

Sweng 837 Summer 2024

Professor Santosh Nalubandhu

Eric Williams

Project:

Generative AI-powered Customer Service Chatbot: Design a chatbot system that uses natural language processing and machine learning to provide instant customer support and personalized recommendations.

Problem Statement and Requirements

Business Requirements:

- a. Clearly define the problem the system aims to solve.
 - a. This system shall provide users with instant customer support using a chatbot that uses natural language processing and machine learning for a car dealership to streamline website interactions for sales, financing, and repairs.
- b. Specify the functionalities the system needs to provide.
 - a. Chatbot shall connect users to the requested service, whether that be sales, financing, or vehicle servicing.
 - i. Sales prompts the chatbot helps the user find a desired car based on their needs and then forwards the user to a sales representative to the most suitable car dealer with the customer's information and car model choices.
 - ii. Financing prompts the chatbot to connect the user to a financing specialist.
 - iii. Vehicle servicing prompts the user to specify details related to servicing needs and schedule an appointment for the user.
 - b. Chatbot shall track usage data and interaction data on the website to measure user engagement and effect on number of sales.
 - c. Chatbot shall keep user data secure.
- c. Identify the target users and their needs.
 - a. Informed buyer
 - i. An informed buyer already knows what type of car they need and can be forwarded directly to a sales representative once they provide the make and model of the car they want.
 - b. Undecided buyer
 - i. An undecided buyer does not know what size car they want, but knows how much money they can spend, how many people the car should seat, a brand preference, and a mileage preference.

- c. Existing Customer
 - i. An existing customer either needs to contact a financing agent or the vehicle service provider. After providing their email address or phone number, along with a password, the chatbot shall recognize the customer and connect them to the appropriate service.
- d. Outline any business goals the system should support.
 - a. The number of car price quotes should increase by 30%.
 - b. Increase total sales by 15%.
 - c. Decrease sales related phone calls by 50%.

Non-Functional Requirements:

- a. Define performance requirements like scalability, response time, and throughput.
 - a. Chatbot response time should be less than 10 seconds per query.
 - b. The chatbot shall be able to work with multiple customers across multiple devices at the same time.
 - c. Chatbot maintains performance across large and small databases.
- b. Specify security requirements like authentication, authorization, and data encryption.
 - a. Existing Customer authentication through the chatbot is encrypted.
 - b. Chatbot does not provide users with unauthorized information such as other customer information or employee information.
- c. Outline maintainability requirements like code modularity, documentation, and testing strategies.
 - a. The system shall be able to handle error handling gracefully if the user gives an invalid prompt.
 - b. The system shall be able to schedule service appointments and email sales representatives seamlessly.
 - c. The system shall be designed to accommodate future enhancements.
- d. Indicate any other non-functional requirements relevant to the system's success.
 - a. The system tracks total interactions with the chatbot.
 - b. The system records the number of messages per session with each user.
 - c. The system tracks what percentage of conversations that users abandon before completing their desired action.
 - d. The system tracks the number of successful outcomes of the chatbot and customers.

UML Use Case Diagrams

This section discusses the actors relevant to this system and their interactions with it.

Description Table for Car Dealership Chatbot System		
Type	Actor	Goal Description
Primary	Buyer	The buyer wants to buy a car. The buyer may know what car they want to purchase, or they may not know what car they want to purchase and need recommendations
Primary	Existing customer	An existing customer has already purchased a car from the dealership and needs the car to be serviced.
Primary	Car Salesman	The car salesman wants to be able to make a sale and wants the buyer to have a car picked out before making a sale.
Supporting	Finance Specialist	The finance specialist provides financing options for buyers who need a loan for their car. They need to know how much money the buyer must loan
Supporting	Vehicle service provider	The vehicle service provider repairs the vehicle. They need to know what time the car will be dropped off at the service plaza and who the car is registered to.
Offstage	Dealership owner	The dealership owner wants his employees to be more productive and make more sales. He believes that this chatbot will streamline the sales process.
Offstage	Regulation committees	AI is under regulatory control for privacy and security

Use Cases

Use Case Section	Comment
Use Case Name	Informed buyer wants to buy a car
Scope	Car sales chatbot system
Level	Sub function
Primary Actor	Buyer
Stakeholder and Interests	Car Salesman

Preconditions	Website can host the chatbot system, buyer opened chatbot on the website, all cars on website are available for purchase
Success Guarantee	Buyer requests a quote on the car
Main Success Scenario	<ol style="list-style-type: none"> 1. Buyer opens website and clicks on chatbot. 2. Buyer queries the chatbot for the car they have in mind. 3. Chatbot finds car options and provides buyer with links. 4. Buyer picks a car that they like and requests a quote. 5. Chatbot requests contact information from buyer 6. Chatbot emails car salesman notifying that the buyer is interested and provides contact information
Extensions	<p>2.a Chatbot cannot find a car that meets the buyer's specifications, so it provides similar options.</p> <p>2.b Chatbot cannot understand buyer's query, so it asks for clarification</p> <p>4.a buyer does not see a car option they like and requests other options</p>
Special Requirements	N/a

Use Case Section	Comment
Use Case Name	Undecided buyer wants to buy a car
Scope	Car sales chatbot system
Level	Sub function
Primary Actor	Buyer
Stakeholder and Interests	Car Salesman
Preconditions	Website can host the chatbot system, buyer opened chatbot on the website, all cars on website are available for purchase
Success Guarantee	Buyer requests a quote on the car
Main Success Scenario	<ol style="list-style-type: none"> 1. Buyer opens website and clicks on chatbot. 2. Buyer queries that they need recommendations for what car they should buy. 3. Chatbot requests for details on what the customer needs 4. Customer provides details of their car needs. 5. Chatbot suggests 5 cars from the company website that meet given specifications, and requests if there are more specifications. 6. Repeat steps 3 to 5 until user selects a car. 7. Buyer picks a car that they like and requests a quote. 8. Chatbot requests contact information from buyer 9. Chatbot emails car salesman notifying that the buyer is interested and provides contact information
Extensions	<p>4.a Customer provides uninterpretable query.</p> <p>4.b Chatbot notes query is uninterpretable and requests a new query from the buyer.</p>

