Use case diagrams

<https://miro.com/welcomeonboard/elBQYnVCdElqQ2lwd2NaN09YZk9NTzE0bTh1bDBCQmp6NnVIQ2U4eEszYkZkRVFiWXdwM1VEd0tXanFCZ0ZtdHwzNDU4NzY0NjAwNTY0MDI5MTk1fDI=?share_link_id=698062456943>

| Explain code: UC-000. | While at the home screen, the user inputs a code snippet. This creates a new thread, sends the snippet to the ChatGPT API and returns an explanation to the user. |
| --- | --- |
| Login: UC-001. | Login to the user account |
| Logout: UC-002. | Logout a user account |
| Open Scenario: UC-003. | Open a Scenario from the list of available Scenarios |
| Find Scenario: UC-004. | The system finds a Scenario based on search conditions |
| Create Scenario: UC-005. | This allows a user to create and save custom Scenarios to their user account |
| Manage Scenario: UC-006. | This opens the scenario profile for the user, it views and edits the profile details |
| Adjust Explanation Scope: UC-007. | This allows the user to change how much code is being analyzed when <Create Explanation> is run |
| Receive Feedback on Code: UC-008 | This allows the user to receive feedback on the code they have entered, once the user has entered code the application will have a button where the user can choose to either receive feedback on errors in code or an explanation as to how the code functions; the user will have the option to choose the degree in specificity of the feedback. |
| Get Syntax Errors: UC-009 | This allows the user to get immediate explanations for syntax errors in their code. The system analyzes the code, highlights syntax errors, and provides suggestions for fixing them. The user can correct the errors based on the system feedback and recheck the code for successful execution. |
| Enter Language: UC-0010 | This allows the user to change the desired language the application uses. |
| Delete Account: UC-0011 | This allows the user to delete their account they’ve created. |
| Create Account: UC-0012 | This allows the user to create an account on the NoCom application. |

| Name/Title: | Explain code |
| --- | --- |
| **ID:** | UC-000 |
| **Description**: | While at the home screen, the user inputs a code snippet. This creates a new thread, sends the snippet to the ChatGPT API, and returns an explanation to the user. |
| **System Under Design:** | NoCom |
| **Primary Actor**: | User |
| **Participants**: | NoCom servers, ChatGPT API |
| **Goal**: | This is NoCom's core function. Send the code snippet to NoCom servers and send a request to ChatGPT API, which will return a description for that code. This will allow the user to receive insight and helpful applications on obscure code snippets. |
| **Related Use Cases**: | **Extensions:**  UC-009 <Get Syntax Errors>, <Adjust Explanation Scope>, <Copy Out>, UC-0010 <Enter Language Code>, UC-008 <Receive Feedback>, UC-003 <Open Scenario> |
| **Invariant**: | The code must be in the proper language |
| **Precondition**: | User enters code |
| **Success Postcondition**: | A new conversation thread will be created, and the user will be presented with an explanation for the given code generated by ChatGPT API. If comments do not already exist, they will be added to the code snippet. |

| **USER STEPS**:  1. The user inputs a code snippet into NoCom  5. The user reads the generated explanation and the returned code snippet with additional comments | **SYSTEM RESPONSE**:  2. Create a new conversation thread  3. Prompt ChatGPT to generate an explanation for the provided code, to a variable level of detail  4. Add comments to the code snippet if there are none |
| --- | --- |

| Name/Title: | Login |
| --- | --- |
| ID: | UC-001 |
| **Description**: | Login to the user account |
| **System Under Design:** | NoCom |
| **Primary Actor**: | Guest |
| **Participants**: | NoCom servers |
| **Goal**: | Change the guest session to a user session if they input valid credentials. |
| **Related Use Cases**: | **Extensions:**  UC-0012 <Create Account> |
| **Precondition**: | Guest session |
| **Success Postcondition**: | The Guest is logged in to a user account |
| **USER STEPS**:  1. The user clicks on the login button.  3. User inputs login credentials. | **SYSTEM RESPONSE**:  2. The system opens the login menu.  4. System matches credentials against an account database.  5. If matches log the guest into the user account session and closes the guest session. |

| Name/Title: | Logout |
| --- | --- |
| ID: | UC-002 |
| **Description**: | Logout a user account. |
| **System Under Design:** | NoCom |
| **Primary Actor**: | User |
| **Participants**: | NoCom servers |
| **Goal**: | Exit a user session into a guest session. |
| **Related Use Cases**: |  |
| **Precondition**: | User session |
| **Success Postcondition**: | The user session is closed and a guest session is open. |
| **USER STEPS**:  1. The user clicks the logout button. | **SYSTEM RESPONSE**  2. The system closes the user account session.  3. The system opens a guest account session. |

