Hazard Analysis
Sandlot

Team 29
Nicholas Fabugais-Inaba
Casra Ghazanfari
Alex Verity
Jung Woo Lee

Table 1: Revision History

Date	**Developer(s)**	**Change**
Oct. 21, 2024	NFI, JL, AV, CG	TA Feedback
Oct. 23, 2024	NFI, JL, AV, CG	Rev0
...	...	...

Contents

1 Introduction

May not need to restate product
Sandlot is intended to be the successor for the current McMaster GSA softball league scheduling and management platform. This project will implement the current functionality from the existing system including additional features such as a new login system, commissioner admin privileges, and an improvement to the robustness, ease of use, and maintainability to the platform.

Hazards are conditions of the product system that could lead to undesirable outcomes. In Sandlot, hazards of security, malfunctions, usability, among others will be analyzed.

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]

If you look at the samples, the components are where the hazards may come from. I think you could probably think of it as what section of the product the hazard will come from. Try using them to figure how how to break Sandlot down into components.

3.1 System Components

The system can be divided into the following components:

- Authentication

  This component authenticates user inputted login information with the database and determines whether a login is successful.

- Scheduling

  This component generates a schedule based on user inputted availability data. Additionally, it allows for rescheduling events in these schedules after they have been generated.

- Accounts

  This component allows users to create, delete or edit accounts. Additionally, account types determine the ways in which a user can interact with the system, its data, and its components.

- Team Structure

  This component is a data strcuture of the system which allows accounts to be grouped together such that a group of accounts interact with the system, its data, and its components in the same way as each other. Additionally, accounts can be added/removed from teams, users can request to join a certain team, and teams can be created/deleted.

- Scoring/Standings

  This component allows users to input score data associated with specific events and will generate cumulative standings based on score data.

- Alerts

  This component alerts users of important information related to the system.

- Database

  This component stores the system's data.

- User Interface

  This component displays the system's data to the user.

3.2 System Boundaries

server

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]

Component	**Failure Mode**	**Failure Effect**	**Failure Cause**	**RPN**	**Recommended Actions**
Example row LEAVEMPTY	Column 1 Failure Mode	Column 2 Failure Effect	e Failure Cause	e RPN	e Recommended Actions

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?