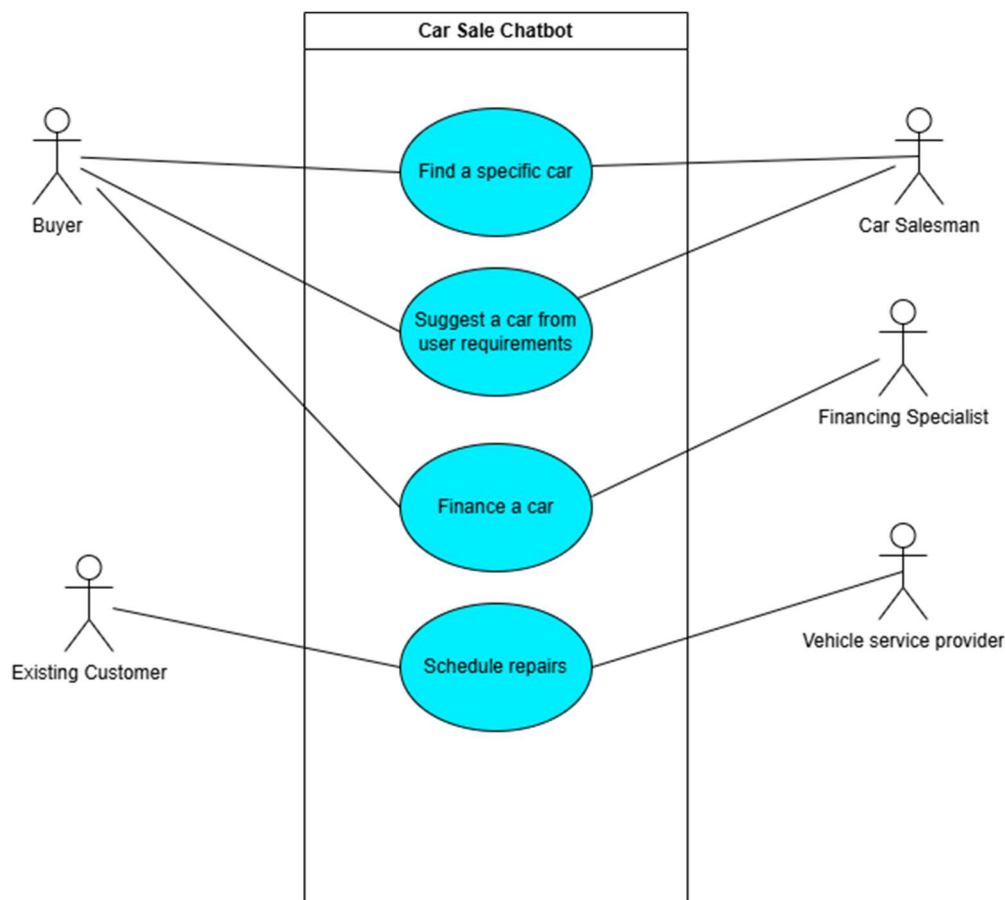
	5.as If chatbot cannot find a car with given specifications, tell user that a car with those specifications cannot be found and request for a different query
Special Requirements	N/a

Use Case Section	Comment
Use Case Name	Buyer wants to finance a car
Scope	Car sales chatbot system
Level	Sub function
Primary Actor	Buyer
Stakeholder and Interests	Car salesman, finance specialist
Preconditions	Website can host the chatbot system, buyer opened chatbot on the website
Success Guarantee	Buyer decides to finance the car or buyer decides not to finance the car
Main Success Scenario	<ol style="list-style-type: none"> 1. Buyer opens website and clicks on chatbot. 2. Buyer queries for financing options on a car they have picked. 3. Chatbot requests for financial details on what the customer needs and desired car 4. Customer provides financial details and car information. 5. Chatbot emails car salesman notifying that the buyer is interested and provides contact information. 6. Chatbot emails finance specialist notifying that the buyer is interested and provides contact information
Extensions	4.a Car is not found on website or is no longer for sale.
Special Requirements	N/a

Use Case Section	Comment
Use Case Name	Existing customer wants to get car repaired
Scope	Car sales chatbot system
Level	Sub function
Primary Actor	Existing customer
Stakeholder and Interests	Vehicle service provider
Preconditions	Website can host the chatbot system, buyer opened chatbot on the website
Success Guarantee	Car repair appointment is scheduled, and vehicle service provider is notified
Main Success Scenario	<ol style="list-style-type: none"> 1. Existing customer opens website and clicks on chatbot.

