

Florida International University
Knight Foundation School of Computing and Information Sciences

Software Engineering Focus

Final Deliverable

Project Title:

Stars Tutoring Chatbot

Team Members: Richard Chong, Andres Lopez, Michael Camelio, Carlos Ramirez, Jason Martinez

Product Owner(s): Dr. Patricia McDermott-Wells

Mentor(s): Dr. Patricia McDermott-Wells

Instructor: Masoud Sadjadi

The MIT License (MIT)

Copyright (c) 2016 Florida International University

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Abstract

This document provides a comprehensive overview of the Stars Tutoring Chatbot project, detailing its objectives, development progress, and design. It begins with an introduction to the current system and the motivation for the new solution. The document outlines implemented and pending user stories, offering insight into user requirements and progress milestones. It includes a project plan covering hardware and software resources and a detailed sprint schedule. The system's design is elaborated through architectural patterns, subsystem decomposition, deployment diagrams, and design patterns, alongside system validation strategies. Supplementary sections feature UML diagrams, user interface designs, sprint review reports, and user manuals. The document concludes with a glossary, appendices, and references, serving as a complete guide for stakeholders.

Table of Contents

INTRODUCTION	5
CURRENT SYSTEM	5
PURPOSE OF NEW SYSTEM	5
USER STORIES	
IMPLEMENTED USER STORIES	7
PENDING USER STORIES	10
PROJECT PLAN	
HARDWARE AND SOFTWARE RESOURCES	12
SPRINTS PLAN	13
<i>Sprint 1</i>	13
<i>Sprint 2</i>	13
<i>Sprint 3</i>	14
<i>Sprint 4</i>	15
<i>Sprint 5</i>	16
<i>Sprint 6</i>	17
<i>Sprint 7</i>	18
SYSTEM DESIGN	
ARCHITECTURAL PATTERNS	20
SYSTEM AND SUBSYSTEM DECOMPOSITION	21
DEPLOYMENT DIAGRAM	22
DESIGN PATTERNS	22
SYSTEM VALIDATION	23
GLOSSARY	37
APPENDIX	38
APPENDIX A - UML DIAGRAMS	38
<i>Static UML Diagrams</i>	38
<i>Dynamic UML Diagrams</i>	40
APPENDIX B - USER INTERFACE DESIGN	52
APPENDIX C - SPRINT REVIEW REPORTS	69
APPENDIX D - USER MANUALS, INSTALLATION/MAINTENANCE DOCUMENT, SHORTCOMINGS/WISHLIST DOCUMENT AND OTHER DOCUMENTS	74
REFERENCES	80

Introduction

STARS is an organization run by Dr. Patricia McDermott-Wells. Dr. Wells happens to be an Associate Teaching Professor at FIU with an extensive background. STARS Tutoring service gets students who want some additional help when it comes to different CS courses. Currently we are trying to help Dr. Wells created a chatbot that can be used by the STARS students to help assist them with their tutoring needs.

Current System

The current system that we initially started with had its flaws. The main reason that we decided to switch things was because the instructions left behind by the previous team were not to clear. So as a team we struggle to fully understand exactly what the previous team had done. On top of that the previous team had left some documentation but it was specific for certain frontend aspects it had to be modified in a specific way. In the documentation we were told that if we did not use ShadCn to make specific components that the code would break. It also did not help that the current system was not able to run properly because the previous people did not let us know how to install crucial components that we needed.

Purpose of New System

The main intention of ours as we created this new system was to keep all prior functionalities and abilities of the previous groups work, while refreshing it and improving upon it as well as adding in new functionalities to expand the apps usability horizon. The current system was rather complicated and consisted of several different files and several different functions, which complicated the process of debugging as well as improving the system efficiently. Initially, we intended to do our best to keep what was previously existing and to build upon their system but after further review and debate, we decided that the cons outweigh the pros of doing so. We then chose to transfer the entire system over to a Streamlit application, allowing for easier initial set up as well as allowing for faster debugging and condensing of the previously overcomplicated code. We wanted to improve on its capabilities of acting as a tutor chat bot for coding but also begin to set its abilities to have a wider range of scope, such as setting up the groundwork for the addition of other APIs and making it behave more as a general tutor that requires feedback step-by-step.

User Stories

The following section provides the detailed user stories that were implemented in this iteration of the project. These user stories served as the basis for the implementation of the project's features. This section also shows the user stories that are to be considered for future development.

Implemented User Stories

- Story 1: As the team leader I will be uploading documents to the discord server to quickly and efficiently manage our class documents.
 - 5story points, upload documents to discord and manage them.
- Story 2: As a team member I will be working on setting up the interactions between the program and the database
 - 10story points, set up postman to send and request successfully with OpenAI API.
 - 10story point, Ensure correct interaction between the OpenAI API, EC2 and the Database.
- Story 4: As a member of the front end I will be developing the main menu/dashboard
 - 20story points, I will look through streamlit site and docs to find a way to get a working front end that we can use to integrate our model with. (Start with streamlit chat applications and I will also be looking through the mongoDB documentation.
 - 10story points, as a team member of the front end i will implement the database to other parts of the page including the main one and the profile page.

