SymTrain Simulation Intelligence Assistant

Andrew Whiteside, Ida Metiam, Rahul Patel, Rowan Amanna

Project Overview/Goal

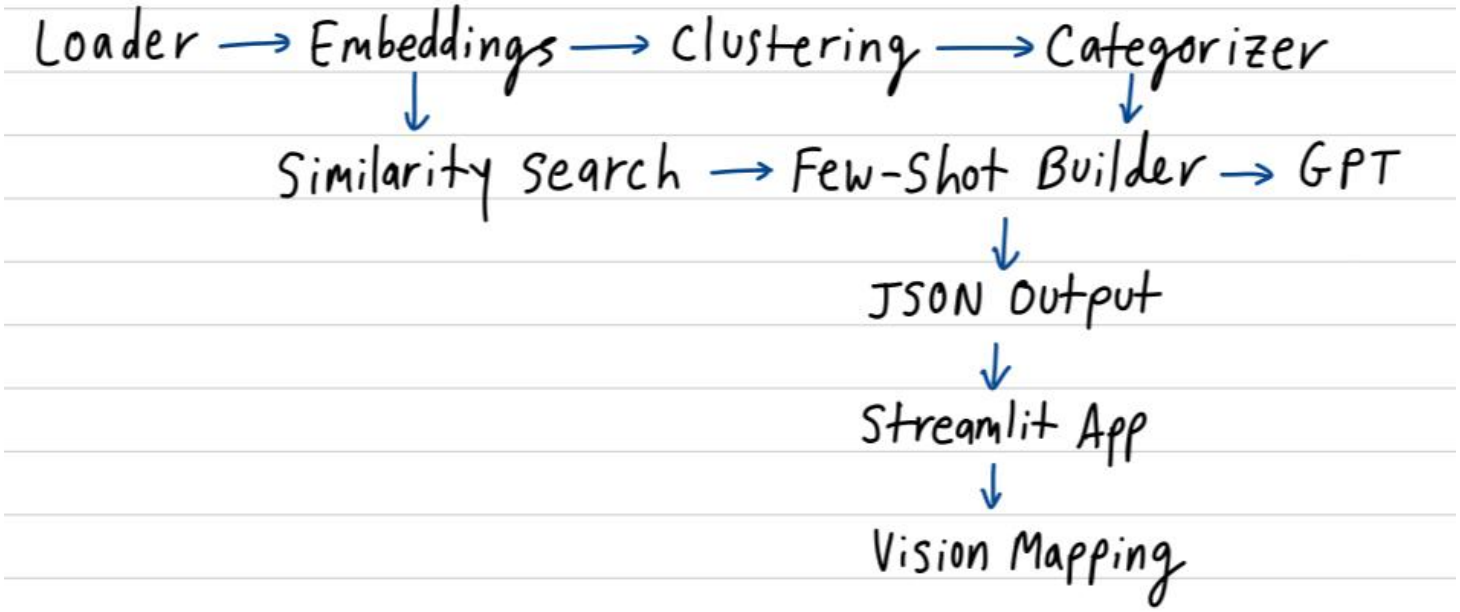
We want to build an automated system that:

- Understands a customer request
- Predicts the request category
- Extracts the reason
- Generates clear, actionable steps
- Delivers everything through a Streamlit App
- Maps steps to UI images

Dataset Overview

- Each simulation contains:
  - JSON transcript (‘audioContentItems’)
  - UI images + hotspot metadata (‘visualContentItems’)
- We processed:
  - Transcript merging
  - Speaker-preserving dialogue
  - Image metadata

System Architecture



Embeddings and Clustering

DistilBERT embeddings used for:

- Categorizing all simulations
- Similarity search
- Selecting few-shot examples

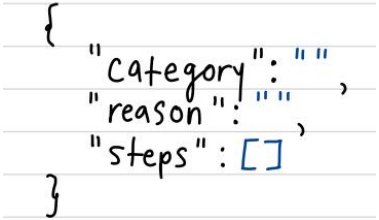
KMeans clustering built categories such as:

- Payment Issues
- Order Status
- Insurance Claims

LLM Pipeline

Steps:

1. Predict category
2. Retrieve category-specific examples
3. Build few-shot prompt
4. Generate:



Output is clean, consistent, step-by-step.

