



BSc. (Hons) in Software Engineering
Department of Information Technology
Faculty of Computing

H.A. Saduni Vindhaya MadhumaliAmarasekara
F20020521004
Software Testing

Software Requirements Specification (SRS) for ChatBot Application

Table of Contents

1.Introduction	
1.1 Purpose.....	2
1.2 Scope	2
1.3 Intended Audience.....	2
1.4 Definitions, Acronyms, Abbreviations.....	2
2.Overall Description	
2.1Product Perspective	3
2.2 Product Functions.....	3
2.3 User Classes and Characteristics.....	3
2.4 Operating Environment.....	4
2.5 Design and Implementation Constrains.....	4
2.6 Assumptions and Dependencies.....	4
3. Specific Requirements.....	
3.1 Functional Requirements.....	5
3.2 Non-Functional Requirements.....	5
4. External Interface Requirements.....	
4.1 User Interface.....	7
4.2 Software Interface.....	7
5.Appendices.....	
5.1 Technology Stack.....	7
5.2 Future Enhancements.....	7

1. Introduction

1.1 Purpose

The purpose of this document is to outline the software requirements for the ChatBot web application. The chatbot allows users to interact with an AI model to get assistance, answers to questions, and general conversational support. This SRS will guide the development and validation of the chatbot system.

1.2 Scope

The chatbot is a web-based application that provides an interactive interface between users and an AI-powered backend via OpenRouter's chat completion API.

- ❖ A web-based interface accessible through modern web browsers.
- ❖ Integration with OpenRouter's chat completion API to process user queries.
- ❖ Support for real-time user input handling through a text input field.
- ❖ Dynamic message rendering for both user and AI responses.
- ❖ Typing indicator displayed while waiting for the AI response.
- ❖ Error handling for API failures, invalid keys, and empty inputs.
- ❖ Designed for general users looking for fast and intelligent AI-driven conversations.
- ❖ Supports basic natural language interaction without requiring technical knowledge.

1.3 Intended Audience

- ❖ **Developers:** To understand functional and technical requirements.
- ❖ **Testers:** To validate the behaviour against the specified requirements.
- ❖ **Project Managers:** To track development progress.
- ❖ **End Users:** For understanding the chatbot's capabilities.

1.4 Definitions, Acronyms, Abbreviations

- ❖ **API:** Application Programming Interface
- ❖ **UI:** User Interface
- ❖ **SRS:** Software Requirements Specification
- ❖ **AI:** Artificial Intelligence

2. Overall Description

2.1 Product Perspective

- ❖ The chatbot is a standalone web application, requiring no installation.
- ❖ It uses **Open Router's Chat API** as the **backend** to process and respond to user messages.
- ❖ The **frontend** is built using **HTML, CSS, and JavaScript** for an interactive user experience.
- ❖ Communication with the **API** is handled using asynchronous **JavaScript** (fetch API).
- ❖ The UI supports dynamic rendering of messages, emojis, and markdown formatting.
- ❖ No additional server-side logic is implemented beyond the API integration.

2.2 Product Functions

- ❖ Accept user input in natural language.
- ❖ Send user messages to a backend AI API.
- ❖ Display AI-generated responses in a chat format.
- ❖ Show typing indicators while waiting for responses.
- ❖ Support emojis, special characters, and long conversations

2.3 User Classes and Characteristics

- ❖ **General Users:** No technical background. Want fast and accurate responses.
- ❖ **Developers:** Use it for testing API responses.
- ❖ **Mobile Users:** Access via responsive design

2.4 Operating Environment

- ❖ Web browsers (Chrome, Firefox, Safari, Edge)
- ❖ Responsive for both desktop and mobile devices
- ❖ Requires internet connection to access API

2.5 Design and Implementation Constrains

- ❖ Dependency on third-party API (OpenRouter)
- ❖ Limited usage based on API key restrictions
- ❖ Real-time interaction needs low-latency response

2.6 Assumptions and Dependencies

- ❖ Internet access is always available.
- ❖ OpenRouter's API is functioning properly.
- ❖ Users have modern browsers with JavaScript enabled.

3. Specific Requirements

3.1 Functional Requirements

- ❖ FR1: System shall allow users to send messages via a text box.
- ❖ FR2: System shall display user and bot messages in a chat format.
- ❖ FR3: System shall call the API and display responses asynchronously.
- ❖ FR4: System shall show a typing indicator while waiting for response.
- ❖ FR5: System shall handle invalid API keys with an error message.

3.2 Non-Functional Requirements

- ❖ NFR1: The response time shall be less than 3 seconds for 90% of messages.
- ❖ NFR2: The chat window shall support responsive layout.
- ❖ NFR3: The system shall handle 1000+ character inputs without crashing.
- ❖ NFR4: The application shall provide error messages on API failure.

4. External Interface Requirements

4.1 User Interfaces

- ❖ Welcome screen with a “Start Chat” button
- ❖ Chat interface with input box, message area, and back button
- ❖ Typing indicator animation

4.1 Software Interface

OpenRouter API: POST <https://openrouter.ai/api/v1/chat/completions>

Headers: Authorization, Content-Type

Body: Model, Messages

5. Appendices

5.1 Technology Stack

- ❖ **Frontend:** HTML, CSS, JavaScript
- ❖ **API Provider:** OpenRouter
- ❖ **AI Model:** deepseek/deepseek-r1:free

5.2 Future Enhancements

- ❖ Add voice input/output support
- ❖ Implement user authentication
- ❖ Maintain conversation history in local storage or database
- ❖ Support multiple language responses