**Software Requirements Specification (SRS) Document**

**F.R.I.D.A.Y AI Assistant**  
**Version:** 1.5  
**Prepared by:** [Your Name]  
**Organization:** [Your Organization]  
**Date:** March 12, 2025

#### 1. ****Introduction****

**1.1 Purpose**  
The purpose of this document is to outline the requirements for the development of **F.R.I.D.A.Y**, an advanced AI assistant designed to listen, think, understand, and execute tasks intelligently. This SRS document will serve as a guide for developers, testers, and stakeholders to understand the system's functionality, features, and design.

**1.2 Scope**  
F.R.I.D.A.Y is an AI-driven assistant that provides real-time information, automation, and AI-based responses. It is capable of handling various types of user queries, including general questions, real-time data requests, automation tasks, and image generation. The system is designed to operate with minimal user input, offering a seamless and interactive user experience.

**1.3 Definitions, Acronyms, and Abbreviations**

* **AI**: Artificial Intelligence
* **LLM**: Large Language Model
* **TTS**: Text-to-Speech
* **GUI**: Graphical User Interface
* **API**: Application Programming Interface
* **SRS**: Software Requirements Specification

**1.4 References**

* Folder Structure.pdf (provided document)
* Python Documentation
* PyQt5 Documentation
* Groq API Documentation
* Hugging Face API Documentation

**1.5 Overview**  
The SRS document is organized into sections that describe the overall system functionality, user interface, system requirements, and design constraints. The document also includes diagrams to illustrate the system architecture and workflow.

#### 2. ****Overall Description****

**2.1 Product Perspective**  
F.R.I.D.A.Y is a standalone AI assistant that integrates various modules to provide a comprehensive user experience. It operates on a desktop environment and interacts with users through a graphical interface and voice commands.

**2.2 Product Functions**

* **Voice Input Processing**: Captures and processes user voice input using speech recognition.
* **Query Classification**: Classifies user queries into real-time, general, automation, or image generation requests.
* **Task Execution**: Executes tasks such as web searches, application control, and image generation.
* **Response Generation**: Provides spoken and visual responses using TTS and GUI.
* **Automation**: Automates system tasks like opening/closing applications and controlling system settings.
* **Real-time Data Handling**: Fetches and processes real-time data such as weather, news, and stock prices.
* **Image Generation**: Generates AI-powered images based on user prompts.

**2.3 User Characteristics**  
The primary users of F.R.I.D.A.Y are individuals who require an intelligent assistant to handle daily tasks, provide real-time information, and automate repetitive actions. Users are expected to have basic knowledge of voice commands and desktop applications.

**2.4 Constraints**

* The system requires a stable internet connection for real-time data fetching and API calls.
* The system is designed to run on desktop environments with Python 3.8 or higher.
* The system depends on third-party APIs (e.g., Groq, Hugging Face) for certain functionalities.

**2.5 Assumptions and Dependencies**

* The system assumes that users will provide clear and concise voice commands.
* The system depends on external APIs for real-time data and image generation.
* The system assumes that the user's hardware supports voice input and output.

#### 3. ****Specific Requirements****

**3.1 Functional Requirements**

| **ID** | **Requirement** | **Description** |
| --- | --- | --- |
| FR-1 | Voice Input Capture | The system shall capture user voice input using a speech recognition module. |
| FR-2 | Query Classification | The system shall classify user queries into real-time, general, automation, or image generation requests. |
| FR-3 | Real-time Data Handling | The system shall fetch and process real-time data for queries requiring up-to-date information. |
| FR-4 | General Query Handling | The system shall provide AI-based responses for general queries using a language model. |
| FR-5 | Automation Task Execution | The system shall execute automation tasks such as opening/closing applications and controlling system settings. |
| FR-6 | Image Generation | The system shall generate AI-powered images based on user prompts. |
| FR-7 | Response Generation | The system shall provide spoken and visual responses using TTS and GUI. |
| FR-8 | Chat Logging | The system shall log all user interactions in a JSON file for future reference. |
| FR-9 | System Monitoring | The system shall display real-time system statistics such as CPU usage, RAM usage, and internet speed. |