| Name/Title: | Receive Feedback on Code |
| --- | --- |
| ID: | UC-008 |
| **Description**: | This allows the user to receive feedback on the code they have entered. Once the user has entered the code, the application will have a button where the user can choose to either receive feedback on errors in the code or an explanation as to how the code functions. The user will have the option to choose the degree of specificity of the feedback. |
| **System Under Design:** | NoCom |
| **Primary Actor**: | User |
| **Participants**: | NoCom servers, ChatGPT API |
| **Goal**: | Provide a varying degree of feedback on code entered by the user, including errors in code and an explanation of how the code works. |
| **Related Use Cases**: | **Instance of:**  UC-003 <Open Scenario>  UC-000 <Explain Code> |
| **Precondition**: | User enters code |
| **Success Postcondition**: | The system shall present a button to report errors related to the code entered by the user. |

| **USER STEPS**:  1. User enters their code.  3. User chooses to show syntax errors in code.  5. User looks at syntax errors or asks the system to give a higher or lower degree of feedback. | **SYSTEM RESPONSE**:  2. Shows button with options for desired level of feedback.  4. Displays syntax errors in code.  4.1 Displays other options for feedback.  6. Explains errors in more or less detail using generative AI. |
| --- | --- |

| Name/Title: | Get Syntax Errors |
| --- | --- |
| **ID:** | UC-009 |
| **Description**: | This allows the user to get immediate explanations for syntax errors in their code. The system analyzes the code, highlights syntax errors, and provides suggestions for fixing them. The user can correct the mistakes based on the system feedback and recheck the code for successful execution. |
| **System Under Design:** | NoCom Platform |
| **Primary Actor**: |  |
| **Participants**: | None |
| **Goal**: | This use case aims to help users understand and correct syntax errors in their code quickly and efficiently. |
| **Related Use Cases**: | **Instance of:**  UC-000 <Explain Code> |
| **Invariant**: | The user must log in and input a code snippet with potential syntax errors. |
| **Precondition**: | The user must be logged into the system.  The user has entered a code snippet containing one or more syntax errors. |
| **Success Postcondition**: | The system identifies and explains syntax errors.  The user corrects the errors, and the system verifies the code as error-free. |

| **USER STEPS**:  1. The user inputs a code snippet with syntax errors.  3. The user views the highlighted errors.  5. The user corrects the syntax errors.  7.The user submits the corrected code. | **SYSTEM RESPONSE**:  2. The system analyzes the code for syntax errors.  4.The system provides explanations and suggestions.  6. The system re-analyzes the code after correction.  8.The system confirms the code is error-free or gives further feedback. |
| --- | --- |

| Name/Title: | Adjust Explanation Scope |
| --- | --- |
| ID: | UC-007 |
| **Description**: | This allows the user to change how much code is being analyzed when <Create Explanation> is run. |
| **System Under Design:** | NoCom |
| **Primary Actor**: | Guest |
| **Participants**: | NoCom servers |
| **Goal**: | Limit the amount of code being fed into <Create Explanation>. |
| **Related Use Cases**: | **Instance of:**  UC-000 <Explain Code> |
| **Precondition**: |  |
| **Success Postcondition**: | New <create explanation> output. |
| **USER STEPS**:  1. The user enters a new scope. | **SYSTEM RESPONSE**:  2. The system enters scoped code into <Create Explanation>.  3. System outputs a new explanation. |

| Name/Title: | Open Scenario |
| --- | --- |
| ID: | UC-003 |
| **Description**: | Open a Scenario from the list of available Scenarios. |
| **System Under Design:** | NoCom |
| **Primary Actor**: | Guest |
| **Participants**: | NoCom servers |
| **Goal**: | To open a scenario session based on the actor's selected scenario. |
| **Related Use Cases**: | **Instance of:**  UC-000 <Explain Code>  **Inclusions:**  UC-004 <Find Scenario>  **Extensions:**  UC-008 <Receive Feedback> |
| **Precondition**: | To already have a scenario made. |
| **Success Postcondition**: | A scenario session based on the selection is open. |
| **USER STEPS**:  1. The user clicks on a scenario | **SYSTEM RESPONSE**:  2. The system opens a scenario session based on selection |

| Name/Title: | Find Scenario |
| --- | --- |
| ID: | UC-004 |
| **Description**: | The system finds a Scenario based on search conditions. |
| **System Under Design:** | NoCom |
| **Primary Actor**: | Guest |
| **Participants**: | NoCom servers, ChatGPT API |
| **Goal**: | To find and open a matching Scenario. |
| **Related Use Cases**: | **Included by:**  UC-003 <Open Scenario>  **Inclusions:**  UC-006 <Manage Scenario> |
| **Precondition**: |  |
| **Success Postcondition**: | A list of scenario entries |
| **USER STEPS**:  1. The user enters the search parameters.  5. User <Opens Scenario>. | **SYSTEM RESPONSE**:  2. The system finds matching Scenarios based on parameters.  3. The System organizes a list based on the best results.  4. The System Returns the list to the User. |

