Hazard Analysis
Software Engineering

Team 21, Alkalytics
Sumanya Gulati
Kate Min
Jennifer Ye
Jason Tran

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

A hazard is any property or condition that has the potential to cause harm. This document serves as a hazard analysis for the application revolving around the capstone project "Alkalytics". This project aims to aids in the data management and analysis of an ocean alkalinity enhancement experiment. This document identifies the components of the system, and then the possible software hazards in these components, as well as ways to mitigate the risks they impose.

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

The following are assumptions made about the software of the system.

- The data provided to this system is validated and correctly formatted before system ingestion
  - Errors can occur from badly formatted or invalid incoming data, which is not something the system can control
- The user of this application is not intentionally trying to misuse it
  - This assumption mitigates the risk of someone intentionally damaging the system
- Internet connection and server infrastructure will always be available, and will not suddenly go down and compromise the system
  - This assumption mitigates the risk of a failing internet connection or fragile server infrastructure interrupting the system
- Users using this application understand how to use the application, whether through documentation or a tutorial

  - This assumptions ensures that user errors caused by lack of knowledge do not occur

- The system is regularly maintained for security and bug fixes
  - This assumption prevents any threats to the system due to poor maintenance

- The system is scalable and possesses enough computational resources to handle any volume of data
  - This assumption mitigates the risk of the system failing due to lack of resources

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]

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?