# Hazard Analysis
# UnderTree

Team 22, Capstoners
Palanichamy Veerash
Kannammalil Kevin
Qureshi Eesha
Ahmed Faiq

Table 1: Revision History

| **Date** | **Developer(s)** | **Change** |
|---|---|---|
| October 18, 2022 | Faiq, Veerash | Came up with system boundaries and FMEA table |
| October 19, 2022 | All members | Finished FMEA table |
| October 19, 2022 | Eesha | Finished section 1 and 2 |
| October 19, 2022 | Faiq | Finished section 4 |
| October 19, 2022 | Veerash | Finished section 3 and 7 |
| October 19, 2022 | Kevin | Finished section 6 |

# Contents

# 1 Introduction

This document outlines the hazard analysis for the Undertree Latex editing software. Undertree is a collaborative Latex editor that allows multiple users to edit a Latex document concurrently, and also provides version control capabilities. The software is complex and consists of multiple components that work together to provide the intended functionalities. As such, each component runs the risk of failure due to hazards that may affect the system's behaviour.

For the purpose of this document a hazard is defined as a state of a system along with an action on the system that would lead to the system integrity being damaged or causing a complete failure.

# 2 Scope and Purpose of Hazard Analysis

The purpose of this document is to provide a detailed analysis of the possible hazards to the system, their causes, and the intended preventative measures that will be taken against them.

First, any critical assumptions will be stated. These are assumptions that will eliminate the need for preventing a hazard as it is assumed to never occur. This is followed by the failure mode and effect analysis which will present a detailed table documenting all components and their potential failures, causes of the failures, effects of the failure, and preventative measures. Then, the safety and security requirements will be derived from the FMEA table and outlined as NFRs. Finally, the document will present a roadmap for when these security requirements will be implemented.

When considering the hazards for this project, the following hazards are out of scope as they are not directly part of the project:

- Issues with the user's browser such as compatibility with out of date or legacy browsers such as Internet Explorer are not part of the scope of the project.
- Issues with the rented hosting hardware and it's uptime are not part of the scope of the project.

# 3 System Boundaries and Components

The system consists of 4 distinct major features that also make up our system boundaries are:

1. Client - The client is the latex editor that will be accesses in a web browser, and it has the following components:
    - Editor
    - Chat box
    - Logging in into the application
2. Server - The server is what the client will be communicating with to acces the main functionality which are:

- Accessing the database
- File Synchronization
- Latex Compilation
- Chat Synchronization
- Authentication
- GitHub Integration
- Latex Editor

3. Github - Our application will also be using GitHub API to carry out various git functionality
4. Database - Our application will also be saving user specific data to the database

# 4 Critical Assumptions

We are making the following assumptions when considering our hazards:

- The user will be using the most up to date version of a Chromium based browser, Firefox, or Safari to prevent any hazards with the front-end system.
- Users and admins will not have direct access to the database, all access to the database will be defined by strict queries executed from the back end
- We will not be considering any issues that may arise on strictly just the front-end from a user acting maliciously, as this will only effect the malicious user them self. For example, if a user tries to create a fake and invalid access token from their browser, this token would not work and cause errors for the user when trying to communicate with the back-end until this token is cleared and a valid token is accessed by logging in.

