

Kristen Renee Haupt

SWENG837

Instructor: Santosh Nalubandhu

Fall 2024

# **SWENG 837 Final Project Documentation**

## **Business Requirements**

### **Business Problem**

Selection 6 - Generative AI-powered Customer Service Chatbot

### **Specifications**

Design a chatbot system that uses natural language processing and machine learning to provide instant customer support and personalized recommendations.

### **Identifying Target Users and Their Needs**

#### **Description Table**

Type	Actor	Goal Description
Primary	Customer	Customers want to be able to get instant responses to customer service issue.
Primary	Chatbot	The Chatbot wants to be able to replace a human customer service representative.
Supporting	Data Scientist(s)	Data Scientist(s) wants to be able to collect data from customer service experience.
Offstage	Company Administration	Admins want to be able to streamline and expedite customer service on their website.

### **Identifying Use Cases**

Use Case 1 – Customer initiates Chatbot interaction.

Use Case 2 – Generative AI conducts a survey to determine positivity rating of interaction.

Use Case 3 – Chatbot adds previous interaction to data pool.

## **Documenting Use Cases as Partially/Fully Dressed**

### **Use Case 1: Customer initiates Chatbot interaction.**

Goal in Context: The customer is prompted on the website to discuss with the Chatbot via a pop-up.

Primary Actors: Customer, Chatbot

Scenario:

- Step 1. Customer selects “customer service” pop-up.
- Step 2. Generative AI says “Hello” and asks opening question.
- Step 3. Customer answers.
- Step 4. Generative AI responds per algorithm.

### **Use Case 2: Generative AI conducts a survey to determine positivity rating of interaction.**

Goal in Context: To judge the effectiveness of the algorithm, the generative AI should conduct a survey to judge the outcome of the interaction for future use in correcting the algorithm.

Primary Actors: Customer, Chatbot

Scenario:

- Step 1. Generative AI completes customer service requests with customer.
- Step 2. Generative AI asks if the customer service interaction was positive or negative.
- Step 3. Customer responds.
- Step 4. Generative AI asks how the interaction could be improved.
- Step 5. Customer responds.

### **Use Case 3: Chatbot adds previous interaction to data pool.**

Goal in Context: The Chatbot will need to be updated periodically so the data obtained during each customer service interaction will need to be saved in a form that can be added to the training/testing/validation data for the algorithm.

Primary Actors: Data Scientist(s), Chatbot

Scenario:

- Step 1. After the survey is complete, the data from the interaction is logged to a .csv file.
- Step 2. Data scientist(s) validate information using AI/ML techniques.
- Step 3. Data scientist(s) update the algorithm periodically.

# **Non-Functional Requirements**

## **Performance Requirements**

1. Scalability – The Chatbot needs to be updated with new information on a schedule determined by the data scientist. The Chatbot should be able to handle 10-100 users concurrently at launch with the ability to scale to 100-1000 users within the first month.
2. Response Time – The latency/processing time should match a natural conversation pattern.
3. Throughput – The chatbot should be able to converse and record 100 interactions concurrently upon release.

## **Security Requirements**

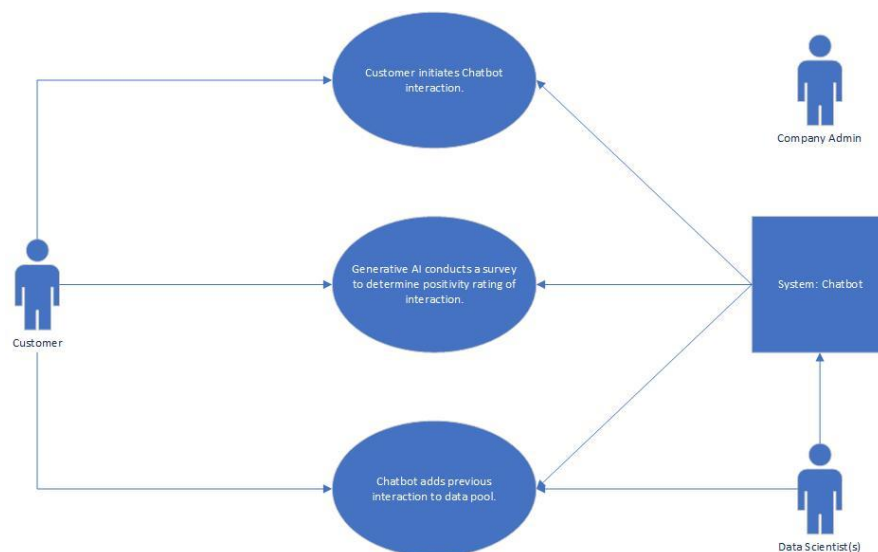
1. Data Encryption – Some business models may deal with sensitive data so XYZ encryption will be used to protect sensitive data from 3<sup>rd</sup> parties.

