Detailed User Interaction Flow and Architecture

Welcome and Introduction

Objective: Provide a warm welcome and an overview of the tool's capabilities to make users comfortable and aware of the features.

Steps:

1. **Welcome Screen**:
   - Display a friendly welcome message.
   - Briefly explain what the tool can do (e.g., "Welcome to API Mockup Tool. This tool helps you create and refine API specifications using natural language inputs or file uploads.")
   - Offer a guided tour for first-time users.

2. **User Options**:
   - **Create New API**: Start a new API specification from scratch.
   - **Modify Existing API**: Edit or refine an existing API specification.

Possible User Actions:

- Click on "Create New API."
- Click on "Modify Existing API."
- Take the guided tour.

Starting a New API

Objective: Gather initial inputs from the user to start building the API specification.

Detailed User Interaction Flow and Architecture

Steps:

1. **Choose Input Method**:
   - **Text Prompt**: Enter a high-level description of the API.
   - **File Upload**: Upload an OpenAPI spec, JSON, or YAML file.

2. **Text Prompt Flow**:
   - User enters a description like "I need an API to fetch user details by user ID."
   - System processes the prompt using OpenAI to generate an initial API specification.
   - Display the generated specification and ask the user for confirmation or additional details.

3. **File Upload Flow**:
   - User uploads an OpenAPI spec, JSON, or YAML file.
   - System parses the file and generates a preliminary API specification.
   - Display the parsed specification for user review.

Possible User Actions:

- Enter a text prompt and submit.
- Upload a file and submit.
- Review the generated or parsed specification.

Processing Initial Input

Objective: Use OpenAI to process the user's input and generate a preliminary API specification.

Detailed User Interaction Flow and Architecture

Steps:

1. **Text Prompt Handling**:

   - System sends the user's text prompt to OpenAI with a well-crafted prompt.

   - Example OpenAI Prompt:

     "Generate an API specification for the following description: {user_prompt}. The response should be in JSON format, including endpoints, parameters, and sample request and response bodies."

   - Receive the response and parse it to create an initial API specification.

   - Validate the response to ensure it is in the correct format.

2. **File Upload Handling**:

   - System parses the uploaded file (OpenAPI spec, JSON, YAML).

   - Generate a preliminary API specification based on the parsed content.

   - Validate the parsed content to ensure it meets the required standards.

Possible User Actions:

- Review the initial API specification.

- Provide feedback or additional details to refine the specification.

Clarifying Requirements

Objective: Ask targeted questions to gather more details and refine the API specification.

Steps:

1. **Identifying Gaps and Ambiguities**:

   - Analyze the initial API specification to identify any missing or unclear details.

- Generate follow-up questions to gather the required information.

2. **Examples of Follow-up Questions**:
   - "What details do you need about the user?"
   - "Do you want to include any query parameters or headers?"
   - "Can you provide a sample request and response?"
   - "What should be the response status codes?"

3. **User Responses**:
   - User provides answers to the follow-up questions.
   - System updates the API specification based on the user's inputs.

4. **Predefined Options for Common Attributes**:
   - Provide predefined options for common attributes (e.g., string, integer, boolean).
   - Allow users to select from these options to simplify the process.

Possible User Actions:
- Answer follow-up questions.
- Select predefined options for attributes.
- Provide additional details or examples.

Iterative Refinement

Objective: Allow the user to iteratively refine the API specification through multiple rounds of questions and answers.

Detailed User Interaction Flow and Architecture

Steps:

1. **Iterative Process**:
   - System asks a series of questions based on the current state of the specification.
   - User answers the questions, and the system updates the specification accordingly.
   - Display the updated specification after each round of questions.

2. **Iteration Limit**:
   - Implement a configurable limit on the number of iterations to prevent infinite loops.
   - Provide an option to manually refine the specification if the iteration limit is reached.

Possible User Actions:

- Continue refining the specification by answering questions.
- Manually edit the specification if needed.
- Confirm the specification when satisfied.

Handling Different User Scenarios

Objective: Cater to both technical and non-technical users by providing tailored interactions and options.

Technical User Scenario:

- **Upload Detailed OpenAPI Spec**: User uploads a comprehensive specification file.
- **System Actions**:
  - Parse the file and identify potential enhancements.

- Ask the user if they want to add more endpoints, parameters, or response types.
- Provide options to manually edit the specification.

Non-Technical User Scenario:

- **Provide High-Level Description**: User provides a general description of the API.
- **System Actions**:
  - Generate a basic specification using OpenAI.
  - Ask simple, guided questions to refine the specification.
  - Provide examples and templates for common API patterns.
  - Allow the user to select from predefined attributes and endpoints.

Possible User Actions:

- Upload a detailed OpenAPI spec and refine it.
- Provide a high-level description and answer guided questions.
- Select predefined options or templates.

Error Handling and Validation

Objective: Ensure robust validation of user inputs and OpenAI responses to prevent errors and provide a seamless experience.

Steps:

1. **Input Validation**:
   - Validate user inputs (text prompts and file uploads) to ensure they are in the correct format.
   - Provide feedback and request corrections if needed.

2. **Response Validation**:
   - Validate OpenAI responses to ensure they are complete and correctly formatted.
   - Handle cases where the response is malformed or incomplete.

3. **Error Messages and Recovery**:
   - Provide clear and informative error messages to the user.
   - Offer options to recover from errors and continue the process.

Possible User Actions:
- Correct inputs based on validation feedback.
- Provide additional details if the OpenAI response is unclear.
- Retry the process if an error occurs.

Confirming and Saving the Specification

Objective: Allow the user to review, confirm, and save the final API specification with version control.

Steps:
1. **Review and Confirmation**:
   - Display the final API specification for user review.
   - Highlight changes and new additions.
   - Ask the user to confirm or make further modifications.

2. **Saving with Version Control**:

   - Save the confirmed specification to the database.

   - Implement version control to track changes and allow reverting to previous versions.

Possible User Actions:

- Review the final specification.

- Confirm and save the specification.

- Make further modifications if needed.

Detailed Mind Map for User Interaction

Detailed Mind Map for User Interaction

User Interaction for API Mockup Tool

??? Welcome Screen

?   ??? Display Welcome Message

?   ??? Explain Tool Capabilities

?   ??? Options: Create New API, Modify Existing API

?   ??? Provide Tutorial/Guided Tour

??? Starting a New API

?   ??? Choose Input Method

?   ?   ??? Text Prompt

?   ?   ?   ??? Enter High-Level Description

?   ?   ?   ??? Example: "Fetch user details by user ID"

?   ?   ??? File Upload

Detailed User Interaction Flow and Architecture

? ? ??? Upload OpenAPI Spec, JSON, or YAML

? ? ??? Parse File and Display Summary

??? Processing Initial Input

? ??? Text Prompt Handling

? ? ??? Use OpenAI for Initial Specification

? ? ??? Example OpenAI Prompt

? ? ??? Receive and Parse Response

? ? ??? Validate Response

? ??? File Upload Handling

? ??? Parse Uploaded File

? ??? Generate Preliminary Specification

? ??? Validate Parsed Content

??? Clarifying Requirements

? ??? Identifying Gaps and Ambiguities

? ? ??? Analyze Initial Specification

? ? ??? Generate Follow-up Questions

? ??? Examples of Follow-up Questions

? ? ??? "What details about the user?"

? ? ??? "Include query parameters/headers?"