

Scoping Document

Group 2: Sean Stapleton, Yash Gambhir, Eric Guldán, Anish Ravipati, David Sun, Jonah Dykhuizen

Github Assistant

The user should be able to learn more about a Github repository and make and respond to pull requests using the assistant. This will be accomplished using three high-level competencies that will be broken down into lower-level competencies. The high-level competencies are the following:

1. Interact with Pull Requests and Issues
2. Request information about the repository
3. Static Code Analysis

Please see below for a full breakdown of these competencies.

1. Interact with Pull Requests and Issues

The user should be able to find, view, update, and create pull requests and issues. Depending on the given state, they will need to provide information such as source and target branches, associated people, and request type. This high-level competency will encompass the following states:

[State Descriptions \(https://www.notion.so/cd248bedaae4e09ae4cf3fa82197ab7\)](https://www.notion.so/cd248bedaae4e09ae4cf3fa82197ab7)

Important States

Search for a pull request or issue

State Name: search_requests

State Type: information

Slot names:

- request_type
- creator
- reviewer
- tag
- source_branch
- target_branch
- filter
- sort

Example utterances:

- Can you show me all of my PRs?
- Can you show me all the PRs I need to review?
- What are the newest issues?
- Has anyone tagged me in an issue?

- Show me the PR for feature1?
- What's David's PR looking like?
- Do I need to look into anything today?
- What are the 3 most recent issues?
- Are there any PRs to merge into master today?

Competency Blueprint

[Transitions \(https://www.notion.so/b7fe45c495e949cca85f8fd048af0cee\)](https://www.notion.so/b7fe45c495e949cca85f8fd048af0cee)

Transition Utterances

- Let me see the top one
- Can you open Yash's second PR there?
- Lemme see what people are saying about issue #1
- Dive into the second one

Business Logic

Generated Slots

- repository
- relevant_requests

Scenarios

- No repository provided
 - The assistant will make a request to the front-end client via the chrome extension to determine the current repository. If this can't be determined, it will attempt to extract the repository name from previous conversations. Otherwise, it will use the repository to which the user has most recently contributed.
- No slots provided
 - The assistant will find all pull requests and issue that are connected to the user and store them in relevant_requests in order of recency
- Sort and/or Filter are provided
 - The assistant will find all pull requests and issue that are connected to the user and store them in relevant_requests in the sort method provided. The assistant will also filter the results based on the provided filter.
- Request type provided
 - The assistant will only include requests of that type in the relevant_requests
- Anything else provided
 - The assistant will again find all pull requests and issues connected to the user. However, it will use the provided information to filter what counts as "connected."

Response Scenarios

- No slots provided
 - "Here are the most recent pull requests and issues I think you might be interested

in for the {{ repository }} repository. Would you like to view any of these in more detail?"

- Request type provided
 - "Here are the most recent {{ request_type }}s I think you might be interested in for the {{ repository }} repository. Would you like to view any of these in more detail?"
- Filter provided
 - "Ok, I've found the {{ filter }} results based on your search. Do you want to dive into any of these?"
- Sort provided
 - "I've sorted the most recent {{ request_types }} associated with your account. Do you want to look at any of these closer?"
- Target provided
 - "Here are all of the pull requests to {{ target }}. Any of these seem interesting?"
- All responses
 - Include {{ relevant requests }} as a list

Create a Pull Request

State Name: create_pr

State Type: action

Slot names:

- source_branch
- target_branch
- pr_message
- reviewer
- assignee

Example utterances:

- Can you create a PR from the feature1 branch to master branch?
- Yo I just finished working on the feature2 branch. Can you create a PR to close out the task?
- Hey can you close out my feature with a PR to master? Say "Added feature X to the landing page" and tag Joe as a reviewer.
- I'm done with feature1. Let's ship the PR.

Business Logic

Generated Slots

- repository
- source_branch

Scenarios

- No repository provided
 - The assistant will make a request to the front-end client via the chrome extension to determine the current repository. If this can't be determined, it will attempt to extract the repository name from previous conversations. Otherwise, it will use the repository to which the user has most recently contributed.
- No source branch provided
 - The assistant will make a request to the Github API to determine the most recently committed branch on that repository by the user. This will fill in the `source_branch`.

