15th International Conference on Business and Information – 2024
ISSN 2961-5283 (Online)

Design and Development of a Reusable Chatbot for Fashion Recommendations: Sinhala Language as a Case Study

Sugathadasa P.D.C.M [1], Yatigammana M.R.K.N [2] and Wijayarathna G. [3]

People communicate about their lifestyles, ideas, and social standing through fashion. Due to the ever-evolving desires and personal preferences and the availability of a wide range of options, selecting the ideal fashion design becomes difficult for customers. As a result, recommendation systems and chatbots have come to the market. In Sri Lanka, these chatbots are only available in English; thus, they are restricted in their ability to serve most Sinhala-speaking community in Sri Lanka. To solve the said problem, this study builds a Sinhala fashion recommendation chatbot. This research focuses on the development of a chatbot developed using Python Flask. It uses OpenAI's GPT-3.5 Turbo to understand and generate human-like responses and Bing Translator to communicate in Sinhala language. The fashion problem solution was developed using the Soft Design Science Research Methodology (SDSRM). The chatbot was designed and developed using component-based software engineering (CBSE), which allows one to easily develop a complex application like a chatbot and export it to other domains. This makes it easy to deploy the chatbot for different purposes in business, education, or any other field with minimal modification. A hosted chatbot prototype was evaluated using a Google Form to obtain feedback. Users reported satisfaction with the personalized style suggestions, indicating the system's potential effectiveness. The unique achievement of this study was building a reusable chatbot module by combining several components; therefore, with minimum modifications, the chatbot can be customized to other domains as well.

Keywords: *Chatbot, Fashion Design, Re-usable, Sinhala Language*

---

[1] Department of Commerce and Financial Management, University of Kelaniya. (chamodsugathadasa@gmail.com)

[2] Department of Commerce and Financial Management, University of Kelaniya. (kaushalya@kln.ac.lk)

[3] Faculty of Computing and Technology, University of Kelaniya. (gamini@kln.ac.lk)

Introduction

The fashion industry is a diverse global business involving producing, marketing, selling, and consuming clothing, accessories, and footwear. It is a dynamic and ever-evolving industry driven by creativity, innovation, and consumer demand. The sector includes designers, manufacturers, wholesalers, retailers, and consumers, each playing a pivotal role in shaping fashion trends and styles. Fashion is a means of self-expression and reflects cultural, social, and economic values. With the advent of e-commerce, marketing has made the fashion industry more accessible and inclusive, allowing individuals to express their style and draw inspiration from worldwide.

One of the significant technological advancements in fashion design is Computer-Aided Design (CAD) software, which has enabled fashion designers to create and manufacture their fashion ideas. With the help of this CAD software, designers can simplify the design process, improve design quality, and reduce production time (Sayem et al., 2010).

On the other hand, the way consumers access fashion information has changed. With the advancement of the internet, many digital media channels are now becoming popular among fashion consumers (Chu & Seock, 2020). Consumers can now access fashion blogs, social media, magazines, events, and personal stylists as sources of information on the latest fashion trends (Michaela & Orna, 2015). On the other hand, fashion marketers have changed their marketing strategies and evolved to include influencer marketing, a popular trend among consumers, especially the younger population (Zietek, 2016). With the help of influencer endorsements, brands can quickly and effectively reach their target audience and lead to higher sales and customer engagement.

Recommendation System

A recommendation system is a computer program that uses machine learning upon data analysis to identify what a user is searching for among an ever-increasing number of options. The system predicts, narrows down, and presents suggestions based on a user's past behaviours and preferences. The recommendation system can offer a range of options and recommendations based on their purchase history, new trends, in-depth information, real-world exhibitions, and personalized advice. However, these recommendation systems have issues such as cold start problems, scalability concerns, data sparsity, and over-specialization issues. (Sharma & Gera, 2013).

Recommendation Chtabots

Recommendation chatbots emerged as a promising solution for the problems in recommendation systems, as they can handle a wide range of inquiries and tasks. The AI-powered assistants are particularly adept at answering frequently asked questions, booking appointments, and making personalized user recommendations. Compared with other recommendation systems, chatbots could be deployed through several platforms, such as websites, social media, and messenger platforms. Further, chatbots' smartphone compatibility facilitates most citizens' use of chatbots in their daily activities (Lin et al., 2023).

Problem Statement

Fashion has become essential to our lives, enabling us to express our personality, culture, and identity. People choose their fashion based on various factors, such as personal style, trendiness, comfort, price, and occasion (Ray & Nayak, 2023).

However, the process of choosing fashion designs has several challenges and difficulties. These include finding fashion designs with the right size that fits a person's body type, choosing the latest fashion designs with constantly changing trends and styles, limited availability of fashion designs in certain regions, and difficulty in finding fashion designs for a unique personal style (Wang et al., 2015).

To address these issues, recommendation systems were developed. However, these systems faced their own limitations, such as the cold start problem, scalability concerns, data sparsity, and over-specialization issues (Sharma & Gera, 2013).

As a solution to these persistent problems, chatbots emerged as a promising technology. Chatbots offer several advantages over traditional recommendation systems. They can be deployed across multiple platforms, including websites, social media, and messaging apps, making them highly accessible. This accessibility is particularly important given the prevalence of mobile devices in modern society. According to Statista, there were 6.92 billion smartphone users worldwide in 2023 (*How Many People Have Smartphones Worldwide (May 2023)*, n.d.). In Sri Lanka specifically, the CEIC reports 30.764 million mobile cellular users (*Sri Lanka Number of Subscriber Mobile, 1960 – 2023 | CEIC Data*, n.d.).

