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

This section outlines the key assumptions made about the system and its operating environment. These assumptions are intended to define the boundaries and to manage potential risks.

Data Collection Limitations

- **Assumption:** All data used for model training and predictions will be sourced from existing datasets provided by CATTLEytics Inc. or relevant partners. No additional data collection will be undertaken by the project team.
- **Rationale:** The scope of the project does not include new data collection activities, limiting the system to work with available historical and provided real-time data. This constraint may limit the model's ability to adapt to emerging conditions not represented in the existing dataset.
- **Mitigation:** Ensure a comprehensive understanding of the provided dataset's limitations, and apply model validation techniques (such as cross-validation) to mitigate risks associated with potential biases or gaps in the data. Regular updates to the dataset from the provider should also be encouraged to keep the model relevant.

Data Accuracy and Availability

- **Assumption:** The historical and real-time data provided to the machine learning model are accurate, complete, and relevant to the cows and their environment.
- **Rationale:** The model's accuracy and reliability depend heavily on the quality of input data. Inaccurate or incomplete data could lead to faulty predictions and management decisions.
- **Mitigation:** Implement data validation procedures to ensure the accuracy and completeness of incoming data.

Stable Operating Conditions

- **Assumption:** The environmental and operational conditions on the farm (such as temperature, feed availability, and animal health monitoring) remain within expected ranges during the model's operation.
- **Rationale:** Significant deviations in farm conditions could affect the model's performance, leading to inaccurate predictions.
- **Mitigation:** Develop contingency measures or notification systems for when conditions fall outside normal ranges.

Model Generalizability

- **Assumption:** The machine learning model generalizes well across different farms and herd conditions without needing extensive retraining for every farm.
- **Rationale:** The model is intended to be a broadly applicable tool across multiple farms with varying conditions. Excessive reliance on farm-specific parameters would increase complexity and reduce scalability.
- **Mitigation:** Incorporate regular model updates and feedback loops to address variability between different environments.

Ethical Use and Compliance

- **Assumption:** Users will adhere to legal and ethical standards when implementing recommendations made by the tool, particularly regarding animal welfare and data privacy.
- **Rationale:** Legal compliance (e.g., PIPEDA) and ethical considerations (animal welfare) are critical to ensuring the tool is used responsibly.
- **Mitigation:** Provide user education and warnings for critical decisions that impact animal welfare, and ensure compliance with data privacy laws.

Technical Infrastructure Reliability

- **Assumption:** The technical infrastructure (servers, farm management platforms, IoT devices) that supports the model operates reliably with minimal downtime or technical failures.

- **Rationale:** System malfunctions or data transmission failures could lead to gaps in model predictions, causing delays in critical decision-making.
- **Mitigation:** Implement fault-tolerance mechanisms and regular system health checks to ensure the infrastructure remains stable.

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?