| Name/Title: | Create Scenario |
| --- | --- |
| ID: | UC-005 |
| **Description**: | This allows users to create and save custom scenarios to their accounts. |
| **System Under Design:** | NoCom |
| **Primary Actor**: | User |
| **Participants**: | NoCom servers |
| **Goal**: | To Create and save a custom Scenario to a user account. |
| **Related Use Cases**: | **Extensions:**  UC-006 <Manage Scenario> |
| **Precondition**: |  |
| **Success Postcondition**: | The creation of a Scenario profile linked to the user account. |
| **USER STEPS**:  1. The user clicks the Create Scenario button.  4. User adds scenarioScero details to profile.  5. User clicks save. | **SYSTEM RESPONSE**:  2. System generates a new scenario profile.  3. System links profile to user account.  6. The system saves the change to the profile. |

| Name/Title: | Manage Scenario |
| --- | --- |
| ID: | UC-006 |
| **Description**: | This opens the scenario profile for the user; it views and edits the profile details. |
| **System Under Design:** | NoCom |
| **Primary Actor**: | User |
| **Participants**: | NoCom servers |
| **Goal**: | To open a selected scenario profile for the user to edit. |
| **Related Use Cases**: | **Instance of:**  UC-005 <Create Scenario>  **Included by:**  UC-004 <Find Scenario> |
| **Precondition**: | Custom Scenario |
| **Success Postcondition**: | A Scenario profile was updated |
| **USER STEPS**:  1. The user opens the Scenario list.  2.User <Find Scenario> based on Scenarios made in <Create Scenario>.  3. User Hits Manage Scenario.  5. User edit profile.  6. User hit save changes. | **SYSTEM RESPONSE**:  4. System Opens Scenario profile.  7. System Saves the changes to the profile. |

| Name/Title: | Enter Code Language |
| --- | --- |
| ID: | UC-010 |
| **Description**: | This allows the user to change the desired language the application uses. |
| **System Under Design:** | NoCom |
| **Primary Actor**: | User |
| **Participants**: | NoCom servers |
| **Goal**: | The goal is to allow the user to change the desired language the application uses. |
| **Related Use Cases**: | **Instance of:**  UC-000 <Explain Code> |
| **Precondition**: |  |
| **Success Postcondition**: | The language the application uses shall change to what the user specifies. |
| **USER STEPS**:  1. The user will click the ‘settings’ button.  3. The user will click the ‘language’ option in settings.  5. The user chooses the specified language. | **SYSTEM RESPONSE**:  2. System shall display users settings.  4. The system shall display the preset list of languages for the user to choose from.  6. The system shall display text in application in the language chosen by the user. |

| Name/Title: | Delete Account |
| --- | --- |
| ID: | UC-011 |
| **Description**: | This allows the user to delete their account they’ve created. |
| **System Under Design:** | NoCom |
| **Primary Actor**: | User |
| **Participants**: | NoCom servers |
| **Goal**: | Shall allow the user to delete their account they have created on the application. |
| **Related Use Cases**: |  |
| **Precondition**: | User must be logged into the application. |
| **Success Postcondition**: | The user's account and information shall be deleted from NoCom servers. |
| **USER STEPS**:  1. The user will click the ‘settings’ button.  3. The user will click the ‘delete account’ option in settings.  5. The user clicks the ‘yes’ button.  5.1 The user clicks the ‘no’ button. | **SYSTEM RESPONSE**:  2. The system shall display users settings.  4. The system shall prompt the user again with a ‘yes’ and a ‘no’ button to ensure they want to delete their account.  6. The system shall delete the user account and account information from NoCom servers.  6.1 The system shall return the user to the users settings display. |

| Name/Title: | Create Account |
| --- | --- |
| ID: | UC-012 |
| **Description**: | This allows the user to create an account on the NoCom application. |
| **System Under Design:** | NoCom |
| **Primary Actor**: | User |
| **Participants**: | NoCom servers |
| **Goal**: | Shall allow the user to create an account on the NoCom application. |
| **Related Use Cases**: | **Instance of:**  UC-001 <Login> |
| **Precondition**: | The user must not be logged into the application. |
| **Success Postcondition**: | Creates an account with the information entered by the user. |
| **USER STEPS**:  1. The user will click the ‘create account’ button.  3. The user will enter username, password, and email. | **SYSTEM RESPONSE**:  2. The system shall direct the user to the create account display.  4. The system shall create an account with username, password, and email entered by the user and save it to the NoCom servers. |