Hazard Analysis
ProgName

Team #, Team Name
Student 1 name
Student 2 name
Student 3 name
Student 4 name

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

3 System Boundaries and Components

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]

- We will assume that any health care provider using our product would not intentionally misuse it.
- We will assume that any health care provider using our product would not use our product with malicious intent and only use it to help their patient.
- We will assume that the libraries and functions we utilize in our code work as expected such as NumPy, Pandas, and SkLearn.

5 Failure Mode and Effect Analysis

[Include your FMEA table here —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]

SR3. The system will be tested periodically to detect crashes and a potential lack of processing power

Rationale: As we build our system out we will be adding more features and increasing the complexity, which increases the processing power required. This may lead to crashes or issues with a lack of processing power that will need to periodically be checked. This is to check the hardware limitations.

Associated Hazards: H2-1

SR4. The system will run through a series of tests before deploying/pushing new builds

Rationale: This will be done to ensure that the system is still working as expected with our new changes and that nothing has broken before we deploy a new build. These tests will help us detect that and allow us to go back and fix any issues that arise.

Associated Hazards: H4-1

SR5. The team will regularly push their code by committing to a repository while working on the code

Rationale: This needs to be done in order to avoid losing code/work. If something were to happen to the team members computer for whatever reason and the work that was done after the last commit was to be deleted/lost, frequent commits would help ensure minimal lose.

Associated Hazards: H5-1

7 Roadmap

[Which safety requirements will be implemented as part of the capstone timeline? Which requirements will be implemented in the future? —SS]