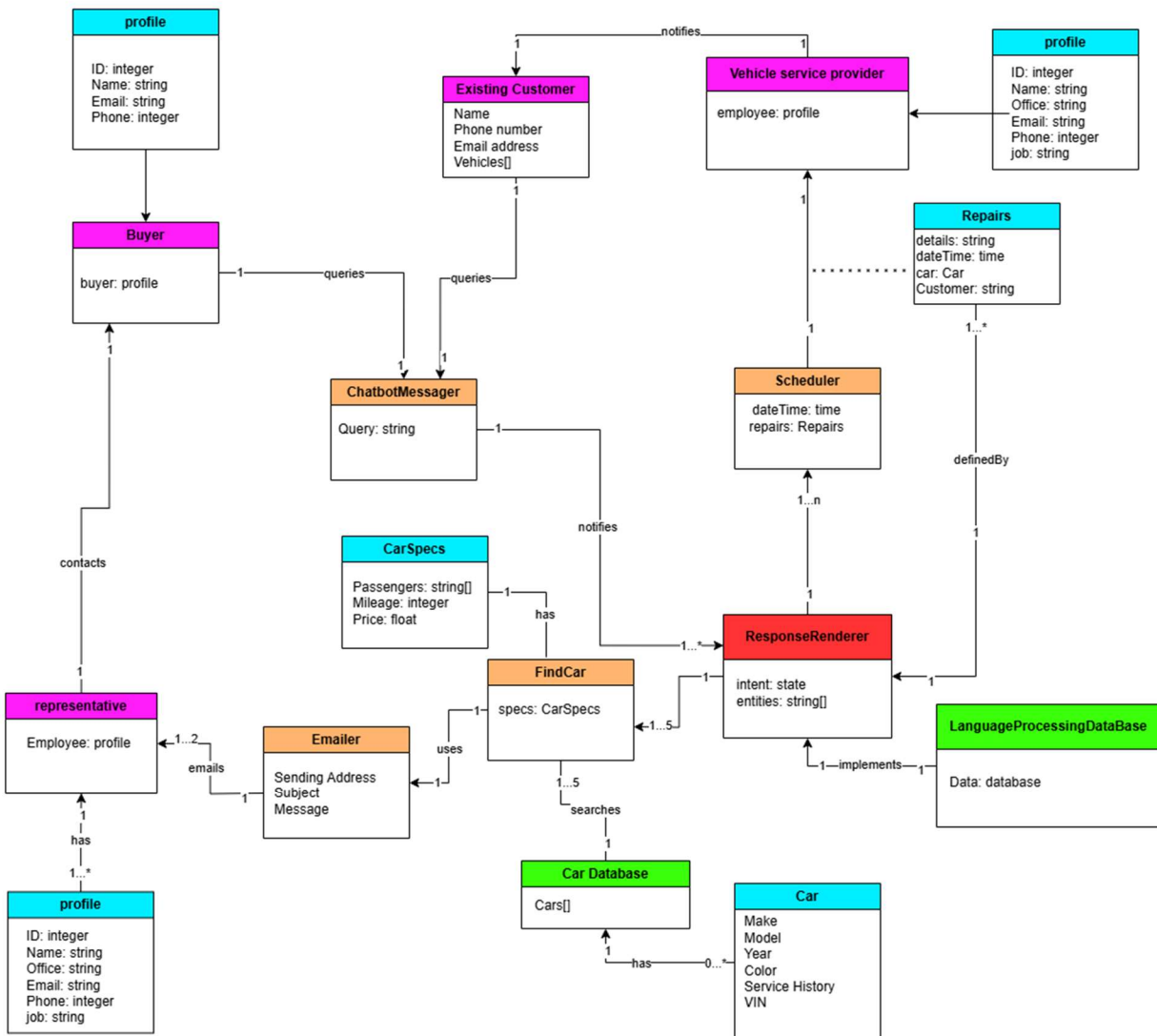
	<ol style="list-style-type: none"> 2. Existing customer queries for car repairs on a car they own. 3. Chatbot checks records for customer's information and car's record, and queries date and time for car drop-off, 4. Customer provides date and time for drop-off of repair. 5. Chatbot queries customer for reasons for drop off. 6. Customer provides reason for drop off. 7. Chatbot schedules appointment for decided date and time, and emails vehicle service provider with details of reason for drop off, date and time for repair, and customer information
Extensions	<ol style="list-style-type: none"> 3.a customer does not exist in database yet. 3.b chatbot request for more information from customer to register. 4.a customer date and time options are not available. 4.b chatbot requests different date and time for drop off
Special Requirements	N/a

Use case diagram:



UML Domain Model

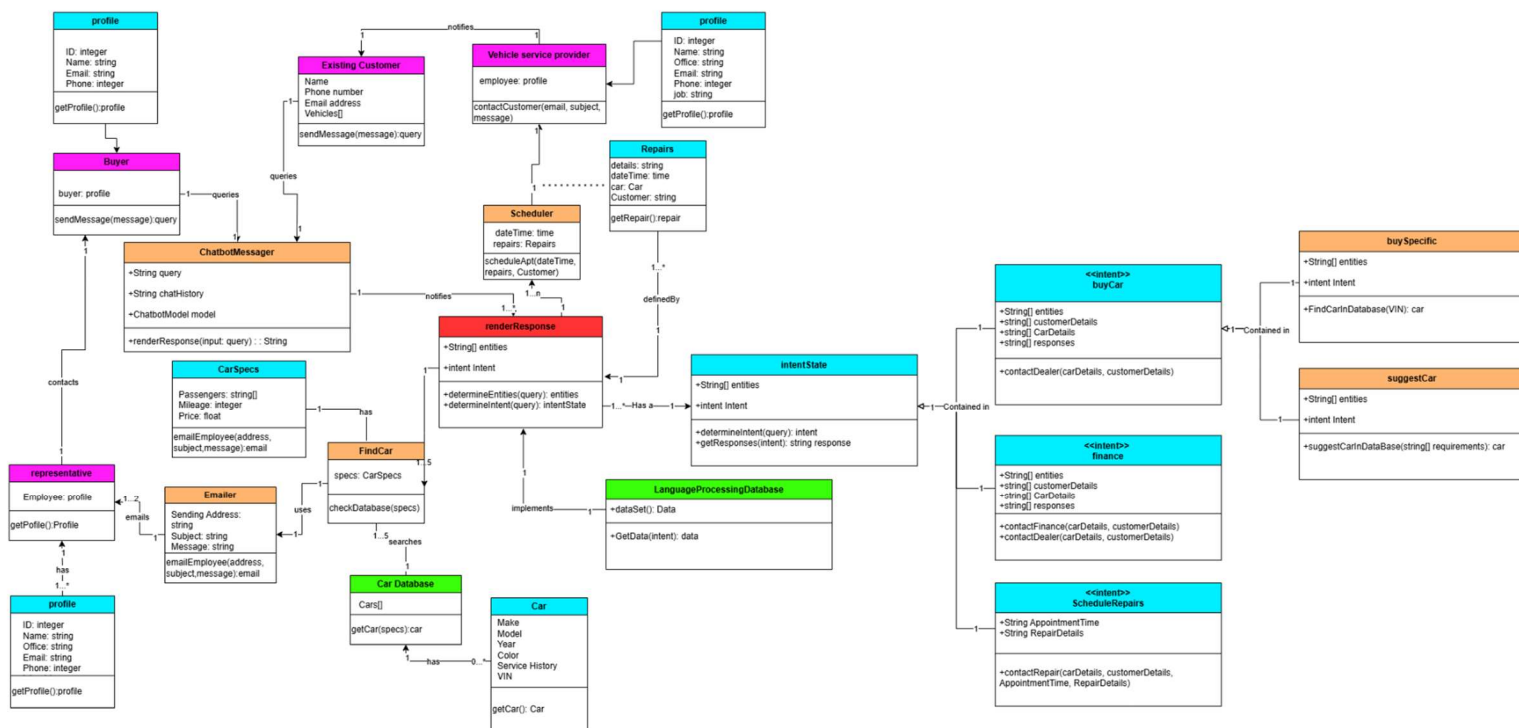
The following diagram shows key objects in the problem domain as well as other objects they are associated with.



The key entities in this domain model are the Response Renderer, the buyer, the language processing database, and employees for the car company like the salesman, finance specialist, and the vehicle service provider. The language processing database is required so that the chatbot can process real language and determine which task a user is attempting to accomplish. It is important to separate car details from the car database to prevent data overload and slow runtimes for the chatbot. Additionally, it is important to note that the car database is constantly being updated as other cars are being bought and sold.

