Online Shopping Assistant

By

Adrian Nzvimbo

H190589W

HIT400 Capstone project Submitted in Partial Fulfillment of the

Requirements of the degree of

Bachelor of Technology

In

Software Engineering

In the

School of Information Sciences and Technology

Harare Institute of Technology

Zimbabwe

Supervisor

Miss Linda Amos

05/2023

Certificate of Declaration

This is to certify that work entitled “HIT400 Research Topic “ *is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology (Hons) in Software Engineering ,Harare Institute of Technology .It is further certified that no part of research has been submitted to any university for the award of any other degree .*

(Supervisor) Signature…….. Date…….

(Mentor) Signature…….. Date……

(Chairman) Signature…….. Date……..

Contents

Abstract

This project proposes the development of a chatbot that will assist customers in selecting the right products and bridge the gap between online and offline shopping. The chatbot will be equipped with features such as personalized recommendations, virtual try-on technology, and product reviews to improve the customer's shopping experience. The aim of this project is to reduce the risk of returns and increase customer satisfaction by providing customers with a more personalized and interactive shopping experience. The feasibility study considers the technical, economic, and operational aspects of the chatbot.

If successful, this chatbot has the potential to revolutionize the ecommerce industry and transform the way customers shop online.

Preface

This project aims to bridge the gap between online and offline shopping by developing a chatbot that provides personalized product recommendations, virtual try-on capabilities, and product review summaries. The project explores the potential of chatbot technology to enhance the customer shopping experience and revolutionize the ecommerce industry. This document outlines the objectives, methodology, and findings of the project, offering insights into the development and implementation of the chatbot system.

Acknowledgements

I would like to express my sincere gratitude to all individuals who have supported and contributed to the completion of this project. I would like to thank my supervisor, Mrs Linda Amos for their guidance, expertise, and invaluable feedback throughout the project. Additionally, I am grateful to the participants who provided their time and feedback during the testing and evaluation phases. Lastly, I would like to acknowledge the support and encouragement from my friends and family, whose unwavering support has been instrumental in the completion of this project.

Dedication

This project is dedicated to all the individuals who strive to improve the online shopping experience and embrace innovative technologies to create positive change. May this project serve as a small step towards enhancing customer satisfaction and revolutionizing the ecommerce industry.

Declaration:

I, Adrian Nzvimbo hereby declare that this project, titled Online Shopping Assistant, is the result of my own work and research, conducted under the supervision of Linda Amos .All information and sources used in this project have been appropriately cited and referenced. I affirm that this project has not been submitted for any other degree or examination. I take full responsibility for the accuracy and originality of the content presented in this document.

Chapter 1: Introduction

1.0 Introduction

The rise of ecommerce has made online shopping more popular, but the lack of personal interaction with a salesperson and inability to physically try on products can lead to unsatisfactory shopping experiences. To address these challenges, a chatbot is proposed to assist customers in selecting the right products and provide personalized recommendations. By integrating virtual try-on technology, the chatbot can reduce the risk of returns and increase customer satisfaction. This chatbot has the potential to revolutionize the ecommerce industry and provide a more enjoyable and efficient shopping experience.

1.1 Background

Ecommerce has become an increasingly popular way for consumers to shop for products and services. With the rise of online shopping, businesses are looking for ways to provide their customers with a better shopping experience. One of the ways they are doing this is by using chatbots. The ecommerce industry has been experiencing explosive growth in recent years, with online sales expected to reach $4.9 trillion by 2021. As a result, many businesses have been investing in chatbots to automate routine tasks and enhance the customer experience. In fact, according to a study by Grand View Research, the global chatbot market size is expected to reach $1.23 billion by 2025, growing at a compound annual growth rate (CAGR) of 24.3% from 2020 to 20mnnjm25.

Chatbots are computer programs that are designed to simulate conversation with human users. They can be used to provide customer support, answer questions, and even complete transactions. In the context of ecommerce, chatbots are being used to provide customers with a more personalized and efficient shopping experience.

The use of chatbots in ecommerce has several benefits. First, chatbots are available 24/7, which means that customers can get the support they need at any time of the day or night. This is particularly important for businesses that operate in different time zones or have customers in different parts of the world.

Second, chatbots are capable of handling a wide range of customer queries and requests. They can provide customers with product recommendations, help them find the products they’re looking for, and even complete transactions. This not only helps customers find what they’re looking for more quickly, but it also helps businesses increase their sales and revenue.

Third, chatbots can provide businesses with valuable insights into their customers' behavior and preferences. By analyzing customer data and purchase history, chatbots can provide businesses with information about what products are popular, what customers are looking for, and what they're willing to pay. This information can be used to improve product offerings, pricing strategies, and marketing campaigns.

1.2 Problem Statement

The rise of ecommerce has led to an increase in online shopping, however, the inability of customers to physically touch and experience products before purchasing can lead to an unsatisfactory shopping experience. There is a gap between online and offline shopping that needs to be bridged. The main problem is that there is no salesperson available online to assist in deciding the right product for the consumer. Additionally, to get a good idea of a product, customers have to go through numerous reviews, which can be time-consuming and tedious. Another major difficulty is the inability to judge how fashion products like clothing and sunglasses would look on them. To address these issues, a chatbot will be developed that can suggest products to the consumer based on their needs, provide virtual try-on technology, and summarize product reviews. The chatbot aims to bridge the gap between online and offline shopping, reduce the risk of returns, and increase customer satisfaction by providing customers with a more personalized and interactive shopping experience.

1.3 Objectives

- To provide personalized product recommendations to customers based on their needs and preferences
- To help customers virtually experience fashion products, such as clothing and sunglasses, in real-time
- To provide a summary of product reviews to help customers make informed decisions

1.4 Justification

The development of a chatbot that can assist customers in selecting the right products and bridge the gap between online and offline shopping is a critical step towards improving the overall customer experience in the ecommerce industry. The inability of customers to physically touch and experience products before purchasing can lead to an unsatisfactory shopping experience, which can result in a higher risk of returns and decreased customer satisfaction.

The proposed chatbot will be equipped with personalized recommendations, virtual try-on technology, and product reviews, which will help customers make informed decisions about the products they wish to purchase. By using machine learning algorithms and customer data, the chatbot can provide customers with product suggestions based on their needs and preferences. This can result in a more personalized shopping experience, which can increase customer satisfaction and loyalty.

The virtual try-on technology integrated into the chatbot will allow customers to try on fashion products such as clothing and sunglasses in real-time, which can help them make more informed purchasing decisions. This feature can reduce the risk of returns, as customers can get a better idea of how a product will look on them before purchasing it.

The chatbot will also summarize product reviews, making it easier for customers to read and understand the feedback from other customers. This can save customers time and effort, as they no longer have to go through numerous reviews to get a good idea of a product's quality and suitability for their needs.

The development of this chatbot has significant potential to revolutionize the ecommerce industry and transform the way customers shop online. The benefits of the chatbot include increased customer satisfaction, decreased risk of returns, and increased sales and revenue for businesses. With the increasing popularity of ecommerce, investing in a chatbot can be a competitive advantage for businesses, allowing them to provide a more personalized and efficient shopping experience to their customers.

1.5 Proposed tools

● React.js for website [It uses virtual DOM that is much faster and allows to create complex UI easily]

● FastAPI for web server [Fastest python web framework and easy to integrate Machine Learning model to the server]

● PyTorch for implementing virtual try-on [Due to its support for dynamic computational graph]

● NLTK for tasks related to Natural Language Processing. [As it has great pre trained models and corpus of data which makes text processing and analysis pretty quick and easy.]

● Docker and Docker-compose for containerizing the web application and the server [Ensures portability of the application]

●Laptop (2.9Ghz processing speed, 8Gb RAM, 500Gb Hard Disk Drive, Windows Operating system(preferably windows 10))

1.6 Feasibility study

A feasibility study is an analysis that takes all of a project's relevant factors into account including economic, technical, legal, and scheduling considerations to ascertain the likelihood of completing the project successfully. Feasibility study is used to discern the pros and cons of undertaking a project before they invest a lot of time and money into it.

1.6.0 Technical feasibility

Technical feasibility aims at analyzing the Web-based chatbot from a technical perspective, that is whether the required technology is available or not and also the availability of the required resources. Technical feasibility of the proposed system may be done by just looking at the software and hardware available and analyzing them without necessarily building the system. The technology for the proposed system is shown below:

1.6.0.1 Hardware requirements

The hardware and infrastructure that is needed for the implantation of the project. Minimum specification of computer to be used are Intel P4 3 GHz processor speed, 2GB RAM, 500Mb Hard Disk Drive, Windows Operating System (32-bit windows 7) or Ubuntu.

Table 1 hardware required

Description	Quantity	Availability
Laptop/desktop	1	yes

As shown on the table above the hardware required to complete this project is readily available, therefore making the proposed system technically feasible.

1.6.0.2 Software requirements

Table 2 software requirements

Description	Quantity	Availability
Windows Operating System	1	yes
Visual studio code	1	yes
Google chrome	1	yes
Python 3.10	1	Yes
Docker desktop	1	Yes
NLTK	1	yes
React.js	1	yes
pyTorch	1	yes

1.6.1 Economic feasibility

The focus here is on checking if the cost of developing the whole system won't surpass the proposed budget. Cost–benefit analysis showed that it was possible for the development of the application to progress. Most developing software I will be using are open source hence they will be available free of charge. The hardware with required processing power is also already available hence the cost is covered hence the project is economically feasible

1.6.2 Operational feasibility

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. It answers question "will the system work?" and how well the system would fit into the current business structures and once implemented how useable would it be. Operational feasibility is dependent on human resources available for the project and involves projecting whether the system will be used if it is developed and implemented. After analyzing these operational requirements, we see that the proposed system is operationally feasible.

1.7 Project Plan

A project plan is a formal document designed to guide the control and execution of a project. A project plan is the key to a successful project and is the most important document that needs to be created when starting any business project.

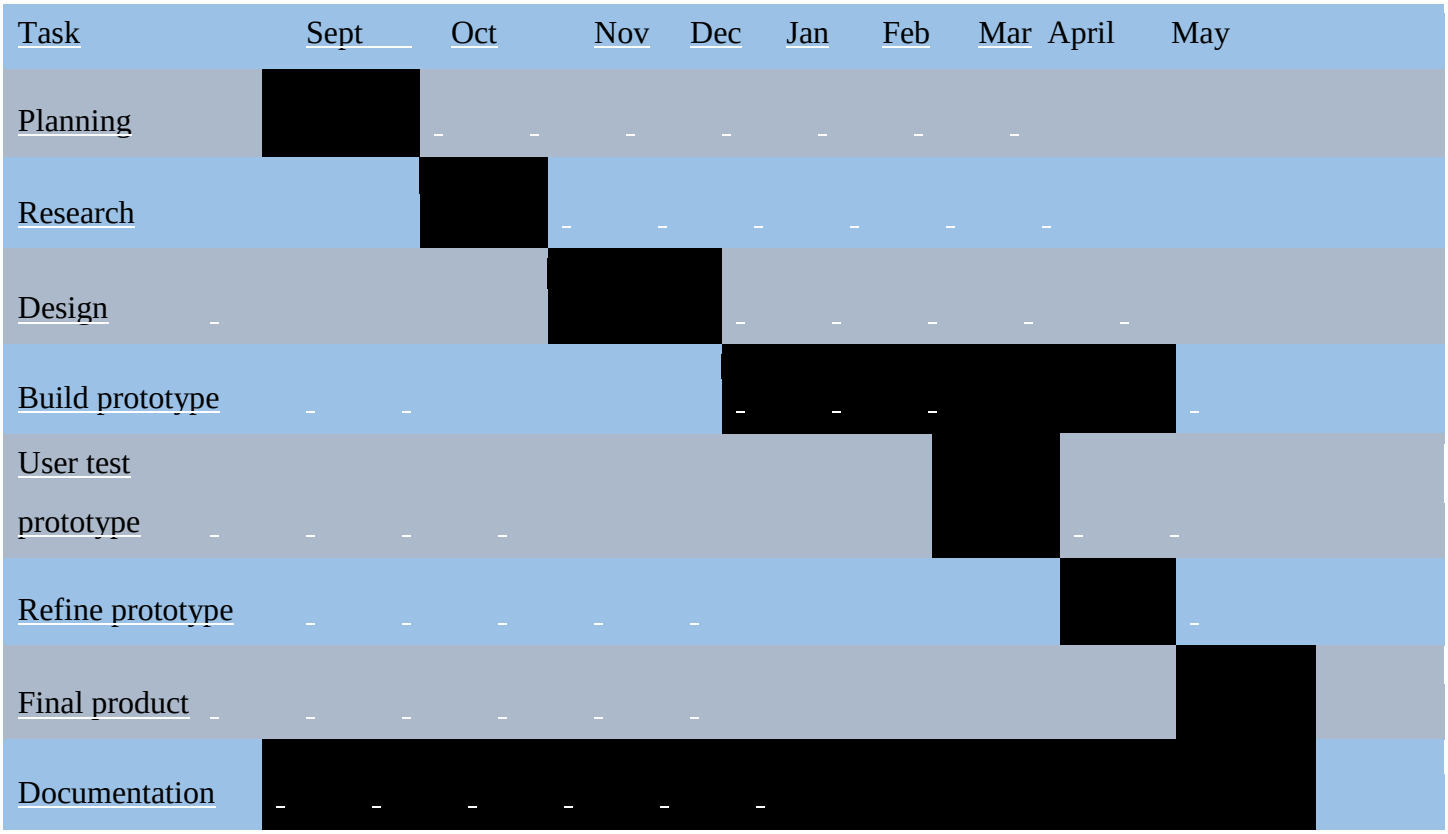
Table 3 project plan

Phase	Duration(days)	Starting date	Ending date
Requirements analysis and definition	35 days	1 September 2022	5 October 2022
System and software design	113 days	6 October 2022	29March 2023
Implementation and unit testing	17 days	1 April 2023	17 April 2023
Integration and system testing	24 days	18 April 2023	11 May 2023

1.7.0 Gantt chart

Gantt charts convey this information visually. They outline all of the tasks involved in a project, and their order, shown against a timescale. This gives you an instant overview of a project, its associated tasks, and when these need to be finished.

Table 4 ganttchart



Chapter 2: Literature Review

2.0 Introduction

As the ecommerce industry continues to grow, businesses are increasingly turning to chatbots to engage with their customers, provide personalized shopping experiences, and drive sales. Chatbots leverage the latest advancements in machine learning and natural language processing to offer customers a seamless and intuitive shopping experience. However, the implementation of an ecommerce chatbot requires careful planning and design to ensure that the chatbot is effective, user-friendly, and meets the needs of the business and its customers. In this literature review, we will explore the existing literature on ecommerce chatbots to gain a better understanding of their potential benefits and key design considerations.

2.1 Related Work

"Improving the Shopping Experience with Chatbots" by Mario Kluser and Bjorn Schuster: This paper discusses the potential of chatbots to improve the shopping experience for customers in the ecommerce industry. The authors argue that chatbots can provide personalized recommendations, simplify the shopping process, and reduce the risk of returns. They also suggest that chatbots can be used to collect customer data and improve business intelligence.

"Virtual Fitting Room: A Key Technology for Enhancing Customers' Online Apparel Shopping Experience" by Jialiang Liu, Chaojun Wu, and Hua Zhang: This paper focuses specifically on the use of virtual fitting room technology to enhance the online shopping experience for customers purchasing apparel. The authors argue that virtual fitting rooms can provide a more interactive and personalized shopping experience, as well as reduce the risk of returns. They also suggest that virtual fitting rooms can increase customer engagement and loyalty.

