SENG 491 – SENIOR PROJECT I

Stratify: AI-Powered Multi-Agent Business Planning Platform

Project Analysis 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

1. Executive Summary

Stratify is an AI-powered framework designed to serve as an intelligent business co-founder for early-stage entrepreneurs. The system addresses the high failure rate of early-stage startups caused by subjective validation and knowledge gaps. By leveraging a Retrieval-Augmented Generation (RAG) pipeline and a multi-agent architecture, Stratify automates market research, financial forecasting, and the creation of an initial web presence. It acts as a unified "Business-in-a-Box" pipeline that transforms either a user profile or a raw idea into a validated, professional business plan.

2. Problem Definition

Entrepreneurs and students face critical challenges during the venture planning phase, including:

- **Subjective Validation:** Founders with existing ideas often lack objective data to verify market need, leading to bias-driven decisions.
- **Knowledge Gaps:** Individuals frequently lack the structured knowledge necessary to conduct formal risk assessments, financial modeling, or compliance checks.
- **Fragmented Tooling:** There is no unified system integrating market research, financial forecasting, and technical execution. Users are forced to stitch together disparate, non-adaptive tools.
- **Execution Barriers:** Non-technical founders struggle to bridge the gap between planning and execution, especially when establishing an initial online presence.

3. Proposed System

3.1 Overview

The Stratify framework acts as an automated venture architect, enforcing data-driven rigor from initial input to final execution. The system operates through a structured pipeline managed by a central orchestrator:

1. I**nput Processing:** The system handles two distinct user scenarios: Mode A (Ideation based on user profiles) and Mode B (Validation of an existing idea).
2. **Context Retrieval:** A RAG pipeline fetches real-time industry data and competitor landscapes from a vector database.
3. **Multi-Agent Analysis:** Specialized AI agents evaluate the concept: Market/Risk agents conduct SWOT and risk analyses, while the Financial agent estimates capital and break-even points.
4. **Automated Delivery:** The system compiles a formal business feasibility report (PDF) and a deployable starter website.

3.2 Functional Requirements

- **FR-01: Dual-Mode Input Processing**
  The system shall accept structured user profiles (skills, risk tolerance, budget) for Mode A. The system shall accept unstructured text descriptions for Mode B.

- **FR-02: Concept Generation (Ideation)**
  The *Idea* Agent shall generate at least three distinct business concepts based on user constraints when operating in Mode A.

- **FR-03: Contextual Data Retrieval**
  The system shall query a vector database to retrieve relevant industry documents based on the chosen business concept.

- **FR-04: Market and Risk Analysis**
  The system shall generate a SWOT analysis using the retrieved context. Additionally, the system shall identify at least three potential risks (operational, regulatory, or market).

- **FR-05: Financial Modeling**
  The system shall estimate initial Capital Expenditure (CAPEX) based on the user's budget and business type. The system shall also calculate a theoretical Break-Even point based on estimated expenses.

- **FR-06: Asset Generation**
  The system shall generate a downloadable PDF Feasibility Report. Furthermore, the system shall generate valid, responsive HTML/CSS code for a tailored landing page.

3.3 Nonfunctional Requirements

- **NFR-01: Performance (Latency)**
  The Idea Agent (Mode A) should generate concepts within 30 seconds. The full "Business-in-a-Box" generation must complete within 3 minutes to maintain user engagement.

- **NFR-02: Safety and Filtering**
  The system must implement content filtering to prevent the generation of illegal or harmful business ideas.

- **NFR-03: Security and Privacy**
  User data (business ideas) must be encrypted in transit. API Keys for LLM services must be stored in environment variables, never in frontend code. User inputs must not be used to train public models without consent.
- **NFR-04: Accuracy**
  Generated website code must be syntactically correct and render without errors in standard browsers. Financial calculations must be mathematically accurate based on agent inputs.

3.4 Pseudo Requirements

- **PR-01: Technology Stack**
    - Frontend: A modern web framework (e.g., React or Vue.js).
    - Backend: Python-based server (e.g., FastAPI or Flask).
    - AI Infrastructure: Integration with an LLM API (e.g., OpenAI or Gemini) and a Vector Database (e.g., Pinecone or ChromaDB).

- **PR-02: Output Limitations**
  Website generation is constrained to a static, single-page landing page (HTML/CSS) for validation, not a full-stack application.

- **PR-03: Ethical Disclaimers**
  The system must clearly disclaim that it provides educational/planning references and not certified financial advice. Users must be warned about potential AI hallucinations requiring human review.