The chatbot not only needs to be able to take language input and interpret it into a specific task, but also use other tools down the line such as an emailing tool and scheduling tool to coordinate with humans in the system.

UML Class diagram



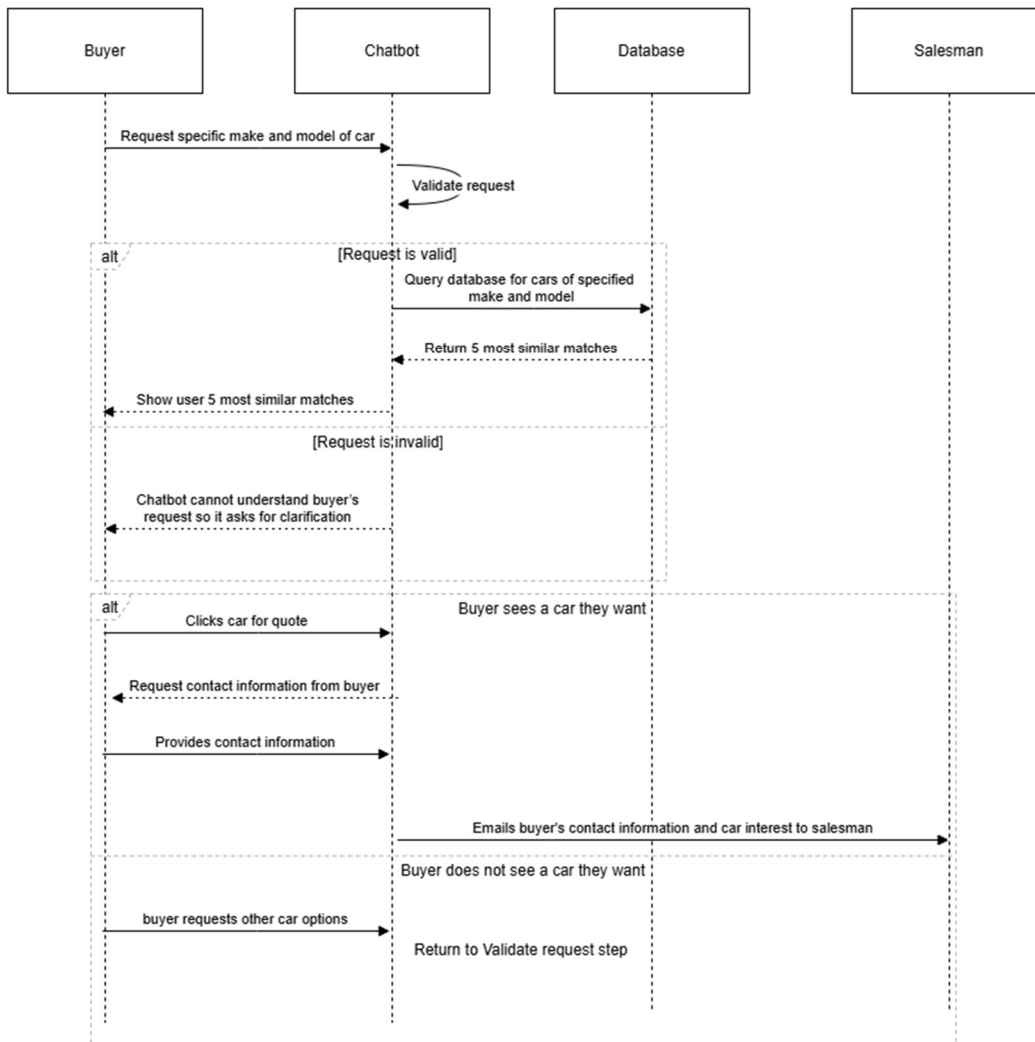
This class diagram elaborates on the domain model primarily focusing on how the chatbot should handle message inputs. In short, this class diagram gives more details on what is happening in the black box of the Machine Learning Database in the domain model.

The main thing that changed with this diagram compared to the domain model is the addition and definition of the intentState module which tells the ResponseRenderer how to determine the intent of the user. This reduces the complexity of the operations and better conveys what each class is designed for. The main technique this diagram employs is aligning the user's intent with a generalized workflow and then fill in the blanks with contextual entities that the user provides in the prompt. If any entities are missing, the chatbot will request the user for the missing entity, until the chatbot has enough information to complete a workflow path.

UML Sequence Diagrams

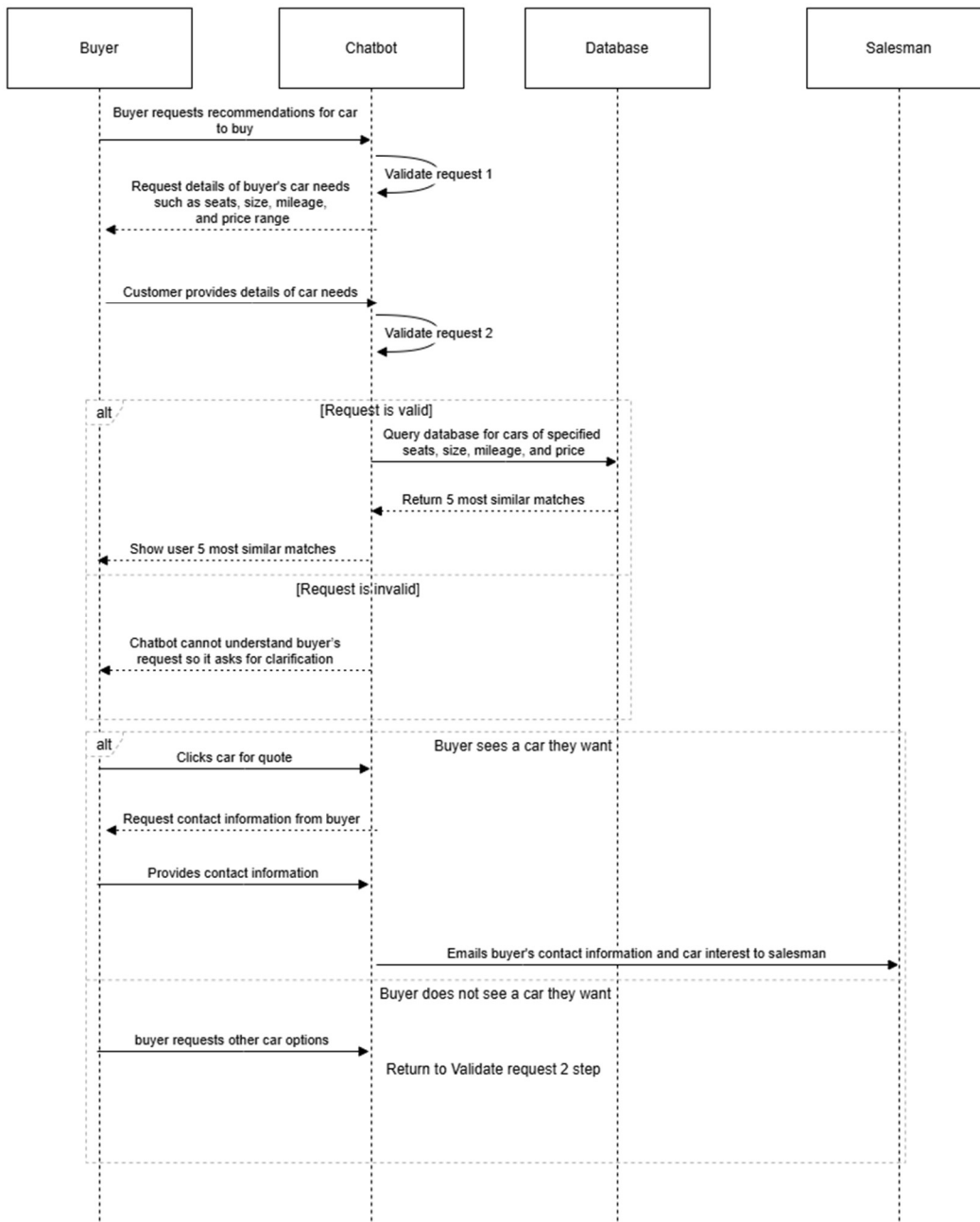
Name of Operation	FindCarInDatabase(string VIN)
Responsibilities	Informed buyer wants to buy a car

