VCU College of Engineering

CS 25-349:
AI-Powered Email Response System Using Fine-Tuned LLMs for Customer Service in React

Project Proposal

Prepared for

Keroles Hakem, CoStar Group

By

Cameron Clyde, Emma Smith,

Angela Harris, Sohil Marreddi

Under the supervision of

Preetam Ghosh

10/11/2024

Executive Summary

As CoStar Group continues to grow, the demands on its customer service teams increase significantly. CoStar Group's internal customer relationship management system is known as Case Management. Case Management employees are responsible for handling complex cases and responding to customer emails. Balancing these responsibilities often results in delayed responses and inconsistent communication quality, as employees struggle to maintain both professionalism and personalization in their emails. Furthermore, replying to emails takes away time employees have to deal with their cases.

These challenges highlight the need for a solution that streamlines the email process, allowing employees to respond quickly and efficiently without compromising the quality of service. This project will leverage Large Language Models (LLMs) to assist in generating quality email responses for Case Management Employees. The overall aim of this project is to create a web application with a fine-tuned LLM on specific customer service data to generate relevant and context-aware email responses. The project goals include implementing a feedback loop to continuously improve the model's performance, allowing employees to review and edit generated drafts before sending them to customers and reducing the time required to respond to customer emails.

The design specifications and constraints outline a system that facilitates the generation of professional email responses for open customer service cases, integrating the React library for frontend development and utilizing FastAPI and PostgreSQL for backend functionality. The key requirements that will be implemented in this design include a feedback loop for AI suggestions, JWT-based authentication, AI response templates that will be customizable for the user, storage management for data related to case assignment logic, AI-generated responses, and representative feedback, and continuous integration and deployment pipelines for automated testing and updates. A successful design will effectively reduce the workload for customer service representatives, handle an increasing number of customer service cases and interactions, minimize customer service response times, and enhance customer satisfaction. In the creation of this design, we will adhere to a budget of $1000, comply with data privacy regulations, and take necessary security measures against vulnerabilities that can present themselves.

In order to meet the necessary design requirements, it's important that we are always applying a set of codes and standards that will ensure quality, reliability, and safety. In this document, a number of codes and standards relating to security, privacy, software development, quality, ethics, design, etc. are given that can be directly applied to our design. Two codes relating to security and privacy that we will be adhering to in our design are the GDPR (General Data Protection Regulation) and CCPA (California Consumer Privacy Act). We will also be applying multiple standards, including ISO/IEC 27701, relating to information security management, ISO/IEC 25010, which outlines the software quality requirements and evaluation, OpenAPI, the standard for API documentation, ISO/IEC TR 24027, the guidelines for evaluating AI-based systems, Material Design Guidelines, which describes the best practices for UI design using Material UI components, etc. With the usage of these codes and standards, we can be sure to create the most efficient and secure design.

The scope of the project is clearly defined to ensure all objectives are met on time and within budget. The web application, CASEflow, will be developed using an Agile methodology with sprints spanning 2-3 weeks each. Deliverables include a fully functional web application that allows employees to log in, access customer emails, and generate AI-assisted responses, fine-tuned on CoStar's customer service data. Key features include an interface for employees to review and edit AI-generated responses, a feedback loop to improve the AI's performance over time, and detailed metrics for response times and model accuracy improvements. Academic deliverables include the project proposal, reports, and presentations required for the Capstone EXPO.

The project will utilize resources such as GitHub for version control, React for frontend development, FastAPI for backend operations, and PostgreSQL for database management. The large language model (LLM) will be fine-tuned using AWS SageMaker, which will be provided by CoStar. The team will focus on continuous integration to ensure a smooth development process, with testing and feedback incorporated at each sprint to avoid scope creep and ensure all deliverables are met. The application aims to streamline email response times, reduce employee workload, and improve the quality and consistency of customer interactions.

