Design Document

|  |  |  |
| --- | --- | --- |
| Version | Date | Description |
| 1.0 | 2023/3/9 | Initial draft of design document |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Project Overview

The goal of this project is to create a software application that accepts a GitHub username and retrieves information about the user's repositories, including all dependencies for each repository. The application will then create a JSON file of all dependencies, which will be visualized using Splunk.

# Architecture

The application will be built using Python and will consist of the following components:

User Interface: A simple web interface that accepts a GitHub username and initiates the data retrieval process.

Data Retrieval: A Python script that retrieves repository information and dependencies using the GitHub API.

Data Storage: A database or flat file that stores the retrieved repository and dependency data.

Data Visualization: A Splunk dashboard that visualizes the retrieved dependency data.

# Project Design

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

| GitHub API |

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

|

|

|

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

| Python Script |

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

|

|

|

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

| | |

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

| GitHub User | | Authentication | | Splunk |

| Interface | | Interface | | Visualization |

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

In this diagram, the GitHub API is shown as the external system that our Python script interacts with to retrieve data. The Python script serves as the main application logic that retrieves the data from the API, processes it, and exports it to Splunk for visualization.

The GitHub User Interface and Authentication Interface represent the components responsible for accepting user inputs (e.g., GitHub username, authentication credentials) and authenticating the user with GitHub API.

The Splunk Visualization component is responsible for displaying the data retrieved from GitHub in a meaningful and intuitive way to the user.

Overall, this diagram provides a high-level overview of the different components and their interactions in the system.

# Data Retrieval

The data retrieval component will use the following process to retrieve repository and dependency data:

Retrieve a list of repositories for the specified GitHub user using the GitHub API.

For each repository, retrieve the contents of the repository using the GitHub API.

For each file in the repository, check if the file is a dependency file (e.g., requirements.txt, package. Json, etc.). If the file is a dependency file, retrieve the contents of the file using the GitHub API and store the dependency information in a database or flat file.

GitHub User +------------>+ Python Script +----------> Splunk

In this diagram, we can see the flow of data between the three main components of the system: the GitHub User Interface, the Python Script, and the Splunk Visualization.

First, the user interacts with the GitHub User Interface, providing a GitHub username and authentication credentials. This data is then passed as input to the Python Script, which uses it to make API requests to retrieve the necessary data from GitHub. Once the Python Script has retrieved the data, it processes it and exports it to Splunk for visualization. The Splunk Visualization component then displays the data to the user in a meaningful way. Overall, this diagram provides a simplified overview of the data flow in the system, focusing on the interactions between the main components.

# Data Storage

The data storage component will use a database or flat file to store the retrieved repository and dependency data. The data schema for the database or flat file will include the following fields:

Repository Name

Repository URL

Dependency Name

Dependency Version

Dependency Type (e.g., Python, JavaScript, Ruby, etc.)

# Data Visualization

The data visualization component will use Splunk to create a dashboard that visualizes the retrieved dependency data. The dashboard will include the following visualizations:

A bar chart that shows the top 10 most common dependencies.

A table that shows all dependencies, sorted by repository.

A pie chart that shows the distribution of dependencies by type.

# Security

The application will use secure authentication mechanisms to protect user data and prevent unauthorized access to the application. Authentication will be required to access the data retrieval component and the data storage component.

# Conclusion

This project will create a software application that retrieves dependency data for GitHub repositories and visualizes the data using Splunk. The application will be secure, scalable, and user-friendly, providing valuable insights into the dependencies used in GitHub projects.