

---

# Software Requirements Specification

for

## ClinicTrends AI Project

**Group: 1**

**Students:**

Luis Guilherme de Barros Andrade Faria - A00187785

Jing Feng Chin - A00178098

Luong Hai Chau - A00117495

**Subject Code:** SEP 401

**Subject Name:** Software Engineering Principles

**Assessment No.:** 2

**Title of Assessment:** Proposal

**Lecturer:** Dr. Ranju Mandal

**Date:** June 2025

# Table of Contents

Table of Contents .....	Error! Bookmark not defined.
Revision History .....	iii
<b>1. Introduction .....</b>	<b>1</b>
1.1 Purpose.....	1
1.2 Product Scope .....	1
<b>2. Overall Description .....</b>	<b>2</b>
2.1 Product Perspective.....	2
2.2 Product Functions .....	3
2.3 Stakeholders.....	3
2.4 Operating Environment.....	4
2.5 Design and Implementation Constraints.....	4
2.6 Assumptions.....	5
2.7 Dependencies .....	5
<b>3. External Interface Requirements .....</b>	<b>5</b>
3.1 User Interfaces .....	5
3.2 Software Interfaces .....	11
3.3 Communications Interfaces .....	12
<b>4. System Features .....</b>	<b>13</b>
4.1 Upload Survey Data.....	13
4.2 Visualize Historical Trends .....	13
4.3 Predict Sentiment Trends.....	14
4.4 Alert system .....	15
4.5 Export Report.....	15
4.6 Data Preview and Validation Report .....	16
<b>5. Other Nonfunctional Requirements .....</b>	<b>17</b>
5.1 Performance Requirements .....	17
5.2 Security Requirements .....	17
5.3 Software Quality Attributes .....	18
5.4 Business Rules .....	18
<b>6. References .....</b>	<b>19</b>
<b>Appendix A: Glossary.....</b>	<b>20</b>
<b>Appendix B: Analysis Models .....</b>	<b>21</b>
<b>7. Conclusion.....</b>	<b>28</b>

## Revision History

Name	Date	Reason For Changes	Version

## 1. Introduction

### 1.1 Purpose

**ClinicTrendsAI** is a software system developed to help businesses analyze and forecast customer satisfaction trends based on historical survey data, such as Net Promoter Score (NPS) feedback. The platform applies machine learning techniques to generate predictive insights, detect early signs of declining satisfaction, and highlight key factors influencing customer sentiment. By transforming raw survey data into actionable intelligence, ClinicTrendsAI enables business managers to make informed, proactive decisions aimed at improving service quality, enhancing client retention, and sustaining long-term business performance.

### 1.2 Product Scope

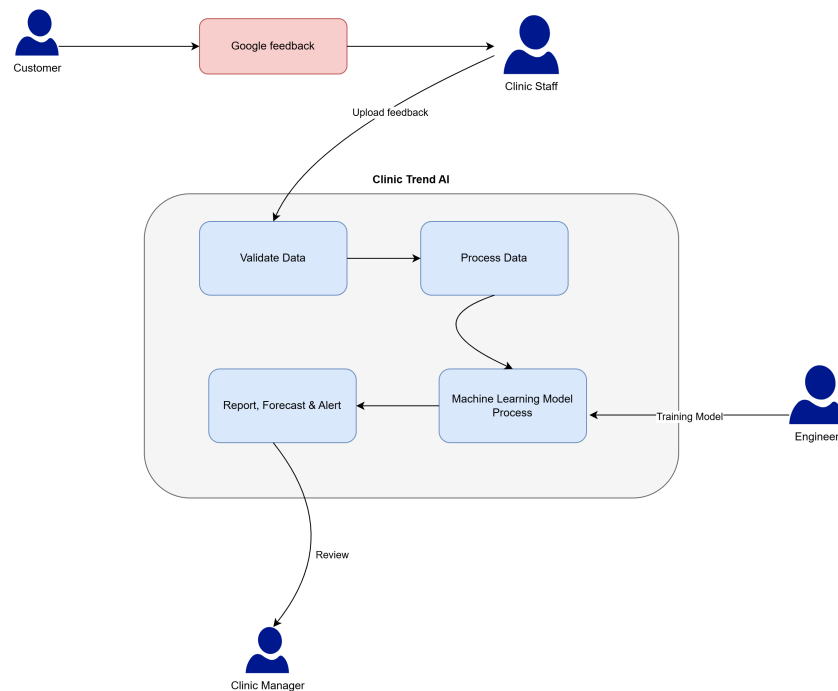
**ClinicTrendsAI** will provide the following core capabilities:

- **Ingest survey datasets** in CSV formats for flexible data integration.
- **Predict future customer satisfaction scores** using machine learning regression models to anticipate emerging trends.
- **Visualize satisfaction trends and alert thresholds** through interactive graphs and automated flagging of concerning patterns.
- **Identify key drivers of satisfaction fluctuations**, using feature importance analysis for actionable insights.
- **Generate exportable reports** to support business decision-making and operational improvements.

## 2. Overall Description

### 2.1 Product Perspective

**ClinicTrendsAI** is a self-contained software product developed as a standalone application to support businesses in proactively managing customer satisfaction. It is not part of a larger existing product family nor a replacement for any legacy system. Instead, it addresses a current gap in affordable, intelligent feedback analysis tools tailored to the needs of small and medium-sized businesses. The application is designed to function independently, requiring only local survey data inputs, including CSV files. Built using **Python** and **Streamlit**, ClinicTrendsAI provides an intuitive interface suitable for non-technical users. It encompasses a lightweight deployment model that ensures accessibility via local or cloud-hosted environments. This product serves as the first iteration in a potential roadmap of intelligent customer insight tools designed for any service sector.



**Figure 1:** The system context diagram of Clinic Trend AI

Figure 1 depicts the high-level view of the Clinic Trend AI system in relation to its actors and data flows. It defines the system boundaries and interactions between the system and its users.

## 2.2 Product Functions

ClinicTrendsAI provides a set of key functions designed to enable business managers to analyze, monitor, and act on customer satisfaction data. The system's functionality is organized into the following core features:

- **User Access Control:** Restrict access to sensitive reports or data
- **Clean & Preprocess Data:** Automatically handle missing values, inconsistent formats, and invalid data entries
- **Upload historical survey data:** Allow users to upload customer satisfaction dataset in .csv format
- **Visualize Historical Trends:** Display satisfaction scores using interactive charts
- **Predict Sentiment Trends:** Applied machine learning to forecast future satisfaction levels
- **Alert System:** Automatically flags predicted satisfaction drops below a critical threshold
- **Export Report:** Allow users to export key analysis outputs and insights

## 2.3 Stakeholders

- **Business Manager:**
  - Role: Primary end user responsible for business operations and customer experiences
  - Expectations: Expect clear insights into satisfaction trends, risks, and improvement opportunities
- **Operation Analyst:**
  - Role: Analyst or data specialist reviewing performance metrics across stores
  - Expectations: Require access to raw data, trends, and feature insights to inform strategic actions.
- **Developer**
  - Role: Responsible for Streamlit UI and features analysis

- Expectations: Need clear UI requirements, data formats, and interaction logic
- **Machine Learning Engineer:**
  - Role: Implement prediction models and feature analysis
  - Expectations: Require clean datasets, evaluation criteria, and feedback on model performance
- **System Admin:**
  - Role: Maintain the application
  - Expectations: Require the design documentation in the details

## **2.4 Operating Environment**

- Web browser (Chrome, Firefox)
- Backend hosted locally or on cloud VM
- Python 3.10+, Streamlit, pandas

