Project Title: LLM-Based Investment Advisor

Team Name: Investment Advisor

Team Members

Lyria Keriane Nzabampema[1]	Mouna Kandekwe[1]
Ibrahim Jimoh[1]	Ayebilla Avoka[1]

Faculty Advisor:
Charles Wiecha[1]
cwiecha@andrew.cmu.edu

1. Carnegie Mellon University Africa

Date of submission: 14/10/2024

2. Problem statement

a) Solution needs:

We will deliver a way for investors to access financial education, receive personalized investment advice, and monitor the potential growth of their investments based on their preferences.

b) Requirements:

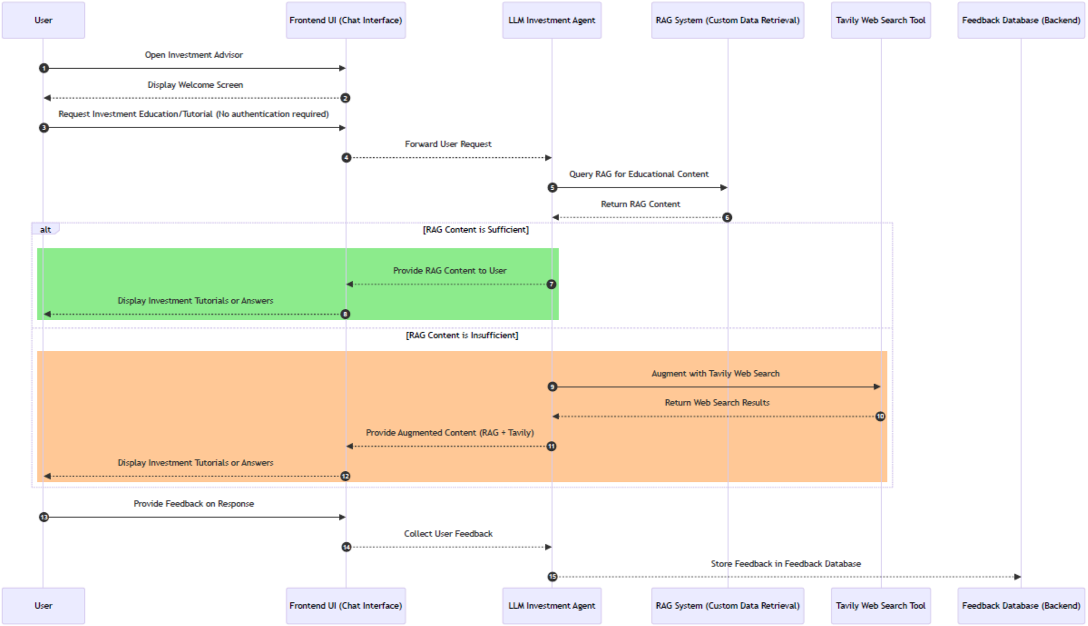
	User (requirements from non-implementers)	System (requirements from implementers)
Functional (What it does)	**1.**The novice investor shall be able to ask questions about financial concepts and investment options (e.g., stocks, bonds) through a chat interface so that they can educate themselves on financial markets. **2.**The investor shall be able to provide their investment preferences (e.g., risk tolerance, budget, time horizon) so that they	**1.**The system shall integrate with GPT-3.5 and Tavily APIs to provide real-time educational content and financial concepts to users through the chat interface. **2.**The investor shall be able to provide their investment preferences (e.g., risk tolerance, budget, time horizon) so that they can receive personalized investment

	can receive personalized investment recommendations. **3.**The user shall be able to manage their account by creating, updating, and deleting it, as well as providing profile information, so that they can maintain control over their personal data. **4.**The investor shall be able to receive predictions of future gains based on their investments so that they can make informed financial decisions. **5.**The user shall be able to log into the system and retrieve their profile information and previous interactions so that they can continue from where they left off. **6.**The user shall be able to provide feedback on investment recommendations and predictions so that they can help improve the system.	recommendations. **3.**The system shall integrate with the external allocation tool to compute and provide users with a breakdown of their resource allocation, which will be displayed on a Dash-based dashboard. **4.**The system shall integrate with the external forecasting tool to generate future gain predictions based on the user's investment allocations and display the results on the dashboard. **5.**The system shall provide authentication using a secure login system and shall store and retrieve user profiles and conversation history in a persistent JSON file-based database. **6.**The system shall collect user feedback on recommendations and predictions and store this data in the feedback database for further analysis and system improvement.
Non-functional (How it does it)	**1.**The user should be able to access a user-friendly interface so that they can navigate the system easily regardless of their technical expertise. **2**.The user should be able to receive real-time feedback when their profile information is being updated so that they can ensure their data is accurate and up-to-date.	**1.**The system should provide an intuitive UI to be user-friendly to non-technical users **2.**The system should respond to user queries within 3 seconds 95% of the time to ensure timely access to information and simulations. **3.**The system should answer at least three financial questions correctly during testing to validate the accuracy of its educational content.

