identifiers may be stored with gaze data in analysis tables
Analyze gaze data	Algorithmic errors in data interpretation	Inaccurate engagement conclusions; invalid study results	Incorrect model logic, poor calibration, inadequate testing	Implement unit tests and verification datasets; peer review of algorithms	All data analysis modules must be verified with ground-truth calibration data
Analyze correlation models	Bias in assumed correlations between gaze and supplementary data	Misleading findings about engagement; false causal interpretations	Overfitting, poor study design, unverified statistical assumptions	Include bias detection procedures; statistical validation before reporting	Correlation models must be reviewed by study supervisor before use
Collect survey responses	Missing or incomplete responses	Reduced accuracy of analysis; gaps in participant context	Participant non-response, survey fatigue, technical form errors	Enforce mandatory fields; data completeness checks; participant reminders	Survey platform must validate completeness before submission
Collect self-assessment data	Misreporting in self-assessment	Inaccurate correlation between gaze and engagement	Participant subjectivity, misunderstanding questions	Include validation questions; statistical outlier checks	Survey must include cross-check items to detect inconsistent answers
Store and manage study data	Data corruption or loss	Loss of study data; incomplete analysis	Disk failure, interrupted file writes, inadequate backup	Regular data backups; use redundant cloud storage	System must maintain daily backups and integrity checksums
Operate eye-tracking hardware	Hardware malfunction	Incorrect gaze readings; unreliable data	Sensor misalignment, firmware issues, physical damage	Periodic calibration and maintenance logs	Device must auto-check calibration before each session
Control experimental environment	Inconsistent environmental conditions (lighting, seating layout)	Reduced accuracy or participant exclusion	Poor classroom setup or uncontrolled environment	Standardize setup; log environmental parameters	Experiment setup must follow documented calibration standards
Manage participant comfort and consent	Student discomfort or awareness of monitoring	Altered natural behavior; reduced validity of gaze data	Obtrusive hardware placement; insufficient consent briefing	Conduct pre-study familiarization sessions; maintain unobtrusive setup	Participants must receive clear consent and explanation before recording
Map gaze data to visual stimuli	Inaccurate handling of gaze data when not aligned to visual stimuli (e.g., off-screen gaze)	Misinterpretation of focus or attention zones	Incomplete mapping algorithms, field-of-view errors	Include "off-screen" classification; improve mapping calibration	Gaze mapping algorithm must identify off-screen events separately
Capture gaze data	Catching private information through gaze data	Unintentional collection of sensitive personal data, breach of user privacy	Excessive gaze capture scope; lack of masking zones	Limit gaze tracking to relevant visual field; apply region masking	Tracking software must enforce spatial limits for data capture

6 Safety and Security Requirements

- **SR: Dashboard must undergo user-centered design testing before deployment** *Rationale:* Prevents confusing layouts or visual errors that could mislead instructors. *How to fake it:* Create a Figma prototype or lightweight web mockup to gather early user feedback before full implementation.

- **SR: Dashboard must display latency indicator and ensure refresh rate $\leq$ 2s delay** *Rationale:* Helps instructors detect real-time lag, maintaining trust in displayed data. *How to fake it:* Simulate delayed data flow in a web app and display a latency counter overlay to visualize timing performance.

- **SR: No personal identifiers may be stored with gaze data in analysis tables** *Rationale:* Prevents accidental re-identification of participants and ensures privacy compliance. *How to fake it:* Use mock datasets with anonymized IDs; verify database schemas exclude personal fields like names or emails.

- **SR: All data analysis modules must be verified with ground-truth calibration data** *Rationale:* Ensures algorithmic correctness and prevents spurious interpretations. *How to fake it:* Compare model outputs against a pre-labeled "gold-standard" dataset and adjust calibration coefficients manually.

- **SR: Correlation models must be reviewed by study supervisor before use** *Rationale:* Reduces the risk of biased or invalid interpretations from flawed correlation assumptions. *How to fake it:* Prepare a simple statistical summary (e.g., scatter plots, R-values) for supervisor review before final reporting.

- **SR: Survey platform must validate completeness before submission** *Rationale:* Prevents missing participant responses that could distort analysis. *How to fake it:* Implement basic "required field" validation in Google Forms or a test web form.