There is a significant body of research on ecommerce chatbots and their potential impact on customer engagement and sales. For example, a study by eMarketer found that 48% of online shoppers would prefer to interact with a chatbot rather than a human, citing convenience and 24/7 availability as the primary

reasons for their preference. Another study by Grand View Research predicts that the global chatbot market will reach $1.23 billion by 2025, driven by the increasing adoption of chatbots in ecommerce.

In terms of design considerations, researchers have identified several key factors that contribute to the success of an ecommerce chatbot. These factors include conversational flow, natural language processing, and user interface design. For example, a study by IBM found that chatbots that use natural language processing to understand the intent of the user and provide relevant responses are more effective than those that rely on rule-based systems. Similarly, a study by Nielsen Norman Group found that chatbots with a clear and intuitive user interface design are more likely to be adopted by users.

In addition, there are several popular ecommerce chatbots that have gained traction in the market. Some of the most popular ecommerce chatbots include:

Amazon's Alexa: Amazon's Alexa is a voice-activated chatbot that allows users to browse and purchase products through voice commands. Alexa can also make recommendations based on the user's purchase history and shopping behavior.

Google Assistant: Google Assistant is a chatbot that allows users to search for products, make purchases, and get personalized recommendations. Google Assistant is also integrated with Google's shopping platform, allowing users to easily find and purchase products.

Sephora's Virtual Artist: Sephora's Virtual Artist is a chatbot that allows users to try on makeup virtually and get personalized product recommendations. The chatbot uses augmented reality technology to create a realistic makeup experience.

eBay's ShopBot: eBay's ShopBot is a chatbot that allows users to search for products, get personalized recommendations, and make purchases directly through the chatbot. The chatbot also provides real-time updates on shipping and delivery.

Conclusion

Ecommerce chatbots have the potential to revolutionize the way that businesses engage with their customers and drive sales. However, the implementation of an effective chatbot requires careful planning, design, and execution. By leveraging the existing literature on ecommerce chatbots and studying popular chatbots in the market, businesses can gain a better understanding of the potential benefits and key design considerations of chatbots. This knowledge can be used to develop highly effective chatbots that meet the needs of the business and its customers.

Chapter 3: Analysis

3.0 Information Gathering Tools

3.0.0 Information Gathering Techniques

To determine the requirements of an application, information gathering must be carried out in order to know what the end user wants the system to be able to do and what the user hopes to achieve through using the application. There are many information gathering techniques, these include :

3.0.1 Questionnaires/Surveys:

this method of information gathering is efficient in collecting data of a large group quickly. Surveys consist of a range of questions such as closed ended questions i.e. where the user is prompted to select a choices/boxes. Open-ended questions allow user to express their opinion or provide whatever feedback they feel is suitable. Other types of questions can be used such as ranges and scales which can be used to gauge how strongly a user may feel. Surveys can be distributed through email, online, paper-based and telephone surveys. The questions used must be clear and precise. Surveys should be tested on a small focus group of user to get feedback on the survey this ensures the questions are unambiguous and unbiased (Jonathan Lazar, J. 2001).

3.0.2 Interviews:

interviews are a structured approach to information gathering. They are conducted face to face to allow the interviewer to ask more detailed open-ended questions and follow up questions to elaborate on certain points to capture the user needs. This method allows the interviewer to take control of the interview and direct it in a way so they can focus on more important aspects of the proposed system. This ensures the most important aspects of the system development are identified within a limited time constraint. One interview tactic interviewers use to determine they are gathering as much information as possible is to ensure the interviewee feels comfortable and relaxed in the interview environment. Open ended questions can be asked resulting in more detailed responses. Follow up questions can be used to ensure a previous response is fully understood. The interviewer can take notes during his process to reflect on and analyze the content later (Graham Oakes, G. 2008).

3.0.3 Discovery Prototyping:

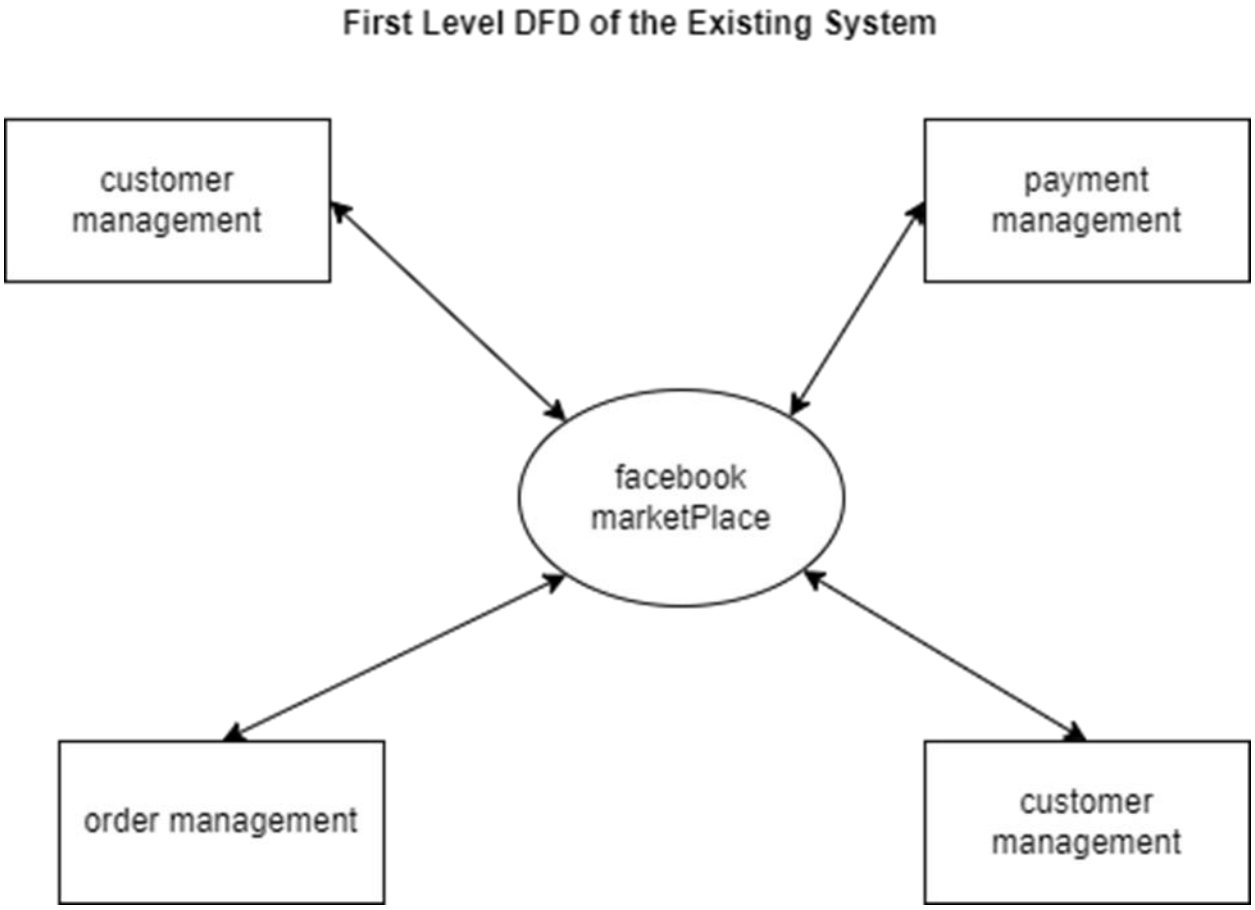
this involves developing a low fidelity throwaway prototype of the proposed system. This approach can be useful in situations where the equirements are vague. This allows the end user to experience and get an impression of the system and determine whether it meets their needs. Developers will meet with the user to

identify the system requirements and investigate what requirements that can be established. Once the requirements are identified the prototype is developed and given to user to assess whether or not it meets the requirements previously discussed. This gives the user an opportunity to be more involved in the project and provide feedback on the prototype that will be used to identify the requirements for the final software to be developed. The prototyping approach allows the developer to get a deeper understanding of the requirements through continuous communication and each prototype iteration or cycle (Tata McGraw-Hill PublishingInstructional Software Research and Development, 2007).

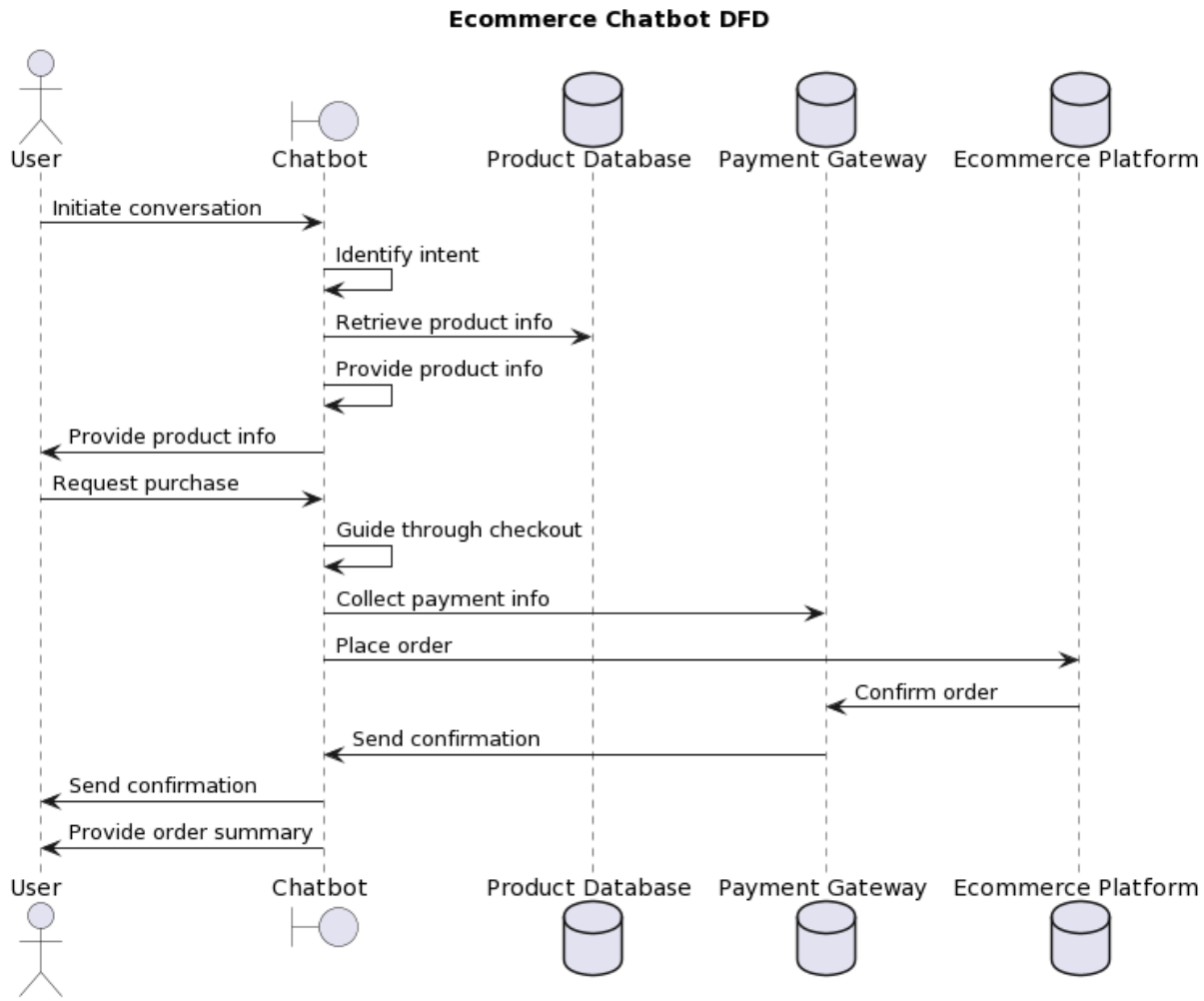
Description of system

3.1 Data analysis –Using UML context diagrams

A UML context diagram is a high-level view of a system that shows its external entities and the interactions between them. It is used to illustrate the system's boundaries and to provide a general understanding of the system's functionality and relationships with other systems or entities.



3.2 DFD of existing system



3.3 Data Analysis

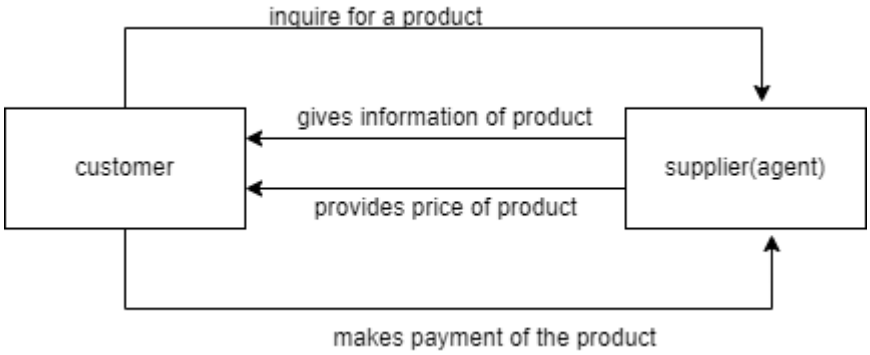
The analysis phase was a crucial activity and the goal was to provide a clear understanding of how the current system works, and its shortfalls. In this chapter, efforts were made to outline possibly all the requirements that the proposed system had to meet. Of concern was the analysis of existing systems, diagrammatic representation, strengths and weaknesses and come up with possible alternatives. The study helped to produce a system that addressed the shortcomings of those previous existing system.

