Team Contribution Report

Group – 2

University of Missouri – Kansas City

COMP\_SCI5588: Data Science Capstone

Yugyung Lee

March 4th, 2024

# Team Contribution Report

## Business Model Slide Deck:

<https://github.com/TarunSiga/DSCapstoneProject/blob/main/ProjectPhase3_1/UniBuddy_Business%20Model%20Slide%20Deck.pptx>

## Pitch Video:

<https://www.youtube.com/watch?v=V67qbIrOyAc>

## Team Contribution:

**Deepak Ayyasamy: Machine Learning Development**

**Ingestion Module Development**:

* Developed ingest.py module responsible for loading documents from specified directories and splitting them into appropriate formats.
* Implemented multithreading and multiprocessing techniques for efficient document loading and processing.
* Incorporated error handling mechanisms to manage exceptions during the document loading process.
* Created a logging system (file\_ingest.log) to track document loading progress and any encountered errors.

**Local GPT Module Development:**

* Developed run\_localGPT.py module responsible for running retrieval question-answer tasks using Language Chain's RetrievalQA pipeline.
* Implemented model loading functions to load pre-trained models based on specified configurations.
* Integrated embeddings and vector stores for efficient text representation and retrieval.
* Leveraged Mistral 7B quantized version as LLM model and “instructor-large” as embedding model.
* Implemented callbacks for streaming responses and managing output verbosity.
* Provided options for users to specify device type, show/hide source documents, use history, and select model types.

**Code Refinement and Documentation:**

* Refactored codebase for improved readability and maintainability.
* Added comprehensive inline documentation to explain module functionalities and code logic.
* Conducted thorough testing and debugging to ensure code correctness and reliability.
* Incorporated user-friendly CLI interfaces using the Click library for easy parameter configuration.

**Challenges Faced:**

* **Concurrency Management**: Implementing efficient concurrency management strategies for document loading and processing required careful consideration to avoid race conditions and deadlocks.
* Addressed issues related to thread and process synchronization to ensure smooth execution.
* **Model Loading and Configuration:** Ensuring compatibility and proper configuration of pre-trained models posed challenges, especially in managing different model types and device types.
* Implemented robust error handling mechanisms to handle model loading failures and configuration errors gracefully.
* **Optimization and Performance Tuning:** Optimizing document loading and processing performance while minimizing resource utilization was a significant challenge.
* Explored various optimization techniques and fine-tuned parameters to achieve a balance between performance and resource efficiency.

**Next Steps**:

* API integration to leverage Local GPT anywhere, anytime.
* Progressing towards building a successful end-end application that can accept multiuser queries and provide solutions.
* Scaling up computing resources to reduce retrieval time of the model.

**Tarun Siga: Front End Development**

**Enhanced user profile management:**

* Implemented a feature for users to edit their profile information such as name, email, profile picture, etc.
* Allow users to customize their profile settings and preferences.

**Improved chat experience:**

* Implemented typing indicators to show when a user is typing a message.
* Added read receipts to indicate when a message has been read by the recipient.
* Implemented emojis and stickers for richer communication.
* Introduced message threading to organize conversations more effectively.

**Challenges Faced:**

* Handling concurrency issues and race conditions in a multi-user chat environment.
* Implemented locking mechanisms or optimistic concurrency control techniques to prevent race conditions and ensure data integrity. Use transactions and atomic operations where necessary to maintain consistency.

**Next Steps:**

**Voice Recognition Integration:**

* Voice recognition integration aims to enhance the user experience of UniBuddy by allowing users to interact with the chatbot using voice commands.
* Voice recognition feature will provide an intuitive and convenient alternative to typing, particularly useful for users with accessibility needs or those on mobile devices.

**Pitch Video Samples:**

<https://www.youtube.com/watch?v=V67qbIrOyAc>

<https://www.youtube.com/watch?v=rYxkBN_WLQU>

**Sai Karthik Naladala: Back End Development**

* Created a visually appealing login/register form using HTML and CSS.
* Integrates Firebase authentication for user registration and login.
* Provides functionality to switch between the login and registration forms.
* Includes a section to display a Streamlit app using an iframe.
* Initializes Firebase using the provided configuration.
* Implements user registration and login functions using Firebase's authentication methods.
* Validates user input for email, password, full name, university, and phone number.
* Redirects the user to the Streamlit application after successful registration or login.
* Defines styles for various elements of the form, including fonts, colors, and dimensions.
* Creates a visually appealing layout with background gradients and rounded corners.
* Ensures consistent styling across different input fields and buttons.
* Sets up a Streamlit app with a chat interface.
* Integrates Firebase Firestore for storing chat history.
* Implements speech recognition using the SpeechRecognition library for voice input.
* Displays chat history and handles user input with appropriate responses.

**Firebase Integration**: Integrated Firebase Admin SDK to interact with Firestore, enabling seamless access to Firestore collections and documents within the Streamlit app.

**Dynamic Collection Retrieval:** Developed a function to dynamically retrieve all collection names from Firestore, ensuring flexibility in selecting data for export.

**Efficient Data Extraction:** Implemented a method to fetch all documents from selected collections, extracting relevant data fields such as prompt and response for export.

**User-Friendly Interface:** Designed an intuitive Streamlit interface allowing users to easily select collections and initiate data export with options for single or multiple selections.

**Automated CSV Generation:** Utilized the CSV module to automatically generate a structured CSV file containing extracted Firestore data, simplifying data management and analysis.

**Challenges faced:**

**Firebase Integration:** Setting up Firebase Admin SDK and Firestore presented challenges due to authentication and connection configurations.

**Streamlit UI:** Designing a user-friendly interface with Streamlit required careful consideration of layout and functionality to ensure smooth user experience.

**CSV Export**: Implementing CSV export functionality involved handling file paths, permissions, and ensuring proper data formatting.

**Data Retrieval:** Retrieving data from Firestore collections involved dealing with asynchronous requests and ensuring correct data transformation for further processing.

**Prompt Integration:** Integrating prompt delivery to a fine-tuned model and retrieving responses required understanding and integrating with external APIs or services, potentially involving authentication and data serialization challenges.

**Error Handling:** Implementing robust error handling mechanisms to handle potential failures in data retrieval, CSV export, or integration with external services.

**Next steps:**

* Currently working on integrating functionality to send prompts to a fine-tuned model and retrieve responses.
* Currently developing Unibuddy's Collaboration Hub, fostering interdisciplinary connections and innovation among students from diverse backgrounds.
* Implementing personalized deadline alerts in Unibuddy, enhancing student engagement and academic support through tailored notifications

**Pitch Video Sample:** <https://www.youtube.com/watch?v=2GFr8rnRd0g>.