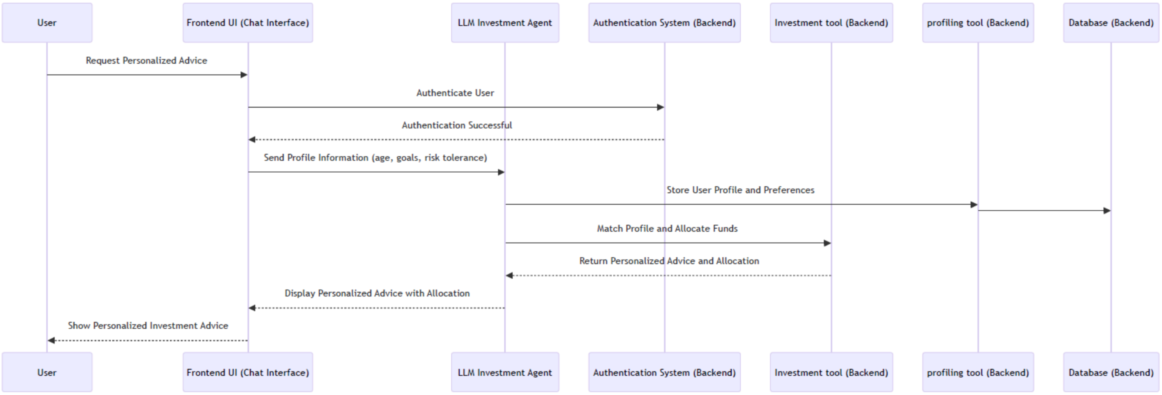
c) Interactions between Users and the system

Sequence Diagrams:

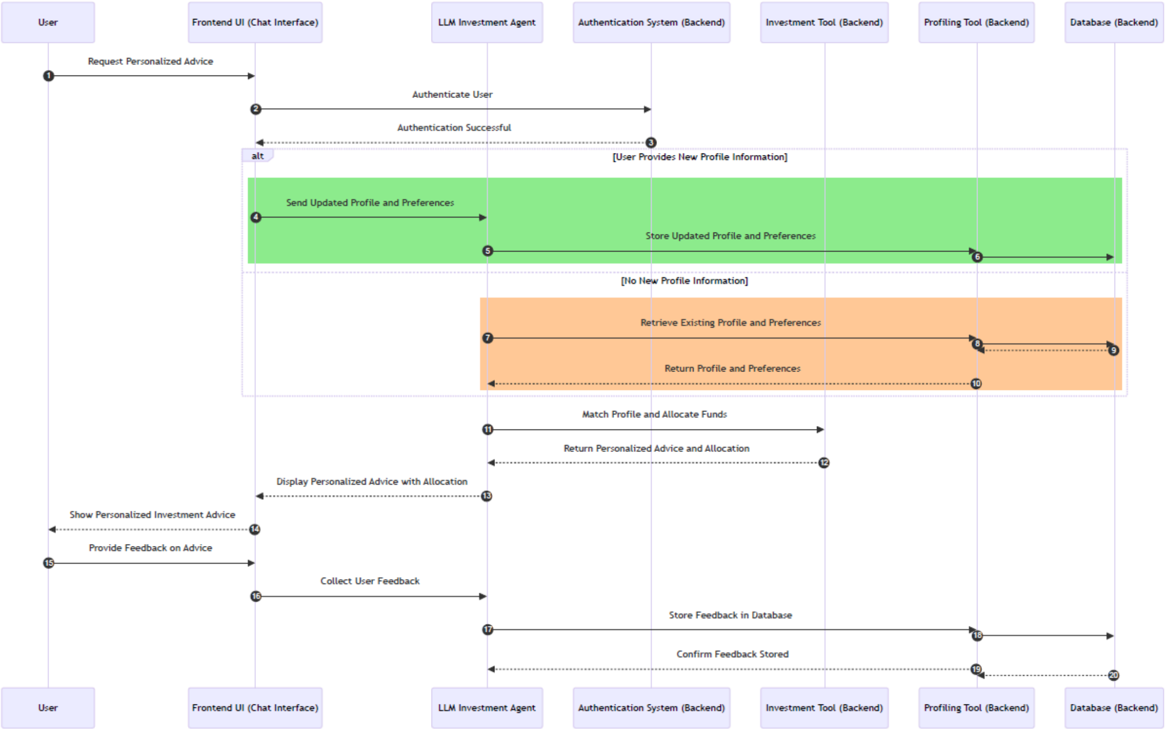
1. Educational query interaction


User
Frontend UI (Chat Interface)
LLM Investment Agent
RAG System (Custom Data Retrieval)
Tavily Web Search Tool
Feedback Database (Backend)
Open Investment Advisor
Display Welcome Screen
Request Investment Education/Tutorial (No authentication required)
Forward User Request
Query RAG for Educational Content
Return RAG Content
alt
[RAG Content is Sufficient]
Provide RAG Content to User
Display Investment Tutorials or Answers
[RAG Content is Insufficient]
Augment with Tavily Web Search
Return Web Search Results
Provide Augmented Content (RAG + Tavily)
Display Investment Tutorials or Answers
Provide Feedback on Response
Collect User Feedback
Store Feedback in Feedback Database

2. Personalized investment advice interaction


User
Frontend UI (Chat Interface)
LLM Investment Agent
Authentication System (Backend)
Investment tool (Backend)
profiling tool (Backend)
Database (Backend)
Request Personalized Advice
Authenticate User
Authentication Successful
Send Profile Information (age, goals, risk tolerance)
Store User Profile and Preferences
Match Profile and Allocate Funds
Return Personalized Advice and Allocation
Display Personalized Advice with Allocation
Show Personalized Investment Advice

3. User profile and resource allocation interaction


User
Frontend UI (Chat Interface)
LLM Investment Agent
Authentication System (Backend)
Investment Tool (Backend)
Profiling Tool (Backend)
Database (Backend)
Request Personalized Advice
Authenticate User
Authentication Successful
alt
[User Provides New Profile Information]
Send Updated Profile and Preferences
Store Updated Profile and Preferences
[No New Profile Information]
Retrieve Existing Profile and Preferences
Return Profile and Preferences
Match Profile and Allocate Funds
Return Personalized Advice and Allocation
Display Personalized Advice with Allocation
Show Personalized Investment Advice
Provide Feedback on Advice
Collect User Feedback
Store Feedback in Database
Confirm Feedback Stored
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

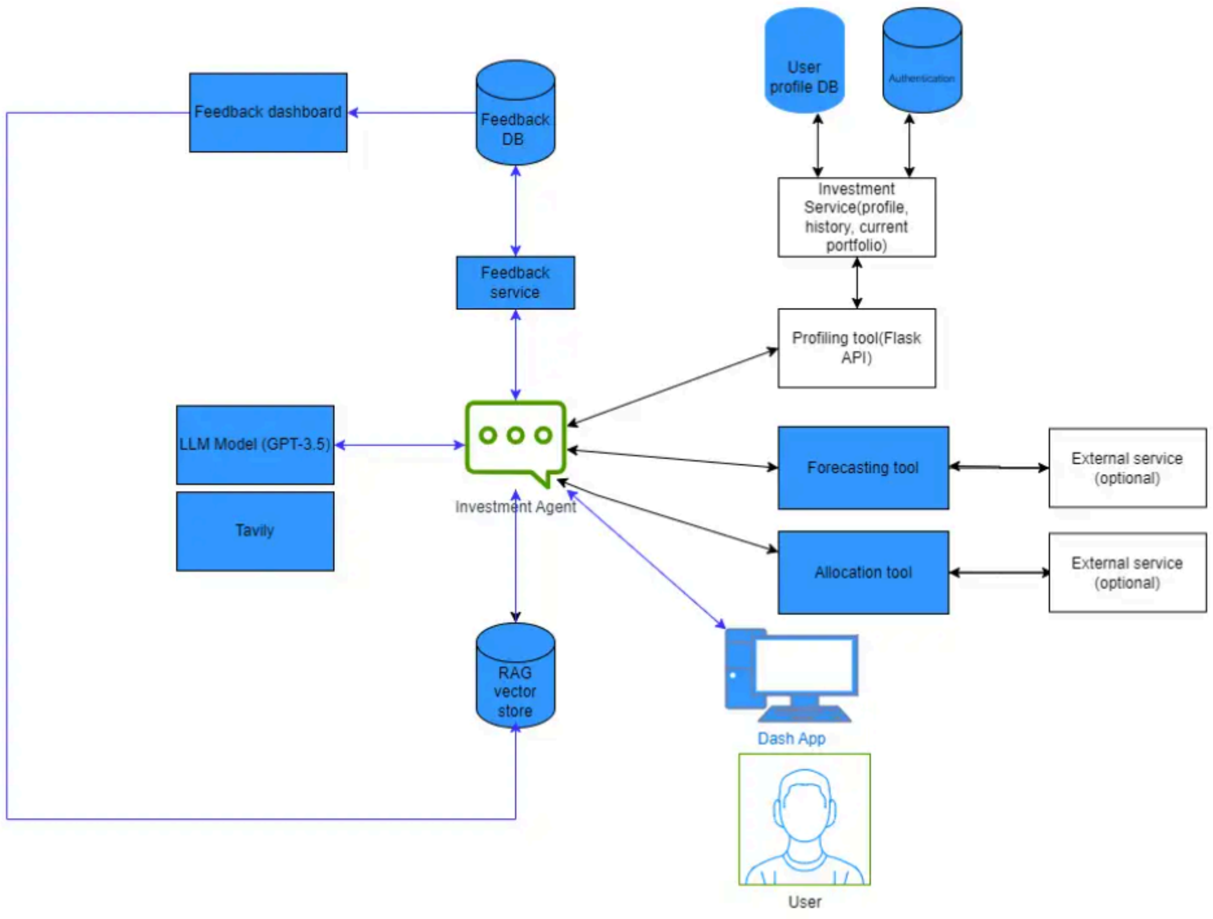
d) Services to Be Provided:

The system will provide the following services to meet the needs of the users:

1. **Educational Service**: Through the integration of GPT-3.5, Tavily and a RAG system, the system will offer financial education by answering user queries and providing explanations of key financial concepts.
2. **Investment Profiling**: The chatbot will collect user investment preferences such as risk tolerance, budget, and time horizon. This data will be used to generate personalized investment recommendations.
3. **Resource Allocation**: The system will integrate with an external allocation tool to process user preferences and distribute their resources across stocks, bonds, and mutual funds.
4. **Prediction and Forecasting**: The system will integrate with an external forecasting tool to predict the potential future gains based on the user's investment allocations.
5. **Visualization**: The user will be able to visualize their resource allocations on a dashboard, with real-time updates reflecting their investment preferences.

6. **Feedback Collection**: The system will provide a feedback mechanism for users to rate and comment on the accuracy of the recommendations and predictions.

System architecture:



3. Project Plan:

a) Major activities and time allocation

Schedule	Activities	Tasks/Efforts	Milestones	Deliverables	Status
Sprint 1 (Week 1 -2)	Educational content	- Integrate GPT-3.5 for financial question answering. **(48 hours)**	**Milestone 1: Demo on 1st of October 2024**	Sprint 1 review demo first functional part	Done
Sprint 2 (Week 3 - 5)	Educational content and authentication	- Implement user authentication (login/registration). **(36 hours)** - Set up connection to RAG System and Tavily Web Search API. **(4 hours)** - Develop and test chatbot interface (Chainlit + Dash). **(12 hours)** - Build persistent user data storage (JSON format). **(8 hours)**	**Milestone 2**: mid-term presentation on **29th October 2024**	We will demo a functional chatbot that can answer financial questions using GPT-3.5 and integrated external sources (Tavily). The demo will also showcase user authentication (login/registration) and persistent storage of user profiles and conversation history.	In progress
	Investment preferences and external allocation	- Extend chatbot to collect investment preferences		We will demo how users can enter their investment	

	Integration	(risk tolerance, budget, etc.). **(6 hours)** - Store preferences in JSON-based user profiles. **(6 hours)** - Integrate with the external allocation tool to generate recommendations. **(6 hours)** - Test recommendations for accurate allocation results. **(1 hour)**		preferences through the chatbot. The demo will include integration with the external allocation tool, showing personalized recommendations based on user preferences. Data persistence will also be demonstrated.	
	Resource allocation dashboard	- Develop a Dash-based dashboard to display resource allocation. **(3 hours)** - Create visualizations (pie charts) for allocations (stocks, bonds, mutual funds). **(3 hours)** - Integrate allocation results from an external tool into the dashboard.		We will demo a dynamic Dash-based dashboard that visualizes resource allocations (e.g., pie charts for stocks, bonds, and mutual funds). The demo will showcase the real-time update of allocation results and user data persistence across sessions.	