The widespread use of mobile devices makes chatbots an ideal solution for reaching a broad audience, including rural citizens of Sri Lanka, with personalized fashion design recommendations. However, despite the potential of chatbots, there isn't a fashion design suggestion chatbot in Sri Lanka that can support the Sinhala-speaking community. Various chatbots recommend fashion designs that are built in the English language, which prevents most of the Sinhala-speaking community from using them.

Landim et al. (2022) also state that implanting such a bot is challenging because it focuses on how difficult human language is and how well the chatbot performs in a fashion environment.

Therefore, this study intends to improve user access to fashion advice in Sri Lanka and other countries where the Sinhala language is prevalent by building and developing a chatbot that can support the Sinhala language.

Thus, the objectives of this study are to:

1. Design and develop a reusable Sinhala Chatbot to provide Fashion Design recommendations for the Sinhala-speaking community.
2. Evaluate the prototype of the Fashion Chatbot with the potential users.

Literature review

Chatbots can emulate human conversations and entertain users. They can also be used in education, information retrieval, business, and e-commerce (Abu Shawar & Atwell, 2007). Chatbot can assist, solve, or automate tasks under the given information (Commands) (Lewandowski et al., 2021). Looking back to the history of Chatbots, the first Chatbot was developed in 1966. That Chatbot is called Eliza. Eliza aimed to act as a psychotherapist and return the user's words as a question (Weizenbaum, 1966). After Eliza Chatbot named ALICE was developed in 1995. ALICE is built on the Artificial Intelligence Markup Language (AIML) and uses a straightforward pattern-matching algorithm. Chatbots were developed inside messenger platforms in 2001. As a result, developers started to build virtual personal assistants like Apple Siri, Microsoft Cortana, Amazon Alexa, Google Assistant, and IBM Watson were developed as the next steps (Adamopoulou & Moussiades, 2020).

According to Gümüş & Çark (2021) there are four types of Chatbots that organizations use to handle their customer services. They are flow-based chatbots, AI chatbots, hybrid chatbots, and Human-powered chatbots.

Shoufan (2023) observed that students like ChatGPT's features and find it helpful for school and jobs. Its human-like interface and well-structured responses and explanations make it straightforward to use. However, many students believe ChatGPT's answers are not always accurate and require solid foundation knowledge since technology cannot replace human intelligence.

The first Sinhala chatbot was developed in 2006. This chatbot was constructed as an implementation of the Sinhala language parsing system. Simple inquiries can be answered using this system. It does not cover a specific domain, and the development purpose was to implement a Chatbot system that uses a Sinhala language and motivates future Sinhala Chatbot developments. The chatbot was created with Java and SWI-Prolog, and its knowledge base is made from the Windows Operating System (Hettige & Karunananda, 2006).

After that system, a domain-specific Sinhala chatbot was developed to answer user inquiries about the Degree Programs at the University of Ruhuna. So, users can effectively answer their inquiries and solve problems with a domain-specific chatbot since they can ask questions directly and reference specific topics, saving time on internet searches. The developer uses an open-source dialogue management tool called RASA to develop this chatbot. There are two main parts of the RASA framework: RASA Core and RASA NLU. Users are empowered by RASA Core's sophisticated interactions and chats. The chatbot deployed in the university website widget and the university’s Facebook page as a messenger bot (Kumanayake, 2021).

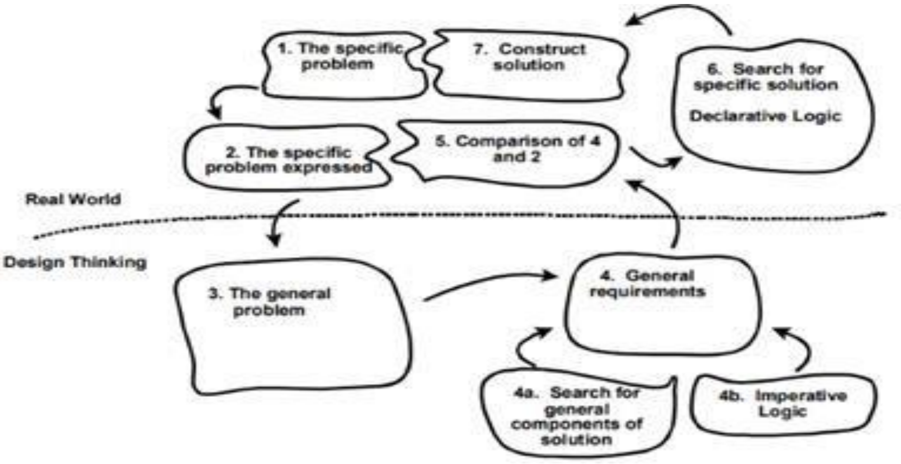
When looking at the recommendation for the Sinhala chatbot, there is one chatbot. It was built for a specific domain, and it is on “Sri Lankan Traditional Dancers.” Researchers make a system for traditional dancers to find creators of dance tracks, costume designers, prop suppliers, and choreographers. So, the article presents a website that offers data based on geography and rating, acting as a system for service providers' recommendations. The "cold start problem" can be resolved by this system (Chandrasena et al., 2021).

Methods

This research used the soft design science research methodology (SDSRM) (Venable et al., 2017).

Figure 1

Steps of SDSRM



Source: (Venable et al., 2017)

SDSRM methodology is inspired by Soft Systems Methodology (SSM). It has seven activities to solve a socio-technical problem.