- 10Story points, as a team member I will implement the new STARS GPT website design using Streamlit
- 10Story points, I will continue to enhance existing work to keep everything up to date with what we are doing.
- Story 5: As a team member working on the back end I will set up the imports for future development and call in package format with Langchain and Streamlit.
 - 5story points, I will make it so users can use the import * from package methods.
 - 5story points, I will make it so users can import sets of functions under predefined names for ease of use.
 - 5story points, Will make it so users can directly call functions or classes without needing to use extended relative imports.
- Story 6: As a team member I will assist with finalizing details on the code in multiple areas.
 - 20story points, Work with team members to complete task as needed .
- Story 7: As a team member, I will be working on working on the main chat page and its functions
 - 10story points, Finalize singleton implementation for Groq API
 - 15story points, Fix current bugs/issues and add new functions with chat page (invalid api key, new conversation button and method to do this, etc)
 - 15story points, Set up function to store user conversations within our database

Pending User Stories

- As a team member, I want to implement a profile page for our users to be able to customize their accounts.
- As a team member, I would like to see the web page fully launched and have users interacting with it.
- As a team member, I would like to make this into a full scale app that users can use.
- As a team member, I would like to implement a history feature for the chats.

Project Plan

This section describes the planning that went into the realization of this project. This project incorporated the agile development techniques and as such required the sprints to be planned. These sprint plannings are detailed in the section. This section also describes the components, both software and hardware, chosen for this project.

Development of our project consisted of distribution of tasks amongst each group member, who would then conduct their own research for their specific tasks and goals and work on achieving them. This would allow each of us to focus on a specific part of the project, allowing us each to research, create, implement, and test each part on our own and ensure it meets all the requirements. We would then upload it to our repository and communicate that we had done so and ensure the rest of the team members could access it and use that specific part to build upon and add their new additions as well. This meant the application was receiving new features and updates consistently, allowing us to build upon each others work.

Hardware and Software Resources

The following is a list of all hardware and software resources that were used in this project:

Hardware:

- **Laptop that can run python and access the internet.**

Software

- **Streamlit**
- **LangChain**
- **Groq API**

- **OpenAI API**
- **Python**
- **Mongo DB**

Sprints

Sprint 1

Summary: Set up a Discord server for class document management, review the existing codebase and database, evaluate AWS services for transitioning from Docker, familiarize with React JS for front-end development, and communicate AWS service options to the team through a presentation.

Planning Meeting

Attendees: Richard Chong, Andres Lopez, Michael Camelio, Carlos Ramirez, Jason Martinez

- Start time: 10:00pm
- End time: 11:00pm

We decided to review the code base and build the team's foundational knowledge of the required stack during this sprint, which was agreed upon by the team. Here is the list of user stories and/or tasks ordered based on their priority.

- Story 1: As the team leader I will be creating a discord server to quickly and efficiently manage our class documents.
 - 5 story points, create discord and any documents needed.
- Story 2: As a team member I will review the codebase and database so that I can familiarize myself with what the previous team accomplished.
 - 30 story points, look through code and get familiar with the repository.
- Story 3: As a team member working on the transition from Docker to AWS, I will review all of the services provided by Amazon to find the set that better fits.
 - 20 story points, reviewing AWS documentation.
- Story 4: As the team member working on the front end I will be familiarizing myself with React JS
 - 20 story points, completing quick tutorial projects and watching videos
- Story 5: As a team member I will make sure to communicate the Services provided by AWS, so that all sub-teams in the project are informed and participate in the final decision.
 - 5 story points, creating a compelling presentation and delivering in a quick format.

The team members were willing to work on the following user stories/tasks.

- Richard Chong willing to work on story 1 and 4 story points, story 1 for 5 points and for story 4 for 10 points
- Andres Lopez willing to work on story 2 for 5 story points, Story 3 for 5 points, and Story 4 for 5 points
- Carlos Ramiez willing to work on Story 2 for 15 story points.
- Michael Camelio, willing to work on story 3 for 20 story points, and 5 for 5 story Points.

Sprint 2

Summary: Set up and organize the server with task delegation, update front-end GPT button options, develop and test the back end with structured documentation, and streamline imports for efficient development and package usage.

Planning Meeting

Attendees: Richard Chong, Andres Lopez, Michael Camelio, Carlos Ramirez, Jason Martinez

- Start time: 8pm
- End time: 8:45pm

We decided that we can now start working on the code that was left behind by the other group, which was agreed upon by the rest of the group. Here is the list of user stories and/or tasks ordered based on their priority.

- Story 1: As the team leader I will be uploading documents to the discord server to quickly and efficiently manage our class documents.
 - 5 story points, upload documents to discord and manage them.
- Story 2: As a team member I will be working on setting up the server to run the original program and I will find a way to run the program and sub-assign tasks to other members of the team.
 - 20 story points, Setting up the server and sub-assign tasks.
 - 20 story points, subtasks as needed.
- Story 3: As a member of the front end I will update the GPT buttons “4o” option and 4 options.
 - 20 story points, implementing the GPT buttons the 4.0 and the 4 options.
- Story 4: As a team member working on the back end I will work on setting up the back end for the new AI system.
 - 5 story points, I will create a repo for the package, plan the process structure, and create a

formal report of tasks in order.

- 10 story points, after identifying the model, and other chain components the base model will be completed.

- 10 story points, Will test the base model and will include documentation for usage for team remember testing

- Story 5: As a team member working on the back end I will set up the imports for future development and calling in package format.

- 5 story points, I will make it so users can use the import * from package methods.

- 5 story points, I will make it so users can import sets of functions under predefined names for ease of use.

- 5 story points, Will make it so users can directly call functions or classes without needing to use extended relative imports.