Response Scenarios

- All scenarios
 - "Hi! I've created a pull request for the {{ repository }} repository from the {{ source_branch }} branch to the {{ target_branch }} branch. Here is a preview. Would you like to submit?"

2. Request information about the repository

The user should be able to request information about a repository, its contributors, or specific files in order to acclimate themselves to the project. They will need to provide the type of information they are looking for (person, statistics, commit info, etc). This competency will include the following states:

[States \(https://www.notion.so/3d387312420045ab8b290b1882df2f7c\)](https://www.notion.so/3d387312420045ab8b290b1882df2f7c)

Important States

Determine which files are being worked on the most

State name: `feature_frequency_ranking`

State type: information

Slot names:

- `sorting_method`
- `time_period`

Example Utterances:

- Hey what features are being worked on the most?
- I want to contribute to this repo, but I'm not sure which features would be good to work on, which files don't have a lot of people working on them?
- What is the most worked on feature in the last month?

[Transitions \(https://www.notion.so/ff173d3d090f493aa6b23b2508b239f2\)](https://www.notion.so/ff173d3d090f493aa6b23b2508b239f2)

Transition Utterances:

- Who knows the most about this feature?

- How many issues are there about this feature?
- Which feature do you think I should work on?

Business Logic

Generated Slots

- repository
- relevant_features

Scenarios

- No repository provided
 - The assistant will make a request to the front-end client via the chrome extension to determine the current repository. If this can't be determined, it will attempt to extract the repository name from previous conversations. Otherwise, it will user for the repository name.
- Once the repository is detected, the assistant will leverage the Github API to determine the most relevant features

Response Scenarios

- No repository specified
 - "Sure! Which repository do you want to know more information about?"
- All other scenarios
 - "Here are the files that have been {{ sorting_method }} actively worked on in the last {{ time_period }}. Is there anything you want to learn more about, specifically?"
- All Scenarios
 - Include {{ relevant requests }} as a list

Find the most active contributor in some specific way (e.g. most active commenter)

State name: most_active

State type: information

Slot names:

- activity
- sorting_method
- time_period

Example Utterances:

- Who has closed the most issues in the last month?
- Who has closed the least issues?
- I don't know who to set as my reviewer for my PR. Who usually reviews pull requests?
- I'm looking to talk to our most insightful developers. Who's submitted the most issues?
- I just submitted an issue but no one has answered. Who should I talk to?

Transitions

n/a

Business Logic

Generated Slots

- repository
- relevant_contributors

Scenarios

- No repository provided
 - The assistant will make a request to the front-end client via the chrome extension to determine the current repository. If this can't be determined, it will attempt to extract the repository name from previous conversations. Otherwise, it will user for the repository name.
- Once the repository is detected, the assistant will leverage the Github API to determine the most relevant contributors

Response Scenarios

- No slots provided
 - "Sure, which activity you would like to track?"
 - Suggestions (pill buttons):
 - Opened issues
 - Closed issues
 - PR reviews
 - PR submissions
 - Comments
- Activity provided
 - "Sure! Here are the contributors who {{ activity }}d the {{ sorting_method }} in the last {{ time period }}."

3. Static Code Analysis

The user should be able to request information about code – its function, documentation, framework, design pattern, best practices, etc. The user might want this information because they are unfamiliar with the codebase and want to learn more without having to reach out to its many contributors or scour through documentation. The user will need to provide the type of analysis they are looking for. This competency will include the following states:

[States \(https://www.notion.so/b11bb040f6764948a301c642f589a50f\)](https://www.notion.so/b11bb040f6764948a301c642f589a50f)

Important States

Summarize code functionality

State name: code_summarization

State type: information

Slot names:

- filename
- function_name
- line number

Example Utterances:

- What does the update_state_parameters function do?
- Hey I'm having trouble understanding. What does the api call on line 21 do?
- What does this method in utilities.py do?
- What's with all this recursion on line 42?
- The get_stuff method is pretty confusing, man. Do you know what it's supposed to do?