1. The authors conducted a literature survey and internet search and found that no fashion design recommendation system can provide recommendations in Sinhala.
2. A survey conducted among Sinhala-speaking customers expressed the need for a fashion design recommendation mechanism in Sinhala. We also found that some fashion design recommendation systems support English-speaking communities. The fashion design recommendation generated in English must be translated into Sinhala.

3. For this project, we need a bilingual translation of Sinhala-English. A new system is also required to communicate with the existing recommendation systems and language translators.
4. The component-based Software engineering (CBSE) process is adopted to design and implement the general solution. We decided to reuse all available components, such as language translators and recommendation systems, and integrate them through a new system to solve our problem identified in Activity 1. This approach reduces the project duration drastically while maintaining the quality of the recommendations generated at a high level. We created a software model for the general requirement at this step.
5. The general design requirements are evaluated against the specific problem to assess compatibility and test the feasibility of the general solution. The problem is reframed regarding the general requirements and the imperative logic to articulate the design.

Table 1

Mapping SDSRM

Specific Problem from Step 2	General Requirements from Step 4
1. It is required to have a bilingual translation between Sinhala and English 2. It is required to interact with existing recommendation systems.	1. Used CBSE to identify components to satisfy the requirements elucidated in the Activity 2 2. Developed a software model using the components identified in the above activity.

6. Conducted a literature survey and Internet search to identify software components available in the market, facilitating the subsequent solution construction. Developed the system and conducted testing and evaluation.
7. Developed the system, implemented the identified components, and conducted thorough testing and evaluation to ensure functionality, usability, and performance.

Chatbot Development

Component-Based Software Engineering (CBSE): A Brief Overview

CBSE is a specialized branch of software engineering that focuses on constructing software systems using reusable, replaceable, and loosely coupled independent components (Crnkovic, 2001). Embracing the separation of concerns principle, CBSE divides the software development process into autonomous elements, each concentrating on a specific component or function of the system.

Components within CBSE are self-contained units in the software architecture, functioning independently and encapsulating distinct tasks. These components adhere to specific interfaces, enabling individual development, deployment, and maintenance. According to IEEE's "Guide to the Software Engineering Body of Knowledge," a software component is described as a "standalone, self-contained, small-sized, and easy-to-understand piece of code with well-defined boundaries" (Bourque, 2014).

This methodology reshapes the traditional waterfall-based development life cycle, which includes requirements, Design, Implementation, Test, Release, and Maintenance, into a streamlined CBSE life cycle: searching, choosing, adjusting, examining, implementing, and replacement. This transformation fosters a more adaptable and modular approach to software design, emphasizing reusability, maintainability, and scalability in constructing software systems.

Why CBSE in this research?

Component-Based Software Engineering (CBSE) offers several advantages that transcend the IT domain so that non-IT undergraduates can grasp the fundamental principles of efficient system development, fostering a more structured and collaborative approach to tackling complex projects, regardless of the specific field they are working in. Building a highly complex system like a chatbot within a year is a time-consuming process that involves years of expertise in areas such as data infrastructure, machine learning, natural language processing, speech recognition, and conversational interfaces.

The studies on software development methodologies have strongly recommended CBSE for several compelling reasons (Gulia & Palak, 2017). This concept suggests that developers can assemble systems by configuring and connecting reusable components from pre-existing packages or libraries instead of developing them from scratch.

Component-based software engineering (CBSE) offers several advantages, including,

1. Timesaving
2. Cost-effectiveness
3. Increased productivity
4. Quality improvement
5. Flexibility and scalability
6. Enhancing reliability

Component-Based Software Engineering Process

Component-based software engineering (CBSE) is a software development approach that supports developing software systems based on the composition of reusable, standardized, and independent components. (Pressman, 2010; Sommerville, 2011)

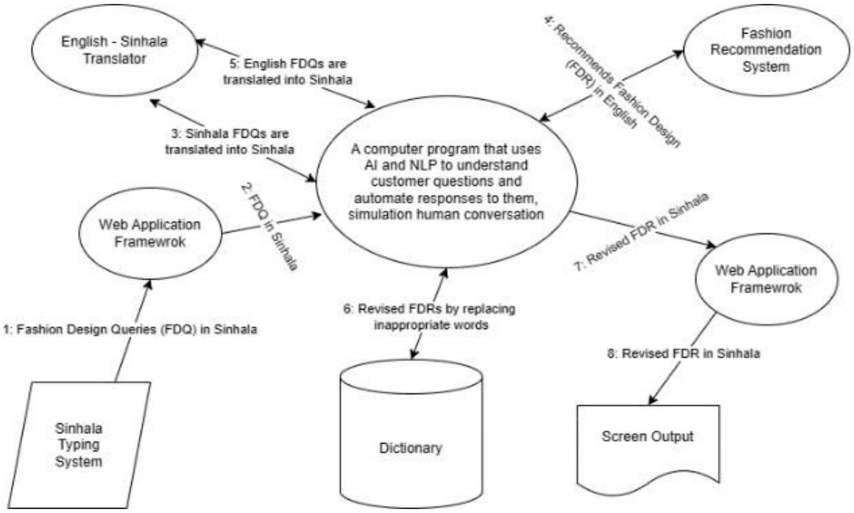
Usually, the following framework activities are included in the development with the reuse CBSE process.