The team members were willing to work on the following user stories/tasks.

- Richard Chong willing to work on story 1, and for story 3
- Andres Lopez willing to work on story 2, 4, and 5
- Carlos Ramiez willing to work on Story 2 and 5.
- Michael Camelio, willing to work on story 3 for 20 story points.
- Jason Martinez, willing to story 5 and 2 for 15 points.

Sprint 3

Summary: Set up and organize the server with task delegation, build the front end using Streamlit components, develop and test the back end with structured documentation, and streamline imports for efficient package usage with Langchain and Streamlit.

Planning Meeting

- Attendees: Richard Chong, Andres Lopez, Michael Camelio, Carlos Ramirez, Jason Martinez
- Start time: 09:15 pm
- End time: 10:30pm

- Story 1: As the team leader I will be uploading documents to the discord server to quickly and efficiently manage our class documents.

- 5 story points, upload documents to discord and manage them.

- Story 2: As a team member I will be working on setting up the server to run the original program and I will find a way to run the program and sub-assign tasks to other members of the team.
 - 20 story points, Setting up the server and sub-assign tasks.
 - 20 story points, subtasks as needed.
- Story 3: As a member of the front end I build the front end using premade streamlit components..
 - 20 story points, implementing the GPT buttons the 4.0 and the 4 options.
- Story 4: As a team member working on the back end I will work on setting up the back end for the new AI system.
 - 5 story points, I will create a repo for the package, plan the process structure, and create a formal report of tasks in order.
 - 10 story points, after identifying the model, and other chain components the base model will be completed.
 - 10 story points, Will test the base model and will include documentation for usage for team member testing
- Story 5: As a team member working on the back end I will set up the imports for future development and calling in package format with Langchain and streamlit.
 - 5 story points, I will make it so users can use the import * from package methods.
 - 5 story points, I will make it so users can import sets of functions under predefined names for ease of use.
 - 5 story points, Will make it so users can directly call functions or classes without needing to use extended relative imports.

The team members were willing to work on the following user stories/tasks.

- Richard Chong willing to work on story 1, and for story 3
- Andres Lopez willing to work on story 2, 4, and 5
- Carlos Ramiez willing to work on Story 2 and 5.
- Michael Camelio, willing to work on story 2 for 20 story points.
- Jason Martinez, willing to story 5 and 2 for 15 points.

Sprint 4

Summary: Set up database interactions and SSL certificates, develop and integrate front-end features like sign-in and chat pages, test and document the AI system's back end, and streamline imports for easier development with Langchain and Streamlit.

Planning Meeting

- Attendees: Richard Chong, Andres Lopez, Michael Camelio, Carlos Ramirez, Jason Martinez
 - Start time: 9pm
 - End time: 10pm
-
- Story 1: As the team leader I will be uploading documents to the discord server to quickly and efficiently manage our class documents.
 - 5 story points, upload documents to discord and manage them.
 - Story 2: As a team member I will be working on setting up the interactions between the program and the database
 - 15 story points, set up database to store user chat history.
 - 10 story point, look into salvaging the old database.
 - Story 2: As a team member I will be working on setting up the SSL Certificates
 - 15 story points, set up SSL Certificates using Let's Encrypt via Certbo.
 - Story 3: As a member of the front end I will be developing the main menu/dashboard
 - 20 story points, I will look through streamlit site and docs to find a working front end that we can use to integrate our model with. (Start with streamlit chat applications.
 - 10 story points, as a team member i will look for front end form submissions (sign in page) that we can use as a template for our sign in
 - 10 Story points, integrate form submission to the chat page.
 - Story 4: As a team member working on the back end I will work on setting up the back end for the new AI system.
 - 10 story points, Will test the base model and will include documentation for usage for team remember testing
 - Story 5: As a team member working on the back end I will set up the imports for future development and calling in package format with Langchain and streamlit.
 - 5 story points, I will make it so users can use the import * from package methods.
 - 5 story points, I will make it so users can import sets of functions under predefined names for ease of use.
 - 5 story points, Will make it so users can directly call functions or classes without needing to use extended relative imports.

The team members were willing to work on the following user stories/tasks.

- Richard Chong willing to work on story 1, and for story 3
- Andres Lopez willing to work on story 2, 4, and 5
- Carlos Ramiez willing to work on Story 2 and 5.
- Michael Camelio, willing to work on story 2 and 3 for 30 story points.
- Jason Martinez, willing to story 5 and 2 for 15 points.

Sprint 5

Summary: Set up database interactions, develop the front end, test and document the back end, streamline imports for easier development, and configure SSL certificates for secure connections.

Planning Meeting

- Attendees: Richard Chong, Andres Lopez, Michael Camelio, Carlos Ramirez, Jason Martinez
 - Start time: 9pm
 - End time: 10pm
-
- Story 1: As the team leader I will be uploading documents to the discord server to quickly and efficiently manage our class documents.
 - 5 story points, upload documents to discord and manage them.
 - Story 2: As a team member I will be working on setting up the interactions between the program and the database
 - 10 story points, set up postman to send and request successfully with OpenAI API.
 - 10 story point, Ensure correct interaction between the OpenAI API, EC2 and the Database.
 - Story 3: As a member of the front end I will be developing the main menu/dashboard
 - 20 story points, I will look through streamlit site and docs to find a working front end that we can use to integrate our model with. (Start with streamlit chat applications.
 - 10 story points, as a team member I will look for the front end. I plan on creating a database temporarily to test out certain login and sign up features.

