Software Requirements Specification

for

Behavioral Cognition Project

Version 2.1 approved

Prepared by

Arturo Rodriguez, Daniel Oniveros, David Santini

Deep Bhakta, Erick Herrera, Iris Ha, Izeth Torres,

Jorge Pena, Shant Hovagimian, Yizhang Cao

**Behavioral Cognition**

**May 8, 2024**

**Table of Contents**

Table of Contents................................................................................................................. pg 2

Revision History................................................................................................................... pg 3

1. Introduction................................................................................................................ pg 4

1.1. Purpose........................................................................................................... pg 4

1.2. Intended Audience and Reading Suggestions................................................ pg 4

1.3. Product Scope................................................................................................ pg 4

1.4. Definitions, Acronyms, and Abbreviations .................................................. pg 4

1.5. References...................................................................................................... pg 4

2. Overall Description.................................................................................................... pg 5

2.1. System Analysis…......................................................................................... pg 5

2.2. Product Perspective........................................................................................... pg 5

2.3. Product Functions........................................................................................... pg 6

2.4. User Classes and Characteristics.................................................................... pg 6

2.5. Operating Environment.................................................................................. pg 6

2.6. Design and Implementation Constraints........................................................ pg 7

2.7. User Documentation...................................................................................... pg 7

2.8. Assumptions and Dependencies.................................................................... pg 7

2.9. Apportioning of Requirements...................................................................... pg 7

3. External Interface Requirements............................................................................... pg 7

3.1. User Interfaces............................................................................................... pg 7

3.2. Hardware Interfaces....................................................................................... pg 8

3.3. Software Interfaces........................................................................................ pg 8

3.4. Communications Interfaces........................................................................... pg 8

4. Requirements Specification....................................................................................... pg 8

4.1. Functional Requirements............................................................................... pg 8

4.2. External Interface Requirements................................................................... pg 9

4.3. Logical Database Requirements.................................................................... pg 9

4.4. Design Constraints......................................................................................... pg 9

5. Other Nonfunctional Requirements........................................................................... pg 9

5.1. Performance Requirements............................................................................ pg 9

5.2. Safety Requirements...................................................................................... pg 9

5.3. Security Requirements................................................................................... pg 9

5.4. Software Quality Attributes........................................................................... pg 9

5.5. Business Rules............................................................................................... pg 10

6. Legal and Ethical Considerations.….......................................................................... pg 10

Appendix A: Glossary........................................................................................................ pg 11

Appendix B: Analysis Models........................................................................................... pg 11

Revision History

| Name | Date | Reason For Changes | Version |
| --- | --- | --- | --- |
| Iris Ha | 12/4/23 | Starting documentation | 1.0 |
| Yizhang Cao | 5/1/2024 | Revising documentation | 2.0 |
| Erick Herrera | 5/1/2024 | Revising documentation | 2.1 |
| Iris Ha | 5/1/2024 | Revising documentation | 2.1 |

**1. Introduction**

California State University Los Angeles Computer science senior design project partnered with Behavioral Cognition to create a toolkit that makes creating language processing easier. The language processing tool kit (lp toolkit) is designed to be easy to use with minimal knowledge of website application and artificial intelligence(AI). Currently, lp toolkit is a chat box that can answer simple questions using langchain openAI.

**1.1 Purpose**

Behavioral Cognition documentation for the language processing toolkit(lp-toolkit) version 2.1. This document will go over the broad and overall architecture of the lp-toolkit software from the usage to restrictions.

**1.2 Intended Audience and Reading Suggestions**

This document is intended for developers and those interested in AI. For developers using lp-toolkit for personal use it is recommended to go through sections 1-5. For developers using it for business related purposes it is recommended to go through sections 1-6. For those interested in AI and intend to use lp-toolkit to build your own chatbot it is recommended to go through sections 2,3 and 4.

**1.3 Product Scope**

lp-toolkit simplifies AI development for developers, allowing them to create their own AI solutions without needing comprehensive understanding of AI principles. The goal of lp-toolkit is to be a toolkit with batteries included; easy to grasp and deployable in a short time.

**1.4 Definitions, Acronyms, and Abbreviations**

* LLM (Large Language Model)
* AI (Artificial Intelligence)
* language processing toolkit (lp-toolkit)
* API (application programming interface)

**1.5 References**

1. **Docker Documentation.** (n.d.). Retrieved Month Day, Year, from https://docs.docker.com/

Official Docker documentation providing detailed guidelines on containerization concepts, setup instructions, and best practices for Docker files and commands.

