

# Verification and Validation Report: Plutos

Team #10, Plutos

Payton Chan

Eric Chen

Fondson Lu

Jason Tan

Angela Wang

March 10, 2025

# 1 Revision History

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

## 2 Symbols, Abbreviations and Acronyms

Refer to Section 1.3 of the [Software Requirements Specification \(SRS\)](#) document for the list of symbols, abbreviations, and acronyms.

In addition, the following abbreviations are used in this document:

Table 1: Symbols, Abbreviations, and Acronyms

symbol	description
V&V	Verification and Validation
UI	User Interface
OCR	Optical Character Recognition
SQL	Structured Query Language
GDPR	General Data Protection Regulation

# Contents

<b>1</b>	<b>Revision History</b>	<b>i</b>
<b>2</b>	<b>Symbols, Abbreviations and Acronyms</b>	<b>ii</b>
<b>3</b>	<b>Functional Requirements Evaluation</b>	<b>1</b>
<b>4</b>	<b>Nonfunctional Requirements Evaluation</b>	<b>2</b>
<b>5</b>	<b>Comparison to Existing Implementation</b>	<b>3</b>
<b>6</b>	<b>Unit Testing</b>	<b>4</b>
6.1	Front-end unit tests . . . . .	4
<b>7</b>	<b>Changes Due to Testing</b>	<b>4</b>
<b>8</b>	<b>Automated Testing</b>	<b>5</b>
<b>9</b>	<b>Trace to Requirements</b>	<b>5</b>
<b>10</b>	<b>Trace to Modules</b>	<b>5</b>
<b>11</b>	<b>Code Coverage Metrics</b>	<b>5</b>

# List of Tables

1	Symbols, Abbreviations, and Acronyms . . . . .	ii
2	Functional Requirements Evaluation . . . . .	1
3	Nonfunctional Requirements Evaluation . . . . .	2
4	Test traceability Table . . . . .	9

# List of Figures

This document reports the results of the Verification and Validation (V&V) process for the Plutosoftware. The V&V plan is documented in the [Verification and Validation Plan](#) document.

### 3 Functional Requirements Evaluation

The functional system tests can be found in Section 4.1 of the [Verification and Validation Plan](#) document. These tests are all performed manually.

Table 2: Functional Requirements Evaluation

Test ID	Pass/Fail	Comments
test-UAM-1	Pass	
test-UAM-2	Pass	
test-UAM-3	Pass	
test-UAM-4	Fail	Not yet implemented
test-UAM-5	Pass	
test-UAM-6	Pass	
test-IP-1	Pass	
test-IP-2	Pass	
test-IP-3	Fail	Not yet implemented
test-MIS-1	Pass	
test-DM-1	Pass	
test-DM-2	Pass	
test-RS-1	Pass	
test-RS-2	Pass	
test-RS-3	Pass	
test-FT-1	Fail	Not yet implemented
test-FT-2	Pass	
test-FT-3	Fail	Not yet implemented

## 4 Nonfunctional Requirements Evaluation

The nonfunctional system tests can be found in Section 4.2 of the [Verification and Validation Plan](#) document. Tests without comments are performed as described in the plan.

Table 3: Nonfunctional Requirements Evaluation

Test ID	Pass/Fail	Comments
test-ACC-1	Pass	<a href="#">Actual output</a> ; accuracy is $47/57 = 82.46\%$ , which meets the threshold of 80%.
test-ACC-2	Pass	By manually comparing the input ( <a href="#">set of receipt images</a> ) with the <a href="#">resulting output</a> , and calculating the accuracy as described in the V&V Plan, current accuracy is 80%, which meets the threshold of 80%. <ul style="list-style-type: none"><li>• <i>foodbasics_1.jpg</i>: <math>13.5/15 = 90\%</math></li><li>• <i>foodbasics_2.jpg</i>: <math>7/9 = 77.78\%</math></li><li>• <i>walmart_1.jpg</i>: <math>7/10 = 70\%</math></li><li>• <i>costco_1.jpg</i>: <math>18.5/23 = 76.09\%</math></li><li>• Overall accuracy: <math>46/57 = 80.70\%</math></li></ul>
test-ACC-3	Pass	
test-ACC-4	Pass	
test-ACC-5	Pass	
test-PERF-1	Pass	
test-PERF-2	Pass	
test-PERF-3	Fail	Load testing has not yet been performed

test-USAB-1	Pass	
test-USAB-2	Pass	
test-USAB-3	Pass	
test-USAB-4	Pass	
<hr/>		
test-SEC-1	Pass	
<hr/>		
test-MTB-1	Pass	System stability has been tested, but application is not backward compatible since it is still under active development.
test-MTB-2	Pass	
test-MTB-3	Pass	
<hr/>		
test-PORT-1	Pass	
test-PORT-2	Pass	
test-PORT-3	Pass	
<hr/>		
test-REUS-1	Pass	
test-REUS-2	Pass	
<hr/>		
test-UND-1	Pass	
test-UND-2	Pass	
test-UND-3	Pass	
<hr/>		
test-LEGAL-1	Fail	The application is still under active development, so it is still using the testing environment and not all security features are active.
<hr/>		

## 5 Comparison to Existing Implementation

This section will not be appropriate for every project.

