A Test Plan

for ClinicTrends AI Project

Design and Creative Technologies

Torrens University, Australia

Group: 1

Students:

Luis Guilherme de Barros Andrade Faria - A00187785

Luong Hai Chau - A00117495

Subject Code:

SEP 401

Subject Name:

Software Engineering Principles

Assessment No.: 3

Title of Assessment:

Proposal

Lecturer: Dr. Ranju Mandal

Date: Aug 2025

Table of Contents

1. Introduction

This Test Plan outlines the systematic approach used to validate the functional and non-functional requirements of **ClinicTrends AI**, as defined in the revised **Software Requirements Specification (SRS**). The purpose of this plan is to ensure that all application features operate as intended, are free of defects, and provide a reliable and user-friendly experience for end users. ClinicTrends AI is a predictive analytics dashboard built for aesthetic clinics, enabling data-driven decision-making through Net Promoter Score (NPS) tracking, sentiment analysis, and machine learning-powered forecasting. To ensure the quality and success of the software, this test plan defines test cases covering the full system scope including data upload, NPS calculation, ML model training, translation, visualization, and system performance.

Each test case includes:

- A unique identification number
- A descriptive title
- Detailed instructions
- Assigned severity (Critical, Important, or Workaround)
- Expected and actual outputs (recorded during test execution)

This document will guide the verification and validation process and serve as a key reference in the final demonstration and presentation of the software. All tests have been designed to be reproducible, measurable, and traceable to their original requirements, ensuring that the software is robust, user-friendly, and ready for deployment.

2. Test Objectives

The objective of this test plan is to ensure the ClinicTrends AI system meets its functional and non-functional requirements as specified in the revised Software Requirements Specification (SRS). The key goals of this testing process include:

- **Validate Functional Requirements**: Ensure all core features—such as data uploading, NPS calculation, sentiment analysis, machine learning predictions, and visualizations—function correctly and align with stakeholder expectations.
- **Verify Data Integrity and Processing Accuracy**: Confirm that the system accurately processes large datasets, calculates correct NPS scores, and classifies feedback sentiment without data loss or misrepresentation.
- **Evaluate Machine Learning Performance**: Assess the performance and reliability of multiple ML models, ensuring predictions are accurate and computation remains within acceptable thresholds for real-time use.
- **Assess System Usability and Responsiveness**: Validate that the user interface is intuitive, responsive, and accessible across different devices and screen sizes, and provides clear feedback and notifications.
- **Test Security and Data Privacy Compliance**: Ensure no customer data is stored persistently, input validation is enforced, and personally identifiable information (PII) is handled in accordance with privacy principles.
- **Check Integration and End-to-End Workflow**: Verify that all modules, from data ingestion to prediction to reporting, operate seamlessly together without conflict or breakdowns.
- **Ensure Stability via Regression and Stress Testing**: Confirm the system remains stable after code changes and under concurrent access or large-scale data loads.

By meeting these objectives, the test plan ensures the final application is robust, secure, efficient, and ready for real-world deployment in clinical settings.

3. Test Scope

The scope of this test plan defines what will and will not be tested for **ClinicTrends AI**, ensuring comprehensive coverage of the application's features as described in the revised Software Requirements Specification (SRS).

In Scope

Testing will cover the following:

- **Functional Testing**
  - Data upload, validation, and error handling for CSV survey files.
  - Calculation and classification of Net Promoter Score (NPS) values.
  - Visualizations including NPS trends, sentiment distribution, and feature importance charts.
  - Sentiment analysis of customer comments and alignment with NPS scores.
  - Machine learning model training, evaluation, and prediction capabilities.
  - Translation features for both single and batch text input.
  - Filtering and alert mechanisms for identifying negative NPS trends.
- **Non-Functional Testing**
  - Performance and scalability with varying dataset sizes and concurrent user loads.
  - Security checks for data privacy, malicious file uploads, and input sanitization.
  - Usability and accessibility, including responsive design and assistive technology compatibility.
  - Integration with external APIs (Translation API, Discord webhook) and internal data flow.
- **Regression Testing**
  - Ensuring updates and changes do not break existing critical functionalities or the user interface.

