UPPSALA
UNIVERSITET

Serienummer
Examensarbete 30 hp
Juni 2026

Style Assistant

A Multimodal RAG-based Conversational Recommender System for Second-Hand Fashion

Molly Börjes och Karin Haglund

Civilingenjörsprogrammet i system i teknik och samhälle

UPPSALA UNIVERSITET

Style Assistant: A Multimodal RAG-based Conversational Recommender System for Second-Hand Fashion

Molly Börjes och Karin Haglund

Abstract

Product recommendations present a critical discovery-bottleneck in online second-hand fashion e-commerce due to the large volume of unique items as traditional search and filtering systems struggle to interpret stylistic language. The aim of this thesis was therefore to propose the Style Assistant, a Multimodal Retrieval-Augmented Generation (MM-RAG)-based Conversational Recommender System (CRS) as a potential solution, by delivering technically robust retrieval and recommendation performance while supporting natural language and multi-turn interactions. Following the Design Science Research (DSR) framework, the Style Assistant artifact was designed, developed and evaluated based on the relevance cycle, design cycle and rigor cycle. The evaluation extended established frameworks through a mixed-method framework combining automated LLM-driven RAGAs metrics with an interactive user experience (UX) evaluation based on ISO 9241-11:2018 usability standards in combination with CRS specific quality dimensions proposed by Jannach (2022). The results demonstrated high system effectiveness, where all participants successfully discovered items they liked and would consider purchasing, while latency was identified as a major limitation. The study concludes that the Style Assistant essentially fulfills its aim, demonstrating significant potential to enhance product discovery in second-hand e-commerce. Specifically, it enables users to discover products by expressing themselves naturally with subjective terms suitable for the highly visual domain of fashion. However, limitations were identified and in order to transition the Style Assistant from prototype to a production ready environment further iterations of designing, implementing and evaluating the system are needed. Future research specifically needs to focus on optimizing pipeline latency, hard filtering constraints to fulfill explicit requirements and integrating user personalization.

Teknisk-naturvetenskapliga fakulteten

Uppsala universitet, Utgivningsort Uppsala/Visby

Handledare: Lukas Frösslund Ämnesgranskare: Jessica Lindblom

Examinator: Elísabet Andrésdóttir

Populärvetenskaplig sammanfattning

Att handla secondhand-kläder online har blivit ett populärt och mer hållbart alternativ till nyproducerat mode som dessutom erbjuder ekonomiska fördelar. Något som skiljer plattformar för secondhand-kläder mot annan e-handel är det enorma och ständigt föränderliga utbudet av unika produkter. Detta gör det ofta svårt för användare att navigera bland produkterna och hitta det de söker, vilket kan göra att en del tvekar inför att använda dessa typer av tjänster.

Plattformar för secondhand-kläder använder idag ofta traditionella sökfunktioner och rekommendationssystem, men dessa har begränsningar i denna kontext. Traditionella sökfunktioner är vanligtvis nyckelordsbaserade och kan därför inte hantera vaga sökningar i naturligt språk. Samtidigt kan det vara svårt för användare att beskriva vad de söker, eftersom mode ofta handlar om en känsla, stil eller tillfälle snarare än specifika krav. Detta skapar en mismatch mellan nyckelordsbaserade söksystem och hur kunder vill kunna uttrycka sina önskemål. Samtidigt är vanliga rekommendationssystem svåra att applicera på secondhand-kläder på grund av det stora antalet unika produkter.

I det här arbetet presenteras Style Assistant, ett konversationsbaserat rekommendationssystem för secondhand-kläder. Projektet omfattar kartläggning av problemet, utveckling samt utvärdering av systemet. Systemet möjliggör interaktiva rekommendationer, där användare stegvis kan förfina sina önskemål genom dialog i naturligt språk. För att generera sina svar använder Style Assistant en stor språkmodell (LLM), vilket är en modell som kan generera språk på en nivå likt människor. Multimodal Retrieval-Augmented Generation (MM-RAG) är en metod som används för att komma ifrån problemen med nyckelordsbaserade sökverktyg, samt grunda rekommendationer och meddelanden i produktkatalogen. MM-RAG möjliggör semantisk sökning, vilket innebär att rekommendationerna grundas på betydelsen av användarens sökning istället för specifika nyckelord. Systemet rekommenderar alltså de produkter som bäst matchar användarens förfrågan. Att metoden är multimodal betyder att textsökning kan matchas mot visuella attribut i produktbilder. För problemidentifiering, utveckling och utvärdering av systemet har Design Science Research använts, vilket är en metod för att utveckla tekniska lösningar till riktiga problem.

Resultaten visar att Style Assistant har potential att förenkla hur användare hittar secondhand-kläder online. Under utvärderingen upplevde testpersonerna systemet som hjälpsamt, och alla hittade produkter som de var intresserade av. Framför allt uppskattades Style Assistants förmåga att förstå vagare ord, såsom en stil eller en “vibe”, samt möjligheten att iterativt kunna förfina sina önskemål under konversationens gång. Däremot identifierades svagheter gällande responstid och att specifika krav ibland inte uppfylls. Style Assistant uppvisade alltså potential i att adressera utmaningarna med att hitta relevanta produkter, men skulle behöva vidare utveckling för att fullt ut möta användarnas behov.

Sammanfattningsvis kan Style Assistant vara ett verktyg som förenklar utforskandet av produkter på onlinebaserade secondhand-plattformar för kläder, genom förenklad navigation genom produktkatalogen samt förståelse för naturligt språk. För att utveckla Style Assistant till ett produktionsfärdigt konversationsbaserat rekommendationssystem skulle vidare studier behöva fokusera på att överkomma de tekniska svagheterna samt utforska nya områden såsom personalisering, gränssnitt och faktisk användning över längre tid.

Table of Contents

1. Introduction

While modern growth of second-hand fashion e-commerce platforms has emerged as a vital ecological and economic alternative to the fast-fashion crisis (Claudio, 2007; Guiot & Roux, 2010), product discovery remains a major challenge in these contexts due to the large volume of unique items. Unlike traditional e-commerce, the uniqueness of second-hand products creates challenges for recommender systems due to data sparsity and cold-start problems (Yu et al., 2020; Khatwani & Chandak, 2016). This makes it difficult for users to navigate large inventories and find relevant products.

Fashion is a highly visual and subjective domain with personal preferences often expressed in subjective terms which traditional search and filtering cannot capture. In the context of fashion e-commerce, it can be difficult to find relevant items with specific attributes due to structured filters not being precise enough to capture a highly specific request. Furthermore, it is often difficult to explore products in a desired style or aesthetic, since traditional keyword-based search lacks the ability to semantically understand broader stylistic terms. (Deldjoo et al., 2025; Laenen et al., 2018). Limited historical data as well as unique items makes this even more challenging in second-hand fashion e-commerce.

This discovery bottleneck requires a more dynamic and context-aware approach, motivating the transition toward Conversational Recommender Systems (CRS). Through Conversational AI, a CRS enables users to express preferences in natural language as well as iteratively refine recommendations through multi-turn feedback (Nawara & Kashef, 2025; Gao et al., 2021). While Large Language Models (LLMs) enable CRSs to generate human-like text for the conversation, they are prone to limitations such as hallucination and a lack of domain-specific language (Gao et al., 2024; Nawara & Kashef, 2025).

To mitigate these limitations, Retrieval Augmented Generation (RAG) uses an external product knowledge base to ground the generated responses and recommendations in actual inventory product data. Specifically, Multimodal RAG (MM-RAG) allows product images and product metadata text to be embedded into a shared vector space. This enables semantic understanding, where users can search for visual attributes using natural language (Zhang et al., 2025). Given the visual nature of the fashion domain, a multimodal approach is essential for supporting users in fashion discovery (Deldjoo et al., 2025).

Sellpy, a large-scale online second-hand retailer with around ten million live items, currently relies on traditional product discovery and recommendation systems. To address the previously mentioned product discovery bottleneck, this thesis presents the Style Assistant, a MM-RAG-based CRS designed for second-hand fashion recommendations.

1.1 Aim and Objectives

The aim of this thesis is to address the product discovery challenge in second-hand fashion e-commerce. A Design Science Research (DSR) (Hevner et al., 2004) approach will be used to design, develop and evaluate the *Style Assistant* artifact, based on the relevance cycle, design cycle and rigor cycle (Hevner, 2007). The Style Assistant proposes a Multimodal Retrieval-Augmented Generation (MM-RAG)-based Conversational Recommender System (CRS) as a potential solution to the product discovery challenge in large-scale second-hand e-commerce with unique items. The system aims to deliver technically robust retrieval and recommendation performance while supporting natural language and multi-turn interactions. To evaluate the system performance and quality, both automated Retrieval-Augmented Generation Assessment (RAGAs) metrics and User Experience (UX) evaluation will be used to assess technical accuracy as well as perceived usability and recommendation relevance.

O1. Analyse user needs and product discovery challenges in the context of second-hand fashion e-commerce to identify the design problem.

O2. Design and implement a multimodal RAG-based CRS artifact including retrieval strategy, conversational interaction flow, and recommendation logic.

O3. Evaluate the system performance using RAGAs as an automated evaluation method to assess retrieval effectiveness and recommendation quality.

O4. Evaluate the perceived quality of the recommendations and conversational interactions through UX evaluation based on usability metrics.

1.2 Scope and Limitations

The scope of this thesis mainly focuses on how a RAG-based Style Assistant can support users in product discovery on Sellpy. Both the implementation and the evaluation therefore focused on the relevance of the retrieved product recommendations and the generated content. The Style Assistant was limited to fashion items in order to narrow the scope of the project, and while Sellpy has a wide product catalog, fashion items represent the largest part.

The system includes a user interface, however the purpose of this is mainly to facilitate UX evaluation and demonstrate the functionality of the Style Assistant. The design and usability of the user interface were out of scope and were not taken into account in the evaluation during this thesis. Security aspects, such as handling harmful or inappropriate generated content, were not within the scope of this thesis, and were not handled any further than what was already provided by the Google Gemini API.

2. Background

This section establishes the theoretical framework for the thesis, beginning with the unique challenges of second-hand fashion e-commerce platforms. It then explores the evolution of recommendation systems toward conversational models and concludes with a technical overview of Retrieval-Augmented Generation (RAG) as well as ethical implications.

2.1 Second-Hand E-commerce

While the rise of fast fashion has decreased the lifespan of fashion items, resulting in a massive environmental crisis (Claudio, 2007), it has simultaneously triggered a significant shift in public consciousness. Today, an ethical and ecological desire to distance oneself from this consumerist retail system has fundamentally shifted the mainstream attitude (Guiot & Roux, 2010). Consequently, second-hand shopping has emerged as a vitally important alternative, with both economic and environmental benefits (Claudio, 2007; Guiot & Roux, 2010). As this consumer movement continues to expand online, e-commerce in general, and especially in the context of fashion, offers a large volume of available items. However, a discovery bottleneck occurs when thousands of products are daily updated or listed, each containing images and textual metadata. The mission to find items of interest by searching among this large dataset based on the user's personal preferences can therefore be a difficult task. (Rubio et al., 2017). Consequently, recommendation systems are vital for e-commerce platforms to present the right item to the right user (Khatwani & Chandak, 2016).

Fashion recommender systems differ from other e-commerce recommender systems in several ways. Firstly, they must account for intertwined relationships, which is not as common for other types of e-commerce. There are both item-user relationships based on style preferences as well as item-item relationships capturing how items match based on color, fabric or functionalities. Visual and aesthetic features also play a significant role in fashion compared to other e-commerce platforms, influencing both how items are matched to a user's style as well as when items are matched together. The rapid cycle of trends within the fashion context requires the recommender system to always remain updated to ensure relevance, while higher levels of brand loyalty add another layer of complexity to the recommender logic. (Deldjoo et al., 2025)

These fashion-specific complexities are further increased in second-hand e-commerce, an environment characterized by a “long-tail” nature where items are unique, short-lived and often accompanied by noisy user inputs (Yu et al., 2020; Shen et al., 2012). This creates severe data sparsity, a major obstacle to recommendation model performance (Khatwani & Chandak, 2016). Furthermore, available metadata is typically restricted and does not capture an item’s full semantic meaning, complicating traditional text classification tasks (Shen et al., 2012). Consequently, traditional recommendation frameworks that rely heavily on historical item interactions are ill-suited and inefficient as they are prone to overfitting when learning per-item

latent factors and face exceptionally aggressive cold-start problems since unique items inherently lack historical data (Ye et al., 2023; Wu et al., 2025). To overcome these static constraints, second-hand e-commerce requires more complex context-aware solutions dynamically adaptable to user intent through interactive dialogue (Nawara & Kashef, 2025).

To address these needs, traditional discovery can be categorized in two interaction modes, firstly by searching based on queries from typed search terms, or secondly by faceted browsing through various available categories (Pradhan et al., 2023). In both contexts, structured filters can help the user to narrow down the results by refining their results through filters on category, size, color, price etc (Sinha, 2020).

During discovery and search, traditional systems particularly handle explicit user intent effectively as these are clearly mappable to structured attributes and can be matched to items. Traditional search and filter systems typically support faceted filtering for refinement, autocompletion and suggestion for query acceleration, catalog-aware retrieval for valid results and personalization hooks based on user history and preferences (Sinha, 2020). However, all this requires the user being able to clearly specify their search goal in terms that align well with the product schema. This leaves e-commerce search with some fundamental challenges to understanding user intent from sometimes short or ambiguous queries as well as understanding the products in the catalog in a broad semantic sense (Wang & Na, 2023). In the context of fashion discovery, and especially in second-hand e-commerce, these tasks become further challenging as user requests often are expressed in subjective language in terms of style, vibe or occasion rather than correctly stated search terms. For example, a user searching for “jeans with holes” might not find good matches if these items are described as “distressed jeans” in the catalog (Laenen et al., 2018). In cases like this, traditional systems alone may underperform, even if the underlying search stack is fast and rich in features (Laenen et al., 2018). This does not mean that search and filtering in e-commerce in this traditional sense is invalidated, but rather it motivates the extension of methods to also handle semantic and conversational intent.

2.2 Conversational Recommendation Systems

Conversational Artificial Intelligence simulates human-like conversations through natural language processing (Mamun et al., 2025). Within e-commerce, Conversational Recommender Systems (CRSs) use this technology to enable purchases and product discovery through interactive chatbots (Sidlauskiene, Joye & Auruskeviciene, 2023). These systems offer real-time, personalized, one-to-one interactions designed to enhance usability, engagement and guidance, ultimately driving platform revenue (Sidlauskiene, Joye & Auruskeviciene, 2023; Freitas & Lotufo, 2024). Broadly, the underlying conversational loop initiates with a user message, where this natural language then is processed and evaluated to generate a response. The system then either stops the interaction there, or it loops continuously to iterate new user inputs to the

dialogue. To generate meaningful user value, this complex and iterative process must remain natural and intuitive (Mamun et al., 2025).

By enabling these iterative inputs, a CRS allows users to progressively refine their preferences through natural language queries (Nawara & Kashef, 2025), rather than requiring a complete and clearly stated intent in a single initial turn (Gao et al., 2021). Unlike traditional search and filter systems, a CRS stores and reuses conversation history to improve the relevance and continuity of the recommendation (Freitas & Lotufo, 2024). This kind of system is particularly suited for domains where product discovery is hindered by vague or subjective user intent (Laenen et al., 2018). Compared to traditional e-commerce platforms strictly driven by structured attributes, fashion e-commerce heavily depends on fluid user intents and stylistic descriptions. This makes CRSs particularly valuable as they enable a dialogue for further refinements (Deldjoo et al., 2025). The ultimate purpose is to combine semantic intent understanding with controlled retrieval and filtering in order to help users find relevant results (Gao et al., 2024).