**3.2 Non-Functional Requirements**

| **ID** | **Requirement** | **Description** |
| --- | --- | --- |
| NFR-1 | Performance | The system shall respond to user queries within 2 seconds. |
| NFR-2 | Availability | The system shall be available 24/7 with minimal downtime. |
| NFR-3 | Usability | The system shall provide an intuitive and user-friendly interface. |
| NFR-4 | Security | The system shall securely handle user data and API keys. |
| NFR-5 | Scalability | The system shall be scalable to handle multiple users and increased query loads. |

**3.3 System Requirements**

| **ID** | **Requirement** | **Description** |
| --- | --- | --- |
| SR-1 | Operating System | The system shall run on Windows, macOS, and Linux. |
| SR-2 | Hardware | The system requires a minimum of 4GB RAM and a dual-core processor. |
| SR-3 | Software | The system requires Python 3.8 or higher and the following libraries: PyQt5, Groq, Hugging Face, and Edge TTS. |
| SR-4 | Internet Connection | The system requires a stable internet connection for real-time data and API calls. |

#### 4. ****System Architecture****

**4.1 High-Level Architecture**  
The system is divided into three main layers:

* **Presentation Layer**: Handles the user interface and interaction.
* **Application Layer**: Manages the core functionality, including query processing, task execution, and response generation.
* **Data Layer**: Manages data storage and retrieval, including chat logs and system settings.

**4.2 Component Diagram**  
Below is a 2D diagram illustrating the high-level architecture of F.R.I.D.A.Y:

Copy

+-------------------+ +-------------------+ +-------------------+

| Presentation | | Application | | Data Layer |

| Layer | | Layer | | |

|-------------------| |-------------------| |-------------------|

| - GUI.py |<----->| - Main.py |<----->| - ChatLog.json |

| - User Interface | | - Module.py | | - System Settings |

| - Voice Input | | - Automation.py | | |

| - System Monitor | | - Chatbot.py | | |

| - Chat Log | | - ImageGeneration | | |

| | | - RealTimeSearch | | |

| | | - SpeechToText | | |

| | | - TextToSpeech | | |

+-------------------+ +-------------------+ +-------------------+

**4.3 Workflow Diagram**  
The workflow of F.R.I.D.A.Y can be represented as follows:

Copy

+-------------------+ +-------------------+ +-------------------+

| Voice Input | ----> | Query Analysis | ----> | Task Execution |

| (SpeechToText) | | (Module.py) | | (Automation.py) |

+-------------------+ +-------------------+ +-------------------+

| | |

| v v

| +-------------------+ +-------------------+

| | Real-time Data | | Image Generation |

| | (RealTimeSearch) | | (ImageGeneration) |

| +-------------------+ +-------------------+

|

v

+-------------------+

| Response |

| (TextToSpeech) |

+-------------------+

#### 5. ****Design Constraints****

* The system must be compatible with Python 3.8 or higher.
* The system must use third-party APIs for real-time data and image generation.
* The system must be designed to handle multiple user queries simultaneously.
* The system must be secure and protect user data and API keys.

