Sprint 0 Planning Document

System Description:

LynxUp Catbot will serve as an invaluable tool for incoming students to become acclimated to their new Rhodes home! The Catbot will serve as an AI guru on all things Rhodes to allow for lightning fast information retrieval and generation.

The system will function like a chat bot with an initial prompt to allow for fixed content selection as well as free-form queries. Fixed content questions will lead the user to Prewritten information as well as useful links to Rhodes webpages for additional support. Think of fixed content as an interactive FAQ page.

The Freeform queries will be directed to a trained AI model that will generate responses based on the information available. Ideally, users should be able to rate the quality of their responses for future improvements.

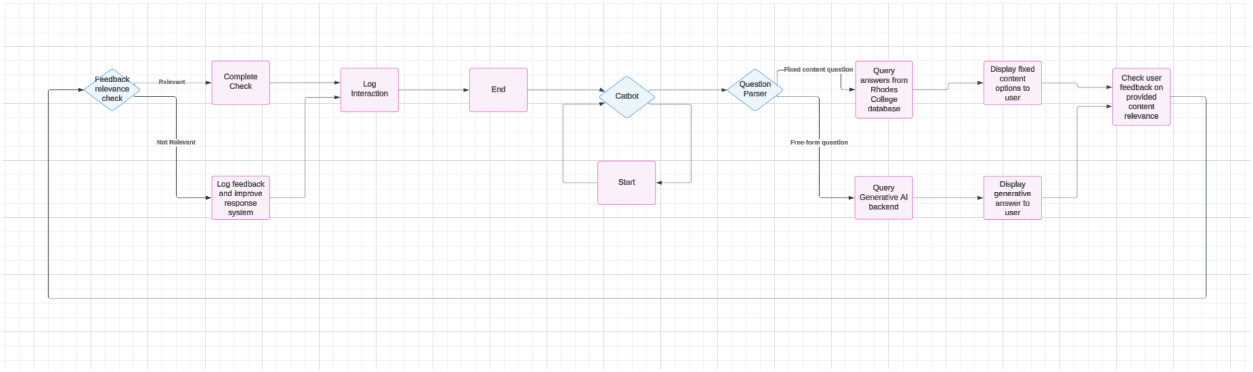
Architectural Overview

Design/architecture description:
From the user standpoint, the user will be able to select from a list of predefined Frequently Asked Questions (FAQs) but also input some free-response prompts that will receive a response upon submission. The feedback look will take the user input and assess the relevance and utility of the responses.

Major system components:

- Database
- User interface
- API between user and backend
- API connection between AI chatbot and backend

System diagram:



Feedback relevance check
Relevant
Not Relevant
Complete Check
Log feedback and improve response system
Log Interaction
End
Catbot
Start
Question Parser
Fixed content question
Free-form question
Query answers from Rhodes College database
Query Generative AI backend
Display fixed content options to user
Display generative answer to user
Check user feedback on provided content relevance

UI, Service Controller, AI Backend

Functional Requirements:

EPIC 1: Interactive User Interface

As a user of Catbot, I want to be able to easily navigate the app and seamlessly prompt the chat feature with various types of prompts so that I can easily get the answers I need without confusion.

EPIC 2: Informative generative AI responses to free form questions

As a user of Catbot, I want to be able to ask any question in my own words and receive a response so that I can get any information I want without having to try and find the right fixed content question.

EPIC 3: Relevant fixed content questions with accurate answers

As a user of Catbot, I want the app to suggest common/relevant questions so that I don't need to type full questions myself and can get an answer quickly and accurately.

EPIC 4: Provide feedback on response relevance and accuracy

As a user of Catbot, I want to be able to tell the app when it's response is irrelevant or inaccurate, so that the model can improve for my future uses and no one else is misinformed.

Non-Functional Requirements:

Reasonably fast response times for both generative and lookup questions.

Able to handle multiple users at the same time.

Technologies and Frameworks:

Primary coding language: Python

Web App front end and API: React, Django, FastAPI

Database: postgreSQL

Generative model: Gemini

Minimum Viable Product (MVP):

The minimum viable product for Catbot would be an accessible user interface with the ability to select from a set of fixed content questions and receive timely and accurate responses from the database. The database would need to be populated with FAQ's most likely to be needed by new students, such as major requirements.

[add references to specific kanban cards]

Semester Roadmap:

Sprint 0 - Feb 6th:

- Quality design document with architecture of project and system and sequence diagrams
- Epics divided and written
- Preliminary user stories added to kanban board
- Preliminary decisions in terms of frameworks and packages to be used

Sprint 1 - Feb 20th

- Basic user interface with functioning API connection and ability to connect to chosen databases structure
- Basic steps made towards database population and generative AI (decisions on HOW to populate and create and train chatbot)

Sprint 2 - March 6th

- Make progress on user interface design and functionality
- Have dataset for AI training created and mostly complete
- Have fixed content question database partially to mostly populated with FAQ information

Sprint 3 - March 20th

- Begin constructing and training AI model
- Continue bug fixes for user interface
- Add to database further and fix any issues that may arise
- MVP complete at end of sprint 3

Sprint 4 - April 3rd

- Finetune and adjust AI model
- Define some test cases for testing web app functionality once we approach project completion
- Further expand user interface for aesthetic appeal (time permitting)

Sprint 5 - April 15th

- Address bugs that have arisen and any issues we have at this point
- Incorporate automatic feedback loop for reinforcement learning in the generative AI model

Sprint 6 - April 29th

- Finalize project and complete any remaining bug fixes
- Prepare symposium presentation