## **2.5 Design and Implementation Constraints**

- Technology Stack Constraint
  - The system is developed using Python and Streamlit
  - Machine learning functionality should be open-source libraries such as scikit-learning and pandas
  - The system is a web application that allows users to access without limitation of platforms
- Data Constraint
  - Input data must be provided in CSV format
  - Required schema includes valid columns, including Date, Store, Score & Comment
  - Must handle CSVs up to 200MB
- Machine Learning Model Constraint
  - Model retraining must occur asynchronously
  - Predictions limited to linear regression and decision tree models in MVP

## 2.6 Assumptions

- Feedback data (e.g., NPS scores) will be representative and sufficient for generating reliable predictions.
- Uploaded data will conform to a consistent structure, containing at minimum the fields: **Date**, **Store**, **Score**, and **Comment**.
- The target users are not highly technical; therefore, the user interface will be kept simple and intuitive.
- Users, including **Businesses Staff & Managers**, have access to the Internet to use the system via a modern browser with the latest versions such as Chrome and Firefox.
- Machine learning models will be trained using clean, pre-processed data prior to integration into the application

## 2.7 Dependencies

- **Python 3.8+** and required libraries (streamlit, pandas, scikit-learn, etc.) must be installed
- **Streamlit Cloud** or local machine with compatible OS for running the application.
- **CSV** files must be supplied by business staff as the primary input source.
- Hugging Face Sentiment API for NLP-based analysis (optional advanced feature)
- Availability of trained ML models or the ability to train models using sample datasets.
- End users will manually upload datasets; no automated ETL or live data integration is currently implemented.

## 3. External Interface Requirements

### 3.1 User Interfaces

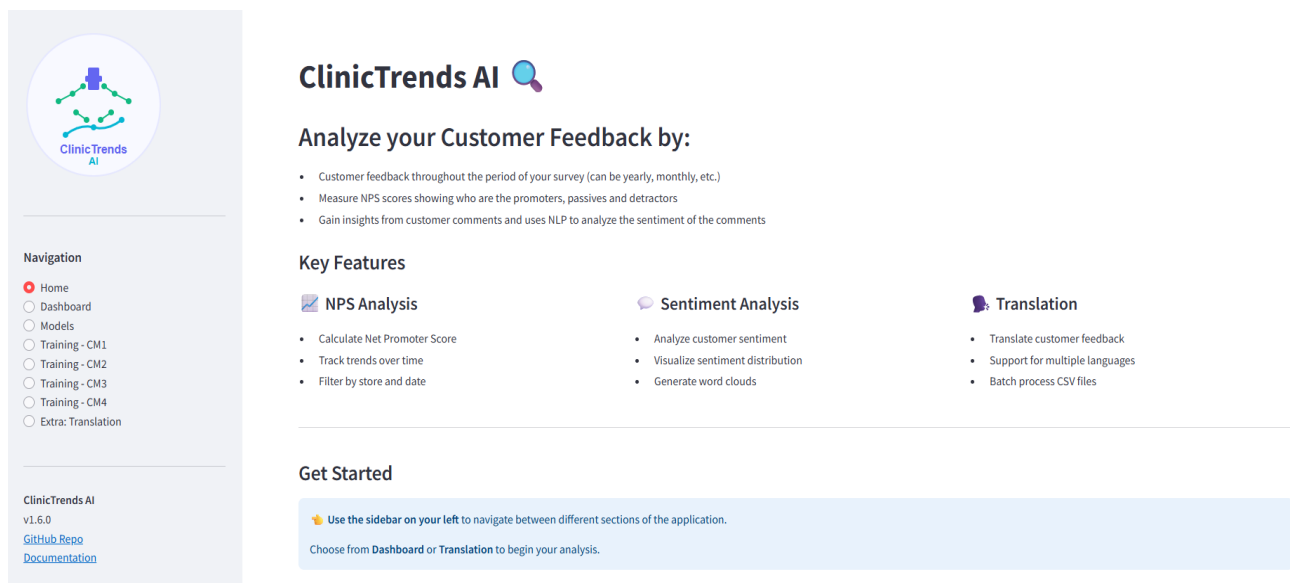
The user interface for *ClinicTrendsAI* is designed to be clean, intuitive, and accessible to non-technical users such as business managers and staff.



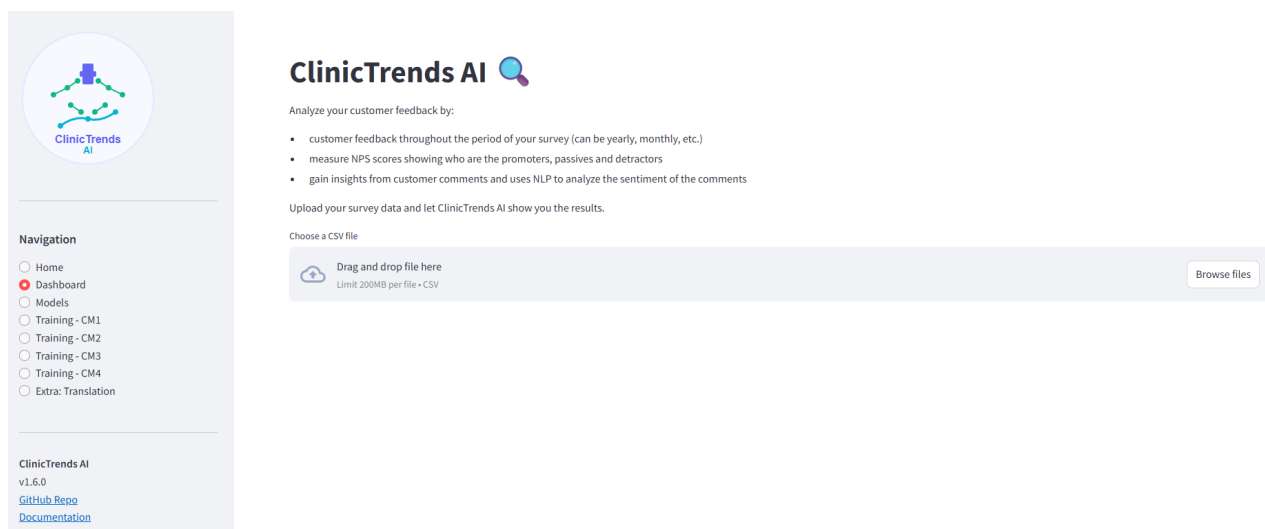
The application uses the **Streamlit web framework**, which delivers a browser-based interface that requires no installation beyond launching a Python script.

## Interface Characteristics

- **Platform:** Web-based interface served via localhost or Streamlit Cloud.
- **Layout:** Left-hand vertical navigation sidebar and dynamic content area on the right.
- **Design Style:** Follows the minimal, consistent aesthetic of Streamlit apps. Uses headers, charts, buttons, and form inputs in a visually structured layout.