- 10 Story points, as a member of the front end I plan on adding some more extra features.
- Story 4: As a team member working on the back end I will work on setting up the back end for the new AI system.
 - 10 story points, Will test the base model and will include documentation for usage for team remember testing
- Story 5: As a team member working on the back end I will set up the imports for future development and calling in package format with Langchain and streamlit.
 - 5 story points, I will make it so users can use the import * from package methods.
 - 5 story points, I will make it so users can import sets of functions under predefined names for ease of use.
 - 5 story points, Will make it so users can directly call functions or classes without needing to use extended relative imports.
- Story 6: As a team member I will be working on setting up the SSL Certificates
 - 15 story points, set up SSL Certificates using Let's Encrypt via Certbo.

The team members were willing to work on the following user stories/tasks.

- Richard Chong willing to work on story 1, and for story 3
- Andres Lopez willing to work on story 2, 4, and 5
- Carlos Ramiez willing to work on Story 2,5
- Michael Camelio, willing to work on story 2 and 6.
- Jason Martinez, willing to story 2 and 3 for 15 points.

Sprint 6

Summary: Ensure Streamlit runs smoothly on EC2, develop and enhance the front end, set up and test the AI system's back end, streamline imports, and implement new features and updated classes.

Planning Meeting

- Attendees: Richard Chong, Andres Lopez, Michael Camelio, Carlos Ramirez, Jason Martinez
- Start time: 9pm
- End time: 10pm

- Story 1: As the team leader I will be uploading documents to the discord server to quickly and efficiently manage our class documents.
 - 5 story points, upload documents to discord and manage them.
- Story 2: As a team member I will work on setting up a delay to the EC2 instance so that Streamlit runs properly.
 - 15 story points, Investigating how to implement the delay on the EC2 Instance.
 - 15 story points, Implement the delay on the EC2 instance so Streamlit can run seamlessly.
- Story 3: As a member of the front end I will be developing the main menu/dashboard
 - 20 story points, I will look through streamlit site and docs to find a working front end that we can use to integrate our model with. (Start with streamlit chat applications.
 - 30 story points, as a team member I will look to develop the front end. I plan on continuing to add features to the database and integrating other items to it.
 - 10 Story points, as a member of the front end I plan on adding a profile page for users so then have a personalized experience.
- Story 4: As a team member working on the back end I will work on setting up the back end for the new AI system.
 - 10 story points, Will integrate back end to the front end.
 - 5 Story points, will test the system and inform dr wells wells that system is ready for testing
- Story 5: As a team member working on the back end I will set up the imports for future development and calling in package format with Langchain and streamlit.
 - 5 story points, I will make it so users can use the import * from package methods.
 - 5 story points, I will make it so users can import sets of functions under predefined names for ease of use.
 - 5 story points, Will make it so users can directly call functions or classes without needing to use extended relative imports.
- Story 6: As a team member working on adding new functionalities and upgrading current ones, I will work on implementing new functions and features
 - 10 story points, Finalize and implement updated singleton class to ensure one instance
 - 15 story points, Begin implementing image generator from open source generator
 - 10 story points, Implement function to ensure that a different image is produced every time regardless whether the input is the same.

The team members were willing to work on the following user stories/tasks.

- Richard Chong willing to work on story 1, and for story 3
- Andres Lopez willing to work on story 2, 4, and 5

- Carlos Ramiez willing to work on Story 3 and 6.
- Michael Camelio, willing to work on story 2
- Jason Martinez, willing to work on story 3

Sprint 7

Summary: Manage class documents, create a project poster, develop the front end, set up and integrate the back end, streamline imports, and add new functionalities for the system.

Planning Meeting

- Attendees: Richard Chong, Andres Lopez, Michael Camelio, Carlos Ramirez, Jason Martinez
 - Start time: 9pm
 - End time: 10pm
-
- Story 1: As the team leader I will be uploading documents to the discord server to quickly and efficiently manage our class documents.
 - 5 story points, upload documents to discord and manage them.
 - Story 2: As a team member I will work on creating a poster to be submitted representing my contributions to the project:
 - 30 story points, design a poster with personal contributions to the project.
 - Story 3: As a member of the front end I will be developing the main menu/dashboard
 - 20 story points, I will look through streamlit site and docs to find a working front end that we can use to integrate our model with. (Start with streamlit chat applications.
 - 30 story points, as a team member I will look to develop the front end. I plan on continuing to add features to the database and integrating other items to it.
 - 10 Story points, as a member of the front end I plan on adding a profile page for users so then have a personalized experience.
 - Story 4: As a team member working on the back end I will work on setting up the back end for the new AI system.
 - 10 story points, Will integrate back end to the front end.
 - 5 Story points, will test the system and inform dr wells wells that system is ready for testing

- Story 5: As a team member working on the back end I will set up the imports for future development and calling in package format with Langchain and streamlit.
 - 5 story points, I will make it so users can use the import * from package methods.
 - 5 story points, I will make it so users can import sets of functions under predefined names for ease of use.
 - 5 story points, Will make it so users can directly call functions or classes without needing to use extended relative imports.
- Story 6: As a team member working on adding new functionalities and upgrading current ones, I will work on implementing new functions and features
 - 20 story points, Implement new conversation button and functions
 - 20 story points, Implement logic to view previously stored user conversations

The team members were willing to work on the following user stories/tasks.

