Final Software Design Document

Author
Author of the project and document.

- Marco Li

Reviewer
Stakeholder of the project and document

- Alex Rossie

Goals and Milestones
Goals and milestones separated in weeks.

Week 1
- Project plan and timeline.
- List of required technologies and tools.
- Initial setup of development environment (IDE, version control, etc.).

Week 2
- Basic web application with user login/ signup functionality.
- Initial layout and design of the application.

Week 3
- The user should be able to get a natural language response from any input via the frontend.

Week 4
- Enhanced chatbot capable of maintaining conversation context.
- UI improvements to display conversation history.

Week 5
- Test cases and testing reports.
- Bug fixes and performance improvements.

Week 6
- Final optimized web application that runs locally.
- A showcase video demo.
- Comprehensive project documentation.

Functional Description
Purpose of the software.

This project aims to create a web-based chatbot using an open-source Large Language Model (LLM) that can run locally. The chatbot will understand natural language questions and give accurate answers and can be entirely run on a local machine. This project will use React as front-end, Express as back-end with Python as a microservice to run Transformers (by huggingface) API to efficiently interface with a desired LLM.

Tools
Softwares and technologies required.

Choice of LLM
- Llama3
  - Pros: Open source, versatile, large community, customizable, new technology
  - Cons: Resource intensive, performance variability, bad at following instructions without guidance.ai
  - Why Llama3: Although resource intensive, it can still be run on the current development machine while maintaining a reasonable amount of model size of 7B parameters.

Alternate choices of LLM
- GPT-J
  - Pros: Open source, powerful, versatile, large community, customizable.
  - Cons: Extremely resource intensive. Unable to run on most machines. Must be fine tuned before use.
- GPT-2
  - Pros: Efficient, open source, good performance.
  - Cons: Low model size, underwhelming responses compared to alternatives.
- BERT
  - Pros: Excellent performance, resource efficient, widely adopted, effective fine-tuning.
  - Not the best for text generation, larger models need resources, slower inference.

Development Environment: VS Code **Version Control:** Git with GitHub **Design:** Figma

Front-end: Next.js **Back-end:** Python **Authentication:** Cognito

Database: DynamoDB **DNS:** Route 53 **Distribution:** Cloudfront

Optional Functionalities
Optional functionalities can be added if time permits.

- User authentication, conversation history, and real-time interactions.
- Front-end in a mobile framework (e.g. React Native)
- Prompt engineering layer to the chatbot.

Component Diagram