1. Component Identification
2. Component Selection
3. Component Adaptation
4. Component Integration
5. Testing and Verification
6. Deployment and Maintenance

CBSE - Component Identification

Figure 2

Component Identification Diagram



As shown in Figure 2, all the components are identified according to the fashion recommendation system’s requirements. The diagram shows how data logically flows through the components. Therefore, the logical process is,

1. The user must use a Sinhala typing tool to type the response to the web UI in Sinhala.
2. Web UI passes user response to Chatbot.
3. The Chatbot passes the user's Sinhala response to the translator, who translates it into English and returns it to the chatbot.
4. The chatbot gives the English-translated user a response to the fashion recommendation system. The recommendation system gives the related recommendations to the chatbot.
5. The chatbot passes the English-style query to the translator, who translates it into Sinhala and passes it back to the chatbot.
6. Chatbot checks Sinhala translated fashion queries with Sinhala language CSV dictionary based on fashion domains and replaces wrongly translated words with correct words.
7. Chabot then delivers the revised Sinhala-style queries to the web UI.
8. Web UI displays Sinhala fashion queries to the user.

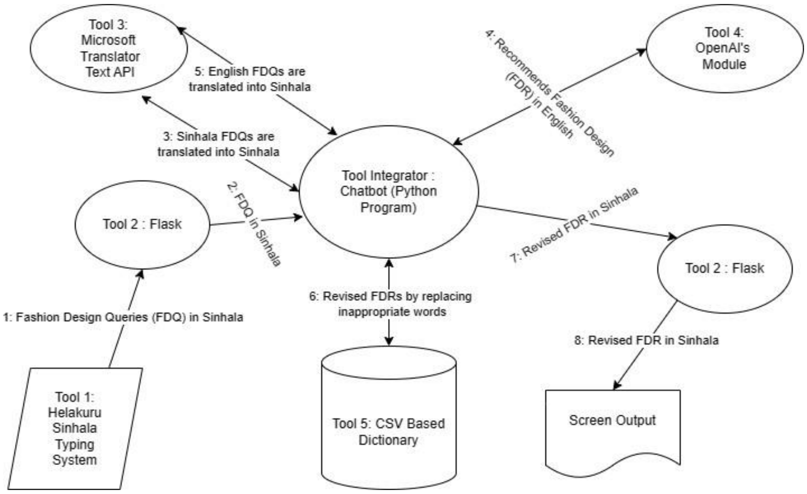
These processes can be repeated because the Chatbot is a conversational model. Users can chat with the Chatbot to get their fashion recommendations.

CBSE - Component Selection

Following the component identified in the first step of CBSE, the required components were selected to develop this Sinhala fashion recommendation chatbot.

Figure 3

Component Selection Diagram



Selecting a Programming Language

Python was selected as the programming language for the development process. Python is an interpreted, high-level programming language suitable for various applications. Its design prioritizes code readability, most notably using significant whitespace. Python's syntax and object-oriented features are intended to help developers create clear, logical code that works well for both small scripts and large-scale projects (Siadati, n.d.). Python is also a popular programming language that has gained widespread adoption across various domains, including academia, research, and industry. Its simplicity, readability, and ease of use make it an ideal choice for non-IT undergraduates new to programming.

Additionally, Python's extensive ecosystem of libraries and packages offers a wealth of functionality, enabling users to tackle a wide range of tasks efficiently. Python's straightforward syntax and clear structure make it highly accessible for beginners, particularly non-IT undergraduates with limited coding experience. The language's design prioritizes readability, facilitating developers' understanding and collaboration. Furthermore, Python boasts an active and supportive community that provides numerous resources, such as documentation, tutorials, and forums, catering to learners at all levels.

Selecting a Web framework

As a web framework, after following the chatbot's requirements, the Python Flask framework was selected among the other Python web frameworks such as Django, CherryPy, Pyramid, Bottle, and Tornado. Flask is a lightweight micro-framework for building web applications quickly. It provides only essential features, allowing developers to add functionality during implementation. As a WSGI application framework, Flask can serve as a backend or handle both backend and frontend tasks. The framework offers an interactive debugger, complete request object, endpoint routing system, and HTTP utilities (Salo, n.d.). Simply Flask allows developers to make web-based applications by combining Python with HTML, CSS, and JS.

One of Flask's key advantages is its status as a microframework, which means it does not require any additional tools or specific libraries, allowing for a more flexible and modular approach to development. This minimalist design philosophy ensures that developers can customize their projects according to their specific needs without being constrained by a rigid set of conventions or dependencies. Overall, Flask's combination of user-friendliness, flexibility, and adherence to industry standards has contributed to its widespread adoption among Python developers, making it a compelling choice for efficiently building dynamic and scalable web applications.

Selecting a Large Language Model

OpenAI’s “GPT-3.5-turbo” model was selected as a fashion recommendation system. GPT-3.5 Turbo is a powerful language model created by OpenAI that builds on the GPT-3 series. It marks a substantial leap in natural language processing skills.

The model's lineage can be traced back to the Davinci architecture, which boasts 175 billion parameters (Ye et al., n.d.).

GPT-3.5 Turbo emerged as an optimization of text-davinci-003, specifically tailored for chat applications. It is the most capable model in the GPT-3.5 series, offering advanced performance at a lower computational cost than its predecessor. This balance of capability and efficiency makes GPT-3.5 Turbo a notable milestone in the evolution of large language models (Ye et al., n.d.).

The OpenAI GPT-3.5-turbo model is capable of prompting. This AI model is equipped with the capability of generating responses based on prompts. To construct a fashion recommendation system using this model, it is required to prompt it with a fashion dataset. Reasons to choose OpenAI modules,

- Only Minimum Coding Required.
- Customization and Personalization.
- Broad Knowledge Base.
- Natural Language Interaction.

