College of Professional Studies: Northeastern University Toronto

ALY 6980 – Capstone Project

Instructor: Dr. Sohom Mandal

Academic Term: Spring 2025

Individual Project Proposal Analysis and Edits

UniAssist: LLM-Powered Conversational Assistant for University Support

Group B

Sheila Kwartemaa Boateng

Charishma Garikapati

Bismark Sarpong

Chu Zhang

May 30, 2025

Introduction

In the modern academic landscape, students encounter a wide range of challenges—from academic pressures to navigating complex university systems—requiring timely and effective support. Traditional channels such as help desks, emails, and phone hotlines are often insufficient to meet the increasing and immediate needs of students, leading institutions to seek innovative technological solutions.

Artificial intelligence (AI)–powered platforms, including chatbots and virtual assistants, are rapidly transforming the way universities provide student support. Leveraging advances in large language models (LLMs) and multi-agent architectures, these systems can deliver round-the-clock, personalized assistance, streamlining communication and improving the overall student experience.

This project focuses on the design, implementation, and evaluation of an AI-powered, multi-agent university chatbot specifically developed for Northeastern University's Toronto campus. Built using state-of-the-art LLMs and a multi-agent framework, the chatbot is tailored to address the unique needs of students, staff, and faculty by providing instant answers to questions about admissions, academics, campus life, and more.

While the current solution is customized for Northeastern Toronto, it is designed with scalability in mind. Future phases of this project will extend the platform to support additional Northeastern University campuses, enabling consistent and efficient student support services across the institution. By analyzing both quantitative and qualitative data related to chatbot usage and effectiveness, this study aims to inform best practices for deploying AI-driven support systems in higher education.

Audience

The primary audience for this survey comprises current students, faculty, and staff at Northeastern University Toronto. This diverse group includes undergraduate and graduate students across various programs, as well as faculty members and administrative staff who interact with campus support services. By engaging this audience, the study aims to capture a comprehensive perspective on the effectiveness, accessibility, and user satisfaction of the AI-powered chatbot platform.

While the initial focus is on the Northeastern Toronto campus, the survey design also considers potential scalability, allowing for future deployment and feedback collection from other Northeastern University campuses as the platform expands.

Model Information and Technologies Used

This project leverages state-of-the-art technologies and a modular architecture to deliver an advanced, scalable AI chatbot platform for university student support. Below is an overview of the key components and choices:

Large Language Model (LLM)

- **Model:** Mixtral-8x7B-Instruct-v0.1, accessed via Together AI API.
- **Integration:** The backend uses the OpenAI v1+ Python client, fully compatible with Together AI endpoints, enabling seamless prompt-based interaction with the Mixtral model.
- **Role:** The LLM serves as the primary conversational engine (“UniBot”), generating natural, context-aware responses to a wide range of university-related and general student queries.

Multi-Agent System

- **Orchestration:** The chatbot backend is structured using a multi-agent architecture, where specialized agents handle searching, summarizing, maintaining chat memory, and managing workflow.
- **Framework:** LangGraph is used for agent orchestration, supporting dynamic routing and coordination between agents.

External APIs

- **Real-time Search:** The Search Agent uses Serper.dev, a Google Search API, to fetch up-to-date information when required.
- **LLM Endpoint:** Together AI provides scalable and reliable access to the Mixtral model, supporting high-availability conversational tasks.

Backend

- **Framework:** FastAPI (Python) is employed for its high performance, asynchronous processing, and easy API definition.
- **Responsibilities:** Handles incoming queries, agent orchestration, session management, and communication with third-party APIs.
- **Deployment:** Hosted on Render, ensuring scalability and reliable uptime.

Frontend

- **Framework:** ReactJS, bundled with Vite, powers a responsive and intuitive chat interface.
- **Features:** Offers a modern, real-time chat experience accessible from desktop and mobile devices.
- **Deployment:** Deployed on Netlify for fast, global content delivery.