Type	System
Cross Reference	Use Case 1
Exception	If request is invalid, chatbot should ask user to clarify
Outputs	Email Car Salesman with customer information
Preconditions	Customer provided valid VIN; car has not already been sold
Post-conditions	None



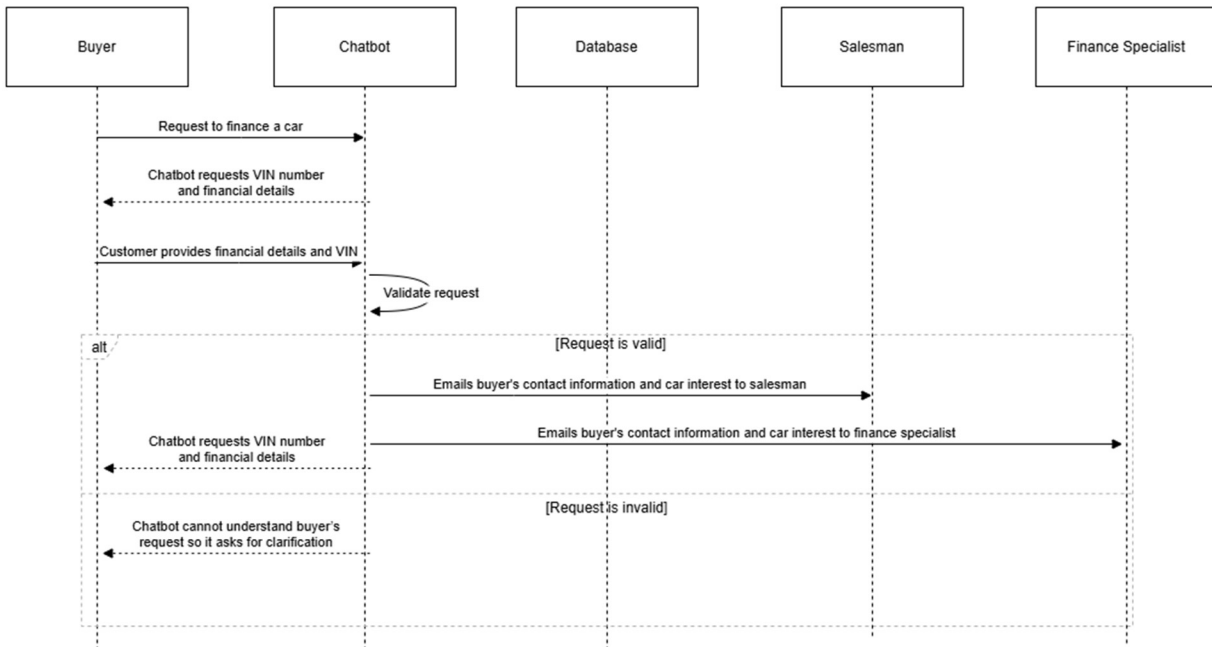
Name of Operation	suggestCarInDatabase(string[] requirements)
Responsibilities	Provide undecided buyer with enough information to choose a car to purchase
Type	System
Cross Reference	Use Case 2

Exception	If request is invalid, chatbot should ask user for more details regarding missing input fields
Outputs	Email Car Salesman with customer information and car choice
Preconditions	Car database is available for chatbot to search
Post-conditions	None