Out of Scope

- Long-term monitoring or analytics in a live production environment.
- Integration with third-party clinic management systems beyond defined APIs.
- Deployment performance on infrastructure configurations outside the approved production setup.

4. Test Environment

Testing for **ClinicTrends AI** will be conducted in controlled environments that closely replicate the target production setup. Separate environments will be used for development, functional testing, integration testing, and performance validation to ensure accurate results and avoid interference between test activities.

4.1 Development Environment

- **Purpose:** Local feature implementation, unit testing, and initial validation.
- **Configuration:**
    - OS: Windows 11 / macOS 13+
    - Python: 3.9+
    - Streamlit: 1.45+
    - Required dependencies installed via requirements.txt
    - Local datasets (small to medium size)
    - Developer debugging tools enabled

4.2 Testing Environment

- **Purpose:** Isolated environment for executing functional, security, integration, and regression tests.
- **Configuration:**
    - Hosted on a virtual machine mirroring production specification
    - OS: Ubuntu 22.04 LTS
    - Python: 3.9+
    - Streamlit: 1.45+
    - Test datasets (1,000 to 100,000+ records)
    - Mock services for Translation API and Discord webhook
    - Logging and monitoring enabled for performance metrics

4.3 Performance & Load Testing Environment

- **Purpose:** Measure application response times, scalability, and stability under high-load conditions.

- **Configuration:**
    - Streamlit Cloud
    - Load simulation tools (JMeter)
    - Dataset size: 25,000 to 100,000+ records
    - Concurrent user simulation: 50+ virtual users
    - Resource monitoring tools for CPU, memory, and network usage

4.4 Production Environment

- **Purpose:** Final deployment environment for end-user access after successful testing.
- **Configuration:**
    - Streamlit Cloud
    - Secure HTTPS access
    - Environment variables for API keys and secure tokens
    - No persistent storage of uploaded datasets

5. Test Scenarios

5.1 Functional Testing

- **TC-001**: Application Startup and Initialization
    - **Description:** Verify application startup and initial page loading
    - **Steps:**
        1. Launch the ClinicTrends AI application
        2. Verify the application loads without errors
        3. Check that the sidebar navigation is displayed
        4. Confirm the home page content is visible
        5. Verify the application title and branding are correct
    - **Expected Output:** Application loads successfully with sidebar navigation and home page content
- **TC-002**: Page Navigation Functionality
    - **Description:** Test navigation between different application pages
    - **Steps:**
        1. Start on the Home page

2. Click on "**NPS Analysis**" in the sidebar
3. Verify the NPS Analysis page loads
4. Click on "**ML Model Comparison**" in the sidebar
5. Verify the ML Model Comparison page loads
6. Click on "**Translation**" in the sidebar
7. Verify the Translation page loads
8. Return to Home page and verify it loads correctly

- **Expected Output:** All pages load correctly with appropriate content and functionality

- **TC-003**: Sidebar Logo and Branding
  - **Description:** Verify sidebar logo and branding elements
  - **Steps:**

    1. Check that the ClinicTrends logo is displayed in the sidebar
    2. Verify the application title "ClinicTrends AI" is visible
    3. Confirm the version information is displayed

  - **Expected Output:** Logo, title, version, and link are all visible and functional

5.2 Data Upload & Processing

- **TC-004**: CSV File Upload Functionality
  - **Description:** Test CSV file upload and validation
  - **Steps:**

    1. Navigate to the NPS Analysis page
    2. Click on "Download Sample CSV Data" button
    3. Verify the sample file downloads successfully
    4. Use the file uploader to upload the downloaded sample file
    5. Verify the file is accepted and processed
    6. Check that a success message is displayed

- **Expected Output:** File uploads successfully with validation and processing confirmation

- **TC-005**: Invalid File Format Handling
  - **Description:** Test handling of invalid file formats
  - **Steps:**