**Figure 2:** Home Page



**Figure 3:** Dashboard UI:

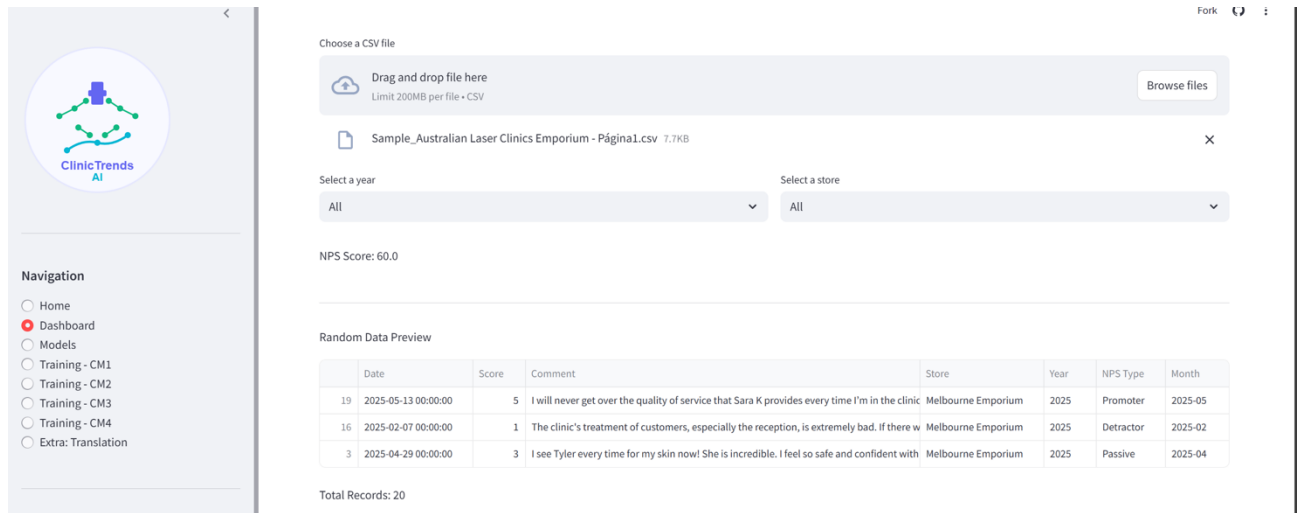
Figures 2 & 3 are designed interfaces of Clinic Trend AI system, following the characteristics of an application in the Streamlit platform.

## Home

- The Home page shows a general introduction to ClinicTrends AI, key features of the system, and a Get Start section for brief guidance. (see Figure 2).

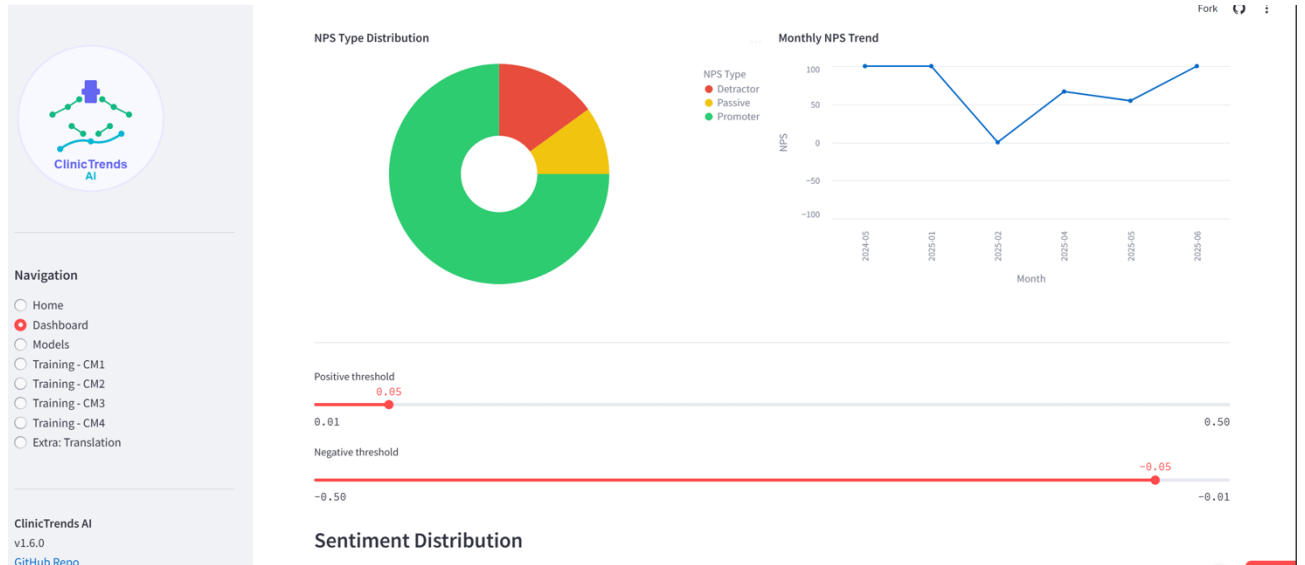
## Dashboard

- The dashboard section can be accessed using the left-hand side bar.
- The page will display a brief description of the type of analysis that will be conducted. (see Figure 3).
- After uploading a valid CSV file, the page will extend and display 2 dropdown menus, allowing users to select year and store for their preference. (see figure 4).



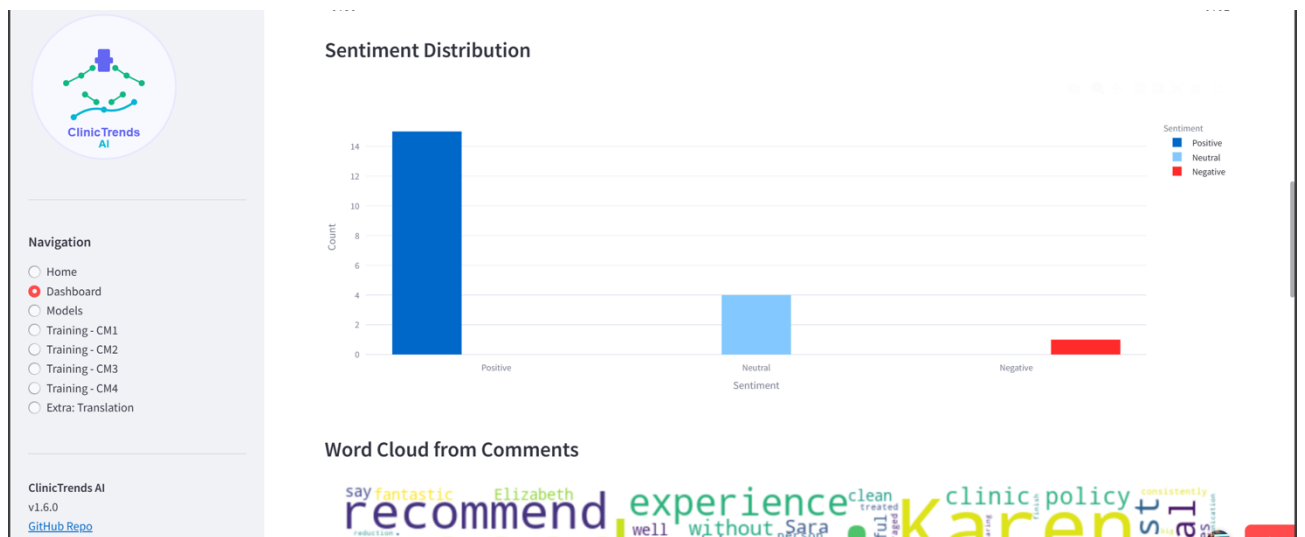
**Figure 4:** Example of processed data, displaying first scroll on the applications' page.

- NPS score and quantity of total records will be shown. There will be a random data preview displaying the information from the original CSV files with updated NPS in the file. (see figure 4).
- Scrolling down, the page will display a pie chart for NPS type distribution, and a line chart for Monthly NPS Trend. Positive threshold and negative threshold are also displayed just right below the charts. (see figure 5).



**Figure 5:** Example of processed data, displaying second scroll on the applications' page.

- A sentiment distribution bar chart will be displayed to provide a visual summary of positive, negative, and neutral comments (see Figure 6).