## **Maintainability Requirements**

1. Code Modularity – The code should not be domain specific. This chatbot should be able to be trained on any business type.
2. Documentation – All lines of code should be documented for future employees to understand and utilize. Documentation should be self-explanatory, clear, and concise.
3. Testing Requirements – The chatbot software should be able to be updated periodically with the caveat of being able to roll back to a previous version if the roll out has bugs.

# **System Design Using Domain Modeling**

## **UML Use Case Diagram**



# UML Domain Model

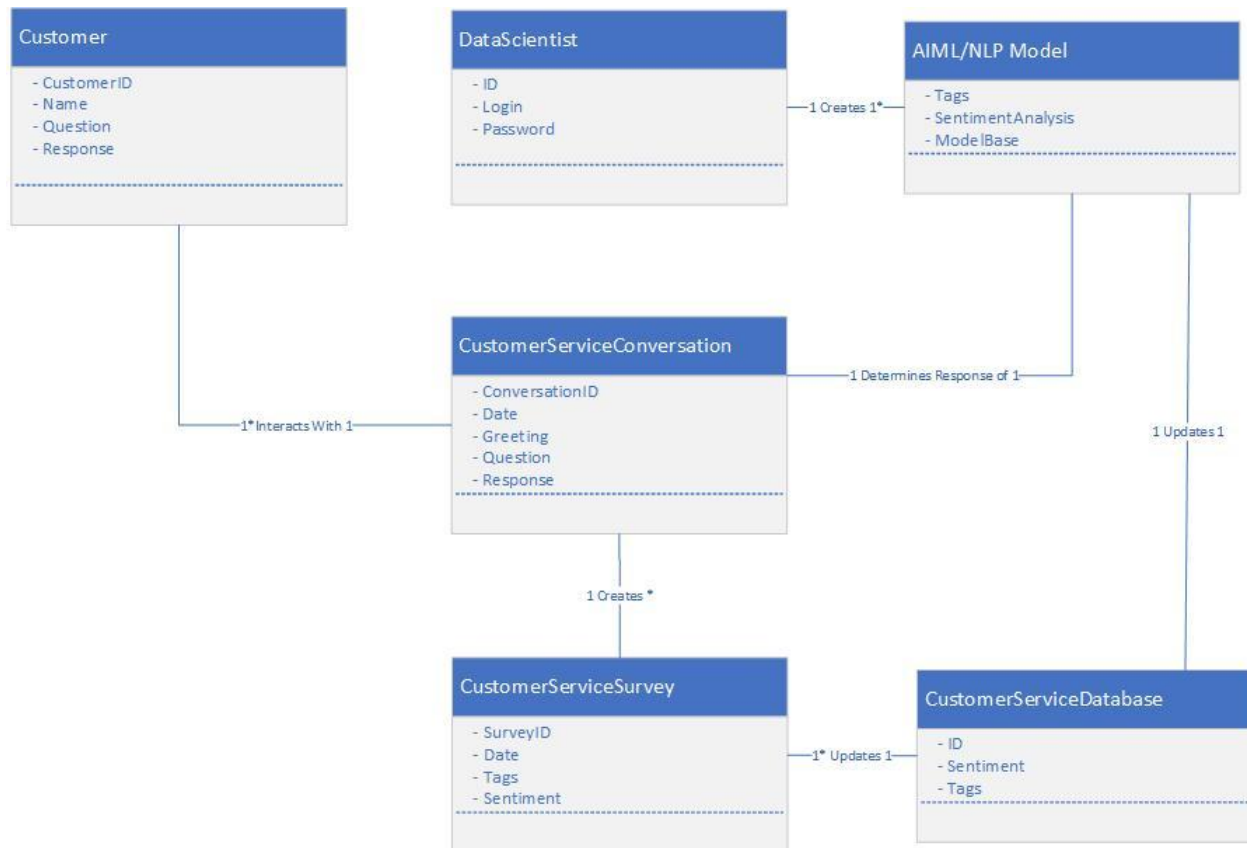
## Conceptual Class Table

Conceptual Class Category	Example
Users	Customer, DataScientist, Admin
System Functions	CustomerServiceConversation, CustomerServiceSurvey, AIML/NLP Model
External Systems	CustomerServiceDatabase, External Database, Security Protocols