    1. Navigate to the NPS Analysis page
    2. Attempt to upload a non-CSV file (e.g., .txt, .xlsx)
    3. Verify an appropriate error message is displayed
    4. Try uploading an empty CSV file
    5. Verify error handling for empty files
    6. Test with a CSV file missing required columns

  - **Expected Output:** Clear error messages for invalid file formats and missing data
- **TC-006**: Large File Processing
  - **Description:** Test processing of large CSV files
  - **Steps:**

    1. Create a CSV file with 50,000+ records
    2. Upload the large file to the NPS Analysis page
    3. Monitor processing time and memory usage
    4. Verify the file processes successfully within 30 seconds
    5. Check that all data is correctly loaded and displayed

  - **Expected Output:** Large files process within acceptable time limits with correct data handling

5.3 NPS Analysis Functionality

- **TC-007**: NPS Score Calculation
  - **Description:** Verify accurate NPS score calculation
  - **Steps:**

1. Upload a CSV file with known NPS data
2. Verify the NPS score is calculated correctly
3. Test with different scoring scales
4. Verify promoter, passive, and detractor classifications
5. Check that the NPS formula is applied correctly

- **Expected Output:** NPS scores calculated accurately with correct classifications

- **TC-008**: NPS Visualization Display
  - **Description:** Test NPS visualization charts and graphs
  - **Steps:**

    1. Upload data and navigate to NPS Analysis
    2. Verify the NPS donut chart is displayed
    3. Check that the chart shows correct proportions of promoters, passives, and detractors
    4. Verify the monthly NPS trend chart is displayed
    5. Test chart interactivity (hover tooltips, zoom, etc.)
    6. Verify chart colors and labels are correct

  - **Expected Output:** All visualizations display correctly with accurate data representation
- **TC-009**: Data Filtering Functionality
  - **Description:** Test year and store filtering capabilities
  - **Steps:**

    1. Upload data with multiple years and stores
    2. Use the year filter to select a specific year
    3. Verify the data and visualizations update accordingly
    4. Use the store filter to select a specific store

5. Verify the data and visualizations update accordingly

6. Test combination of year and store filters

7. Verify "All" option works for both filters

- **Expected Output:** Filters work correctly and update all relevant data and visualizations

5.4 Sentiment Analysis Functionality

- **TC-010**: Sentiment Analysis Processing
  - **Description:** Test sentiment analysis of customer comments
  - **Steps:**

    1. Upload data with customer comments

    2. Verify sentiment analysis is performed on comments

    3. Check that positive, negative, and neutral sentiments are correctly identified

    4. Verify sentiment distribution chart is displayed

    5. Test with various comment types (positive, negative, neutral, mixed)

  - **Expected Output:** Sentiment analysis accurately classifies comments with proper visualization
- **TC-011**: Word Cloud Generation
  - **Description:** Test word cloud visualization for comments
  - **Steps:**

    1. Upload data with customer comments

    2. Verify word cloud is generated and displayed

    3. Check that common words appear larger in the cloud

    4. Verify stop words are properly filtered out

    5. Test word cloud responsiveness and interactivity

  - **Expected Output:** Word cloud displays correctly with appropriate word sizing and filtering

- **TC-012**: Sentiment vs NPS Alignment
    - **Description:** Test analysis of sentiment and NPS alignment
    - **Steps:**

        1. Upload data with both NPS scores and comments
        2. Verify sentiment analysis is performed alongside NPS calculation
        3. Check for potential mismatches between NPS scores and sentiment
        4. Verify insights about sentiment-NPS alignment are provided
        5. Test edge cases where sentiment and NPS don't align

    - **Expected Output:** System identifies and highlights sentiment-NPS alignment patterns

5.4 Machine Learning Model Functionality

- **TC-013**: ML Model Training
    - **Description:** Test machine learning model training process
    - **Steps:**

