**User Requirements:**

* Given a repository and a GitHub username, the site shall return several statistics and visualizations related to the user's contribution to the repo.
* Given a repo file path, the system shall display the change information, such as the date/time and username of who made the last modification, for every line in the given file.

**System Requirements:**

* The system should accept a keyword that the user input.
* The system should provide historical modification information for a given repo file path.
* The system should provide a project's information for a user's search results.
* Given a username and repository, the system will validate and make API calls based on the given inputs
* The system will modify the information of the API (remove fields that are not necessary).
* The system should display the information that the API returns on the screen in a user-friendly and easy-to-navigate manner.
* The system shall be easily understandable, portable, upgradeable, user-friendly, and responsive.
* The system should be scalable by updating or modifying information based on user requirements.

**Risk Analysis:**

* GitHub API rate limiting
* Not enough knowledge of React.js
* Limited knowledge of REST and GraphQL API calls
* GitHub API may not have services that the GitHubGo website needs
* API stops communicating; the GitHubGo website heavily relies on GitHub API, so if GitHub API is down, it will also take down the GitHubGo website’s services
* Poor network connection; the site might not respond or take longer than usual to load results based on the user's internet connection