3.5 System Models

3.5.1 Scenarios

Scenario A: Ideation (Mode A)

1. **User:** Selects "Mode A: Ideation" and inputs their skills ("Python, Marketing") and budget ("$1000").
2. **Stratify:** The Idea Agent generates 3 viable startup concepts based on these constraints.
3. **User:** Selects one concept to proceed.
4. **Stratify:** System queries the Vector Database for market trends. Market & Risk Agents conduct a SWOT analysis, and the Financial Agent estimates CAPEX.
5. **Result:** User downloads the generated PDF Feasibility Report and the starter website HTML file.

Scenario B: Validation (Mode B)

1. **User**: Selects "Mode B: Validation" and enters a detailed description of an existing business idea.
2. **Stratify:** The system skips the Idea Agent generation phase and passes the input directly to the analysis pipeline.
3. **Stratify:** The RAG pipeline retrieves relevant industry documents based on the exact concept.
4. **Stratify:** Agents compile risks and financial models specific to the inputted idea.
5. **Result:** User receives an objective, data-grounded report verifying the market need and operational risks of their idea.

3.5.2 Use Case Model

The system interacts primarily with the User (Entrepreneur/Student) and external APIs (LLM, Vector DB).

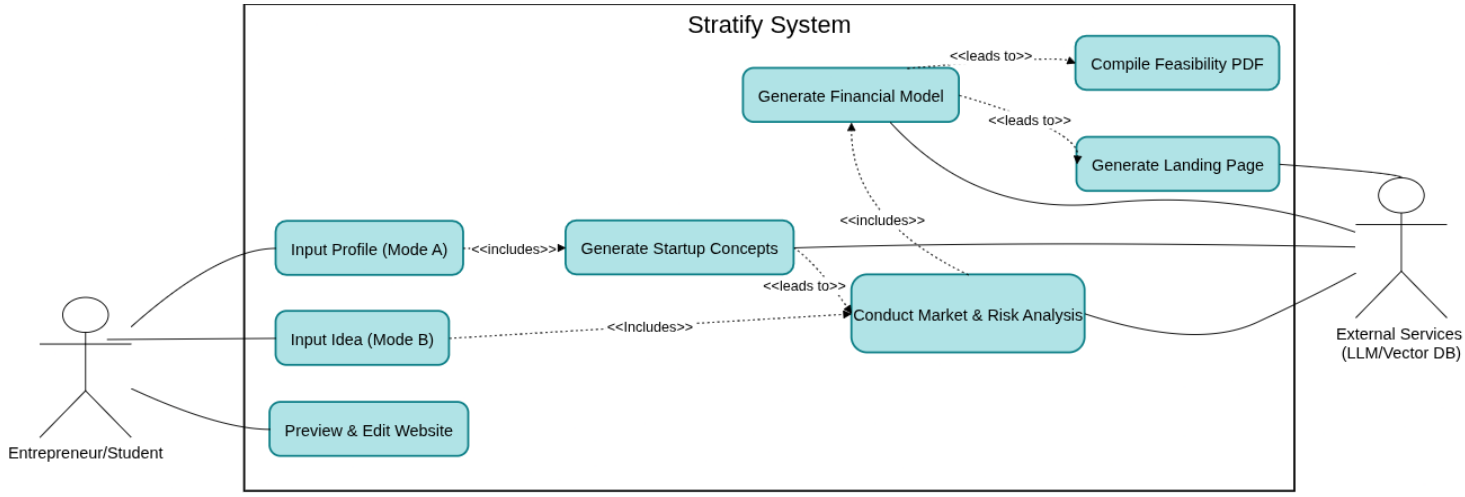

Figure 1: Use Case Diagram

3.5.3 Object and Class Model

This model represents the core structural components of the Stratify backend, highlighting the Orchestrator, the abstract Agent base class, and the specific agent implementations.

