Software Requirements Specification (SRS)

Introduction

The "Code Explainer" software system is an application that utilizes the Groq LLaMA 3 language model to provide explanations for Python code snippets. The system is designed to assist users in understanding complex code by generating human-readable explanations. The software shall be capable of processing Python code files, interacting with the Groq LLaMA 3 API, and rendering explanations in a user-friendly interface.

Overall Description

The Code Explainer software system consists of the following main components:

1. **User Interface (UI):** The UI is built using Streamlit, a Python library for creating web applications. It allows users to upload a code file, view the code content, and initiate the code explanation process.
2. **Groq LLaMA 3 API:** The system interacts with the Groq LLaMA 3 API to generate explanations for the uploaded code. The API is used to send code snippets to the LLaMA 3 model and retrieve explanation responses.
3. **OpenAI Client:** The system uses OpenAI's Python client library to interact with the Groq LLaMA 3 API.

Specific Requirements

1. **Performance:** The system shall respond to user input and generate explanations within 5 seconds (average response time).
2. **Security:** The system shall securely store and process user-uploaded code files, ensuring no

unauthorized access or data breaches.

3. **Design:** The UI shall be accessible and intuitive, with clear instructions and visual indicators for user interaction.

External Interface Requirements

1. **Groq LLaMA 3 API:** The system shall interact with the Groq LLaMA 3 API to send code snippets and retrieve explanation responses.
2. **OpenAI Client:** The system shall use OpenAI's Python client library to interact with the Groq LLaMA 3 API.

Functional Requirements

1. **Code Upload and Viewing:** The system shall allow users to upload Python code files (with supported extensions) and view the code content in the UI.
2. **Explain Code:** The system shall enable users to initiate the code explanation process by clicking a button.
3. **Generate Explanation:** The system shall use the Groq LLaMA 3 API to generate explanations for the uploaded code and display them in the UI.

Non-functional Requirements

1. **Scalability:** The system shall be designed to handle multiple users and simultaneous code explanation requests, ensuring efficient processing and minimal downtime.
2. **Usability:** The UI shall be user-friendly, with clear instructions, minimal errors, and informative feedback.
3. **Data Integrity:** The system shall ensure data integrity by securely storing and processing

user-uploaded code files, and maintaining accurate explanation responses.