1. **MongoDB Manual.** (n.d.). Retrieved Month Day, Year, from https://docs.mongodb.com/manual/

MongoDB’s official manual, including details on installation, configuration, and best practices for database design and performance tuning.

1. **Node.js Documentation.** (n.d.). Retrieved Month Day, Year, from https://nodejs.org/en/docs/

Documentation for Node.js, offering insights into its asynchronous programming model and various APIs supporting backend development.

1. **Stack Overflow. (n.d.)**. Discussions on containerization with Docker. Retrieved Month Day, Year, from https://stackoverflow.com/questions/tagged/docker

A collection of community discussions and problem-solving related to the use of Docker in development environments, providing practical insights and troubleshooting tips.

**2. Overall Description**

Computer science senior design project partnering with Behavioral Cognition to make a toolkit that makes creating language processing easier. The language processing tool kit (lp toolkit) is made to be easy to use without needing the knowledge of how to build a web application and AI. At the current stage lp toolkit is a simple chat box that can answer simple questions using langchain openAI.

**2.1 System Analysis**

lp-toolkit stands as a research-driven endeavor crafted specifically to cater to developers, offering a streamlined avenue to construct their personalized AI systems. Designed with a "batteries included" approach, it provides developers the flexibility to integrate any Language Model they prefer into their AI projects. This approach liberates developers from model constraints, enabling them to use the power of diverse Language Models to tailor AI solutions that precisely meet their unique requirements and objectives.

**2.2 Product Perspective**

lp-toolkit operates within a Docker environment, leveraging containerization for its build and execution processes, while the user interface is accessed through standard web browsers. Conceptually akin to platforms like ChatGPT and Google Bard, lp-toolkit aligns itself with these AI-driven tools in offering a user-friendly interface for specific tasks. In a similar vein, ChatClient stands as another product in the realm of lp-toolkit's objectives, providing an online platform where users can construct personalized AI chatbots, reflecting a shared goal of empowering users in harnessing AI capabilities for their needs.

**2.3 Product Functions**

Users can choose between two models currently implemented; ChatGPT and Huggingface. After choosing one of the two models the user can enter a query through the chat box and receive a response to the query. The selected model will answer the query using AI.

The project should be able to use other LLM and users should be able to use it via the frontend in the same manner as stated previously.

**2.4 User Classes and Characteristics**

In the lp-toolkit ecosystem, two distinct user levels define the access and functionalities available: regular users and admin users. Regular users are granted access to the browser-based chat user interface, providing them with the primary interaction platform for utilizing the toolkit's features. This interface offers a streamlined experience tailored to meet the specific needs and tasks of regular users, facilitating their engagement with the toolkit's functionalities.

In contrast, admin users enjoy elevated access privileges that encompass not only the regular user interface but also an additional admin user interface web page. This expanded access gives admin users enhanced functionalities, allowing them to oversee and manage various aspects of the toolkit. Through the admin interface, they gain control over administrative tasks, such as user management, system configurations, and potentially additional tools or settings exclusive to administrative roles.

By differentiating between regular and admin users, the lp-toolkit ensures a tailored experience for each user level, catering to their respective needs and responsibilities within the toolkit's environment. This hierarchical structure aims to optimize usability while providing necessary controls and capabilities to effectively manage the toolkit's operations.

**2.5 Operating Environment**

The lp-toolkit is a versatile web-based tool designed for seamless execution via standard web browsers. It harnesses the power of Docker, offering a robust environment for users to optimize workflows and streamline processes. While Docker isn't mandatory for lp-toolkit, its utilization is highly recommended to ensure optimal performance and compatibility across diverse systems. Users engage with the toolkit through simple yet powerful commands executed within their command line or terminal environment. These commands encompass building and running lp-toolkit, providing users with the flexibility and control to navigate the system effortlessly.

By leveraging Docker, lp-toolkit simplifies deployment while maintaining consistency across various platforms. This approach not only ensures a consistent operational environment but also simplifies the deployment process across different systems. By prioritizing accessibility and ease of use, lp-toolkit enables users, regardless of their technical expertise, to seamlessly integrate the toolkit into their workflows. This user-friendly approach fosters inclusivity, empowering a diverse range of users to benefit from lp-toolkit's resources and functionalities with minimal setup and maximum efficiency.

**2.6 Design and Implementation Constraints**

Current Implementation uses two LLM, ChatGPT and Hugging face. ChaptGPT will require an API key. Tools required for general implementation are npm, docker and docker-compose.

