Design of Conversational Experiences

Assignment 2: Feature Formulator Bot

Submitted by

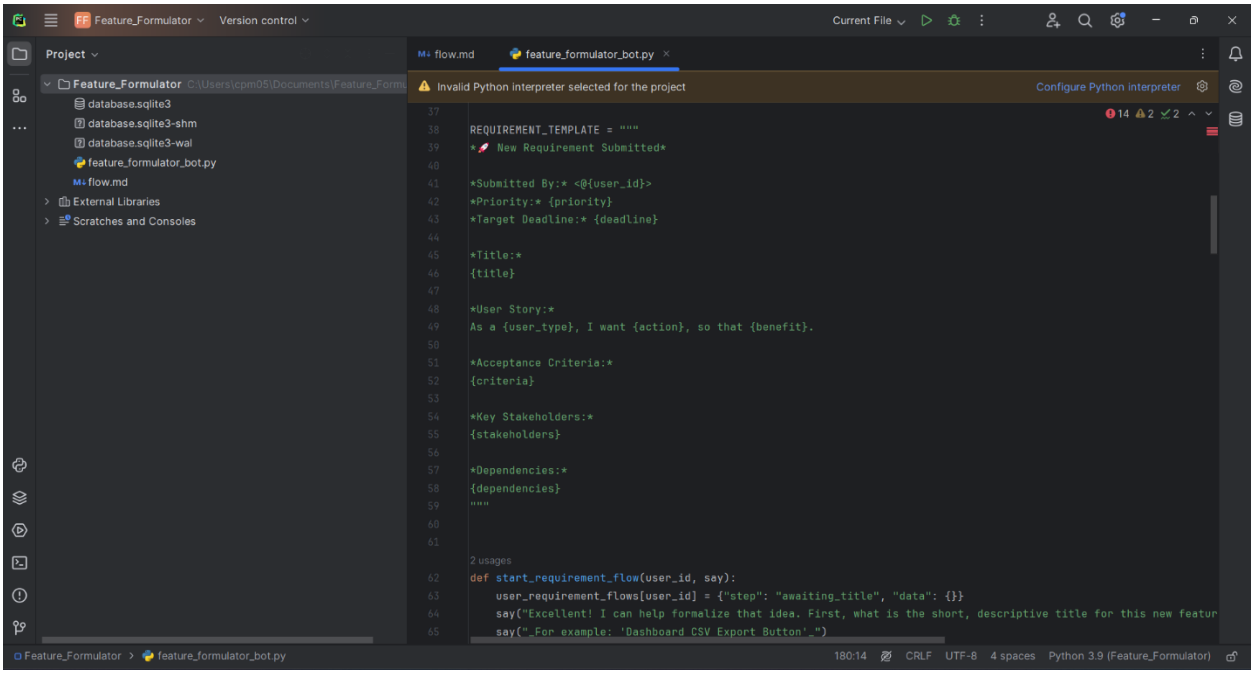
Chandru M P

2024TM93120

1. Design the Bot Persona

This section defines the identity, purpose, and personality of the Feature Formulator bot.

- **Bot Name:** Feature Formulator
- **Environment:** The bot operates exclusively within the Slack messaging platform, interacting with users via Direct Messages (DMs).
- **Target Audience:** Any member of a product development team, including Product Managers, Engineers, Designers, QA Testers, and other internal stakeholders who wish to submit a feature idea.
- **Purpose (Single Goal):** To guide a user through a structured, goal-oriented conversation to capture all the necessary details for a new feature requirement, thereby transforming a simple idea into an actionable ticket.
- **Runtime Variations:** The bot operates in two distinct modes:
  - **Conversational Mode:** When not in an active flow, the bot uses the ChatterBot library to engage in simple, non-goal-oriented conversation (e.g., greetings).
  - **Data Gathering Mode:** Once a requirement flow is initiated, the bot follows a strict, linear path to collect specific pieces of information. It does not deviate from this path unless the user cancels.
- **Service Branding:** The bot presents itself as a professional, efficient, and helpful assistant. Its branding is an extension of the development team's focus on clarity and process.
- **Core Values:**
  - **Clarity:** Aims to remove ambiguity from new ideas.
  - **Efficiency:** Saves time by collecting all information in a single interaction.
  - **Collaboration:** Creates a single, accessible entry point for all team members to contribute.
- **Derived Personality Traits:**
  - **Helpful & Guiding:** Proactively provides examples and clear instructions.
  - **Patient:** Follows a step-by-step process without rushing the user.
  - **Meticulous & Organized:** Cares about getting the details right.
  - **Politely Formal:** Maintains a professional but approachable tone.