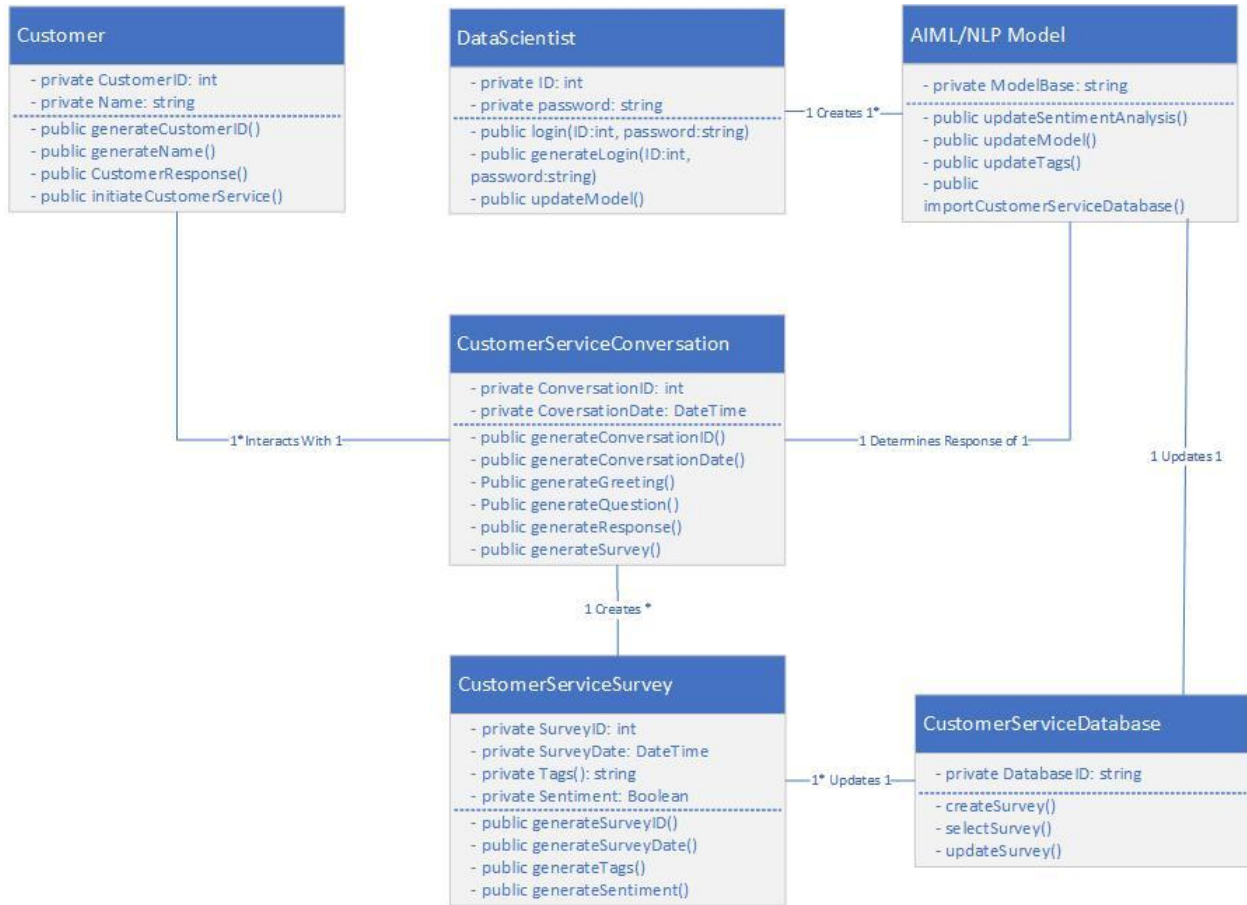
## Class Pruning

Retained	Pruned
Customer	Admin
DataScientist	External Database
CustomerServiceConversation	Security Protocols
CustomerServiceSurvey	
AIML/NLP Model	
CustomerServiceDatabase	