The selection of the GPT-3.5 Turbo module over alternatives like GPT-3 or hypothetical versions such as GPT-4 for the fashion recommendation chatbot project is underpinned by its augmented capabilities tailored to natural language processing tasks. Compared to GPT-3, GPT-3.5 Turbo offers enhancements like a more extensive model size, refined training data, and improved language understanding, aligning closely with the project's requirements. Additionally, considering hypothetical options, the proven performance and practicality of GPT-3.5 Turbo render it a preferable choice over an unproven and potentially costlier GPT-4 model. This decision reflects an optimal balance of performance, relevance to project objectives, and practical considerations within the current landscape of AI technology.

Selecting a Language Translator

As a language translator for chatbot, Microsoft Translator Text API service was selected among the other language translator services, such as Google Translator, DeepL, Yandex, and IBM Watson, because Microsoft translator service provides higher quality translations than other services.

The Python CSV library was selected to develop a fashion domain-based dictionary. The necessity of a CSV-based vocabulary was raised due to the limitations of machine language translators, especially in a specialized domain like fashion.

CBSE - Component Adaptation

Adapting the OpenAI Module

OpenAI’s "GPT-3.5-turbo" model has been prompted to use a fashion dataset to create the fashion recommendations system. For prompt testing, the OpenAI playground has been used. After achieving the desired functionality, the chatbot system was easily integrated using code generated from the playground.

Adapting Flask Web Framework

The chatbot's UI is a web-based user interface built using HTML, CSS, and JavaScript. HTML form elements receive user input and represent outputs. CSS styles the web UI, providing a better user experience. Relevant functionalities were added to the web UI with the support of JavaScript. The connection between the Chatbot program and the Web UI was implemented using Flask’s “Request, Render Template, and Jsonify” functions.

Adapting Microsoft Language Translator

The relevant translation functions were created using the Microsoft Azure Python library's capabilities. Following the Azure documentation, after including all the relevant translation functionalities, the defined function was securely stored as a component.

Developing CSV Based Dictionary

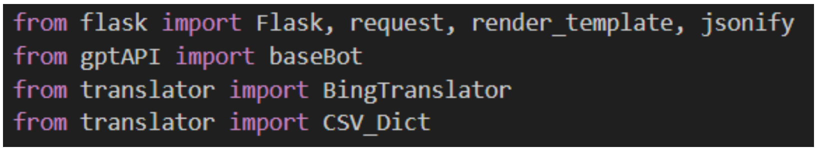
The CSV-based dictionary is essential for finding and swapping out "wrongly translated words" with their correct translations. The Python function ‘DictReader,’ which is part of the ‘CSV’ module, was used to read CSV files as dictionaries. Python regular expressions, which are essential to the restoration process, significantly enhances the implementation. A CSV-based dictionary enhances a chatbot's linguistic accuracy in fashion, ensuring more accurate and appropriate responses.

CBSE - Component Integration

Tool integrator: The chatbot serves as the integration tool to bring these components together.

Figure 4

Python Libraries



```
from flask import Flask, request, render_template, jsonify
from gptAPI import baseBot
from translator import BingTranslator
from translator import CSV_Dict
```

First, all required libraries and functions inside App.py are called.

Figure 5

Defining Home Route

```
app = Flask(__name__)

# Define home route
@app.route('/')
def home():
    return render_template('chatbot.html')
```

Then, basic web application routing and view processing were done using Flask, and a home () view function was created that renders the UI template 'Chatbot.html'.

Figure 6

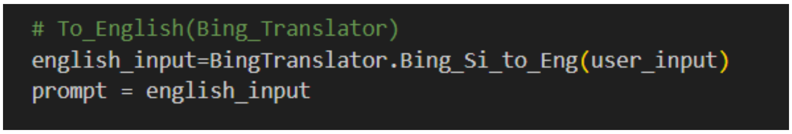
Defining Data Endpoint

```
@app.route("/ask", methods=['POST'])
def ask():
    user_input = str(request.form['messageText'])
```

Then, the Flask route '/ask' was defined as an additional Flask path to request Chatbot responses. This route will integrate all the components. The first logic takes the user text input in Sinhala from a UI field and stores it in a variable called “user_input.”

Figure 7

Calling Sinhala to English Translator function

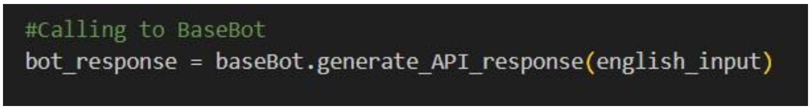


```
# To_English(Bing_Translator)
english_input=BingTranslator.Bing_Si_to_Eng(user_input)
prompt = english_input
```

After that, the user input is given to the language translation function “Bing_Si_to_Eng” to convert the Sinhala text to English. The results are stored in the English input variable.

Figure 8

Calling to GPT prompted module function

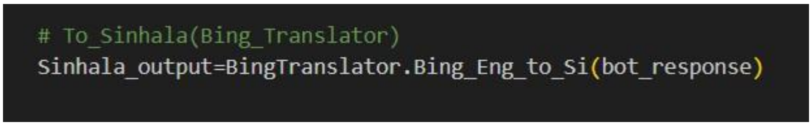


```
#Calling to BaseBot
bot_response = baseBot.generate_API_response(english_input)
```

“english_input” is then passed to the “generate_API_response” function, which interacts with the OpenAI’s API. The “bot_response” variable will save the English response of OpenAI’s module.

