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

1 Introduction

Hazard Analysis is a key step in the engineering process, which is used to identify potential risks and dangers in a system or process. It helps us to ensure the safety and risk management of a system. By systematically analyzing potential risks of the system, we can work to mitigate these potential harms and any consequences that may arise. This document is a key part of the overall safety of the Realm app. It aims to help our stakeholders understand the possible risks of the app and all precautions we have in place to prevent such risks.

2 Scope and Purpose of Hazard Analysis

The scope of this Hazard Analysis covers the identification, evaluation, and mitigation of hazards as it relates to the entire development process of the Realm project. Hazards which are considered in this document include features within the app, and external hazards through the environment.

Certain losses that could be incurred because of hazards are loss of privacy, including unauthorized tracking of users location and unauthorized sharing of personal data, such as email or password. Another loss is health risks from bright flashes within the app, which can trigger seizures in users that may be suffering from photosensitive epilepsy. Furthermore, human injury may occur from accidents because of users being distracted from AR content.

3 System Boundaries and Components

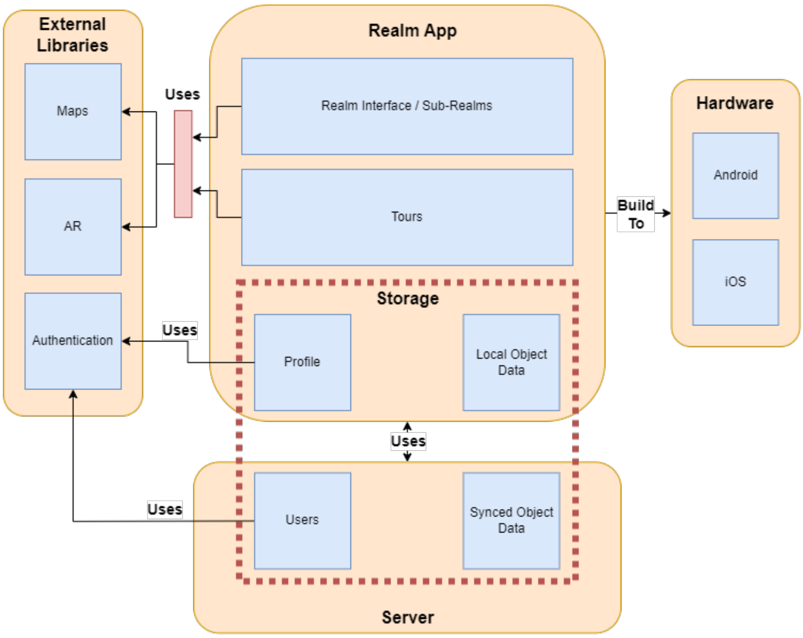

Figure 1: System Boundaries and Components

1. **Realm App**
   - (a) Realm Interface / Sub-Realms
   - (b) Tours
2. **External Libraries**
   - (a) Maps
   - (b) AR
   - (c) Authentication

3. **Cloud**
   (a) Accounts
   (b) Synced Object Data
4. **Hardware**
   (a) Android Devices
   (b) iOS Devices

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	ROR-1, ROR-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	ROR-3, ROR-4	H2-1
Privacy and Data Protection	User data is exposed to unauthorized users	Loss of user trust, potential legal implications, data breaches	Weak encryption, improper access control policies and other security vulnerabilities	Regular security audits, reports of unauthorized access	Implement strong encryption protocols, two-factor authentication, and regular security updates	SR-5, SR-6	H3-1
AR Object Rendering	AR objects fail to render or display incorrectly in the user's environment	Users are unable to see placed objects or experience visual glitches	Device camera issues, insufficient processing power, software bugs, network issues	User-reported issues, monitoring rendering logs	Optimize rendering algorithms for performance; implement fallback modes for low-performance devices	SR-7	H4-1
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	ROR-1, ROR-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	ROR-3, ROR-4	H2-1
Viewing AR objects accurately in the Realm screen	Location access is disabled	User is unable to accurately view object instances in their surroundings	Permission for location denied by mobile device, Location access disabled by the user, Bugs in software component synchronizing object and device location with each other	Having periodic updates of device location	Prompting user to grant location access or transitioning to an offline view of the Realm screen	ACR-1, ACR-2	H3-1
Navigating to AR object cluster	AR objects within selected cluster are deleted by the owner during navigation	User may arrive at destination with no AR objects present, Bugs in navigation software may cause app crashes	All AR objects within the targeted cluster are deleted by respective owners	Keeping track of AR object cluster count	Before starting navigation, check for existence of AR object cluster. Notify user about objects being deleted by owners. Provide option to start navigation back to original starting point	ACR-3, ROR-1	H4-1
	Location access is disabled	System is unable to present current user location, system is unable to display next instruction	Location access disabled by the user or device system	Having periodic updates of device location	Prompt user to grant location access to continue navigation	ACR-2	H4-2

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]

SR-1 The system shall implement encryption protocols to protect user data when storing.

SR-2 The system shall provide a multi-factor authentication option for user accounts to enhance security.

SR-3 The system shall provide fallback modes for rendering AR objects on low-performance devices to ensure accessibility for all users.

SR-4 The system shall allow users to customize rendering settings (e.g., brightness, effects) to minimize discomfort or health risks associated with viewing objects.

SR-5 The system shall display indicators to alert users when rendering might cause discomfort (e.g., rapid movement or flashing effects).

6.1 Definitions

1. **Encryption standard** - An encryption standard is a set of algorithms used to encode data to ensure that it can be viewed by authorized users only.

6.2 Safety Requirements

SAR-1 The system should not distract users from their surroundings to the extent that they lose awareness of potential collisions or inadvertently enter restricted areas.

SAR-2 The system shall have warnings for bright flashes or loud noises.

SAR-3 The system should have the option to disable bright lights and loud noises.

SAR-4 The system shall be designed to operate with minimal battery usage, ensuring that its consumption does not exceed the average battery usage of comparable applications within the same category.

6.3 Security Requirements

SER-1 The system should follow an 1encryption standard for communication between users and with the administrator.

SER-2 The system should use a secure method of authenticating user access to system.

SER-3 The system shall encrypt all user data stored using an 1encryption standard.

SER-4 The system should not reveal the general user location to other general users.

6.4 Accessibility Requirements (accessibility of product features)

ACR-1 The system shall have an offline (without location syncing) view for interactive components.

ACR-2 The system shall notify the user to grant access to needed device data.

ACR-3 The system should allow the user to terminate navigation in the Maps components.

6.5 Robustness Requirements

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

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

ROR-3 The system must automatically back up databases daily

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

ROR-5 The system shall keep track of object instance count for AR object clusters in the Maps component.

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?

Ans. Answer 1

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

Ans. Answer 2

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?

Ans. Answer 3

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?

Ans. Answer 4