

# 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

symbol	description
T	Test

[symbols, abbreviations or acronyms – you can reference the SRS tables if needed —SS]

# 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>1</b>
4.1	Usability . . . . .	1
4.2	Performance . . . . .	1
4.3	etc. . . . .	1
<b>5</b>	<b>Comparison to Existing Implementation</b>	<b>1</b>
<b>6</b>	<b>Unit Testing</b>	<b>1</b>
6.1	Front-end unit tests . . . . .	1
<b>7</b>	<b>Changes Due to Testing</b>	<b>1</b>
<b>8</b>	<b>Automated Testing</b>	<b>2</b>
<b>9</b>	<b>Trace to Requirements</b>	<b>2</b>
<b>10</b>	<b>Trace to Modules</b>	<b>2</b>
<b>11</b>	<b>Code Coverage Metrics</b>	<b>2</b>

## List of Tables

1	Unit Testing Table . . . . .	4
---	------------------------------	---

## List of Figures

This document ...

### **3 Functional Requirements Evaluation**

### **4 Nonfunctional Requirements Evaluation**

#### **4.1 Usability**

#### **4.2 Performance**

#### **4.3 etc.**

### **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 [1](#) for unit test traceability table

### **7 Changes Due to Testing**

[This section should highlight how feedback from the users and from the supervisor (when one exists) shaped the final product. In particular the feedback from the Rev 0 demo to the supervisor (or to potential users) should be highlighted. —SS]

- 8 Automated Testing
- 9 Trace to Requirements
- 10 Trace to Modules
- 11 Code Coverage Metrics

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

Test	Testing plan
test-UAM-1: Account creation	
test-UAM-2: User login	
test-UAM-3: User logout	
test-UAM-4: Account update	
test-UAM-5: Authorization access	
test-UAM-6: Password reset	Manual testing
test-IP-1: Image upload	Manual testing
test-IP-2: Image preview	
test-IP-3: Image upload file size limit	
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 1: Unit Testing Table