Software Requirement Specifications

MujahidGPT

Submitted by
Mujahid Hussain (F22BINFT1M01235)

Submitted to
Mr. Muzamil ur Rehman

Department of Information Technology
Faculty of Computing
The Islamia University of Bahawalpur

Sr No	Details	Date	Supervisor Signature

Summary

MujahidGPT is a web-based intelligent personal assistant developed as a Final Year Project for the Department of Information Technology, The Islamia University of Bahawalpur, under the supervision of Mr. Muzamil ur Rehman. The project aims to design and implement an AI-powered chatbot that serves as a virtual representative for JHK Solution, providing instant, accurate, and consistent information to clients at any time.

The system is designed to address the common challenge of limited staff availability by offering an always-accessible platform through which users can inquire about the company's services, background, pricing, team, and contact details. By automating routine client interactions, MujahidGPT enhances customer satisfaction while reducing the operational workload on company staff.

MujahidGPT is implemented using Python for backend logic, HTML and CSS for user interface styling, and Gradio to deliver a responsive and user-friendly chat experience. The application is hosted on Hugging Face Spaces, with source code and version control managed through GitHub, and additional visibility provided via Google Sites. For natural language understanding and response generation, the system integrates the Groq API, ensuring fast and intelligent conversational responses.

The project follows a structured Software Requirements Specification (SRS) that clearly defines functional and nonfunctional requirements, system features, user classes, operating environment, constraints, and assumptions. Key functionalities include real-time conversational interaction, accurate retrieval of pre-defined company information, guided service inquiry handling when human support is unavailable, and basic feedback collection for continuous improvement.

Overall, MujahidGPT demonstrates the practical application of modern AI and web technologies to solve real-world business problems. The project delivers a cost-effective, scalable, and maintainable solution that meets both academic standards and industry-oriented requirements, showcasing the student's ability to analyze, design, and implement an intelligent software system in a professional manner.

Table of Contents

1. Introduction

1.1 Purpose

This document specifies the software requirements for the Personal Assistant for JHK Solution, named MujahidGPT. It is a chatbot system designed to provide comprehensive information about JHK Solution's services, company details, and assist clients in obtaining services when company personnel are unavailable. This SRS covers the full scope of the system, including its core chatbot functionality, integration with APIs, and user interaction mechanisms. The revision number is 1.0, representing the initial release.

1.2 Document Conventions

This SRS follows standard typographical conventions: Bold text is used for section headings, italics for emphasis on key terms, and bullet points for lists of functions or requirements. Priorities for requirements are explicitly stated where applicable (High, Medium, Low), and higher-level requirements are inherited by sub-requirements unless otherwise noted. All requirements are numbered sequentially (e.g., REQ-1). "TBD" is used as a placeholder for any undecided elements.

1.3 Intended Audience and Reading Suggestions

This document is intended for developers, project supervisors (e.g., Mr. Muzamil ur Rehman), clients from JHK Solution, testers, and documentation writers. Developers should focus on sections 3, 4, and 5 for implementation details. Supervisors and clients may start with sections 1 and 2 for an overview, then proceed to section 4 for features. Testers should review section 5 for nonfunctional requirements. Read in sequence from section 1 for a complete understanding, skipping appendices unless needed for definitions.

1.4 Product Scope

MujahidGPT is a web-based personal assistant chatbot that enhances client engagement for JHK Solution by providing 24/7 access to company information and services. Benefits include improved customer satisfaction through instant responses, reduced workload on human staff, and alignment with JHK Solution's goal of efficient service delivery. Objectives are to handle inquiries reliably, ensure cross-platform accessibility, and integrate seamlessly with existing tools. This SRS does not cover hardware procurement or post-deployment maintenance beyond initial setup. For vision details, refer to the project proposal document.

1.5 References

- Hugging Face Documentation: https://huggingface.co/docs (for model hosting and integration).