2. Design the Conversation Flow

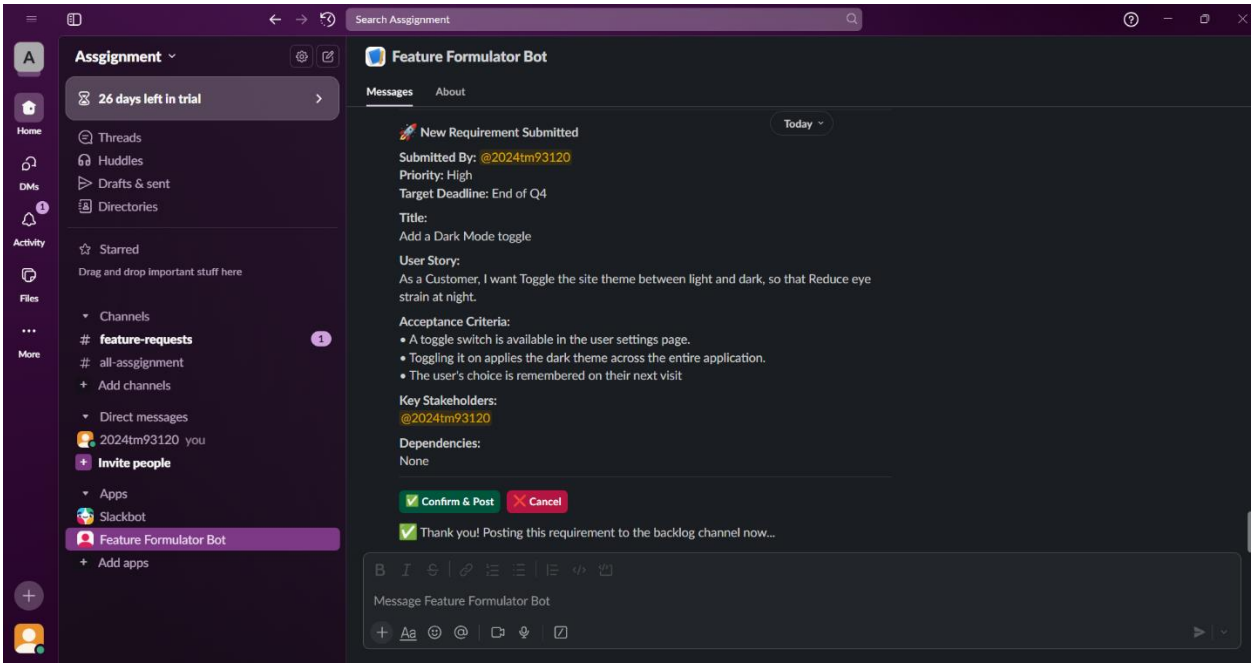
This section outlines the conversational logic, including user inputs, bot intents, and the paths the conversation can take.

2.1. Key Utterances, Intents, and Entities

- **Intents (User's Goal):**
  - start_requirement_flow: The user wants to submit a new feature.
  - provide_information: The user is providing an answer to a bot's question.
  - cancel_flow: The user wants to stop the current process.
  - greet: The user is engaging in simple conversation.
- **Utterances (Examples of User Input):**
  - start_requirement_flow: "I have a new feature idea," "we should implement dark mode," "new requirement."
  - cancel_flow: "cancel."
- **Entities (Data Extracted by the Bot):**
  - title: The name of the requirement.
  - user_type, action, benefit: The three parts of the user story.
  - criteria: The list of acceptance criteria.
  - stakeholders: A list of mentioned Slack users.
  - dependencies: A description of dependencies.
  - deadline: The target date or timeline.
  - priority: The selected priority level.