**2.7 User Documentation**

Instructions for building and running lp-toolkit will be provided in the README.md files in the main directory. Build and run instructions for sub directories(frontend, backend, langchain, etc.) are also provided in README.md file in respective sub directories.

**2.8 Assumptions and Dependencies**

The user will have already installed Docker to run the program in one command. Users should also have an API key when the LLM demands it.

**2.9 Apportioning of Requirements**

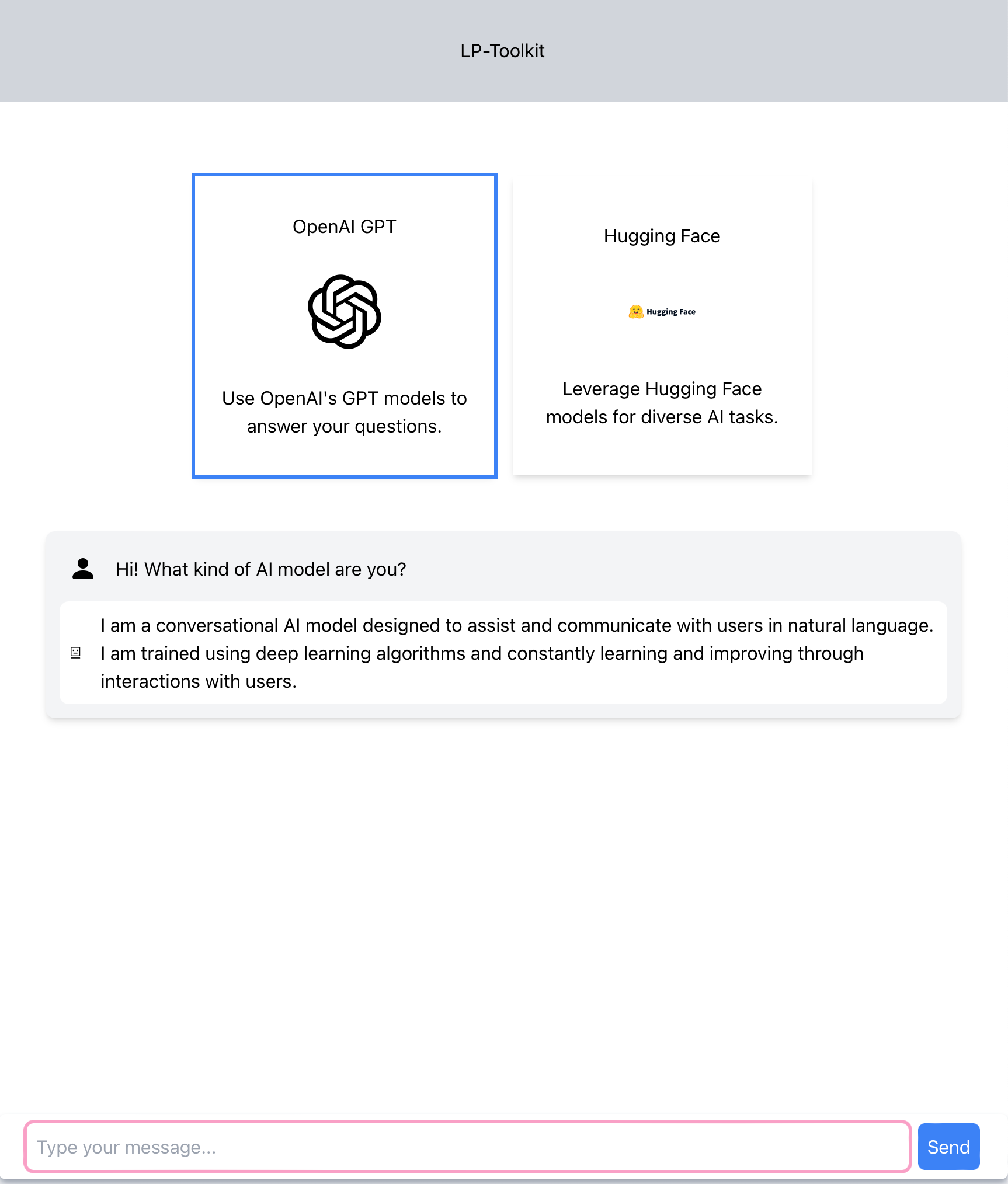
A Vectorizer is to be implemented. A continuous chat log from switching between the implemented models should also be added.