3.4 Analysis of existing system

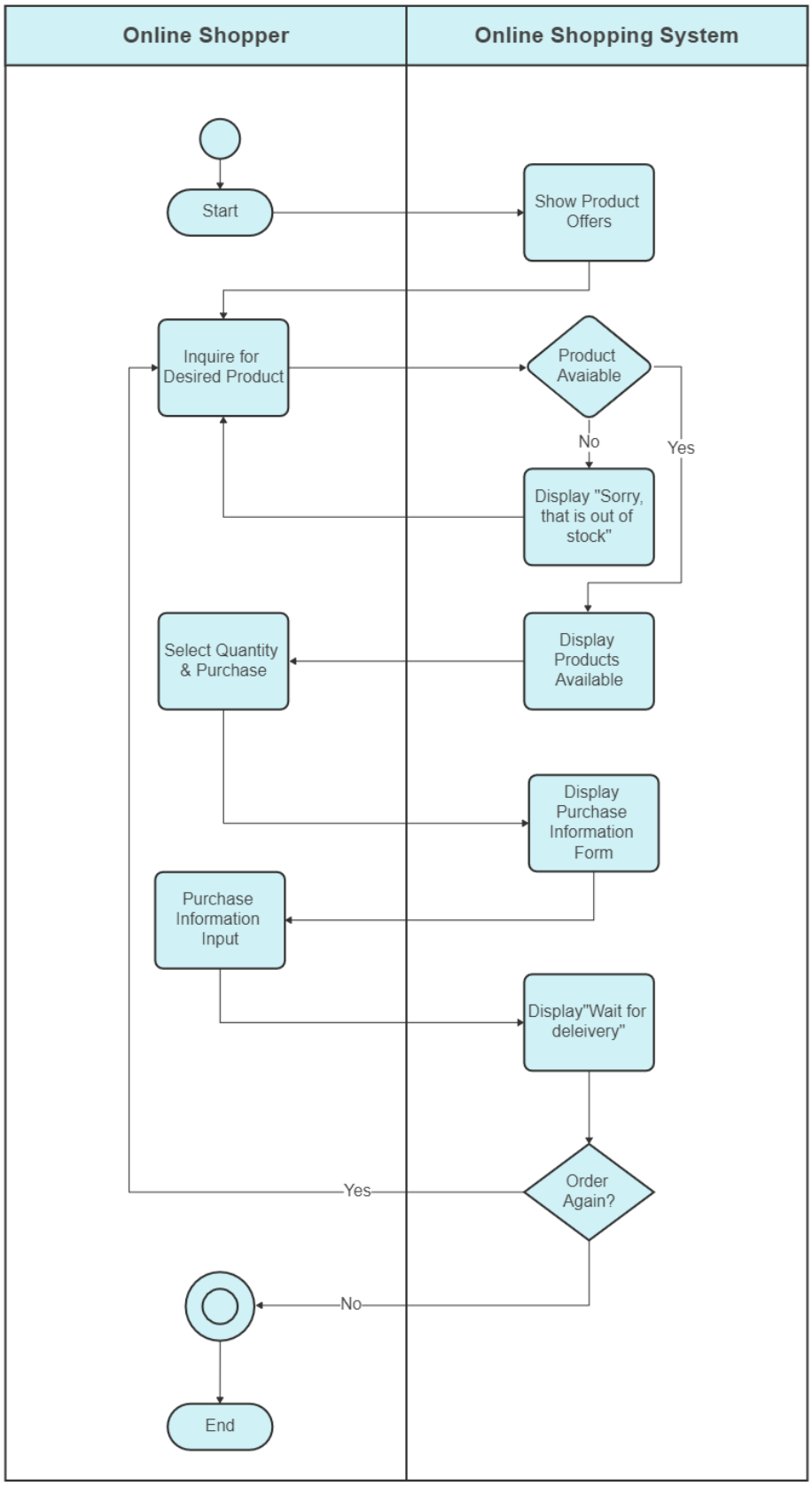
Most organizations and individuals advertise their products and services on the internet and on different social media such as Facebook, twitter, whatsapp, instagram etc. So when a customer or client needs a service or product provided by the organization they respond to the advert. When a customer makes an inquiry and agent responds to that inquiry.

3.5 Data flow diagram of existing system

A data flow diagram (DFD) is a graphical representation of the flow of data through a system. It shows how data is input, processed, and output by a system, and how it is stored or transmitted between components of the system. DFDs are used to model the process flow and data movement within a system, and to identify potential bottlenecks or areas of optimization.



3.6 Activity diagram of existing system



3.7 Functional Analysis of Proposed System-Functional and Non-functional Requirements

3.7.0 .Functional and Non-Functional Requirements

Functional and Non-functional requirements are identified through the analysis of the data collected from the survey. Functional requirements are the features and functionality that the system must have or be able to perform whereas non-functional requirements define the manner or characteristic the system must have such as:

Performance, usability, modifiability, maintainability, security, scalability, reliability, availability, configurability and design constraints.

3.7.1 Functional Requirements

they shall be able to chat with Snoop (chatbot), request for items available in the stock database, pay for items via the chatbot platform, have a they want to buy, and know the price of the items they are enquiring about or wish to purchase.

Users will be able to converse with the Chat bot text commands and it will understand what the user is saying through NLP (Natural Language Processing) to analyse the context of a conversation. It can identify the intent of a question to provide an accurate answer and suggest options to confirm or resolve the issue.

3.7.2 Non-Functional Requirements

Non-functional requirements provided by the system include quality of service features such as efficiency and quality. The system should be fast for users to participate and improve user experience.

Non-functional requirements include the following:

(1) Security: Unauthorized users should have no access to the system.

(2) Usability: he proposed system should be easy for the user to operate, enter data, and interpret the output.

(3) Scalability: The system should always perform adequately regardless of updates.

(4) Compatibility: The proposed system should be compatible with all web browsers.

Chapter 4: Design

4.0 Introduction

The requirements and the collection of the analysis phase in chapter three produced both data requirements and functional requirements which are in turn used as a source for the design phase. This phase entails the process of designing the architecture, components, interfaces, and other characteristics of the system. At this phase things to be address are the problem definition that is, the actual solution to the problem at hand will be realized. The main goal of this phase is to develop the concept and basic framework of the chatbot

4.1.0 System Inputs

The chat bot will have a few inputs

- Customer requests for product recommendations or virtual try-on
- Customer preferences and needs
- Customer queries and questions
- Product and customer data from databases

4.1.1 System processing

- Natural language processing to understand customer requests and queries
- Analysis of customer data and preferences to provide personalized product recommendations
- Use of 3D models to allow virtual try-on of fashion products
- Summarization of product reviews to help customers make informed purchase decisions

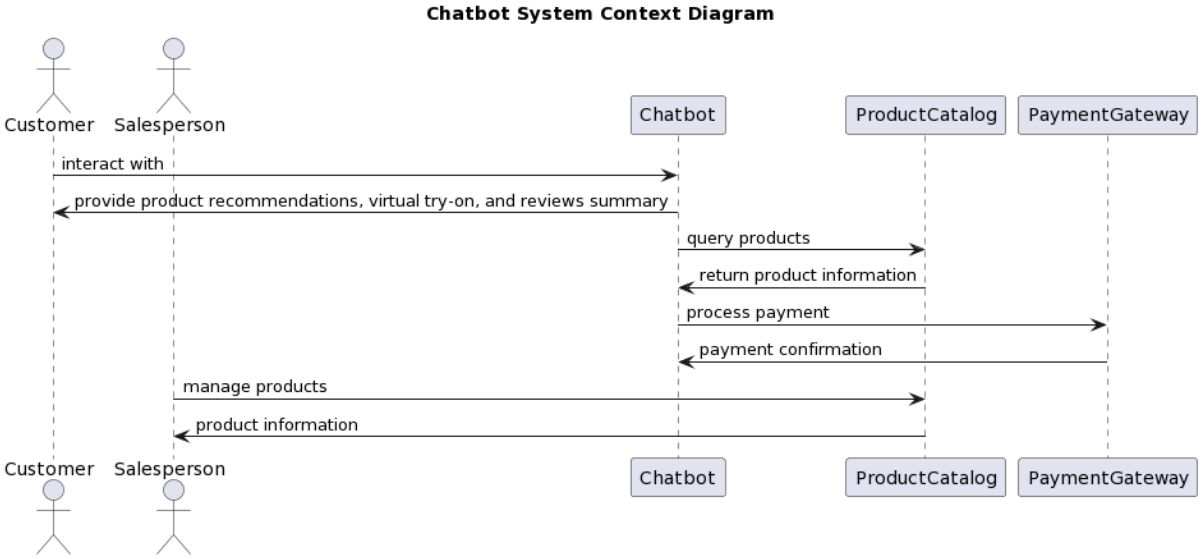
4.1.2 System output

- Personalized product recommendations for customers
- Virtual try-on experience for customers to see how products look
- Summary of product reviews to help customers make informed decisions

4.2 Design diagrams

Design diagrams are a set of graphical notations used by software engineers to represent various aspects of software design. These diagrams help software engineers to visualize, communicate, and document software designs.

4.2.1 UML CONTEXT DIAGRAM



4.2.2 Data Flow Diagrams (DFD)

It is a graphical representation of a system that shows how data flows through the system, including how it is input, processed, and output.

A DFD consists of a set of interconnected bubbles, each of which represents a process that transforms data. The data flows between processes are shown as arrows, and the data stores where data is stored are represented as rectangles.

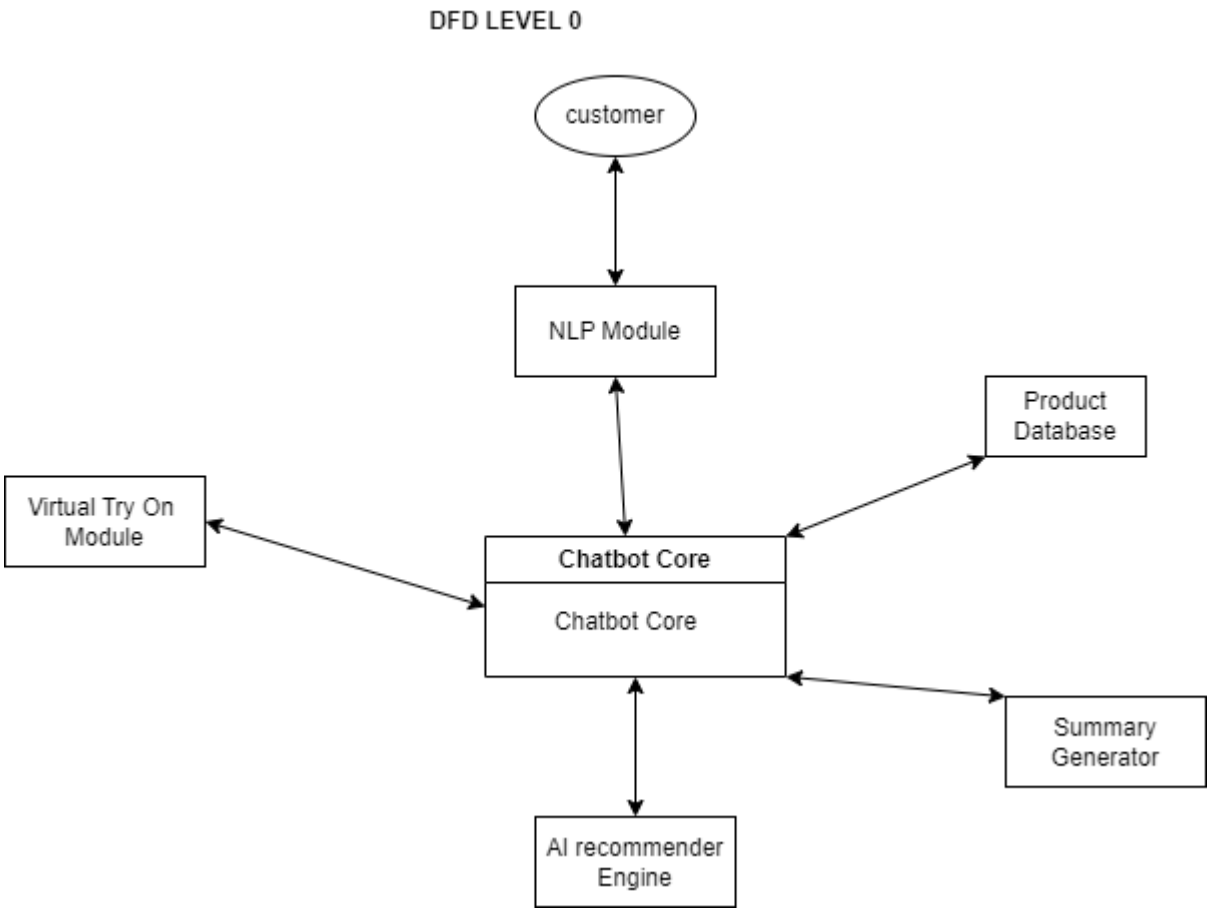
DFDs are typically drawn at different levels of detail, with the most abstract level showing the high-level processes and data flows and the lower-level diagrams showing more detailed processes and data flows.

DFDs are useful for identifying the data requirements of a system, the processes that operate on that data, and the relationships between those processes. They are also helpful in identifying areas for improvement and optimization in a system.

4.2.2.0 DFD LEVEL 0

A DFD Level 0, also known as a Context Diagram, is the highest-level DFD that represents a system as a single process with its interactions with external entities. The Context Diagram provides an overview of the system and its environment, without going into the details of the internal processes.

In a DFD Level 0, the system is represented as a single process, and the external entities that interact with the system are represented as squares or rectangles. The data flows between the system and the external entities are represented as arrows.



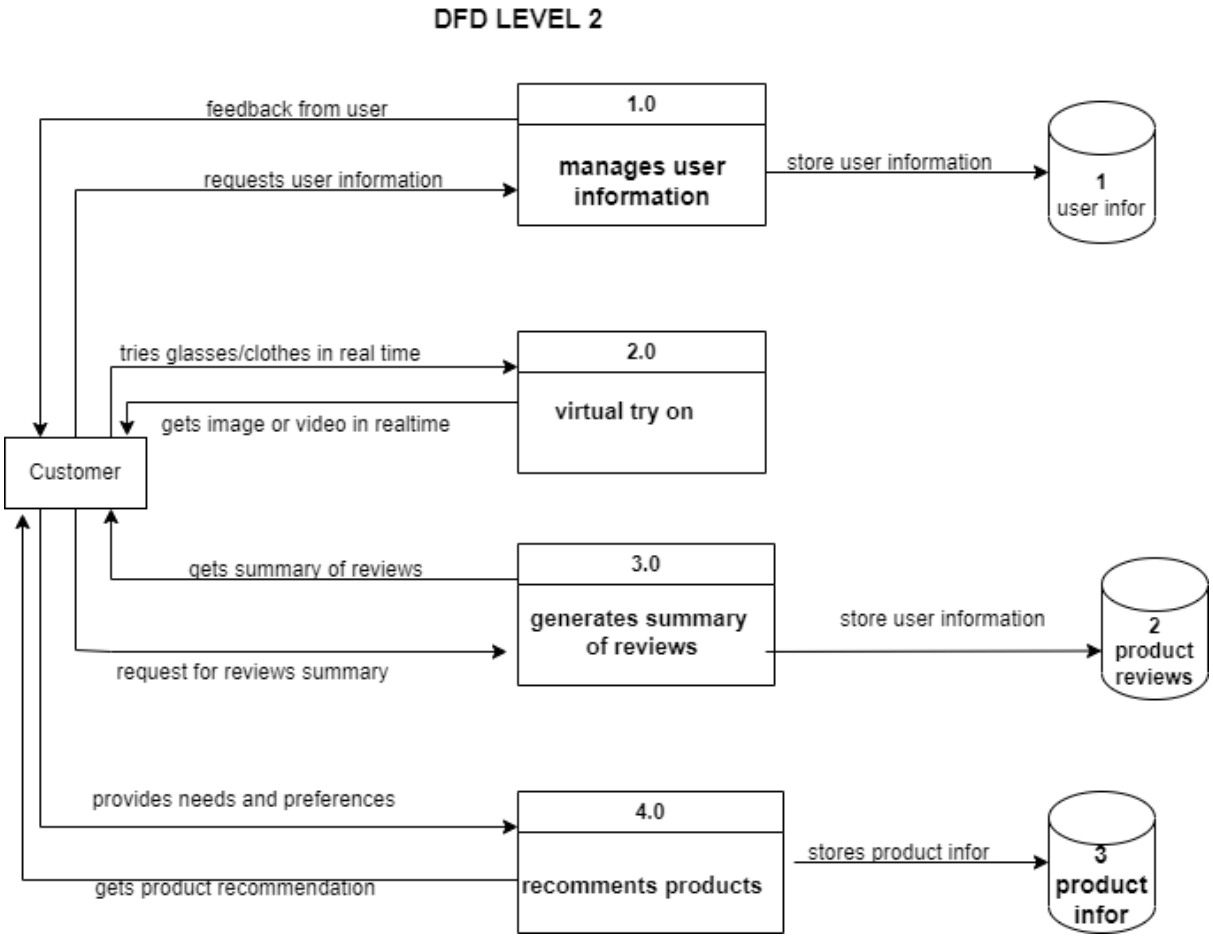
4.2.2.1 DFD LEVEL 1

A DFD Level 1 is a more detailed version of the Data Flow Diagram that represents a system as a set of processes that are interconnected by data flows. A Level 1 DFD breaks down the system into its functional components and shows how data flows between these components.