		(6 hours) - Ensure persistent data display on subsequent logins. **(6 hours)**			
Sprint 3 (Week 6 - 7)	Prediction and external forecasting tool integration.	- Integrate with the external forecasting tool for predicting future gains. **(24 hours)** - Retrieve and display predictions on the dashboard. **(24 hours)** - Ensure predictions update dynamically based on user allocation changes. **(24 hours)** - Perform end-to-end testing for system functionality. **(24 hours)**	**Milestone 3**: Final presentation on **December 3rd 2024**	We will demo the integration of the external forecasting tool that predicts potential future gains based on the user's investment allocations. The demo will include how predictions are updated dynamically in the dashboard based on allocation changes.	Not started
Sprint 4 (Week 8 -9)	Feedback system, review and final integration	- Implement a feedback collection system. **(24 hours)** - Display past	**Milestone 3**: Final presentation on **December 3rd 2024**	We will demo the complete system, including user feedback collection, the ability to view past interactions, and a fully	Not started

		conversations and historical data for returning users. **(24 hours)** - Store user feedback in the feedback database. **(24 hours)** - Conduct final testing for all system components (education, recommendation, predictions, feedback). **(24 hours)**		functional investment advisor system. The demo will show the integration of all components: education, recommendations, predictions, and feedback handling.	
Sprint 5 (Week 10 -11)	Testing and feedback	- Functional and integration testing. **(16 hours)** -Limited user testing and feedback. **(16 hours)** -Address Critical Issues. **(16 hours**) Final project deliverables mentioned in the syllabus. **(48 hours)**	**Milestone 3**: Final presentation on **December 3rd 2024**	Comprehensive testing result Functional and integration test reports. User feedback insights and recommendations Resolved critical issues and system refinements.	Not started

Note: Each sprint is allocated a total of 96 hours.

b) Resources needs

1. **Computing resources**:
   For the project, we will require access to several APIs and external tools to power the system. This includes the OpenAI GPT-3.5 API, which will be used to handle financial education queries, and the Tavily Web Search API for retrieving additional educational content. Additionally, we will integrate an external allocation tool for generating personalized investment recommendations based on user preferences. To predict future gains, we will also use an external forecasting tool. These tools will form the backbone of the system, ensuring accurate data retrieval, recommendations, and predictions.
2. **Development environment**:
   The backend of the system will be developed using the Flask framework, which is well-suited for building scalable web applications. The dashboard and visualization components will be built using Dash, a Python-based framework for creating interactive web applications. For persistent storage of user data and interactions, we will utilize a JSON file-based database, ensuring that user profiles and preferences are securely stored and can be accessed during subsequent sessions.
3. **Hosting**:
   The system will require cloud hosting to run the chatbot, backend, and dashboard components. Services like AWS may be used to host the entire system, providing scalability and availability for multiple users. While the team will handle all development efforts, the API keys and hosting capabilities will be provided or funded by CMU.

4. Plan for success:

a) Criteria for measuring outcomes:

1. **Milestone verification:** We will conduct scheduled demos on October 29th, 2024, and December 14th, 2024, showcasing completed features (e.g., chatbot, resource allocation, prediction system) to verify they meet user and system requirements.
2. **Deliverable reports:** Progress reports will accompany each milestone and be sent to our faculty advisor.
3. **Key performance indicators:** Users should be able to view a pie chart displaying their resource allocation (stocks, bonds, mutual funds) on the dashboard, and the system shall answer at least three financial questions correctly to ensure the accuracy of educational content provided by GPT-3.5 and Tavily integrations. Additionally, we will collect input-output data to verify that the system's responses match expected results and use benchmark datasets to further evaluate system accuracy and performance.
4. **User feedback:** After each major demo, we will collect feedback from the advisor and colleagues (classmates) to evaluate the system's usability, educational content, and recommendation accuracy.

5. **Testing and validation:** Each sprint will include unit and integration testing, with automated test cases ensuring functionality across education queries, user inputs, and dashboard updates.
6. **User acceptance testing**: In the final sprint, we will perform UAT with novice investors to gather feedback on the system's effectiveness and share the results in the final report.

b) Risk identification and handling:

1. **Organizational risks**:
   Communication delays could disrupt project flow. To mitigate this, we will hold regular team meetings and use project management tools to track tasks and deadlines efficiently. Additionally, time constraints may pose a challenge, so we will follow a scrum methodology to ensure incremental progress while allowing flexibility to adjust workloads based on project needs and unforeseen delays.
2. **Technological risks**:
   There is a risk that general LLM models may not be niche-specific enough for financial topics. To address this, we will integrate external APIs such as **Tavily** and fine-tune the models using relevant financial data to improve accuracy and relevance. Another concern is the lack of advanced web development expertise within the team. To overcome this, we will use user-friendly frameworks like Dash, prioritizing simplicity and functionality over visual complexity.
3. **Dependency risks**:
   Lack of access to premium financial data APIs could hinder the system's ability to provide accurate market data. To prevent this, we will secure the necessary API subscriptions early in the project.