2.3 Large Language Models

The emergence of Large Language Models (LLMs) has transformed the fields of conversational AI and Natural Language Processing (NLP). Constructed using the transformer architecture and trained on massive text corpora, LLMs are capable of processing and generating natural language at a level comparable to humans on many NLP tasks (Teubner et al., 2023). Modern implementations increasingly extend traditional recommendation frameworks through the usage of LLMs to handle limitations like data sparsity, cold-start constraints and limited contextual reasoning. Because LLMs offer the possibility to understand and generate human-like text, they allow the system to dynamically match user preferences while adapting to changing multi-turn contexts and intents (Nawara & Kashef, 2025). Furthermore, LLMs can in some cases improve transparency and consumer trust by generating natural-language explanations for their suggestions (McInerney et al., 2018; Nawara & Kashef, 2025). In fashion e-commerce, these dialogue-based recommendations can reduce decision uncertainty and capture abstract style preferences outside standard categorization, allowing interactive adjustments even for small details. By integrating mixed-modality retrieval users are enabled to interact directly with visual features, maximizing their ability to effectively communicate style preferences. (Deldjoo et al., 2025)

Even though LLMs have a wide set of capabilities, they still carry limitations that complicate their integration into many real applications (Gao et al., 2024). At a foundational level, implementations using LLMs can be interpreted as “black-box” models in cases where explanations of their outputs might only offer limited meaningful interpretability (Mamun et al., 2025; Milano et al., 2020). This opaque decision making constrains systematic accountability and raises challenges for justifying why specific results are presented to the user (Milano et al., 2020). Furthermore, the relational dynamics between the user and the system positions communication and trust as key factors in human-AI interaction (Mamun et al., 2025). One way

to improve trust is through "grounding in communication", a method that decreases misunderstanding by explicitly establishing what has been said to demonstrate systematic comprehension (Mamun et al., 2025). Establishing this transparency and perceived reliability directly influences how safely and effectively users can evaluate options and proceed with confidence (Lopez-Lopez & Iniesta, 2025). Moreover, a main limitation is the models' proneness to hallucination and their lack of domain-specific knowledge. Because LLMs are trained on fixed datasets, they lack a capability to provide domain specific or sometimes up-to-date information. Hallucination refers to LLMs producing linguistically coherent and highly plausible outputs containing factually incorrect information (Gao et al., 2024; Teubner et al., 2023; Jing et al., 2024; Nawara & Kashef, 2025). The models are particularly prone to hallucination in domain-specific or knowledge-intense contexts, where queries exceed the scope of their training data (Gao et al., 2024).

These limitations directly affect the core design considerations when building systems that use LLMs. Strategies include decreasing the model's extent of responsibility in creating all responses, minimizing its usage in tasks especially prone to hallucination, or avoiding completely relying on LLMs during important subtasks directly affecting the user. By applying lighter NLP approaches composed of transformer-based message classifiers and pre-written responses as a first step before using LLMs, the system becomes more robust as well as saves computational costs. (Freitas & Lotufo, 2024).

2.4 Retrieval-Augmented Generation (RAG)

To overcome the challenges with LLMs in the context of CRSs, Retrieval-Augmented Generation (RAG) functions as a key technique by incorporating an external knowledge base. This retrieved data is used to ground the generated response. RAG architectures can be categorized into multiple variants, where naive RAG technology consists of three components; retrieval, augmentation and generation, and where advanced RAG follows the same fundamental pipeline but implements additional pre-retreival and post-retrieval processing to improve performance. (Gao et al., 2024).

While traditional RAG applications are limited to text as a single modality, real-world data often exists in multiple modalities (Shu & Yu, 2025). With the development of multimodal large language models (MLLMs), RAG technologies can handle diverse data types, including images, audio, video, and code. Multimodal RAG (MM-RAG) enables both RAG with multimodal inputs and retrieval, as well as multimodal generation. This architecture supports both multimodal inputs and retrieval, as well as multimodal generation, enabling a system to process queries in one modality and retrieve or generate content in another (Zhang et al., 2025). For instance, it allows a textual conversation to interact directly with visual product attributes.

2.4.1 Vector Database and Multimodal Embeddings

The foundational component of a RAG system is the knowledge base, implemented as a vector database where information is stored as high-dimensional mathematical representations. Before data can be stored, it must be processed in a pre-retrieval phase where its format is standardized and then segmented into smaller chunks. A critical factor in this phase is the retrieval granularity. Coarse-grained retrieval provides more semantic context but may introduce noise and higher computational costs, while fine-grained retrieval offers higher precision for specific matches but risks losing the semantic meaning. Determining the ideal granularity is essential for optimizing the overall performance of the RAG architecture but may vary for different tasks (Gao et al., 2024).

Once segmented, each data chunk is encoded into a high-dimensional vector representation using an embedding model. In MM-RAG, these embedding models must be capable of processing multimodal data and mapping it into a shared high-dimensional vector space. By projecting images and text into a shared vector space, multimodal embeddings allow comparison of data across different modalities, for example allowing a textual query to retrieve semantically relevant matches in forms of images. This vector space represents semantics by ensuring that objects with similar attributes are positioned in close proximity to each other. Similarity within this vector space is determined by calculating the distance between vectors using measures such as cosine similarity or dot product. (Shu & Yu, 2025)

2.4.2 Pre-Retrieval

Original user queries can be ambiguous, incomplete or misaligned with the stored vector data, making them insufficient for direct retrieval. Core challenges in query processing include query complexity, poorly formulated or unorganized language and difficulties in processing domain-specific language. To address these issues, query optimization is employed to refine the user's input into a more precise formulation suitable for vector similarity search. The primary query optimization strategies described in the literature include query expansion, query rewriting, multi-query and query routing. (Zhang et al., 2025; Gao et al., 2024)

Query expansion aims to enrich the original query by incorporating additional semantic information into the query. The expansion is typically executed by an LLM that analyzes the original query and redefines it with semantically stronger information before retrieval takes place. *Query rewriting* instead focuses on rephrasing the query with different wording while preserving the original semantic meaning to account for alternative linguistic expressions. Furthermore, the *multi-query* approach aims at handling complex inputs by dividing the original query into smaller sub-queries whose separate results are later integrated into a single unified response during the augmentation stage. Dividing the query can be achieved using different methods such as query splitting or query decomposition. *Query splitting* divides the query into smaller components, such as sentences or keywords and uses these isolated fragments for retrieval. *Query decomposition* alternatively tries to capture different dimensions of the query, by

splitting the original input into sub-queries that isolate distinct semantic dimensions. (Zhang et al., 2025). Finally, *query routing* aims to improve overall retrieval by directing the query to the most appropriate search pipeline. This process includes, for example, extracting keywords or semantic features to decrease the active search space during retrieval. (Gao et al., 2024)

2.4.3 Retrieval

During the retrieval phase, the optimized query is used to retrieve the most relevant data from the knowledge base. RAG systems can use different types of retrievers; *sparse*, *dense* or *hybrid* (Gao et al., 2024). *Sparse retrievers* use sparse high-dimensional vectors where each dimension corresponds to a specific keyword, meaning that relevance is determined through exact keyword matching. While sparse retrieval is computationally efficient and effective for finding specific terms, it is limited strictly to the text modality and lacks a deeper semantic understanding of linguistic context. *Dense retrieval* instead uses dense high-dimensional vectors to enable semantic similarity matching, allowing the system to identify relevant matches even when they do not share exact keywords with the query (Wang et al., 2024). These dense retrievers are essential for MM-RAG frameworks because they enable similarity searches across diverse modalities within a shared embedding space. Moreover, *hybrid retrievers* combine multiple approaches by combining the exact keyword-match precision of sparse methods with the contextual semantic depth of dense methods, resulting in significantly higher retrieval performance (Zhang et al., 2025).

2.4.4 Post-Retrieval Processing and Augmentation

Augmentation is the process of preparing the retrieved data before the generation step. In a naive RAG architecture this step simply adds the retrieved data to the user query as additional context. However, in an advanced RAG framework, this stage includes post-retrieval processes designed to optimize the quality and relevance of the retrieved data before generation. Such augmentation steps can include reranking, summarizing and fusion. (Winterflood, 2026).

Reranking refers to reassessing the initial retrieved results and assigning weights to them in order to ensure that the most relevant information is prioritized. *Fusion* is the process of combining results from multiple retrieval legs or modalities, which results in a unified set of candidates. Another technique is *information compression* where the retrieved context is reduced in size to fit model constraints while also maintaining its semantic core. *Noise removal* is another preparation step where the generated response is cleaned from irrelevant or distracting segments to improve precision. Similarly, *summarization* is a process where the retrieved information is condensed to provide a more focused context for the generator. (Winterflood, 2026)

The general purpose of these augmentations is to reduce the cognitive load on the generation in order to minimize the risk of the model focusing on irrelevant parts within the retrieved data. Consequently, the augmentation step results in clean and highly relevant prompts for the final component of the pipeline: the generation. (Zhang et al., 2025).

2.4.5 Generation

The generator is the final component of the RAG pipeline, which is responsible for synthesizing the original query in combination with the refined context into a coherent and human-like written response (Shu & Yu, 2025). While traditional RAG uses a standard LLM for textual output, MM-RAG utilizes a MLLM to support various combinations of multimodal inputs and outputs (Zhang et al., 2025). The architecture of a MLLM generator generally consists of five components: a modality encoder, an input projector, a LLM backbone, an output projector and a modality generator (Zhang et al., 2025).

The modality encoder turns raw inputs into features which capture modality-specific semantic representations. Then the input projector maps these features into a text vector space later used by the LLM backbone, which is the core model responsible for the reasoning and synthesis, based on integrated multimodal semantics. The output projector then maps token embeddings from the backbone to a feature space for the modality generator which produces the final output, often using latent diffusion models for non-text modalities. (Zhang et al., 2025)

Performance can be further improved through targeted prompt engineering where the model is guided through input design or fine-tuning. This updates the parameters to align the generator with the retriever or any domain-specific knowledge (Zhang et al., 2025; Gao et al., 2024). All together, these methods allow the system to decrease hallucinations and instead provide factually grounded recommendations (Gao et al., 2024).

2.5 User Experience

The success of a conversational recommender system depends less on the underlying level of AI sophistication and more on the quality of the user experience. If the system fails to engage the user, the technical complexity of the AI is irrelevant as this would lead the system to being unused (Lopez-Lopez & Iniesta, 2025). The usability of a system is defined by the three pillars of effectiveness, efficiency and satisfaction (ISO 9241-11:2018).

Effectiveness measures the accuracy and completeness with which a user achieves their goal and serves as an internal diagnostic for the success of subtasks (ISO 9241-11:2018). In an e-commerce CRS, achieving this accuracy heavily relies on bridging raw user queries and the fixed item metadata (Wang & Na, 2023). This constraint is particularly evident in the context of fashion, where users often use fuzzy language when describing these subjective visual items (Laenen et al., 2018). Tightly related to this is how a system processes negations, which represent negative constraints. If a user specifies that they want a dress but explicitly asks for it “without sleeves”, the system must handle this accurately to provide an effective retrieval (An et al., 2025). To sustain this effectiveness over long-term interactions, the CRS must continuously navigate the “explore-exploit” dilemma. While the system can maximize immediate engagement through exploitation, which means recommending safe and predictable options, it must also introduce exploration by presenting uncertain or diverse items. (McInerney et al., 2018; Yang et

al., 2023). Balancing the two is crucial for minimizing user fatigue and becomes especially critical for cold-start users where the system has limited historical interactions to base recommendations on (Gao et al., 2021).

Efficiency centers on the resources expended, such as time, effort and costs, in relation to the results achieved (ISO 9241-11:2018). Users have limited time and energy, and a failed exploration may expose them to recommendations with low confidence. Unsuccessful exploration represents a lost interaction opportunity, which can negatively impact the user's perception of the system (Gao et al., 2021). Meanwhile, asking many follow-up questions with the purpose of understanding the user's preferences can at the same time be perceived as too demanding, potentially leading to boredom and user abandonment. (Gao et al., 2021) Because multimodal models often incur higher latency and resource consumption, maintaining this efficiency is a central design challenge (Zhang et al., 2025). Interactions typically end naturally when the user either is satisfied with the recommendations or becomes too impatient to continue their search (Mamun et al., 2025; Gao et al., 2021). Therefore, the system strategy should aim to use the least number of turns possible while still giving good enough results (Gao et al., 2021).

Ultimately, satisfaction is defined by the extent to which user responses meet needs and expectations, emphasizing comfort and acceptability (ISO 9241-11:2018). This aligns with the perceived quality of the conversation, as users value naturalness, relevance, and speed for continued usage and loyalty (Lopez-Lopez & Iniesta, 2025; Jannach, 2022). While functional values reflect the perceived usefulness of the system, emotional values reflect emotional attachment and enjoyment. Traditionally, the main driver has been functional values, however the importance of emotional values is increasing as users value enjoyable experiences higher. These emotional values require the AI to comprehend human communication and reply appropriately (Mamun et al., 2025). Human-like attributes are vital design elements since perceived personalization from human-like traits leads to familiarity, similarity, and liability (Lopez-Lopez & Iniesta, 2025; Sidlauskiene et al., 2023). If the system fails to engage the user emotionally, the underlying technical complexity remains irrelevant, as the system will likely go unused (Lopez-Lopez & Iniesta, 2025).

3. Methodology

This study was based on Design Science Research (DSR) methodology (Hevner et al., 2004), since the aim was to design, develop and evaluate the Style Assistant as a technological artifact to address a real-world product discovery challenge while ensuring methodological rigor. The methodology was structured according to the three cycles of DSR: the relevance cycle, the design cycle and the rigor cycle.

3.1 Design Science Research

Design Science Research (DSR) is an Information Systems (IS) paradigm, focusing on the creation and evaluation of innovative artifacts that address real-world organizational challenges. Utility is the primary goal of DSR, referring to an artifact being effective and useful within its context (Hevner et al., 2004). According to Hevner et al. (2004), this search for an effective artifact requires employing available *means* to reach desired *ends*, while also satisfying environmental laws. The design process can be divided into three closely related cycles: the relevance cycle, the design cycle and the rigor cycle (Hevner, 2007). In the relevance cycle, the design problem is identified by analyzing opportunities and problems in the application environment (Hevner, 2007). This cycle establishes the *ends* of the research, defined by Hevner et al. (2004) as identifying specific goals, requirements and constraints that the final artifact must satisfy. The design cycle is the central cycle in DSR according to Hevner (2007), executing a rapid internal loop dedicated to the construction of the design artifact. Within this cycle, the required functional *ends* and constraints established by the relevance cycle are iteratively transformed into concrete technical choices until a satisfactory design is achieved. Lastly, the rigor cycle, according to Hevner (2007), positions the entire research project in the context of the existing knowledge base of scientific theories and engineering *means* to ensure structural innovation and guarantee that the artifact produced contributes to current research (Hevner et al., 2004). The rigor cycle grounds the research on appropriate evaluation frameworks to evaluate the artifact as a whole (Hevner, 2007).

