Hazard Analysis
SFWRENG 4G06

Team 28, Cowvolution Minds
Aryan Patel
Harshpreet Chinjer
Krish Patel
Martin Ivanov
Shazim Rahman

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

[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]

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

6.1 Safety Requirements

SFR1: Prevent Inbreeding Between Sire and Dam

- **Description**: The system must prevent inbreeding by detecting when a sire and dam are closely related and disallowing or flagging those breeding pairs. The system should calculate the relationship coefficient between breeding pairs and raise an alert if it exceeds a predefined threshold.
- **Rationale**: Inbreeding increases the risk of genetic disorders, reduced fertility, and poor overall herd health. Preventing inbreeding ensures better genetic diversity and healthier offspring.
- **Fit Criterion**: The system should compare genetic records of sires and dams, ensuring that no breeding pair with a relationship coefficient above a set threshold is approved. This value can currently set to 10%, but will likely to re-evaluated later. A warning or error message must be displayed for all invalid breeding pairs, with 100% accuracy in flagging related pairs during tests.

6.2 Security Requirements

SCR1: Backup and Recovery Mechanism

- **Description**: The system must regularly back up its data to ensure that critical information, such as local breeding data, developed models, and so forth are not lost in case of system failure or crash. It should also provide a reliable recovery mechanism to restore data after an incident.
- **Rationale**: Data loss can lead to incorrect predictions, poor breeding decisions, and financial losses for farmers. A backup and recovery mechanism ensures that the system remains reliable and that users can recover from potential failures.
- **Fit Criterion**: The system must successfully back up data every 24 hours, witmah at least 99% success in restoring data during tests.

SCR2: Multi-Factor Authentication (MFA)

- **Description**: The system must enforce multi-factor authentication for all administrative users to ensure secure access to sensitive data, such as genetic and breeding records. MFA should require users to authenticate using two or more credentials, such as passwords and authentication tokens.
- **Rationale**: Sensitive data requires an additional layer of security to prevent unauthorized access. By using MFA, the risk of data breaches and unauthorized access is minimized.
- **Fit Criterion**: MFA must be implemented and enforced for all administrative-level accounts, with at least 95% of users successfully authenticating during tests.

7 Roadmap

SFR1 will be included in the capstone timeline as it involves safety and ethical concerns. It will be implemented when the system is being built. The other safety concerns are not of high priority and will therefore may be inlcuded if there is extra time.

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?