Table of Contents

Section A. Problem Statement

CoStar Group is the global leader in the digitalization of commercial real estate. Since their start in 1986, they have expanded into the world of residential real estate and become a leader in this industry. Their mission is to digitize the world's real estate and make it easier for people and companies to discover properties. They provide real estate information, analytics, and online marketplaces (CoStar Group, 2024). In their mission to digitize real estate around the world, CoStar Group continues to grow every year. CoStar Group acquires new brands and companies to help achieve this mission. CoStar Group has over 25 brands, including Homes.com and Apartments.com.

Merging these businesses into CoStar Group can be difficult. Every company that CoStar Group acquires does their business differently. In particular, most companies have their own subscriptions with different Customer Relationship Management (CRM) systems. CRM systems are a strategic tool that integrate technology, customer knowledge and relationships to enhance business efficiency, foster customer loyalty, and drive economic growth (Gil-Gomez et al., 2020). It is evident that CRMs are essential, but CoStar Group does not want to deal with multiple different CRMs and pay fees for each CRM. To solve this, they decided to run it all internally and develop their own CRM, known as Case Management.

In CoStar's existing customer service response process, representatives begin by reading the incoming email along with the associated email history to fully understand the customer's issue. After reviewing the context, they select an appropriate response template from a set of predefined options that best aligns with the customer's inquiry or concern. The representative then customizes the chosen template by filling in relevant details specific to the case, such as the customer's name, case number, and any additional information that addresses the issue.

As the amount of cases increases, the current process begins to show limitations. This process becomes time consuming, labor intensive, and prone to human error as representatives are required to manually fill each template (Mesquita et al., 2022). The reliance on templates can result in repetitive and impersonal responses, which may not fully address each customer's issue. This adds additional time and effort as representatives must write their own reply. While technologies have been proposed to automatically query and select customer data to fill response templates, they are not completely accurate and still fail to address customer issues that are not covered by templates (Malik et al,. 2007). As case volumes rise, it becomes increasingly difficult for employees to manage the workload effectively, leading to delayed responses, inconsistencies in communication, and reduced overall quality of customer service (Sheth et al., 2024). This issue creates the need for a more scalable solution that can maintain personalization without sacrificing speed or accuracy.

Our project aims to improve and streamline this process by developing an email response system that utilizes a fine-tuned large language model (LLM) to generate personalized responses. This system will analyze the incoming email and its history to understand the customer's issue, then automatically create a response without relying on premade templates. The LLM will use the context of the email history to create responses more tailored to the specific case, while automatically filling in relevant customer data, which increases both the speed and quality of customer service interactions.

Section B. Engineering Design Requirements

B.1 Project Goals (i.e. Client Needs)

This project aims to assist customer service employees at CoStar Group with the high-volume of emails received. The demands for customer service employees grow as CoStar Group acquires new businesses. It can be difficult for employees to manage their existing work on top of the many emails they receive. Responding to emails in a personalized and professional way is time consuming. When trying to balance work on existing complex cases, writing these emails can lead to inconsistencies. With this in mind, the project goals are as follows:

- To design an app that reduces the time required to respond to customer emails
- To enable employees to focus on high priority work by automating the generation of strong email replies
- To improve the personalization and professionalism of email responses from customer service employees
- To allow employees to review and edit generated drafts, ensuring accuracy and appropriateness
- To enhance the AI model through employee feedback to continuously improve the model's performance

B.2 Design Objectives

The following are key objectives of the design:

- The design will reduce email response times for employees by generating draft replies. This will be measurable through employee feedback and satisfaction, with a target of reducing response time by 30%.
- The design will create a web app, with an authentication system to ensure only CoStar Group Case representatives can access this information. This structure will be completed halfway through the fall semester, with testing to ensure functionality as expected.
- The design will fine-tune a Large Language Model (LLM) using customer service data to generate relevant, context-aware email responses. The speed and accuracy will be measured before fine tuning. After fine-tuning, the speed and accuracy should improve by 20%.
- The design will implement a feedback loop where employees can rate the email responses produced by the model to improve the model. This will be measured by the model's ability to learn from feedback.
- The design will include a user interface allowing employees to review and edit responses for accuracy and appropriateness. This will be measurable by employee usage and edit rates.

