CodeCompass Test Plan and Report
By: Mann, Vincent, Evan, Nickolas, Stephen, Hao

System Test scenarios: Provide a list of system test scenarios. Ideally, the scenarios should relate to specific user stories and associated acceptance criteria.

A scenario is a list of system level interactions (including precise input and output) a user would follow to determine that each user story has been completed; i.e., all the acceptance criteria for a user story are met.

Example:
A. User story 1: As a user I want to create an account so that I can use the toilet location system.
B. User story 2: As a registered user I want to view a map showing locations of all toilets in the 5 mile radius of my current location so that I can visually choose more information about toilets of interest.

Scenario 1: Create Account (Pass/Fail)
1. start Toilet app; select ‘new user’; type
    name = <Linda Werner>
    password = <LLWPass9>
    password confirmation = <LLWPass9>
    press Enter Key
2. user should see verification message that account is now active

Scenario 2: Show toilets in 5 mile radius (Pass/Fail)
1. select ‘view toilets in 5 mile radius’;
2. User should see map of 5 miles radius of current location with all toilets within the toilet database marked.

Note for each scenario whether the test passed or failed for the system release version.

Unit tests: If you developed automated tests, reference the relevant directories in your release branch. Note if any of the tests failed for the released system version. Be prepared to run the tests during the project review.

—--

User Story 1: As a user, I want to be able to upload codebase from both Public & Private Github repos and Local Folders (Pass)
Scenario 1: Authorize CodeCompass on GitHub

1. If user has not installed GitHub App, click ‘Install GitHub App’
2. Click "Connect GitHub Account" button
3. User should be redirected to GitHub OAuth authorization page
4. Click "Authorize CodeCompass" on GitHub

Scenario 2: Local Folders

1. Start Organization page; select ‘New repository’;
2. Type name = <repo name>
3. Choose local repository type
4. Upload a .zip file of the local repository
5. Press create repository
6. User should see a codebase on your organization page

Scenario 3: GitHub/GitLabFolders

1. Start Organization page; select ‘New repository’;
2. type name = <repo name>
3. Choose GitHub/Gitlab repository type
4. Type link = <repository GitHub/GitLab link>
5. Press create repository
6. User should see a codebase on your organization page

User Story 2: As a user, I want to be able to upload multiple codebases (i.e., Multiple projects) (Pass)

1. start Organization page; (with one codebase already inputted); select ‘New repository’;
2. Fill in required fields and upload the codebase
3. User should see multiple codebases in the organization page
4. User should click through each codebase being able to swap in the organization page

User Story 3 : As a user, I want automatic AI-generated documentation for the codebase

1. start Organization page; (with one codebase already inputted); select ‘New repository’;
2. Fill in required fields and upload the codebase (local folder, GitHub, GitLab, etc.)
3. Click on project
4. Click ‘Index Codebase’
5. Documentation is displayed when processing is complete

User Story 4: As a user, I want to have an AI-powered RAG-Chat Interface to interact with the codebase

1. Start Project page

2. Fill in required fields and upload the codebase (local folder, GitHub, GitLab, etc.)
3. Click on project
4. Documentation is displayed when processing is complete

User Story 5: As a user, I want to have an A RAG retrieval engine retrieve relevant code chunks to the user query from the input codebase

1. start Organization page; (with one codebase already inputted); select ‘New repository’;
2. Fill in required fields and upload the codebase
3. Wait for indexing to finish
4. Open chat and ask desired queries
5. Observe right-hand side: code chunks displayed with line numbers
    a. Left-hand side contains LLM response