- Richard Chong willing to work on story 1, and for story 3, 2
- Andres Lopez willing to work on story 2, 4, and 5
- Carlos Ramiez willing to work on Story 6.
- Michael Camelio, willing to work on story 2
- Jason Martinez, willing to work on story 3 and 2

System Design

This section contains information on the design decisions that went into this project. The architecture patterns are outlined and explained. The entire system is shown in a package diagram and the subsystems are explained. Finally, the design patterns used in the project are discussed.

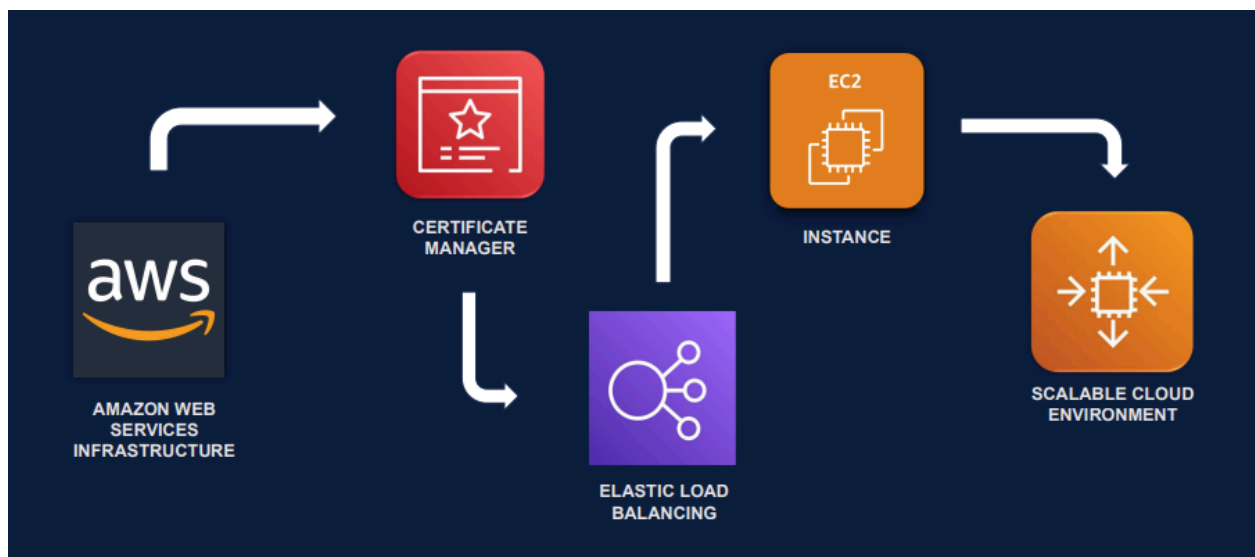
Architectural Patterns

For the Architectural pattern it will be flowing from the AWS server to, Streamlit and from streamlit it will go to the database. From there you will access the Openai api key.

System and Subsystem Decomposition

1. The start is at the login screen. Where the user has two choices: the login button and the sign up button.
 - For the Login we needed to create a login form.
 - We also needed a Signup form.
 - All that information needed to be sent to the database as soon as they hit enter.
 - Also, we need to confirm that the username and password in sign is a valid FIU email and the password meets the criteria.
2. In the main page you have the Tutor bot and the Code bot.
 - From there if you choose the code bot you just have to interact with the bot and ask it a question.
 - It will generate answers based on the question given to the bot.
 - This applies to both bots.
 - Each bot has a different functionality

Deployment Diagram



Design Patterns

For the design pattern we ended up deciding to go with a minimalist pattern. This keeps things nice and clean while also looking nice.

System Validation

We employed several different types of test case scenarios throughout the process of creating our application to ensure that we covered as many issues that could occur as possible. This ranged from load times when launching the application to testing input validation on parts that could take user input, or even strange interactions that would initially occur if the user decided to swap between the OpenAI and Groq models. We each would test our implementations separately, to first ensure it is doing everything it should do as well as preventing things that shouldn't happen from occurring, such as testing that the chain object for the chat wouldn't be created when an improper API key was entered. This method of testing made the combining implementation of each of our work easier, as we each had already gone through unit testing, meaning if any issues arise it was due to the way the code was interacting with one another.

