PURDUE UNIVERSITY | The Data Mine

Way cool Aircraft Database Engine (WADE)

Markee Jackson, Noah Aguilera, Van Nguyen, Volker Cloud, Alec Magsombol, Aisha Lawrence, Ashley Torres, Rahul Paul,
Yair Torres, Carol Wang, David Wang, Micheal Tomdio, Sergio Villarreal

INTRODUCTION/BACKGROUND

This project aims to develop a chatbot that helps and provides information regarding aircraft manuals and other resources. It takes a long to time to read through a manual that can be hundreds of pages long. This chatbot will help V2X employees, developers, and engineers who want to learn the basic functions and features of relevant products and save them time by providing the information they need almost instantaneously.

METHODOLOGY

The chatbot consists of three main components: a front-end interface for user interaction, a back-end powered by a Large Language Model (LLM), and an API that connects the front and back end while also filtering for potentially malicious input. The LLM is trained on publicly available Pam Am aircraft manuals from Miami EDUs Digital Collections Library.

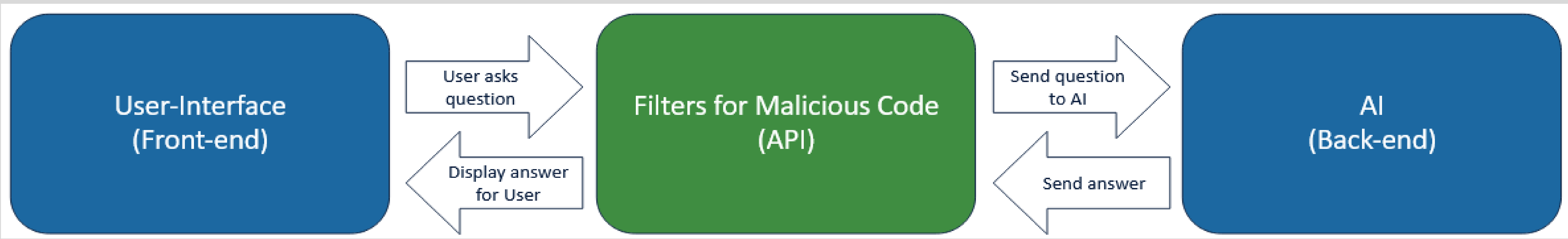

Figure 1. Chatbot Structure

FRONT-END PROCESS

- Using Miro, we designed a simplistic user interface for users to interact with the chatbot
- It is designed to look familiar to other popular chatbots so experienced users feel comfortable using it
- The design was kept minimalistic to not overwhelm first time users.
- We utilized HTML and CSS to translate the design to the web Django framework
- JavaScript was used to add dynamic functionality so users can interact with the chatbot in real time