        1. Navigate to ML Model Comparison page
        2. Upload training data with comments and NPS classifications
        3. Initiate model training for all four models
        4. Verify training progress is displayed
        5. Check that training completes within 30 seconds for 25,000+ records
        6. Verify training metrics are calculated and displayed

    - **Expected Output:** All models train successfully with performance metrics displayed
- **TC-014**: Model Performance Comparison
    - **Description:** Test ML model performance comparison and benchmarking
    - **Steps:**

        1. Complete model training for all four models

2. Verify performance comparison table is displayed

3. Check that accuracy, precision, recall, and F1-scores are shown

4. Verify the best performing model is highlighted

5. Test model comparison visualizations (bar charts, etc.)

6. Verify training time comparisons are displayed

- **Expected Output:** Comprehensive model comparison with accurate metrics and visualizations

- **TC-015**: Model Prediction Functionality
  - **Description:** Test ML model prediction capabilities
  - **Steps:**

    1. Train models with sample data

    2. Input new customer comments for prediction

    3. Verify predictions are generated for all models

    4. Check that prediction confidence scores are displayed

    5. Test with various comment types and lengths

    6. Verify prediction consistency across models

  - **Expected Output:** Accurate predictions with confidence scores for all models

- **TC-015**: Feature Engineering Validation
  - **Description:** Test feature engineering and optimization
  - **Steps:**

    1. Monitor feature engineering process during model training

    2. Verify optimal feature count is determined automatically

    3. Check that TF-IDF vectorization is performed correctly

    4. Test feature fusion for comment-score combination

5. Verify feature selection improves model performance

- **Expected Output:** Feature engineering optimizes model performance with automatic feature selection

5.5 Translation Functionality

- **TC-016**: Text Translation
  - **Description:** Test text translation capabilities
  - **Steps:**

    1. Navigate to Translation page
    2. Input text in English
    3. Select a target language (e.g., Spanish, French)
    4. Click translate button
    5. Verify translation is generated and displayed
    6. Test translation quality and accuracy
    7. Test with different text lengths and complexity

  - **Expected Output:** Accurate translations with proper language detection and formatting

5.6 Severity Matrix

TC	Name	Severity
TC-001	Application Startup & Initialization	Critical
TC-002	Page Navigation Functionality	Critical
TC-003	Sidebar Logo and Branding	Important
TC-004	CSV File Upload Functionality	Critical
TC-005	Invalid File Format Handling	Important
TC-006	Large File Processing	Important
TC-007	NPS Score Calculation	Critical
TC-008	NPS Visualization Display	Critical
TC-009	Data Filtering Functionality	Important
TC-010	Sentiment Analysis Processing	Critical
TC-011	Word Cloud Generation	Important
TC-012	Sentiment vs NPS Alignment	Important

TC-013	ML Model Training	Critical
TC-014	Model Performance Comparison	Critical
TC-015	Model Prediction Functionality	Important
TC-016	Feature Engineering Validation	Important

6. Performance Testing

6.1 Response Time Testing

- **TC-017**: Page Load Performance
  - **Description:** Test page load times meet performance requirements
  - **Steps:**
    1. Measure time to load Home page
    2. Measure time to load NPS Analysis page
    3. Measure time to load ML Model Comparison page
    4. Measure time to load Translation page
    5. Verify all pages load within 3 seconds
  - **Expected Output:** All pages load within 3 seconds under normal conditions
- **TC-018**: Data Processing Performance
  - **Description:** Test data processing performance with large datasets
  - **Steps:**
    1. Upload CSV file with 100,000+ records
    2. Measure time for data loading and processing
    3. Measure time for NPS calculation
    4. Measure time for sentiment analysis
    5. Verify processing completes within acceptable time limits
  - **Expected Output:** Large datasets process within 30 seconds with all calculations

6.2 Severity Matrix

TC	Name	Severity

TC-017	Page Load Performance	Important
TC-018	Data Processing Performance	Important

7. Acceptance Criteria

- All Critical test cases must pass
- All Important test cases must pass or have approved workarounds
- Performance requirements must be met
- All response time requirements met