



---

# SENG 491 – SENIOR PROJECT I

---

**Stratify: AI-Powered Multi-Agent Business Planning Platform**

**Software Requirement Specification Report**

**Team Members:**

- Shahdad Peykani – 99111760026 – Software Engineering
- Muhammad Ozar Mirza – 99687965054 – Software Engineering
- Abdirahman Tahlil Farah Hussein – 99717977422 – Software Engineering

**Supervisor:** Prof. Ayşenur Birtürk

**Jury Members:**

- Prof. Eren Ulu
- Prof. Elif Kurtaran Özbudak

## Table of Contents

1. Introduction .....	4
1.1 Purpose .....	4
1.2 Product Scope .....	4
1.3 Description .....	4
2. Overall Description .....	4
2.1 Product Perspective .....	4
2.2 Product Functions .....	4
2.3 Operating Environment .....	5
2.4 Constraints .....	5
2.5 Professional and Ethical Issues .....	5
3. External Interface Requirements .....	5
3.1 User Interface .....	5
3.2 Hardware Interfaces .....	6
3.3 Software Interfaces .....	6
4. System Features .....	6
4.1 Dual-Mode Input & Orchestration .....	6
4.2 RAG-Enhanced Market and Risk Analysis .....	7
4.3 Financial Feasibility Modeling .....	7
4.4 "Business-in-a-Box" Asset Generation .....	8
5. Other Nonfunctional Requirements .....	8
5.1 Performance Requirements .....	8
5.2 Safety Requirements .....	8
5.3 Security Requirements .....	8
5.4 Usability Requirements .....	8
5.5 Accuracy Requirements .....	9
6. Appendices .....	9

6.1 Appendix A: Glossary .....	9
6.2 Appendix B: Analysis Models .....	9
6.3 Appendix C: To Be Determined (TBD) .....	10
7. References .....	10

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to define the Software Requirements Specification (SRS) for Stratify, an AI-powered multi-agent business planning platform. This document details the functional and non-functional requirements, system interfaces, and constraints to guide the development team, project supervisor, and jury members.

## 1.2 Product Scope

Stratify addresses the high failure rate of early-stage startups caused by subjective validation and knowledge gaps. It is a unified "Business-in-a-Box" pipeline that transforms a user profile or a raw idea into a validated business plan. By leveraging Retrieval-Augmented Generation (RAG) and a multi-agent architecture, Stratify automates market research, financial forecasting, and initial web presence creation.

## 1.3 Description

Entrepreneurs often lack the tools to objectively validate ideas or bridge the gap between planning and execution. Stratify solves this by offering two modes: Ideation (generating ideas based on user skills/budget) and Validation (analyzing an existing concept). The system employs specialized AI agents (Market, Risk, Financial, and Web Generation) to produce a comprehensive feasibility report and a deployable starter website.

# 2. Overall Description

## 2.1 Product Perspective

Stratify is a web-based application that serves as an intelligent "co-founder". It operates as an orchestration layer between the user and various external services, including Large Language Models (LLMs) for reasoning and Vector Databases for retrieving real-time market data (RAG).

## 2.2 Product Functions

The core functions of Stratify are:

- **Dual-mode Input Processing:** Accepting user profiles for ideation or detailed descriptions for validation
- **RAG-Driven Market Research:** Retrieving relevant industry data and competitor landscapes from a knowledge base.
- **Multi-Agent Analysis:** Performing SWOT analysis, risk assessment, and financial modeling (CAPEX/OPEX estimation).

- **Automated Asset Generation:** Creating a PDF business feasibility report and a single-page landing website.

## 2.3 Operating Environment

- **Frontend:** A modern web framework (e.g., React or Vue.js) for the user dashboard and wizard interface.
- **Backend:** Python-based server (e.g., FastAPI or Flask) to handle agent orchestration.
- **AI/Data Infrastructure:** Integration with an LLM API (e.g., OpenAI or Gemini) and a Vector Database (e.g., Pinecone or ChromaDB) for the RAG pipeline.

## 2.4 Constraints

- **LLM Dependencies:** The quality of the business advice is directly tied to the reasoning capabilities of the underlying LLM.
- **Data Freshness:** The RAG system is limited by the currency of the data ingested into the vector database; it may not reflect real-time news events occurring today.
- **Financial Estimation:** The financial agent provides estimates based on industry averages, not guaranteed financial advice.
- **Scope of Web Generation:** The website generation is constrained to a static, single-page landing page (HTML/CSS) for validation purposes, not a full-stack web application.