Figure 9

Calling to Sinhala to English Translator function

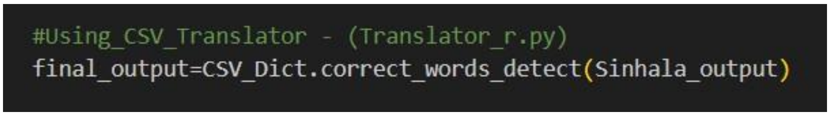


```
# To_Sinhala(Bing_Translator)
Sinhala_output=BingTranslator.Bing_Eng_to_Si(bot_response)
```

"bot_response" is translated from English to Sinhala using the "Bing_Eng_to_Si" function. Sinhala texts are stored in the “sinhala_output” variable.

Figure 10

Calling to Sinhala to Sinhala Word Dictionary Function

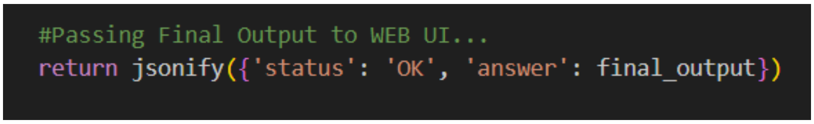


```
#Using_CSV_Translator - (Translator_r.py)
final_output=CSV_Dict.correct_words_detect(Sinhala_output)
```

The CSV dictionary is called in this step, and the variable value "Sinhala_output" is passed to the function parameter. The result is saved in the "final_output" variable.

Figure 11

Pass to Chatbot's Response to Web UI



```
#Passing Final Output to WEB UI...
return jsonify({'status': 'OK', 'answer': final_output})
```

As the final step, using Flask's jsonify function returns the final response stored in the "final_output" variable to the web UI. Finally, the Web UI displays the Chatbot's response.

CBSE – Testing and Verification

The testing scenario is the “Style Preference Exploration.” In this scenario, user queries and responses are as follows.

In Sinhala,

“මම සති අන්ත නිවාඩුවක් සඳහා සාමාන්‍ය ඇඳුමක් සොයමින් සිටිමි. ඔබට හැඩකාර නමුත් සුවපහසු දෙයක් යෝජනා කළ හැකිද?”

English Translation,

“I am looking for a casual outfit for a weekend getaway. Can you suggest something stylish yet comfortable?”

The Chatbot gives the response as,

In Sinhala,

“ඇත්තෙන්ම! විලාසිතාමය හා සුවපහසු සති අන්ත නිවාඩු ඇඳුමක් සඳහා, මම ඉහළ ඉණ සහිත Jeans හෝ ඩෙනිම් කලිසම් යුගලයක් ප්‍රවාහක බ්ලවුස් හෝ ග්‍රැෆික් ටී-ෂර්ට් එකක් සමඟ යුගල කර නිර්දේශ කරමි. සමහර සුවපහසු ස්නීකර් හෝ සෙරෙප්පු සමඟ පෙනුම සම්පූර්ණ කර ක්‍රොස්බොඩි බෑගයක් සහ අව් කණ්ණාඩි සමඟ ප්‍රවේශ වන්න. මෙම ඇඳුම ඔබේ සති අන්තය පුරාම චික් පෙනුමක් සහ සුවපහසුවක් දෙනු ඇත.”

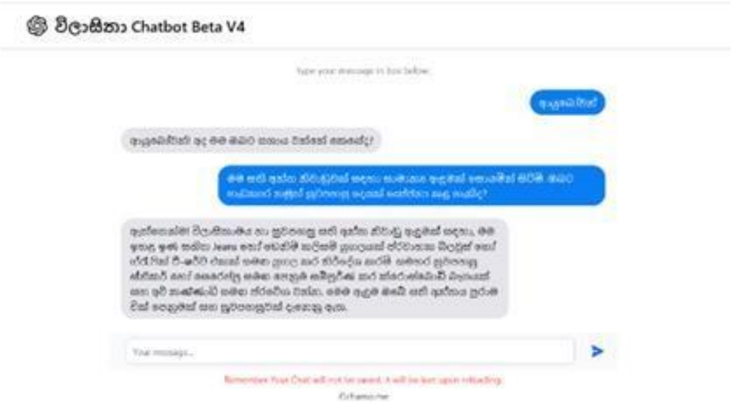
English Translation,

"For sure! For a stylish and comfortable weekend getaway outfit, I recommend a pair of high-waisted jeans or denim pants paired with a flowy blouse or graphic t-shirt. Complete the look with some comfortable sneakers or sandals and accessorize with a crossbody bag and sunglasses. This dress will keep you looking chic and feeling comfortable all weekend.”

Figure 12

Chatbot Testing Scenario

CBSE -



Deployment and Maintenance

Deployment

The chatbot developed is deployed and hosted on a Python application cloud called "PythonAnywhere" (*Host, Run, and Code Python in the Cloud: PythonAnywhere,* n.d.). This cloud platform helps integrate development environments (IDEs) for easily deploying and hosting Python applications. It was hosted to provide users with scalability, high availability, and continuous access to the fashion chatbot.

Maintenance

Maintenance of a fashion chatbot involves several crucial aspects to ensure its effectiveness, relevance, and continuous improvement. These includes.

- Upgrade the fashion dataset.
- Improve the conversation ability.
- Update Sinhala fashion data dictionary.
- Upgrade OpenAI modules.

Challenges

Ambiguous User quarries handling

This chatbot is created using OpenAI's GPT-3.5 module. Its primary purpose is to answer user queries related to fashion recommendations. If a user attempts to ask questions outside the scope of fashion-related topics, the chatbot is designed to recognize such queries as ambiguous or irrelevant. In such cases, the chatbot will politely remind the user that it is specifically designed to handle fashion-related inquiries and will kindly request the user to rephrase their question within fashion recommendations.

