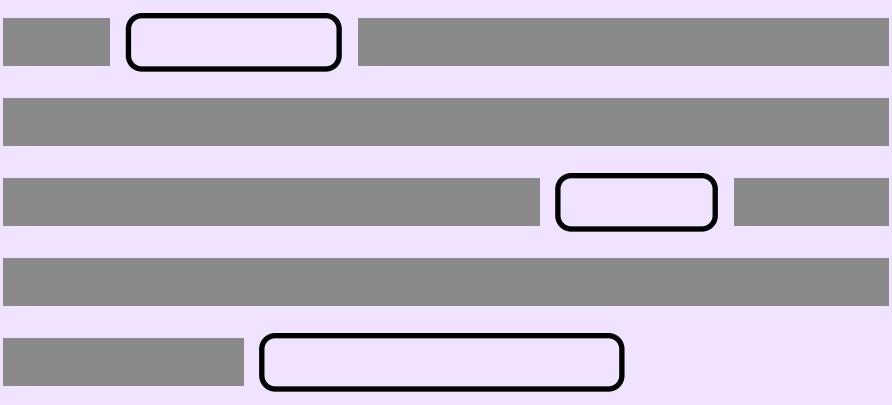
SLOT-FILLING

Streamlining user input for seamless task completion



Overview

Slot-filling is an interaction pattern where the copilot gathers specific pieces of information (slots) from the user to complete a task or answer a query. This approach involves breaking down complex requests into manageable components and guiding the user to provide the necessary details step-by-step. Slot-filling is particularly useful for tasks that require structured data input, such as "Book a meeting" or "Generate a report."

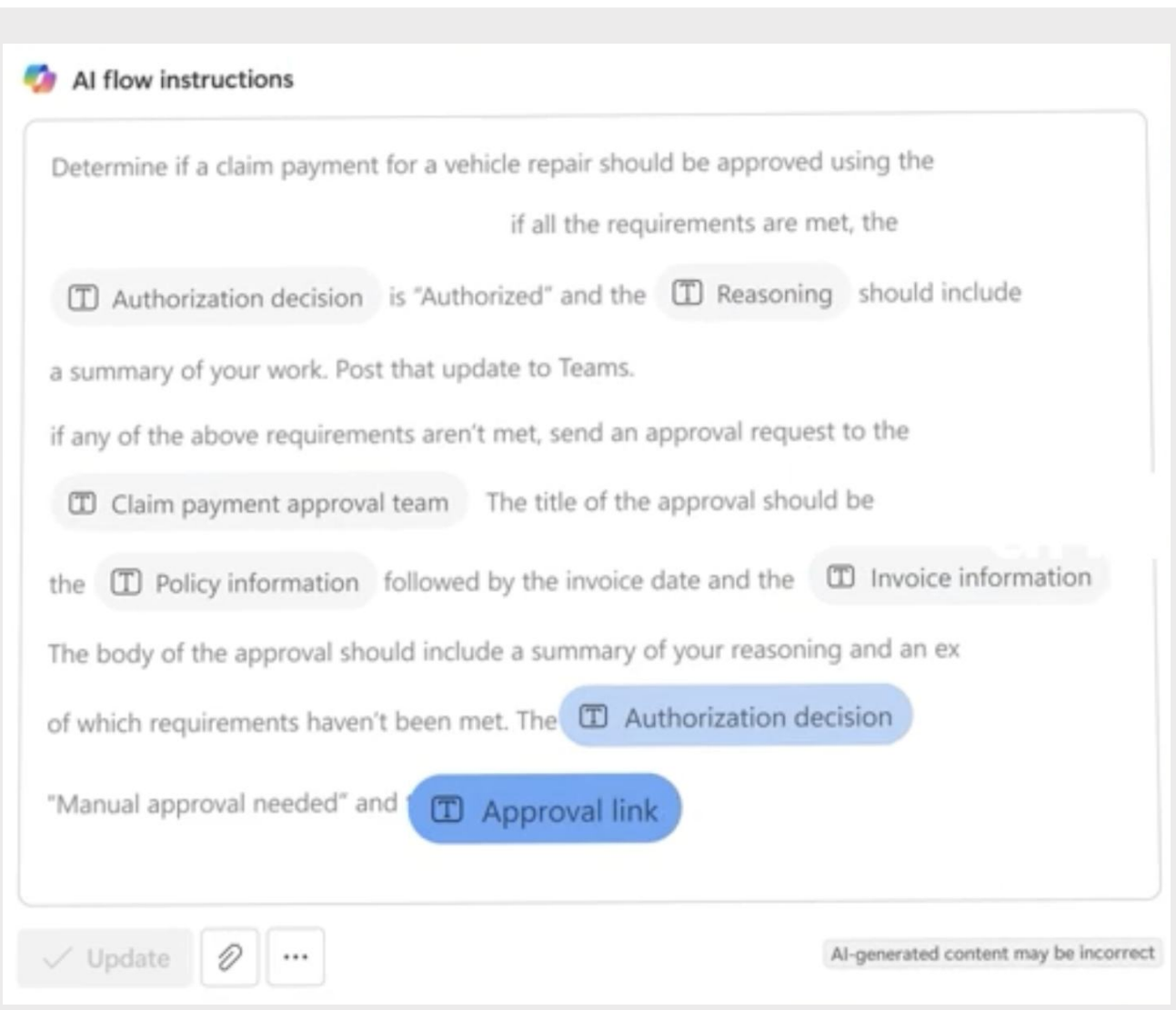
Common Issues
Users often provide partial information, making it difficult for the system to fulfil their requests accurately. Without clear guidance, users may be unsure what information is needed or how to provide it, leading to confusion. Repeatedly asking for missing information can also be time-consuming and frustrating for users. Furthermore, incorrect or ambiguous inputs can result in errors in task completion or require additional clarification steps, hindering efficiency and accuracy.