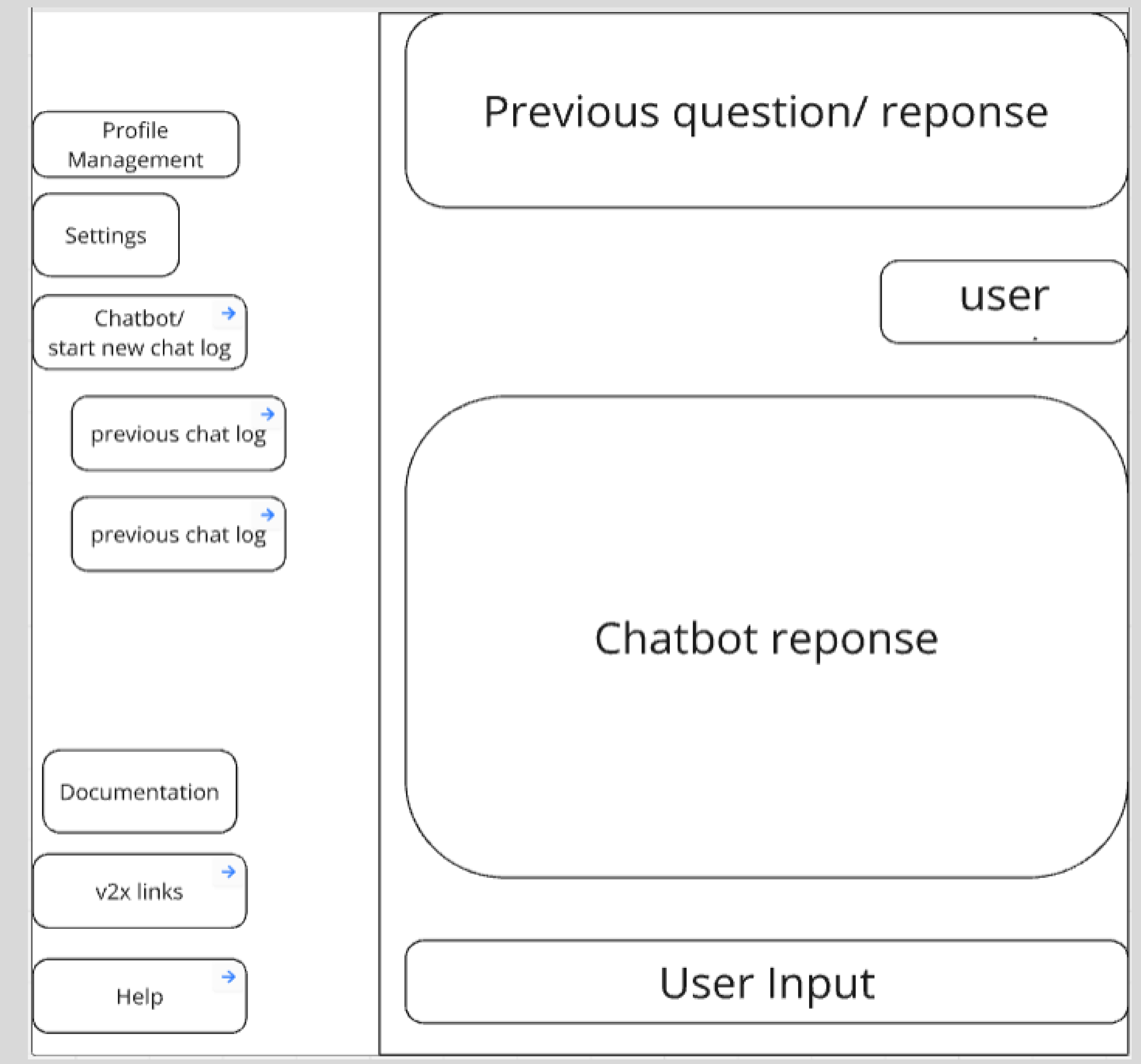

Figure 2. Sketch of chatbot's user interface in Miro

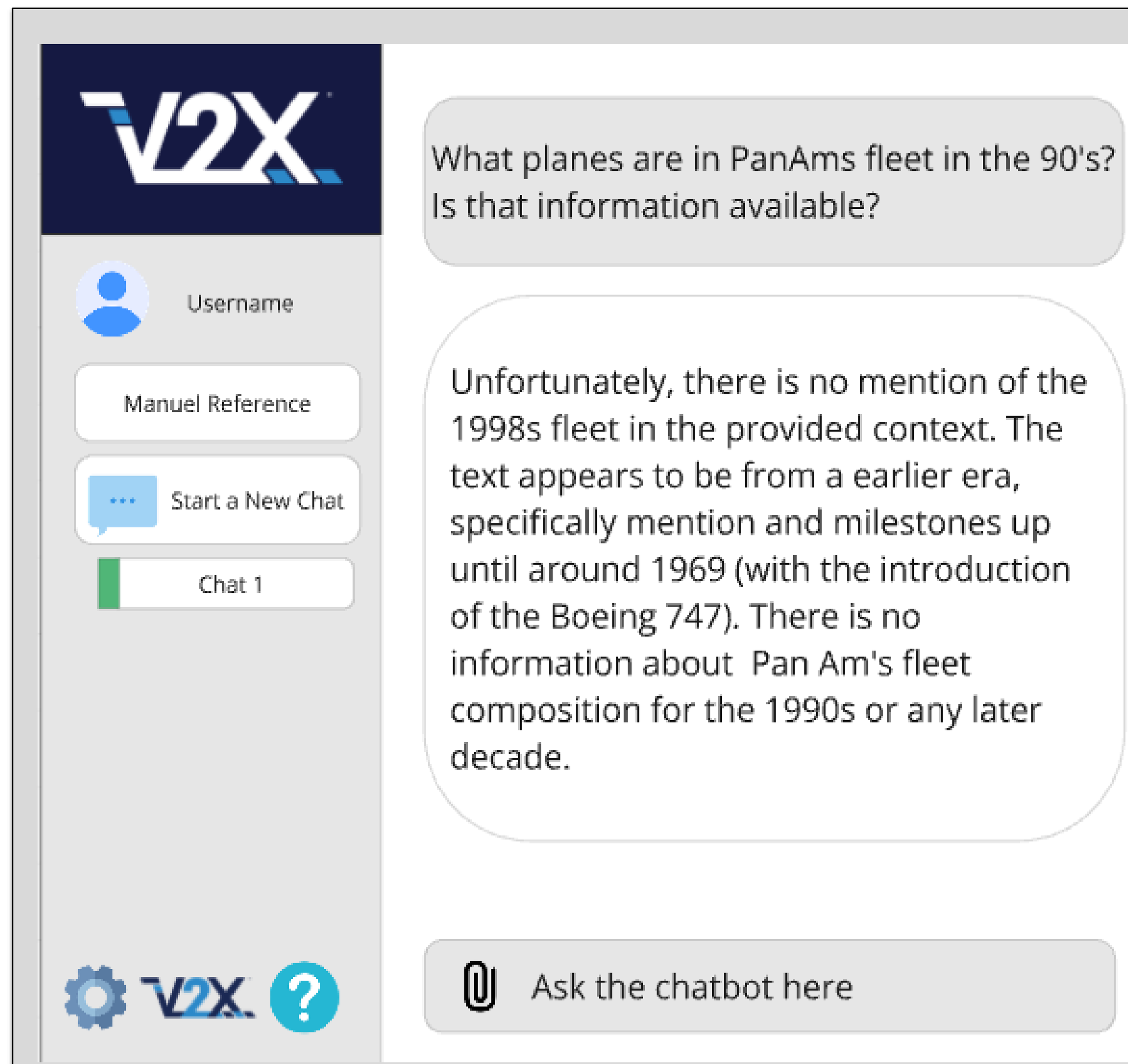

Figure 3. Example of a user interacting with the chatbot

BACK-END PROCESS

Data: Pan American World Airways Records
- Contains 114 scanned PDF aircraft manuals from Pan Am
- Hosted on Miami EDUs Digital Collections Library

