Statement of Work SOW

Price Negotiator System

Generic statement:

People prefer to purchase items on the internet. There are several websites that can help you with this. They do not need to go to the store to purchase the items they require. Websites are required, and this necessitates the targeting of the appropriate clientele. Ecommerce refers to websites that allow consumers to purchase things they want quickly and simply. What if there is an application that can provide some chatbot system in order to negotiate the prices of the products? Yes, using the price negotiator ecommerce chatbot system application is possible.

Business problem statement and Proposed solution:

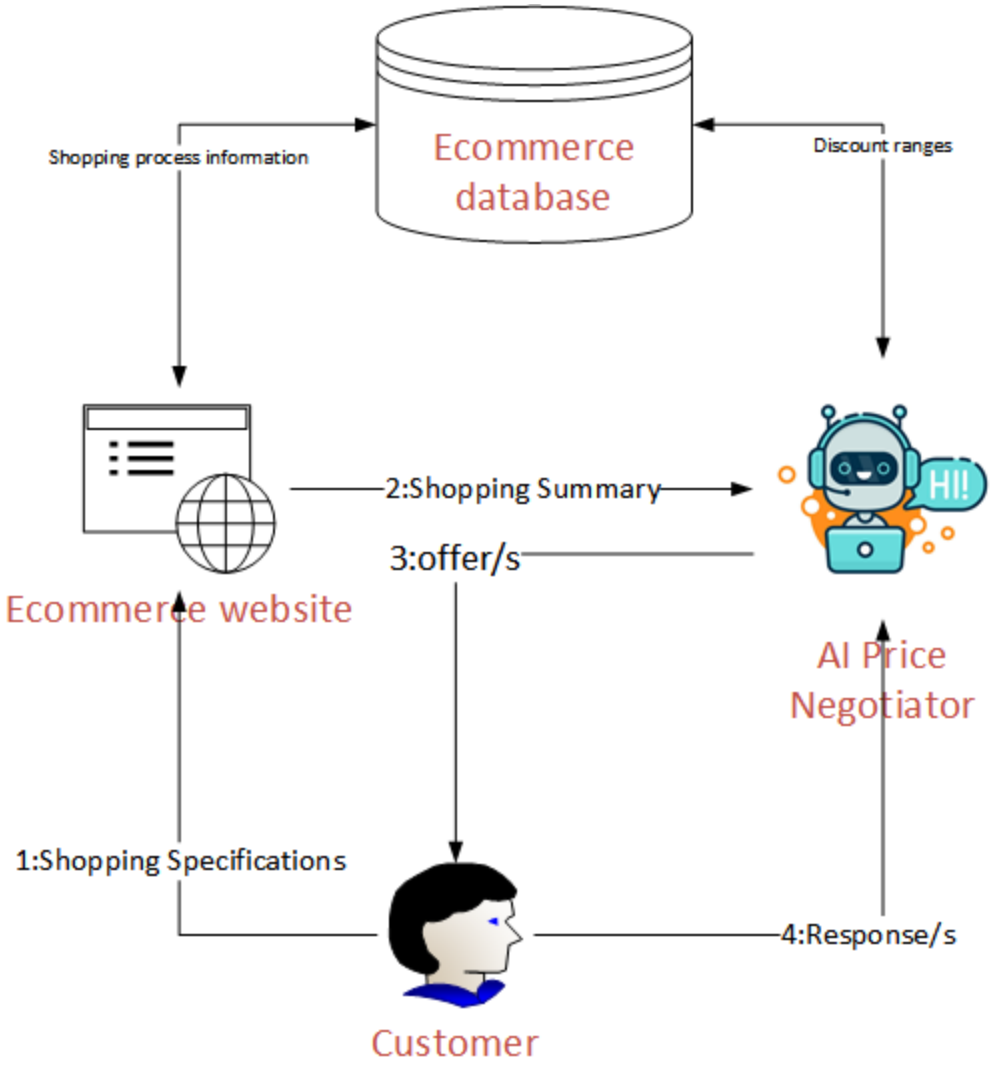
In the retail online business there are many factors that play a role from seller and buyer side. The seller wants to give the best product to the customer at the best price although no comparison process is done before presenting the product to the seller. Having a better price which is not a fixed one will increase the payoff among sellers and buyers. Thus, an automated agent for negotiation has been designed to maintain a flexible and considerable price instead of a fixed price. Chat-bot is mainly used to provide conversation between both human and machine. Team feeds some knowledge to the machine so that the machine can identify the sentences (NLTK) and take a decision itself in response to answering a question. People can rely on this application with great ease using this application.