3.2 The Relevance Cycle

The relevance cycle is the first phase of Design Science Research, as presented by Hevner (2007). This phase starts by analyzing the contextual environment of the research project, resulting in requirements based on identified opportunities and problems. The goal is to create a foundation for developing an artifact that improves its research environment or certain practices within it. (Hevner, 2007)

Therefore, this section analyzes the problem relevance with the purpose of identifying opportunities and limitations in the current product discovery functionalities. These insights formed the required *ends*, which led to the foundational design of the Style Assistant as an

artifact intended to support users in product discovery in an improved way. To systematically capture these *ends*, this relevance cycle consisted of two main sources of collected insights from the environment: an assessment of current product discovery functionalities in the application, and a questionnaire aiming to collect user and stakeholder expectations of the Style Assistant. These were further complemented by discussions about current product discovery challenges and potential future solutions.

3.2.1 Situation Assessment

Sellpy is a large-scale online second-hand retail company, with approximately ten million of unique items live on site. Product discovery is a fundamental challenge in second-hand e-commerce due to interaction data sparsity, long-tail distribution and cold-start problems. (Yu et al., 2020; Khatwani & Chandak, 2016). Currently, Sellpy has several functionalities for product discovery including a search function with filtering options, user recommendations and pre-curated stores. Despite this, product discovery remains as a main challenge for users, translating into an organizational challenge. Two main sub-problems were identified: limited specific search, and difficult exploration.

The limited specific search problem can be related to the search engine and filtering options which relies on keyword matching and structured metadata but lacks the semantic understanding provided by dense vector retrieval (Sinha, 2020; Gao et al., 2024). This means that a user's subjective and fuzzy fashion language often doesn't align with the fixed metadata (Shen et al., 2012; Laenen et al., 2018). For Sellpy, this means that attributes not explicitly captured in the metadata, such as a particular fit or style, remain unsearchable. The current filtering system is optimized for structured attributes like size, brand, condition, price, demography, color, pattern, material, fabric etc, but it lacks the ability to capture the interface for users to express more complex and subjective fashion intents.

The second subproblem involves discovery without explicit prior intent, where users browse for inspiration. At Sellpy, this is currently limited to browsing through pre-defined categories, which means that the user needs to identify what they are looking for in a context where it is difficult to navigate to a certain style. Because the current search engine cannot handle visual style attributes, it cannot recommend items that share a vibe or suit a specific occasion unless those terms are already present in the metadata (Zhang et al., 2025), making exploration bounded to pre-defined categories.

3.2.2 Expectations and Use Cases

To assess user expectations of the Style Assistant a survey was conducted, see Appendix A for the full questionnaire. The survey consisted of four questions focusing on expected functionalities, typical use cases and example queries to examine how users would interact with the new system. A total of 11 responses were collected, where all participants were both Sellpy

employees and Sellpy users. Responses were analyzed to identify recurring expectations and use cases of the Style Assistant.

Analysis of the responses showed that the Style Assistant was expected to be effective and able to give more advanced semantic recommendations compared to traditional keyword-matching. Some recurring expected functionalities were the ability to recommend outfits, match items to occasions and handle user feedback. It was noted that users would use the Style Assistant as a way to overcome filter limitations, both regarding specific product searches and for getting broader inspirational searches. Furthermore, it was important that the system understood broad stylistic fashion terms and concepts in order to give more accurate recommendations. Also, it was required to handle filters like brand, price and material. Overall, the results suggested that users expected the Style Assistant to support more flexible and natural interactions compared to existing product discovery functionalities.

3.2.3 Style Assistant

The problem analysis, including a situation assessment and a conducted survey on user expectations resulted in a positioning of the problem relevance for the new artifact. The result of the design cycle is therefore the Style Assistant artifact, which aimed to be an effective solution by utilizing available *means* to reach desired *ends* while satisfying environmental laws, which are the inherent rules and limitations of the environment to which a design must adapt, as defined by Hevner et al. (2004). In the context of second-hand e-commerce, these environmental laws relate to the uniqueness and short lifespan of individual second-hand items. Consequently, the identified design problem is to develop and evaluate a conversational recommender system capable of supporting both specific and exploratory fashion searches through natural language and semantic understanding, as well as handling iterative user feedback.

In a design problem, Hevner et al. (2004) refers to *ends* as the goals and constraints of the solution. For the Style Assistant artifact, a number of different *ends* were identified from the design problem analysis which the final artifact was required to satisfy. These foundational *ends* required the user to be able to use natural language to describe what fashion items they were looking for and to be able to specify details or attributes that the current search and filtering functionalities do not allow. The system must be able to both understand specific requirements, such as “I want an oversized dark wash denim jacket with distressed details” and broader stylistic terms not captured by filters, for example “I am looking for professional clothing for a job interview”. Also, users had to be able to address what they do not want in natural language, making handling negations an *end*.

The filtering process was designed to be handled automatically by interpreting user input and extracting filters such as brand, material and price from it. Multi-turn conversation had to be handled by the Style Assistant, where previously stated user preferences must be remembered throughout the conversation. The user also needs to be able to refine their requirements by

multi-turn conversation, for example by writing refinements such as “I want a lighter color” or “More bohemian style”. Moreover, the generated messages were intended to feel natural to the user, as well as be grounded in the conversation and retrieval. Based on these *ends*, the Style Assistant aims to offer easier navigation through the large product catalog by enhancing the relevance of a smaller amount of items presented to the users.

While the *ends* establish the objectives, *means* instead refers to the resources and actions available to construct a solution. To reach the desired *ends* of a design problem, available *means* are utilized (Hevner et al., 2004). To address this design problem, the Style Assistant artifact was designed as a RAG-based conversational recommender system as its specific technical *means*, intended to improve product discovery beyond traditional keyword-based approaches. The RAG-based approach contributed to research rigor (Hevner et al., 2004), since RAG is an established method in similar systems for enabling semantic product retrieval as well as grounding generated responses in retrieved product data. The *means* used to achieve the Style Assistant design problem solution are further presented in Chapter 4.

3.3 The Design Cycle

Hevner (2007) refers to the design cycle as the heart of DSR, where the required *ends* from the relevance cycle are matched against technical *means* and requirements from the rigor cycle to ensure methodological rigor. While the design cycle is intertwined with the other two cycles, it still serves independently while the design research is performed. By iterating between construction of the artifact, evaluation and following feedback, the design cycle further refines the design of the artifact. This means that the design cycle iteratively evaluates the artifact against the requirements until they are satisfied. Both the construction and evaluation of the artifact must be based on problem relevance and rigor, and it is important to balance the effort spent on these (Hevner, 2007).

Aligned with DSR, the Style Assistant was developed and evaluated as an artifact aimed to address the product discovery challenge in second-hand fashion e-commerce, identified in the relevance cycle. Multimodal Retrieval-Augmented Generation (MM-RAG) was identified as a rigorous method for building a CRS. This approach allowed for natural language communication as well as multi-turn refinements, while ensuring that generated messages and retrieved product recommendations were grounded in actual product inventory data (Gao et al., 2021; Nawara & Kashef, 2025). During the design cycle the Style Assistant was developed to address the identified design problem and it progressed through a series of organic development iterations focusing on different features of the artifact, all in relation to the established requirements from the relevance cycle. These development phases were generally categorized as follows:

Iteration 1: Baseline Retrieval and Simple Prompting

Iteration 2: Intent Classification and Query Rewriting

Iteration 3: Filter Extraction Heuristics and LLM Fallback

Iteration 4: Reference Pinning, Vector Weighting, and Negations

The construction of the final Style Assistant pipeline is further examined in Section 4. While these rapid internal design cycles are based on continuous, informal smaller evaluations to refine the artifact's technical features, Hevner (2007) notes that a design science artifact ultimately had to be evaluated in a scientifically grounded manner. Consequently, the resulting artifact must be evaluated through rigorous methodologies as presented by the rigor cycle in the next section.

3.4 The Rigor Cycle

The rigor cycle as presented by Hevner (2007), is where the research project is positioned in the context of past knowledge to ensure innovation and guarantee that the produced designs contribute new insights to the research. In practice, this includes the selection and application of appropriate theories and methods for constructing and evaluating the artifact to ensure methodological rigor. In this context, the evaluation frameworks from the knowledge base serve as a scientific evaluation *means* necessary to verify whether the completed artifact effectively achieves its established *ends*. Research contributions are defined as any extension to the original theories and methods made during the research. (Hevner, 2007)

In the context of second-hand e-commerce, MM-RAG based CRSs is positioned as an extension to the traditional product discovery methods. In this research project, the MM-RAG based CRS was used to construct the Style Assistant which was later evaluated through a hybrid of existing frameworks. Due to the complex nature of CRSs, a mixed-method evaluation that combines objective computational measures with subjective UX measures is beneficial (Jannach, 2022). This kind of solution is especially relevant in the context of fashion recommendations as such preferences are highly personal and subjective (Deldjoo et al., 2025). More specifically, the technical performance of the Style Assistant was evaluated through the RAGAs framework proposed by Es et al. (2024) which evaluates the RAG pipeline through automated metrics. However, RAGAs does not capture the subjective perception of the recommendations, which required it to be complemented by an UX evaluation. The subjective evaluation of the Style Assistant’s usability was based on a hybrid approach where usability standards of ISO 9241-11:2018 were extended with CRS specific quality dimensions proposed by Jannach (2022).

3.4.1 RAGAs Evaluation Framework

RAGAs (Retrieval Augmented Generation Assessment) is a framework proposed by Es et al. (2024) for reference-free evaluation of RAG systems. The RAGAs framework aims to evaluate both the retrieval and the generation of a RAG system without relying on a ground truth by using LLM as a judge. Retrieval and generation performance are evaluated separately with respective metrics. For retrieval, *context relevance* is proposed as a quality metric referring to that the retrieved documents should exclusively be relevant to answer the query. For generation, the two

proposed metrics are *answer faithfulness* and *answer relevance.* Answer faithfulness refers to that the answer should be grounded in the context, aiming to avoid hallucinations in the generation. Answer relevance refers to that the generated answer should answer the query input. All RAGAs metrics provide scores on a continuous scale between 0 and 1, where higher values indicate better performance. (Es et al., 2024)

These metrics are evaluated based on query-passage-answer triples. Whereas the query is the initial user request or question, the passage is the retrieval context containing a relevant knowledge base used for retrieval to help answer the query, and the answer being the final response generated by the system (Es et al., 2024). RAGAs automatically evaluate these triples based on the three quality aspects using an LLM and specific prompting strategies which focuses on the alignment and relevance between the components of the triples. The way RAGAs uses LLM as a judge has been shown to align more closely with human judgement than other similar baselines. (Es et al., 2024)

3.4.2 RAGAs Evaluation Protocol

To evaluate the technical quality of the Style Assistant RAG pipeline without having to rely on manually authored ground-truth answers, RAGAs was applied to logged query-context-answer triples. This suited open-ended, multi-turn searches where many replies were acceptable and where strict correct answers were often undefined. (Es et al., 2024). The aim was to provide an automated analysis of how successfully the Style Assistants specific *means* meet their corresponding functional *ends*.

The evaluation was driven by a scenario catalog built of 29 different scenarios divided into 16 different groups, which aimed to evaluate different types of queries. These groups included specific search, exploration, filters, UI filters, negations, reference, and refinements. Dividing the scenarios into groups enabled assessment of the system performance on different types of queries. In total, the catalog contained 41 step-level chat turns where each step produced one evaluation row. Evaluation scenarios were executed in a reproducible manner based on a script that also preserved session order for multi-turn cases. Each step produced one row in the exported table and those rows were passed to RAGAs which scored each row individually. Summary statistics were computed overall, per scenario and per scenario groups.

Each evaluation scenario consisted of a user search query, which was captured as a text message, and an accompanying “extras” payload. This payload contained explicit non-visual filters, such as demography, size and reference item data. Some scenarios represented multi-turn dialogue and therefore included several iterations of messages and extras. The scenarios were used for building a RAGAs dataset by running the scenarios against the Style Assistant API. The RAGAs dataset is then used for computing the desired RAGAs metrics. For each scenario, the RAGAs dataset is constructed using the query, which was the user message; the context, which was the textualized retrieved content of the retrieved products with corresponding vector database

payload attached; and the answer, which was the assistant's generated answer. The textualized retrieved content consisted of following attributes: rank, id, title, brand, color, size, price, score, and optional material, fabric, pattern, sleeve length and garment length.

During the evaluation, three RAGAs metrics were used. To evaluate the retrieval, *context relevance* determined how much of everything retrieved was actually useful for answering the query. That meant that irrelevant hits, such as items from the wrong category or results not fulfilling the requirements constraints, led to a lower context relevancy score. (Es et al., 2024). For example if the user's query asked for a floral short-sleeved shirt in linen, and the results were products with these attributes stated in product metadata, this typically resulted in a high context relevance score. In contrast, a low context relevance score resulted if the results contained unrelated attributes, for example wrong patterns or product types. In practice, this was calculated by the judge LLM based on the query, the model answer and the single context chunk for each retrieved context, which then decides whether the context was useful for producing the answer (Es et al., 2024; Exploding Gradients, 2026). The exact system prompt template is provided in Appendix B.

Answer faithfulness was the metric used to decide if the assistant claims were supported by the logged retrieval context or not (Es et al., 2024). High answer faithfulness meant that the reply only described the actual retrieved results, in this case floral short-sleeved linen shirt, meanwhile, a low answer faithfulness score was recorded if the assistant for example claimed that it had avoided all other patterns, while some products in the context included other patterns than floral. In practice, faithfulness uses two LLM passes (Es et al., 2024), firstly statement decomposition from the pair of question and answer, and secondly verification for each statement against the retrieved context, using the prompt sequence shown in Appendix B.

The third evaluated quality metric was answer relevancy which checked whether the reply matched the user's request by generating a potential input query from the answer and computing the similarity between the original query and the generated queries (Es et al., 2024). A high answer relevancy score in this case would mean that the reply directly addressed floral short-sleeved linen shirts. On the other hand, a low score meant that the answer drifted off-topic, for example by discussing unrelated subjects. To evaluate the answer's relevance without relying on any ground truth, the LLM was prompted in a way to generate several potential questions based on the generated answer; the specific evaluation prompt formulation is provided in Appendix B (Es et al., 2024; Exploding Gradients, 2026).

One limitation of RAGAs as evaluation in the context of the Style Assistant was the fact that it does not use product images, and instead its metrics only used textualised retrieval. As a result, the scores reflected the logged text and not any visual similarities. This meant that visual features not expressed in the metadata, such as style, vibe, occasion, fit, were not captured by the RAGAs evaluation. The lack of visual understanding potentially introduced lower RAGAs scores despite

the recommendations being stylistically relevant. Furthermore, RAGAs could measure perceived usability, trust or other recommendation satisfaction, meaning it could not fully evaluate subjective UX *ends*. For this reason, automated RAGAs results, which highlighted the technical performance of the *means*, were interpreted together with UX findings from the following section. This combined analysis enabled a more complete assessment of both pipeline quality as well as experienced quality by users, capturing additional aspects beyond metadata.

3.4.3 User Experience Evaluation Framework

The UX evaluation of the Style Assistant is based on a hybrid framework, integrating the usability standards of ISO 9241-11:2018 with the quality dimensions proposed by Jannach (2022) specifically for the conversational recommender systems. The hybrid approach adapts the usability definition to the Style Assistant domain, ensuring rigorous usability evaluation of the Style Assistant artifact. Usability is defined by ISO 9241-11:2018 by three measurable pillars that describe the quality of user interaction with a system: *Effectiveness, Efficiency* and *Satisfaction*. This standard treats usability as an outcome of interaction rather than a static product attribute.

Within the context of CRS, evaluation is inherently interactive and user-centric, requiring both objective and subjective measures to capture the user experience and subjective quality perceptions (Jannach, 2022). While traditional quality dimensions for recommender systems include prediction accuracy, item coverage, novelty, serendipity and diversity, four primary UX dimensions for CRS evaluation have been identified in the literature: Effectiveness of Task Support, Efficiency of Task Support, Quality of the Conversation and Usability, and Effectiveness of Subtask (Jannach, 2022).