**3. External Interface Requirements**

**3.1 User Interfaces**

At any point of the conversation, any of the implemented models can be used as the AI that carries out the tasks given. A button with a brief description of the AI should be implemented. Clicking this button switches the AI model.

Current user interaction is limited to asking a question via the textbox, viewing past questions and answers given(when implemented), and the buttons that switch AI models.

****

**3.2 Hardware Interfaces**

This project does not interact with hardware interfaces; it is purely software.

**3.3 Software Interfaces**

HuggingFace, Langchain, FastAPI, ExpressJs, React, MongoDB 7.0, OpenAI, Docker

**3.4 Communications Interfaces**

This program uses a browser to display the GUI which users can interact with. From executing the project the frontend is running on port 3000/tcp. The backend is running through port 5050/tcp and MongoDB is running through port 27017/tcp.

**4. Requirements Specification**

**4.1 Functional Requirements**

The system must take requests through the proxy which routes the request to the frontend and backend. API calls are made to the backend while static assets, HTML, css and images are made to the frontend. Backend API calls to the database which then routes into data analytics and data modeling.

**4.2 External Interface Requirements**

User Input: Input given will be a question to be passed to the AI to answer. Users will type their question into the textbox and submit using the “Send” button.

AI model Input: Button that chooses which AI model to use to answer User Input.

Response Output: Output will be the response that the AI gives when given User Input.

**4.3 Logical Database Requirements**

When queries are sent to the backend, the MongoDB database will take data and send the corresponding data into the Data Analytics and into the Data Modeling.

**4.4 Design Constraints**

The system must operate efficiently under the load of up to 10,000 concurrent users.

The interface must be compatible with major browsers like Chrome, Firefox, and Edge. Should be able to easily implement other LLMs. A simple button press is all that is needed to switch LLM.

**5. Other Nonfunctional Requirements**

**5.1 Performance Requirements**

All queries sent shall be processed depending on the LLM chosen. Should be able to handle 10,000 concurrent users. Information to be handled will be in the form of text.

**5.2 Safety Requirements**

It is important to note that AI is not perfect as it can sometimes give faulty or even harmful information. A disclaimer should be made towards the user that the AI will not always be 100% correct and to not allow the chatbox to make big decisions(medical, law advice, etc.).

**5.3 Security Requirements**

As this project runs locally on the user’s machine, there is no current need for user identity authentication.

**5.4 Software Quality Attributes**

Developers will be able to easily implement any LLM and make API calls to use them. Any LLM can be used and should work with this software. Installation of this software should be as seamless as possible as accessibility is key.

**5.5 Business Rules**

Users should be able to customize the AI by setting the temperature and/or setting the token limit. Other features offered by the chosen LLM should also be displayed and accessible to the user. Customization should be easily presented to the user whenever possible to create an AI that is catered specifically for the user.