## 2.5 Professional and Ethical Issues

- **Financial Disclaimer:** Stratify provides educational/planning references. The system must clearly disclaim that it does not provide certified financial advice.
- **Hallucination Risk:** Users must be warned that AI agents can occasionally generate plausible but incorrect facts (hallucinations), requiring human review of the final report.
- **Data Privacy:** User business ideas are intellectual property; the system should not use user inputs to train public models without consent.

# 3. External Interface Requirements

## 3.1 User interface

The system shall provide a web application with the following components:

- **Onboarding Wizard:** A step-by-step form to select Mode A (Ideation) or Mode B (Validation) and input relevant data (skills, budget, idea description).

- **Progress Dashboard:** A visual indicator showing the status of different agents (e.g., "Market Agent analyzing...", "Generating Website...").
- **Report Viewer:** A built-in PDF viewer or download link for the Feasibility Report.
- **Web Previewer:** An iframe or preview pane to view the generated landing page code rendered in real-time.

## 3.2 Hardware Interfaces

- **Server-side:** Standard cloud computing instances (e.g., AWS EC2 or Google Cloud).
- **Client-side:** Accessible via standard web browsers (Chrome, Firefox, Edge).

## 3.3 Software Interfaces

- **LLM API:** For all generative tasks (Ideation, Analysis, Coding).
- **Vector Database API:** For retrieving context documents during the RAG process.
- **PDF Generation Library:** (e.g., ReportLab or FPDF) To compile text outputs into a formal document.

# 4. System Features

## 4.1 Dual-Mode Input & Orchestration

### 4.1.1 Description and Priority

High Priority. The entry point of the application. It must distinguish between users who need ideas and users who have ideas.

### 4.1.2 Stimulus/Response Sequences

- **Stimulus:** User selects "Mode A: Ideation" and inputs skills (e.g., "Python, Marketing") and budget (e.g., "\$1000").
- **Response:** The Idea Agent generates 3 viable startup concepts. The user selects one to proceed.
- **Stimulus:** User selects "Mode B: Validation" and inputs a business description.
- **Response:** The system skips generation and passes the input directly to the analysis pipeline.

### 4.1.3 Functional requirements

- **FR-1:** The system shall accept structured user profiles (skills, risk tolerance, budget) for Mode A.
- **FR-2:** The system shall accept unstructured text descriptions for Mode B.

- **FR-3:** The Idea Agent shall generate at least three distinct business concepts based on user constraints in Mode A.

## 4.2 RAG-Enhanced Market and Risk Analysis

### 4.2.1 Description and Priority

High Priority. This feature ensures the advice is grounded in data rather than just LLM training data.

### 4.2.2 Stimulus/Response Sequences

- **Stimulus:** A business concept is selected/provided.
- **Response:** The system queries the Vector Database for similar business models and market trends.
- **Stimulus:** Context is retrieved.
- **Response:** The Market & Risk Agents generate a SWOT analysis (Strengths, Weaknesses, Opportunities, Threats) and identify saturation risks.

### 4.2.3 Functional Requirements

- **FR-4:** The system shall query a vector database to retrieve relevant industry documents based on the business concept.
- **FR-5:** The system shall generate a SWOT analysis using the retrieved context.
- **FR-6:** The system shall identify at least three potential risks (operational, regulatory, or market).

## 4.3 Financial Feasibility Modeling

### 4.3.1 Description and Priority

Medium Priority. Provides a "sanity check" on the economics of the idea.

### 4.3.2 Stimulus/Response Sequences

- **Stimulus:** Market analysis is complete.
- **Response:** The Financial Agent estimates startup costs and potential revenue streams based on the business type.

### 4.3.3 Functional Requirements

- **FR-7:** The system shall estimate initial Capital Expenditure (CAPEX) based on the user's stated budget and business type.
- **FR-8:** The system shall calculate a theoretical Break-Even point based on estimated monthly expenses and projected unit economics.

## 4.4 "Business-in-a-Box" Asset Generation

### 4.4.1 Description and Priority

High Priority. The final tangible output for the user.

### 4.4.2 Stimulus/Response Sequences

- **Stimulus:** All analysis agents have finished.
- **Response:** The system compiles a PDF report containing Executive Summary, Strategy, Financials, and Risk.
- **Response:** The Web Generation Agent produces a single HTML file with a Hero section, Value Proposition, and "Coming Soon" email capture form.