The system allows the customer to browse the e-commerce website and add required products to the shopping cart. As the customer clicks on the checkout button, it provides checkout or cancels the shopping process. If the customer is satisfied with the total amount, he/she can choose to checkout. However, if he is not happy with the total amount and he/she wants a better price, he clicks on the Cancel button. As the customer clicks on the Cancel button, it redirects to the chatbot. When the chatbot gets initiated, it pulls the cart details, including the total price amount and the purchased product information. The chatbot accesses the database to retrieve the appropriate discount range concerning the total price amount. Then the AI Chatbot gives the user an option of a minimum discount from the discount range. If the customer agrees, the chatbot redirects to the cart with the applied discount, where the customer can checkout and receive an email with the product description, discount applied and total amount. If the user disagrees with the first minimum discount, the chatbot proposes a higher deal, the value being between the minimum and maximum range. If the user agrees, the chatbot redirects to the cart and the same process applies. However, suppose the user is still not satisfied with the discount. In that case, the chatbot proposes the maximum and final discount, to which if the customer accepts to pay, redirects to checkout to follow the same

process. Alternatively, if the customer is still unwilling to take the discount, the chatbot stops the negotiating, ends the deal and redirects to the checkout (cart) page.

In a case where the customer has a total price out of range for any discount to apply, the chatbot informs the customer that there is no discount for the particular price range, closes the deal and redirects to the checkout (cart) page.

The following figure shows the project architecture.



Data Requirements and Data Elements:

For constructing a backend database, we use Relational data structure and star schema.

[Website Database]

To create a chatbot feeding system, we feed data from an E-Commerce website which we created. There are mainly 5 tables where data is stored. The following 4 tables Category, Product, Customer, and Order , are used to run the ecommerce website . The fifth tableDiscount_Ranges is used from the chatbot section to retrieve the discount percentages ranges that the chatbot needs to build the negotiation process .

Table Name		**Products**	
Field name		Type	Reference
PK	id	Auto increment field	
	name	`CharField(max_length=60)`	
FK 1	`category`	`IntegerField(default=0)`	Category
	`description`	`CharField (max_length=250, default='', blank=True, null=True`	

Table Name		**Category**	
Field name		Type	Reference
PK	id	Auto increment field	
	name	`CharField(max_length=50)`	

Table Name		**Customer**	
Field name		Type	Reference
PK	id	Auto increment field	
	`first_name`	`CharField(max_length=50)`	
	`last_name`	`CharField(max_length=50)`	
	phone	`CharField(max_length=10)`	
	email	`EmailField()`	
	password	`CharField(max_length=100)`	

Table Name		**Order**	
Field name		Type	Reference
PK	id	Auto increment field	
FK1	`product`	`ForeignKey(Products, on_delete=models.CASCADE)`	Product
FK 2	`customer`	`ForeignKey(Customer, on_delete=models.CASCADE)`	Customer
	quantity	`IntegerField(default=1)`	
	price	`IntegerField`	

	phone	CharField (max_length=50, default='', blank=True)	
	date	DateField (default=datetime.datetime.today)	
	status	BooleanField (default=False)	

Table Name		**Discount_Ranges**	
Field name		Type	Reference
PK	id	Auto increment field	
	amont_from	IntegerField	
	amont_to	IntegerField	
	discount_from	models.DecimalField(decimal_places=2,max_digits=4)	
	Discount_to	models.DecimalField(decimal_places=2,max_digits=4)	

For chatbot training and identification of proper product names, we put intents.json for training and it works because of the collection of words from the NLTK library.

Data Assumption, limitations, and constraints:

Data Assumptions:

Chatbot: We are assuming that the customer will agree on any one of the discount ranges.

Assuming that the NLTK library has sufficient data to train, test for feeding the data to Chatbot from the E-commerce website, we took samples of products from various categories. Data in size is not that huge which we see on real e-commerce websites. It is for demonstration purposes and training purposes. We tried to feed Chatbot the words through the NLTK library (input from customers which they write and the AI chatbot takes as input) which is in range of our project vicinity only.

