Hazard Analysis
Plutos

Team #10, Plutos
Payton Chan
Eric Chen
Fondson Lu
Jason Tan
Angela Wang

Table 1: Revision History

Date	**Developer(s)**	**Change**
10/23/2024	Angela	Initial draft
...	...	...

Contents

1 Introduction

A hazard in the context of this document is any property or condition that may lead to harm or damage to the Plutos system or its users. Potential losses due to these hazards may include loss of application functionality, performance, or accuracy, or breaches of user privacy or data. The following sections will identify hazards within the system and discuss the controls in place for their mitigation.

2 Scope and Purpose of Hazard Analysis

This document aims to provide a comprehensive hazard analysis of the Plutos system. It identifies hazards within the system, outlines measures to mitigate them, and specifies the safety and security requirements derived from this analysis. The analysis will follow the Failure Mode and Effect Analysis (FMEA) approach. The analysis aims to discover the potential failure modes within the system and develop a mitigation plan to reduce the risk of failure.

3 System Boundaries and Components

The system will be divided into the following components:

1. The Plutos application, which consists of:
   (a) **The database**: The database is where the user's receipts and profile data will be stored.
   (b) **The backend server**: The backend server is responsible for handling and serving requests from the client. It will interact with all the other components listed here.
   (c) **The frontend/user interface**: The frontend/user interface is responsible for displaying the appropriate views to the user and handling user interactions.
   (d) **The machine learning (ML) model**: The ML model is responsible for parsing and categorizing items from a picture of an itemized receipt.
2. The user's mobile device and camera setup

4 Critical Assumptions

The project will be making the following critical assumptions:

1. The users will be using a mobile device running an up-to-date version of iOS or Android.
2. Users are not expected to repeatedly input invalid images into the system (i.e., images that do not contain a receipt). While it is anticipated that users may occasionally submit an invalid image, it is assumed to not be a significant concern.

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]

Table 2: Failure Mode and Effect Analysis Table

Design Function	Failure Modes	Effects of Failure	Causes of Failure	Recommended Action	SR	Ref
...	...	...	...	...	...	...

6 Safety and Security Requirements

1. **Receipt Image Protection:** Applying image sanitization techniques to ensure that malicious content cannot be injected into the app via uploaded images (e.g. XSS). Will also ensure that images of receipts are securely stored using access controls and encryption and should be deleted when they are no longer needed or upon user request.
2. **User Awareness and Education:** Providing users with information about security best practices and secure usage such as how to create strong passwords, recognize phishing attempts, and manage data privacy.
3. **Regular Security Updates:** Our app will be regularly updated to address newly discovered vulnerabilities, both in the app code along with any third-party libraries of frameworks. This will be done on an iterative basis with periodic patch releases (e.g. patch release every 3 weeks).
4. **Data Anonymization:** Before storing or processing any receipts that are uploaded, all sensitive information such as account numbers or personal details will be anonymized to protect user privacy. This is especially important when using data for AI training or analysis.
5. **Physical Device Security:** While the assumption is that users will be on a mobile device, we do require the app to be run on a secure device (e.g. detecting if the device is jailbroken or rooted), This can help against local data theft or tampering. Additionally, we will also provide users with the ability to remotely wipe sensitive financial data and receipts from their devices in the case of loss or theft.

7 Roadmap

Due to the time constraint of the capstone timeline, there will be only time to implement certain safety/security requirements. The rest of the safety requirements will be implemented in the future (likely following the capstone period).

The requirements that will be implemented as part of the capstone time will be:

1. User Awareness and Education
2. Physical Device Security

The requirements that will be implemented in the future will be:

1. Receipt Image Protection
2. Regular Security Updates
3. Data Anonymization

Appendix — Reflection

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?

   The overall process while writing this deliverable was smooth and efficient as we were quickly able to identify the potential hazards related to our project. We brainstormed several ambiguous sections or things we thought were a bit unclear within this analysis document, and were able to get very clear answers from our helpful TA. The team worked well together as we all put in our best efforts and supported one another when completing this task.

   Using the FMEA (Failure Mode and Effect Analysis) approach helped streamline the hazard identification process. Breaking down the system into components allowed for a clear understanding of where risks might occur. Writing the deliverable helped the team clarify and solidify their understanding of how the receipt scanner, the AI model, and other system components interact, making it easier to identify hazards.

2. What pain points did you experience during this deliverable, and how did you resolve them?

   At first, it was challenging to define the potential failure modes, especially for components like the machine learning model. The team resolved this by conducting additional research on common failure points in similar systems and reviewing how AI models typically behave with poor input data. Another challenge was balancing realistic assumptions about user behavior with potential risks. For example, while assuming users won't repeatedly input invalid images, we acknowledged this could still happen. We resolved this by planning mitigation strategies for those edge cases.

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?

   We had already considered risks related to image quality (e.g., blurry or incomplete receipt images) and network connectivity issues (e.g., users not being able to connect to the database).

   While working on the hazard analysis, we realized potential risks like Optical Character Recognition (OCR) misinterpretation due to varied receipt fonts and ML model processing time under different device conditions (e.g., low memory or poor network). These came up while brainstorming as a team and thinking about the specific steps in image processing and how the system handles diverse input.

4. Other than the risk of physical harm, list at least 2 other types of risk in software products. Why are they important to consider?

Two other risks that are apparent in software products are security and reliability risks.

Security vulnerabilities can lead to issues such as data breaches, unauthorized access or identity theft, as well as collateral damages, whether it be financial losses or reputational damage. This is considered a risk and is important to consider as it creates an opportunity for malicious users to exploit weaknesses in software systems, which can have a range of detrimental consequences. Examples include operational disruptions, intellectual property theft, ransomware attacks, etc.

As for reliability, it is mostly concerned with when software fails to function consistently, such as having frequent downtimes. This can affect the user's experience, leading to a loss of productivity or customer dissatisfaction. Both of these can lead to potential revenue loss. This is classified as a risk and is important to consider because unreliable software can lead to negative consequences, which affect not only the users but also the organization that provides the software. The damages can be both monetary and non-monetary, such as losing user trust/loyalty, reputational damage, and associated compliance and legal risks.