- **SR: Survey must include cross-check items to detect inconsistent answers** *Rationale:* Improves reliability of self-reported data by identifying careless or contradictory responses. *How to fake it:* Add duplicated or rephrased items (e.g., "I often multitask in class" vs. "I rarely focus on one thing") and verify internal consistency.

- **SR: System must maintain daily backups and integrity checksums** *Rationale:* Ensures data persistence and prevents silent corruption or accidental loss. *How to fake it:* Automate daily file copies to a secondary directory; generate simple MD5 checksums to confirm integrity.

- **SR: Device must auto-check calibration before each session** *Rationale:* Guarantees consistent hardware accuracy and avoids drift-related errors. *How to fake it:* Simulate calibration pop-ups in the user interface or log "calibration passed" events before recording.

- **SR: Experiment setup must follow documented calibration standards** *Rationale:* Reduces environmental variability and improves reproducibility between sessions. *How to fake it:* Create a short setup checklist (lighting, distance, seating layout) and follow it during each test run.

- **SR: Participants must receive clear consent and explanation before recording** *Rationale:* Ensures ethical compliance and participant comfort. *How to fake it:* Use a digital consent form that requires an explicit "I agree" before data collection begins.

- **SR: Gaze mapping algorithm must identify off-screen events separately** *Rationale:* Prevents incorrect attribution of gaze to visual elements, improving data validity. *How to fake it:* Add an "off-screen" flag to simulated gaze data whenever coordinates fall outside display bounds.

- **SR: Tracking software must enforce spatial limits for data capture** *Rationale:* Prevents unintentional recording of private or irrelevant information. *How to fake it:* Restrict capture areas in code or simulate bounding boxes in visualization tools (e.g., OpenCV or Figma overlays).

7 Roadmap

The safety requirements that will be implemented during the capstone are the following:

- Dashboard must undergo user-centered design testing before deployment
- Dashboard must display latency indicator and ensure refresh rate $\leq$ 2s delay
- No personal identifiers may be stored with gaze data in analysis tables

The other safety requirements will be referred to once again to see which safety requirements have been met and which ones will still need to be worked on.

Appendix — Reflection

Angela

1. **What went well while writing this deliverable?** Since we worked on the requirements in the SRS, it was easier to link hazards to specific system functions. The examples from class helped a lot with structuring the FMEA and understanding how to connect risks to our design.

2. **What pain points did you experience and how did you resolve them?** It was hard to know which hazards were most relevant since some requirements are still TBD. Rating severity and detectability was also tricky. I made simple assumptions for now and noted what to confirm with the supervisors later.

3. **Which risks were known before, and which emerged during the Hazard Analysis?** We already knew about privacy and data handling risks. During the analysis, we found new ones like incorrect gaze mapping and sync errors that could affect data accuracy.

4. **Beyond physical harm, list at least two other types of software risk and why they matter.** *Privacy and Ethical Risks:* Gaze data could expose personal information if not handled properly. *System Reliability Risks:* Delayed or incorrect data could give instructors the wrong impression of engagement.

   (a) **What went well while writing this deliverable?** Referring to the lecture slides and examples made it straightforward to define hazards and structure the FMEA table. The example provided in class gave us a clear reference for formatting and scope. Collaboration also went smoothly since each team member contributed based on their familiarity with different system components.

   (b) **What pain points did you experience and how did you resolve them?** The hardest part was brainstorming realistic failure modes and their corresponding causes, effects, and mitigations. There were many possible directions, and it took time to identify the most relevant ones. We resolved this by discussing ideas as a team, researching similar projects, and organizing potential risks in a shared spreadsheet until we reached consensus.

   (c) **Which risks were known before, and which emerged during the Hazard Analysis?** A key known risk was data privacy and consent management, which we had anticipated early on. During the analysis, new risks emerged — such as algorithmic bias, calibration errors, and inconsistent environmental conditions — which helped us better understand how technical and human factors interact.

   (d) **Beyond physical harm, list at least two other types of software risk and why they matter.** *Privacy and Ethical Risks:* Mishandling user data or collecting more information than necessary can breach confidentiality and damage user trust. *Data Integrity Risks:* Inaccurate or corrupted data can lead to false conclusions or invalid results, reducing confidence in the system's output.