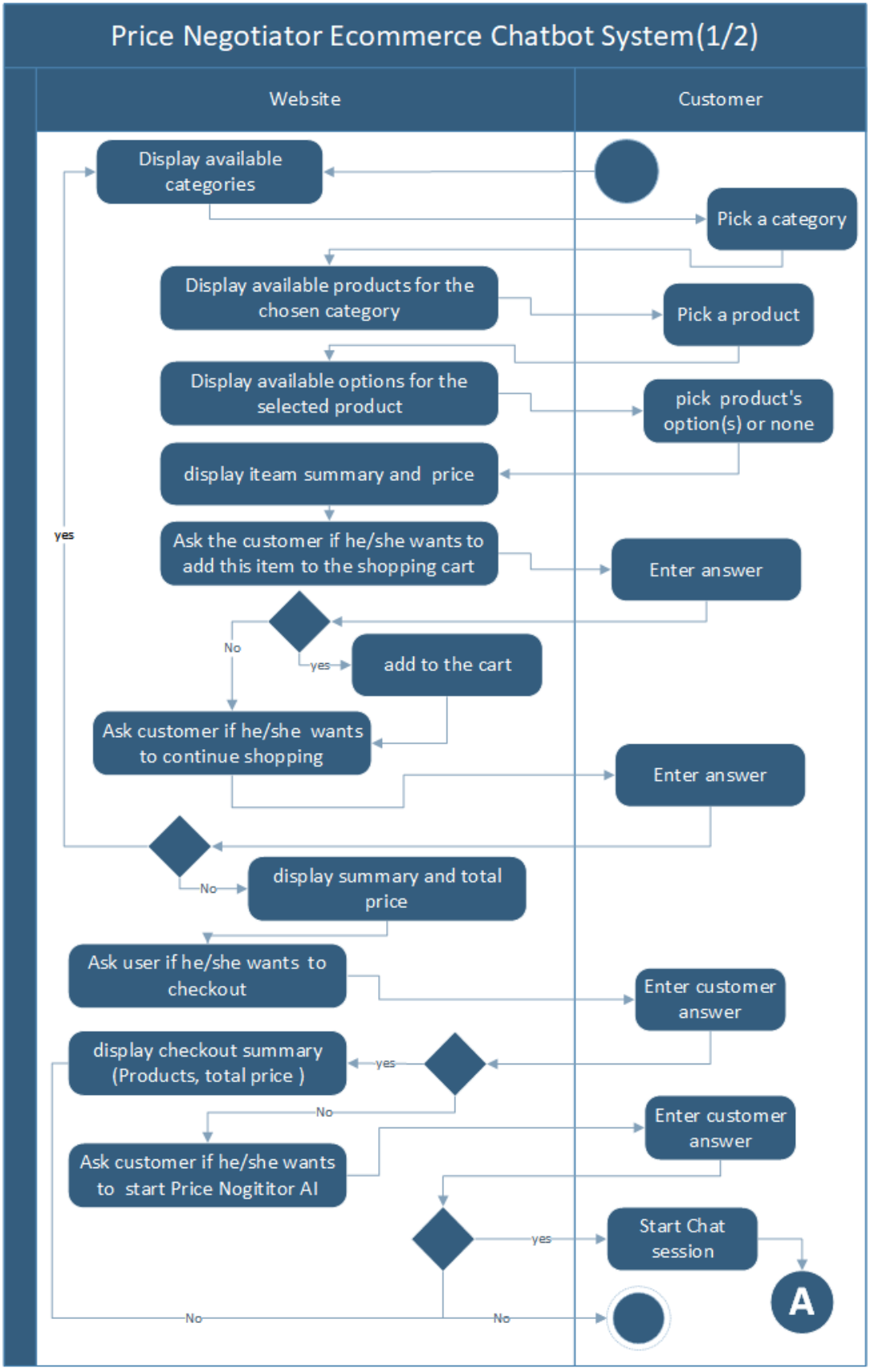
As of now we are not facing any major roadblock or impediment.