Streamlit Application

Features:

- Accepts user request
- Displays category + reason
- Generates procedural steps
- Shows mapped UI images

Provides a fully interactive user-facing demo.

Example Test Input

Input: “I need to update my payment method.”

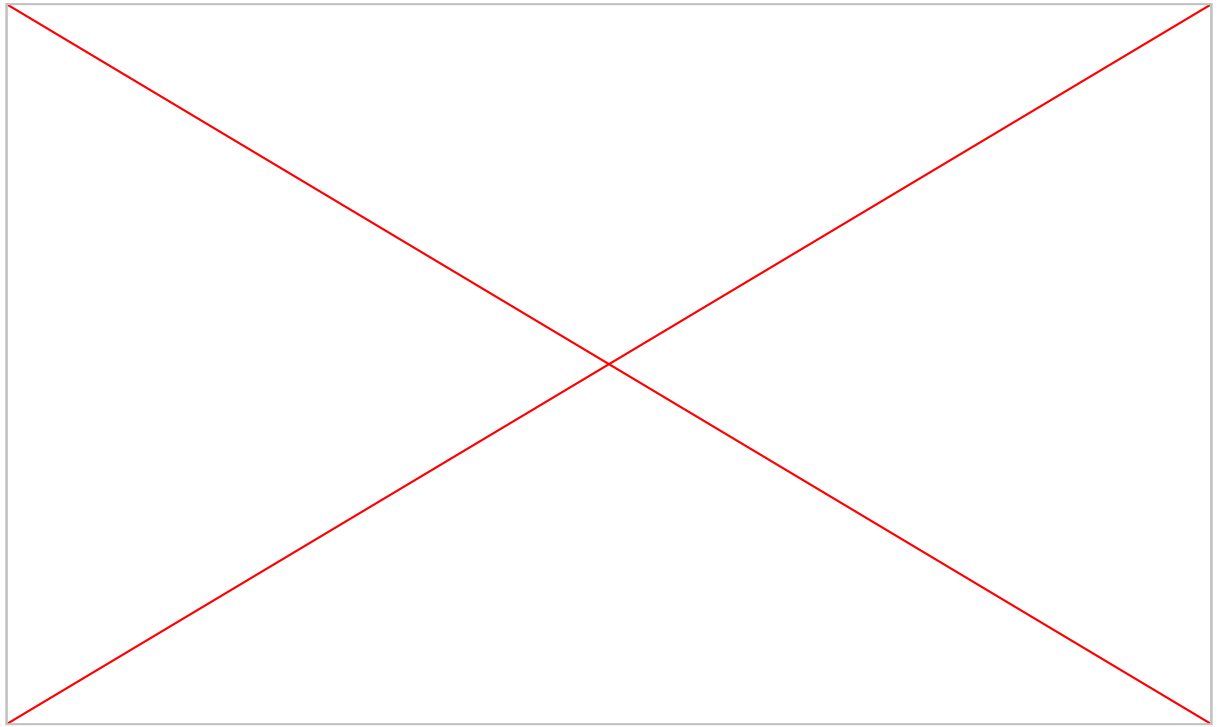
Output:

- Category: Payment Issue
- Reason: Update card information

Steps:

1. Open Account Settings
2. Select Payment Methods
3. Click Edit
4. Enter new card details
5. Save changes

Demo



Vision-Based Step Mapping

Implemented:

- Semantic matching between step text + hotspot labels
- Relevance scoring
- Image identification (‘fileID’)

Results:

- 5/5 steps mapped + accurate!

Vision Mapping Example

Example Step: “Click on Account Settings…”

Mapped to:

- Image: account_settings.png
- Hotspot: “Account Settings”
- Type: button
- Relevance: 1.00

Dockerization

We containerized the entire project using:

- Dockerfile
- docker-compose.yml

Includes:

- Streamlit service
- HuggingFace cache
- Ollama model endpoint

Ensures reproducibility for grading.

Conclusion

We built a complete, end-to-end customer support system that:

- Loads & processes simulations
- Categorizes requests
- Extracts reasons
- Generates clean procedural steps
- Runs inside Streamlit & Docker
- Includes the fully implemented Vision Task