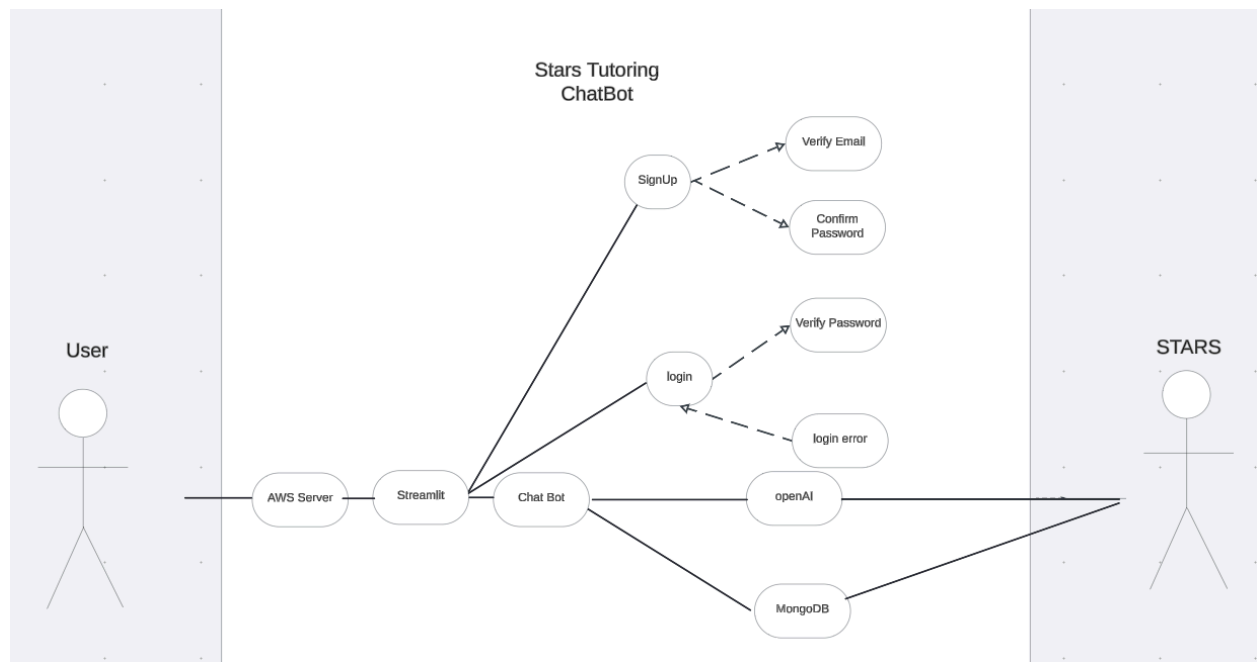
Glossary

API - Application Programming Interface.

LLM - Large Language Model.