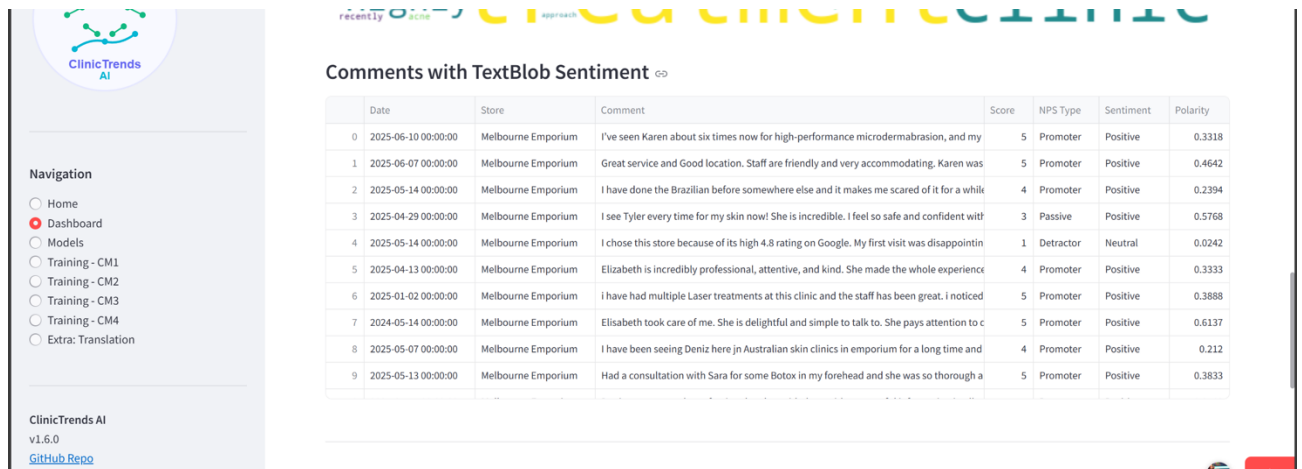
**Figure 6:** Example of graphics displayed in the third scroll on the applications' page.

- Below that, a word cloud will be generated to highlight the most frequently used words in the comments. (see Figure 7).



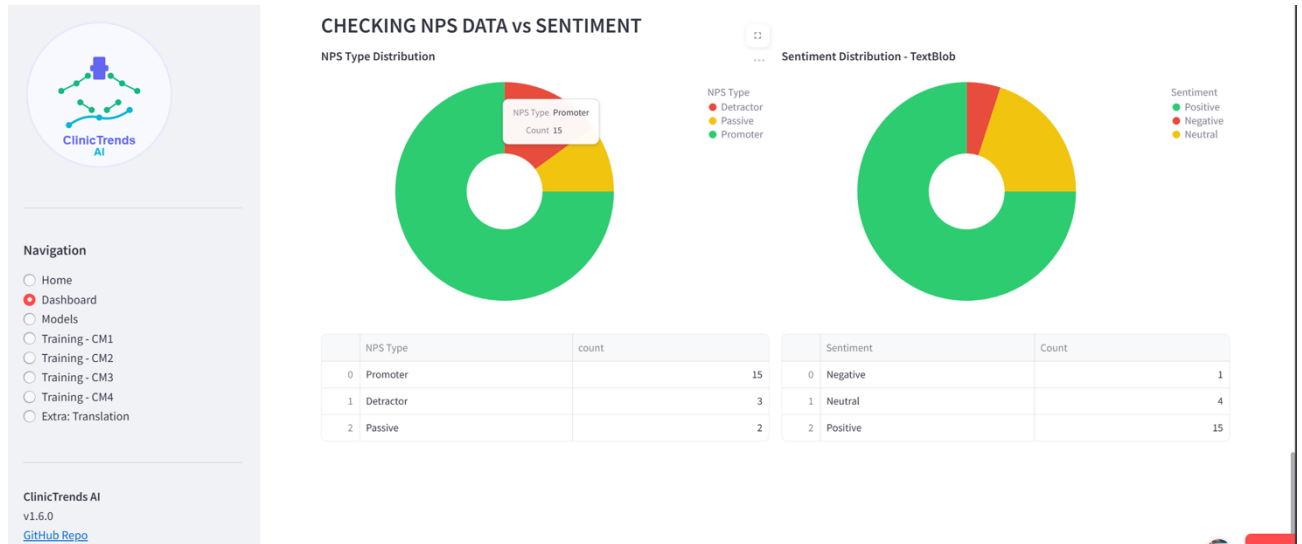
**Figure 7:** Example of word cloud shown in the fourth scroll on the applications' page.

- The page will also include a table of analyzed comments. Each tagged with sentiment score as determined by TextBlob.(see figure 8).



**Figure 8:** Example of table data displayed in the fifth scroll on the applications' page.

- Finally, the page will present a comparison between NPS scores and sentiment distribution, helping users explore potential relationships between quantitative ratings and textual feedback. (see Figure 9).



**Figure 9:** Example of the charts present at the sixth scroll on the applications' page.

## Translation (Figure 10 & 11)

- In the Translation page, it will display 2 tabs, that are Text Translation and File Translation.  
(see figure 10).

**Text & File Translation**

Translate customer feedback between different languages. You can either translate individual pieces of text or process entire CSV files.

**Text Translation** | File Translation

**Text Translation**

Quickly translate individual pieces of text.

Enter text to translate:

From:  To:

**Translate**

How to use

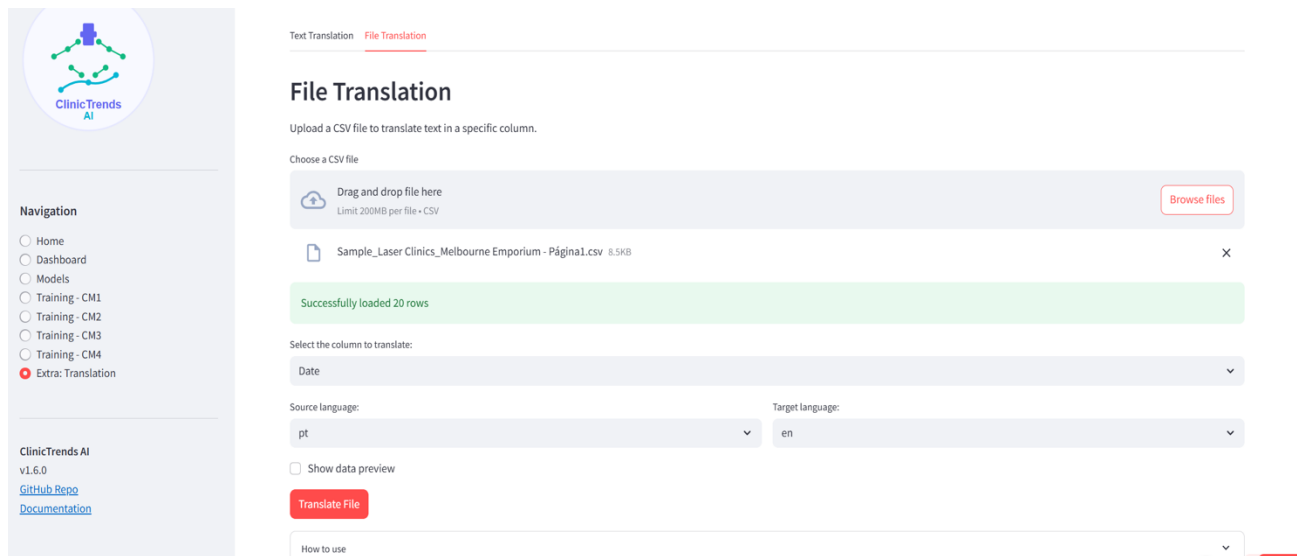
**Navigation**

- ☐ Home
- ☐ Dashboard
- ☐ Models
- ☐ Training - CM1
- ☐ Training - CM2
- ☐ Training - CM3
- ☐ Training - CM4
- ☒ Extra: Translation

ClinicTrends AI  
v1.6.0  
[GitHub Repo](#)  
[Documentation](#)

**Figure 10:** Example of Text Translation input page.

- When users select Text Translation, they can manually enter the text they wish to translate. Two dropdown menus, from and to, are provided, allowing users to choose the original and target language. (see figure 10).
- Choosing File Translation allows users to upload a CSV file and specify which column of text they want to translate. An optional "Show Data Preview" checkbox lets users view their file before translating. (see figure 11).