2.2. Main Conversation Flow ("Happy Path")

1. **Initiation:** The user DMs the bot with a keyword (e.g., "new idea"). The bot recognizes the start_requirement_flow intent and begins the guided conversation.
2. **Title Collection:** The bot asks for and stores the title.
3. **User Story Collection:** The bot asks for and stores the user_type, action, and benefit in three separate sub-steps.
4. **Acceptance Criteria Collection:** The bot asks for and stores the criteria.
5. **Stakeholder Collection:** The bot asks for and stores the stakeholders.
6. **Dependency Collection:** The bot asks for and stores the dependencies.
7. **Deadline Collection:** The bot asks for and stores the deadline.
8. **Priority Selection:** The bot presents interactive buttons to capture the priority.
9. **Confirmation:** The bot displays a complete summary of all collected entities and asks for final confirmation with "Confirm & Post" and "Cancel" buttons.
10. **Submission:** Upon confirmation, the bot posts the formatted requirement to the designated Slack channel.



Search Assignment

Assignment

26 days left in trial

Threads
Huddles
Drafts & sent
Directories

Starred
Drag and drop important stuff here

Channels
feature-requests 1
all-assignment
Add channels

Direct messages
2024tm93120 you
Invite people

Apps
Slackbot
Feature Formulator Bot
Add apps

Home
DMs
Activity
Files
More

Feature Formulator Bot

Messages About

Today

Feature Formulator Bot APP 17:14
Got it. Are there any dependencies on other teams or services? If not, type None.

2024tm93120 17:14
None

Feature Formulator Bot APP 17:15
Almost done! Is there a target deadline or release date? If flexible, type None.

2024tm93120 17:15
End of Q4

Feature Formulator Bot APP 17:15
That's everything I need. What is the priority of this request?

Low | Medium | High | Critical

Great! Please review the requirement below before I post it.

🚀 **New Requirement Submitted**

Submitted By: @2024tm93120
Priority: High
Target Deadline: End of Q4

Title:
Add a Dark Mode toggle

User Story:

Message Feature Formulator Bot

Search Assignment

Assignment

26 days left in trial

Threads
Huddles
Drafts & sent
Directories

Starred
Drag and drop important stuff here

Channels
feature-requests 1
all-assignment
Add channels

Direct messages
2024tm93120 you
Invite people

Apps
Slackbot
Feature Formulator Bot
Add apps

Home
DMs
Activity
Files
More

Feature Formulator Bot

Messages About

Today

2024tm93120 17:14
Toggle the site theme between light and dark

Feature Formulator Bot APP 17:14
Perfect. And finally, what is the benefit or goal they achieve by doing this?

For example: 'so that I can perform offline analysis in Excel'

2024tm93120 17:14
Reduce eye strain at night

Feature Formulator Bot APP 17:14
Great user story! Now, please list the Acceptance Criteria.

Example:
- The button is on the main dashboard.
- The downloaded file is named 'export.csv'_