- Gradio Documentation: https://gradio.app/docs (for UI building).
- Google Sites Guide: https://support.google.com/sites (for hosting).
- GitHub API Reference: https://docs.github.com (for version control).
- Groq API Documentation: Available at thenajafigroup.online (for API keys and usage).
- Python 3.10+ Standard Library: https://docs.python.org/3/.
- HTML5 and CSS3 Standards: https://www.w3.org/TR/html5/ and https://www.w3.org/Style/CSS/.
- IEEE SRS Template: IEEE Std 830-1998 (adapted for this project).

2. Overall Description

2.1 Product Perspective

MujahidGPT is a new, self-contained product developed as a final year project to serve as a virtual assistant for JHK Solution. It builds on existing chatbot technologies but is customized for JHK Solution's specific needs, such as providing company history, service listings, pricing, and contact details. Unlike general chatbots, it focuses on business-specific queries and acts as a fallback when staff are unavailable. The system interfaces with web APIs for real-time data and is hosted on platforms like Hugging Face Spaces. A high-level diagram would show: Client (user) -> Web Interface (Gradio) -> Chatbot Logic (Python) -> API Calls (Groq) -> Response Generation.

2.2 Product Functions

The major functions include:

- Handling natural language queries about JHK Solution's services, history, team, and contact information.
- Providing instant responses using pre-trained models and API integrations.
- Collecting user feedback for improvements.
- Logging interactions for analysis by JHK Solution admins.
- Supporting multi-language queries if expanded (initially English). A top-level data flow: User input -> NLP processing -> Database/API query -> Response output.

2.3 User Classes and Characteristics

- Clients: Frequent users seeking services; may have low technical expertise, expect simple interactions; high importance.
- Company Admins: Occasional users for monitoring; technical background, access to logs; medium importance.
- Developers/Supervisors: Infrequent users for testing; high technical expertise; low priority for satisfaction but critical for maintenance. Users are differentiated by privilege levels: Clients (read-only), Admins (read/write).

2.4 Operating Environment

The software operates on web browsers across Mac, Linux, and Windows platforms. Hardware: Standard desktop/laptop with internet access (no specific CPU/RAM requirements beyond browser capabilities). Operating systems: Any supporting modern browsers (Chrome, Firefox, Safari). It coexists with other web apps without conflicts, hosted on Hugging Face Spaces and Google Sites.

2.5 Design and Implementation Constraints

- Languages: Python for backend logic, HTML/CSS for frontend styling.
- Tools: Gradio for UI, Hugging Face for model hosting, GitHub for source control, Google Sites for deployment.
- APIs: Must use Groq API keys from thenajafigroup.online; no other AI APIs without approval.
- Constraints: No mobile app version (web-only); adhere to free-tier limits of hosting platforms; ensure cross-browser compatibility; security protocols for API calls (HTTPS).
- Standards: Follow web accessibility guidelines (WCAG 2.1) for usability on all OS.

2.6 User Documentation

Delivered components: User manual (PDF), online help via in-chat prompts, and tutorials on GitHub README. Formats: Markdown for online, PDF for printable. Standards: Clear, step-by-step guides with screenshots.

2.7 Assumptions and Dependencies

Assumptions: Internet connectivity is always available for users; Groq API remains free/accessible; JHK Solution provides accurate company data for training. Dependencies: Groq API for NLP processing; Hugging Face/Gradio for hosting (project fails if platforms change terms); GitHub for version control. If assumptions change, requirements may need revision.

3. External Interface Requirements

3.1 User Interfaces

Logical characteristics: Web-based chat interface with text input box, response display area, and buttons for feedback (e.g., thumbs up/down). Follow Gradio UI standards: Simple layout, responsive design. Standard functions: Help button for FAQs, clear error messages (e.g., "Sorry, I didn't understand that"). Keyboard shortcuts: Enter to send message. Error display: Red text for invalid inputs. Details in separate UI spec document.

3.2 Hardware Interfaces

Supports standard web hardware: Keyboard/mouse for input, screen for output. No special devices; data interactions via HTTP. Supported types: Any device with browser (laptops, desktops). Protocols: Standard web protocols.