In a Level 1 DFD, each process shown in the Context Diagram (Level 0) is decomposed into a set of subprocesses that perform specific functions. These subprocesses are represented as bubbles, and the data flows between them are shown as arrows. Data stores are also represented in the Level 1 DFD, usually as rectangles.

The purpose of a Level 1 DFD is to provide a more detailed view of the system and its processes than the Context Diagram. It helps to identify the functions that the system performs and the relationships between these functions and the data they process.

Level 1 DFDs are usually created for each process shown in the Context Diagram, and they provide a more detailed view of the system. However, they are still relatively high-level and do not show the detailed logic of individual processes or the data structures used by the system.

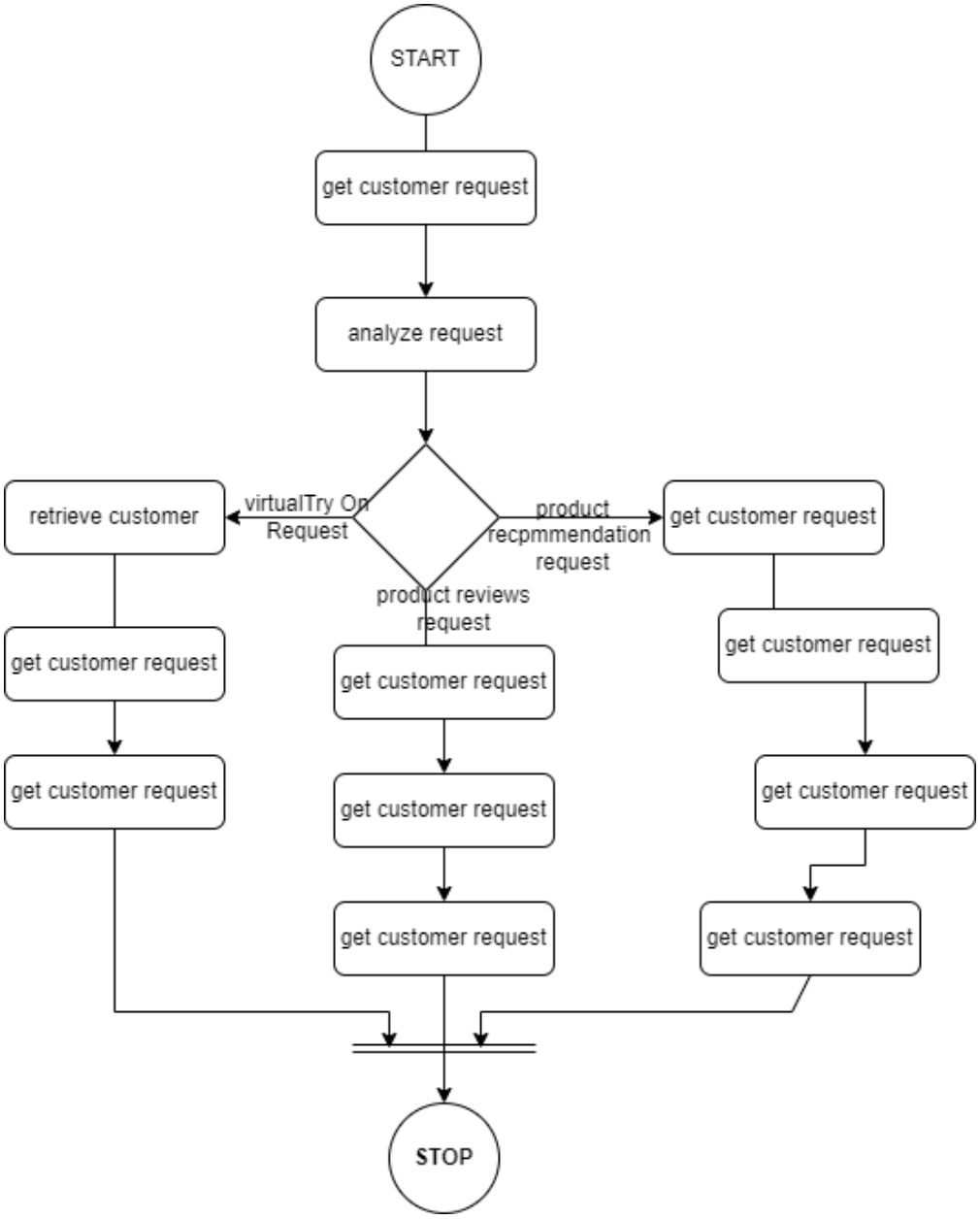


4.2.3 Activity Diagram

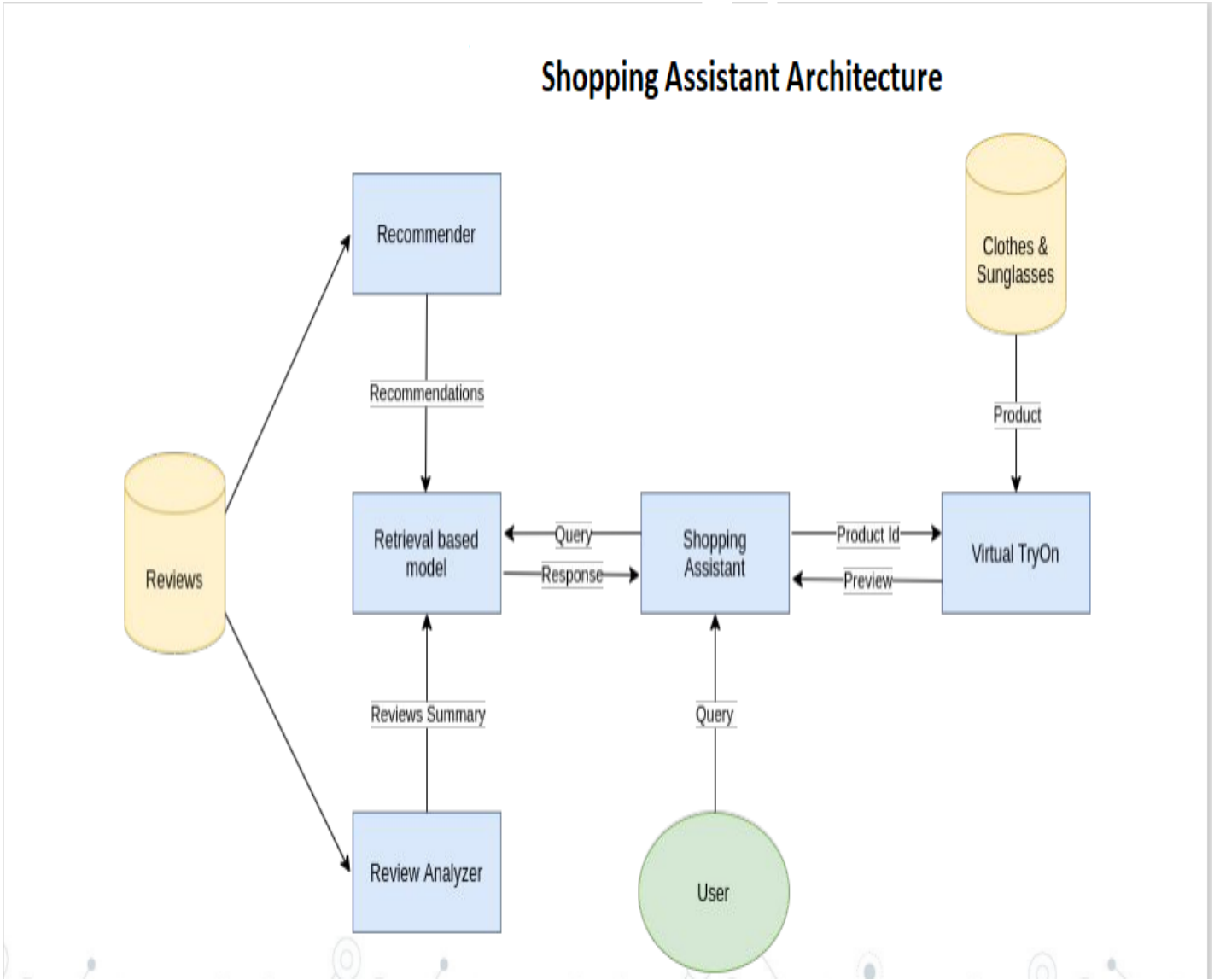
An activity diagram is a type of UML (Unified Modeling Language) diagram used in software engineering to model the flow of activities and actions within a system. It is a graphical representation of a workflow that shows the sequence of activities, decisions, and conditions that are involved in a process.

In an activity diagram, the activities are represented as rounded rectangles, and the transitions between activities are represented as arrows. The decisions in the workflow are represented as diamonds, and the conditions that determine the next activity or decision are shown on the outgoing arrows from the diamond.

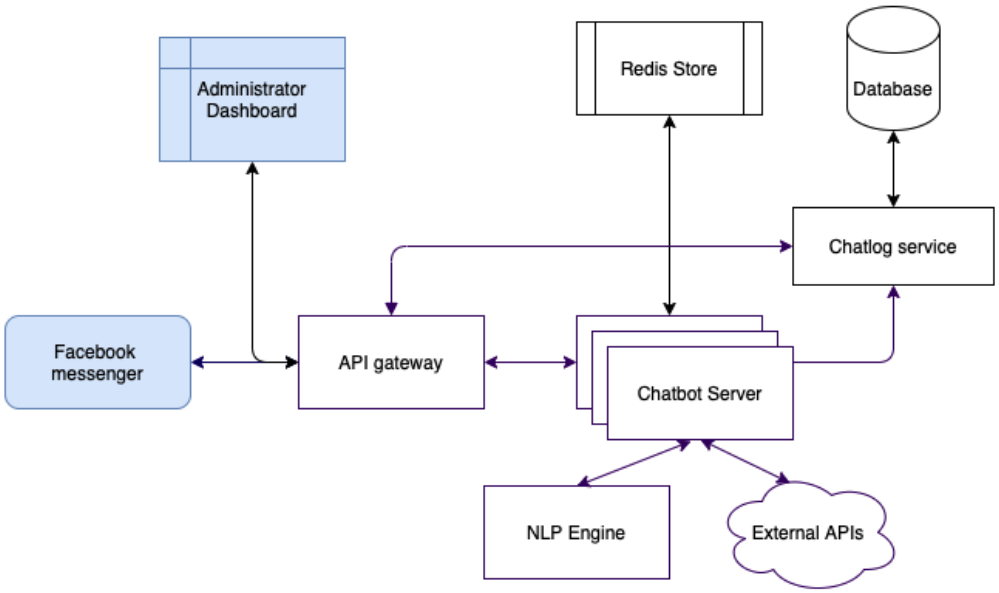
Activity diagrams can be used to model a wide range of processes, such as business processes, software processes, and system processes. They are particularly useful in modeling complex processes that involve multiple decision points and parallel activities.


START
get customer request
analyze request
retrieve customer
virtualTry On Request
product recpmmendation request
get customer request
product reviews request
get customer request
get customer request
get customer request
get customer request
get customer request
get customer request
get customer request
get customer request
STOP

4.2.4 Shopping Assistant Workflow or Architecture



Chatbot System Architecture



4.2.5 Entity Relationship Diagram

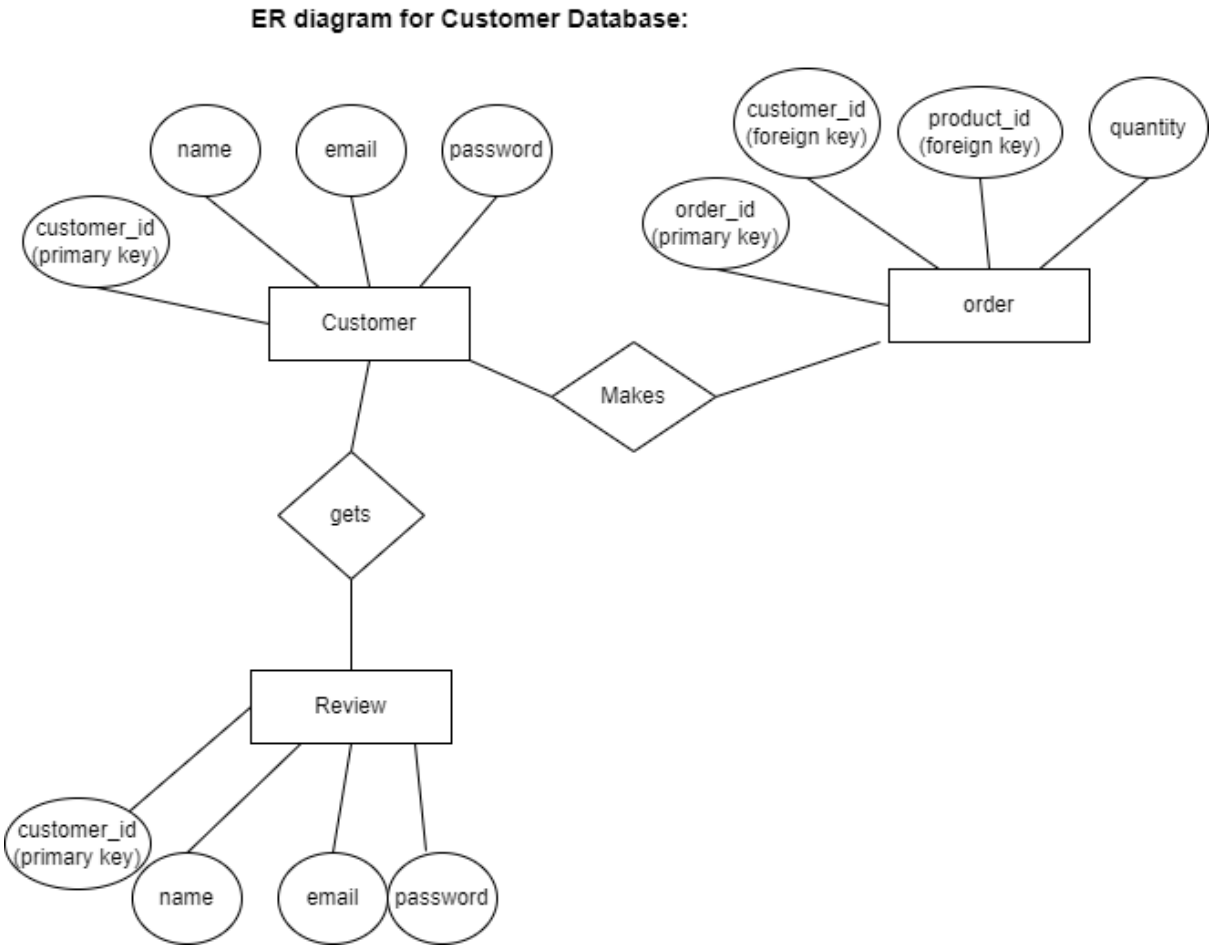
An Entity Relationship Diagram (ERD) for a chatbot would show the relationships between the entities that are involved in the chatbot's operation. The relationships would include the various types of data that the chatbot needs to operate, such as user information, chatbot responses, and chat histories.

ER diagram for Customer Database:

[An ER diagram showing the Customer Database with entities and their attributes.]