### 4.4.3 Functional Requirements

- **FR-9:** The system shall generate a downloadable PDF Feasibility Report.
- **FR-10:** The system shall generate valid, responsive HTML/CSS code for a landing page tailored to the business concept.

## 5. Other Nonfunctional Requirements

### 5.1 Performance Requirements

- The *Idea* Agent (Mode A) should generate concepts within 30 seconds.
- The full "Business-in-a-Box" generation (Analysis + Report + Website) should complete within 3 minutes to maintain user engagement.

### 5.2 Safety Requirements

- The system must implement content filtering to prevent the generation of illegal or harmful business ideas (e.g., scams, prohibited goods).

### 5.3 Security Requirements

- User data (business ideas) must be encrypted in transit.
- API Keys for LLM services must be stored in environment variables, never in the frontend code.

### 5.4 Usability Requirements

- The interface shall use non-technical language where possible (e.g., explaining "CAPEX" simply as "Startup Costs").
- The system shall allow users to edit the generated website text within the browser before downloading.



## 5.5 Accuracy Requirements

- Generated website code must be syntactically correct and render without errors in standard browsers.
- Financial calculations must be mathematically accurate based on the estimated inputs provided by the agents.

## 6. Appendices

### 6.1 Appendix A: Glossary

- **RAG (Retrieval-Augmented Generation):** A technique to optimize LLM output by referencing an authoritative knowledge base outside its training data.
- **SWOT Analysis:** A strategic planning technique used to help a person or organization identify Strengths, Weaknesses, Opportunities, and Threats related to business competition.
- **CAPEX:** Capital Expenditure; funds used by a company to acquire, upgrade, and maintain physical assets.