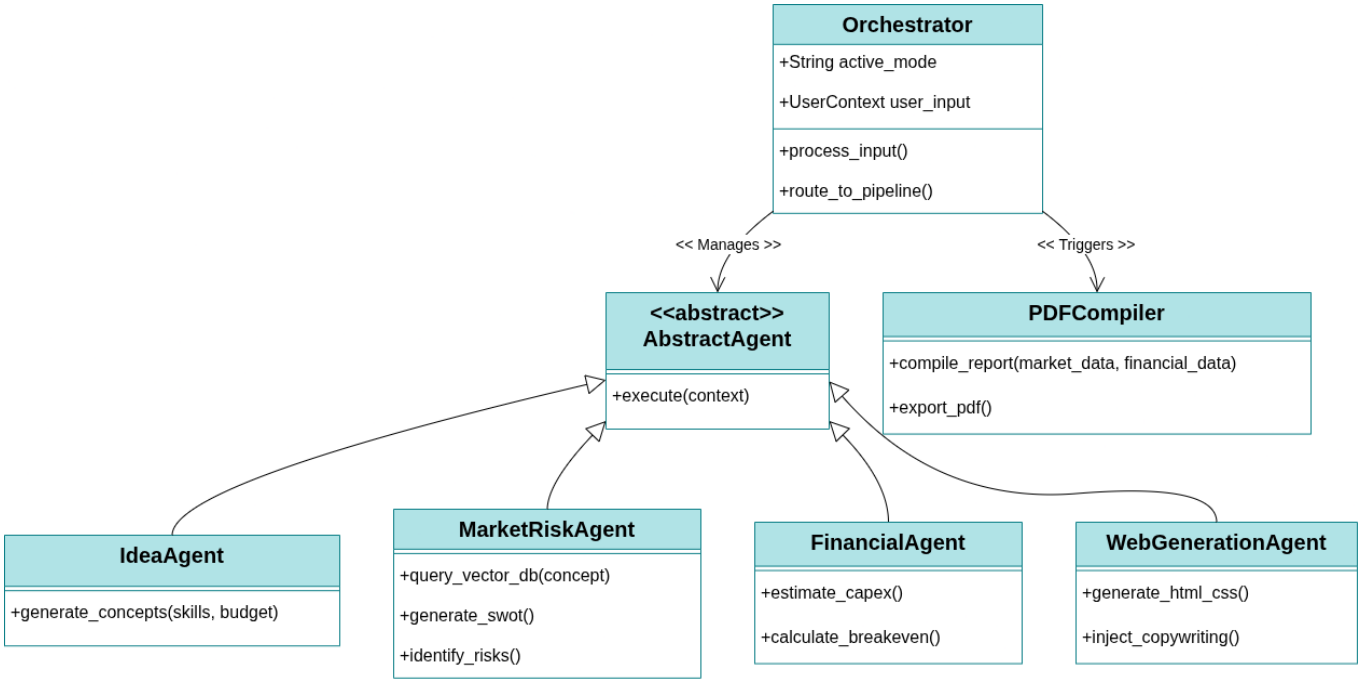

Figure 2: Class Diagram

3.5.4 Dynamic Models

The sequence diagram below illustrates the flow of the application depending on whether the user selects Mode A or Mode B, and how the Orchestrator passes data through the multi-agent pipeline.

