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

Table 2: Failure Mode and Effect Analysis Table

Design Function	Failure Modes	Effects of Failure	Causes of Failure	Recommended Action	SR	Ref
Authenticate user	Non-human accounts exist on the server	Increased traffic to the server, invalid inputs, performance deterioration	A bot account logs into the application	Ensure account creation includes a captcha	FR3	H1-1
Accept image	Camera does not open	The user cannot input an image to be processed by the system.	• Application does not have access to user's camera • The application is bugged out	• Prompt the user to allow camera access for the application • The user should restart the application.	FR7	H2-1
	Photo library doesn't open	Same as H2-1	Same as H2-1	Same as H2-1	FR8	H2-2
Process receipt image	Image is not processable	Poor input into machine learning model; may cause incorrect results or cause model to run for a long time	• Poor quality receipt (e.g., paper is crumpled or wrinkled, light ink) • Poor photo quality (e.g., blurry, part of receipt cut out from frame, low lighting) • The image does not contain a receipt	• Prompt the user to retry due to poor photo quality and make sure that the input is actually a receipt • Same as H3-1a • Run image validation before passing the image through the ML model (e.g., check that there is a paper with text in the image)	FR16	H3-1
	Optical character recognition (OCR) failing or incorrectly parsing text	Incorrectly parsed text, missing items, incorrect categorization or costs	• Receipt uses different font sizes or styles • Receipt quality causes some letters to be difficult to recognize (e.g., 0 and O, T and I)	• Inform user that the OCR isn't working after 3 failed attempts and ask user to manually input receipt expenses	FR16, NFR4	H3-2

Continued from previous page

Design Function	Failure Modes	Effects of Failure	Causes of Failure	Recommended Action	SR	Ref
	Incorrect item categorization	Items are miscategorized, resulting in incorrect insights	• OCR incorrectly parses text or fails to read some text • Ambiguous item name • Item has not been identified in the past	• Refer to H3-2. • Prompt the user to confirm the category of this object and give it an alias for helping in future recognition. • See H3-3b	FR17, NFR2	H3-3
	Processing image takes more than X seconds to complete	• Lower user satisfaction	• User's device is low on memory or has other tasks running in the background • Large image size/high image resolution • OCR model not optimized (especially for that specific use case) • Poor pre-processing	• Inform the user that the task is taking longer than expected and show troubleshooting tips. • See H3-2	NFR8, NFR16	H3-4
Save new receipt input (image details and model results)	Cannot connect to database	Receipt input cannot be saved after being processed.	• Poor network connection • Database server downtime	• Prompt the user to check their network connection • Inform the user of the server error; system will try again at a later time (data will be stored locally and then backup up)	FR15	H4-1

Continued from previous page

Design Function	Failure Modes	Effects of Failure	Causes of Failure	Recommended Action	SR	Ref
The system suggests user budget and goals	The system suggests a budget that is lower than the user's total monthly expenses (i.e. unattainable budget)	• Lower user satisfaction • User is unable to budget correspondingly to the suggested system budget	• Algorithm to calculate the budget fails	• Provide an option to the user to recalculate monthly budget with a given minimum monthly expense (user enters minimum monthly expense and system recalculates budget)	NFR5	H5-1

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

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.