**Figure 11:** Example of successful upload of csv file for translation.

### 3.2 Software Interfaces

This section outlines the key software components, tools, libraries, and APIs that *ClinicTrendsAI* depends on, along with their communication patterns, data exchange mechanisms, and integration characteristics

- Libraries and Tools:
  - Streamlit 1.45.1
  - Pandas 2.3.0
  - TextBlob 0.19.0
  - Altair 5.5.0

- Scikit-learn 1.5.2
- Transformers 4.52.4
- Deep-translator 1.11.4
- Tenacity 9.1.2

### **3.3 Communications Interfaces**

**ClinicTrendsAI** is primarily a web-based application. It optionally utilizes cloud services and APIs depending on deployment settings and feature activation. This section describes the communication interfaces, protocols, standards, and security considerations for all communication between the system and external components.

- **Web Browser Interface**

- **Interface Type:** Localhost (default) or Streamlit Cloud
- **Protocol:** HTTP (default), HTTPS (Streamlit Cloud)
- **Requirements:**
  - Compatible with modern browsers: Chrome, Firefox, Safari, Edge
- **Purpose:** Provides the graphical user interface for all system interactions

- **External API Communication**

- **Hugging Face API**
  - **Protocol:** HTTPS
  - Purpose: Sentiment analysis
- **Google Translate API**
  - Protocol: HTTPS

## 4. System Features

### 4.1 Upload Survey Data

#### 4.1.1 Description and Priority

This function allows users to upload historical customer feedback in .csv format, which serves as the primary data source for all downstream analytics. This feature is assigned **High Priority** because it is the system's entry point for processing and generating insights.

#### 4.1.2 Stimulus/Response Sequences

- User selects a file via the upload interface
- System validates file structure and content.
- If valid, the system loads data into the active session for further processing.
- If invalid, an error message is shown with instructions.

#### 4.1.3 Functional Requirements

- **REQ-01:** The system shall support the upload of .csv file with UTF-8 encoding.
- **REQ-02:** The system shall validate file structure, ensuring the presence of columns: **Date**, **Store**, **Score**, and **Comment**.
- **REQ-03:** The system shall display an appropriate error message if the uploaded file is invalid or missing required columns.
- **REQ-04:** The system shall preview the first five rows of data after successful upload for user confirmation

### 4.2 Visualize Historical Trends

#### 4.2.1 Description and Priority

This feature presents satisfaction data (e.g., NPS) over time using interactive visualizations. Users can segment data by store, staff group, or date range to identify patterns. This is a **High Priority** feature due to its role in monitoring ongoing satisfaction levels.



#### **4.2.2 Stimulus/Response Sequences**

- User uploads a valid dataset.
- The system generates line/bar charts representing satisfaction scores.
- User filters data by store, date range, or category.
- Visualizations are updated dynamically.

#### **4.2.3 Functional Requirements**

- **REQ-05:** The system shall generate Altair-based interactive time-series charts to visualize NPS scores.
- **REQ-06:** The system shall allow users to filter visualizations by store, date range, and customer group.
- **REQ-07:** The system shall compute and display average NPS per period selected.
- **REQ-08:** The system shall provide tooltips and legends for chart clarity.

### **4.3 Predict Sentiment Trends**

#### **4.3.1 Description and Priority**

This feature uses a machine learning model to predict future satisfaction scores based on historical data and comment sentiment. Assigned **High Priority** due to its forecasting function.

#### **4.3.2 Stimulus/Response Sequences**

- User clicks “Analyze Sentiment” button
- System preprocesses data
- Model returns sentiment analysis + scores + confidence intervals

#### **4.3.3 Functional Requirements**

- **REQ-09:** The system shall apply a regression model (e.g., Linear Regression using scikit-learn) to predict satisfaction scores.

- **REQ-10:** The system shall output a 95% confidence interval for predicted values.
- **REQ-11:** The system shall allow users to adjust forecast parameters (e.g., time window).
- **REQ-12:** The system shall preprocess text using NLP techniques prior to prediction if comment analysis is enabled.

## **4.4 Alert system**

### **4.4.1 Description and Priority**

This component monitors forecasted satisfaction scores and alerts the user when values drop below a critical threshold. Priority is set to **Medium**, as it supports proactive decision-making.

### **4.4.2 Stimulus/Response Sequences**

- Model detects predicted NPS below the threshold (e.g., 50).
- UI highlights the issue via visual cue (e.g., red icon, warning message).
- Top contributing factors are listed.

### **4.4.3 Functional Requirements**

- **REQ-13:** The system shall flag predicted scores that fall below a defined threshold (default: 50).
- **REQ-14:** The system shall display visual alerts (e.g., color-coded warning indicators) when a risk is detected.
- **REQ-15:** The system shall list the top 3 features contributing to the drop using feature importance metrics.

## **4.5 Export Report**

### **4.5.1 Description and Priority**

Allows users to export analysis of results, predictions, and trend charts in PDF or CSV formats. This is a **Medium Priority** feature that supports documentation and sharing.

#### **4.5.2 Stimulus/Response Sequences**

- User clicks the “Export Report” button.
- System compiles insights and visualizations.
- The system generates downloadable files.
- File is saved to the user’s device.

#### **4.5.3 Functional Requirements**

- **REQ-16:** The system shall generate downloadable reports in .pdf or .csv format.
- **REQ-17:** Each report shall include:
  - Timestamp of generation
  - Store/branch identifiers
  - Predicted NPS scores
  - Historical NPS chart
  - Alert summaries (if any)
- **REQ-18:** The system shall allow the user to select the file format (PDF or CSV) prior to export.

### **4.6 Data Preview and Validation Report**

#### **4.6.1 Description and Priority**

This feature enhances trust and transparency in data handling by allowing users to preview uploaded data and review validation results. This is **Medium Priority**, supporting usability and error prevention.

#### **4.6.2 Stimulus/Response Sequences**

- User uploads dataset

- System validates and loads data
- A preview table and validation report are displayed

#### **4.6.3 Functional Requirements**

- **REQ-22:** The system shall show a preview of the first five rows of data uploaded.
- **REQ-23:** The system shall validate schema and report issues such as missing values, incorrect types, or duplicates.
- **REQ-24:** The system shall display a summary report showing row count, number of valid entries, and detected warnings.

### **5. Other Nonfunctional Requirements**

#### **5.1 Performance Requirements**

- The system shall return predictions within 10 seconds of data upload.
- The system shall upload and parse survey files up to 200MB in 15 seconds.
- The system shall validate uploaded data schema within 3 seconds after file upload.
- The system shall display a preview of uploaded data (first 5 rows) in under 2 seconds.
- The system shall export report in PDF/CSV within 5 seconds after the users initiate the export action.

#### **5.2 Security Requirements**

- The system shall encrypt all uploaded files in transit via HTTPS.
- User authentication should be required for access to sensitive reports.
- The system will not store any data uploaded by users.

### **5.3 Software Quality Attributes**

- System shall achieve 95% uptime.
- User interactions should not require more than 2 clicks for any core feature.
- The user interface shall be simple and straightforward, no additional technical training required for users.
- The system shall provide clear tooltips to guide users throughout their sessions.
- The system shall run on all modern web browsers (Chrome, Edge, Firefox, Safari) without any prior installation.