B.3 Design Specifications and Constraints

Design Specifications

Functional Requirements

- The system must be powered by fine-tuned LLMs
- Design must facilitate the generation of professional email responses for open customer service cases
- The system must integrate the React library for frontend development
- Design must integrate with existing customer service platforms
- Design must implement a feedback loop, allowing employees to rate AI-generated suggestions
- Design must integrate FastAPI and PostgreSQL for backend functionality
- Design must implement JWT-based authentication for user login and signup on the backend
- Design must implement login and signup pages on the frontend
- Design must be able to store data related to case assignment logic, AI-generated responses, and representative feedback on those responses
- Design must allow for customization of the AI response templates
- Design must follow *IEEE P7003 Algorithmic Bias Considerations* to ensure that AI-generated responses are free from bias and are transparent to both employees and customers
- The design must include unit tests, integration tests, and automated testing pipelines to follow ISO/IEC/IEEE 29119 software testing standards
- Design must implement continuous integration and deployment pipelines for automated testing and seamless updates (CI/CD best practices)

Non-Functional Requirements

- Design must effectively reduce the workload for customer service representatives
- The system should be designed to handle an increasing number of customer service cases and employee interactions
- The AI model must generate responses quickly to ensure minimal delays in customer service workflow
- The design must follow *ISO/IEC 25010* to ensure software quality, including functionality, usability, and maintainability
- The design must adhere to *WCAG 2.1 guidelines* to ensure that the interface is accessible to users with disabilities
- AI model must undergo regular evaluation to ensure that its suggestions are accurate, relevant, and helpful
- Design should aim to minimize response customer service response times
- Design must enhance customer satisfaction

Design Constraints

Cost

- Design must adhere to a budget of $1000

Security and Data Privacy

- All user data must comply with data privacy regulations
- Design must implement data retention policies that comply with *GDPR/CCPA* standards to ensure user data is only stored for the required period
- Design must comply with *GDPR* guidelines, ensuring that user data is protected and the necessary opt-ins and transparency measures are in place
- Design must ensure protections against the top 10 web application vulnerabilities, including secure authentication, access control, and input validation (OWASP Top Ten)
- The system must follow standards for managing personally identifiable information (ISO/IEC 27701 Compliance)
- Design must securely store JWTs within the system

Quality Assurance and Testing

- Design must ensure that tests cover security vulnerabilities and performance bottlenecks, which limit the system's performance (ISTQB guidelines)
- Design must include robust error handling

B.4 Codes and Standards

Codes:

Security and Privacy

- GDPR (General Data Protection Regulation): Data protection and privacy law for EU residents
- CCPA (California Consumer Privacy Act): Privacy law for California residents, focusing on data transparency and user control over personal data

Standards:

Security

- OWASP Top Ten: Guidelines for mitigating common web vulnerabilities
- NIST Cybersecurity Framework: Best practices for managing cybersecurity risks
- ISO/IEC 27701: Information security management systems standard
- JWT Best Practices: Guidelines for secure handling of JSON Web Tokens
- CIS Benchmarks (Center for Internet Security): Best practices for cloud infrastructure security

Privacy

- ISO/IEC 27701: Privacy extension to ISO 27001, focusing on managing personal data (PII)

Software Development Standards

- ISO/IEC 25010 (Software Quality Requirements and Evaluation): Software quality metrics, such as usability and reliability
- IEEE 12207: Framework for software development lifecycle (SDLC)
- SOLID Principles: Object-oriented design principles for maintainability
- Agile Development Framework: Practices for iterative, adaptive development

API and Web Standards

- RESTful API Best Practices: Guidelines for designing RESTful APIs (proper use of HTTP methods, statelessness)
- OpenAPI: Standard for API documentation
- W3C Web Standards: Accessibility, responsive design, and cross-browser compatibility
- WCAG 2.1 (Web Content Accessibility Guidelines): Guidelines for designing accessible web content