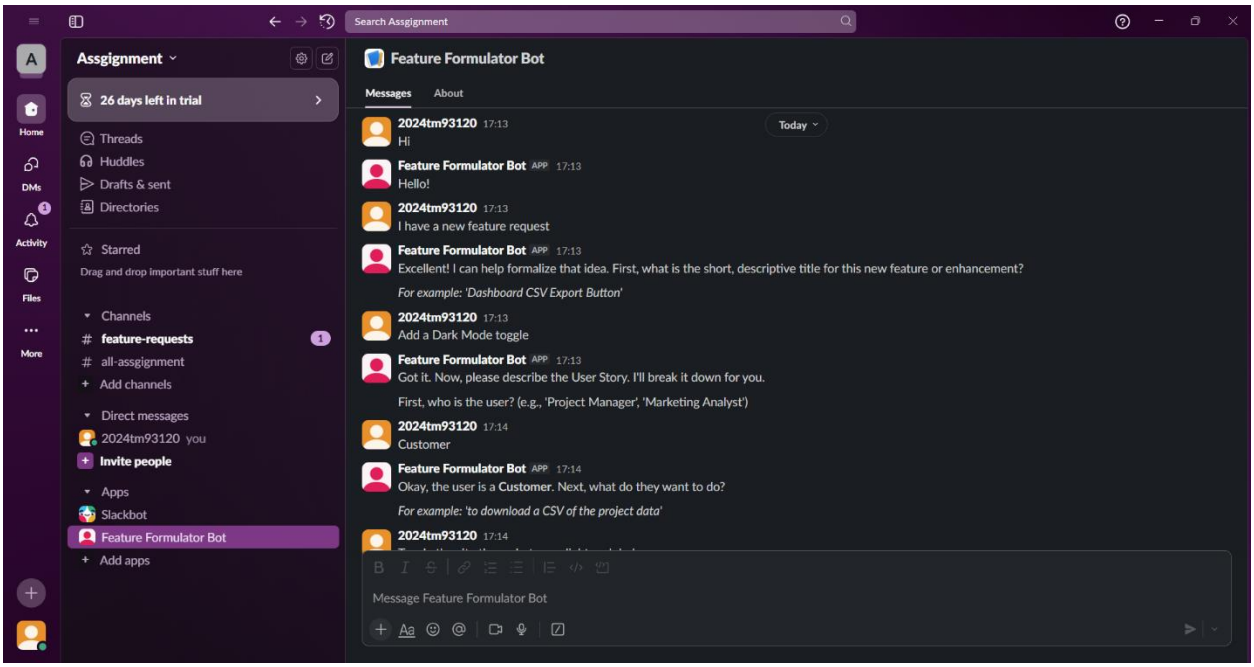
2024tm93120 17:14
- A toggle switch is available in the user settings page.
- Toggling it on applies the dark theme across the entire application.
- The user's choice is remembered on their next visit

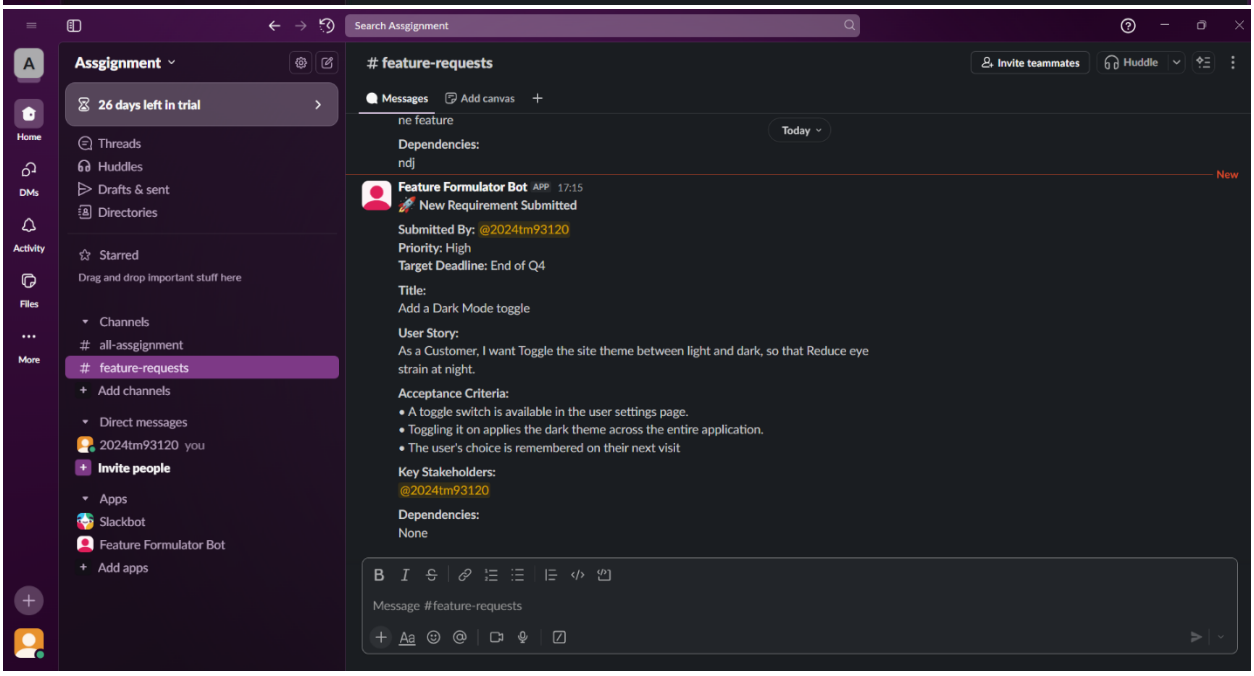
Feature Formulator Bot APP 17:14
Thank you. Now, please @mention any key stakeholders. If there are none, just type None.