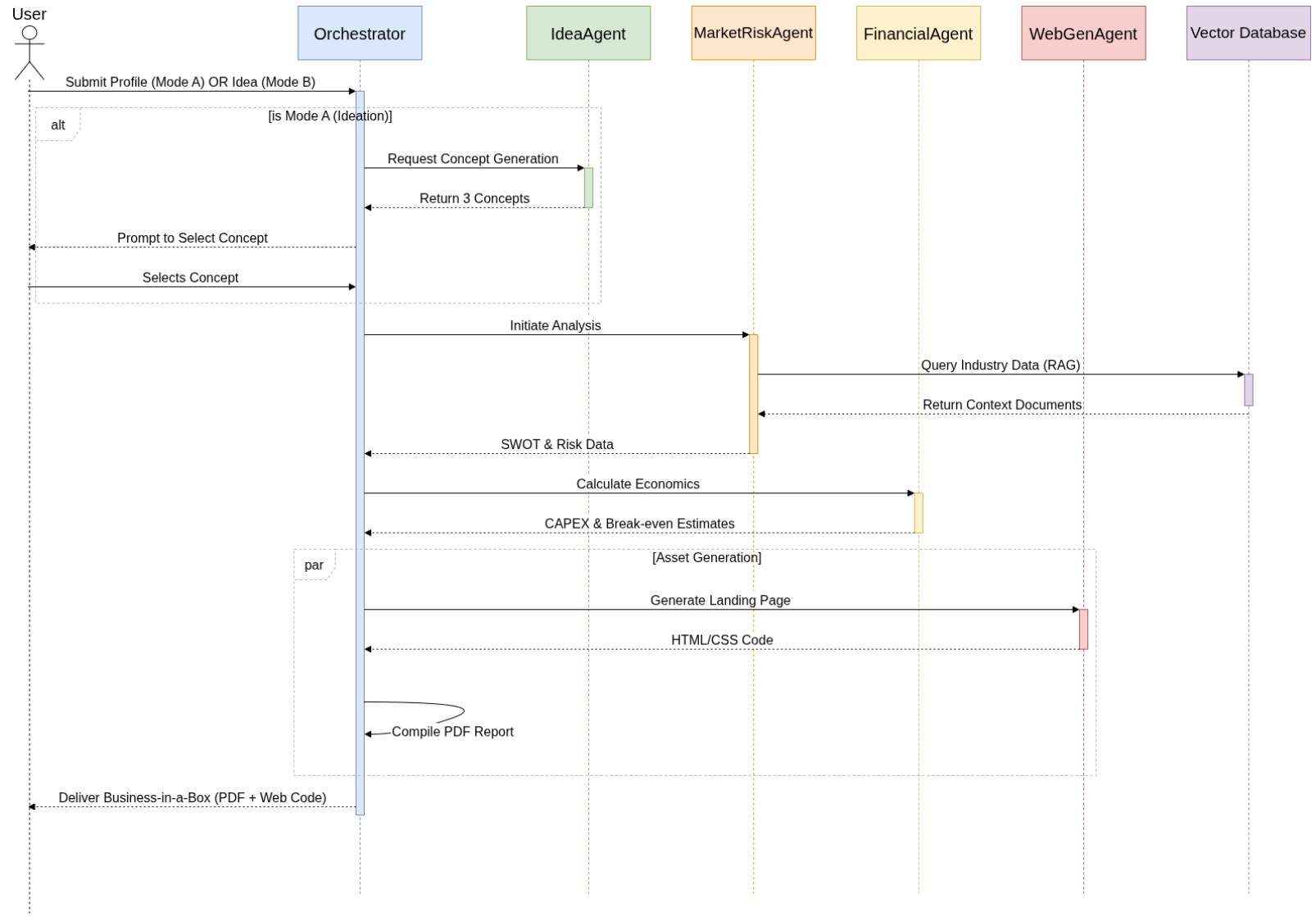

Figure 3: Sequence Diagram

3.5.5 User Interface/Navigational Paths and Screen Mock-ups

The user interface consists of several core components to guide the user seamlessly from input to execution:

- **Onboarding Wizard:** A step-by-step form to select Mode A or Mode B and input relevant data.
- **Progress Dashboard:** A visual indicator showing the status of different background agents (e.g., "Market Agent analyzing...").
- **Report Viewer:** A built-in PDF viewer or direct download link for the completed feasibility report.
- **Web Previewer:** An iframe or preview pane to view the generated landing page code rendered in real-time, allowing users to edit text before downloading.