[Transitions \(https://www.notion.so/f9aa82f4546948669d472c21250e1f26\)](https://www.notion.so/f9aa82f4546948669d472c21250e1f26)

Transition Utterances:

- Ah ok, so what does parameter1 do then?
- So the client variable must be a reference to the chrome driver. Where should that be defined?

Business Logic

Generated Slots

- repository
- analyser_response

Scenarios

- No repository provided
 - The assistant will make a request to the front-end client via the chrome extension to determine the current repository. If this can't be determined, it will attempt to extract the repository name from previous conversations. Otherwise, it will user for the repository name.
- All necessary information provided
 - The assistant will produce a string representing the code located using the provided information. It will then pass that string to the static analysis API and set analyser_response to the response returned by the API.

Response Scenarios

- No repository specified
 - "Sure! Which repository do you want to know more information about?"
- Function name provided, but can't determine filename
 - "Sweet, I'll take a look at {{ function_name }}. Which file is that in?"

- Filename provided but no function name or line number
 - "Is there a specific method or line number in {{ filename }} you would like to learn more about?"
- Line number and filename provided
 - "Great, give me one second while I take a look at that line"
 - {{ analyser_response }}
- Function name and filename provided
 - "Great, give me one second while I take a look at the code"
 - {{ analyser_response }}

Identify framework used in code

State name: identify_framework

State type: information

Slot names:

- filename

Example Utterances:

- Hey what framework does this repository use
- What frameworks are being used in train_model.py?
- Can you list the tools used in this repo?

[Transitions \(https://www.notion.so/958edc3316d041ee9769025b56406e15\)](https://www.notion.so/958edc3316d041ee9769025b56406e15)

Transition Utterances:

- Where can I learn more about these frameworks?
- I want to learn {{ framework }}!

Business Logic

Generated Slots

- repository
- framework

Scenarios

- No repository provided
 - The assistant will make a request to the front-end client via the chrome extension to determine the current repository. If this can't be determined, it will attempt to extract the repository name from previous conversations. Otherwise, it will user for the repository name.
- No filename provided
 - Assistant will summarize tools and frameworks used in the whole repository

Response Scenarios

- No repository specified
 - "Sure! Which repository do you want to know more information about?"
- No Filename provided
 - "Here are the frameworks used in the whole repo: "
- No framework in context
 - "I'm sorry, I wasn't able to determine what framework is being used."

Identify company/project standards for common tasks

State name: indentify_standards

State type: information

Slot names:

- action

Example Utterances:

- Hey I want to make an http request. What library should I use?
- Yo I want to create some unit tests for the React components. Is there a standard library for unit testing?
- What should I use for documenting my code?
- Need to encrypt and salt some passwords. What should I use?

Note*: since it would be extremely inefficient to generate a representation of every dependency and match them to the passed action, we will probably need to compile a list of actions we will support. This should be done by categories (web dev, data manipulation, etc).

[Transitions \(https://www.notion.so/bd002b2c3f614b75ae2cd5eb061361af\)](https://www.notion.so/bd002b2c3f614b75ae2cd5eb061361af)

Transition Utterances:

- Ok sweet, where can I learn more about that?
- Sick, how do I use that?

Business Logic

Generated Slots

- repository
- standards

Scenarios

- No repository provided
 - The assistant will make a request to the front-end client via the chrome extension to determine the current repository. If this can't be determined, it will attempt to extract the repository name from previous conversations. Otherwise, it will user for the repository name.

- No action provided
 - Make a call to the static analyser API and set standards to the response – the overall most frequently used dependencies in the project. Then, prompt the user for their action.
- Action provided
 - Make a request to the static analyser API and set standards equal to the returned response

Response Scenarios

- No repository specified
 - "Sure! Which repository do you want to know more information about?"
- No action specified
 - "Of course. Here are some of the most frequently used dependencies in this project. What are you trying to do?"
 - Display {{ standards }}
- Action specified, one standard found
 - "Looks like the project standard is the {{ standards[0] }} library"
- Action specified, multiple standards found
 - "Looks like there are a few dependencies that have been used for {{ action }}. Here they are in order of most frequently used."
 - Display {{ standards }}