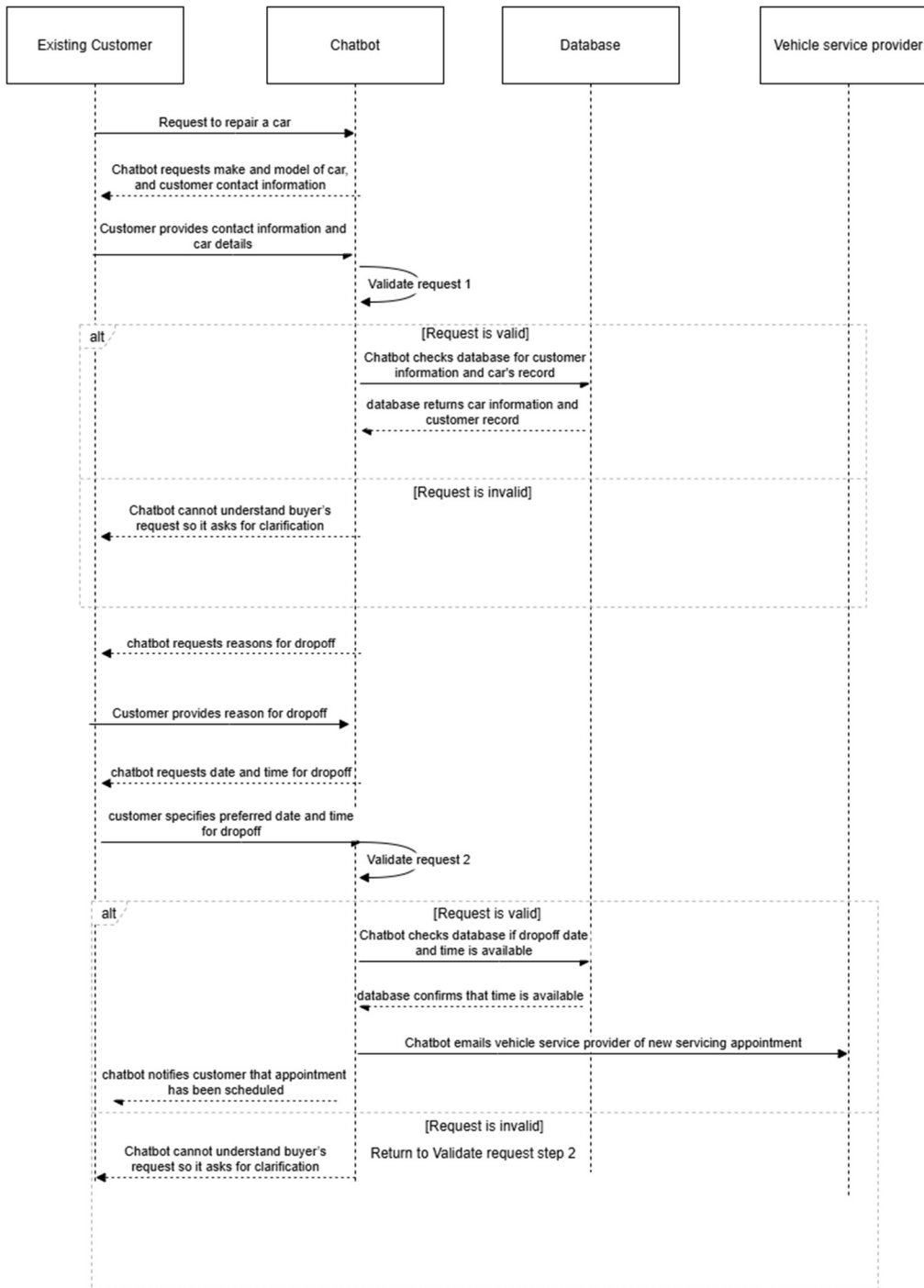


Name of Operation	emailRepresentative (email,subject, text)
Responsibilities	Connect buyer with a desired car model with a sales member and finance specialist, and provide financial details to specialist
Type	System

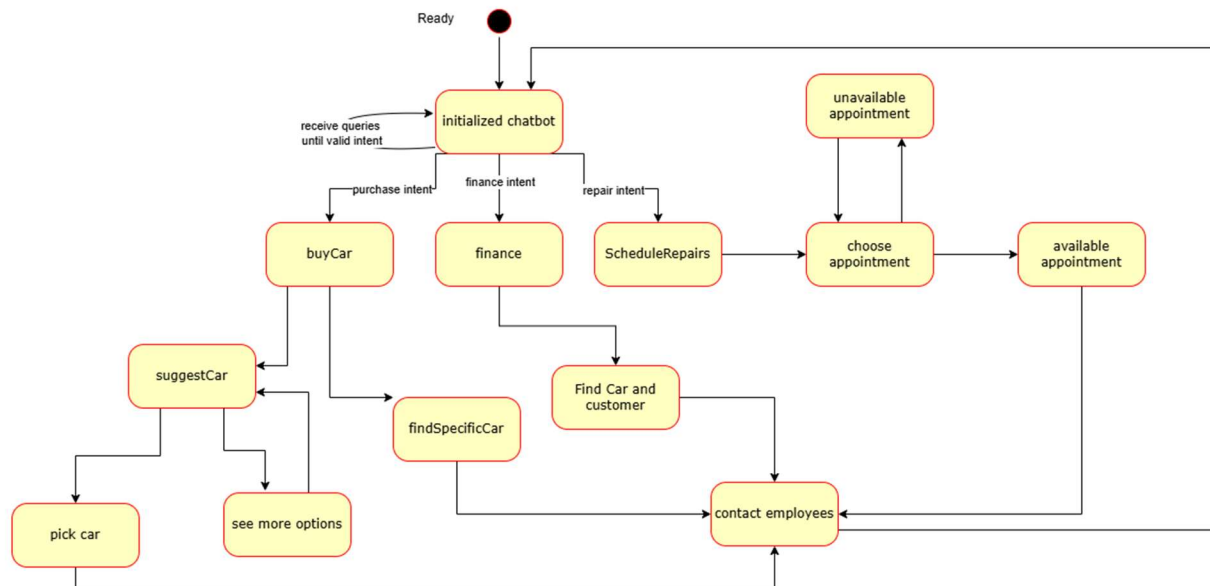
Cross Reference	Use Case 3
Exception	If request is invalid, chatbot should ask user for more details regarding missing input fields
Outputs	Email Car Salesman and finance specialist with customer information and car choice
Preconditions	Car database is available for chatbot to search
Post-conditions	None



Name of Operation	scheduleRepair(car details, string[] customerDetails, string AppointmentTime, string RepairDetails)
Responsibilities	Schedule a car repair after a user provides a valid appointment time
Type	System
Cross Reference	Use Case 4
Exception	If the appointment time is invalid, request a different appointment time.
Outputs	Notify user and vehicle service provider that an appointment for car repair has been scheduled
Preconditions	Car database is available for chatbot to search
Post-conditions	None