To adapt the three ISO pillars for a conversational recommender system, these are mapped to Jannach's (2022) four evaluation dimensions, resulting in the following combined evaluation metrics:

Effectiveness: The ISO standard defines this as the accuracy and completeness with which users achieve goals (ISO 9241-11:2018), which in the context of the Style Assistant measures the external outcome whether the users found items they would purchase. This pillar integrates two concepts from Jannach (2022): *Effectiveness of Task Support*, which is the primary measure of the system's ability to help users in product discovery and includes traditional recommendation quality metrics, and *Effectiveness of Subtask*, which serves as an internal diagnostic focusing on the success of individual components like intent classification, filtering and prompt generation.

Efficiency: Based on the ISO definition, this measures the resources used, such as time, effort and costs, in relation to the results achieved (ISO 9241-11:2018). This is applied through Jannach's *Efficiency of Task Support*, which evaluates how well the system performs its tasks in terms of time or perceived effort.

Satisfaction: The ISO standard defines satisfaction as the extent to which user responses meet their needs and expectations, emphasizing comfort and acceptability (ISO 9241-11:2018). This corresponds with Jannach's dimension of *Quality of the Conversation and Usability*, which discusses the subjective experience and linguistic aspects of the conversation, aligning the perceived usability of the system with ISO's holistic focus on Satisfaction.

These combined dimensions take the interactive nature of CRS into account while still incorporating traditional usability metrics. (ISO 9241-11:2018; Jannach, 2022)

3.4.4 User Experience Evaluation Protocol

To complement the quantified and controlled RAGAs evaluation, the UX evaluation of the system aimed to capture a subjective and human-oriented dimension of the performance and quality of the CRS. This allowed for the assessment of the perceived relevance of the recommendations and system performance, as well as how likely users would be to use the Style Assistant again in real-life scenarios. A total of nine participants took part in the evaluation. This sample size was considered sufficient to capture insights regarding user perception of the CRS. The participants were required to have prior experience in using Sellpy as a second-hand shopping platform, as well as an interest in second-hand shopping, since Sellpy users represented the main target group of the system. Other than that, no domain-specific knowledge was required.

Each session started with a short introduction of the main functionalities of the Style Assistant. Since the goal of the user-centered evaluation was to evaluate the system performance in regards to its recommendations and not any aspects of the user interface, the participants were briefly instructed how to interact with the UI. The evaluation consisted of two main scenarios, each repeated twice to improve coverage and consistency. For each scenario, the user was asked to produce a search query based on products that they were currently interested in, in order to actually be able to determine whether the recommendations were good or not. During the evaluation session, each user produced two specific search queries and two broader exploratory searches. The user was encouraged to interact with the system until they either found a suitable product or gave up.

Scenario 1–2 (specific search): Participants were asked to generate two specific product queries with clearly defined requirements. The goal was to evaluate the system's ability to support specific search and recommend products following explicit requirements.

Scenario 3–4 (exploration): Participants were asked to generate broader queries based on style, inspiration, aesthetic, or occasion. The goal was to evaluate the system's ability to support exploratory search and provide diverse, inspirational recommendations.

At the end of the session, two open questions about the system's strengths and weaknesses were asked in order to capture the user's general perception of the system. Afterwards, the user was

asked to fill in a questionnaire based on ISOs usability definition and the four evaluation dimensions of conversational recommendation systems suggested by Jannach (2022). The complete post-evaluation questionnaire is provided in Appendix C. The questionnaire includes 21 questions grouped based on the usability dimensions. Most questions used a Likert scale where the user rated how well the question corresponded to their perception from 1 to 7. Some questions were direct yes/no questions to determine whether the user succeeded or failed with the main task of the system.

During the UX evaluation, all data from the sessions, including response time and the number of iterations, was collected in a NoSQL database. Additionally, it was noted whether each session resulted in a success or failure, defined by whether the user found something they liked or if they gave up. This data was later used for computing the average response time, average number of iterations and fractions of success versus failure to be used during the results and discussion. By combining objective metrics, subjective Likert-ratings and open feedback, the questionnaire aims to provide an in-depth assessment of how well the RAG-based CRS supported product discovery and the quality of its recommendations.

To protect research participants and ensure good research practice in this part of the research involving humans, four general requirements for research ethics were followed. First, the information requirement which established that the researchers must inform the participants about the purpose of the research project, the methods and the terms and conditions for participation. Secondly, the consent requirement ensured that the participants had a right to decide for themselves whether they wanted to participate and that they had the right to withdraw their participation at any time, without negative consequences. Thirdly, the confidentiality requirement stated that any personal data had to be stored and handled in such a way that unauthorised persons could not access them. Lastly, the use requirement: ensures that the collected data would only be used for research purposes and not for commercial or other non-scientific purposes; this also applied to how the data was stored and for how long (Vetenskapsrådet, 2024). In the beginning of the session, the participants were informed about these four principles in order to ensure good and ethical research.

4. Artifact Design and Implementation

The outcome of the design cycle is the built artifact, the Style Assistant. Based on the required *ends* from the relevance cycle and the established *means* identified in the design cycle, grounded in the environmental laws and established methods as presented in the rigor cycle, this artifact aims to improve the established research area. In the context of second-hand e-commerce fashion, this includes enabling searches through natural language with a semantic understanding as well as a multi-turn conversation for further refinements. In practice, this CRS uses a RAG pipeline, as visualized in Figure 1, which starts with the user input, in the form of a search request in natural language. Then this query is processed in certain pre-retrieval steps, including intent classification, filter extraction, query optimization, multi-turn continuity, negations and reference item handling. Thereafter, the retrieval phase conducts a similarity search where this request is matched against the vector database of the available products. These results are later structured and prepared during the post-processing phase and from this the system generates the result in the form of a short acknowledgment, the actual recommendations and three example prompts. These are then presented in the UI for the user to see and interact with, enabling a loop over this pipeline as its functionality as a CRS enables the user to continue the conversation by sending new requests. All these steps will be further explained in the following sections.

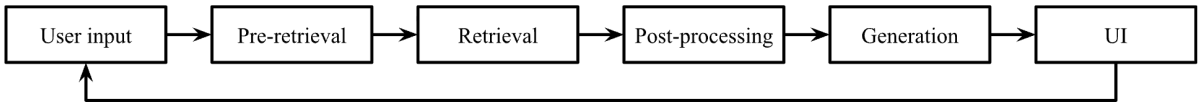

Figure 1: The Artifact System Pipeline.

4.1 Technical and Software Details

The backend was developed in Python using the FastAPI framework to ensure efficient integration with the organization's existing technology stack. Together with the Google Gemini API, specifically the Gemini 2.5 Flash model, for LLM-based tasks and Qdrant as the vector database, these software choices were the foundational technical *means* selected for the system. Qdrant was selected as the vector database, as it was already established within the organization's infrastructure to manage the high-dimensional product embeddings. The frontend was built using React for a responsive user interface and the entire application was deployed via Amazon Web Services to ensure compatibility with existing software within the organization.

4.2 Embedding Model and Vector Database

The embedding model encodes each product as a vector representation. In this case, a multimodal embedding model encodes the product image, brand, size and metadata into a shared 2241-dimensional multimodal vector representation. This vector space is used in retrieval for finding the nearest neighbors to a search query, using the dot product as the similarity metric.

This dense retrieval enables semantic similarity matching as vectors are closer if they share visual attributes, such as shades of colors or styles (Wang et al., 2024). By using a multimodal embedding model, images and text are embedded into the same vector space, meaning that users can search on image attributes using text, which is an advantage in fashion retrieval because of its visual importance (Deldjoo et al., 2025). Apart from the vector representation, each Qdrant point also has an attached payload, which consists of all item metadata. During the retrieval step, Qdrant supports payload filtering.

4.3 Pre-Retrieval Processing

The pre-retrieval step applies the technical *means* of the RAG pipeline for the Style Assistant aiming to achieve system *ends* by preparing the user input query for the retrieval phase, focusing on intent classification, filter extraction and query optimization.

4.3.1 Intent classification

Based on current limitations with product discovery within second-hand e-commerce, the relevance cycle identified three main use cases for the Style Assistant. These require the new system to enable searches based on specific requirements, broader inspirational searches and lastly refinements through multi-turn feedback. The Style Assistant artifact was therefore built around three main intents; search, explore and refine. The search intent corresponds to specific search queries, where the user has specified certain attributes about the product that they are looking for. The exploration intent on the other hand corresponds to explorative search queries that reference a certain style, vibe, occasion, location etc. The refine intent acts as the conversational *means* to store and reuse previous interactions for continuity and relevance. This capability is a core function of a conversational system (Freitas & Lotufo, 2024).

Intent classification is the first pre-retrieval processing step in the RAG pipeline, aiming to understand the user intent behind the request in order to give relevant results. To understand user intent from the user query is one main challenge within e-commerce in general (Wang & Na, 2023) while also being especially relevant in domains like fashion where user intent can be vague or subjective (Laenen et al., 2018). As a technical *means*, an LLM is used for classifying the intent of the user message. By making the system intent-aware, the rest of the RAG pipeline is adapted to the classified intent through query routing (Gao et al., 2024). Additionally, the intent classifier of the Style Assistant determines whether the user has specified an item type in their message, such as “dress”, ”rain jacket” or “bags”. The following steps of the pipeline are then handled differently depending on whether an item type was specified or not. Query routing then decides the rest of the pipeline for the pre-retrieval processing steps and the vector retrieval.

4.3.2 Filter Extraction

Aligning with expectations from the user survey, the Style Assistant processes filters such as brand, price and color directly from the user message. All extracted constraints are managed through the Qdrant payload.

To optimize integration, demography and size filters are handled directly via the UI, leveraging existing Sellpy accounts and their size profiles. Such explicit filters are passed straight to the backend candidate pool. On the other hand, brand, price and material filters are dynamically parsed from user messages using a hybrid extraction strategy. To maximize execution efficiency, heuristic functions first check the user query. If any facets remain unresolved, a single combined LLM call is invoked to extract the remaining bounds or facets. This selective approach prioritizes robustness and minimizes computational costs by invoking the LLM only when necessary (Freitas & Lotufo, 2024).

The system normalizes all natural language extractions to ensure compatibility with catalog vocabularies. For price, the system detects both explicit numeric constraints, such as “SEK” or “kr” bounds, as well as relative changes such as “cheaper” or “more expensive”. For brands and materials, heuristics isolate inclusions or exclusions, which are then mapped to existing values in order to sanitize inputs and by grouping related variants together into a single entity. This mapping includes all possible options for material, and for brands the top 1000 most common brands.

In contrast, highly visual attributes shown in the product images, such as color shades, fit, fabric or length, are excluded from hard payload filtering. Instead, they are included as a part of the text query vector. This is possible since the multimodal embedding model can handle searching for visual attributes in an image using text (Rubio et al., 2017; Shu & Yu, 2025). Hard filtering these attributes could constrain the flexibility identified as an architectural *end*, compared to traditional methods that further rely on structured filters. Consequently, as a technical *means*, strict payload filters are exclusively used for non-visual parameters like brand, price and materials, as discussed previously.

4.3.3 Query Optimization

An intent-aware query rewriting step was implemented to optimize the queries for vector similarity retrieval. Different strategies for query optimization are used depending on the previously classified intent as well as whether the user specifies an item type or not. The general aim of the query optimization step is to translate a query from natural language into a query suitable for vector retrieval with stronger semantic meaning (Zhang et al., 2025; Gao et al., 2024).

For the search intent, an LLM-based query rewriting strategy is used for query optimization. As described by Zhang et al. (2025), query rewriting aims to rephrase the query to improve it for

vector retrieval, while preserving its original semantic meaning. In this implementation, the query rewriting aims to remove filler words with low semantic relevance to enhance the query for vector retrieval. The rewritten query therefore only includes words relevant for vector search, often corresponding to keywords used for analyzing the product image.

For the explore intent, multi-query-based rewriting strategies are implemented for query optimization. Regarding the multi-query strategy, an LLM-based method that combines query decomposition and query expansion, both suggested by Zhang et al. (2025), is implemented. Query decomposition aims to divide the query into multiple queries that capture different dimensions of the original query, while query expansion aims to semantically enrich the query by providing additional information (Zhang et al., 2025). For the Style Assistant, the multi-query pipeline depends on whether an item type is specified or not as extracted by the intent classifier.

If the user specified no item type is specified, category-based multi-query rewriting is implemented. This is handled by an LLM that is prompted to produce four category-based queries; two clothing specific queries, one shoe-specific query and one accessory-specific query. Category-based multi-query rewriting aims to capture different dimensions of the original search query, as stated for the query decomposition strategy by Zhang et al. (2025). For the Style Assistant, this also means ensuring diversity in the results to provide more inspirational product recommendations. Additionally to the category division, the LLM also enriches the queries semantically to translate abstract stylistic requests into a richer query with fashion-appropriate terms capturing the semantic meaning of the original query. This corresponds to query expansion described by Zhang et al. (2025).

If the user specifies an item type, the multi-query pipeline instead prompts the LLM to produce three separate search queries, where the item type is included in the queries. Again, this is based on query decomposition and query expansion suggested by Zhang et al. (2025), and aims to capture different dimensions of the request, such as providing different styles or vibes to introduce more diverse results, as well as semantically enriching the queries for vector retrieval. After the multi-query expansion, the multiple queries are merged into a twelve word long search query containing the most frequent words across all queries. This is kept in the conversation memory and used as a reference for later refinements.

For the refine intent, the query optimization steps aim to combine the previous conversation history with new refinements. This is important in order to make the Style Assistant able to use previous messages to iteratively refine the recommendations, which is one of the core functionalities of a CRS according to Freitas & Lotufo (2024). An LLM is prompted to merge the previous search query with the new refinement by analyzing which parts of the query that should be rewritten. If an item type is specified, a single query is used for the rewriting and retrieval. If no item type is specified, the same category-based multi-query rewriting strategy as previously described is used, ensuring that it incorporates the refinement into all four queries.

4.3.4 Multi-Turn Continuity

A core functionality in a CRS is handling the history of previous turns (Freitas & Lotufo, 2024). Multi-turn conversation requires the current message, prior history and potential filters to be combined. By incorporating previous messages in the RAG pipeline, multi-turn conversation is enabled (Gao et al., 2024). Multi-turn functionality needs to combine the current user message, metadata from any prior user turns and potential filters being carried. For each run, the system needs to determine continuity where the carry states of the history are either updated or cleared. The trade-off lies in keeping coherent refinements by carrying applicable filters but still ensuring that pivots do not drag old constraints or irrelevant thread text into next searches.

For the Style Assistant, conversation continuity is based on the user search context, analyzing if they move from one search context to another. Two separate continuity systems are implemented, one for turn continuity (negations and filters) and one for reference items. Keeping the reference item in a separate continuity system is necessary since this continuity is often independent of other context changes. First, continuity is determined by a heuristic function that catches clear changes. If the heuristic cannot determine this, an LLM is invoked to analyze and classify whether the search context has changed. This approach is motivated by Freitas & Lotufo's (2024) idea of only relying on LLMs when necessary.

Previous user messages are not dropped in this manner as they often imply context that is needed in future conversation turns. For example, if the user types "I want a dress in bohemian style" and during later iteration states "What about a skirt instead?", these new results should also match the previous request stating the style. As stated by Freitas & Lotufo (2024), this storage of conversation history is important in order to be able to provide continuity as well as iterative refinements in a CRS. If the next query instead is "I also need something for a halloween party", the system should determine this as a clear intent change, resulting in dropping previous messages.