### **5.4 Business Rules**

- The system should handle incomplete rows gracefully.
- Only authorized users may export reports.
- The system shall allow users to download all analyzed data from the current session.
- The system will not retain any uploaded or processed data once the user session ends.

## 6. References

- Wiegers, K. E. (1999). Software Requirements Specification Template. Retrieved from Karl Wiegers SRS Templates
- Streamlit Inc. (2025). Streamlit Documentation. <https://docs.streamlit.io/>
- Scikit-learn Developers. (2025). Scikit-Learn Documentation. <https://scikit-learn.org/stable/documentation.html>
- Altair Developers. (2025). Altair Documentation. <https://altair-viz.github.io/>
- Hugging Face. (2025). Hugging Face Transformers Documentation. <https://huggingface.co/docs/transformers>
- Pandas Development Team. (2025). pandas Documentation. <https://pandas.pydata.org/docs/>

**Appendix A: Glossary**

Term	Definition
SEP401	Software Engineering Principles subject we are studying
NPS	Net Promoter Score, a metric to gauge customer loyalty and satisfaction.
ML	Machine Learning: computer algorithms that learn patterns from data.
MVP	Minimum Viable Product: the simplest version of a product delivering usable value.
API	Application Programming Interface: a set of functions enabling software to communicate.
Altair	A Python library for declarative statistical visualization.
CSV	Comma-Separated Values: a text format for tabular data storage.
Dashboard	The main visual interface in Streamlit where charts, tables, and results are displayed.
Data Frame	A two-dimensional labeled data structure used in pandas for data manipulation.
ETL	Extract, Transform, Load: the process of moving and transforming data.
Regression Model	A type of ML model predicting numeric values based on input features.
Scikit-learn	A Python library for machine learning tasks.
Sentiment Analysis	A process of determining emotional tone from text data.
Streamlit	A Python framework for creating data-driven web applications.

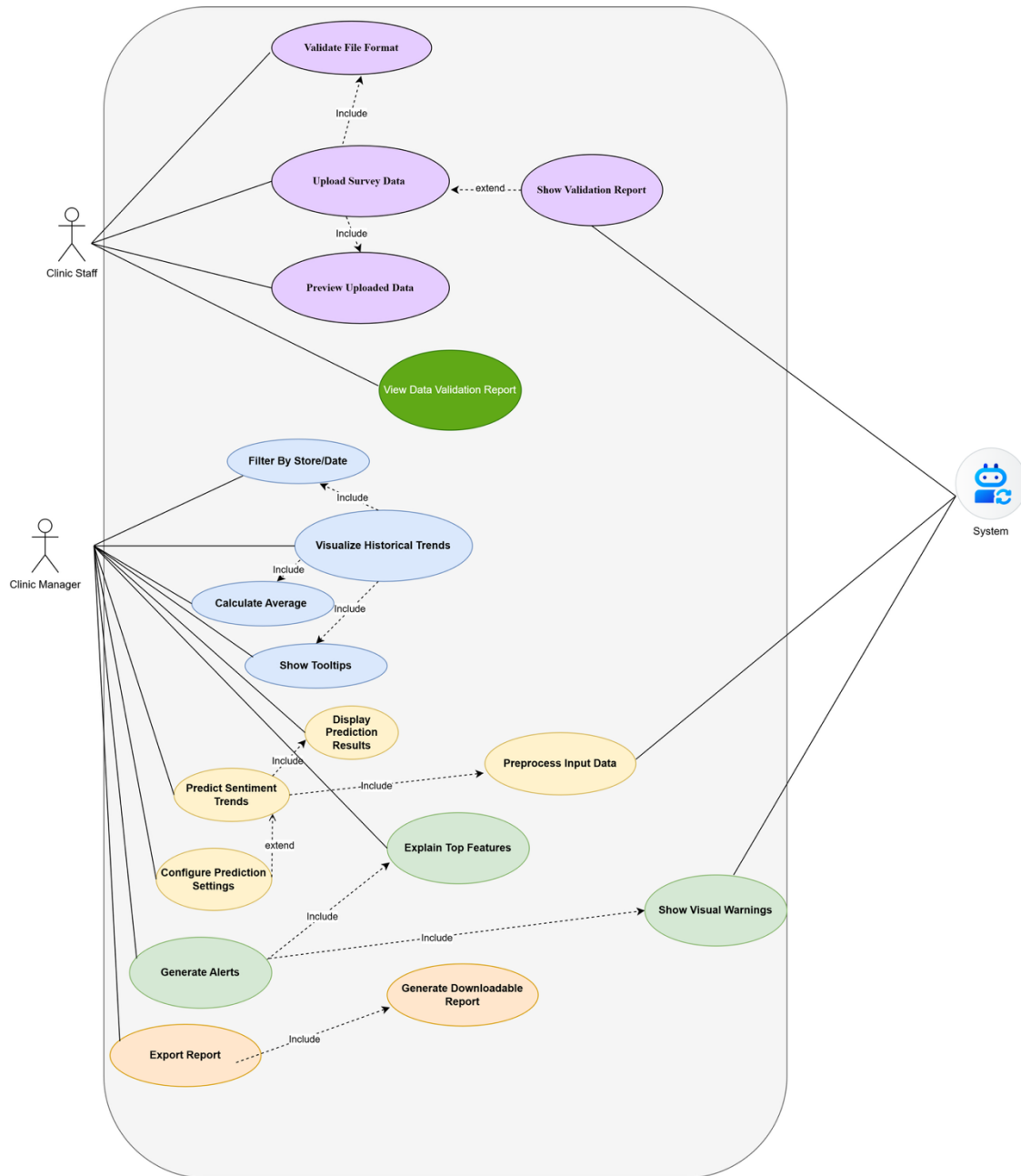
**Appendix B: Analysis Models****Use cases:**

#	Name	Description
<b>UC-01</b>	Upload Survey Data	User uploads a CSV file containing historical customer feedback for analysis
UC-01.1	Validate File Format	System checks if the uploaded file is valid and contains required fields.
UC-01.2	Preview Uploaded Data	Displays the first five rows of uploaded data for user confirmation
UC-01.3	Show Validation Report	Displays warnings about missing or malformed data.
<b>UC-02</b>	Visualize Historical Trends	Displays satisfaction data (e.g., NPS) in interactive charts by store, period, or group
UC-02.1	Filter by Store/Date	User filters visualizations based on store, date range, or customer group.
UC-02.2	Calculate Average NPS	System computes average satisfaction scores over selected period
UC-02.3	Show Chart Tooltips & Legends	The system provides visual cues for better interpretation of charts.
<b>UC-03</b>	Predict Sentiment Trends	Uses a machine learning model to forecast satisfaction scores
UC-03.1	Preprocess Input Data	Cleans and structures the data before analysis.
UC-03.2	Configure Prediction Settings	Allows users to modify prediction parameters like forecast range.
UC-03.3	Display Prediction Results	Presents forecast results, including confidence intervals.
<b>UC-04</b>	Generate Alerts	Triggers when predicted NPS falls below a defined threshold.

