Subject Code and Title: SEP401 (N06748) Software Engineering Principles
Student Name: Luis Guilherme de Barros Andrade Faria
Student ID: A00187785

Assessment 1, Part B: Software Requirements Specifications

Member Name	**Student Id**
Luis Guilherme de Barros Andrade Faria	A00187785
Jing Feng Chin	A00178098
Hai Chau Luong	

Project Details

IEEE Software Requirements Specification (SRS)

Project Title: ClinicTrendsAI: Customer Satisfaction Trend Prediction System

Version: 1.0

Date: June 3, 2025

Author: Luis G. B. A. Faria

---

1. Introduction

1.1 Purpose

ClinicTrendsAI is a software system designed to analyze and forecast customer satisfaction trends using historical survey data (e.g., NPS-style feedback) collected by aesthetic clinics. It leverages basic machine learning techniques to provide predictive insights, identify high-risk satisfaction drops, and support data-driven decision-making by clinic managers.

1.2 Document Conventions

This document uses standard IEEE SRS formatting. All system features are documented in Section 4 using the "shall" notation. UML and data flow diagrams are provided in Appendix B.

1.3 Intended Audience and Reading Suggestions

- **Software Developers**: for implementation guidance
- **Stakeholders (clinic managers)**: for understanding functional deliverables
- **QA Engineers**: for building test cases
- **Instructors**: for evaluating planning and design rigor

1.4 Product Scope

ClinicTrendsAI will:

- Ingest survey datasets (CSV, JSON)
- Predict future satisfaction scores using regression models
- Display trend graphs and alert flags
- Identify key drivers behind satisfaction changes (feature importance)
- Provide exportable reports for clinic use

1.5 References

- IEEE SRS Format
- Scikit-learn Documentation
- Agile Manifesto
- PMBOK Guide (for risk tracking and change management)

---

2. Overall Description

2.1 Product Perspective

ClinicTrendsAI is a standalone web-based system, designed with a 2-tier architecture:

- **Frontend**: Streamlit UI
- **Backend**: Python logic + pandas + scikit-learn

2.2 Product Functions

- Upload historical survey data

- Visualize satisfaction trends by date, store, or segment
- Predict next-month satisfaction scores
- Generate risk alerts based on trend projections
- Export PDF or CSV summary reports

2.3 User Classes and Characteristics

- **Clinic Manager**: Non-technical user seeking insights
- **Data Analyst**: Interested in feature breakdowns
- **System Admin**: For uploading datasets, managing app state

2.4 Operating Environment

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

2.5 Design and Implementation Constraints

- Must handle CSVs up to 100MB
- Predictions limited to linear regression and decision tree models in MVP
- Model retraining must occur asynchronously

2.6 User Documentation

- Quick-start guide (PDF)
- Tooltip hints in UI

2.7 Assumptions and Dependencies

- Survey datasets are clean or preprocessed
- Backend runs with access to Python ML environment

---

3. External Interface Requirements

3.1 User Interfaces

- Streamlit dashboard with upload button, chart views, prediction box, and alerts panel.

3.2 Hardware Interfaces

- None

3.3 Software Interfaces

- Python packages: scikit-learn, pandas, matplotlib

3.4 Communications Interfaces

- Localhost or HTTP (future cloud deployment may require HTTPS)

4. System Features

4.1 Upload Survey Data

- The user shall be able to upload CSV or JSON survey datasets.

4.2 Visualize Historical Trends

- The system shall plot NPS scores over time.
- The system shall allow filtering by store or segment.

4.3 Predict Satisfaction Trends

- The system shall use ML models to predict future satisfaction scores.
- The prediction shall be displayed with confidence intervals.

4.4 Alert System

- The system shall flag trends where predicted score drops below 50.
- The system shall highlight risky features influencing the dip.

4.5 Export Report

- The user shall be able to download a PDF or CSV summary of predictions.

5. Non-functional Requirements

5.1 Performance

- The system shall return predictions within 10 seconds of data upload.

5.2 Usability

- The UI shall be intuitive for non-technical users.

5.3 Reliability

- The system shall handle incomplete rows gracefully.

5.4 Maintainability

- The codebase shall follow PEP8 standards and include documentation.

5.5 Portability

- The system shall run on Windows, macOS, and Linux with Python installed.

Appendix A: Glossary

- **NPS**: Net Promoter Score
- **ML**: Machine Learning
- **MVP**: Minimum Viable Product

Appendix B: Analysis Models

- Use Case Diagram (to be inserted)
- System Architecture Diagram (2-tier)
- Data Flow Diagram (CSV input → prediction → alert + export)