4.3.5 Negations

One functionality of the Style Assistant is how it handles negations, satisfying the system *end* which requires that users are able to address negative constraints in natural language. In other words, these are negative constraints where the user explicitly states what they do not want (An et al., 2025), for example, user inputs along the lines of "Nothing that is made out of wool" or "Can you find me something that isn't floral". The negation term is extracted from the raw user message. First, a simple heuristic function is used to quickly identify any explicitly stated negations. If the heuristic cannot extract any negation terms, an LLM is prompted to analyze the user input and extract potential negations. This two-stage approach is chosen in order to only invoke the LLM when necessary, as suggested by Freitas & Lotufo (2024).

Identified negation terms are then embedded separately. The negation vectors are used as *means* to adjust the retrieval vector by pushing it away from these unwanted concepts. Mathematically,

the input vector and negation vectors are L2-normalized. Then, alignment between these two is computed by measuring the dot product (Wang et al., 2024). A positive dot product indicates that the vector still points toward the negation term. In this case, a fraction of the aligned part of the vector is removed, scaled with a default push-away strength. After each removal the unit length is renormalized. In short, the system iteratively projects the query onto each forbidden direction, subtracts a scaled version of this projection and renormalizes. Since dense retrieval computes similarity between the query vector and the product vectors (Zhang et al., 2025; Gao et al., 2024), the negation term vector aims to reduce similarity to any products semantically close to the negation term.

4.3.6 Reference Item

A reference item functionality is implemented in order to make it possible to do smaller refinements using another item as a reference. Referencing an item is done by selecting it using the pin button in the UI and stating alterations in the input message. This means for example referencing an item and writing something like “I like the fit of this but I want it in black”, “I like the style of this” or “I want it to be longer”. As stated by Deldjoo et al. (2025), it is often difficult to express stylistic concepts using words, and referencing an image can therefore be a good way to narrow down user preferences. The reference item functionality therefore aims to put the referenced item as a visual anchor for retrieval. This is enabled by multimodal embeddings, where a blended vector is created by weighting the referenced image vector with the textual refinement vector in a common vector space (Deldjoo et al., 2025; Rubio et al., 2017).

Regarding weighting, there is no objectively correct weighting between image and text. After experimentation, refinements are weighted heavier than the reference image in order to keep alterations heavier than the original image. This prevents the results from feeling like the visual nearest neighbors, but instead clearly takes the alteration into consideration. In the case of negative constraints in combination with a reference item, the same image and text weight is used, however the negation term is applied through a further vector push-away on the blended query, similarly to the normal case but with a higher strength for exclusions to remain effective.

4.4 Retrieval

The retrieval step retrieves catalog items by semantic similarity to the user’s rewritten query. Optional payload filters from the pre-retrieval steps restrict which products enter the candidate pool. The Style Assistant uses dense retrieval, meaning the optimized query is encoded into a high-dimensional vector representation used for similarity matching (Wang et al., 2024) against the Qdrant vector database. Specifically, the query is encoded using the same multimodal embedding model as the knowledge base and is padded to a 2241-dimensional vector to align with the product vectors. This vector is then used for an approximate nearest neighbor search within Qdrant, using dot product as the similarity metric. Payload filters ensure that only

products satisfying the extracted constraints and available on site are returned. In short, retrieval ranks the catalog by semantic proximity to the query (Shu & Yu, 2025).

The system executes either single-query or multi-query retrieval depending on the routing pipeline. During multi-query retrieval, several queries are run in parallel as separate retrieval legs. Category-based retrieval, performed when no item type is specified, uses Qdrant payload filtering to apply clothing, shoes or accessory filters to each retrieval leg. Each leg returns its own list of items which are then merged and deduplicated, keeping the highest scoring items when present in multiple legs. To further diversify the retrieved items in cases where no item type is specified, category limits are applied in the Qdrant retrieval. Firstly, each item type is limited to a maximum of three items. Secondly, certain item types such as underwear are excluded unless they are stated by the user as they were otherwise often overrepresented. Lastly, tops were often shown to be overrepresented. In order to ensure that the recommended items include at least one bottom, an additional bottom retrieval is implemented in cases where no bottoms are retrieved originally. Then, the lowest scoring clothing item is replaced by the highest scoring bottom.

In order to improve the recommendations, a larger pool of products than what is shown is often retrieved. Also, Qdrant's nearest neighbor algorithm uses Maximal Marginal Relevance (MMR), which trades a small amount of top similarity for other more varied neighbors. This is meant to balance the exploitation of strong matches with the exploration of alternative options (McInerney et al., 2018; Yang et al., 2023).

Together, MMR, per-type caps, category mixing in multi-queries, and bottom enrichment together represent a computational *means*. These methods in combination are designed to achieve the *end* of reshaping the raw Qdrant ranking, leading to recommendation diversity without changing the semantic query (Jannach, 2022; Zhang et al., 2025).

4.5 Post-Retrieval Processing & Augmentation

Post-retrieval covers all steps that turn raw vector neighbors into a presentation-ready set of recommended items (Gao et al., 2024). The list of results is capped to 15 product cards to match the user interface, regardless of whether retrieval has considered more candidates internally.

Results from multi-query integration are then re-sorted by similarity score (Winterflood, 2026). For category-based multi-query retrieval, reserved slots for shoes and accessories help ensure a mixed list of results rather than strict global top scores of clothing items. By shuffling the merged list, the results are not strictly ordered by retrieval leg.

If no products within the catalog match the query, the system returns a preset message explaining that no results were found. This response is rendered in the user interface without any other results. Avoiding LLM generation in cases without results prevents hallucinated answers from

the LLM (Nawara & Kashef, 2025), which could otherwise result in the system recommending incorrect items or giving false acknowledgments. However, in the standard case of valid product matches, the retrieved items and their metadata are forwarded to the generation stage.

4.6 Generation

The final phase of the multimodal RAG pipeline is the generation stage, which translates the retrieved data into a response in natural language to be displayed for the user. While previous steps have focused on the actual recommendations, the generation step transforms the information into a cohesive response to present in the dialogue with the user. In the case of at least one item recommendation, the system initializes the generation step. The response consists of a short acknowledgment text, the actual recommendations and three following example prompts.

To create these, the model receives relevant context regarding the conversation from the post-retrieval phase in order to ground the written results. Grounding the textual outputs in context aims to maintain conversational relevance and continuity (Freitas & Lotufo, 2024). The acknowledgment is intended to reflect how the request is interpreted and how the shown results relate to it, aiming at grounding in communication to support trust (Mamun et al., 2025). The system is prompted to avoid false certainty, not to state prices and to respect referenced items, in order to limit hallucination, a known risk with large language models (Nawara & Kashef, 2025). Moreover, the three example prompts are meant as small inspirational search queries grounded in the same conversation, which aim to complement the input field, helping the users to understand what kind of requests the assistant can handle and what level of details that might be useful. The system instructs the LLM to propose one narrow refinement of the current query, one complementary constraint on the same goal, and lastly one that opens a new dimension of the overall style. This is structured with the purpose of providing both exploiting and exploring recommendations (McInerney et al., 2018; Yang et al., 2023). The main goal is to keep the dialogue helpful to avoid user abandonment (Gao et al., 2021).

4.7 User Interface

The user interface of the Style Assistant aims to complete the system, however, it was not a primary focus of the thesis objectives and should therefore be interpreted as infrastructure for system evaluation rather than a final product ready for production deployment. The development of the Style Assistant user interface aims to match existing Sellpy design by adopting colors and fonts, as well as some internally pre-made components used throughout the existing website. In this section, the UI is explained in the order in which it is used.

The Style Assistant is exposed as a dedicated route in the Sellpy frontend. From a user perspective, navigation is integrated into the main header through a specific “AI search” icon, see Figure 2.

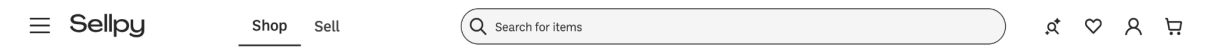

Figure 2. Navigation button to Style Assistant from Sellpy banner, accessible via the icon directly to the right of the main search input field represented by a magnifying glass with sparkles.

The landing page of the Style Assistant, as seen in Figure 3, consists of a title, “Sellpy Style Assistant”, an illustration and three randomly rendered example prompts sampled from a larger predefined pool. These are displayed as large clickable example prompts designed to lower the initial cognitive effort required to start using the Style Assistant. These aim to improve the natural and intuitive feel of the interaction, hopefully leading the user to continue using the interface (Mamun et al., 2025).

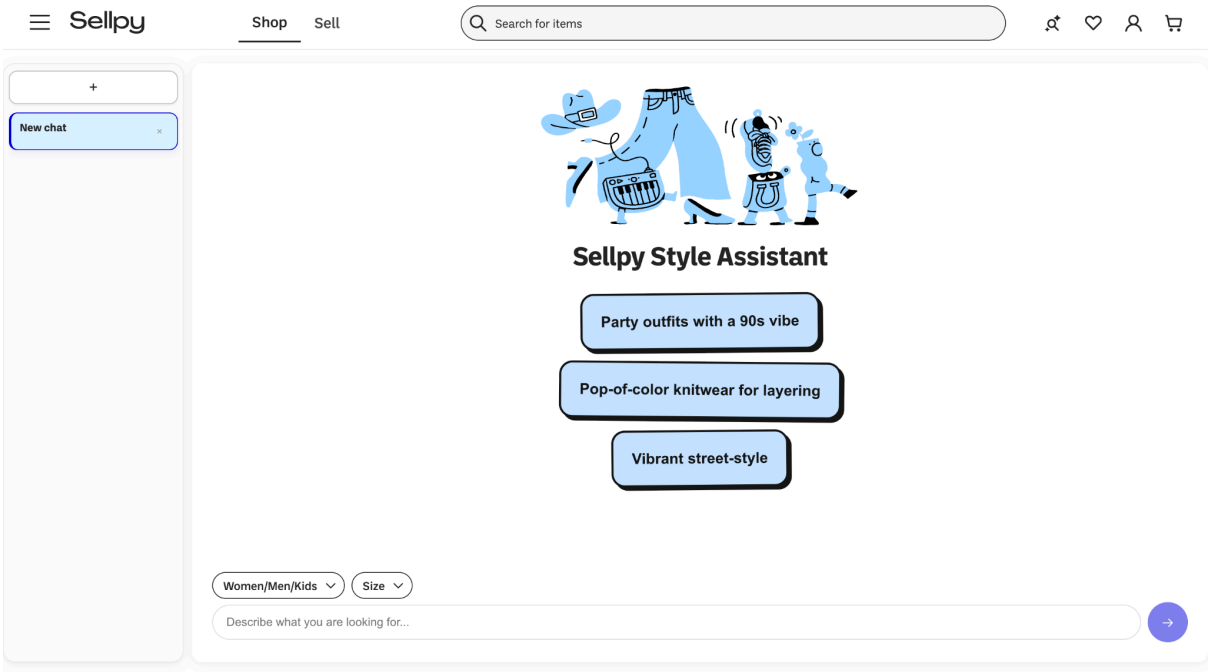

Figure 3. Style Assistant landing page.

The Style Assistant uses tabs to support multiple parallel conversations, each containing the current message history, active filters, pinned reference item and other metadata such as the rendered title. Tab titles are initialized with a default label, “New chat”, which are later changed based on the search content, aiming to improve the overview and retrieval of earlier sessions, see Figure 4. The purpose of the tab system is to enable users to explore different style intents without losing previous context and therefore also revisit earlier searches by returning to the same context where they left off. As previously mentioned, a key functionality of conversational recommender systems is to reuse previous interactions and conversation history, therefore this needs to be preserved (Freitas & Lotufo, 2024).

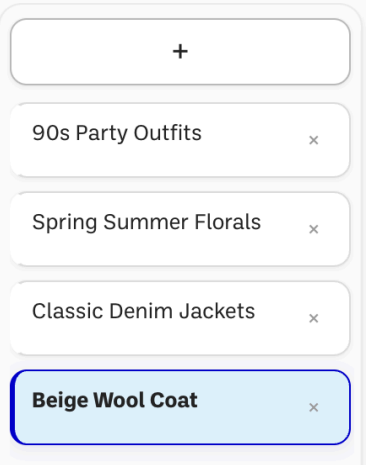

Figure 4: Session tabs.

Search queries can be formatted however the user wants to, however, empty searches are not submittable and are avoided as the button to send the request remains disabled until the user has typed their search query. Before sending their request, the user can apply demography and size filters through buttons as seen in Figure 5. These are handled in the same way as in other parts of the website including the user's predefined size profile if they have created one and logged in to their account. For the demography and size filters to be active, they need to be set before the message has been sent. However, they can be kept as unset if the user does not have such preferences. Thereafter the filters are kept as a preference in the conversation context and kept for later retrieval as long as the user does not clear or change them in later iterations.

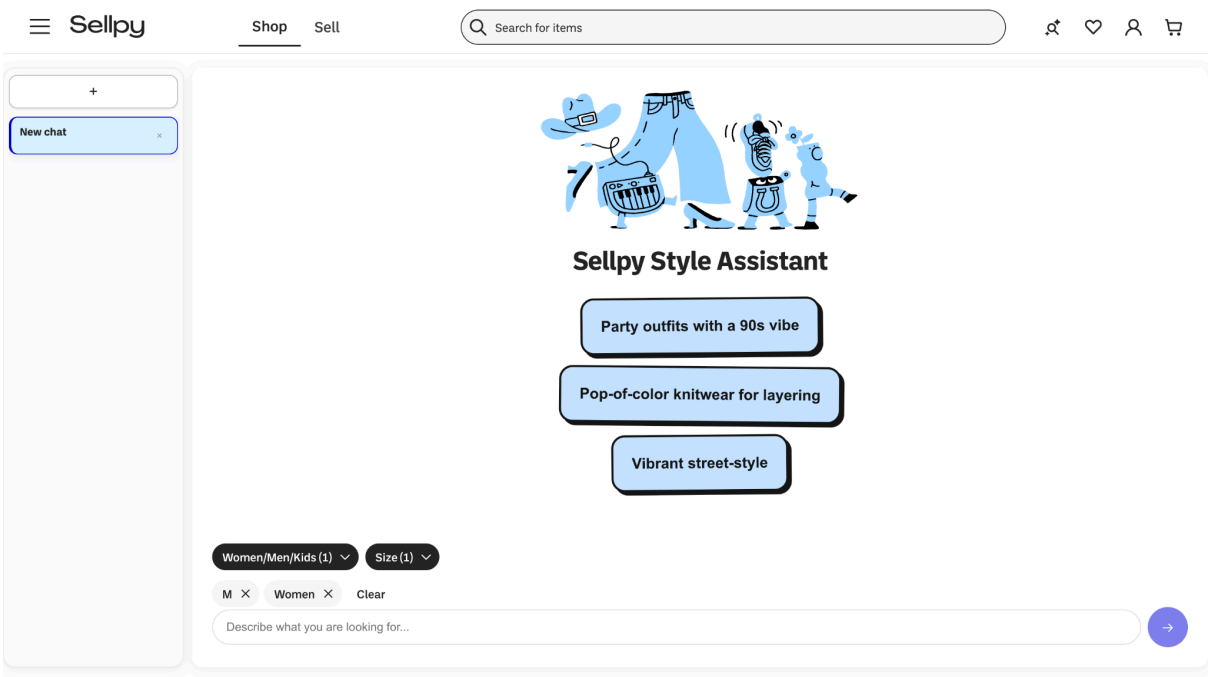

Figure 5: Landing page with active filters.