Entities:

- Customer: Stores information about the customers who use the e-commerce store
- Order: Stores information about the orders placed by customers
- Review: Stores information about the product reviews written by customers



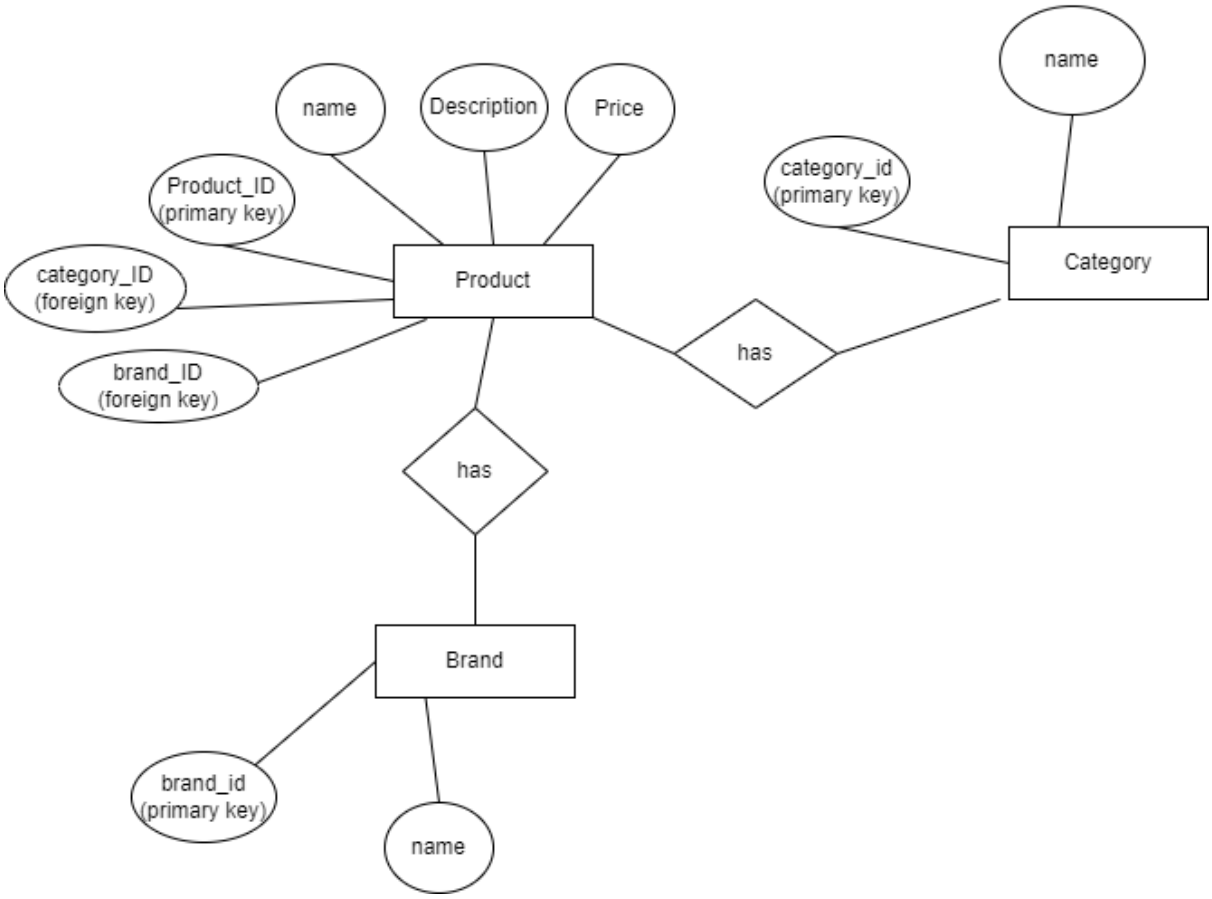
ER diagram for Product Database:

[An ER diagram showing the Product Database with entities and their attributes.]

Entities:

- Product: Stores information about the products offered by the e-commerce store
- Category: Stores information about different categories of products
- Brand: Stores information about different brands of products

ER diagram for Product Database:



4.2.6 Normalized Databases

We can start with the first normal form (1NF) which requires each table to have a primary key and each attribute to be atomic (indivisible).

Customer Table:

ID	Name	Email	Password	Address

| | | | | |

Product Table:

ID	Name	Description	Price	Category

Review Table:

ID	Customer_ID	Product_ID	Rating	Comment

Next, we move to the second normal form (2NF) which requires all non-key attributes to be fully dependent on the primary key.

Customer Table:

ID	Name	Email	Password	Address

Product Table:

ID	Name	Description	Price	Category

Review Table:

ID	Customer_ID	Product_ID	Rating	Comment

| | | | | |

Next, we move to the third normal form (3NF) which requires that all non-key attributes are dependent only on the primary key and not on other non-key attributes.

Customer Table:

ID	Name	Email	Password	Address

Product Table:

ID	Name	Description	Price	Category

Review Table:

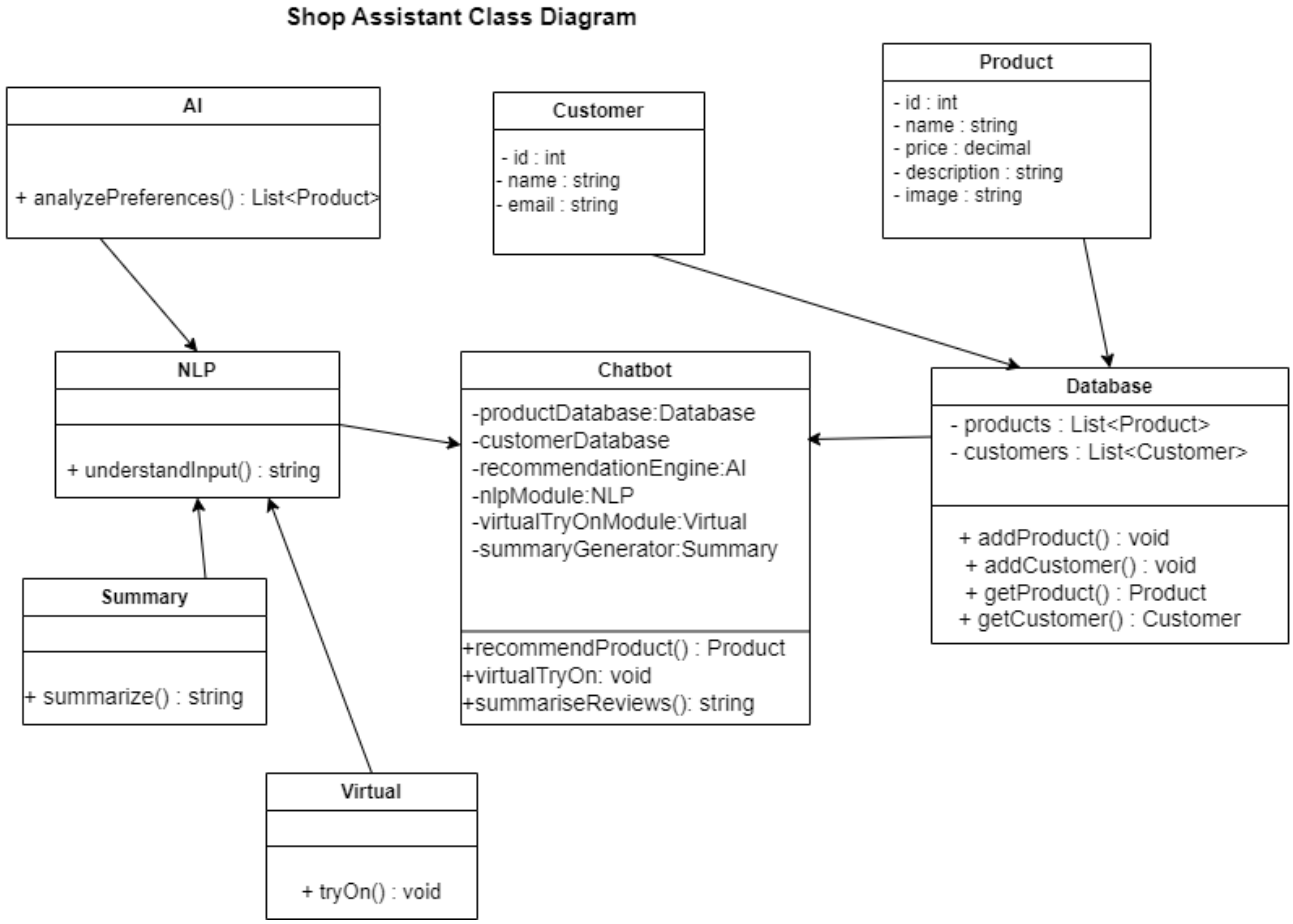
ID	Customer_ID	Product_ID	Rating	Comment

This is the final normalized database design for the chatbot system.

4.2.7 Class Diagram

A class diagram is a type of UML (Unified Modeling Language) diagram that represents the structure and relationships of classes in an object-oriented system. It shows the classes, their attributes, operations, and the relationships between them.

In a class diagram, a class is represented as a rectangle, with the class name at the top. The attributes of the class are listed below the class name, and the operations or methods of the class are listed below the attributes. The relationships between classes are represented as lines connecting the classes.

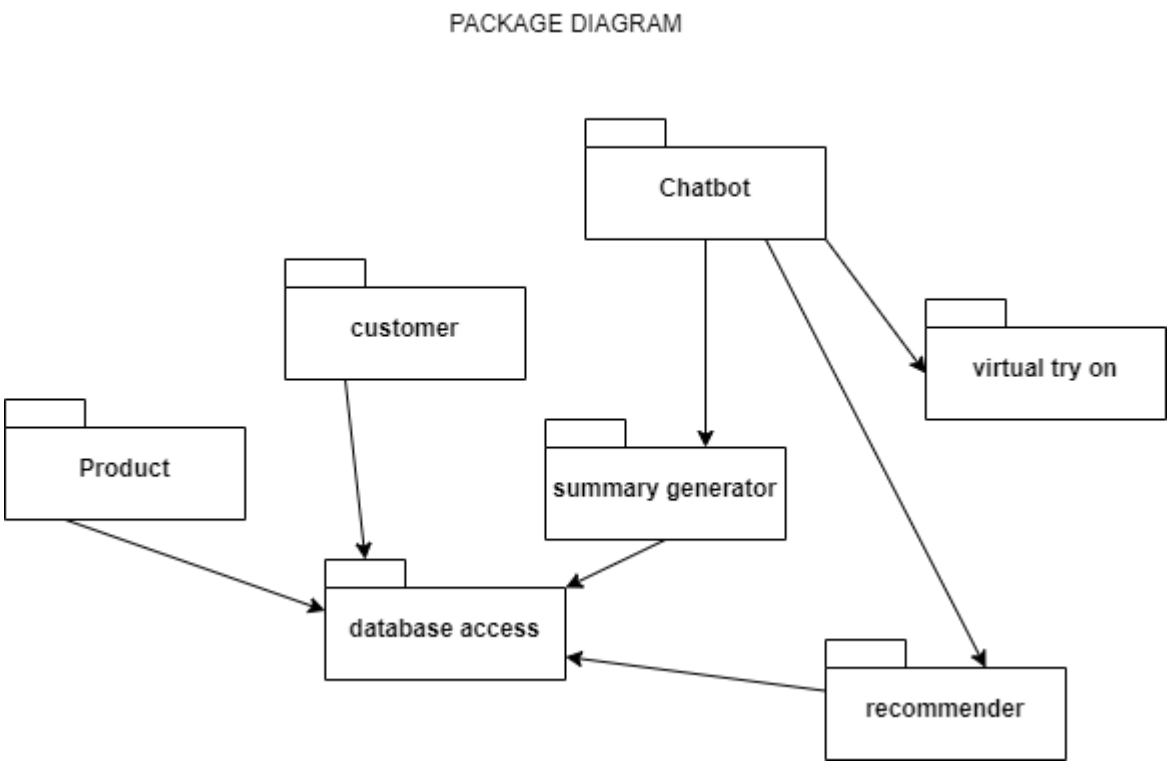


4.2.8 Package Diagram

A package diagram is a type of UML (Unified Modeling Language) diagram that shows the organization and dependencies of packages in a system. It is used to illustrate the high-level structure of a system and how the components within the system are organized into packages.

In a package diagram, the packages are represented as rectangles, with the package name at the top. The dependencies between packages are represented as arrows, with the arrow pointing from the dependent package to the package it depends on.

Package diagrams show the relationships and dependencies between packages within a system, which can help to ensure that the system is well-organized and modular. They are useful for identifying potential issues or inefficiencies in the system's design, such as circular dependencies or overly complex package structures.

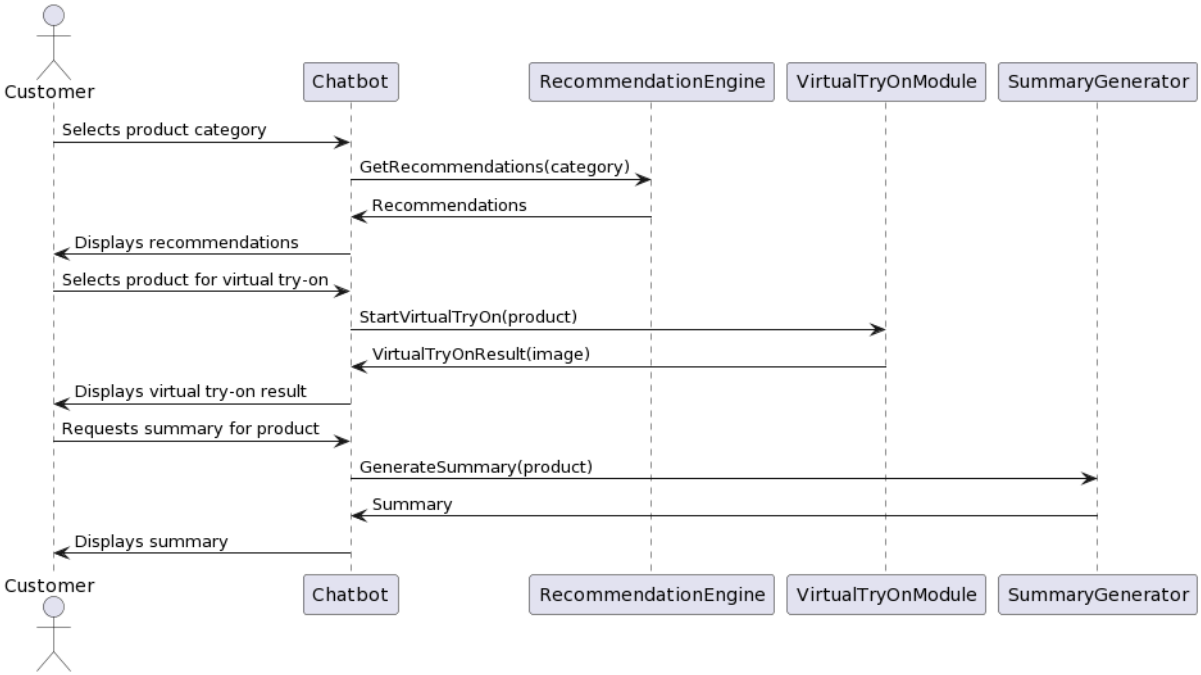


4.2.9 Sequence Diagram

A sequence diagram is a type of UML (Unified Modeling Language) diagram that represents the interactions between objects or components in a system over time. It shows the sequence of messages exchanged between the objects, along with the lifeline of each object.