AI Ethics and Guidelines

- ISO/IEC TR 24027: Guidelines for evaluating AI-based systems
- IEEE P7003 Algorithmic Bias Considerations: Recommendations for avoiding bias in AI systems

Quality Assurance Standards

- ISTQB (International Software Testing Qualifications Board): Best practices for software testing
- ISO/IEC/IEEE 29119: Software testing standards for comprehensive testing
- CI/CD Best Practices: Guidelines for automate testing, deployment, and version control

Frontend UI/UX Design Standards

- Material Design Guidelines: Best practices for UI design using Material UI components
- User-Centered Design (UCD): Focus on user needs during the design process (ISO 9241-210)

Database Standards

- GDPR/CCPA Data Retention Policies: Clear policies for data storage and retention, especially regarding personal and sensitive data

Section C. Scope of Work

C.1 Deliverables

1. A Web Application named CASEflow, a secure web application which allows users to login to access protected data(emails), and get an AI generated email response which improves on the Large Language Model we Finetune.

    1.1 Given real company data (data cleaned for privacy), we are to provide Fine Tuning to the existing Large Language Model to improve speed and performance of the AI response.

    1.2 A Feature for employees to rate AI-generated suggestions, improving the LLM over time.

    1.3 Functionality for employees to review, edit, and send the AI-generated email responses using the application.

    1.4 Metrics and reports on response times and AI accuracy improvements using statistics from user feedback(this would be done through testing as we Fine-Tune the LLM).

2. Provide technical documentation, a user manual, and build steps for running the application CASEflow.

3. Academic Deliverables

    3.1 Team Contract
    3.2 Project Proposal
    3.3 Preliminary Design Report
    3.4 Fall Poster and presentation
    3.5 Final design report
    3.6 Capstone EXPO poster and presentation

Some important issues to discuss with the design team, sponsor, and faculty advisor include the following:

The only third party vendor we might be unable to get access to is Amazon Web Services accounts. In our current plan of action we need an AWS account to work on the large language model which is a core feature of our application CASEflow.

Method of deployment, issues with our deployment, whether that our application is failing to build, or not enough testing being done because everything is being run on our local environment until deployment.

All deliverables can be worked on remotely; however, due to a large volume of remote work the group has multiple means of communication including Discord, Slack, email, and Zoom.

Meeting Minutes and other information is all organized in shared google drives for effective remote work.

C.2 Milestones

Our team will be utilizing Agile Development with a focus on Continuous Integration. The project will be divided into distinct phases, or 'sprints,' each spanning 2-3 weeks, depending on the complexity of the tasks. The length and content of each sprint will be collaboratively decided by the student team, project mentor, and faculty advisor. The deliverables from each sprint will serve as milestones to track progress and ensure that the final project objectives are completed on time.

Sprints:

Sprint 1: Set up basic project structure Date: 9/9/24 - 9/23/24

- Set up folder structure for Frontend(react), Backend(FastAPi), and connect database(PostgreSQL).

Sprint 2: Authentication Date: 9/23/24 - 10/7/24

- Complete Sign in and Log in page UI(react)
- Create JWT tokens to enable secure logins, Connect frontend and backend for testing.

Sprint 3: Case Management & Representative Schema Date: 10/7/24 - 10/21/24

- Create routes for Different pages
- Create Casetable(frontend)
- Create database models for Casetable(backend)
- Create API Endpoints for retrieval and entry of the CaseTable information.
- Create Form for users to submit new cases, and a dashboard to view assigned cases(frontend)

Milestone: Completion of basic project design Date: 10/21/24

Sprint 4 (Weeks 8-10): Connecting LLM Date: 10/21/24 - 11/4/24

- Connect LLM to backend VIA AWS Sagemaker (pending aws account approval)
- Test all components of application before fine tuning next sprint

Sprint 5 - 8: LLM Fine-Tuning Date: 11/4/24 - 12/16/24

Milestone: Completion of LLM Fine Tuning Date: 12/16/24

Sprint 9 - 10: LLM Feedback Date: 1/13/25 - 2/10/25

- LLM Feedback and metrics report testing.

