

Testing Document and Specification

Test Plan

Team Nova
CS_451R, 2025

Haley Nilsen
Abdul Mohammed

Sid Nsude
Kaleb Flores Lopez

Introduction

This document outlines the test plan for the ClariFi application produced by Team Nova. As outlined in the project Requirements Document, this system needs to allow users to track their daily expenses and have the ability to set financial goals. This system will also provide a LLM Chat Assistant for users to ask questions. The testing activities discussed in this document will verify that the software for the ClariFi system meets the needs of the customer by verifying that the requirements for this system, as outlined in the Requirements Document, are met.

The testing routine will test the ClariFi system's response to logging in, sending a prompt to the LLM Chatbot, adding/editing a goal, and changing account information. The results of this testing procedure will enable the creators of this system to gauge project success as outlined in the Project Plan.

Terminology

Throughout this document the terms System, User, Personal User, Business-Admin User, Sub-User, and Site will be used frequently therefore, formal definitions will be given.

System- the database and bulletin board that supports the backend of the website

User- people affiliated with ClariFi or Commerce that will interact with the frontend

Personal User: Any non-business user who is using the application.

Business-Admin User: The user in a business with elevated privileges. Typically this is the business owner or team owner.

Sub-User: These are the team members on a business team. They are only allowed read-only access.

Site- the front-end website (main page)

Username/Password- unique identifiers that authenticate and validates a user

Items Tested

Items that will be tested during the testing phase as laid out by the Project Plan will be but are not limited to:

- Ability for an individual to login and be taken to their personalized dashboard
Test Case: 1.1.1.1, 1.1.5.1
- Ability for a user to select a prompt and chat with the LLM Assistant
Test Case: 1.1.2.1
- Ability for a user to add or edit a financial goal
Test Case: 1.1.3.1, 1.1.3.2
- Ability for users to update their name, email, and password in Settings
Test Case: 1.1.4.1, 1.1.4.2, 1.1.4.3

Items Not Tested

There are features that will not be included in the current testing procedure. This does not mean that these features will not be implemented, but that they have not been implemented and are not available for testing. Those features include but are not limited to:

- Active responses from the LLM. The user types any prompt and it will type out a response back to the user instead of the user clicking prompts.
- Session handling to remember user login credentials for a quicker login process

Approach

The overall method to this testing procedure is manual system testing. Each test case created will have a direct link to the requirements as laid out in the Requirements Document. Test cases that include similar Feature methods will be tested together (ex: the Personal User and Business User have access to the same tasks, so they are grouped into one test case). Examples of these features include logging into a personalized dashboard, creating/editing a goal, sending a prompt to the LLM Assistant, and updating information in settings. Each of these features will be tested within their own test cases.

Manual system testing will continue throughout the second and third iteration of the project. For each iteration, both old and newly implemented features will be tested. Adding new features or functionality can sometimes interfere with the functionality of old features and to ensure product/project success, all features implemented should function as intended throughout the life of the software.

Item Pass/Fail Criteria

The minimum requirements for this software system were laid out in the Requirements Document and the Project Plan outlined what the creators of the software considered project success.

Implemented features that meet the requirements as determined by the customer, meaning the feature does what the user wants it to do with very little difficulty, passes the testing procedure. Difficulty, as used here, is determined by user comprehension and user ability to use the feature with little to no training.

Features that contain major defects will fail the testing procedure and will be documented via an incident report and turned over to the developer for investigation and revision.

Test Deliverables

In addition to the Test Plan, other test deliverables include the Test Specification which outlines the specific test cases and expected results of each test, and Test reports which is comprised of Incidents, Defects and Changes.

Testing Tasks

The following list the testing deliverables and the activities required to produce the deliverable.

Deliverables	Activities
Test Plan	<ul style="list-style-type: none">• Analyze Requirements for System Features• Determine Testable/Non-Testable Features• Develop Approach/Method for testing• Determine Task and Estimate Efforts• Develop Schedule for Testing
Test Specifications	<ul style="list-style-type: none">• Analyze Requirements• Define Test Cases for Testable Features as Outlined by the Test Plan
Test reports	<ul style="list-style-type: none">• Implement Test Cases as Outlined by the Test Specifications• Document Incidents and Defects• Determine Severity of Incidents and Defects• Determine Changes that Need to be Made to System