In a sequence diagram, the objects are represented as vertical lines, called lifelines. The messages exchanged between the objects are represented as horizontal arrows, with the message name and any parameters shown above the arrow.

The sequence diagram shows the order in which the messages are sent and received, along with any conditions or loops that occur during the interactions. It can also show the concurrent interactions between objects that occur simultaneously.

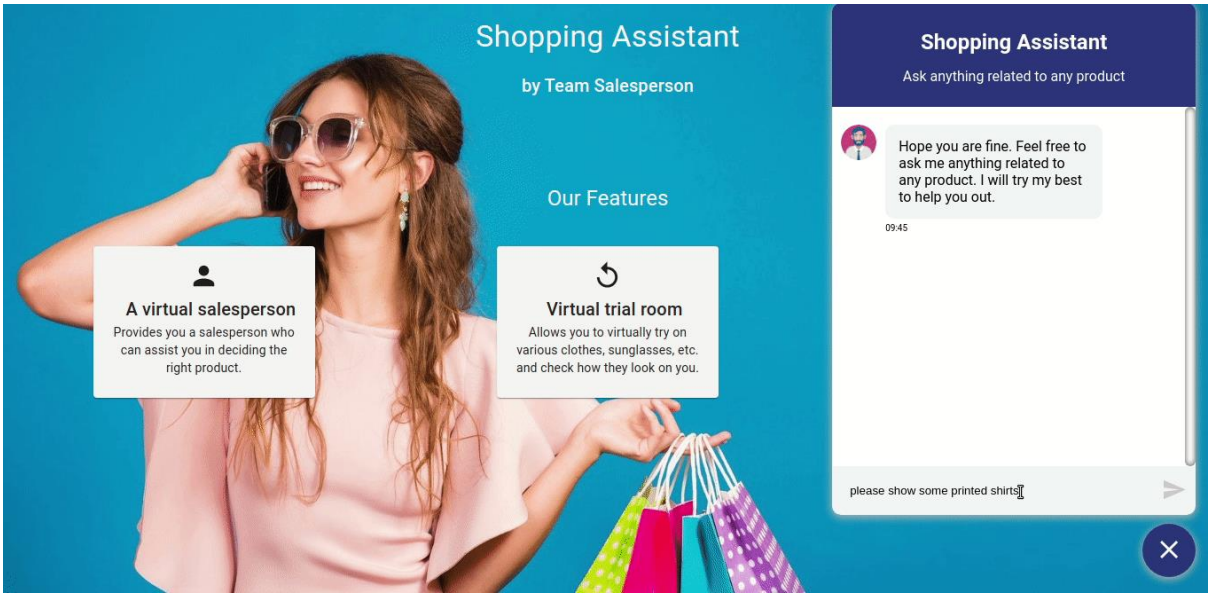


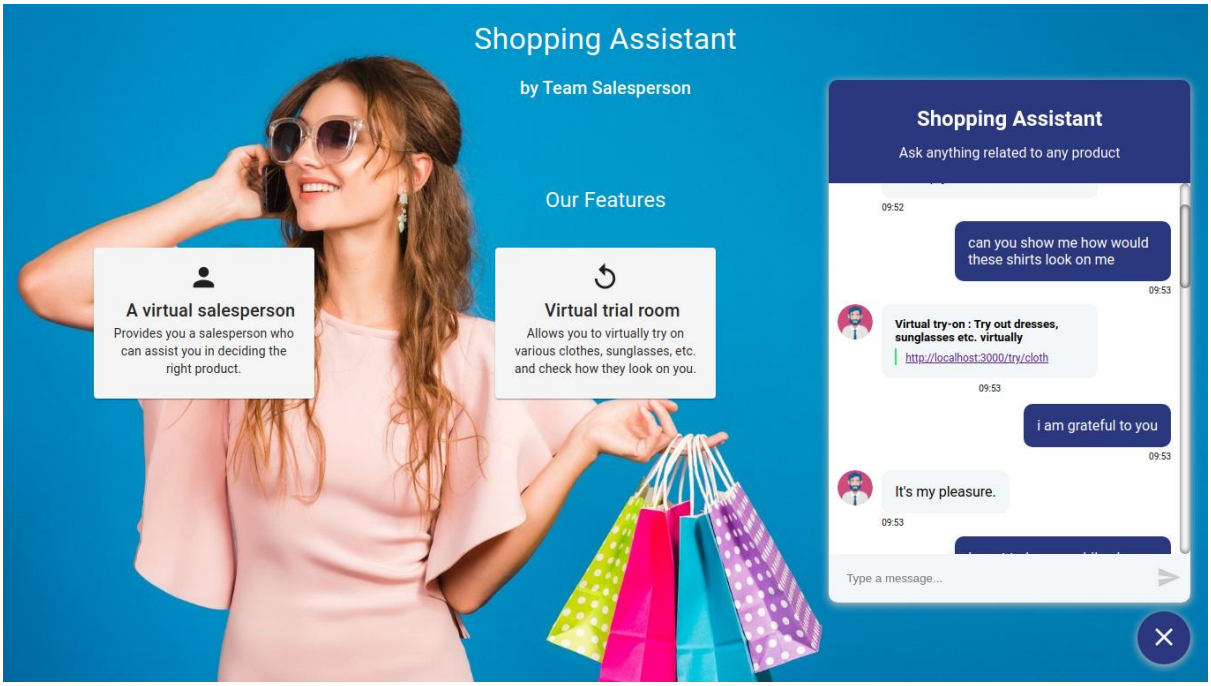
4.2.10 Pseudo Code

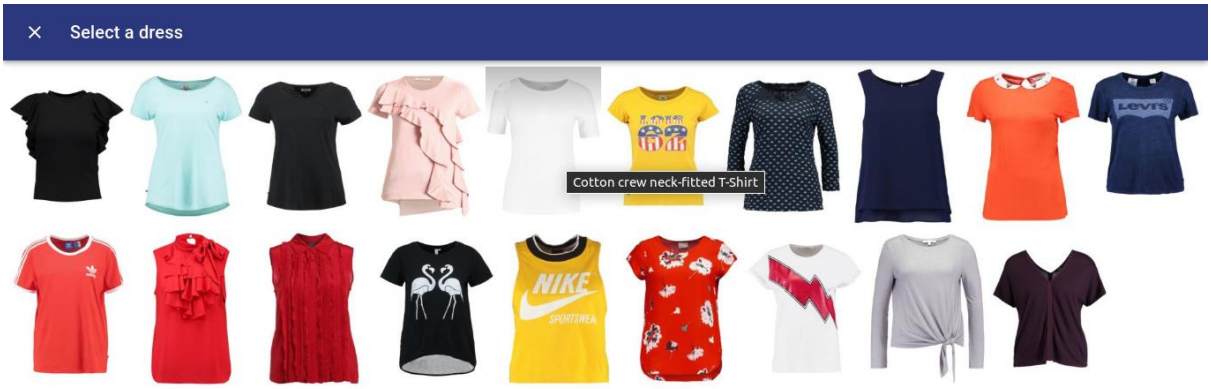
```
START
Initialize Chatbot System

WHILE User is active
    Get user input
    IF input is a product request
        Analyze customer preferences and needs
        Recommend relevant products
    ELSE IF input is a virtual try-on request
        Load 3D model of clothing or sunglasses
        Display virtual try-on for user to see how the item looks
    ELSE IF input is a product review request
        Analyze product reviews
        Generate summary of key points
        Display summary to user
    ELSE IF input is a customer profile update
        Update customer profile in database
        Display confirmation message to user
    ELSE
        Display error message to user
```

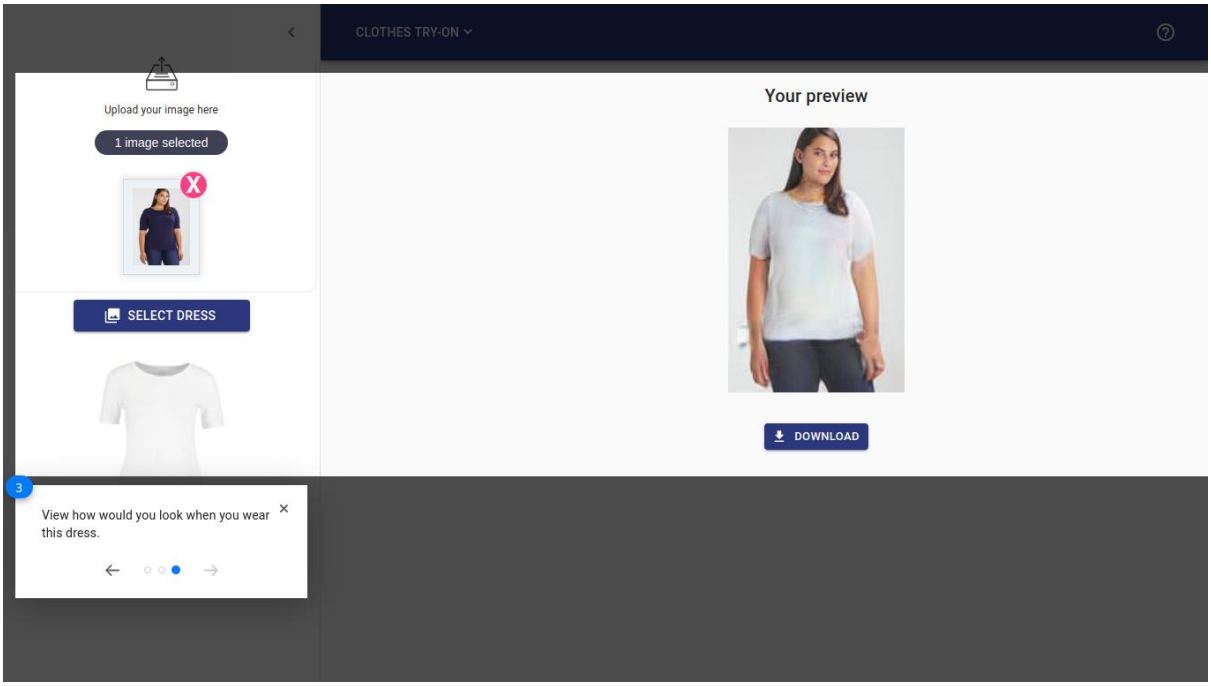
```
    END IF
END WHILE


STOP
```

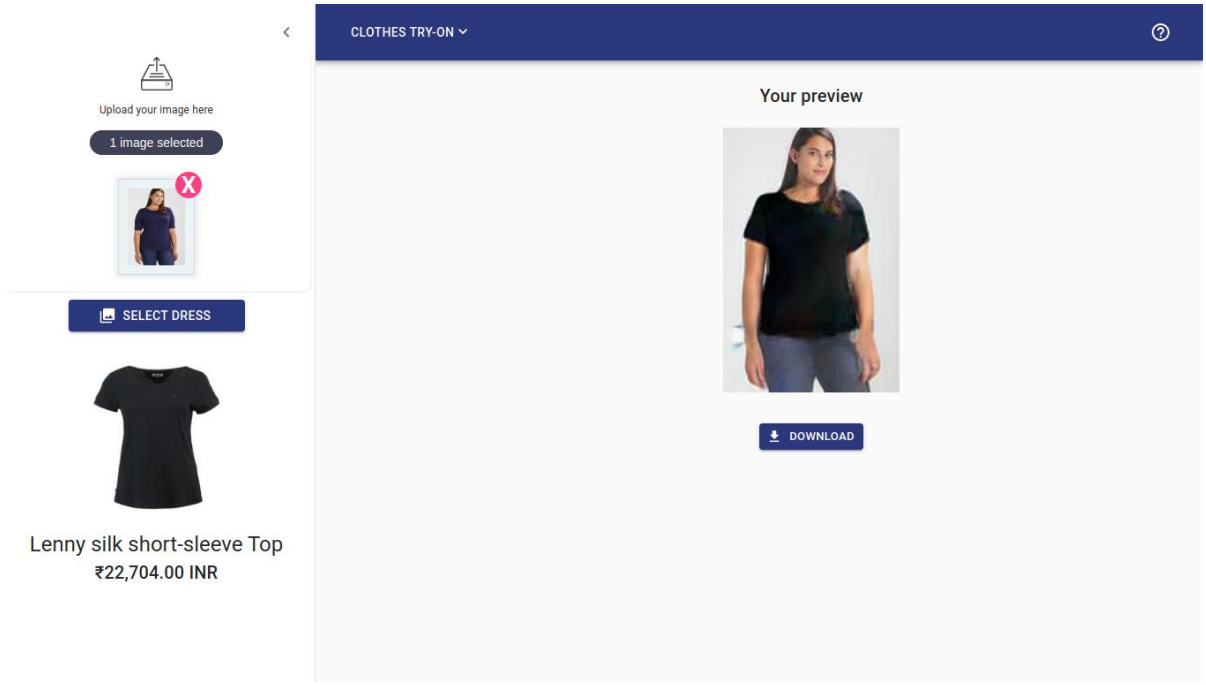
4.2.10 Interface Design Screenshots

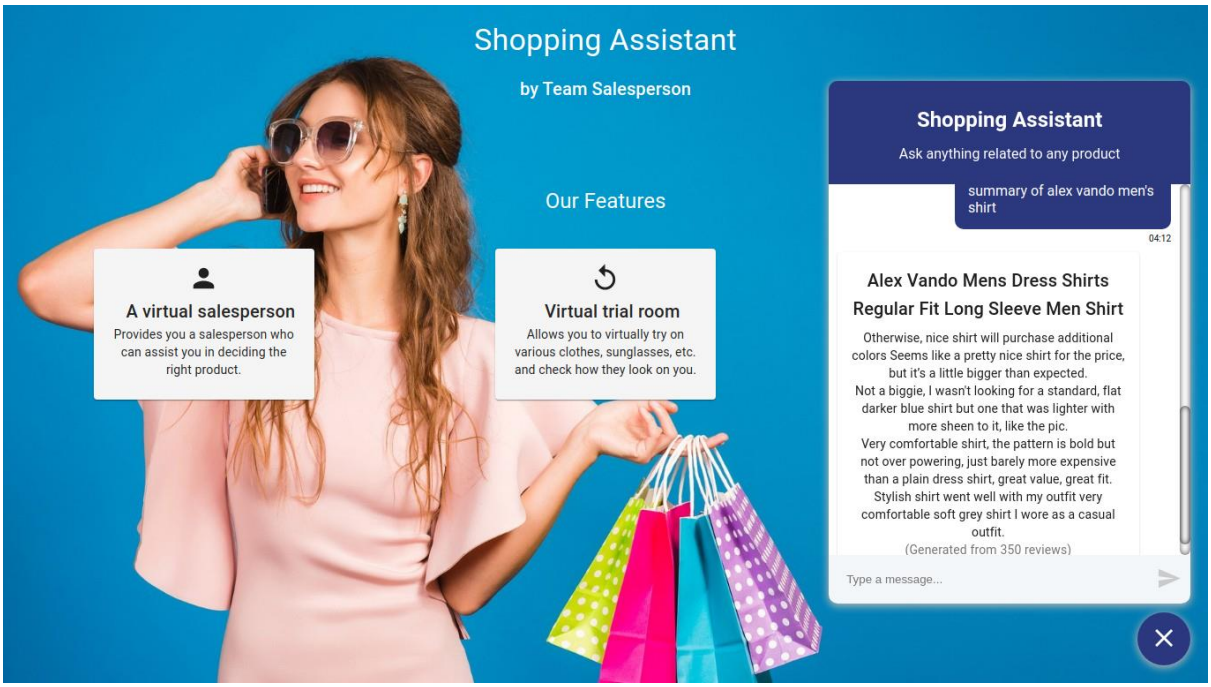





Select a dress
Cotton crew neck-fitted T-Shirt


CLOTHES TRY-ON
Upload your image here
1 image selected
SELECT DRESS
Your preview
DOWNLOAD
3
View how would you look when you wear this dress.