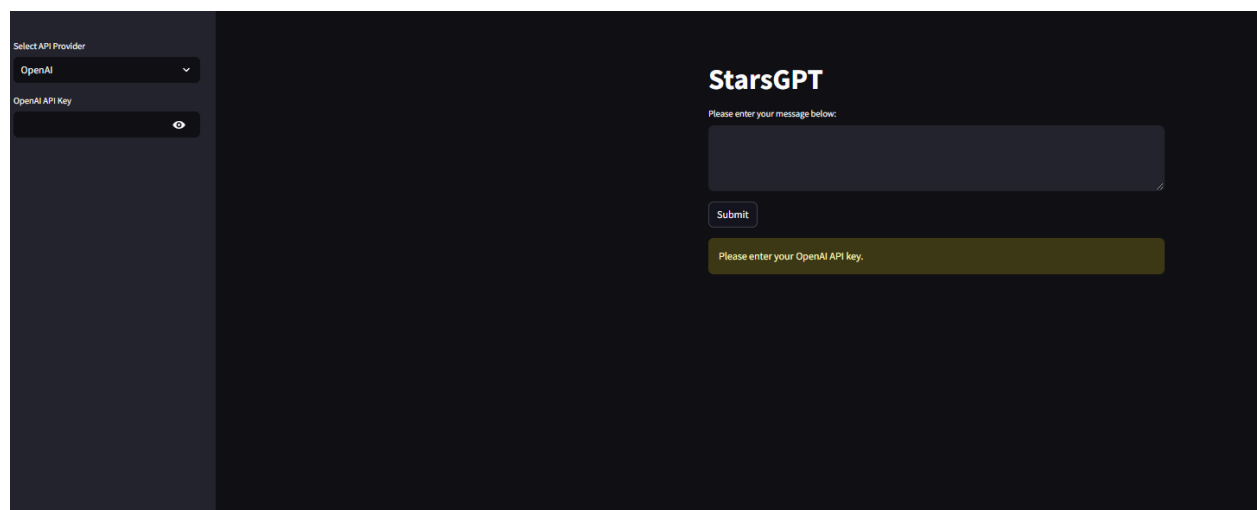
Appendix

Appendix A - UML Diagrams

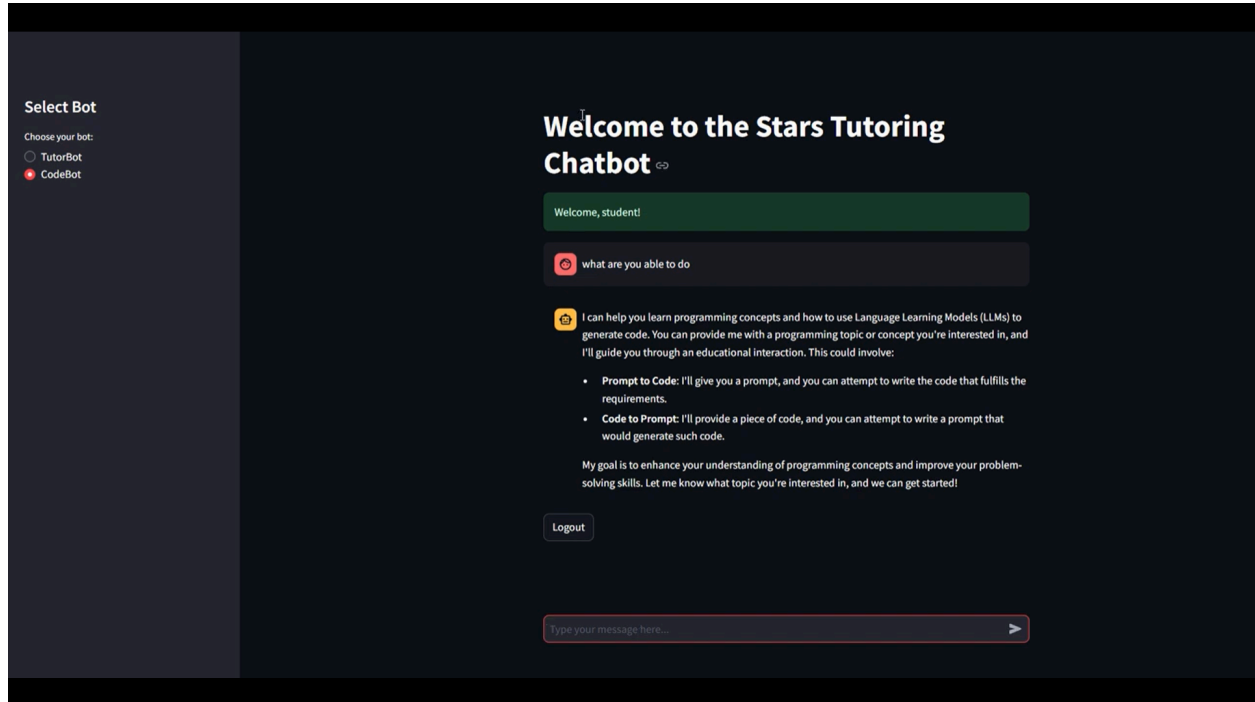


Appendix B - User Interface Design

We chose to maintain a modern and simplistic design, allowing the entire focus of the application to be kept on the user's interaction with the chat bot.



Earlier version of application which asked user to choose between either Groq or OpenAI and to enter their API key.



This is the final version of the frontend user interface. Which allows users interact with the chat bot and ask questions.

Appendix C - Sprint Review Reports

Sprint 1:

- Attendees: Richard Chong, Andres Lopez, Michael Camelio, Carlos Ramirez, Jason Martinez
- Start time: 8pm
- End time: 8:45pm

After a show-and-tell presentation, we consolidated our findings as follows. ○ Features/functions that were tested thoroughly and worked as expected without any errors/issues.

- Currently we are all trying to review and catch up with the code that was left behind for us.

○ Features/functions that were tested thoroughly and did NOT work as expected. Here is the list of issues/errors observed.

- No inconvenience has been relevant since the original code continues to be reviewed by the team.

○ These are features/functions that were tested thoroughly and would benefit from some improvement/enhancement. Here is the list of suggested enhancements/improvements.

■ None yet.

Sprint 2:

- Attendees: Richard Chong, Andres Lopez, Michael Camelio, Carlos Ramirez, Jason Martinez
- Start time: 9pm
- End time: 9:45pm

After a show-and-tell presentation, we consolidated our findings as follows.

○ Features/functions that were tested thoroughly and worked as expected without any errors/issues.

■ Currently the team is testing the backend and should be working fine just needs to create an environment for it.

○ Features/functions that were tested thoroughly and did NOT work as expected. Here is the list of issues/errors observed.

■ There were issues with the previous database serving architect preventing testing.

○ These are features/functions that were tested thoroughly and would benefit from some improvement/enhancement. Here is the list of suggested enhancements/improvements.

■ None yet.

Sprint 3:

Sprint Review Meeting Minutes

- Project: STARGPT

- Attendees: Richard Chong, Andres Lopez, Michael Camelio, Carlos Ramirez, Jason Martinez

- Start time: 9pm

- End time: 9:45pm

After a show-and-tell presentation, we consolidated our findings as follows.

- Features/functions that were tested thoroughly and worked as expected without any errors/issues.
 - Currently the team is testing the backend and should be working fine just needs to create an environment for it.
 - Created a login page as well as a chat page where users can enter their API key for OpenAI or Groq and chat with them. Currently not connected to our backend but there is a dummy instance backend that does store the current input and output text.
- Features/functions that were tested thoroughly and did NOT work as expected. Here is the list of issues/errors observed.
 - There were issues with the previous database serving architect preventing testing.
 - Issue with EC2 instance key pair
 - Issue with endpoints preventing testing of the API with AWS database.
- These are features/functions that were tested thoroughly and would benefit from some improvement/enhancement. Here is the list of suggested enhancements/improvements.
 - None yet

Sprint 4:

- Attendees: Richard Chong, Andres Lopez, Michael Camelio, Carlos Ramirez, Jason Martinez
- Start time: 9pm
- End time: 9:45pm

After a show-and-tell presentation, we consolidated our findings as follows.

- Features/functions that were tested thoroughly and worked as expected without any errors/issues.
 - Currently the team is testing the backend and should be working fine just needs to create an environment for it.
 - Created a login page as well as a chat page where users can enter their API key for OpenAI or Groq and chat with them. Currently not connected to our backend but there is a dummy instance backend that does store the current input and output text.
- Features/functions that were tested thoroughly and did NOT work as expected. Here is the list of issues/errors observed.
 - There were issues with the previous database serving architect preventing testing.

- Issue with implementing the previous singleton method, has to be updated and adapted to work properly with our current model.
- Issues with uploading the AI to pypi due to some unforeseen errors.
- These are features/functions that were tested thoroughly and would benefit from some improvement/enhancement. Here is the list of suggested enhancements/improvements.
 - Login page
 - Main Page
 - AI chat box

Sprint 5:

- Attendees: Richard Chong, Andres Lopez, Michael Camelio, Carlos Ramirez, Jason Martinez
- Start time: 9pm
- End time: 9:45pm

After a show-and-tell presentation, we consolidated our findings as follows.