After a user sends a request, the fixed text “Assistant is thinking…” is immediately shown as a response while the results are loading, see Figure 6. In this state, no new messages can be sent, and the example prompts are disabled. To illustrate that the request is being processed, the user is shown an animation representing the loading of the system as well as the three toggling dots. This is a critical moment of the pipeline as interactions usually end if the user becomes too impatient to continue the search (Mamun et al., 2025; Gao et al., 2021), therefore effects like these aim to engage the user and convey the understanding that the results are on its way.

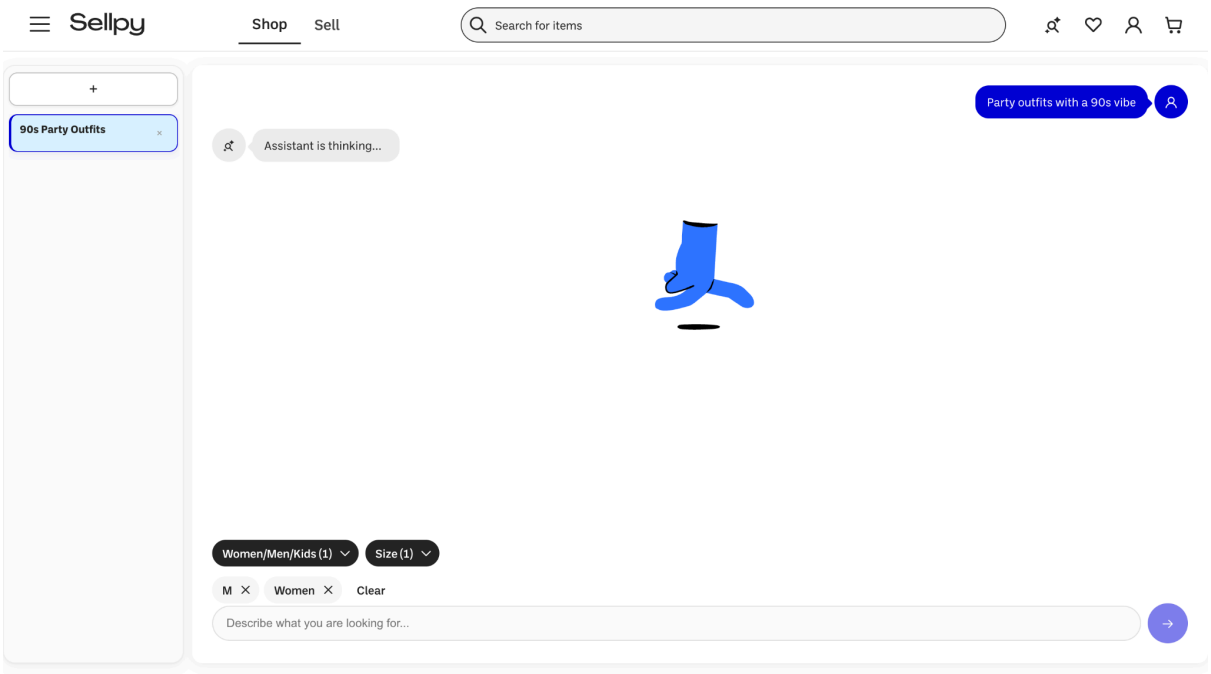

Figure 6: Loading page after sent query.

When the backend reply arrives, the loading state is cleared and the UI renders the response and recommendations, as shown in Figure 7. The response has three elements: an generated acknowledgment message, the recommended items and three follow-up example prompts. First, the assistant acknowledgment is shown as the assistant turn of the conversation and is revealed progressively with a typewriter effect for conversational pacing. User engagement is crucial for the interaction quality, therefore, effects like these aim to improve the design of the system (Lopez-Lopez & Iniesta, 2025), especially in scenarios like these where the user might have had to wait for a period of time. The second part of the response is the actual recommendations in the form of product cards. Lastly, follow-up example prompts are displayed underneath the actual results to support iterative refinement.

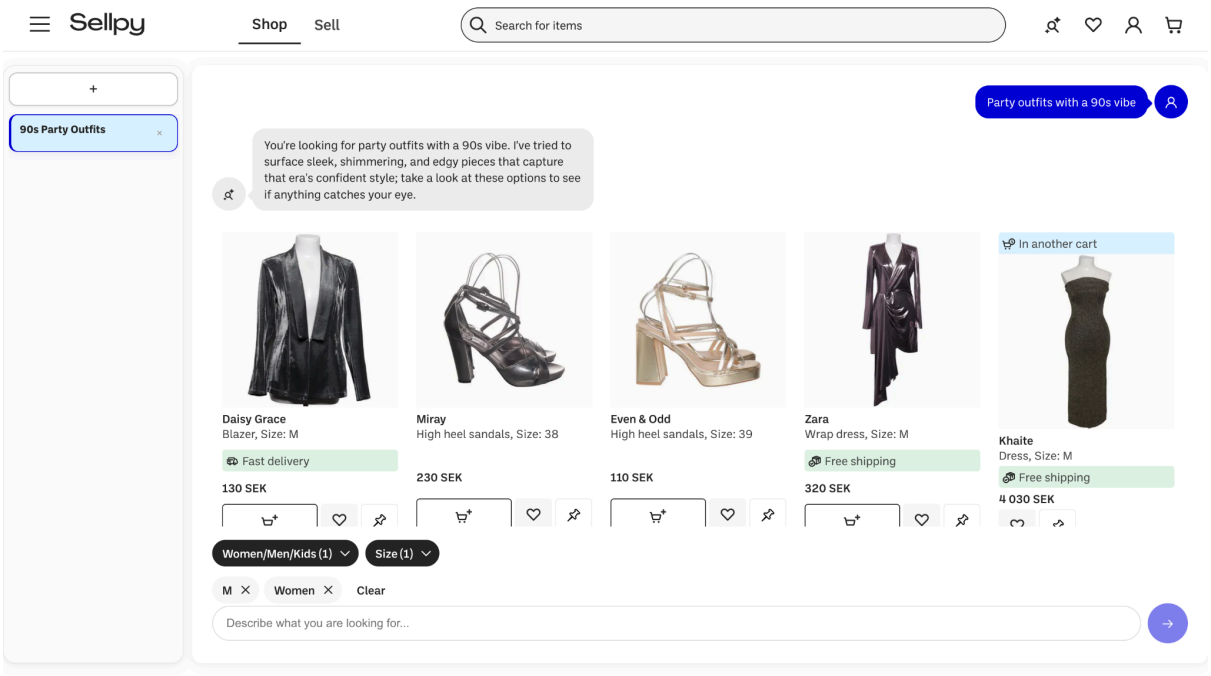

Figure 7: Answer and recommendations results.

Unique for the Style Assistant is the additional third button for pinning an item, as seen in Figure 8, allowing the user to pin an item as a reference for the next iteration. The UI of this button is directly inspired by the other buttons and is either empty or filled depending on whether the item is pinned. As a result of pinning an item from the list of results, it gets attached to the search bar to make the current reference context explicit before sending the message, see Figure 8.

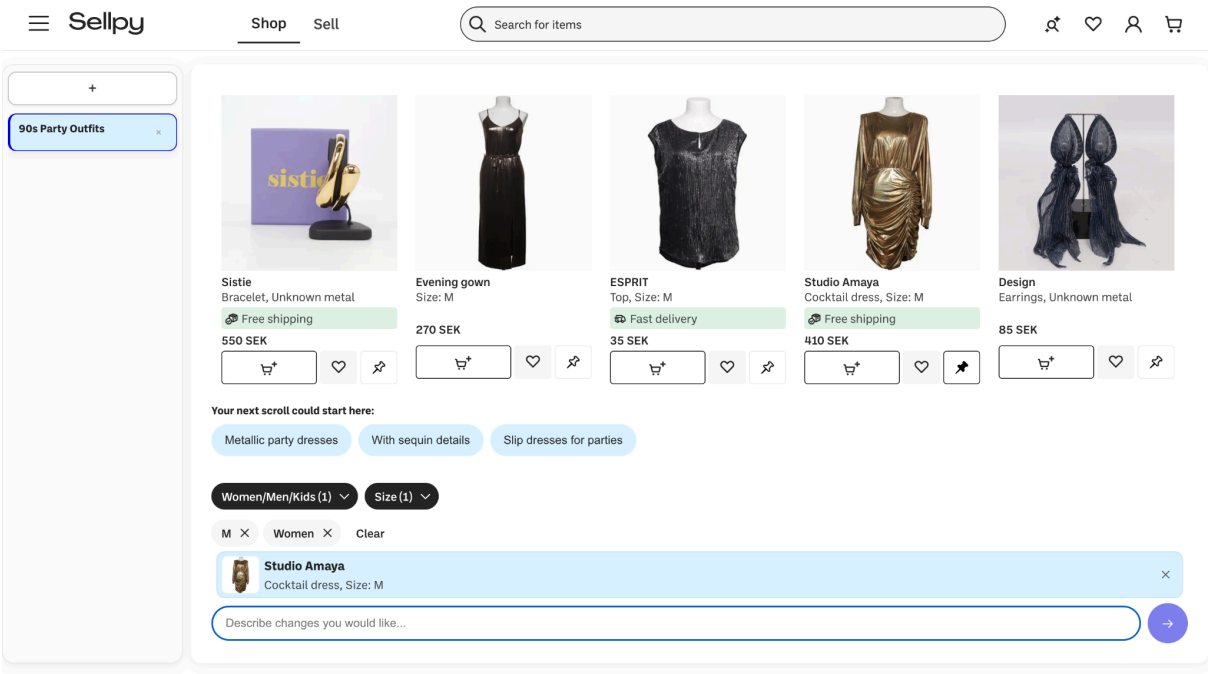

Figure 8: Pinned item.

When a pinned item is used in a sent message it is displayed together with the corresponding textual input in the chat timeline as shown in Figure 9. Then, if the user scrolls back in the conversation history they can see the link between the written message and pinned reference item, see Figure 9. This design aims to improve interface traceability by explicitly showing which item was referenced that specific turn. This explicit connection makes it easier for the user to understand why the specific recommendations were presented, in contrast to the major limitation of LLMs operating as “black-box” models (Milano et al., 2020). By pinning the referenced item in the interface, the system establishes a form of “grounding in communication” (Mamun et al., 2025), transforming a potentially untraceable process into a transparent interaction that demonstrates comprehension to enhance user trust.

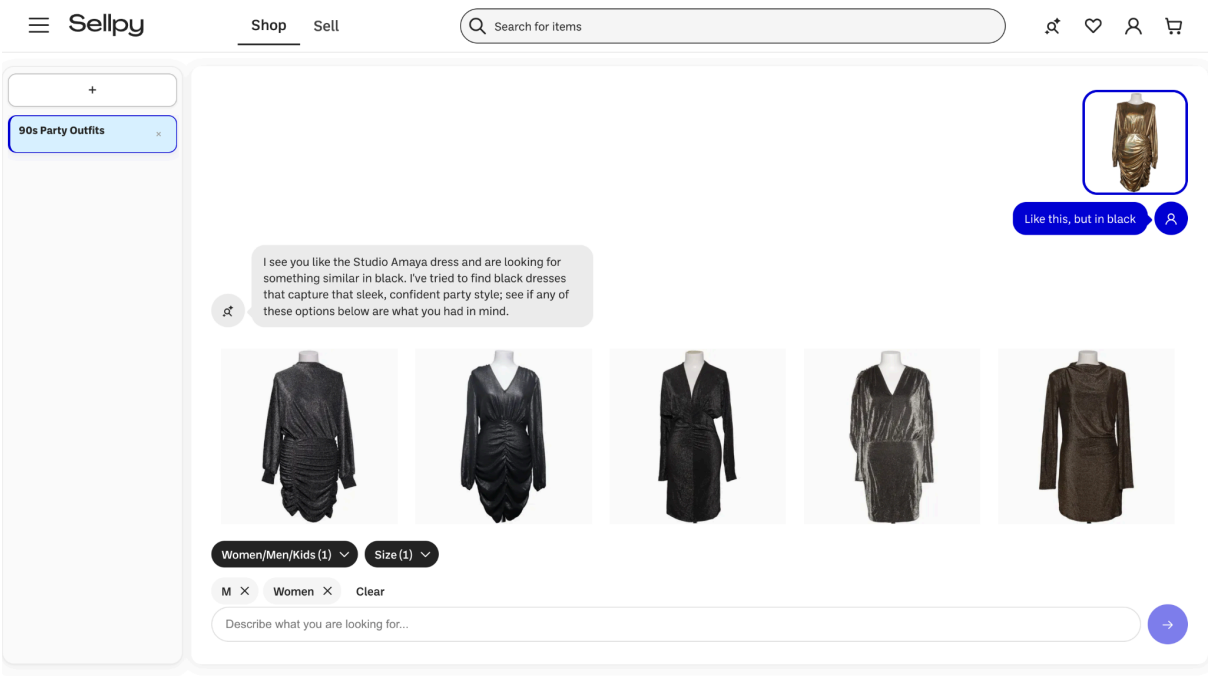

Figure 9: Results after sent query with pinned item and alteration.

As previously mentioned, the example prompts are meant to inspire the user to continue using the system by suggesting meaningful next actions. Before the follow-up prompts the interface displays a short guidance heading such as “Not quite it? Try one of these searches:” or “Adjust or broaden what you're looking for:” etc, as illustrated in Figure 10. This heading is randomly selected from a fixed predefined list of equivalent variants. This aims to keep the conversational tone varied while preserving consistent meaning. When hovered, the example prompts become highlighted and, when pressed, they are immediately sent as an user message.

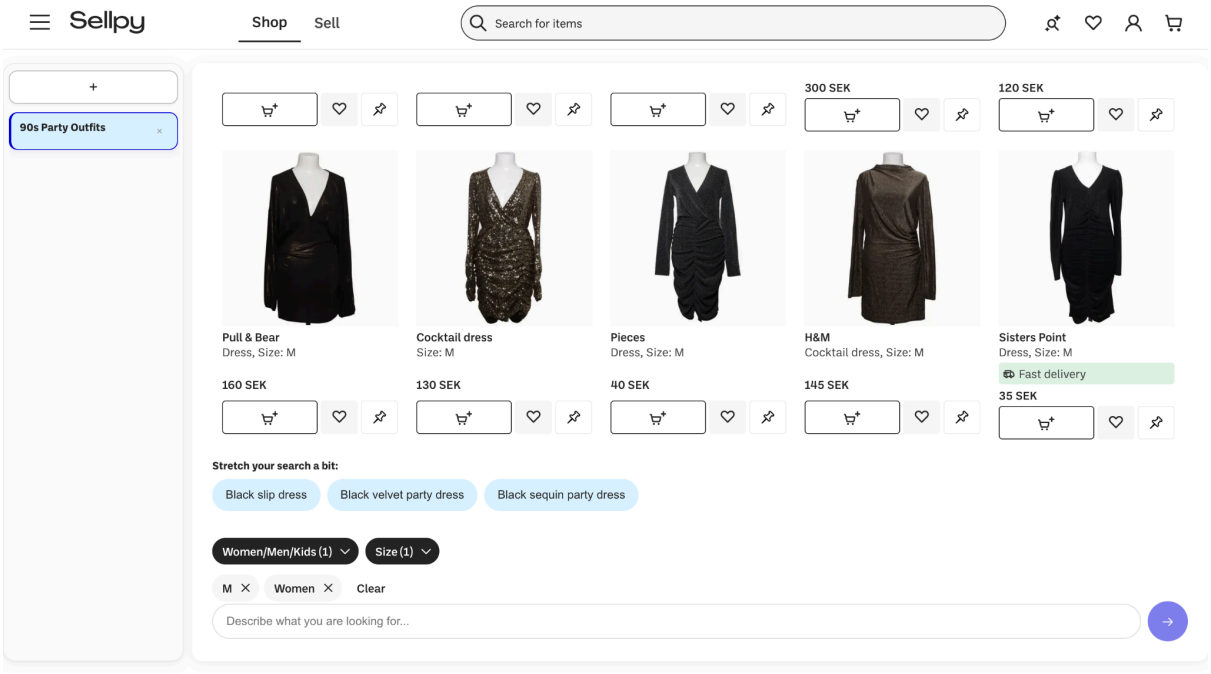

Figure 10: Example prompts shown with the results.