Fashion data set update strategies

As mentioned earlier, this chatbot was built by integrating various components. One of these components is the fashion dataset. Fashion is a domain that evolves daily, and numerous fashion variants exist worldwide. Consequently, it is challenging to upgrade the chatbot's fashion dataset instantly. A strategy has been developed to update the fashion dataset monthly by incorporating the latest global fashion trends to address this issue. The primary purpose of this dataset updating process is to ensure that the fashion chatbot remains up-to-date and relevant for users, providing them with the most current fashion recommendations and information.

Furthermore, the fashion industry is a dynamic and ever-changing landscape, with continually emerging new styles, trends, and designs. To maintain the chatbot's relevance and accuracy, it is crucial to refresh its knowledge base with the latest fashion data regularly. By adopting a monthly update schedule, the chatbot can stay abreast of the rapidly evolving fashion world, ensuring that its recommendations and responses reflect the most recent developments in the industry. This proactive approach to dataset maintenance enhances the chatbot's usefulness and fosters user trust and satisfaction, as they can rely on the chatbot to provide the most current and comprehensive fashion guidance.

Evaluation

A questionnaire was created to evaluate the fashion recommendation chatbot, and responses were collected from 20 independent participants.

Table 2

Chatbot Evaluation

Questions	User Answers (As a Percentage)
1. Did the chatbot understand your initial request or question?	● Yes : 100%. ● No: 0%.
2. How well did the chatbot recommend fashion items relevant to what you asked for?	● Very relevant recommendations. : 70%. ● Somewhat relevant recommendations. : 30%. ● Not very relevant recommendations. : 0%.
3. Was the chatbot's Sinhala response fluent and easy to understand?	● Very fluent and understandable : 40%. ● Somewhat fluent : 60%. ● Not very fluent or hard to understand : 0%.
4. Could you carry on a conversational dialogue with the chatbot?	● Yes, conversation flowed well : 85%. ● Some back and forth but not very natural. :15%. ● No, responses were too limited for a conversation : 0%.
5. Did the chatbot provide recommendations specific to Sinhala fashion preferences?	● Very specific for Sinhala fashion :65%. ● Somewhat tailored for Sinhala fashion : 35%. ● Not tailored at all : 0%.
6. Were you satisfied overall with the fashion recommendations from the chatbot?	● Very satisfied : 65%. ● Somewhat satisfied : 35%. ● Not satisfied: 0%.

The majority of the 20 users reviewed the chatbot positively, indicating that it provided useful Sinhala fashion recommendations in response to natural language questions.

Conclusion

Significant of the study

Significance in the Fashion Industry

The fashion design recommendation chatbot represents a significant advancement in the fashion industry. By leveraging AI technology, it offers personalized style advice to consumers, potentially increasing customer satisfaction and sales. This tool can help bridge the gap between online shopping and the personalized experience of in-store stylists, making fashion more accessible to a broader audience. Additionally, it can assist designers by analyzing trends and consumer preferences, streamlining the design process, and reducing waste in production.

Adaptability to Other Domains

The chatbot's design uses component-based software engineering (CBSE), which makes it highly adaptable to other domains beyond fashion. This adaptability is significant because,

- It can be quickly modified for use in interior design, product recommendations, or even career advice.
- The cost and time for developing similar AI-powered recommendation systems in different industries can be significantly reduced.
- It demonstrates a model for creating flexible AI applications that can evolve with changing market needs or technological advancements.

Future Research

Future research directions stemming from this study could include:

- Explore the integration of visual recognition AI to provide recommendations based on user-uploaded images.
- Investigate the long-term impact of AI-powered fashion recommendations on consumer behavior and the fashion industry.
- Develop methods to ensure the chatbot's recommendations remain culturally sensitive and diverse.
- Study how to incorporate real-time fashion trend data to keep the chatbot's knowledge current.
- Examine ways to make the chatbot more interactive through augmented reality features for virtual try-ons.

Limitations

AI Expertise Gap

Building a complex chatbot requires deep AI knowledge, including machine learning and natural language processing. This poses a challenge for non-ICT undergraduates.

Library Selection

Finding an AI library that supports Sinhala language processing is crucial but challenging. Thorough research is needed to identify a suitable option.

Sinhala API Scarcity

Locating or developing APIs for Sinhala language processing and fashion domain knowledge is challenging due to limited resources for less common languages.

Conclusion

This research explores designing and developing a Sinhala fashion recommendation chatbot by developing a reusable chatbot module using the CBSE with a minimum coding knowledge suitable for non-ICT undergraduates. This study aimed to build a reusable chatbot module by combining several components. It is a system that can identify integrated components separately, like a building block structure. If someone wants to create a chatbot based on another non-fashion domain, those developers can easily modify the required components of this Chatbot block separately.

Reference.

Abu Shawar, B., & Atwell, E. (2007). Chatbots: are they really useful? *Empirical Evidence during the Implementation of an Educational Chatbot with the Electroencephalogram Metric*.