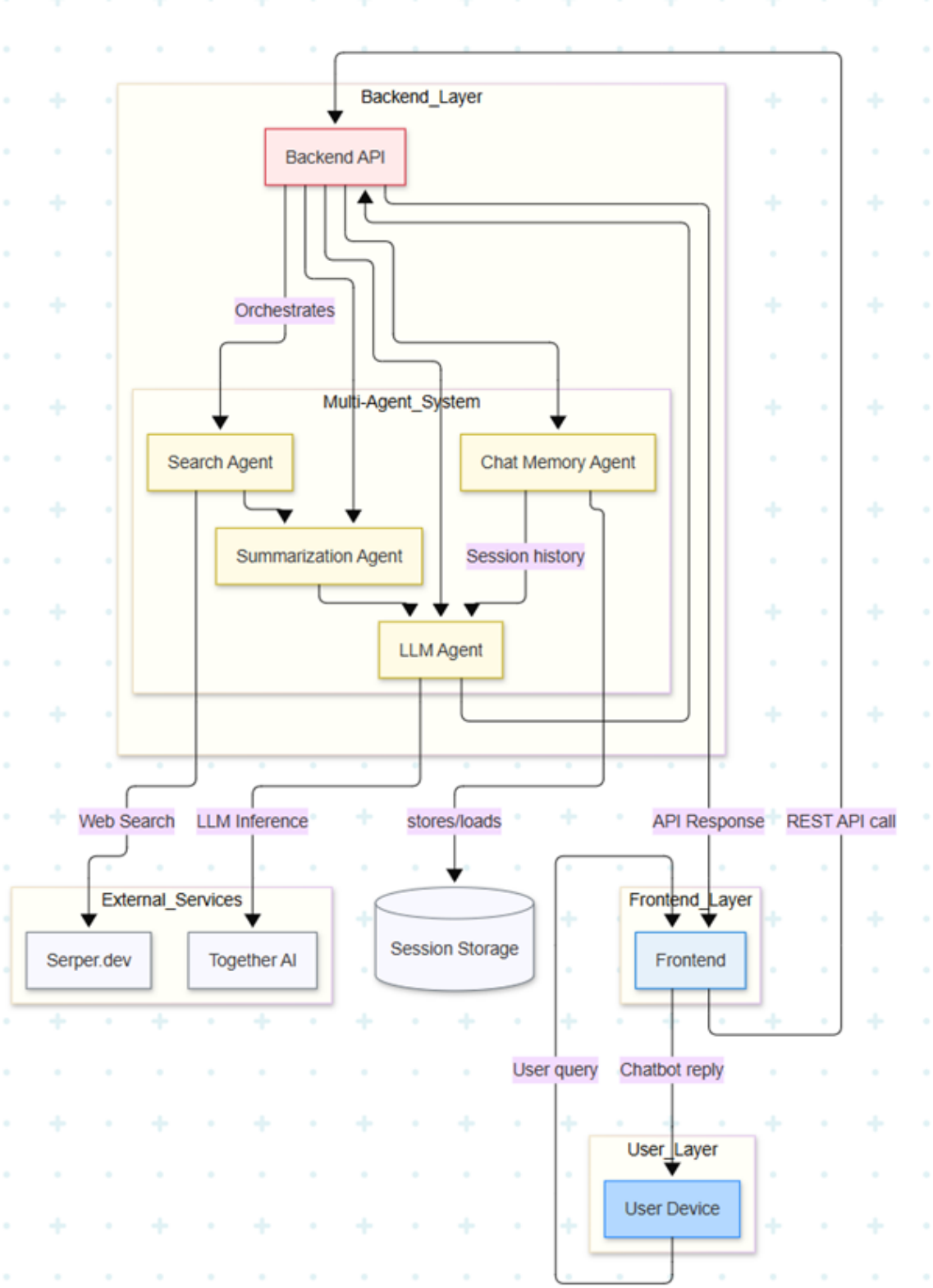
Project Scope and Future Scalability

- **Current Implementation:** The system is customized for Northeastern University Toronto, with tailored prompts and resource integration.
- **Scalability:** The modular architecture is designed for easy extension to additional Northeastern University campuses in future phases.

Summary Table:

Component	Technology/Service	Purpose
LLM	Mixtral-8x7B-Instruct-v0.1 (Together AI)	Natural language understanding/generation
Multi-Agent System	Python, LangGraph	Modular orchestration of tasks
Search Agent	Serper.dev (Google Search API)	Real-time information retrieval
Backend	FastAPI	API, orchestration, session management
Frontend	ReactJS, Vite	User chat interface
Deployment	Render (Backend), Netlify (Frontend)	Cloud hosting and scaling

Architecture Diagram



System architecture of the university chatbot platform. The design separates the user interface (frontend) from the backend API, which orchestrates a multi-agent system for web search, summarization, memory management, and large language model inference. The architecture incorporates persistent session storage and external APIs for both search and AI-powered response generation.

Flow Description

1. **User** sends a question via the web chat interface.
2. The **Frontend** relays the request to the **Backend API**.
3. The **Backend** orchestrates specialized agents to:
    - Search the web (Search Agent via Serper.dev)
    - Summarize retrieved info (Summarization Agent)
    - Maintain session context (Chat Memory Agent, with storage)
    - Generate a conversational answer (LLM Agent via Together AI)
4. The final answer, with supporting information, is sent back through the API to the **Frontend** and displayed to the user.

Project Links

- **Frontend (Live Demo):** https://candid-yeot-40275f.netlify.app/
- **Backend (API Endpoint):** https://university-bot-e4rm.onrender.com/ask
- **GitHub Repository:** https://github.com/charishmag21/university-bot/tree/dev

Conclusion

This project demonstrates the design and implementation of a modern, AI-powered university chatbot system, leveraging a multi-agent architecture and state-of-the-art large language models to deliver effective, scalable, and responsive support for students, faculty, and staff. By integrating real-time web search, summarization, conversation memory, and advanced language generation, the platform addresses common gaps in traditional student support services—improving access to information and enhancing the overall user experience.

The modular design ensures that the solution is not only tailored to the unique needs of Northeastern University Toronto but is also easily adaptable to other campuses and academic institutions in the future. Through quantitative and qualitative analysis of user interactions and satisfaction, the project lays the groundwork for continuous improvement and data-driven optimization of digital student services.

Looking ahead, the system's architecture supports seamless integration of additional features, further customization, and broader deployment. As artificial intelligence continues to advance, this chatbot platform can serve as a foundation for innovative, student-centric support solutions across higher education.

References

- Together Computer. (2024). *Mixtral-8x7B-Instruct-v0.1 Model Inference API Documentation*. Together AI. https://docs.together.ai/docs/inference
- LangChain. (2024). *LangGraph Documentation*. Read the Docs. https://langgraph.readthedocs.io/
- Serper. (2024). *Serper.dev API Documentation*. Serper. https://serper.dev/
- Tiangolo, S. (2024). *FastAPI Documentation*. FastAPI. https://fastapi.tiangolo.com/
- Meta (React Contributors). (2024). *React Documentation*. React. https://react.dev/
- Vite Contributors. (2024). *Vite Documentation*. Vite. https://vitejs.dev/
- Render. (2024). *Render Documentation*. Render. https://render.com/docs
- Netlify. (2024). *Netlify Documentation*. Netlify. https://docs.netlify.com/
- OpenAI. (2024). *OpenAI Python API Reference*. OpenAI. https://platform.openai.com/docs/api-reference
- GitHub, Inc. (2024). *GitHub Documentation*. GitHub. https://docs.github.com/