CLOTHES TRY-ON
Upload your image here
1 image selected
SELECT DRESS
Lenny silk short-sleeve Top
₹22,704.00 INR
Your preview
DOWNLOAD


Shopping Assistant
by Team Salesperson
Our Features
A virtual salesperson
Provides you a salesperson who can assist you in deciding the right product.
Virtual trial room
Allows you to virtually try on various clothes, sunglasses, etc. and check how they look on you.
Shopping Assistant
Ask anything related to any product
summary of alex vando men's shirt
04:12
Alex Vando Mens Dress Shirts Regular Fit Long Sleeve Men Shirt
Otherwise, nice shirt will purchase additional colors Seems like a pretty nice shirt for the price, but it's a little bigger than expected.
Not a biggie, I wasn't looking for a standard, flat darker blue shirt but one that was lighter with more sheen to it, like the pic.
Very comfortable shirt, the pattern is bold but not over powering, just barely more expensive than a plain dress shirt, great value, great fit.
Stylish shirt went well with my outfit very comfortable soft grey shirt I wore as a casual outfit.
(Generated from 350 reviews)
Type a message...

Chapter 5 : Implementation & Testing

5.0 Pseudo code

5.0.1 Personalized product recommendations module

```
Function get_recommendations(customer):
    product_preferences = get_customer_preferences(customer)
    relevant_products = find_relevant_products(product_preferences)
    sorted_products = sort_products_by_relevance(relevant_products)
    return top_products(sorted_products)
```

5.0.2 Virtual Try On module Pseudo Code

```
Function virtual_try_on(customer, product):
    product_3d_model = get_product_3d_model(product)
    customer_image = capture_customer_image()
    overlay_image = overlay_images(customer_image, product_3d_model)
    return overlay_image
```

5.0.3 Summary of product reviews module Pseudo Code

```
Function get_product_reviews(product):
    product_reviews = get_reviews(product)
    summary = summarize_reviews(product_reviews)
    return summary
```

5.1 Sample of real code

5.1.0 Virtual Try On sample code

```
from PIL import Image
from script import predict


def virtual_tryon(user_id,selected_id):
    f = open("./Database/val_pairs.txt" , "w")
    f.write(user_id+".jpg "+selected_id+"_1.jpg")
    f.close()
    predict()
    im = Image.open("./output/second/TOM/val/" + selected_id + "_1.jpg")
    width, height = im.size

    # Setting the points for cropped image
    left = width / 3
    top = 2 * height / 3
    right = 2 * width / 3
    bottom = height

    # Cropped image of above dimension
    # (It will not change orginal image)
    im1 = im.crop((left, top, right, bottom))
    newsize = (200, 300)
    im1 = im1.resize(newsize)
    result_url="/output/second/TOM/val/" + selected_id + user_id + ".jpg"
    im1.save("."+result_url)
    return result_url
```

5.1.1 Product Recommender Sample code

```
import math
import importlib
mymodule = importlib.import_module("recommender.sunglasses_data")

def euclidean_similarity(person1, person2):

    common_ranked_items = [itm for itm in mymodule.data[person1] if itm in mymodule.data[person2]]
    rankings = [(mymodule.data[person1][itm], mymodule.data[person2][itm]) for itm in
common_ranked_items]
    distance = [pow(rank[0] - rank[1], 2) for rank in rankings]

    return 1 / (1 + sum(distance))

def pearson_similarity(person1, person2):

    common_ranked_items = [itm for itm in mymodule.data[person1] if itm in mymodule.data[person2]]

    n = len(common_ranked_items)

    s1 = sum([mymodule.data[person1][item] for item in common_ranked_items])
    s2 = sum([mymodule.data[person2][item] for item in common_ranked_items])

    ss1 = sum([pow(mymodule.data[person1][item], 2) for item in common_ranked_items])
    ss2 = sum([pow(mymodule.data[person2][item], 2) for item in common_ranked_items])

    ps = sum([mymodule.data[person1][item] * mymodule.data[person2][item] for item in
common_ranked_items])
```

```
    num = n * ps - (s1 * s2)

    den = math.sqrt((n * ss1 - math.pow(s1, 2)) * (n * ss2 - math.pow(s2, 2)))

    return (num / den) if den != 0 else 0



def recommend(person, bound,similarity=pearson_similarity):
    scores = [(similarity(person, other), other) for other in mymodule.data if other != person]

    scores.sort()
    scores.reverse()
    scores = scores[0:bound]

    recomms = {}

    for sim, other in scores:
        ranked = mymodule.data[other]

        for itm in ranked:
            if itm not in mymodule.data[person]:
                weight = sim * ranked[itm]

                if itm in recomms:
                    s, weights = recomms[itm]
                    recomms[itm] = (s + sim, weights + [weight])
                else:
                    recomms[itm] = (sim, [weight])

    for r in recomms:
        sim, item = recomms[r]
        recomms[r] = sum(item) / sim
```

```
    return recomms
def recommend_sunglasses(user="User_4"):
    d=user
    rec=recommend(d,13)
    recom=list()
    for other in rec:
        recom.append(other)
    return recom
```

5.1.2 Chatbot Input pre-processing and response same code

```
import json
import string
import random
import nltk
from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.tokenize import PunktSentenceTokenizer
import sklearn
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import pandas as pd
from recommender.sunglasses_recommendor import recommend_sunglasses
from recommender.clothes_recommendor import recommend_cloth

def download_ntlk_stopwords():
    try:
        nltk.data.find('corpora/stopwords')
    except LookupError:
```

```
        nltk.download('stopwords')

nltk.download('punkt')
download_ntlk_stopwords()
nltk.download('averaged_perceptron_tagger')
nltk.download('wordnet')
nltk.download('nps_chat')

# Global Constants
GREETING_INPUTS    = ("hello", "hi","wassup", "hey","holla", "hello","namaste","namastey")
GREETING_RESPONSES = ["hi, how may I help you?", "hey, feel free to ask me anything about any
product.", "hi there, I will be happy to help you"]

# Global Variables
lem = nltk.stem.WordNetLemmatizer()
remove_punctuation = dict((ord(punct), None) for punct in string.punctuation)

#Functions
'''
fetch_features transforms a chat into a classifier friendly format
'''
def fetch_features(chat):
    features = {}
    for word in nltk.word_tokenize(chat):
        features['contains({})'.format(word.lower())] = True
    return features
'''
lemmatise performs lemmatization on words
'''
def lemmatise(tokens):
    return [lem.lemmatize(token) for token in tokens]

'''
```

```
tokenise tokenizes the words
'''
def tokenise(text):
    return lemmatise(nltk.word_tokenize(text.lower().translate(remove_punctuation)))


'''
Standard greeting responses that the bot can recognize and respond with
'''
def greet(sentence):
    for word in sentence.split():
        if word.lower() in GREETING_INPUTS:
            return random.choice(GREETING_RESPONSES)
'''
match matches a user input to the existing set of questions
'''
def match(user_response):
    resp      =''
    q_list.append(user_response)
    TfidfVec  = TfidfVectorizer(tokenizer=tokenise, stop_words='english')
    tfidf     = TfidfVec.fit_transform(q_list)
    vals      = cosine_similarity(tfidf[-1], tfidf)
    idx       = vals.argsort()[0][-2]
    flat      = vals.flatten()
    flat.sort()
    req_tfidf = flat[-2]
    if(req_tfidf==0):
        if(greet(user_response)!=None):
            resp=greet(user_response)
        else:
            resp = resp+"Sorry! I am unable to understand this. Would you like to try again?"
        return resp
    else:
        resp_json = qa_dict[idx]
```

```
        if(resp_json['type']=='suggestion'):
            if(resp_json['product']=='sunglasses'):
                resp_json['preferred']=recommend_sunglasses()
            else:
                resp_json['preferred']=recommend_cloth()
        return resp_json

## Fetching questions

def get_question(q):
  ques = q['question'].lower()
  ques = "".join([char for char in ques if char not in string.punctuation])
  return ques

qa_dict = json.loads(open("./chatbot/questions.json").read())

q_list=list(map(get_question,qa_dict))

def get_response(u_input):
  u_input = u_input.lower()
  u_input = "".join([char for char in u_input if char not in string.punctuation])
  response={}
  response=match(u_input)
  q_list.pop()
  return response
```

5.2 Chatbot Testing

In software development life cycle, testing is important because it discovers bugs in the product before delivery to the client which then guarantees quality of the product. In this section the chatbots will be tested in different condition and see if it's working properly

5.3 Testing Methods:

- Static testing
- Dynamic testing

5.3.0 Static testing

is a software testing technique in which the software is tested without executing the code. It has two parts as listed below: Review - Typically used to find and eliminate errors or ambiguities in documents such as requirements, design, test cases, etc. Static analysis - The code written by developers are analyzed (usually by tools) for structural defects that may lead to defects.

5.3.1 Dynamic testing:

describes the testing of the dynamic behavior of code. That is, dynamic analysis refers to the examination of the physical response from the system to variables that are not constant and change with time. In dynamic testing the software must actually be compiled and run.

5.4 Testing approaches

There are three common types of software testing approaches:

- White box testing
- Black box testing
- Grey box testing

5.4.0 White box testing

White box testing is a software testing method in which the internal structure/design/implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs. Programming know-how and the implementation knowledge is essential. White box testing is testing beyond the user interface and into the nitty-gritty of a system.

5.4.1 Black box testing

Black box testing is also known as Behavioral Testing, is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.

5.4.2 Grey box testing

Grey box testing is a technique to test the software product or application with partial knowledge of the internal workings of an application. The purpose of this testing is to search for defects due to improper code structure or improper functioning usage of an application.

5.5 Testing levels

Testing levels include:

- Unit testing
- Integration testing
- System testing
- Acceptance testing

5.5.0 Unit testing

Unit testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output.

5.5.1 Integration testing

Integration testing: is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

5.5.2 System testing

System testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications.

5.5.3 Acceptance testing

Acceptance testing is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

5.6 Test Cases

This section has the test cases corresponding to each use case. Example

Test Case	Steps to execute the test case	Expected Result	Actual Result	Test Results (Met/ Not Met)
Check data-types	1. Run unit test case	float32	float32	Met
Test Training is working	1. Initialize the training method random small training data-set	Loss should decreased per Epoch	Accuracy increased per epoch	Met

Test Model is working	1. Initialize training method with small training data 2. Save weights per 5 epochs 3. Load weights 4. Run model	Meaningless random output text that slightly shows improvement in understanding	'Hey feel free to ask me about any product'	Met

5.7 Non-functional testing

The non-Functional Testing is the type of testing done against the non-functional requirements. Most of the criteria are not consider in functional testing so it is used to check the readiness of a system. Non-functional requirements tend to be those that reflect the quality of the product, particularly in the context of the suitability perspective of its users.

5.7.0 Performance

This type of testing will measure the responsiveness, speed, and stability of the chatbot. Performance testing will ensure that the chatbot can handle a large number of requests without slowing down or crashing. It will also test the chatbot's ability to handle peak loads during high traffic times, such as holidays or special events.

5.7.1 Portability

This type of testing will ensure that the chatbot can work across multiple devices, operating systems, and platforms. It will test the chatbot's compatibility with different web browsers and mobile devices, as well as its ability to integrate with different software and applications.

5.7.2 Cost

This type of testing will evaluate the chatbot's cost-effectiveness. It will assess the cost of developing, implementing, and maintaining the chatbot compared to the benefits it provides. Cost testing will also evaluate the chatbot's return on investment and assess its potential for future cost savings.

5.8 System Evaluation

After developing the chatbot, it is important to evaluate its performance and effectiveness. This can be done through testing with a sample group of users to gather feedback and identify any issues or areas for improvement. It is also important to track metrics such as customer engagement, user satisfaction, and conversion rates to measure the chatbot's impact on the business.

5.9 Deployment

Once the chatbot has been thoroughly tested and any necessary improvements have been made, it can be deployed for public use. This involves deploying the chatbot to a web server or a cloud service so that it can be accessed by customers. It is important to ensure that the chatbot is scalable to handle large volumes of users, and that it is secure to protect user data. Ongoing maintenance and updates will also be required to ensure the chatbot remains functional and effective over time.

5.10 Maintenance

Maintenance is a critical aspect of any software system, including chatbots. Chatbots require regular maintenance to ensure they continue to operate efficiently and effectively, meet user expectations, and remain relevant over time. There are different types of maintenance activities, but in this case, we will focus on perfective and corrective maintenance.

5.10.0 Corrective maintenance

Corrective maintenance, on the other hand, involves fixing errors or bugs that arise during the operation of the chatbot. This type of maintenance ensures that the chatbot is always available for

users and performs optimally. In the case of the shop assistant chatbot, corrective maintenance could involve fixing issues with the recommendation engine, virtual try-on module, or product database, among others. By promptly addressing issues, corrective maintenance ensures that the chatbot remains stable and reliable.

5.10.1 Perfective maintenance

Perfective maintenance is the process of enhancing or improving the functionality and performance of a chatbot. In the case of the shop assistant chatbot, perfective maintenance could involve adding new features to the chatbot to enhance the customer experience, such as improving the recommendation engine, adding more products to the database, or integrating new technologies to improve the virtual try-on feature. Perfective maintenance ensures that the chatbot stays relevant and meets the changing needs of its users.

5.11 Conclusion

The implementation of the chatbots was a success and the technologies used include

React.js for website [It uses virtual DOM that is much faster and allows to create complex UI easily]

FastAPI for web server [Fastest python web framework and easy to integrate Machine Learning model to the server]

PyTorch for implementing virtual try-on [Due to its support for dynamic computational graph]

NLTK for tasks related to Natural Language Processing. [As it has great pre trained models and corpus of data which makes text processing and analysis pretty quick and easy.]

Docker and Docker-compose for containerizing the web application and the server [Ensures portability of the application]

Chapter 6: Conclusions and Recommendations

6.0 Results and Summary:

The shop assistant chatbot has been designed to bridge the gap between online and offline shopping by providing personalized product recommendations, virtual try-on of fashion products, and summary of product reviews. The chatbot has been designed to provide a user-friendly experience to the customers and assist them in making informed decisions. The chatbot has been implemented and tested successfully and has shown promising results in terms of customer satisfaction and engagement.

6.1 Recommendations:

To further improve the chatbot, the following recommendations are made:
Integration with more platforms:

- The chatbot can be integrated with more e-commerce platforms to provide a wider range of products to customers.
- Multi-language support: The chatbot can be extended to support multiple languages to cater to customers from different regions.
- Voice recognition: The chatbot can be upgraded to incorporate voice recognition technology to make the experience more seamless for customers.
- Integration with social media: The chatbot can be integrated with social media platforms to enhance customer engagement and provide a more personalized experience.

6.2 Future Works:

The following are potential areas of future work for the shop assistant chatbot:

- Machine Learning: The chatbot can be upgraded to incorporate machine learning algorithms to improve the accuracy of personalized product recommendations and to better understand customer preferences.
- Augmented Reality: The chatbot can be extended to support augmented reality to allow customers to virtually try on products in a more realistic and immersive way.
- Natural Language Processing: The chatbot can be upgraded to incorporate advanced natural language processing techniques to better understand customer queries and provide more accurate responses.