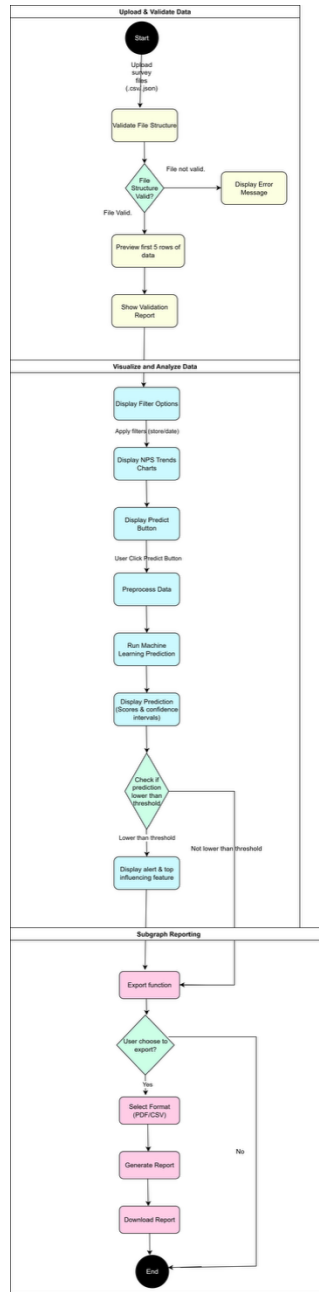


UC-04.1	Show Visual Warnings	Displays alert messages or red icons in the interface.
<b>UC-05</b>	Export Report	Allows users to export current insights, predictions, and visualizations.
UC-05.1	Select Export Format	Enables users to choose between CSV and PDF.
UC-05.2	Generate Downloadable Report	Compiles charts, alerts, and metrics into a formatted file.
<b>UC-06</b>	View Data Validation Report	Provides detailed summary of uploaded data quality and schema issues.

## Use Case Diagram:

**Figure 12:** Comprehensive Use Case Diagram displaying ClinicTrendsAI features.

## Activity Diagram:



**Figure 13:** Activity Diagram for loading feedback data and running sentiment analysis in ClinicTrendsAI.

## Data Modelling Concepts:

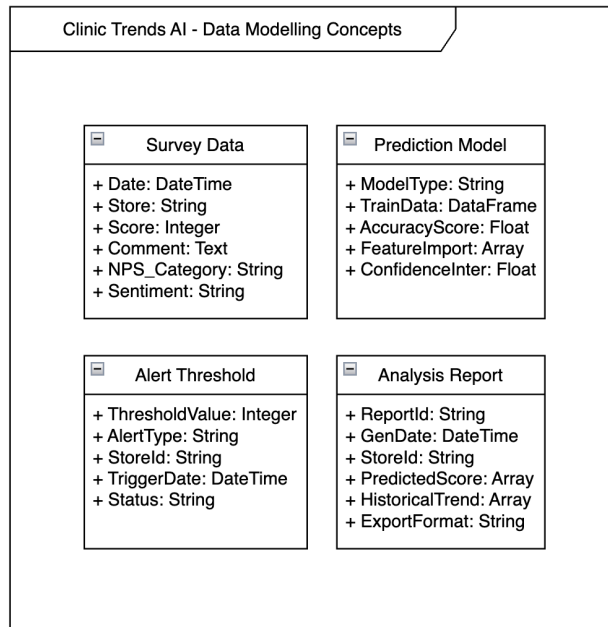


Figure 14: Data Modelling Concepts for ClinicTrendsAI.

## Class Diagram:

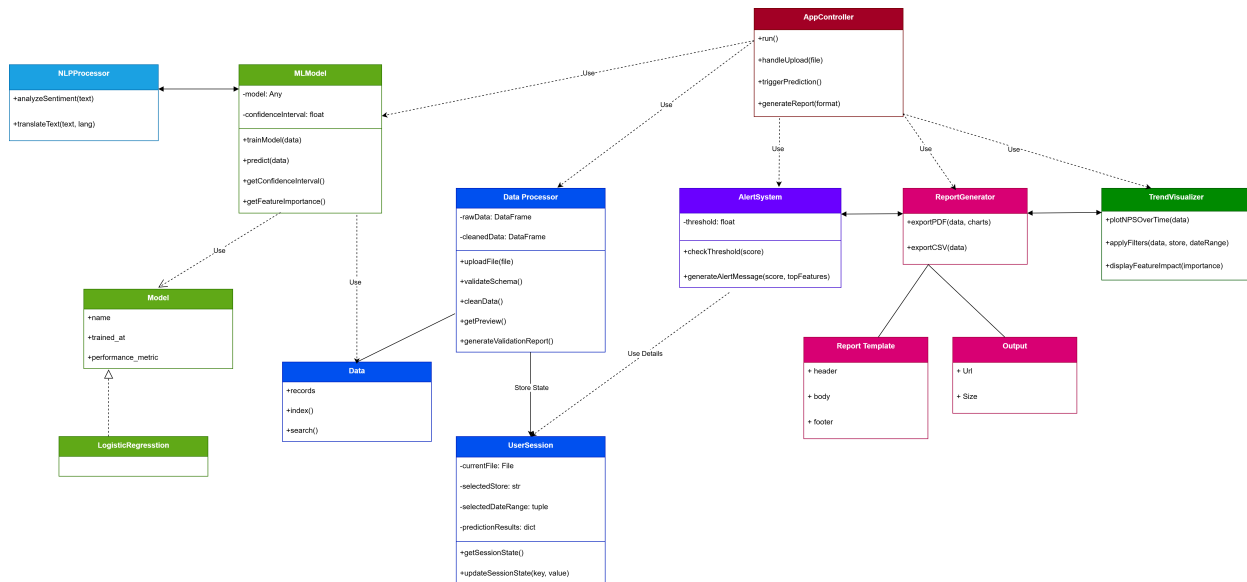
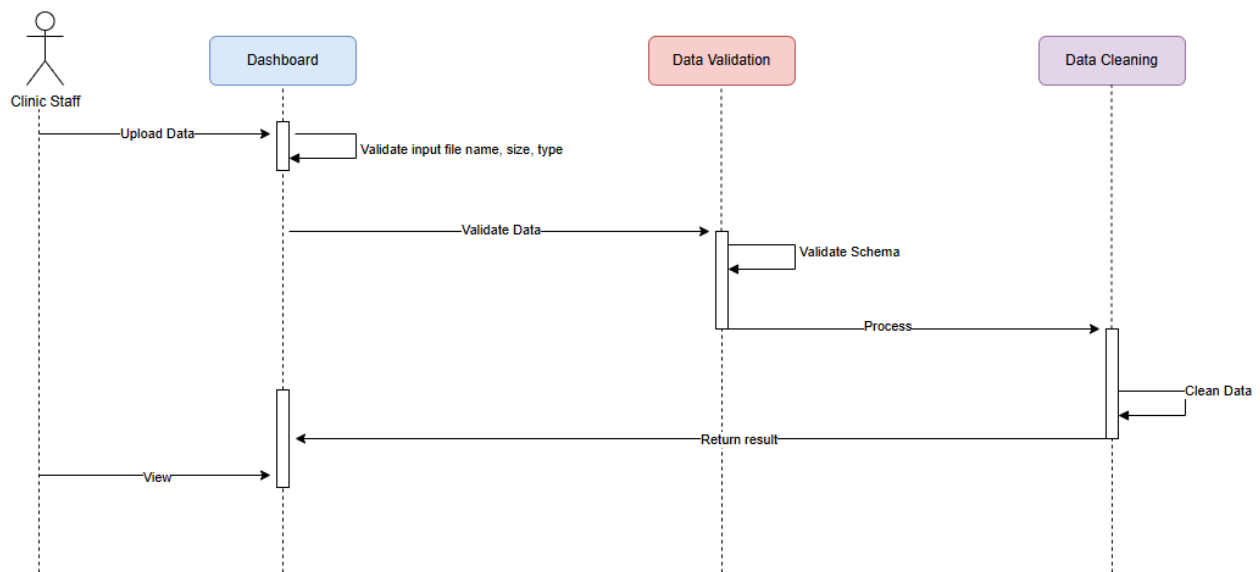


Figure 15: The Class Diagram for ClinicTrendsAI.