Adamopoulou, E., & Moussiades, L. (2020). An Overview of Chatbot Technology. *IFIP Advances in Information and Communication Technology*, *584 IFIP*, 373–383. https://doi.org/10.1007/978-3-030-49186-4_31

Chandrasena, W., Gunesekera, A., Uwanthika, I., Chandrasena, J., Gunasekara, A., & Uwanthika, G. (2021). *Sinhala Chatbot with Recommendation System for Sri Lankan Traditional Dancers Service Provider Recommender System with Sinhala Chatbot for Sri Lankan Traditional Dancers View project human resource management View project Sinhala Chatbot with Recommendation System for Sri Lankan Traditional Dancers*. https://www.researchgate.net/publication/360247377

Chu, S.-C., & Seock, Y.-K. (2020). The Power of Social Media in Fashion Advertising. *Journal of Interactive Advertising*, *20*(2), 93–94. https://doi.org/10.1080/15252019.2020.1802955

Gulia, P., & Palak, P. (2017). Component Based Software Development Life Cycle Models: A Comparative Review. *Oriental Journal of Computer Science and Technology*, *10*(2), 467–473. https://doi.org/10.13005/ojcst/10.02.30

Gümüş, N., & Çark, Ö. (2021). THE EFFECT OF CUSTOMERS' ATTITUDES TOWARDS CHATBOTS ON THEIR EXPERIENCE AND BEHAVIORAL INTENTION IN TURKEY. *Interdisciplinary Description of Complex Systems*, *19*(3), 420–436. https://doi.org/10.7906/indecs19.3.6

Hettige, B., & Karunananda, A. (2006). *Sri Lanka Association for Artificial Intelligence (SLAAI) Proceedings of the third Annual Sessions 15 th*. https://www.researchgate.net/publication/230688513

Host, run, and code Python in the cloud: PythonAnywhere. (n.d.). Retrieved October 15, 2024, from https://www.pythonanywhere.com/

How Many People Have Smartphones Worldwide (May 2023). (n.d.). Retrieved May 29, 2023, from https://www.bankmycell.com/blog/how-many-phones-are-in-the-world

Kumanayake, U. E. (2021). *A Sinhala chatbot for user inquiries regarding Degree Programs at University of Ruhuna*. https://dl.ucsc.cmb.ac.lk/jspui/handle/123456789/4374

Landim, A. R. D. B., Pereira, A. M., Vieira, T., de, E., Moura, J. A. B., Wanick, V., & Bazaki, E. (2022). Chatbot design approaches for fashion E-commerce: an interdisciplinary review. *International Journal of Fashion Design, Technology and Education*, *15*(2), 200–210. https://doi.org/10.1080/17543266.2021.1990417

Lewandowski, T., Grotherr, C., & Böhmann, T. (2021). *State-of-the-Art Analysis of Adopting AI-based Conversational Agents in Organizations: A Systematic Literature Review ExTEND-Engineering of Service Systems for User-Generated Services View project INSTANT View project*. https://aisel.aisnet.org/pacis2021/167

Lin, C. C., Huang, A. Y. Q., & Yang, S. J. H. (2023). A Review of AI-Driven Conversational Chatbots Implementation Methodologies and Challenges (1999–2022). In *Sustainability (Switzerland)* (Vol. 15, Issue 5). MDPI. https://doi.org/10.3390/su15054012

Michaela, E., & Orna, S. L. (2015). Fashion Conscious Consumers, Fast Fashion and the Impact of Social Media on Purchase Intention. *Academic Journal of Interdisciplinary Studies*. https://doi.org/10.5901/ajis.2015.v4n3s1p173

Pressman, R. S. (2010). *Software Engineering: A Practitioner's Approach*. (7th Edition).

Ray, S., & Nayak, L. (2023). Marketing Sustainable Fashion: Trends and Future Directions. *Sustainability*, *15*(7), 6202. https://doi.org/10.3390/su15076202

Salo, K. (n.d.). *Comparative study on Python web frameworks: Flask and Django*.

Sayem, A. S. M., Kennon, R., & Clarke, N. (2010). 3D CAD systems for the clothing industry. *International Journal of Fashion Design, Technology and Education*, *3*(2), 45–53. https://doi.org/10.1080/17543261003689888

Sharma, L., & Gera, A. (2013). A Survey of Recommendation System: Research Challenges. *International Journal of Engineering Trends and Technology (IJETT)*. http://www.ijettjournal.org

Shoufan, A. (2023). Exploring Students' Perceptions of ChatGPT: Thematic Analysis and Follow-Up Survey. *IEEE Access*, *11*, 38805–38818. https://doi.org/10.1109/ACCESS.2023.3268224

Siadati, S. (n.d.). *Fundamentals of Python programming*. https://doi.org/10.13140/RG.2.2.13071.20642

Sommerville, I. (2011). *Software engineering*. Pearson.

Venable, J. R., Pries-Heje, J., Baskerville, R. L., Venable, J. R. ;, Pries-Heje, J. ;, & Baskerville, R. (2017). Choosing a Design Science Research Methodology. In *Australia Choosing a Design Science Research Methodology* (Vol. 2). https://aisel.aisnet.org/acis2017/112

Wang, L. C., Zeng, X. Y., Koehl, L., & Chen, Y. (2015). Intelligent Fashion Recommender System: Fuzzy Logic in Personalized Garment Design. *IEEE Transactions on Human-Machine Systems*, *45*(1), 95–109. https://doi.org/10.1109/THMS.2014.2364398

Weizenbaum, J. (1966). ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, *9*(1), 36–45. https://doi.org/10.1145/365153.365168

Ye, J., Chen, X., Xu, N., Zu, C., Shao, Z., Liu, S., Cui, Y., Zhou, Z., Gong, C., Shen, Y., Zhou, J., Chen, S., Gui, T., Zhang, Q., & Huang, X. (n.d.). *A Comprehensive*

Capability Analysis of GPT-3 and GPT-3.5 Series Models. https://platform.openai.com/docs/model-index-for-researchers

Zietek, N. (2016). *INFLUENCER MARKETING-THE CHARACTERISTICS AND COMPONENTS OF FASHION INFLUENCER MARKETING Thesis for One-Year Master, 15 ECTS Textile Management.*