## 6 Unit Testing

### 6.1 Front-end unit tests

All front-end unit tests can be found in the [test directory](#)  
Refer to Table 4 for unit test traceability table

## 7 Changes Due to Testing

In accordance to the feedback provided through the Rev 0 demo and other users, the application will undergo small modifications to address these responses. A key component that was brought up during the Rev 0 demo was the importance for users to be able to modify the data entries within the app with relative ease. Following the initial development schedule, the categorization model is constantly being refined and has a possibility of misclassification. The system should allow the users to make changes to these inputs manually to improve overall usability. A few other suggestions revolving around usability were small features such as autofilling item entries and providing adaptable user metrics.

One small change was made to the requirements regarding accuracy. The accuracy of the categorization model, which was initially targeted to reach around 90%, will now only require an 80% level of accuracy. This adjustment was made to enhance the model's generalization, which in turn reduces ambiguity during data classification. Allowing the model to be less rigid helps it handle real-world, diverse data more effectively. Additionally, the ability for users to modify misclassified entries will further help address any discrepancies and improve the overall accuracy.

At the moment, the core functional components of the application have been tested by a select group of users, providing feedback on its performance and usability. However, we have yet to conduct larger-scale usability testing to assess the flow of the entire app. Once we finalize the implementation of the smaller features, such as autofill options and adaptable user metrics, we plan to conduct further tests to optimize usability. This will involve a broader range of users to ensure that every criteria set for the system at the beginning of the project was met. By focusing on the complete flow of the



application, we aim to have testers evaluate all the smaller aspects of the app and how they work cohesively with the main functional components, ensuring that each part functions seamlessly together for an optimal user experience.

## **8 Automated Testing**

## **9 Trace to Requirements**

## **10 Trace to Modules**

## **11 Code Coverage Metrics**

## **References**

## Appendix — Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of 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?

While writing the VnV report, it was noteworthy that the collaborative effort among team members was highly effective. As we assigned each member a different area of testing, we ensured that the results brought into our meetings were insightful and understood by all of the other group members. This collaboration not only enhanced the quality of the content but also ensured that all perspectives in response to the feedback were considered. Finally, the thorough testing conducted prior to writing the report provided a good foundation to plan out the future steps of development. This allowed us to present concrete evidence of our findings, validating the findings outlined in the report.

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

One challenge we faced was managing all of the feedback received from Rev 0 as well as the feedback provided from the users. There were some differing opinions as to how we should address these responses moving forward. We held a meeting to discuss the feedback together, allowing us to prioritize the most important points and agree on how to approach them within our implementation. Another challenge was ensuring the

technical details, especially in the testing and results sections, were accurate. Some team members had more knowledge of their assigned testing areas, which could create gaps. To mitigate these gaps we had several meetings where team members shared their expertise, helping improve our understanding and ensure the report was well written and that all group members had a solid understanding of all testing aspects.

3. Which parts of this document stemmed from speaking to your client(s) or a proxy (e.g. your peers)? Which ones were not, and why?

Several sections of the VnV report were shaped by feedback from clients and peers. For example, the "Changes Due to Testing" section was influenced by the input provided by the instructor and TA during the Rev 0 demo, which highlighted the need for better usability features. This led us to adjust our focus to user-centered features. Peer reviews also helped refine our assessment of functional and nonfunctional requirements. On the contrary, some parts of the report, like the technical details of testing methods and code coverage metrics, were based on internal discussions and research. These sections relied on our team's expertise and the VnV Plan, as they required a deeper understanding of the system's technical aspects, while the feedback focused more on the project's overall direction.

4. In what ways was the Verification and Validation (VnV) Plan different from the activities that were actually conducted for VnV? If there were differences, what changes required the modification in the plan? Why did these changes occur? Would you be able to anticipate these changes in future projects? If there weren't any differences, how was your team able to clearly predict a feasible amount of effort and the right tasks needed to build the evidence that demonstrates the required quality? (It is expected that most teams will have had to deviate from their original VnV Plan.)

The content of the VnV report differed from the original plan due to continuous feedback, unforeseen technical challenges, and a greater emphasis on features revolving around usability. We had to modify our approach by expanding user-focused testing, introducing additional test cases, and addressing unexpected issues. These changes occurred as we gained new insights and encountered limitations in our initial assumptions. To improve future planning, we can allocate more buffer

time, conduct early individual pilot tests for core functional features, and allow for more flexibility in our initial VnV plan. Despite these deviations, our team effectively adapted to ensure a thorough validation process with sufficient coverage to guide further development.

<b>Test</b>	<b>Testing plan</b>
test-UAM-1: Account creation	test_users.py
test-UAM-2: User login	Manual testing
test-UAM-3: User logout	Manual testing
test-UAM-4: Account update	test_users.py
test-UAM-5: Authorization access	test_users.py
test-UAM-6: Password reset	Manual testing
test-IP-1: Image upload	Manual testing
test-IP-2: Image preview	Manual testing
test-IP-3: Image upload file size limit	Manual testing
test-MIS-1: Manual input expense	AddExpenseView.test.tsx, AddExpenseModal.test.tsx
test-FT-1: View spending history and trends	ExpensesList.test.tsx, HomePageMetricsBox.test.tsx, SpendingDetails.test.tsx
test-FT-2: Set and track budget	BudgetBoxDetails.test.tsx, MyBudgetsBox.test.tsx, NewBudgetModal.test.tsx
test-FT-3: Notification when user approaching limit	Not implemented

Table 4: Test traceability Table