- **Data Handling** is managed by DataProcessor, which loads, cleans, validates, and previews survey data. It interacts with the Data class for basic operations and UserSession to maintain application state.
- **Machine Learning** functionality is encapsulated in MLModel, which uses sentiment analysis from NLPProcessor and inherits model metadata from the abstract Model class (e.g., LogisticRegression).
- **Alerting** is handled by the AlertSystem, which checks prediction thresholds and generates messages when customer satisfaction is expected to drop.
- **Visualization** of NPS trends is managed by TrendVisualizer, which supports filtering and feature impact chartseport Generation is carried out by ReportGenerator, which uses ReportTemplate and Output components to export data in PDF or CSV formats.
- **UserSession** preserves user-specific selections (e.g., uploaded files, filters, predictions) across interactions.

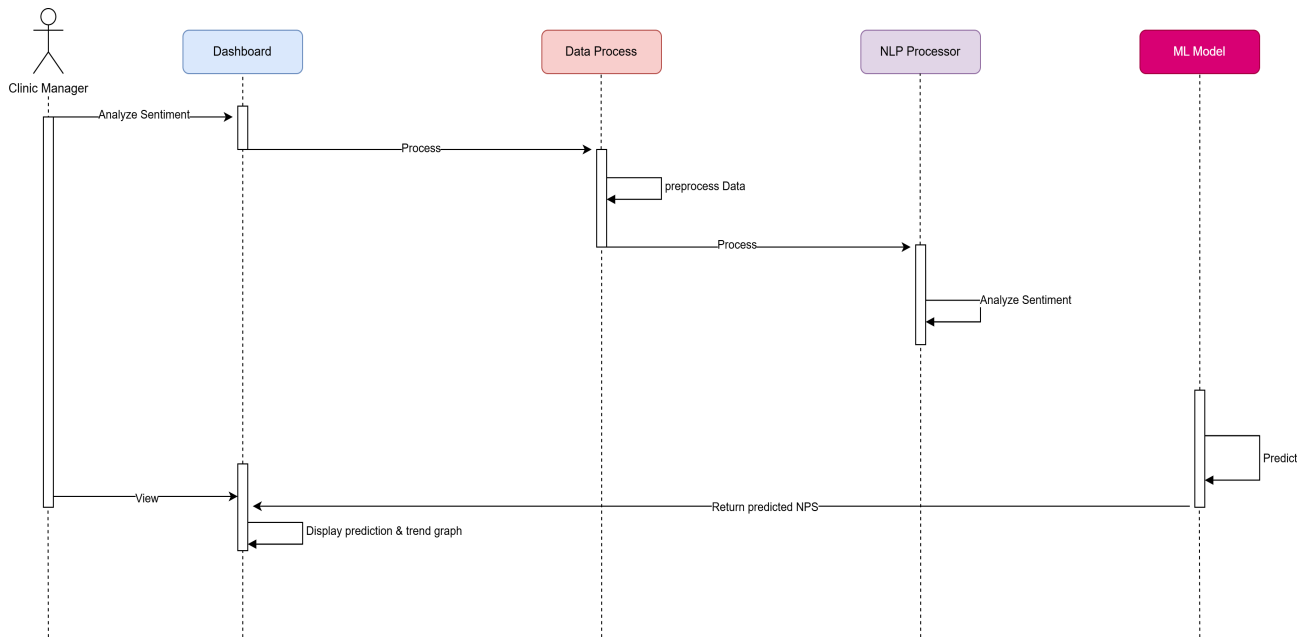
### Sequence Diagram:



**Figure 16:** The Sequence Diagram for Uploading CSV file.

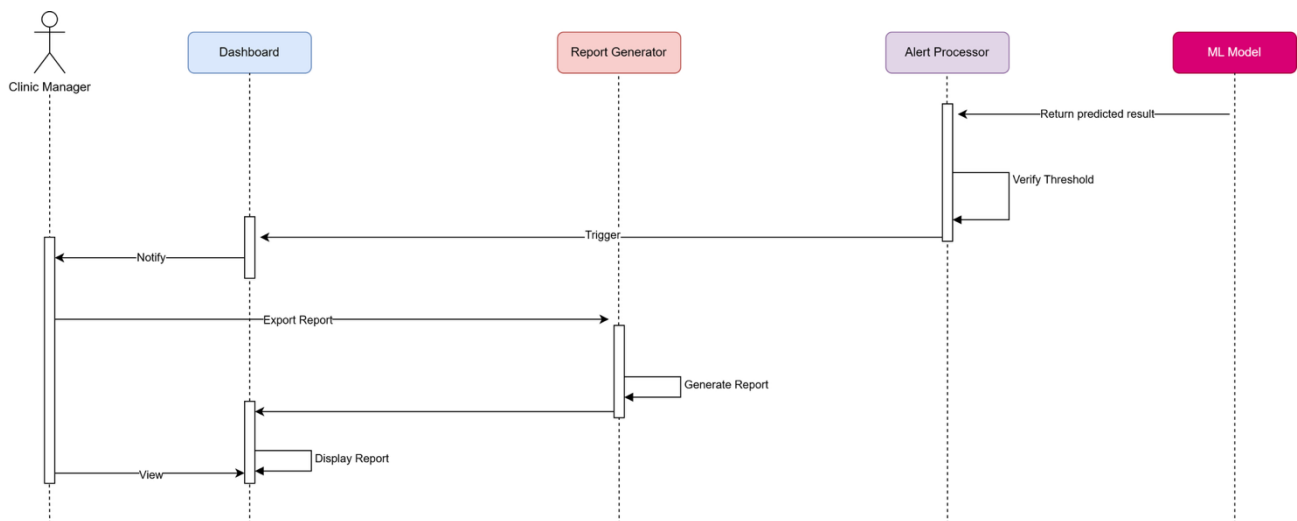
This diagram illustrates the sequence of interactions between the **Clinic Staff**, **Dashboard**, and **Data Processes** when uploading a feedback file. The process begins with the user selecting a .csv or .json file. The system then validates the structure, cleans the data, and returns a preview of the first few

rows along with a validation summary. This is the entry point of the application and is essential for initiating further analysis.



**Figure 17:** The Sequence Diagram for Predicting Sentiment Trends.

This diagram shows how the **Clinic Manager** initiates a prediction workflow by requesting sentiment analysis. The system first preprocesses the dataset and uses the **NLP Processor** to analyze sentiment. The processed data is then passed to the **ML Model**, which generates predicted satisfaction scores with a confidence interval. Results are returned to the user interface for visualization.



**Figure 18:** The Sequence Diagram for Generating Alert and Exporting Report.

This diagram captures the workflow after predictions are made. If predicted NPS scores fall below a threshold, the **Alert Processor** will be triggered to display visual warnings and highlight the most influential features. The **Clinic Manager** can then choose to export a report. The **Report Generator** compiles insights and creates a downloadable PDF or CSV, which is returned to the user for download.

## 7. Conclusion

*ClinicTrendsAI* represents a significant advancement in transforming customer feedback into actionable insights for aesthetic clinics and similar businesses. Its intuitive web-based interface, integrated machine learning capabilities, and robust data visualization tools empower managers and analysts to monitor satisfaction trends proactively and make data-driven decisions with confidence.

Future iterations of *ClinicTrendsAI* may include enhanced model accuracy through fine-tuned transformers, real-time data integrations, and advanced alerting systems to further support strategic planning and elevate customer experience management.