

WalletWatch – User Acceptance Testing (UAT) Plan

Team: Group 013-4

Semester: Fall 2025

Environment: Localhost (Docker), Chrome Browser

Testers: Team members + 1 external volunteer

Feature 1: Add Transaction Functionality

Feature Description

Users can manually add a transaction, including amount, category, description, and date. This transaction should appear in the user's dashboard and be saved to the database.

Test Data

| Field | Example Value |
|-------------|----------------|
| Amount | 25.00 |
| Category | Food |
| Description | Chipotle lunch |
| Date | 2025-10-12 |

Acceptance Criteria

1. User cannot submit the form unless all mandatory fields are completed.
2. Successful submission creates a new transaction row in the database.

3. Transaction instantly appears on the dashboard without refreshing (if implemented).
 4. Form provides error messages for invalid or missing fields.
 5. Page should not crash or reload unexpectedly.
-

UAT Test Cases

Test Case 1: Successful Add Transaction

Steps

1. Login as a valid user.
2. Navigate to **Add Transaction** page.
3. Fill out all fields with valid data.
4. Click **Submit**.

Expected Result

- User sees success confirmation.
- Transaction appears on dashboard.
- Transaction exists in `transactions` table.

Pass Criteria

- ✓ Transaction saved
 - ✓ UI updated
 - ✓ No errors
-

Test Case 2: Missing Required Fields

Steps

1. Leave one or more required fields blank.
2. Click **Submit**.

Expected Result

- Form displays descriptive errors (“Amount required”, etc.)
- No database insertion occurs.

Pass Criteria

- ✓ No transaction created
 - ✓ Correct error messages shown
-

Test Case 3: Invalid Input Format

Example: Non-numeric amount, date in wrong format, etc.

Expected Result

- System blocks the submission.
- Shows appropriate error feedback.

Pass Criteria

- ✓ Validation prevents bad data
 - ✓ No DB insertion
-
-

Feature 2: Settings Page Functionality

Feature Description

The settings page allows the user to update profile settings such as username, email, password, or theme preferences.

Test Data

| Field | Example Value |
|--------------|---------------------|
| New Username | Kishore2025 |
| New Email | kishore@example.com |
| New Password | Test1234! |
| Theme | Dark mode |

Acceptance Criteria

1. Form validates input before updating user information.
 2. Email must be in valid format.
 3. Password updates must follow password rules.
 4. Changes persist in the database.
 5. User receives confirmation feedback.
-

UAT Test Cases

Test Case 1: Update Username Successfully

Steps

1. Go to Settings.
2. Change username to a valid new one.
3. Submit form.

Expected Result

- System shows success message.
 - New username appears across the app.
 - Username updates in the `users` table.
-

Test Case 2: Update Email With Invalid Format

Steps

- Enter `kishore123@com`
- Click Save

Expected Result

- Error: “Invalid email format”
 - No DB update
-

Test Case 3: Update Password With Weak Password

Steps

- Enter password “123”
- Submit

Expected Result

- Error: “Password must be 8+ characters with letters & numbers”
- No DB update



Feature 3: Leaderboard Using Real User Data

Feature Description

Leaderboard displays real users sorted by total spending, total earnings, or “Pride Points” (however your group defines ranking).

Test Data

Manually inserted sample users:

| Username | Total Points |
|----------|--------------|
| Kishore | 120 |
| Jordan | 95 |
| Emma | 60 |

Acceptance Criteria

1. Leaderboard fetches live data from the DB.
2. Users must appear in correct sorted order (descending).
3. Ties are handled correctly.
4. Page loads without error even if DB has 0 users.
5. Design matches the project’s UI standards.

UAT Test Cases

Test Case 1: Leaderboard Shows Sorted User Data

Steps

1. Ensure DB has 3–5 users with different scores.
2. Open `/leaderboard`.
3. Observe order.

Expected Result

- Highest scoring user appears first
- Sorted descending
- No missing users

Test Case 2: Leaderboard With No Data

Steps

- Clear leaderboard data (or start with empty DB).
- Open leaderboard page.

Expected Result

- Page loads with friendly “No entries yet” message
- No crashes

Test Case 3: Data Updates After Adding User Activity

Steps

1. Add new transaction or points to a user.
2. Refresh leaderboard.

Expected Result

- Ranking updates correctly
 - Most recent totals are displayed
-
-



Test Environment

- **Browser:** Chrome v120+
 - **Backend:** Node.js 18 (Dockerized)
 - **Database:** PostgreSQL 15 (Dockerized)
 - **Testing Location:** Localhost (127.0.0.1:3000)
 - **Tools:** Docker Compose, VS Code, Postman (optional)
-



UAT Execution Summary (Week 4 – to be filled later)

| Feature | Tester | Pass/Fail | Note |
|-----------------|--------|-----------|------|
| | I | S | |
| Add Transaction | — | — | — |
| Settings Page | — | — | — |

Risks

Organizational Risks

- Team members unavailable during testing
- Limited access to user testers

Technical Risks

- Untested edge cases
- Unexpected DB failures
- Docker environment differences across machines

Business Risks

- Users misunderstanding UI
- Poor performance leading to bad user experience