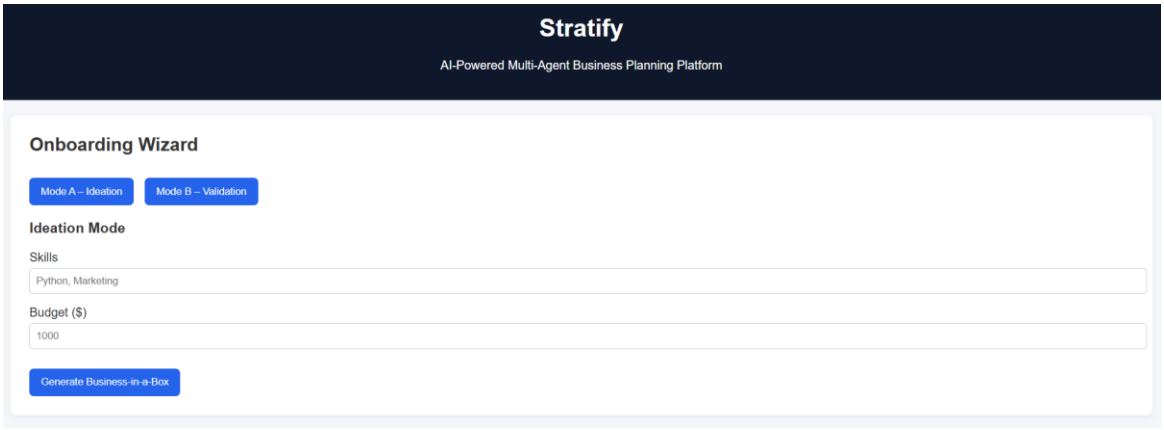

Figure 4: Project Context Input Interface for Mode A Initialization

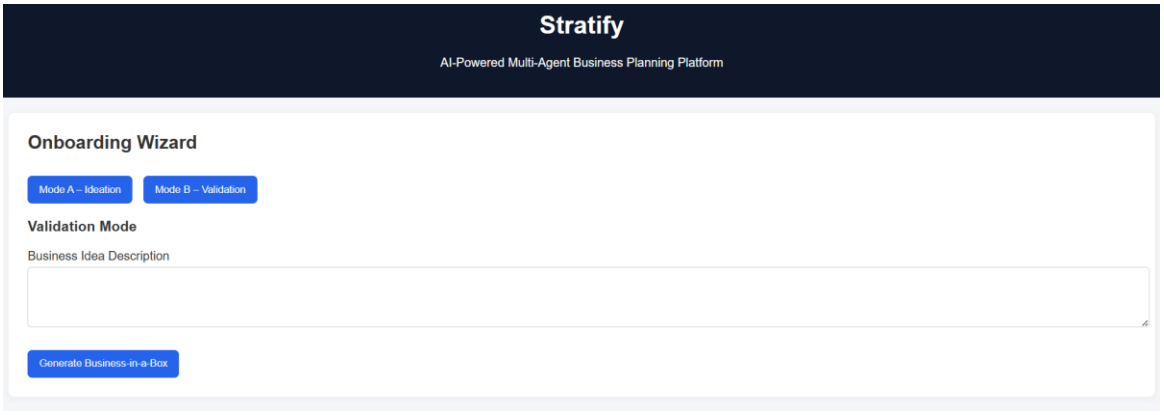

Figure 5: Project Context Input Interface for Mode B Initialization

Figure 6: Idea Generation and Multi-Agent Pipeline Execution Interface

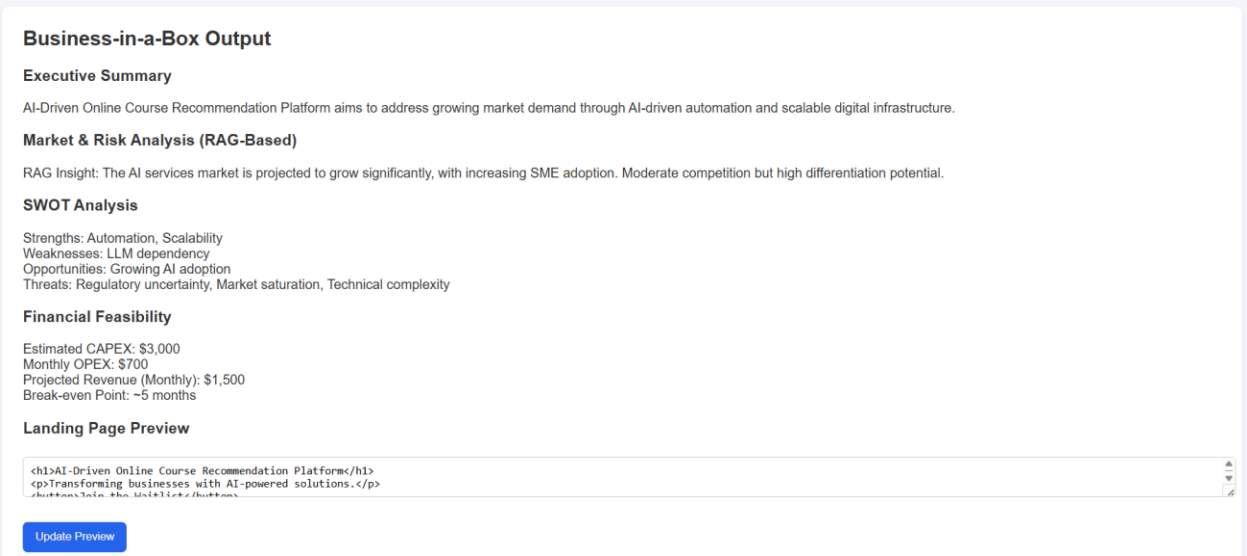

Figure 7: Business-in-a-Box Output Interface

4. 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 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 (Startup Costs).

5. References

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

[2] IEEE, "IEEE Code of Ethics," IEEE Policies, Section 7.8, 2024. 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. Available: https://fastapi.tiangolo.com/

[6] Meta Open Source, "React: The Library for Web and Native User Interfaces," 2024. Available: https://react.dev/