#### 6. ****Diagrams****

**6.1 Class Diagram**  
The class diagram illustrates the relationships between the main classes in the system:

Copy

+-------------------+ +-------------------+ +-------------------+

| FridayUI | | FridayCore | | Module |

|-------------------| |-------------------| |-------------------|

| - initUI() |<----->| - initialize() |<----->| - classifyQuery() |

| - createMicBtn() | | - processCommand()| | - executeTask() |

| - sendMessage() | | - handleQuery() | | - generateImage() |

+-------------------+ +-------------------+ +-------------------+

| | |

| v v

| +-------------------+ +-------------------+

| | Automation | | Chatbot |

| |-------------------| |-------------------|

| | - openApp() | | - answerQuery() |

| | - closeApp() | | - logChat() |

| +-------------------+ +-------------------+

|

v

+-------------------+

| TextToSpeech |

|-------------------|

| - speak() |

+-------------------+

**6.2 Use Case Diagram**  
The use case diagram shows the interactions between the user and the system:

Copy

+-------------------+ +-------------------+ +-------------------+

| User | | F.R.I.D.A.Y | | System |

|-------------------| |-------------------| |-------------------|

| - Speak Command | ----> | - Process Query | ----> | - Execute Task |

| - View Response | | - Generate Image | | - Fetch Data |

| - Monitor System | | - Log Interaction | | - Control Apps |

+-------------------+ +-------------------+ +-------------------+

**6.3 Activity Diagram**  
The activity diagram illustrates the flow of activities in the system:

Copy

+-------------------+ +-------------------+ +-------------------+

| Start | ----> | Capture Voice | ----> | Classify Query |

+-------------------+ +-------------------+ +-------------------+

| | |

| v v

| +-------------------+ +-------------------+

| | Real-time Data | | Automation Task |

| | Fetch Data | | Execute Task |

| +-------------------+ +-------------------+

|

v

+-------------------+

| Generate Response|

| (TTS/GUI) |

+-------------------+

**6.4 ER Diagram**  
The ER diagram shows the relationships between the data entities in the system:

Copy

+-------------------+ +-------------------+ +-------------------+

| User | | ChatLog | | SystemSettings |

|-------------------| |-------------------| |-------------------|

| - UserID (PK) |<----->| - LogID (PK) |<----->| - SettingID (PK) |

| - Username | | - UserID (FK) | | - UserID (FK) |

| - VoiceInput | | - Query | | - MicStatus |

| - Response | | - Response | | - AssistantStatus |

+-------------------+ +-------------------+ +-------------------+

**6.5 DFD Diagram (3 Layers)**  
The Data Flow Diagram (DFD) shows the flow of data through the system across three layers:

**Layer 1: Presentation Layer**

Copy

+-------------------+ +-------------------+ +-------------------+

| User | ----> | GUI | ----> | Voice Input |

|-------------------| |-------------------| |-------------------|

| - Speak Command | | - Display Response| | - Capture Voice |

| - View Response | | - Show System Info| | - Send to Backend |

+-------------------+ +-------------------+ +-------------------+

**Layer 2: Application Layer**

Copy

+-------------------+ +-------------------+ +-------------------+

| Voice Input | ----> | Query Analysis | ----> | Task Execution |

|-------------------| |-------------------| |-------------------|

| - SpeechToText | | - Classify Query | | - Automation |

| - Send to Backend | | - Fetch Data | | - Generate Image |

+-------------------+ +-------------------+ +-------------------+

**Layer 3: Data Layer**

Copy

+-------------------+ +-------------------+ +-------------------+

| Task Execution | ----> | Data Storage | ----> | ChatLog |

|-------------------| |-------------------| |-------------------|

| - Execute Task | | - Store Data | | - Log Interaction |

| - Generate Image | | - Retrieve Data | | - Save Settings |

+-------------------+ +-------------------+ +-------------------+

#### 7. ****Future Enhancements****

* **Multi-language Support**: Add support for multiple languages.
* **Mobile Integration**: Develop a mobile version of F.R.I.D.A.Y.
* **Enhanced Automation**: Add more automation tasks, such as email management and calendar scheduling.
* **Machine Learning**: Integrate machine learning models for better query understanding and response generation.

#### 8. ****Conclusion****

This SRS document outlines the requirements for the development of F.R.I.D.A.Y, an advanced AI assistant. The document provides a detailed description of the system's functionality, architecture, and design constraints. The system is designed to offer a seamless and interactive user experience, with the ability to handle various types of user queries and tasks.

### 2D Diagrams

1. **High-Level Architecture Diagram**
2. **Class Diagram**
3. **Use Case Diagram**
4. **Activity Diagram**
5. **ER Diagram**
6. **DFD Diagram (3 Layers)**

This concludes the SRS document for F.R.I.D.A.Y.