- Features/functions that were tested thoroughly and worked as expected without any errors/issues.
 - Currently the team is testing the backend and should be working fine just needs to create an environment for it.
 - Created a login page as well as a chat page where users can enter their API key for OpenAI or Groq and chat with them. Currently not connected to our backend but there is a dummy instance backend that does store the current input and output text.
 - Currently setting up the mongoDB database and going to be testing soon to make sure it functions properly.
 - Streamlit deployable version of API implemented chat has begun testing. Needs to be integrated into AWS still.
- Features/functions that were tested thoroughly and did NOT work as expected. Here is the list of issues/errors observed.
 - There were issues with the previous database serving architect preventing testing.
 - Issue with EC2 instance key pair
 - Issue with endpoints preventing testing of the API with AWS database.

- Issue with implementing the previous singleton method, has to be updated and adapted to work properly with our current model.
- Tried deploying streamlit deploy singularly and function worked, attempting AWS deployment did not work as expected.
- These are features/functions that were tested thoroughly and would benefit from some improvement/enhancement. Here is the list of suggested enhancements/improvements.
 - Login page
 - Main Page
 - AI chat box

Sprint 6:

- Attendees: Richard Chong, Andres Lopez, Michael Camelio, Carlos Ramirez, Jason Martinez
- Start time: 9pm
- End time: 9:45pm After a show-and-tell presentation, we consolidated our findings as follows.
- Features/functions that were tested thoroughly and worked as expected without any errors/issues.
 - Currently the team is testing the backend and should be working fine just needs to create an environment for it.
 - Created a login page as well as a chat page where users can enter their API key for OpenAI or Groq and chat with them. Currently not connected to our backend but there is a dummy instance backend that does store the current input and output text.
 - Currently finished setting up the database and works super smoothly with no issues.
 - Streamlit deployable version of API implemented chat has begun testing. Needs to be integrated into AWS still.
 - Currently integrating chat memory for the LLM api to the database without intermediary questions in the context, working thus far without complications(this is the contextual memory that includes decision steps and not the “chat memory”) output to the user
 - The profile page has been fully tested and functions properly now.
 - The login page is fully functional now and has been tested.
 - Chat page allows users to flawlessly swap in between Groq and OpenAI models, storing each respective conversation independently.

- Chat page has API key validation, ensuring that the user's input is tested first before attempting to create a chain instance.
- Features/functions that were tested thoroughly and did NOT work as expected. Here is the list of issues/errors observed.
 - There were issues with the previous database serving architect preventing testing.
 - Issue with EC2 instance key pair
 - Issue with endpoints preventing testing of the API with AWS database.
 - Issue with implementing the previous singleton method, has to be updated and adapted to work properly with our current model.
 - Tried deploying streamlit deploy singularly and function worked, attempting AWS deployment did not work as expected.
 - At first some of the updating current information from the database was an issue but now it has been resolved.
- These are features/functions that were tested thoroughly and would benefit from some improvement/enhancement. Here is the list of suggested enhancements/improvements.
 - Login page
 - Main Page
 - AI chat box
 - Profile Page

Sprint 7:

- Project: STARGPT
- Attendees: Richard Chong, Andres Lopez, Michael Camelio, Carlos Ramirez, Jason Martinez
- Start time: 9pm
- End time: 9:45pm

After a show-and-tell presentation, we consolidated our findings as follows.

- Features/functions that were tested thoroughly and worked as expected without any errors/issues.
 - Currently the team is testing the backend and should be working fine just needs to create an environment for it.
 - Created a login page as well as a chat page where users can enter their API key for OpenAI or Groq and chat with them. Currently not connected to our backend but there is a dummy instance backend that does store the current input and output text.
 - Currently finished setting up the database and works super smoothly with no issues.
 - Streamlit deployable version of API implemented chat has begun testing. Needs to be integrated into AWS still.
 - Currently integrating chat memory for the LLM api to the database without intermediary questions in the context, working thus far without complications(this is the contextual memory that includes decision steps and not the “chat memory”) output to the user
 - The profile page has been fully tested and functions properly now.
 - The login page is fully functional now and has been tested.
 - Chat page allows users to flawlessly swap in between Groq and OpenAI models, storing each respective conversation independently.
 - Chat page has API key validation, ensuring that the user’s input is tested first before attempting to create a chain instance.
- Features/functions that were tested thoroughly and did NOT work as expected. Here is the list of issues/errors observed.
 - There were issues with the previous database serving architect preventing testing.
 - Issue with EC2 instance key pair
 - Issue with endpoints preventing testing of the API with AWS database.
 - Issue with implementing the previous singleton method, has to be updated and adapted to work properly with our current model.
 - Tried deploying streamlit deploy singularly and function worked, attempting AWS deployment did not work as expected.
 - At first some of the updating current information from the database was an issue but now it has been resolved.
- These are features/functions that were tested thoroughly and would benefit from some improvement/enhancement. Here is the list of suggested enhancements/improvements.
 - Login page
 - Main Page
 - AI chat box
 - Profile Page

Appendix D - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents

User Manual:

<https://docs.google.com/document/d/1yu1O6kDB6aBBgZYvlHISn6Xae6YiWpnY/edit>

Installation/Maintenance Documentation:

<https://docs.google.com/document/d/1PQPb3gfOzjALl2DWzAEqewoCcz2Qojz8/edit>

References

Streamlit Documentation: <https://docs.streamlit.io/>

MongoDB Documentation: <https://www.mongodb.com/docs/>