Solution

We can leverage slot-filling to address these common issues by implementing a structured and user-friendly approach to information gathering. By breaking down requests into specific pieces of information, we ensure all necessary details are collected systematically, minimising the risk of incomplete or incorrect inputs. We can design clear and concise prompts for each slot, guiding users to provide the required information and reducing confusion about what is needed at each step.

Benefits of slot-filling

- Efficiency: Automates repetitive and boilerplate tasks, saving time for developers.
- Accuracy: Reduces the chance of errors by ensuring that placeholders are filled with contextually appropriate data.
- Consistency: Ensures consistent application of patterns and structures across different parts of a project.



Example of designing slots for an AI flow in Microsoft Power Automate

Step 1: Identify key information slots

Determine critical information needed for Nintex tasks, such as:
- User details
- Process-specific data
- Dates
- Document types

Step 2: Define slot types and validation rules

- Text slots: Names, descriptions, etc. Use Regex for validation.
- Number slots: Quantities, IDs, etc. Use range checks.
- Date slots: Scheduling, deadlines, etc. Use date format validation.
- Boolean slots: Yes/No answers. Simple true/false validation.

Step 3: Create user-friendly prompts

- Design clear, context-aware prompts (e.g. “Please enter the workflow ID”
- Provide default values or examples to guide users

Step 4: Implement dynamic slot-filling

- Use dynamic forms and adaptive interfaces to adjust based on user inputs
- Provide real-time feedback and validation to users as they fill slots

Step 5: Ensure data integrity and security

- Apply appropriate data validation and sanitisation techniques
- Ensure secure handling of sensitive information

Step 6: Iterative testing and improvement

- Conduct usability testing to refine the slot-filling experience
- Gather feedback and iterate to improve usability and efficiency

Example use case: Employee onboarding

1. Identify Slots: Employee name, role, start date, department, manager, etc.
2. Prompts: “Please enter the employee’s full name”, “Select the employee’s start date”, etc.
3. Validation: Ensure the start date is a future date, role matches predefined roles, etc.
4. Dynamic Interface: If the role is “Manager”, prompt for additional manager-specific information.