UML State Diagram

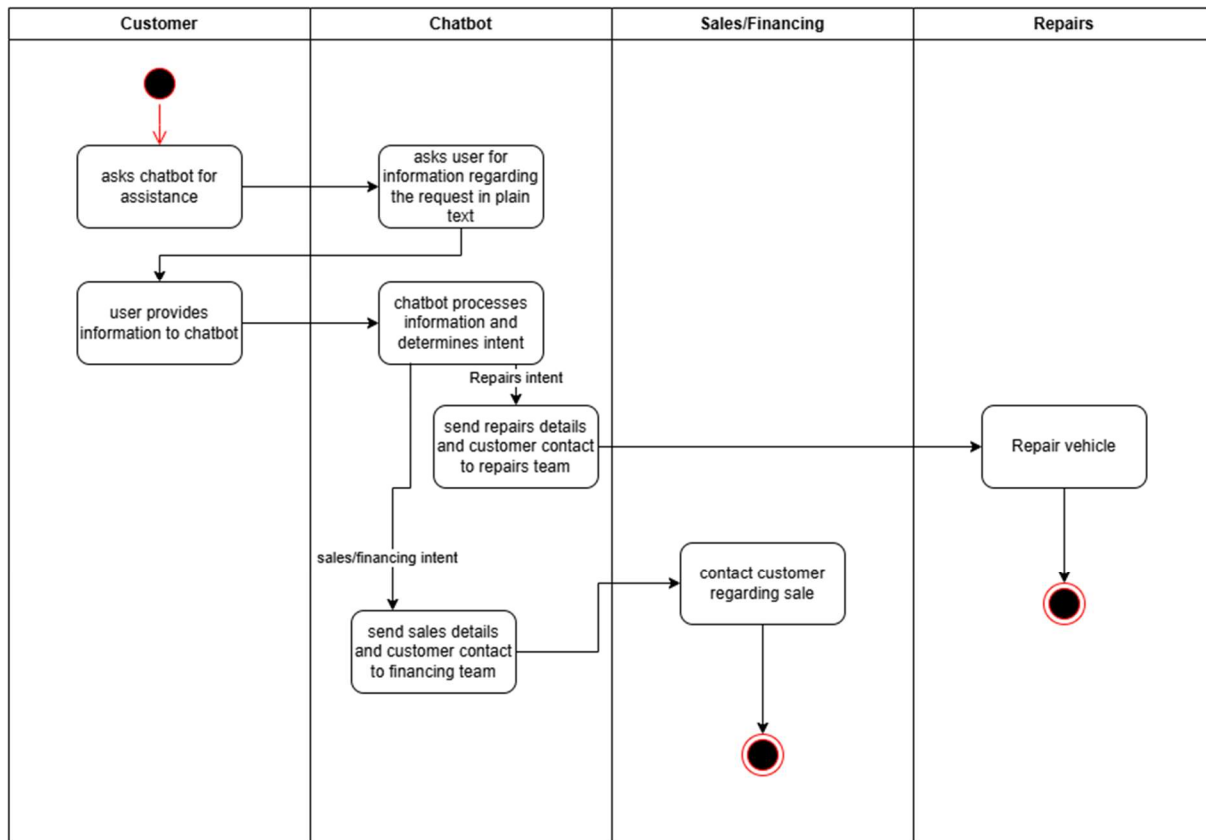


The start of the state diagram begins with a user interacting with the chatbot. From there, the chatbot interacts with a user until it determines that a user wants to buy a car, contact a financing agent, or schedule repairs. If the user wants to buy a car, the chatbot must determine if the user already has a car they intend to purchase or if they need suggestions. After a car is chosen, the chatbot contacts relevant car dealership employees, specifically salesmen. It then returns to its initialized state.

If the chatbot determined that the user wants to speak to a financing agent, the chatbot attempts to determine more information about the customer as well as details about the car they intend to take out a loan for. After this information is gathered, the chatbot forwards this information to a finance specialist.

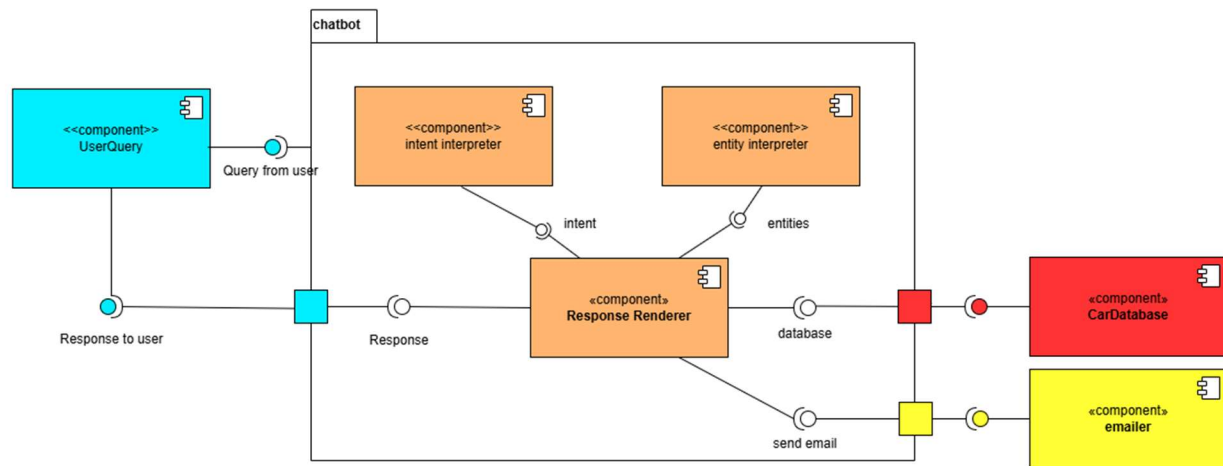
Lastly, if the chatbot determines that a user wants to schedule their car for repairs, they communicate with a user attempting to select a valid appointment date and time. If the appointment is available, the chatbot will provide a vehicle service provider with information about the car and a notification that the appointment has been scheduled.