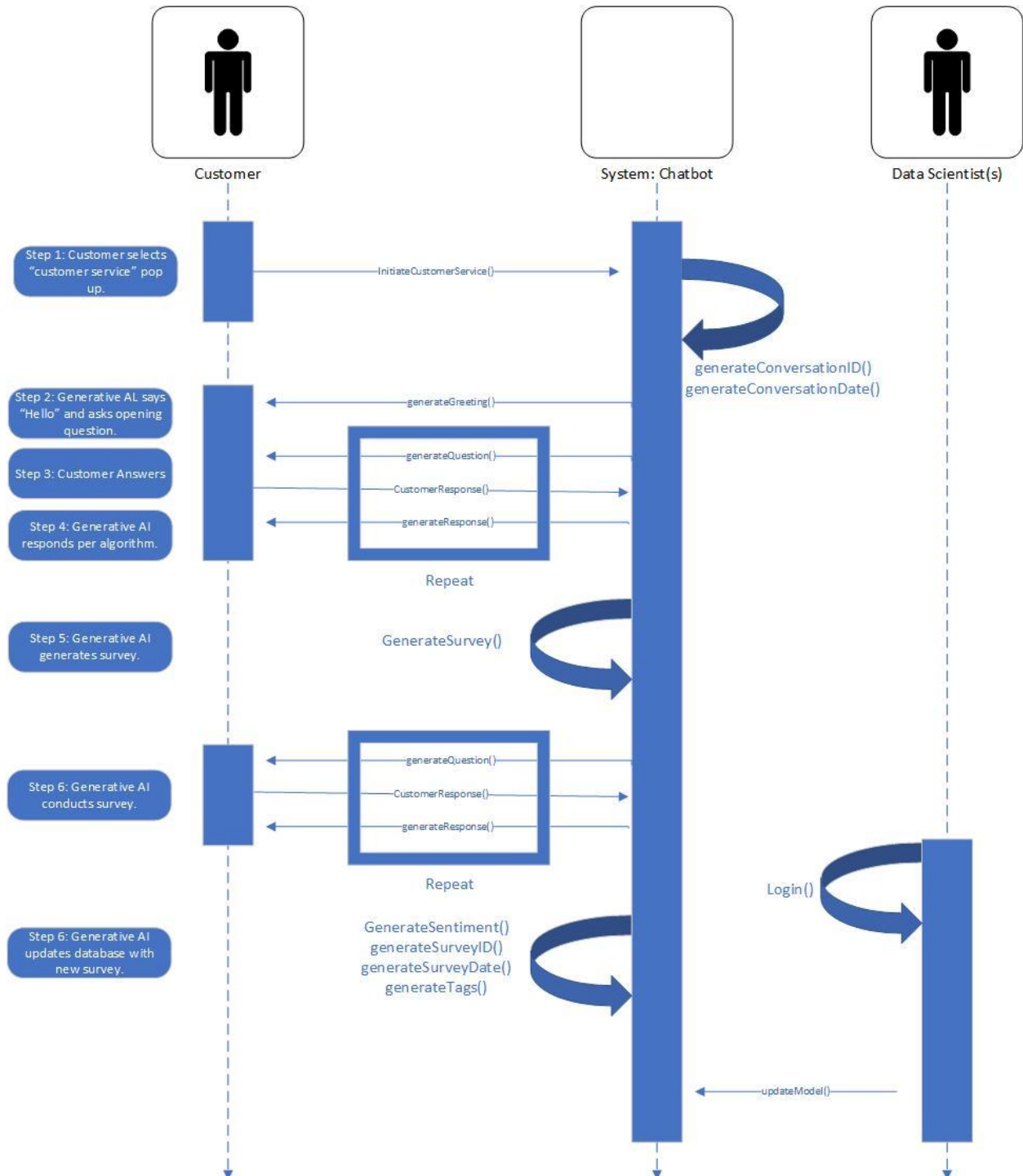
## Domain Model



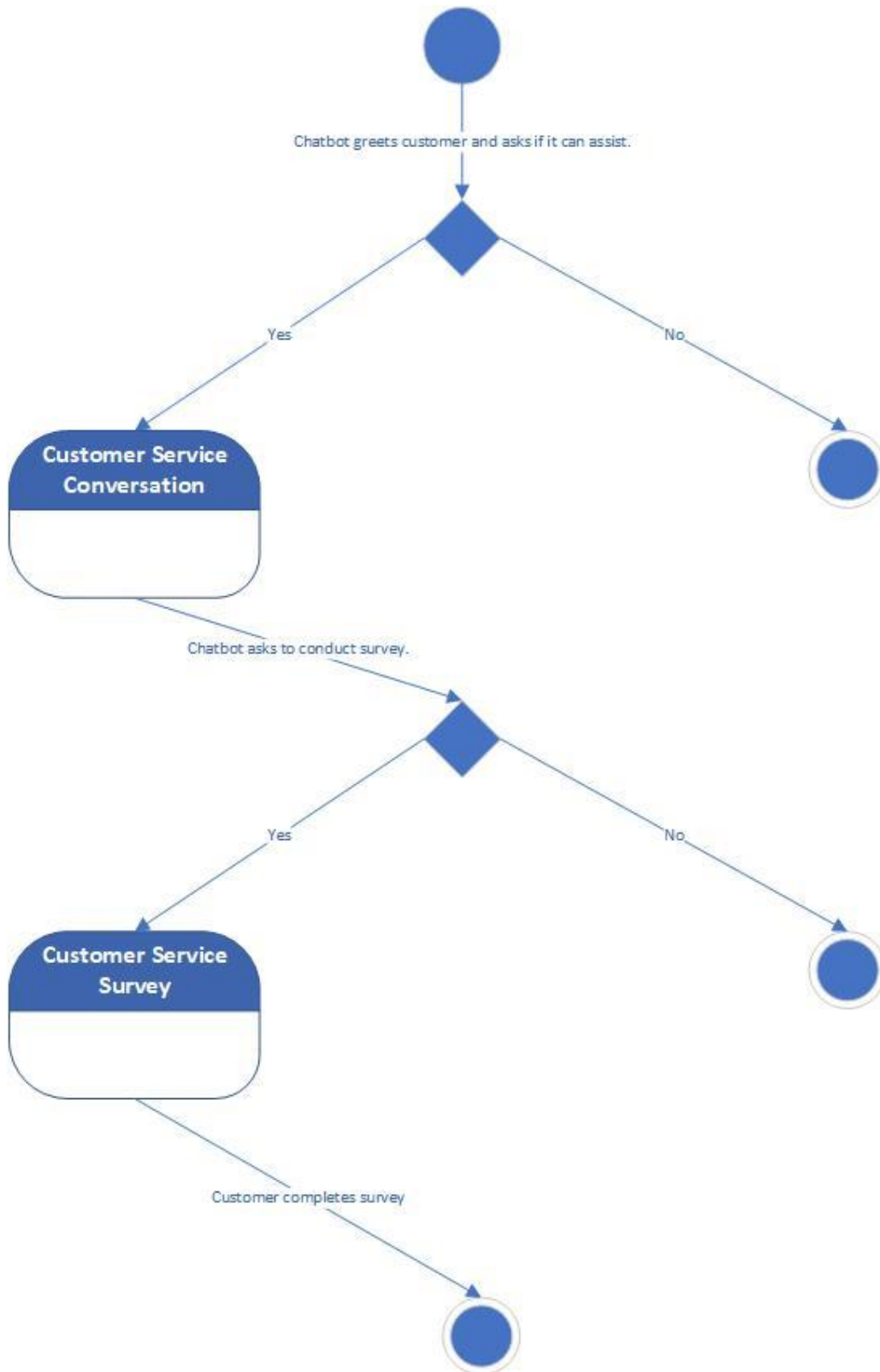
# Class Diagram



## Sequence Diagram

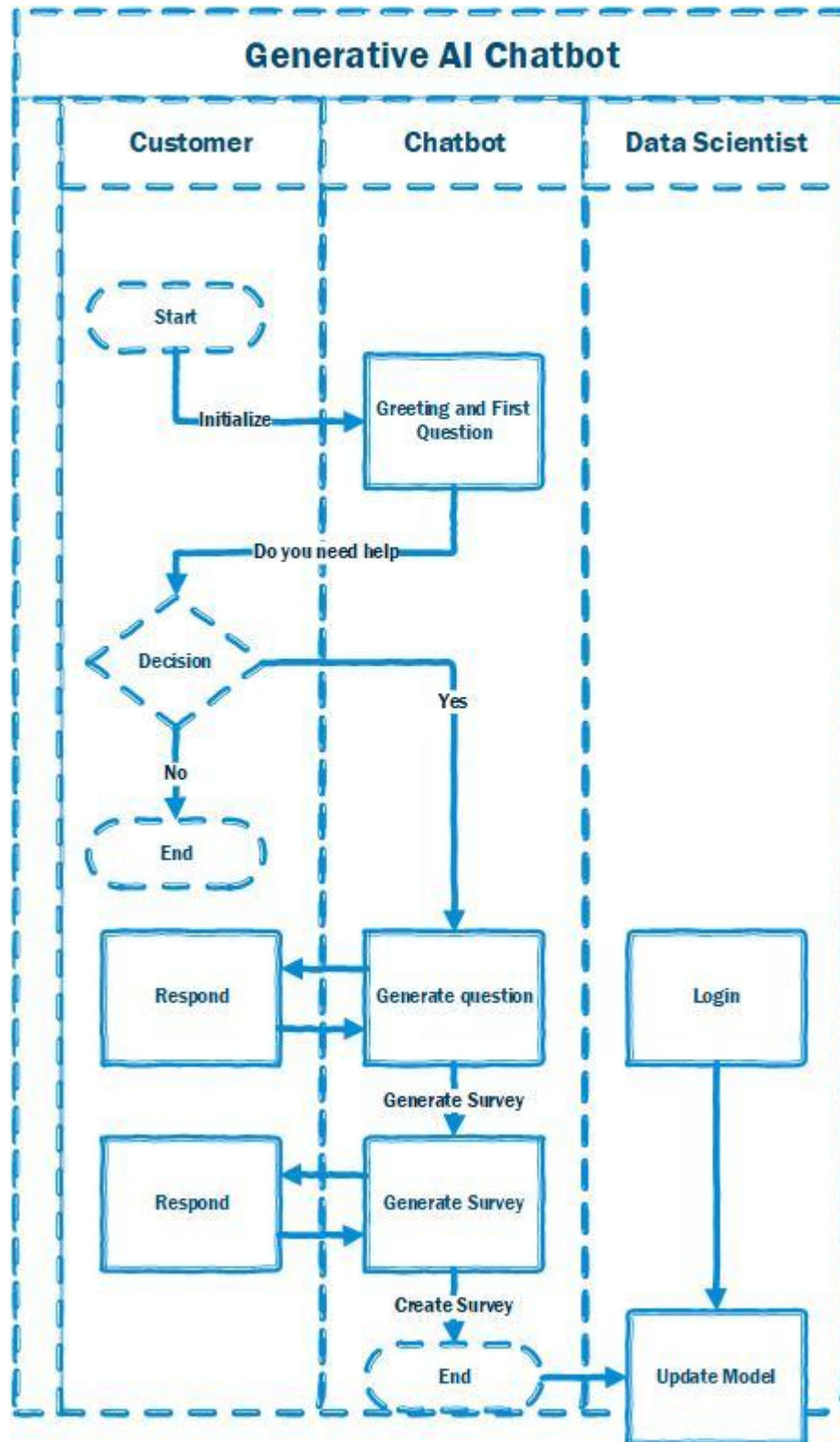


## State Diagram

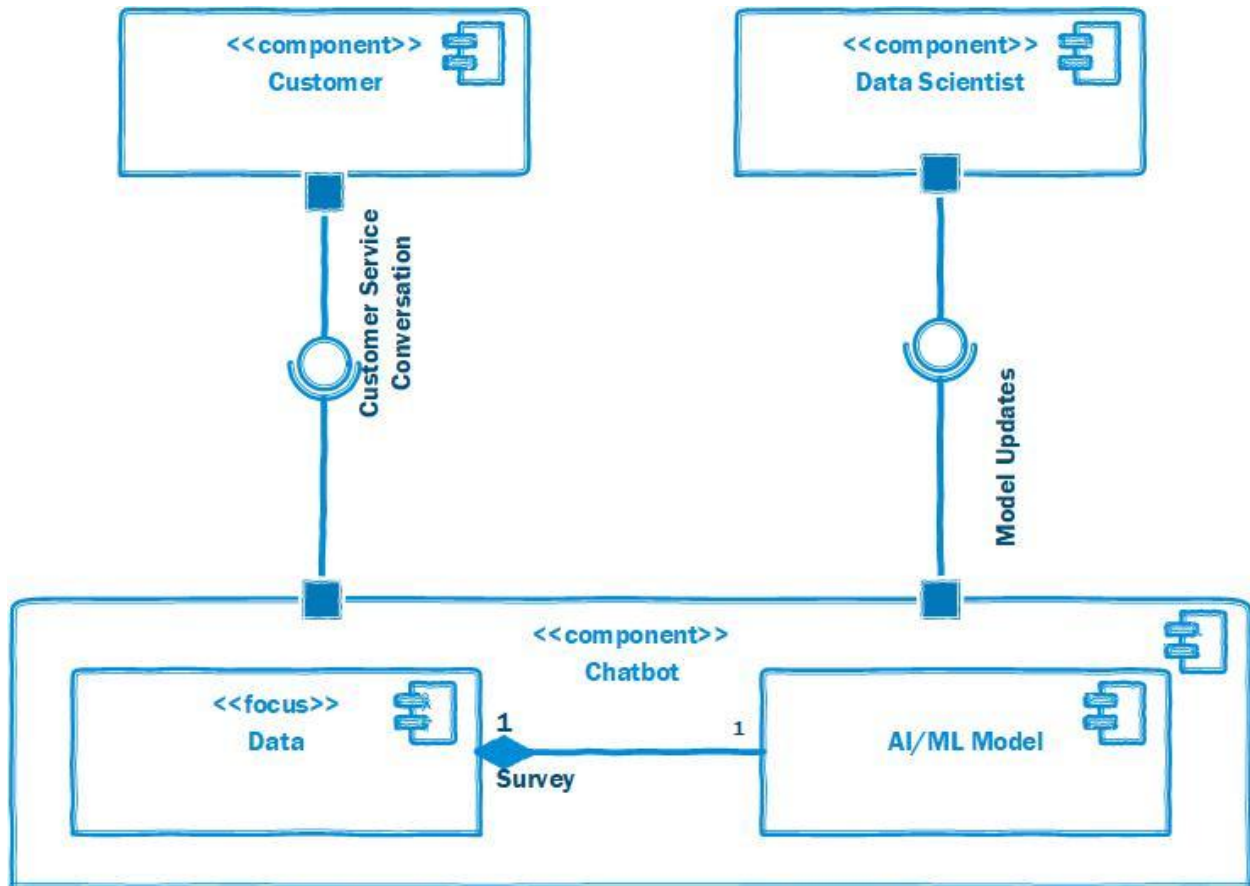




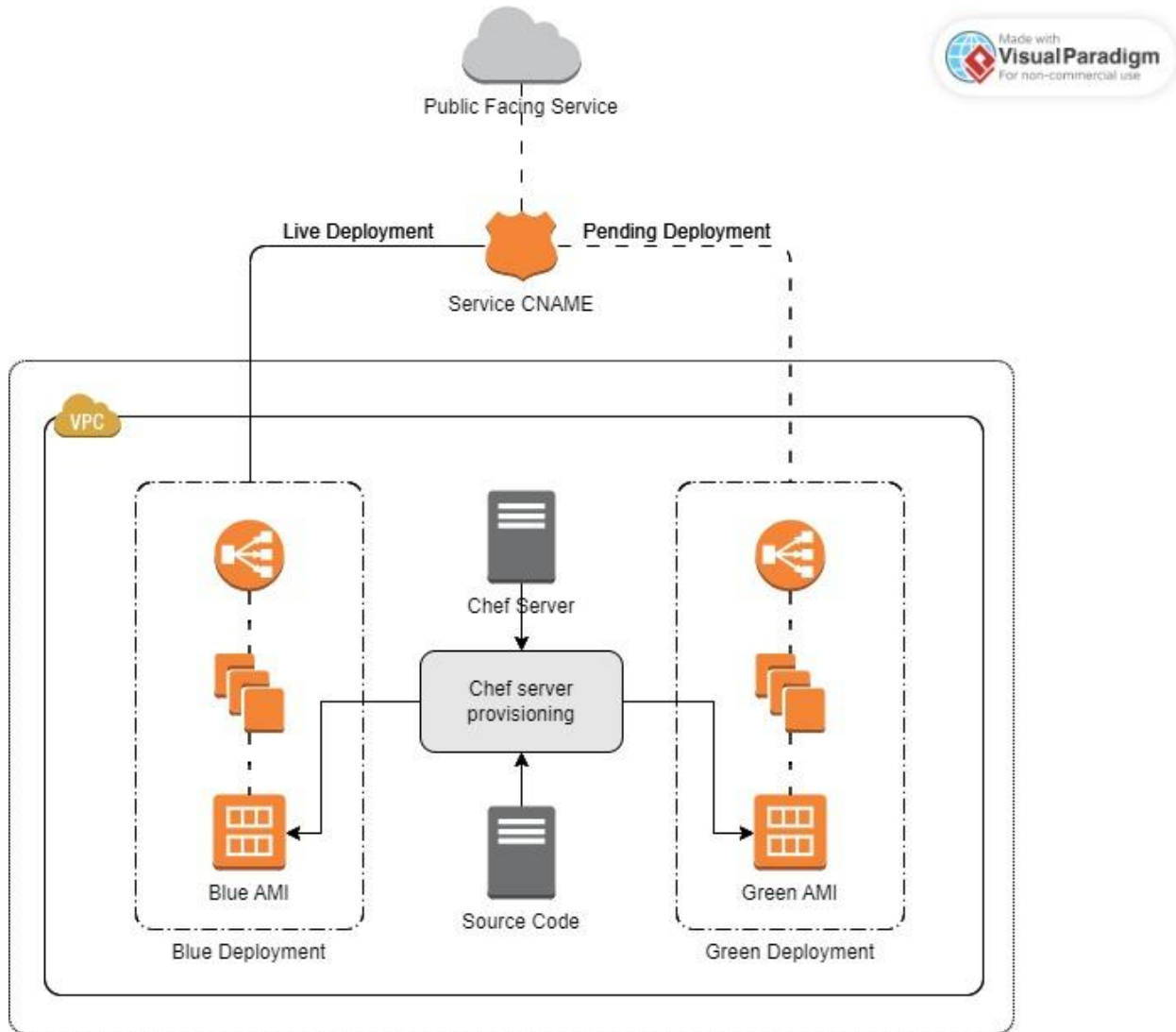
## Swimlane Diagram



## Component Diagram



# Cloud Deployment Diagram



From: <https://aws.amazon.com/solutions/case-studies/expedia/>