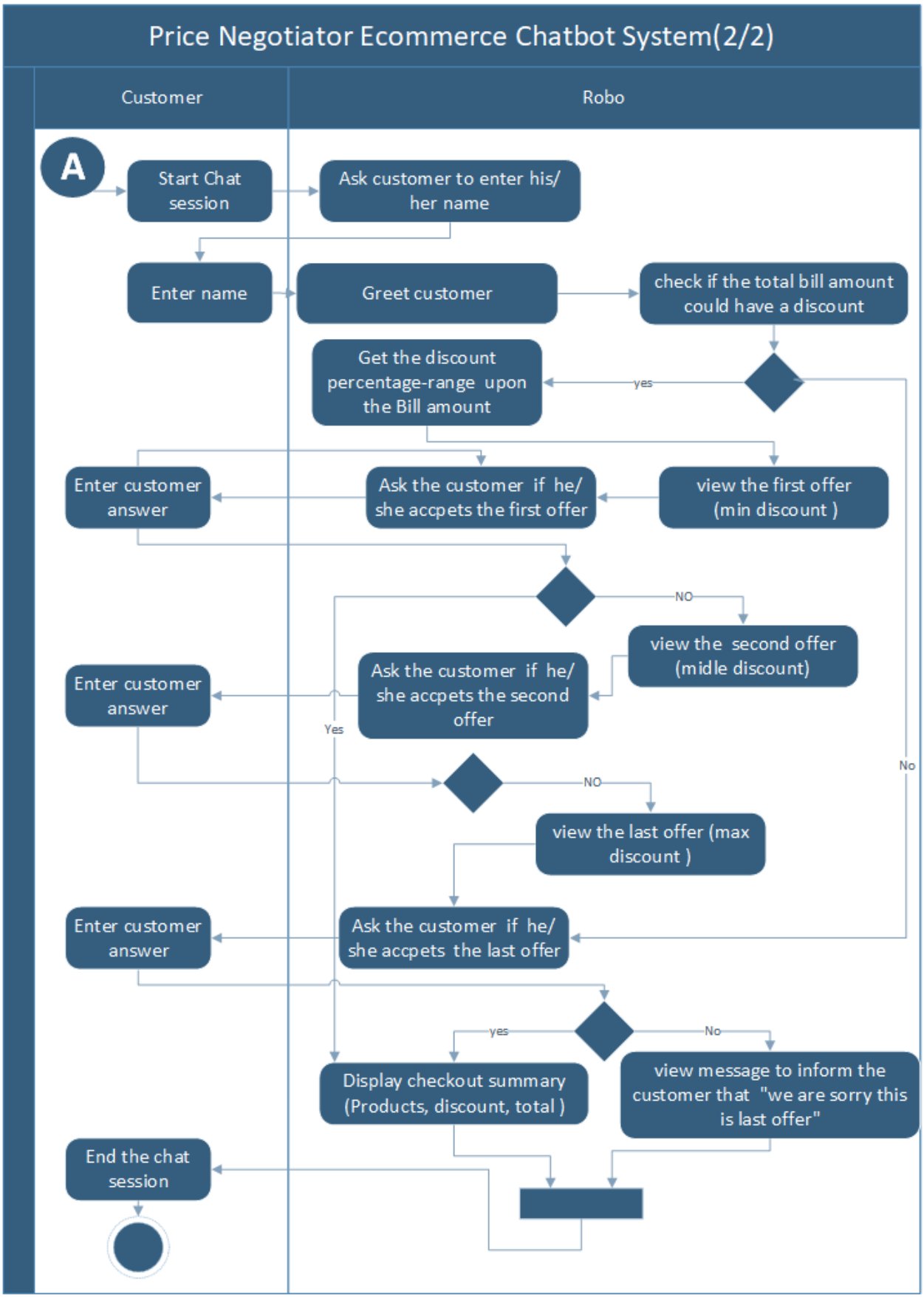
Test Process for quality of work:

At every phase of our project, we test our separate system for functionality testing (e.g., Ecommerce website). We also test code through A/B testing. Currently for separate components we are doing Alpha testing. When the prototype is ready from our end, we will do Beta testing before the final product launch.

During each test(alpha and Beta), unit tests should be made to test each functionality. Integration tests will be executed to check that each function effectively works when embedded and does not negatively impact other system functions. As the system has two sections (e-commerce website and AI price negotiator chatbot), both parts connect to the same database; the integration test for this system is an important stage during the testing phase.

End-user test (acceptance test) should be run before the product's final delivery to ensure that the project meets the user's expectations. The following flowchart shows the desired scenarios that will be assumed to create the end-user use cases test.


Price Negotiator Ecommerce Chatbot System(1/2)
Website
Customer
Display available categories
Pick a category
Display available products for the chosen category
Pick a product
Display available options for the selected product
pick product's option(s) or none
display iteam summary and price
Ask the customer if he/she wants to add this item to the shopping cart
Enter answer
yes
No
yes
add to the cart
Ask customer if he/she wants to continue shopping
Enter answer
No
display summary and total price
Ask user if he/she wants to checkout
Enter customer answer
display checkout summary (Products, total price)
yes
No
Enter customer answer
Ask customer if he/she wants to start Price Nogititor AI
yes
Start Chat session
A
No
No


Price Negotiator Ecommerce Chatbot System(2/2)
Customer
Robo
A
Start Chat session
Ask customer to enter his/ her name
Enter name
Greet customer
check if the total bill amount could have a discount
Get the discount percentage-range upon the Bill amount
yes
Enter customer answer
Ask the customer if he/ she accpets the first offer
view the first offer (min discount)
NO
view the second offer (midle discount)
Enter customer answer
Ask the customer if he/ she accpets the second offer
Yes
No
NO
view the last offer (max discount)
Enter customer answer
Ask the customer if he/ she accpets the last offer
yes
No
Display checkout summary (Products, discount, total)
view message to inform the customer that "we are sorry this is last offer"
End the chat session