# 5 Failure Mode and Effect Analysis

Table 2: FMEA Table

| Component | Failure Modes | Effects of Failure | Causes of Failure | Recommended Action | SR | Ref. |
|---|---|---|---|---|---|---|
| Database | Data is unintentionally deleted | User data is lost | 1. Database failure<br>2. Malicious Actor | 1. Backup data frequently<br>2. Establish strict queries to fetch and add data to database, and escape all user provided information that is used in queries. | 1. NFR32<br>2. NFR27<br>NFR35 | H1-1 |
| | Database is unavailable | User loses access to personal data | 1. Database failure<br>2. Database Maintenance<br>3. Malicious Actor | 1. Display an error stating that the work is unable to be synchronized with the database and allow them to continue working offline.<br>2. Display an error stating that the database is down for maintenance and give an expected downtime. Allow users to continue working offline.<br>3. Display an error stating that the website is down. Prevent access to website while the scope of the attack by the malicious actor is investigated. | 1. NFR28<br>2. NFR28<br>3. NFR28 | H1-2 |
| | Data overwritten | User data is lost | 1. Identical keys | Enforce unique primary key constraints | NFR35 | H1-3 |
| File Synchronization | Changes to file overwritten | Inaccurate file version stored | 1. Race condition | Implement synchronization between concurrent changes to file data store | NFR35 | H2-1 |
| | Back-end Unavailable to synchronize changes | Inaccurate file version stored | 1. Backend server down/unreachable | Store current version locally in client side | NFR34 | H2-2 |
| Latex Compilation | Compilation Crashes/Stuck | User does not see built PDF | 1. Compilation times out due to error | Stop the compilation and display an error message to the user stating that the compilation timed out | NFR28 | H3-1 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Chat synchronization | Messages overwritten | Overwritten message would not be delivered to all users | 1. Race condition | Implement synchronization between concurrent changes to messages data store | NFR35 | H4-1 |
| | Back-end unavailable to synchronize messages | New messages would be unable to be delivered | 1. Error in back end causing crash | Display a message stating that the messages cannot be synchronized and allow users to continue working in offline mode | NFR28 | H4-2 |
| GitHub Integration | GitHub services are unavailable | User is not able to use any of the GitHub functionality | 1. GitHub servers are not reachable<br>2. Maintenance with GitHub services | Display a message stating the commit and push functionalities are currently unavailable | NFR28 | H5-1 |
| | GitHub API key is invalid | User is not able to use any of the authorized GitHub functionality | 1. GitHub API key is expired<br>2. GitHub API key is invalid | 1. Renew the key using the refresh token<br>2. Ask the user to login again | 1. NFR40<br>2. NFR42 | H5-2 |
| Latex editor | File is unintentionally deleted | File data is lost | 1. Misclick | User should be prompted when file is deleted. Most recently edited files should be also stored even after deleted for some time | NFR29 | H6-1 |
| | Project is unintentionally deleted | Project belonging to the user are lost | 1. Misclick | User should be prompted when project is deleted | NFR29 | H6-2 |
| Authentication | Unauthorized user gets access to privileged data | Unauthorized user will be able to get access to and modify data of other users | 1. Unsecure authentication system | Inform users of the breach and restore database backup to undo any changes made by malicious actor | NFR26<br>NFR27<br>NFR36<br>NFR38 | H7-1 |

# 6 Safety and Security Requirements

## 6.1 Access Requirements

NFR24. The system's code will only be view-able to the public through the Git repository
NFR25. The system restricts it's editing privileges for the code to only the maintainers
Fit Criterion: Users will be able to view the source code on the **GitHub** repository page
NFR26. Users can only access projects that they are authorized to do so
NFR27. The system will retrieve the minimum required data needed for the user
NFR28. The system will provide appropriate errors to communicate system issues to user
NFR29. The system will provide necessary confirmations to crucial changes

## 6.2 Integrity Requirements

NFR30. The system will not manipulate or modify any of the user's data that is stored on it
NFR31. The system will protect itself from intentional abuse
NFR32. The system will back up the data in the database frequently
NFR33. The system will retain user's projects even after being deleted for a few days
NFR34. The system will store the editor data locally on the client's device
NFR35. The system will implement strict measures in the back end to prevent unintentional behaviour

## 6.3 Privacy Requirements

NFR36. The system will require the user to create an account
NFR37. The system will not use the user's personal information for anything than what is required and consented to by the user
NFR38. The system will store all user credentials securely
NFR39. Users can only access projects that they are authorized to do so
NFR40. The system will renew API keys regularly
NFR41. The system will re-authenticate the user when required

## 6.4 Audit Requirements

N/A

## 6.5 Immunity Requirements

N/A

# 7 Roadmap

Some of the new requirements that were discovered as part of this Hazard Analysis will be phased in alongside our Requirements mentioned in SRS document. Due to project time constraints, some of the requirements will not be able to implemented within the final project deadline. However since most of the hazards are integral towards the functional requirements, we will be looking to implement solutions for all hazards aside from the database backup.