Sprint 11 (Weeks 19-20): Testing Date: 2/10/25 - 2/24/25

- Testing and Documentation write up

Milestone: Completion of testing	**Date: 2/24/25**
Sprint 12: Bug Fixes	Date: 2/24/25 - 3/10/25
Sprint 13: Deployment	Date: 3/10/25 - 3/24/25
Milestone: Completion of project	**Date: 3/24/25**

Academic Milestones:

CMSC 451 Deliverables	
Team Contract	Date: 9/4/24
Project Proposal	Date: 10/11/24
Preliminary Design Report	Date: 11/15/24
Fall Design Poster and Presentation	Date: 12/12/24
CMSC 452 Deliverables	
Final Design Report	Date: 2/23/25
Capstone EXPO Abstract	Date: 3/5/25
Capstone EXPO Poster	Date: 4/18/25
Capstone EXPO Presentation	Date: 4/25/25

C.3 Resources

Paid Resources

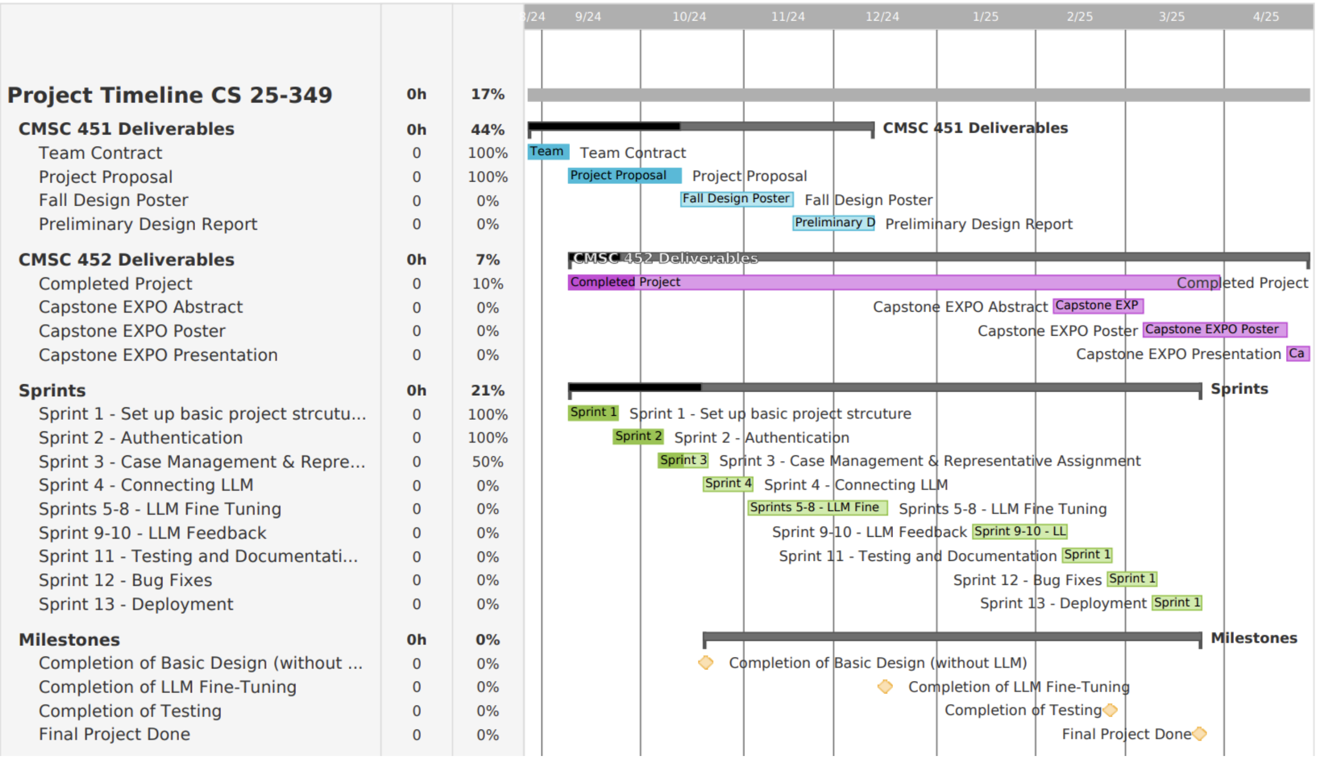
Access to Amazon web service account, needed to utilize AWS SageMaker for LLM fine-tuning and machine learning aspect. These accounts will be provided by Project Sponsor (CoStar Group).