3.3 Software Interfaces

Connections: Python scripts interface with Gradio (version 3.x) for UI, Hugging Face libraries for models, Groq API (via HTTP requests). Data in/out: User queries (strings) in, responses (text) out. Services: API calls for NLP; shared data via session variables. Constraints: Use RESTful APIs; no global data areas.

3.4 Communications Interfaces

Requirements: Web browser for client-server communication; HTTPS for secure API calls. Message formatting: JSON for API responses. Standards: HTTP/HTTPS, no FTP. Security: API key encryption; transfer rates: <2s per response; synchronization: Real-time via websockets if implemented in Gradio.

---

4. System Features

Features organized by core services: Interaction, Retrieval, Handling, Feedback.

4.1 System Feature 1: Chatbot Interaction

4.1.1 Description and Priority

Interactive chat for user queries; High priority (benefit:9, penalty:9, cost:5, risk:3).

4.1.2 Stimulus/Response Sequences

User types query -> System processes -> Response displayed; Error if invalid -> Retry prompt.

4.1.3 Functional Requirements
REQ-1: System shall accept text inputs up to 500 characters.
REQ-2: System shall respond within 2 seconds using Groq API.

REQ-3: Handle errors gracefully with fallback messages.

4.2 System Feature 2: Information Retrieval

4.2.1 Description and Priority

Retrieve company info (services, contacts); High priority (benefit:8, penalty:7, cost:4, risk:2).

4.2.2 Stimulus/Response Sequences

Query like "What services does JHK offer?" -> Fetch data -> Display list.

4.2.3 Functional Requirements

REQ-4: Access pre-loaded company database.
REQ-5: Support keyword-based searches.

REQ-6: Update info via admin interface (TBD).

4.3 System Feature 3: Service Inquiry Handling

4.3.1 Description and Priority

Guide users on unavailable services; Medium priority (benefit:6, penalty:5, cost:3, risk:2).

4.3.2 Stimulus/Response Sequences

Inquiry when staff unavailable -> Provide alternatives or schedule.

4.3.3 Functional Requirements

REQ-7: Detect staff availability via API (TBD).

REQ-8: Suggest email/contact forms.

4.4 System Feature 4: User Feedback Collection

4.4.1 Description and Priority

Collect ratings/comments; Low priority (benefit:4, penalty:3, cost:2, risk:1).

4.4.2 Stimulus/Response Sequences

After response -> Prompt feedback -> Store.

4.4.3 Functional Requirements

REQ-9: Store feedback in log file.

REQ-10: Anonymize user data.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

System must handle 100 concurrent users with <2s response time. Rationale: Ensure fast access for clients. Specific: Real-time processing on standard hardware; individual features like retrieval <1s.

5.2 Safety Requirements

No direct harm risks, but prevent misinformation: Validate responses against company data. Refer to no external safety policies; no certifications needed. Actions: Log all interactions for audit.

5.3 Security Requirements

Protect API keys; use HTTPS. Authenticate admins via passwords. Refer to GDPR-like privacy; no certifications. User identity: Anonymous for clients.

5.4 Software Quality Attributes

Reliable (99% uptime), easy (intuitive UI), fast (low latency), accessible (cross-OS: Mac, Linux, Windows). Quantitative: Availability 99%, usability score >80% in tests. Preferences: Ease of use over learning curve. Other: Maintainable code, portable web app.

5.5 Business Rules

Admins can update info; clients read-only. Enforce: Only JHK-approved data displayed. Implies REQ-11: Role-based access.

6. Other Requirements

Database: Use lightweight JSON for storage. Internationalization: English only initially. Legal: Comply with data protection laws. Reuse: Leverage open-source Gradio components.

Appendix A: Glossary

- Chatbot: AI-driven conversational agent.
- Groq API: Fast AI inference service.
- Gradio: Python library for web UIs.
- NLP: Natural Language Processing.
- JHK Solution: The company this assistant serves.

Appendix B: To Be Determined List

1. Exact database schema for company info.
2. Multi-language support implementation.
3. Advanced analytics for logs.