## Tables Definitions

Sentiment	Customer ID	Customer Name	Conversation ID	Conversation Date	Survey ID	Survey Date	Conversation Text	Tag 1	Tag n
bool	int	varchar(30)	int	datetime	int	DateTime	text	varchar(15)	Varchar(15)

**Sentiment** - Type: Boolean Explanation: 0 for negative sentiment, 1 for positive sentiment.

**Customer ID** – Type: Integer Explanation: Numeric identifier for the customer.

**Customer Name** – Type: Varchar(30) Explanation: 30 position variable width character string used to make the AI seem more human.

**Conversation ID** – Type: Integer Explanation: Numeric identifier for the customer service conversation.

**Conversation Date** – Type: Datetime Explanation: Time stamp of when the customer service conversation begins.

**Survey ID** – Type: Integer Explanation: Numeric identifier for the customer service survey.

**Survey Date** – Type: Datetime Explanation: Time stamp of when the customer service survey begins.

**Conversation Text** – Type: Text Explanation: A transcript log of the conversation between the customer and the chatbot.

**Tag 1 through n** - Type: Varchar(15) Explanation: 15 position variable width character string used to aid the data scientist in classification and further model training.

## Design Patterns

**KISS method so the chatbot can be used across numerous industries.**

- The idea was to keep the chatbot simple, at least as simple as one can make a chatbot using AIML techniques.
- The conversation between the customer and chatbot should be clear and concise, as well as avoid jargon.
- In terms of consistency, the chatbot should mimic the standard conversation via text that has been around for decades. We are not trying to reinvent the wheel here.
- The use of Tags that can be changed between different industries keeps this software flexible.
- The text chat type conversation should make this usable for any type of customer, while the back end model training features available to the data scientist should make fine tuning the model a breeze.