Free Resources

We will be using GitHub for our Version Control System, which is accessible to all VCU students. Programming language, frameworks and technologies are all available to open source download. We will be using React, FastAPI, PostgreSQL, SwaggerUI and Visual Studio Code. For Communication we are using Slack, Discord, Zoom, and Email.

Appendix 1: Project Timeline

The figure below illustrates the timeline of our project in Gantt Chart form. The categories of tasks are as follows: CMSC 451 Deliverables, CMSC 452 Deliverables, Sprints, and Project Milestones. The sprints section will change as the team is assigned sprints.



Project Timeline CS 25-349	**0h**	**17%**
CMSC 451 Deliverables	**0h**	**44%**
Team Contract	0	100%
Project Proposal	0	100%
Fall Design Poster	0	0%
Preliminary Design Report	0	0%
CMSC 452 Deliverables	**0h**	**7%**
Completed Project	0	10%
Capstone EXPO Abstract	0	0%
Capstone EXPO Poster	0	0%
Capstone EXPO Presentation	0	0%
Sprints	**0h**	**21%**
Sprint 1 - Set up basic project strcutu...	0	100%
Sprint 2 - Authentication	0	100%
Sprint 3 - Case Management & Repre...	0	50%
Sprint 4 - Connecting LLM	0	0%
Sprints 5-8 - LLM Fine Tuning	0	0%
Sprint 9-10 - LLM Feedback	0	0%
Sprint 11 - Testing and Documentati...	0	0%
Sprint 12 - Bug Fixes	0	0%
Sprint 13 - Deployment	0	0%
Milestones	**0h**	**0%**
Completion of Basic Design (without ...	0	0%
Completion of LLM Fine-Tuning	0	0%
Completion of Testing	0	0%
Final Project Done	0	0%

Appendix 2: Team Contract (i.e. Team Organization)

Step 1: Get to Know One Another. Gather Basic Information.

Team Member Name	***Strengths each member bring to the group***	***Other Info***	***Contact Info***
Angela Harris	Organization, quick learner, communication, flexibility	Strong believer in the idea that there are no “dumb” questions	harrisam2@vcu.edu
Emma Smith	Problem-solving, React experience, leadership	I am passionate about learning new things and enjoy collaborating with others.	smither3@vcu.edu
Sohil Marreddi	Eager to learn from others, previous project experience	Intrigued to build real world products and create that product from scratch.	marreddiss@vcu.edu
Cameron Clyde	Adaptable, team-work, quick learner, front-end experience.	I'm always looking for ways to improve my knowledge and skills.	clydecp@vcu.edu

Other Stakeholders	***Notes***	***Contact Info***
Faculty Advisor: Preetam Ghosh, VCU Engineering		pghosh@vcu.edu
Sponsor: Keroles Hakem, CoStar Group		khakem@costar.com

Step 2: Team Culture. Clarify the Group's Purpose and Culture Goals.