While the user interface was not the primary focus of the design cycle, as previously mentioned, it serves as an essential interface enabling users to interact with the MM-RAG-based CRS, which is necessary for evaluating the Style Assistant as an artifact.

5. Results

This chapter presents the findings from the evaluation of the Style Assistant, highlighting both a technical and UX perspective to evaluate the quality of the artifact.

5.1 RAGAs Evaluation Results

The results showed varying performance across the three different RAGAs evaluation metrics used; context relevance, answer faithfulness and answer relevancy. Context relevance showed a moderate performance but high variability, with a mean of 0.583 and a standard deviation of 0.380, see Table 1, indicating varying performance for different retrieval queries. Answer faithfulness generally achieved very low scores, with a mean of 0.233, see Table 1, indicating that the generated message was not supported by the retrieved context. This could be explained by the limited information in the product metadata, which is further discussed in later sections. Answer relevance achieved the highest score, with a mean of 0.710, see Table 1, indicating that the generated message corresponded well to the user input queries.

Table 1: Average RAGAs scores for all scenarios.

Metric	Mean	Std	Median	Min	Max
Context relevance	0.583	0.380	0.646	0.0	1.0
Answer faithfulness	0.233	0.264	0.167	0.0	1.0
Answer relevancy	0.710	0.156	0.711	0.0	0.937

To investigate the RAGAs performance across different groups of search queries, group-based statistics for the RAGAs performance metrics were computed. Context relevance varied a lot between different query groups. More specific queries, belonging to groups like specific search, filters, reference and refinement, generally showed high performance scores while exploratory queries yielded very low scores, see Table 2. For the answer faithfulness metric, the scores were generally low for all scenario groups, but queries involving filters and refinements performed slightly better in comparison to other groups, while exploration and negation scenarios received very low scores, see Table 2. The answer relevance score was rather high for all query groups.

Table 2: RAGAs performance metrics per scenario group.

Scenario Group	n	Context relevance	Answer Faithfulness	Answer relevancy
Explore	5	0.0336	0.0836	0.864
Specific search	6	0.653	0.117	0.734
Filters	5	0.791	0.399	0.691
UI filters	3	0.674	0.383	0.638
Negation	4	0.332	0.0357	0.771
Reference	10	0.747	0.233	0.641
Refinement	6	0.643	0.357	0.711

5.2 User Experience Evaluation Results

The UX evaluation resulted in user perceptions about the Style Assistant, collected performance metrics and Likert scale statistics from the questionnaire. In this section, these three types of results from the UX evaluation are presented.

5.2.1 Qualitative Findings

Users generally had a positive attitude towards the Style Assistant as a system for product discovery. Some users explicitly stated that the curated selection of 15 results was favorable, as this limited navigation through the large catalog. The conversational flow and the potential for refining the results, for example with a pinned reference item, as well as the familiar LLM chat format, were appreciated by many users. Generally, users liked the Style Assistants ability to understand styles and vibes better than a regular keyword-based search.

One recurring issue during the UX evaluation was that users did not apply their sizes accurately, which caused them to get results in the wrong demography and size. However, for users who had complete personal size profiles, this was not a problem. Other frustrations included the system's response time, as well as the fact that it sometimes recommended items that did not fulfill explicitly stated requirements, for example for certain item types or colors. Several users also stated that they would like the possibility to pin multiple items in order to convey the general feel of what they were looking for.

5.2.2 System Performance Metrics

During the UX part of the evaluation, system performance data was collected to evaluate the system performance and user behavior. This data was later used to identify the average response time of the system, the average number of iterations per session and the fraction of users that successfully found something they liked during this part of the evaluation, see Table 3.

Table 3: Measured system performance and user behavior collected from the UX evaluation.

Sessions	Average response time (s)	Average total iterations (n)	Fraction success
Total	18.4	4.4	0.77
Search	17.0	4.2	0.78
Explore	19.9	4.4	0.76

5.2.3 Questionnaire

Regarding basic success metrics of the Style Assistant evaluation, all users stated that they found an item that they liked and an item they would consider buying, see Table 4. The following section examines different aspects of the Style Assistant's performance from the UX evaluation.

Table 4: Statistics of direct success/failures for the usability evaluation.

Statement	Yes	n
I found something that I liked	100 %	9
I found something that I would consider buying	100 %	9

Users stated that the initial results felt somewhat relevant, but there is a rather high variance, see Table 5. The responses collected through the questionnaire use the seven point Likert scale as described in Section 5.2.3. Some users experienced that the results felt random or did not fulfill their stated requirements in terms of, for example, brand or color, which might correspond to a lower score. At the same time, several users explained that they really felt like the results were relevant and immediately found items that they were interested in. Looking at the median score, the users generally found the system helpful and would use it again, see Table 5.

Table 5: Statistics for general effectiveness of task support statements.

Statement	Median	Range (Min-Max)
The initial results felt relevant	5	2-7
I found the system helpful to find suitable items	6	4-7
I would use this system again	6	5-7

For the search intent, users were somewhat content with the performance of the Style Assistant, although as seen in Table 6, there was a rather high variance in the scores that the users assigned. Some users got very relevant products on their first try, feeling like the recommendations were exactly what they were thinking of. On the other hand, some users struggled with getting the Style Assistant to understand their specific requirements in terms of, for example, brand or colors. In some cases, the system also did not fulfill requirements in terms of, for example, only recommending products of a specific type.

Table 6: Statistics for effectiveness of task support statements for search intent.

Statement	Median	Range (Min-Max)
The results fulfilled my request of specific requirements	5	2-7
I felt like the system understood what I was searching for	5	3-7

Users generally felt like the results were somewhat suitable to their request in terms of style or occasion, see Table 7. Several users expressed that they liked that the system provided recommendations for an outfit, comprising clothing, shoes and accessories. Also, they expressed that the system captured the vibe or aesthetic they were looking for very well, especially compared to the normal search function. Some users instead experienced that the results felt a bit more random, and while they could be a good fit for the sent message, they were not what the users were looking for. Some users experienced that the recommended products sometimes felt out of place for their request.

Table 7: Statistics for effectiveness of task support statements for explore intent.

Statement	Median	Range (Min-Max)
The results felt suitable to my request in terms of style/occasion	5	3-7
The system inspired me to explore new items that I would not search for myself	6	1-7

Most users successfully refined their results using natural language, which is reflected in the median score 6, see Table 8. Furthermore, most users found that refining the results helped them find what they were looking for, see Table 8. Refining in terms of prices, colors, fits or other attributes generally worked well. At the same time, some users experienced problems with refining, where they tried to adjust the products several times without success, which could be

explained by the wide range of scores, see Table 8. For example, some users tried to refine the color palette several times with no success.

Table 8: Statistics for effectiveness of task support statements for refinements.

Statement	Median	Range (Min-Max)
The system understood my refinements	6	2-6
Refining the results helped me find what I was looking for	5	2-7

The reference item statements generally received high scores with a median of 6 and a relatively low variance, see Table 9. This shows that users generally appreciated the reference item functionality. Several users expressed that this was the best and most helpful functionality of the Style Assistant for guiding the recommendations in a desired direction, especially since the user did not have to express what they wanted in words, but could instead use the visual attributes as a reference. Some users found that the reference item focused a bit too much on the image, forgetting previous requirements in the conversation.

Several users used the reference item functionality to find matching products, which was not an implemented functionality, and therefore resulted in failure. Also, several users expressed that they wanted to pin multiple items to guide the Style Assistant in the direction of the style that they liked.

Table 9: Statistics for effectiveness of task support statements for reference item refinements.

Statement	Median	Range (Min-Max)
The system understood what I wanted to refine with help of the reference item	6	5-6
Refining the results with a reference item helped me find what I was looking for	6	4-7

Regarding how much effort was needed to find products that the user likes, the results were rather varied with a median of 3, see Table 10. Several users expressed that they liked that the Style Assistant showed them a smaller selection of products, lowering the effort of finding relevant products compared to browsing the full product catalog, which could feel overwhelming. At the same time, some users gave up before finding relevant products, see Table 10, indicating that the effort was too high. Some users iterated many times before finding products that they liked, while others found something almost immediately.

Table 10: Statistics for efficiency of task support statements.

Statement	Median	Range (Min-Max)
Much effort was needed to find something that I like	3	1-6

The median scores regarding the subtasks of the system showed that most users found them to work well, see Table 11. When asked if the system understood their intent, the users scored it in a range from 2 to 7, where a few users did not perceive the results to truly match what they meant by their search. Most of the users found the results to follow their applied filters, as seen in Table 11, although a quite common issue was that the users missed to apply their demography and size with the buttons, or instead only wrote them in the message. Not filtering correctly sometimes resulted in the results not being relevant to the user.

In general, users somewhat agreed that the system understood negations, with a median score of 5, see Table 11. Most users found the predefined example prompts to be relevant, see Table 11, whereas some explicitly used them while some others completely skipped them.

Table 11: Statistics for effectiveness of subtask statements.

Statement	Median	Range (Min-Max)
The system understood my intent	5	2-7
The results followed the applied filters	6	3-7
The system understood negations	5	3-7
The predefined example prompts felt relevant	5	4-7

Overall, the users were satisfied with the system, except for the slow response time, see Table 12. The response time was a recurring frustration among the users, with a very low median score regarding the response time being fast enough, see Table 12. Across all sessions, the average response time was 18.4 seconds, as seen in Table 12. This was a point commonly brought up during the open questions when asked what generally did not work well with the system.

Generally, the dialogue was thought of as consistent throughout the conversations and mostly remembered the user's preferences in later iterations, according to the users, see Table 12. For some of the users, while exploring different categories, the system seemed to drop the initial preferences requesting a certain style or context when iterating further or changing to search for another item type, resulting in the user having to remind the Style Assistant of these previous requirements. However, for some other users, this was not an issue and in those cases the system remembered the conversation history well.

When asked if the language felt natural and easy to understand, the users responded that this applied to the system with a median of score 6, so the general opinion was that this was the case, see Table 12. However, a few users stated that they did not really read the acknowledgments that the Style Assistant provided. Some users pointed out a few uncommon words from the example prompts such as “moisture-wicking’, as well as the Style Assistant claiming the recommendations to be a certain incorrect item type.

Most of the users stated that they trusted the system's recommendations, see Table 12, although a point made by one user was that sometimes the recommendations felt a bit random in the early iterations as only a little information had been provided regarding what the user was looking for, and therefore the user did not completely trust those recommendations.

Table 12: Statistics for satisfaction statements.

Statement	Median	Range (Min-Max)
The system responded fast enough	2	1-5
The dialogue felt consistent throughout the session	5	4-7
The system remembered my preferences throughout the conversation	5	2-6
The language was natural and easy to understand	6	3-7
I trusted the recommendations from the system	5	3-6

6. Discussion

The aim of this thesis was to address product discovery challenges in second-hand e-commerce. A Design Science Research (DSR) (Hevner et al., 2004) approach was used, where the Style Assistant artifact was designed, developed and evaluated in relation to the identified objectives. Therefore, this chapter evaluates the Style Assistant's ability to address product discovery challenges. By combining the problem analysis (O1), the system design (O2) and the mixed-method evaluation results (O3 and O4), the discussion reflects on the extent to which this MM-RAG-based CRS addresses product discovery in the unique domain of second-hand fashion e-commerce. The section concludes with a reflection on the limitations that point toward future research within the field.

6.1 Problem Relevance: Second-Hand E-Commerce

Objective 1*: Analyse user needs and product discovery challenges in the context of second-hand fashion e-commerce to identify the design problem,* which tightly corresponds to the relevance cycle as presented by Hevner (2007). By identifying the environmental complexities of product discovery during the relevance cycle, explained by Hevner (2007) as analyzing opportunities and problems in the research environment, the requirements were iteratively mapped against grounding technical *means* inside the design cycle to continuously refine the artifact. By bridging constraints of the relevance cycle with the iterative execution of the design cycle, the true design problem could be addressed.

During this process certain *ends* were established which the artifact must satisfy, as Hevner (2007) explains based on identified goals, requirements and constraints, which in the context of the Style Assistant laid the foundation for later cycles during the development. As the environment of second-hand e-commerce is especially unique and complex, standard generic assumptions were not applicable, creating the need for a further conducted problem analysis. It is worth mentioning that this cycle would need further iterations as presented by Hevner (2007) to solve the organizational challenges through iteratively balancing the identified *ends* and *means*.

The problem analysis highlights the fundamental limitations within the second-hand e-commerce domain with unique product discovery challenges. This environment is characterized by a "discovery bottleneck" (Shen et al., 2012) due to its large amount of unique items as well as a lack of available data on historical interactions, in contrast to traditional e-commerce environments. Contextual findings from the situation assessment showed that this context led to specific challenges that require solutions that extend traditional recommender systems, directly aligning with related literature (Khatwani & Chandak, 2016). This highlights the true challenge based on data sparsity rather than the large catalog volume. As products are unique and short-lived, traditional collaborative filtering lacks the historical behavioral anchor required to create recommendations in such a manner.

A key insight from the situation assessment is the identification of two main subproblems with traditional discovery functionality: specific search and exploration. Traditional search and filtering methods struggle to handle subjective or semantic requests due to their hard reliance on structured metadata (Ye et al., 2023). This lays the foundation for a new artifact being capable of handling users' wishes to use broader fashion terms, such as matching items to specific occasions or styles, as seen in the user survey on expectations. These findings validated the gap identified by Laenen et al. (2018) and Deldjoo et al. (2025) regarding the fact that fashion intent is highly subjective to personal vibes or preferences. This created the discovery bottleneck where items with certain visual attributes remain unsearchable to the user. In practice, these subproblems mean that structured metadata schemas act as a functional filter completely ignoring visual styles. As fluid concepts like vibes or occasions are not directly mapped as item attributes, this forces users to translate their requests into mappable terms for rigid catalog schemas.

This technical bottleneck directly links to a broader societal problem where traditional search and filtering systems restrict sustainable consumption by making large scaled second-hand product catalogs frustrating to navigate. As presented by Guiot & Roux (2010) the general public has an ethical, ecological and economical desire to actively distance themselves from traditional fast-fashion. However, a barrier between consumers and second-hand items requires a new solution to simplify sustainable customer choices.

To address the identified gap between the limitations of traditional systems and the subjective nature of second-hand fashion e-commerce, the Style Assistant and its identified requirements were positioned as a relevant technological solution. The user expectations required the system to support specific and explorative searches through natural language, handle iterative user feedback and enhance item relevance by combining images with textual product information during retrieval. This aimed to extend traditional recommender systems which, according to Khatwani & Chandak (2016), are ill-suited for environments with data sparsity. The identified *means* and *ends* of this initial phase demonstrated that the fashion context created a need for a context-aware solution that can handle the dynamic nature of user intents and semantic searches, as explained in the literature (Nawara & Kashef, 2025; Ye et al., 2023). This laid the foundation of the design cycle, established by Hevner (2007) and directly connected to the next objective of this thesis, where the Style Assistant was designed and built based on this problem analysis.