### 6.2 Appendix B: Analysis Models

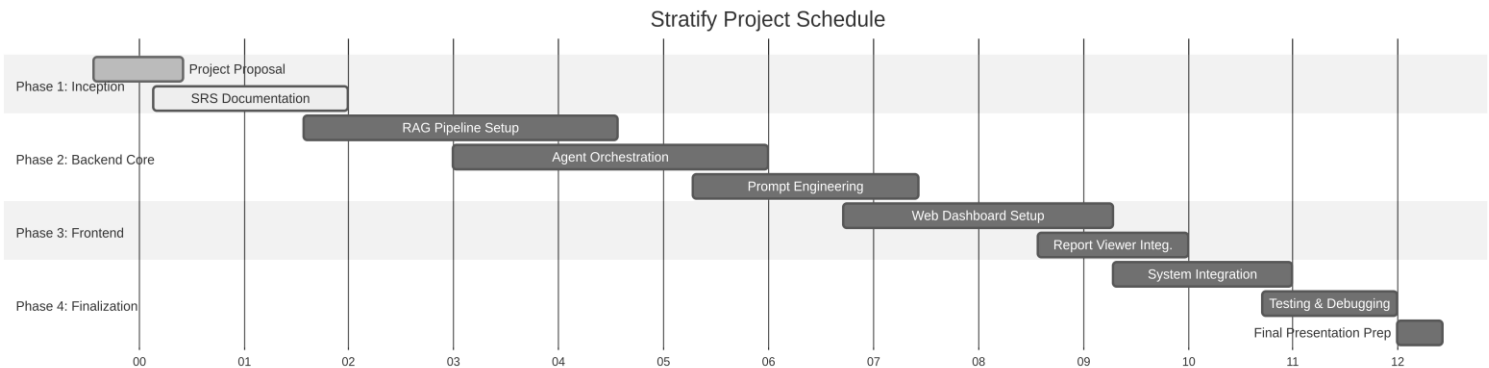


Figure 1: Proposed Gantt chart

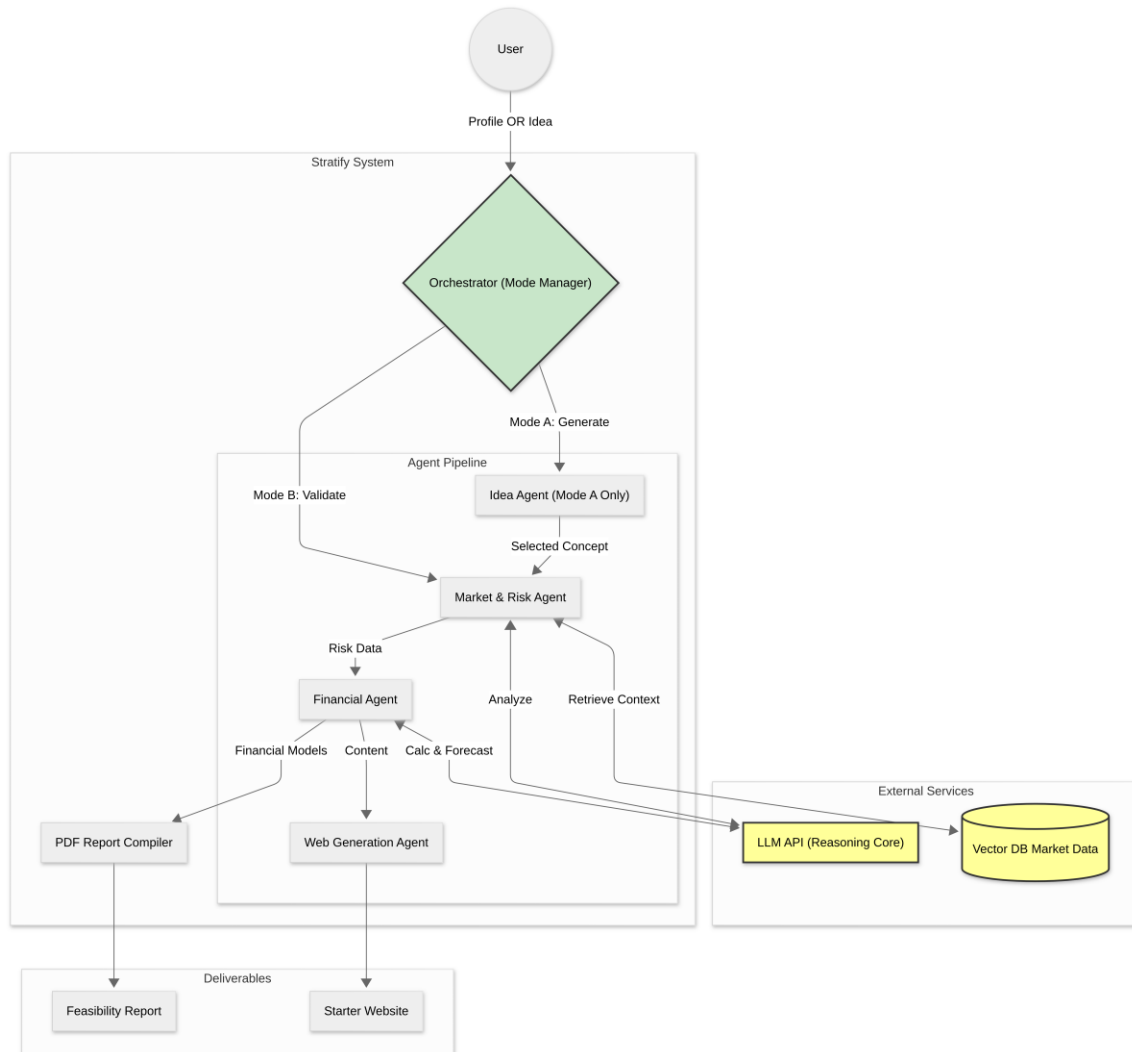


Figure 2: High-Level System Architecture

### 6.3 Appendix C: To Be Determined (TBD)

- Specific list of data sources for the Vector Database (e.g., Crunchbase public data, government small business guides).
- Selection of specific UI framework (React vs. Streamlit).

## 7. References

[1] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Proc. NeurIPS, 2020. [Online]. Available: <https://arxiv.org/abs/2005.11401>

- [2] IEEE, "IEEE Code of Ethics," IEEE Policies, Section 7.8, 2024. [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8.html>
- [3] A. Osterwalder and Y. Pigneur, Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. Hoboken, NJ: John Wiley & Sons, 2010.
- [4] K. Wiegers and J. Beatty, Software Requirements, 3rd ed. Redmond, WA: Microsoft Press, 2013.
- [5] T. I. Tiangolo, "FastAPI High performance, easy to learn, fast to code, ready for production," FastAPI Documentation, 2024. [Online]. Available: <https://fastapi.tiangolo.com/>
- [6] Meta Open Source, "React: The Library for Web and Native User Interfaces," 2024. [Online]. Available: <https://react.dev/>
- [7] "OWASP Top Ten," OWASP Foundation, 2024. [Online]. Available: <https://owasp.org/www-project-top-ten/>