LLM Insights
Features Document

Feature 1: SSO Login – User

Flow of Events:

- Users access the application.
- They are presented with an SSO login page.
- Users must log in through GitHub's SSO.
- The system verifies grants access if they succeed.
- Upon successful login, users are redirected to their respective profiles or dashboards.

Acceptance Criteria:

- Users can view their account information.
- The system securely stores user information.
- Login requires valid SSO credentials.
- Successful login redirects users to the appropriate interface.

Feature 2: Dashboard for all User Repositories

Flow of Events:

- As users log in to their accounts.
- They access their GitHub's Repositories.
- Users can remove repositories from their respective dashboards.

Acceptance Criteria:

- The dashboard displays relevant repos information for the user.
- Users can manage their profiles and settings.

Feature 3: Dashboard for each Repo.

Flow of Events:

- Users log in to their accounts.
- They access the dashboard for each repo.
- The dashboard displays Repo analysis, clones, commits and statistics.
- Users can check the LLM based insights offered by the application for that repo (new section in that dashboard).

Acceptance Criteria:

- Users can view repo analysis, clones, commits and statistics.
- Types of LLM based insights can be accessed.
- Open PR's clones, contributions, branches, etc

Feature 4: Dashboard for insights

Flow of Events:

- Users access the application.
- The app must be securely connected with the LLM for seamless transactions.
- The app determines the preferred insights from the list provided.
- The system checks for required information from the repositories.

Acceptance Criteria:

- The application successfully generates insights using any LLM model .

LLM Insights for the application

Insight 1: Collaboration Analysis

Description:

As a user, I want to identify active contributors and their impact on the repository. This feature will analyze collaboration patterns, such as who works together frequently on issues and pull requests.

Acceptance Criteria:

- The user will select the repo to use.
- The user will select the collaboration analysis tab to know which developers when working together improve the code quality/smells of the project.

Insight 2 Common code mistakes

Description:

As a user, I want to see the common types of mistakes occurring in the code by using the code review comments.

Acceptance Criteria:

- I will choose a project of my choice.
- The search response will provide me with common code mistakes with the help of code-review comments.

Insight 3: Bug Detection in Application Flow

Description:

As a user, I want to employ a Language Model (LLM) to analyze any application's major flows to identify potential issues or gaps. This feature is different from traditional code review. Here, I want to consider code pushed by the developer in any open PR that can introduce unexpected bugs in the application's major flows.

Acceptance Criteria:

- For any repo of interest, there will be a list of open PR(s).
- A task runner/workflow will run, and it will automatically conduct bug detection to find unexpected bugs being introduced by pushed code in the application flows.
- I want to display actionable recommendations for resolving these bugs on the front end.
- Also, I want to display alerts if any PR is introducing any bug in the application's major flows.

Insight 4: Code Quality Enhancement Insights

Description:

As a user, I want to leverage a Language Model (LLM) to suggest code improvements based on different criteria like Readability, Performance, Correctness, Scalability, etc. on open Pull Requests (PRs) by a developer, so that we can ensure code quality, adherence to coding standards, and catch potential issues early in the development process.

Acceptance Criteria:

- Against any repo of interest, there will be a list of PR(s) opened by the developer(s).
- I want to view a chart displaying code improvement suggestions for the whole list of open PR(s) by selecting the repository's name.
- Also, I want to view the code improvement suggestions chart for open PR(s) by specifying both the repository name and the developer's name.
- The chart will display scores across the vertical axis (0 to 5) and improvement criteria like readability, performance, etc along the horizontal axis.