Hazard Analysis
Software Engineering

Team 2, SyntaxSentinals
Mohammad Mohsin Khan
Lucas Chen
Dennis Fong
Julian Cecchini
Luigi Quattrociocchi

Table 1: Revision History

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

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

The purpose of this hazard analysis is to identify, evaluate, and mitigate potential risks that could lead to system failures or undesired outcomes. In the context of this project, the primary losses incurred due to hazards could include:

- Unauthorized interception of sensitive data, such as code submissions or plagiarism reports which could lead to privacy breaches.
- Misidentification of plagiarism cases, either false positives (innocent submissions flagged) or false negatives (plagiarized submissions unflagged).
- Disruption of service leading to user dissatisfaction, especially in time-sensitive code competition environments leading to loss of reputation.
- Inaccurate similarity scores, which could result in biased or incorrect decisions by professors or competition organizers.

The scope of this hazard analysis will cover the following areas:

- Risks associated with data handling.
- Risks in the plagiarism detection algorithms and model performance.
- User authentication and access control risks.
- Potential human errors in adjusting plagiarism detection thresholds.

The analysis aims to minimize these risks and ensure the robustness, security, and accuracy of the system while maintaining a high level of user trust and system reliability.

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

- Adequate computational resources exist for the real time analysis of the code snippets
- Users do not intend to misuse the product
- Third party resources that support this product will always be functionally correct
- All components on the cloud will provide sufficient scalability and security
- The system will be maintained regularly with bug fixes/performance enhancements
- The criteria for plagiarism is agreed upon by all users

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

[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?