2024tm93120 17:14
@2024tm93120

Feature Formulator Bot APP 17:14

Message Feature Formulator Bot

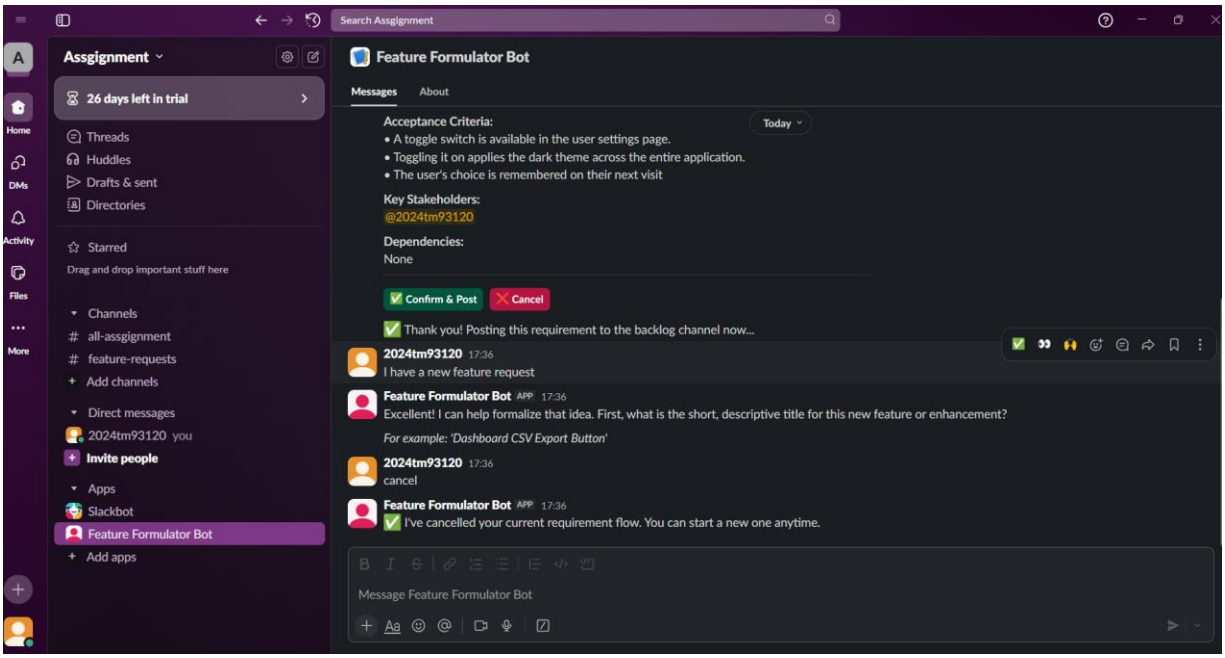




2.3. Alternate Conversation Flows

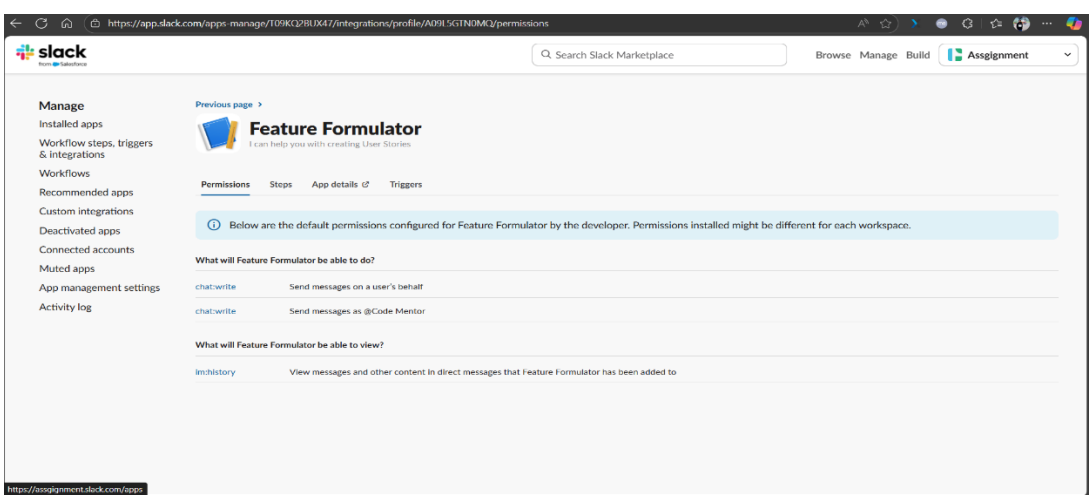
- **User Cancels Mid-Flow:**
  - **Trigger:** User types cancel at any point during the data collection.
  - **Flow:** The bot detects the cancel_flow intent, deletes the in-progress requirement data, confirms the cancellation with the user, and exits the flow.
- **User Cancels at Confirmation:**

    - **Trigger:** User clicks the " Cancel" button on the final review screen.
    - **Flow:** The bot deletes the requirement data and confirms the cancellation.
- **User Provides Unrelated Input:**
    - **Trigger:** User is not in a flow and says something that doesn't match a keyword (e.g., "hello").
    - **Flow:** The bot passes the input to ChatterBot, which provides a general conversational response based on its training. The main flow is not initiated.



2.4. Required Integrations

- **Slack API:** The entire bot is built on the Slack platform, using the Slack Bolt for Python SDK to handle events, messages, and API calls.
- No other external integrations are required for the core functionality.



3. Build and Test the Bot

This section covers the technical implementation and the strategy for verifying its functionality.

3.1. Implementation (Build)

The bot is built as a single Python script (server.py) with the following architecture:

- **Language:** Python
- **Primary Framework:** slack_bolt (Official Slack SDK for Python)
- **Connection Protocol:** Socket Mode, which provides a secure connection to Slack's APIs without requiring a public URL or exposing the server to the internet.
- **Conversational AI:** ChatterBot is used for basic, non-goal-oriented conversations. Intent recognition for starting the main flow is handled by a simple keyword-matching system.