Culture Goals	***Actions***	***Warning Signs***
Attend every meeting	- Set up meetings in shared calendar - Send reminder message in group chat day of meeting - Discuss when there are schedule conflicts/reschedule as needed	- Student misses meeting without communication
Holding each other accountable to stay on top of work	- Work on project weekly - Discuss goals for the week and add them to meeting notes - Set reasonable deadlines for goals	- Student shows up for weekly meeting with no considerable work done
Creating a safe environment with open communication	- Talking through problems/conflicts as they arise - Communicate schedule - Open to new ideas	- Shutting down other teammates ideas - Passive aggression

Step 3: Time Commitments, Meeting Structure, and Communication

Meeting Participants	*Frequency* *Dates and Times / Locations*	*Meeting Goals* *Responsible Party*
Students Only	*As Needed, On Discord Voice Channel or Zoom*	*Discuss updates on progress and challenges* *Work through challenges as group and discuss potential solutions* *(Angela will record notes and upload them to shared Google Drive)*
Students Only	*Every Monday 7-9pm in library room*	*Actively work on project, discuss any difficulties* *Review previous weeks progress and discuss goals for upcoming week* *(Emma will take notes and upload them to shared Google Drive)*
Students + Faculty advisor	*As Needed*	*Update faculty advisor and get answers to our questions* *(Sohil will take notes on shared Google Docs; Angela will organize and lead meeting)*
Students + Project Sponsor + Faculty advisor	*Thursday at 6PM on Zoom*	*Update project sponsor to ensure we are on the right track* *Ask questions and discuss challenges* *(Cameron will take notes; Emma will organize and lead meeting; Sohil will demo prototype so far and give updates)*

Step 4: Determine Individual Roles and Responsibilities

Team Member	***Role(s)***	***Responsibilities***
Angela Harris	Logistics Manager	- Primary contact for communication with faculty advisor - Documenting meeting times - Obtaining information for the team - Following up on communication of commitments
Emma Smith	Project Manager	- Primary contact for communication with sponsor - Create a welcoming environment at meetings - Document and organize team goals for week - Schedule weekly meetings with sponsor - Book library rooms for student meetings
Cameron Clyde	Test Engineer/Financial Manager	- Monitors team budget. - Oversees experimental design, test plan, procedures and data analysis - Leads presentation of experimental finding and resulting recommendations
Sohil Marreddi	Systems Engineer	- Takes notes of client/sponsor requirements to put together solutions - Works with team members to design architecture of the software

Step 5: Agree to the above team contract

Team Member: Angela Harris *Signature: Angela Harris*

Team Member: Sohil Marreddi *Signature: Sohil Marreddi*

Team Member: Cameron Clyde *Signature: Cameron Clyde*

Team Member: Emma Smith *Signature: Emma Smith*

References

[1] A global leader in the digital transformation of the $300+ trillion real estate industry. Home | CoStar Group. (2024). https://www.costargroup.com/

[2] Gil-Gomez, H., Guerola-Navarro, V., Oltra-Badenes, R., & Lozano-Quilis, J. A. (2020). Customer relationship management: Digital Transformation and Sustainable Business Model Innovation. *Economic Research-Ekonomska Istraživanja*, *33*(1), 2733–2750. https://doi.org/10.1080/1331677x.2019.1676283

[3] Mesquita, T., Martins, B., & Almeida, M. (2022, October). Dense template retrieval for customer support. In Proceedings of the 29th International Conference on Computational Linguistics (pp. 1106-1115). https://aclanthology.org/2022.coling-1.94/

[4] Sheth, J. N., Jain, V., & Ambika, A. (2024). Designing an empathetic user-centric customer support organization: Practitioners' perspectives. European Journal of Marketing, 58(4), 845–868. https://doi.org/10.1108/ejm-05-2022-0350

[5] Malik, R., Subramaniam, L. V., & Kaushik, S. (2007, January). Automatically selecting answer templates to respond to customer emails. In IJCAI (Vol. 7, No. 1659, p. 3015). https://www.researchgate.net/profile/Lv-Subramaniam/publication/220815935_Automatically_Selecting_Answer_Templates_to_Respond_to_Customer_Emails/links/55b6580e08aec0e5f436fe0e/Automatically-Selecting-Answer-Templates-to-Respond-to-Customer-Emails.pdf