UML Activity Diagram



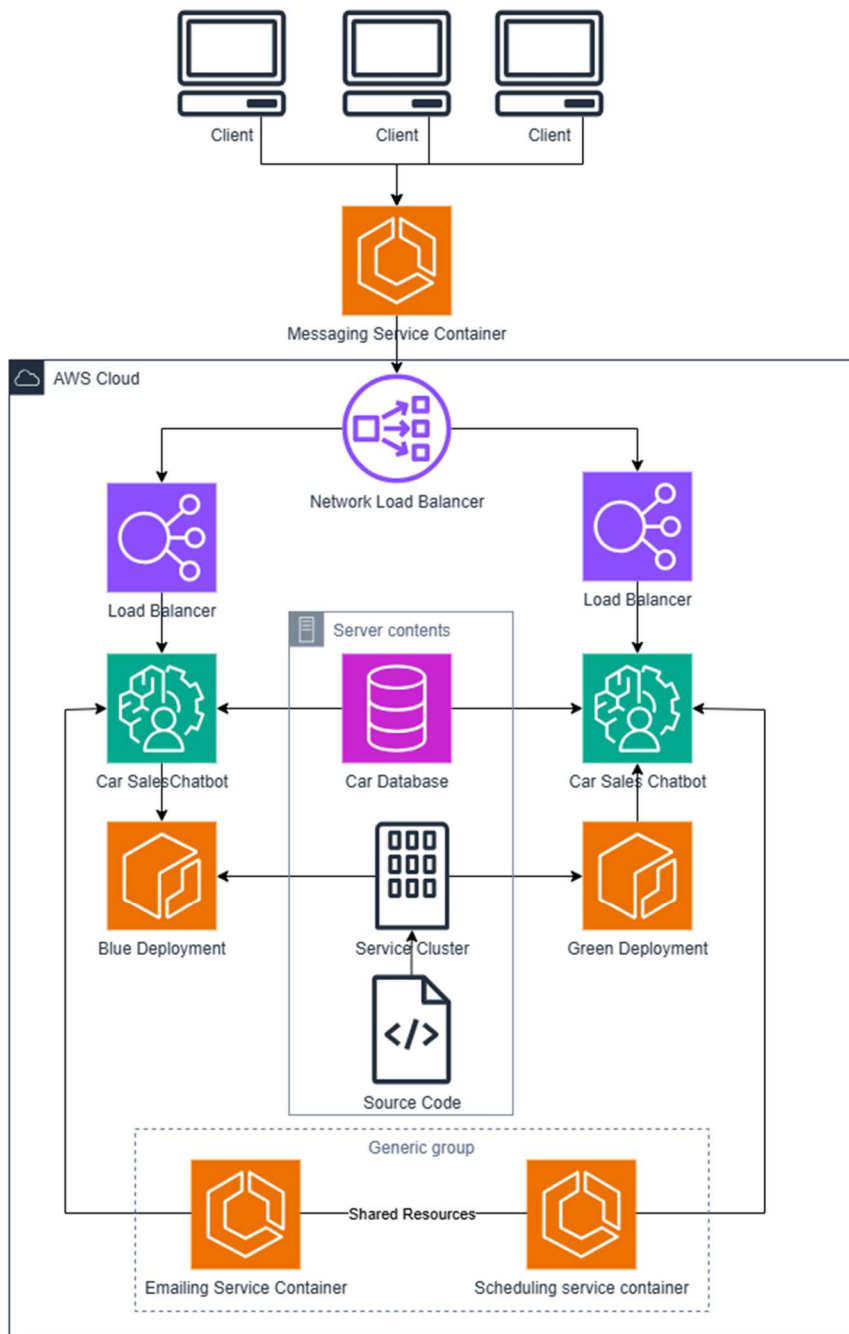
For the above diagram, there are 4 actors: Customer, Chatbot, Sales/financing, and repairs. The customer is either a buyer or an existing customer. The chatbot connects the Customer with whatever service they need. Sales/Financing represents either the car salesman or the financing specialist that perform the sale. These two were merged to reduce complexity. Repairs represents the vehicle service provider who repairs the vehicle.

UML Component Diagram



The above component diagram demonstrates how user queries are interpreted by the chatbot. The chatbot receives an input of a query from a user and must use context of the message to determine the intent of the message as well as track all the entities listed in the query. Once this information is handled the chatbot begins to render its response after consulting the database containing all information regarding the cars that are available for sale, cars that need financing, or cars that need servicing. Once cars have been selected, the user emails the appropriate employees regarding the intended car sale, financing transaction, or repairs appointment.

Cloud Deployment Diagram



Rationale:

I decided to use the Blue Green Deployment pattern as it makes it so that the chatbot will remain online even though the database is constantly updated due to daily sales. Minimizing the downtime means that these changes to the database will not prevent a user from requesting a quote for a car or scheduling a repair. This also makes it easy to revert any changes as if there is

a server to roll back to if data is determined to be invalid. Additionally, the load balancer directs network traffic to whichever deployment is currently the active version of the chatbot, while the other version can be updated for the next iteration.

Skeleton Classes and Tables Definition

Classes:

Class findCar{

 Private specs CarSpecs

 public car checkDatabase(make, model, year, color, history, VIN);

}

Class emailer{

 Public void EmailEmployee(sendAddress, subject, message);

}

Class scheduler{

 Private Time dateTime;

 Private repairs Repairs;

 Public void scheduleApt(dateTime, repairs, customer);

}

Class chatbotMessenger{

 public string query;

 public string ChatHistory;

 private ChatbotModel model;

 Public string renderResponse(query);

}

Tables:

1. Car

Field	Data Type	Comment
-------	-----------	---------

ID	Integer	Identifier of the car in the database
VIN	String	Vehicle ID number
Make	String	Manufacturer
Model	String	Model of car
Type	String	Type of car like sedan or SUV
Year	Integer	Year car was released
Milage	Integer	Number of miles traveled in car
Price	Double	Price of car
Color	String	Color of car
History	String[]	List of all repairs

2. Employee

Field	Data Type	Comment
ID	Integer	Identifier of the employee in the database
Name	String	Employee name
Office	String	Geographic work location
Email	String	Email address
Phone	integer	Phone number to contact
Job	String	Identifier of responsibilities, like salesman

Design Patterns

This section discusses which design patterns and best practices were used throughout the design process of this system.

The architecture used for this design was cloud native design. Since the system needs to be connected to a learning database, it made more sense in terms of speed, safety, and scalability to have all resources be stored in the cloud rather than internal servers. It also allows the chatbot to easily perform other services down the line, or even work across multiple dealerships as all the data is stored in the cloud.

Some design patterns that were used were generalization classes, description classes, states, and strategy patterns. Description classes were a useful way for defining what was required to be stored in each component of the database such as cars; each car had tons of details to track, but a description class helped keep the diagrams uncluttered. The strategy and generalization patterns seemed to be very similar to each other. By defining all user behavior as one of 4 intents or strategies it made the system much easier to generalize what actions needed to be accomplished.

Some design principles that were used were KISS and YAGNI. With this design, it would have been easy to let scope creep and have the chatbot do everything for the company, but it seemed more manageable to start the bot off with a few core responsibilities with the potential to add more responsibilities down the line as they were needed. The focus of this design was to have 3 primary workflows the chatbot could complete before listing off every single possible behavior it could help with which would have led to both unnecessary complexity and even behavior that is not required for successful operation.