

Software Requirements Specification

for

Behavioral Cognition Project

Version 1.0 approved

Prepared by Deep Bhakta, Yizhang Cao, Iris Ha, Erick Agustin Herrera, Shant Hovagimian, Daniel Ontiveros, Jorge Pena, Arturo Rodriguez, David Santini, Izeth Torres

Behavioral Cognition

September 22, 2023

Table of Contents

Table of Contents.....	pg 2
Revision History.....	pg 3
1. Introduction.....	pg 4
1.1. Purpose.....	pg #
1.2. Intended Audience and Reading Suggestions.....	pg #
1.3. Product Scope.....	pg #
1.4. Definitions, Acronyms, and Abbreviations	pg #
1.5. References.....	pg #
2. Overall Description.....	pg #
2.1. System Analysis.....	pg #
2.2. Product Perspective.....	pg #
2.3. Product Functions.....	pg #
2.4. User Classes and Characteristics.....	pg #
2.5. Operating Environment.....	pg #
2.6. Design and Implementation Constraints.....	pg #
2.7. User Documentation.....	pg #
2.8. Assumptions and Dependencies.....	pg #
2.9. Apportioning of Requirements.....	pg #
3. External Interface Requirements.....	pg #
3.1. User Interfaces.....	pg #
3.2. Hardware Interfaces.....	pg #
3.3. Software Interfaces.....	pg #
3.4. Communications Interfaces.....	pg #
4. Requirements Specification.....	pg #
4.1. Functional Requirements.....	pg #
4.2. External Interface Requirements.....	pg #
4.3. Logical Database Requirements.....	pg #
4.4. Design Constraints.....	pg #
5. Other Nonfunctional Requirements.....	pg #
5.1. Performance Requirements.....	pg #
5.2. Safety Requirements.....	pg #
5.3. Security Requirements.....	pg #
5.4. Software Quality Attributes.....	pg #
5.5. Business Rules.....	pg #
6. Legal and Ethical Considerations.....	pg #
Appendix A: Glossary.....	pg #
Appendix B: Analysis Models.....	pg #
Appendix C: To Be Determined List.....	pg #

Revision History

Name	Date	Reason For Changes	Version
Iris Ha	12/4/23	Starting documentation	1.0

<Add rows as necessary when the document is revised. This document should be consistently updated and maintained throughout your project. If ANY requirements are changed, added, removed, etc., immediately revise your document.>

1. Introduction

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.

1.1 Purpose

Behavioral Cognition documentation for the language processing toolkit(lp-toolkit) version 1.0. This document will go over everything there is about lp-toolkit 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 & 4.

1.3 Product Scope

lp-toolkit simplifies AI development for developers, allowing them to create their own AI solutions without requiring a comprehensive understanding of AI principles. The goal of lp-toolkit is to be a toolkit with batteries included. Easy to grasp how to use in a short time.

1.4 Definitions, Acronyms, and Abbreviations

- LLM (Large Language Model)
- AI (Artificial Intelligence)
- language processing toolkit (lp-toolkit)

1.5 References

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 them 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 harness 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

lp-toolkit has a chatbox where the user can enter a query and receive a response to the query.

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 empowers admin users with 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

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

2.9 Apportioning of Requirements

3. External Interface Requirements

3.1 User Interfaces

3.2 Hardware Interfaces

3.3 Software Interfaces

3.4 Communications Interfaces

4. Requirements Specification

4.1 Functional Requirements

4.2 External Interface Requirements

4.3 Logical Database Requirements

4.4 Design Constraints

5. Other Nonfunctional Requirements

5.1 Performance Requirements

5.2 Safety Requirements

5.3 Security Requirements

5.4 Software Quality Attributes

5.5 Business Rules

6. Legal and Ethical Considerations

Appendix A: Glossary

Appendix B: Analysis Models

Appendix C: To Be Determined List