Hazard Analysis
Software Engineering

Team #13, ARC
Avanish Ahluwalia
Russell Davidson
Rafey Malik
Abdul Zulfiqar

Table 1: Revision History

Date	Version	Author(s)	Notes
2024-10-18	1.0	All	Initial Hazard Analysis

Contents

[You are free to modify this template. —SS]

1 Introduction

[You can include your definition of what a hazard is here. —SS]

2 Scope and Purpose of Hazard Analysis

[You should say what **loss** could be incurred because of the hazards. —SS]

3 System Boundaries and Components

[Dividing the system into components will help you brainstorm the hazards. You shouldn't do a full design of the components, just get a feel for the major ones. For projects that involve hardware, the components will typically include each individual piece of hardware. If your software will have a database, or an important library, these are also potential components. —SS]

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]

- The software system is only used in the intended software environments (unmodified iOS and Android Versions 16.0+ and 12.0+ respectively as per distribution requirement DI-D1)
- The software system is only used on devices that meet the minimum hardware requirements (GPS, camera, and all required sensors present)

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]

5.1 Hazards Out of Scope

Hazards resulting from the failure of the following components will not be considered in this analysis as the software system cannot mitigate hazards in these external systems

- User device hardware
- Back-end server hardware

5.2 Failure Mode and Effects Analysis Table

Table 2: FMEA Table

Design Function	Failure Modes	Effects of Failure	Causes of Failure	Detection	Recommended Action	Req	Ref.
Object Placement	System fails to store AR object instance in database	User has to redo the object placement workflow, wasting their time	Database failure, Back-end overwhelmed with traffic	Provide useful error messages from back-end to app client	Implement automatic retry mechanism for AR object instance storage in the case of storage failure	RR-1, RR-2	H1-1
Object Instance Storage	Database becomes corrupted	Users lose access to their (and other's) AR object instances	Faulty storage devices on server, Bugs in database management software	Automated periodic database testing	Implement a mechanism to restore the database from a backup if necessary, based on automated database testing	RR-3, RR-4	H2-1

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]

6.1 Robustness Requirements

RR-1 The system must have an automated mechanism to retry the upload and storage of object instances when an initial attempt fails

RR-2 All internal APIs of the system must provide useful error messages in the case of system failures

RR-3 The system must automatically back up databases daily

RR-4 The system must have a mechanism to restore a database from a backup in the case of unrecoverable failure / corruption

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

[Not required for CAS 741 —SS]

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