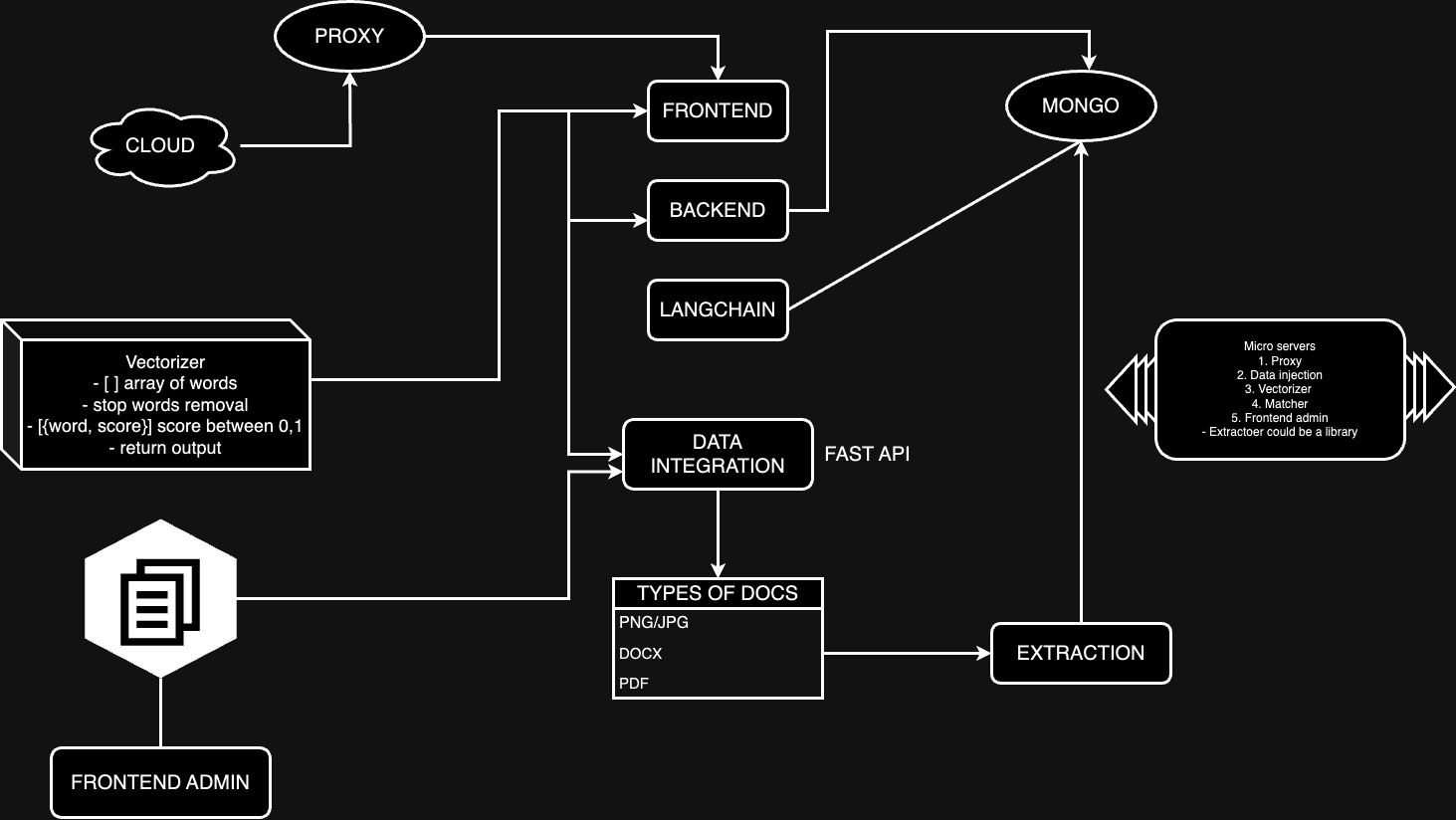
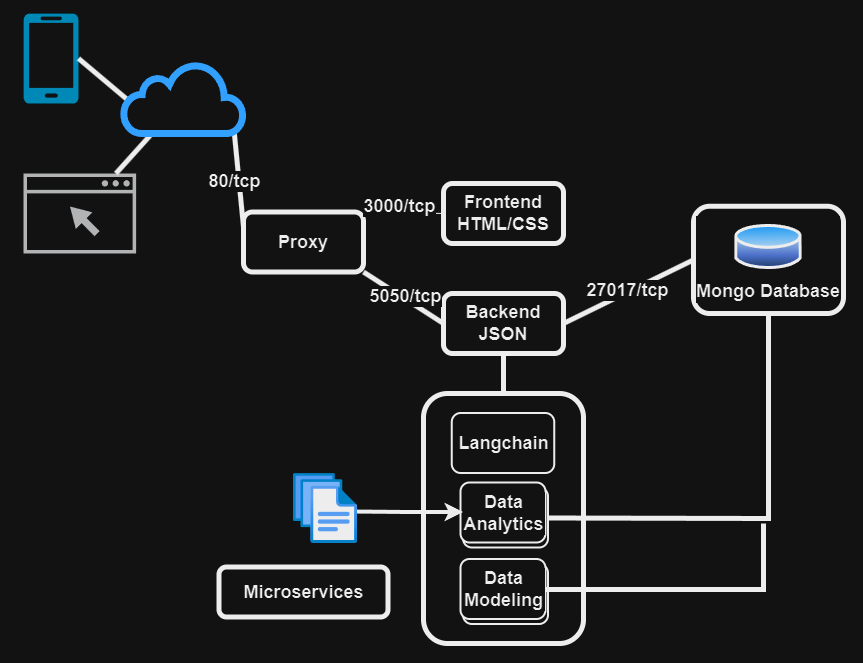
**6. Legal and Ethical Considerations**

As this project uses AI, it is possible that the AI gives unethical or dangerous advice, moreover the AI can be taught to give advice/instruction on how to do unethical/illegal activities. It is assumed that the user will use the software for reasonable uses.

**Appendix A: Glossary**

* LLM (Large Language Model)
* AI (Artificial Intelligence)
* language processing toolkit (lp-toolkit)
* API (application programming interface)

**Appendix B: Analysis Models**

****