- **State Management:** The bot uses a simple in-memory Python dictionary to track each user's progress through the requirement gathering flow. This is lightweight but will reset if the bot restarts.

3.2. Verification

- **Manual / User Acceptance Testing (UAT):**
  - Execute the "Happy Path" flow from start to finish and verify the requirement is posted correctly.
  - Test the cancel command at various stages of the flow.
  - Test the "Cancel" button on the final confirmation screen.
  - Send non-keyword messages to the bot (e.g., "hello") to verify that the ChatterBot fallback is working.
  - Verify that the bot correctly formats and posts requirements with multi-line inputs for criteria and dependencies.

4. Deploy the Bot

This section describes how to host the bot for continuous operation.

- **Platform / Channel:** The bot is designed for and deployed on the **Slack** platform.
- **Hosting Strategy:** The Python script is a long-running process and must be hosted on a server that is always online. Suitable hosting options include:

    - A cloud provider like **Heroku**, **AWS EC2**, or **Google Cloud Run**.
    - An on-premise server within the organization.
- **Deployment Steps:**
    1. **Prepare the Server:** Set up a server environment with Python installed.
    2. **Install Dependencies:** Install all required packages using pip install -r requirements.txt.
    3. **Set Environment Variables:** Securely set the SLACK_BOT_TOKEN, SLACK_APP_TOKEN, and REQUIREMENT_CHANNEL_ID as environment variables on the server. **Do not** hard-code them or use a .env file in a production environment.
    4. **Run the Process:** Start the bot using a process manager like systemd (on Linux) or pm2 to ensure that it runs continuously and automatically restarts if it crashes.
       Ex: pm2 start server.py --name feature-formulator-bot