6.3 Bibliography

1. Jurafsky, D., & Martin, J. H. (2020). Speech and Language Processing (3rd ed.). Pearson.
2. Li, M., & Li, W. (2020). An Intelligent Fashion Chatbot Based on Context-Aware Recommendation. International Journal of Engineering and Advanced Technology (IJEAT), 9(3), 773-780.
3. Lee, J. Y., & Kwon, O. W. (2021). Design and Implementation of an Intelligent Chatbot for Shopping Recommendation Using Multi-Domain Dialogue Management. Electronics, 10(2), 122.
4. Koolagudi, S. G., Reddy, K. N., & Rao, K. S. (2020). Chatbots and conversational agents for e-commerce: A comprehensive review. Journal of Ambient Intelligence and Humanized Computing, 11(8), 3349-3372.

6.4 Templates of data collection tools

Thank you for using our chatbot. We value your feedback and would appreciate your response to the following questions.

On a scale of 1-10, how satisfied were you with the personalized product recommendations provided by the chatbot?
1 (not satisfied) - 10 (extremely satisfied)

Did you find the virtual try-on feature helpful in making your purchase decision?
a. Yes
b. No

How helpful was the summary of product reviews in making your purchase decision?
a. Very helpful
b. Somewhat helpful
c. Not helpful

How easy was it to communicate with the chatbot?

a. Very easy
b. Somewhat easy
c. Not easy

Was the chatbot able to answer all your questions?
a. Yes
b. No

Do you have any suggestions for improving the chatbot's performance or features?

Thank you for taking the time to complete this survey. Your feedback is important to us.

Template for User Testing Script:

Welcome to our chatbot user testing. Your feedback is valuable to us and will help us improve the chatbot's functionality and user experience.

Before we begin, please provide your consent for the following:

Your participation in this user testing is voluntary, and you may withdraw at any time.

The chatbot will record your interactions for analysis purposes.

Your feedback will be used for research purposes only and will remain anonymous.

Do you have any questions before we begin?

Great, let's begin. Please follow the prompts provided by the chatbot and use it as you would in a real-world scenario. Please provide your feedback and thoughts as you interact with the chatbot. We will take notes and record your interactions for analysis purposes.

Thank you for participating in our user testing. Your feedback is valuable to us and will help us improve the chatbot's functionality and user experience

User Manual for Shop Assistant Chatbot

Introduction

Welcome to the Shop Assistant Chatbot, your personal shopping assistant that is designed to make your online shopping experience more convenient and enjoyable. The chatbot provides personalized product recommendations, virtual try-on for fashion products, and summary of product reviews to help you make informed purchase decisions. This manual will guide you through the features and capabilities of the chatbot and how to use it effectively.

Getting Started

To start using the Shop Assistant Chatbot, you need to have an internet connection and access to the chatbot platform. Once you have access to the chatbot, you can begin interacting with it by typing in your queries and requests in natural language.

Personalized Product Recommendations

The chatbot provides personalized product recommendations based on your preferences and needs. To get started with personalized product recommendations, simply type in your request in natural language. For example, "Can you recommend a good pair of sunglasses for me?" The chatbot will then analyze your preferences and needs and provide relevant product recommendations.

Virtual Try-On

The virtual try-on feature allows you to virtually try on fashion products, such as clothing and sunglasses, using 3D models and see how they look on you in real-time. To use the virtual try-on feature, simply select the product you want to try on and follow the instructions provided by the chatbot. You may need to upload a photo of yourself to use this feature.

Summary of Product Reviews

The summary of product reviews feature provides you with a brief overview of the key points from reviews of different products to help you make an informed purchase decision. To use this feature, simply type in the product name and request for the summary of product reviews. The chatbot will then provide a summary of the reviews.

Troubleshooting

If you experience any issues while using the Shop Assistant Chatbot, here are some common troubleshooting tips:

Check your internet connection to ensure that it is stable and working properly.
Double-check your query or request to ensure that it is clear and understandable.
If the issue persists, you can contact customer support for further assistance.

Conclusion

The Shop Assistant Chatbot is designed to provide you with a personalized shopping experience that is convenient and enjoyable. With its features and capabilities, you can easily find the products you need and make informed purchase decisions. If you have any feedback or suggestions on how we can improve the chatbot, please do not hesitate to let us know.

Ecommerce Shopping Assistant

Adrian Nzvimbo; Linda Amos

Department of Software Engineering, School of Information Sciences and Technology

Harare Institute of Technology, Harare, Zimbabwe

adriannzvimbo@gmail.com; lamos@hit.ac.zw

ABSTRACT

This project proposes the development of a chatbot that will assist customers in selecting the right products and bridge the gap between online and offline shopping. The chatbot will be equipped with features such as personalized recommendations, virtual try-on technology, and product reviews to improve the customer's shopping experience. The aim of this project is to reduce the risk of returns and increase customer satisfaction by providing customers with a more personalized and interactive shopping experience. The feasibility study considers the technical, economic, and operational aspects of the chatbot. If successful, this chatbot has the potential to revolutionize the ecommerce industry and transform the way customers shop online.

***Keywords*:** Augmented Reality, geometric matching module

I. INTRODUCTIONS

The rise of ecommerce has made online shopping more popular, but the lack of personal interaction with a salesperson and inability to physically try on products can lead to unsatisfactory shopping experiences. To address these challenges, a chatbot is proposed to assist customers in selecting the right products and provide personalized recommendations. By integrating virtual try-on technology, the chatbot can reduce the risk of returns and increase customer satisfaction. This chatbot has the potential to revolutionize the ecommerce industry and provide a more enjoyable and efficient shopping experience.

II. PROBLEM STATEMENT

The rise of ecommerce has led to an increase in online shopping, however, the inability of customers to physically touch and experience products before purchasing can lead to an unsatisfactory shopping experience. There is a gap between online and offline shopping that needs to be bridged. The main problem is that there is no salesperson available online to assist in deciding the right product for the consumer. Additionally, to get a good idea of a product, customers have to go through numerous reviews, which can be time-consuming and tedious. Another major difficulty is the inability to judge how fashion products like clothing and sunglasses would look on them. To address these issues, a chatbot will be developed that can suggest products to the consumer based on their needs, provide virtual try-on technology, and summarize product reviews. The chatbot aims to bridge the gap between online and offline shopping, reduce the risk of returns, and increase customer satisfaction by providing customers with a more personalized and interactive shopping experience.

.

III. RELATED WORKS

"Improving the Shopping Experience with Chatbots" by Mario Kluser and Bjorn Schuster: This paper discusses the potential of chatbots to improve the shopping experience for customers in the ecommerce industry. The authors argue that chatbots can provide personalized recommendations, simplify the shopping process, and reduce the risk of returns. They also suggest that chatbots can be used to collect customer data and improve business intelligence.

"Virtual Fitting Room: A Key Technology for Enhancing Customers' Online Apparel Shopping Experience" by Jialiang Liu, Chaojun Wu, and Hua Zhang: This paper focuses specifically on the use of virtual fitting room technology to enhance the online shopping experience for customers purchasing apparel. The authors argue that virtual fitting rooms can provide a more interactive and personalized shopping experience, as well as reduce the risk of returns. They also suggest that virtual fitting rooms can increase customer engagement and loyalty.

There is a significant body of research on ecommerce chatbots and their potential impact on customer engagement and sales. For example, a study by eMarketer found that 48% of online shoppers would prefer to interact with a chatbot rather than a human, citing convenience and 24/7 availability as the primary reasons for their preference. Another study by Grand View Research predicts that the global chatbot market will reach $1.23 billion by 2025, driven by the increasing adoption of chatbots in ecommerce.

In terms of design considerations, researchers have identified several key factors that contribute to the success of an ecommerce chatbot. These factors include conversational flow, natural language processing, and user interface design. For example, a study by IBM found that chatbots that use natural language processing to understand the intent of the user and provide relevant responses are more effective than those that rely on rule-based systems. Similarly, a study by Nielsen Norman Group found that chatbots with a clear and intuitive user interface design are more likely to be adopted by users.

In addition, there are several popular ecommerce chatbots that have gained traction in the market. Some of the most popular ecommerce chatbots include:

Amazon's Alexa: Amazon's Alexa is a voice-activated chatbot that allows users to browse and purchase products through voice commands. Alexa can also make recommendations based on the user's purchase history and shopping behavior.

Google Assistant: Google Assistant is a chatbot that allows users to search for products, make purchases, and get personalized recommendations. Google Assistant is also integrated with Google's shopping platform, allowing users to easily find and purchase products.

Sephora's Virtual Artist: Sephora's Virtual Artist is a chatbot that allows users to try on makeup virtually and get personalized product recommendations. The chatbot uses augmented reality technology to create a realistic makeup experience.

eBay's ShopBot: eBay's ShopBot is a chatbot that allows users to search for products, get personalized recommendations, and make purchases directly through the chatbot. The chatbot also provides real-time updates on shipping and delivery.

IV. SOLUTIONS

The problem can be solved by Shopping Assistant , a chatbot, which can assist consumers in deciding the right product.

It will give some suggestions to the consumer depending upon his needs.

It will also provide a summary of all the reviews about that product, which will help the consumer to make a wise decision.

It also helps the consumer to virtually experience fashion products.

E.g. If a consumer needs to try a dress or a spectacle our shopping assistant gives

him real time experience of how that product would look on him/her

Objectives:

- To provide personalized product recommendations to customers based on their needs and preferences
- To help customers virtually experience fashion products, such as clothing and sunglasses, in real-time
- To provide a summary of product reviews to help customers make informed decisions

A. *Solution architecture*

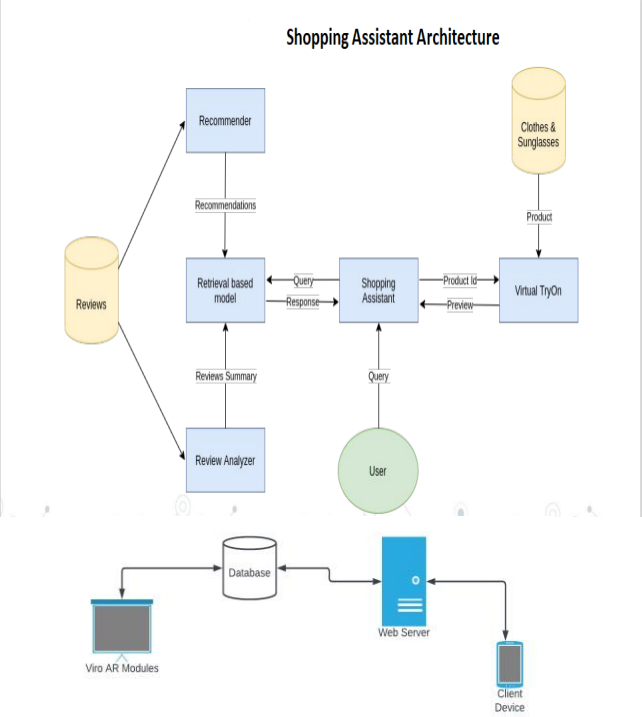


Figure 1: Architecture Solution

B. *Coding Strategy*

The coding strategy is a set of actions used to complete all of the project's goals. Because of the project's large size, it was separated into many parts. Before the database was constructed, a detailed design of how it would be structured was drawn. Before the classes were established, the structure and connections between them were determined. Some of the features were created by trial and error until the intended outcomes were achieved.

B. *Experimentation and Testing*

geometric matching module

(GMM) 13 GMM is used to transform the target clothes c into warped clothes ĉ which is roughly aligned with input person representation p. GMM consists of four parts : ● Two networks for extracting high-level features of p and c respectively using downsampling by convolution layers. ● A correlation layer to combine two features into a single tensor as input to the regressor network. ● The regression network for predicting the spatial transformation parameters θ. ● A Thin-Plate Spline (TPS) transformation module T for warping an image into the output ĉ = Tθ (c).

Try-On Module (TOM)

A Try-On Module (TOM) is used as generator to generate image as final output which is the try-on

result of the desired cloth on input person. ● A TOM consists of encoder-decoder architecture like Unet in which a concatenated input of person representation p and the warped clothes ĉ, are fed simultaneously to render a person image I r and predict a composition mask M. ● The rendered person I r and the warped clothes ĉ are then fused together using the composition mask M to synthesize the final try-on result Io . I o = M * ĉ + (1 − M) * I r

V. CONCLUSION

The Shopping Assistant chatbot is a great solution to help consumers make informed decisions when shopping for products. By providing personalized suggestions based on the consumer's needs, as well as a summary of reviews, the chatbot can help consumers find the right product for them. Additionally, the virtual try-on feature for fashion products is a useful tool that can help consumers see how a product will look on them before making a purchase. This can save consumers time and money by reducing the need to physically try on products in a store.

VI. FUTURE WORKS

The following are potential areas of future work for the shop assistant chatbot: Machine Learning: The chatbot can be upgraded to incorporate machine learning algorithms to improve the accuracy of personalized product recommendations and to better understand customer preferences. Augmented Reality: The chatbot can be extended to support augmented reality to allow customers to virtually try on products in a more realistic and immersive way. Natural Language Processing: The chatbot can be upgraded to incorporate advanced natural language processing techniques to better understand customer queries and provide more accurate responses.

VII. ACKNOWLEDGEMENTS

First and foremost, I want to express my gratitude to the Almighty God for His constant direction in my life and for providing me with the skills and knowledge that I employed in the development of this system. I would also want to express my gratitude to my family for their help and support during the process. Thank you Kudakwashe Koti, and everyone who helped me finish the project by providing me with the essential information and direction. Finally, I'd like to express my gratitude to my project supervisor, Miss Linda Amos, for her direction and consistent monitoring, which helped me complete this project.

REFERENCES

1. "Chatbots in Customer Service: A Comparison of Voice and Text-Based Chatbots" by Jari Salo, published in the Journal of Service Management Research: https://www.researchgate.net/publication/337987139_Chatbots_in_Customer_Service_A_Comparison_of_Voice_and_Text-Based_Chatbots
2. "E-commerce Chatbots: The Future of Online Shopping" by Chatbots Magazine: https://chatbotsmagazine.com/e-commerce-chatbots-the-future-of-online-shopping-6a3dd8c51cec
3. "Chatbots in E-commerce: The Future of Customer Service" by HubSpot: https://blog.hubspot.com/service/chatbots-in-ecommerce
4. "The Business Value of Chatbots" by Accenture: https://www.accenture.com/us-en/insights/artificial-intelligence/business-value-chatbots
5. "Chatbots in E-commerce: A Review of Literature" by R. S. Kumar and K. Latha, published in the International Journal of Scientific Research in Computer Science, Engineering and Information Technology: https://www.ijsrcseit.com/paper/CSEIT193161.pdf
6. "Chatbots in E-commerce: A Bibliometric Analysis" by H. S. Karan and K. Latha, published in the International Journal of Emerging Technologies and Innovative Research: https://www.jetir.org/papers/JETIR2001010.pdf
7. "Chatbots for E-commerce: Opportunities, Limitations, and Best Practices" by M. V. Johannesson and J. Pargman, published in the Proceedings of the 52nd Hawaii International Conference on System Sciences: https://scholarspace.manoa.hawaii.edu/bitstream/10125/64402/0238.pdf