Large Language Model: LLaMA 3.1
- Open-weight model from Meta, optimized for low-latency inference and fine-tuning
- Used as the backbone for retrieval-augmented generation (RAG) in our chatbot

Optical Character Recognition: PaddleOCR
- Lightweight, high-accuracy OCR system developed by Baidu
- Used to extract structured text from scanned aircraft manual PDFs

Text Preprocessing: NLTK
- Remove non-ASCII characters, extra whitespace, HTML tags, etc
- Lemmatization (reduce words to their simplest form)

Embedding Model: nomic-embed-text
- Transformer-based model from Nomic AI optimized for text embeddings
- Converts text chunks into vector representations for semantic search

Vector Database: ChromaDB
- Lightweight and fast embedding database built for RAG pipelines
- Enables efficient similarity search across over 120,000 document chunks

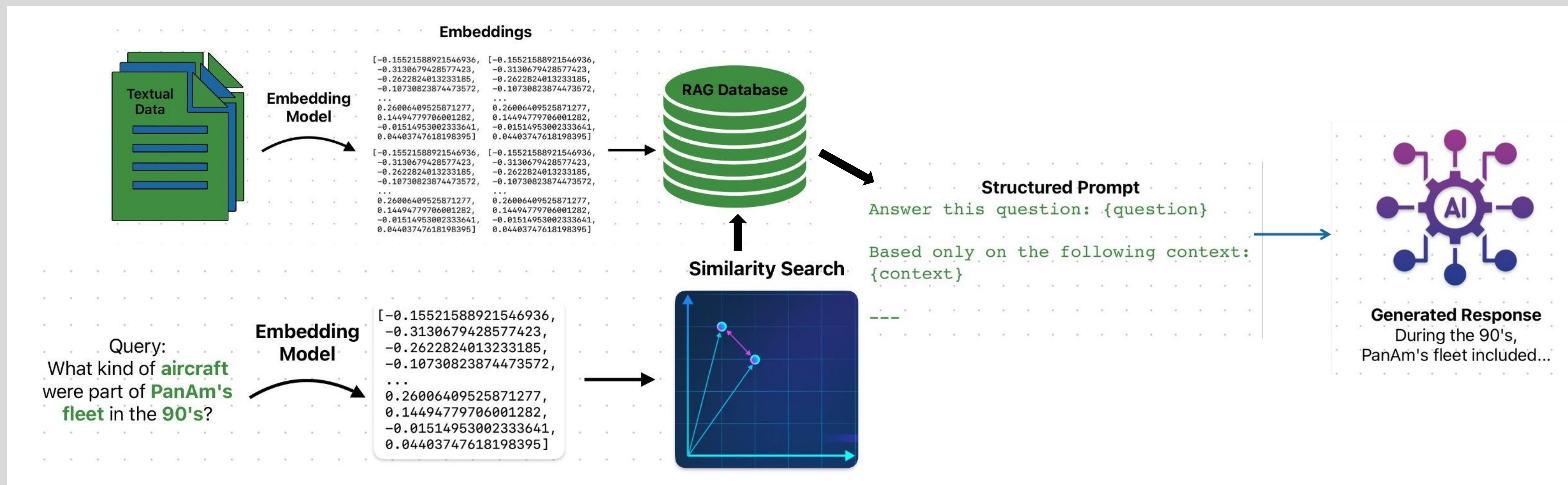

Figure 4. Retrieval Augmented Generation System Overview

API PROCESS

- Using Django's project and app structure. This makes it easier to maintain, scale, and manage the chatbot as the project evolves
- App Configuration handles the registration of the API app, which is essential for making sure Django recognizes the project
- Middleware that processes requests and responses and handles system security
- URL Routing is where we define the API endpoints such as handling chatbot requests

ACKNOWLEDGMENTS

Special thanks to our mentor Doug, Jennifer, Dustin from V2X, our TA Elias, and Doug, Pete and everyone else from the Data Mine Staff!
Some icons used in the chatbot are designed by Flaticon

FUTURE WORK

We hope to further improve the design by moving away from the blocky one-colored design, improve security with better filtering in user's prompt and improve model response by using a different embedding model.

CONCLUSION

From this project we learned to collaborate as a team while individually working on separate task as well as. We also learned how to train a model for that chatbot as well as create and design a user interface for it.

The Data Mine Corporate Partners Symposium
2025