6.2 The Artifact: MM-RAG-based CRS

Objective 2: *Design and implement a RAG-based CRS including retrieval strategy, conversational interaction flow and recommendation logic* focused on the creation of the Style Assistant artifact. The Style Assistant artifact serves as a rigorous technical response to the identified design problem regarding product discovery in second-hand e-commerce from the relevance cycle. Furthermore, this section discusses the design and development of the artifact in

relation to the design cycle, as well as the incorporation of previous CRS and RAG research corresponding to the rigor cycle (Hevner, 2007).

This thesis addressed the product discovery bottleneck by designing and developing the Style Assistant, a MM-RAG-based CRS. As stated in previous literature, CRS enables more dynamic product recommendations through natural language understanding and iterative refinements (Nawara & Kashef, 2025; Freitas & Lotufo, 2024) and therefore serves as a promising solution to the product discovery challenge. The relevance cycle showed that users expect the Style Assistant to understand natural language beyond keyword matching and filters as well as to be able to give semantically advanced recommendations. Multimodal RAG was identified in the rigor cycle as a suitable technical *means* for enabling the matching of textual queries to visual attributes, as well as grounding recommendations in the product catalog, contributing to rigor as RAG is an established method for this kind of system.

The ability to project images and text into the same vector space (Zhang et al., 2025) was the main reason why MM-RAG was considered a suitable *means* for the stated design problem. For the Style Assistant, this enables users to use natural language to search for visual attributes. The incorporation of visual features means that users can both search for details that are otherwise not incorporated in filters or product metadata, and use broader stylistic terms such as styles or vibes to describe what they are looking for. Additionally, MM-RAG ensures that generated messages and recommendations are grounded in the actual product inventory, which is crucial for the Style Assistant in order to guarantee accuracy and trustworthiness.

An advanced RAG pipeline (Gao et al., 2024) was implemented, consisting of pre-retrieval steps, retrieval, post-retrieval steps and generation. To adapt the architecture to the domain of a CRS for second-hand fashion recommendations, certain steps were included in the pipeline. The fashion domain is both characterized by the importance of visuals, as well as the fuzzy language used for expressing stylistic preferences (Deldjoo et al., 2025). For the preprocessing steps, intent classification, filter extraction and query rewriting was implemented aiming to capture what the user actually means by their natural language request, and enhancing it into a query suitable for vector search. This enables more semantically advanced recommendations, where users can discover desired products without having to rely on filters or specific keywords. Separating the pipeline based on user intent also aims to improve the Style Assistant's ability to recommend products based on different types of user input. Being able to both handle specific requirements and broader inspirational searches was a requirement based on the user expectations identified in the relevance cycle. At the same time, filter extraction ensures that the retrieved product recommendations follow explicit requirements, which is important in order for the recommendations to actually be relevant to the user.

Regarding conversation history, the Style Assistant handles iterative refinements, which is expressed as one of the main functionalities in a CRS. For textual refinements, this is done using

an LLM. A reference product feature was implemented as an additional functionality for refining the recommendations. This allows users to directly interact with visual features by combining textual refinements with a product image, simplifying the process of narrowing down the recommendations. This aligns with Deldjoo et al.'s (2025) idea of multimodal embeddings enabling users to express style preferences in a simpler way.

The Style Assistant's response consists of a generated message, 15 product recommendations and three example prompts. Limiting the amount of products to 15 for each iteration aimed to help users navigate a large product inventory, which was expressed as one of the main issues both in the situation assessment and in previous literature. By using example prompts grounded in the conversation context, the user is inspired to continue iterating on the search to find relevant products.

6.3 Evaluation: Technical Performance vs. User Experience

To ensure methodological evaluation rigor (Hevner et al., 2004), a mixed-method evaluation of the Style Assistant artifact was conducted. An automated RAGAs evaluation corresponds to Objective 3: *Evaluate the system performance using RAGAs as an automated evaluation method to assess retrieval effectiveness and recommendation quality.* This technical evaluation was combined with UX research including a usability evaluation, open questions and a questionnaire, to fulfill Objective 4: *Evaluate the perceived quality of the recommendations and conversational interactions through UX evaluation with a focus on usability metrics.* Consequently, this section examines the findings from the UX evaluation using the ISO usability framework (ISO 9241-11:2018) in combination with the domain-specific evaluation dimensions proposed by Jannach (2022). By contrasting these qualitative user insights directly against the quantitative performance metrics generated by the RAGAs framework, this integrated discussion ensures a comprehensive assessment in relation to the identified *ends* and implemented *means* of the Style Assistant artifact.

6.3.1 Usability Discussion

Based on Design Science Research (Hevner et al., 2004), the evaluation of the Style Assistant must critically analyze how the interaction between defined *ends* and technical *means* addresses the real-world product discovery bottleneck in second-hand e-commerce. By framing the UX results within the ISO usability framework (ISO 9241-11:2018) and integrating Jannach's (2022) CRS evaluation dimensions, this section will analyze the observed tensions between interaction flexibility, semantic understanding and execution efficiency.

Effectiveness

According to the ISO definition, effectiveness refers to the accuracy and completeness of goal achievement, a definition that directly intersects with Jannach's (2022) view of task support effectiveness in product discovery platforms. To establish problem relevance, as defined by

Hevner et al. (2004), the Style Assistant was designed and implemented as a purposeful technological artifact to address the real-world challenge of product discovery in second-hand fashion e-commerce, a domain heavily constrained by a large inventory of unique products.

The Style Assistant generally demonstrated strong effectiveness in task support, the fact that 100% of participants stated that they found something that they liked and would consider buying suggests that the artifact successfully addressed the underlying challenge of product discovery. Although the Style Assistant successfully supported the overall discovery journey, the UX evaluation showed that 23% of the individual interactions ended by the user giving up on their search. These two results need to be distinguished as the abandonment rate of 23% captures each individual session, whereas the 100% success rate reflects the users' general perception of the system at the end of the evaluation. This observation is a key finding as it highlights how interaction friction quickly exceeds user patience, strictly aligning with related literature by Gao et al. (2021), which suggests that too demanding or tedious interactions lead to higher rates of user abandonment. These results indicate that the system did not achieve perfect accuracy in every recommendation, but the overall results showed that the Style Assistant successfully supported users in fashion discovery and, as an artifact, therefore fulfilled the ISO standard for general effectiveness. While this general effectiveness was successful, a thorough analysis evaluates how the specific *ends* achieved effectiveness through the deployment of their corresponding technical *means* (Hevner et al., 2004).

Results showed that users found the system to understand subjective and stylistic terms such as styles and vibes, highlighting the effectiveness of this primary *end*. This finding directly supports previous research regarding fashion discovery's highly visual and subjective nature which traditional systems often fail to interpret (Deldjoo et al., 2025; Laenen et al., 2018). By successfully processing this, the multimodal RAG architecture proved to be an effective technical *means* for implementing semantically based searching. Consequently, the pre-retrieval processing steps, including intent classification and query rewriting outlined by Zhang et al. (2025), served as a well-fitted technical *means* for translating natural language queries into search queries optimized for semantic vector retrieval. In this manner, the Style Assistant effectively bridges the gap between fashion related semantic searches and the environment of second-hand e-commerce, mapping to the work of Laenen et al. (2018) that illustrates this complexity through user searches among the lines of "jeans with holes" might not match items described as "distressed jeans" in the product catalog.

Regarding the explore intent, several users responded positively to the system presenting whole outfit recommendations that integrated clothing, shoes and accessories. Furthermore, the questionnaire confirmed that users felt inspired to discover new items that they would not have manually searched for themselves. This positive finding demonstrates that the category-based multi-query retrieval strategy functioned as a highly successful technical *means* to support inspirational and exploratory *ends*. By providing diversity while satisfying specific constraints,

this approach fulfilled one of the primary product discovery goals established for the Style Assistant artifact during the problem analysis phase.

A critical insight highlights the tension of the chosen technical *means* managing explicit user constraints such as filters and negations to fulfill the Style Assistant's *end* of understanding natural language. While users found the system to generally apply explicitly stated filters, such as size and demography, several users expressed frustration that the system occasionally recommended items that did not fully correspond to their requirements, for example a specific color or brand. This exposes a fundamental trade-off between purely semantic vector retrieval and hard filtering. The system handles attributes like color and product type as purely semantic, leading to conversational and stylistic flexibility. However, this approach of soft filtering reduces absolute effectiveness in cases where users require certain constraints, for example by including noise as related shades of the specified color instead of filtering on mappable metadata. In the case of brand matching, this trade-off highlights an operational limitation of the artifact's current implementation, since only the 1000 most popular brands are represented, as well as misspelled brands failing to match the correct brand. Ultimately, while a purely semantic retrieval strategy increases interaction flexibility and is suitable for fluid stylistic discovery, the evaluation demonstrated that a lack of hard filtering constraints to enforce the stated *ends* can reduce the effectiveness of the *means* when managing explicit user requirements. This is especially true as the conversation develops further and users might transition from an exploratory mindset to a more targeted search, a behavioral shift that extends the core challenge identified by Wang & Na (2023) concerning the difficulty of accurately interpreting ambiguous or short user requests in relation to structured product metadata. As this transition occurs, the users' tolerance for stray results likely decreases. This introduces a crucial challenge for the system built on soft filtering as it fails to respect strict boundaries and ultimately fails to understand the true intent of the user.

A similar observation concerns the main *end* for the Style Assistant artifact to handle multi-turn interactions, including the user being able to iteratively refine their requirements while keeping previous statements in the conversation memory. The results showed that most users successfully narrowed down their search by refining, especially with the help of the image reference functionality, which was described as one of the most helpful mechanisms of the system. This indicates that multimodal vector retrieval is a helpful technical *means* for combining an image and refinement text to retrieve relevant products. Adding an image as reference decreases the need for explicitly stating user preferences, and is therefore seen as a major contribution of this thesis as this functionality is not present in the current application. This idea also corresponds to the essentialness of multimodal embeddings in fashion due to its strong visual nature as suggested by Deldjoo et al. (2025). However one frustration with refinement users experienced was that this technical *means* sometimes had a tendency to forget previous requirements in the conversation. This suggests that the consistency of the retrieval throughout the conversation was sometimes a challenge. When setting the weights of the combined vector the goal was to preserve text refinements by boosting them further to avoid being overshadowed by the image,

however these weights highly depend on the context and might have led to aggressively pruning or deprioritizing earlier constraints. The trade-off of maintaining coherence is similarly described by Gao et al. (2024) as the need to carry applied filters while avoiding dragging old, irrelevant constraint requests into the following iterations. As discussed previously, a possible solution could be to introduce hard filters to ensure that results follow explicit requirements, although this might introduce unwanted limitations, such as excluding semantically similar attributes when strictly filtering on the explicitly stated terms, for example specific shades of colors. Consequently, the artifact needs to address the dilemma that the more conversational turns a user takes to refine their search, the higher the probability that the system loses consistency, shifting the user from a state of progressive discovery to conversational frustration.

Efficiency

The efficiency of a system, defined by ISO 9241-11:2018 and Jannach (2022) as the relationship between needed resources, such as time, effort and cost, and the achieved results, presents an important trade-off within the Style Assistant. On one hand, the low perceived cognitive effort according to the users validated the system's capacity to recommend relevant products. By creating a smaller set of recommended items, the Style Assistant successfully minimized the efforts needed of manual browsing, which as previously explained is typically even more difficult in the context of second-hand fashion.

On the other hand, the reduced cognitive effort was heavily offset by the system response time. Performance results, see Table 3, showed an average response time of 18.4 seconds which represented a major architectural bottleneck. A latency of this magnitude disrupts the natural flow of human-computer interaction, transforming what should be a dynamic conversational dialogue into a set of waiting periods. As explained by Gao et al. (2021), systems perceived as too demanding may lead to boredom and user abandonment, and is therefore a crucial pain point of the system. The latency is a direct consequence of the technical *means* chosen to achieve recommendations of high quality in the form of highly serial steps in an advanced RAG pipeline. As presented in literature by Zhang et al. (2025), multimodal models often incur higher latency and resource consumption making this a known central design challenge, especially in combination with this increasing the risk of user abandonment presented by Gao et al. (2021). Therefore, to make the system feel smooth and effortless to align with user expectations for a CRS, the response time would need to be lower, especially as the ISO (9241-11:2018) definition of efficiency states time spent as a non-negotiable metric of usability.

Satisfaction

Satisfaction refers to the extent to which the system meets user needs and expectations according to the ISO definition (ISO 9241-11:2018). For a conversational recommender system, this relates to Jannach's (2022) dimension of Quality of Conversation and Usability, referring to the user's

subjective perception and linguistic quality of the system. Regarding the conversational quality of the Style Assistant, users agreed that the generated language felt natural and easy to understand, see Table 12, indicating that users were satisfied with the LLM-based text generation approach in the RAG pipeline. This means that the benefits of using an LLM in the generation step of the RAG pipeline in combination with the typewriter effect to pace the dialogue, helped to mimic a human interaction, which lowered the barrier for users to engage in interactions with AI-based systems. As presented in the literature, human-like traits lead to familiarity, similarity and liability as explained by Lopez-Lopez & Iniesta (2025) as well as Sidlauskiene et al. (2023). Anomalies seen, such as a couple of uncommon terms and referring to the incorrect item type, is therefore a direct consequence of open-ended prompts for the LLM to interpret. However, the generally high user satisfaction with the conversational naturalness points to this being successful. Confirming and rephrasing the user queries in the acknowledgment, an approach that was highly valued by users, aligned with the literature on "grounding in communication" as defined by Mamun et al. (2025) as a method of enhancing trust by demonstrating systematic comprehension. From an ethical standpoint, this strategy directly addresses the transparency dilemmas inherited by the "black-box" architecture, as described by Milano et al. (2020). Explicitly establishing this transparency decreases the ethical risk of untraceable AI behavior to create user trust as explained by Lopez-Lopez & Iniesta (2025). However, as some users pointed out that they skipped the acknowledgment highlights an interesting UX trend where users use CRSs as a search tool rather than a standard social chatbot by instead focusing on the visual product cards. This is an interesting finding that can be compared to Lopez-Lopez & Iniesta's (2025) explanation that whether a system engages the user or not is more important than any underlying technical complexity, as it also determines if it will be used or not.

Some users felt like the set of recommendations were random, especially during early iterations. This exposes a typical cold-start problem, directly aligning with previous literature. In such cases, the usage of personalization would be helpful as further discussed in Section 6.4.2 as it provides further information about the users preferences. When a user provides a vague initial query, semantic search returns broad matches that the user may interpret as a lack of system intelligence, instantly damaging the initial trust and satisfaction with the system. From an ethical perspective this dynamic reveals how fragile user trust is in cases where communication fails, directly aligning with Mamun et al. (2025) discussion on human-AI interaction dynamics. Scientifically, this finding highlights a constraint in contexts without anchors or historical interactions where the vector proximity alone cannot fully bridge the gap between somewhat abstract user language and precise stylistic expectations.

Generally, users were satisfied with the consistency and memory of the system, highlighting that the iterative refinement was a helpful way of narrowing down the product inventory. At the same time, some users experienced that earlier requirements were lost in the conversation during the refinement, which users found frustrating. This showed that the perception of the Style Assistant's consistency was somewhat mixed, contributing to both satisfaction and

dissatisfaction. These results can be explained by the dual pre-retrieval mechanism where explicit constraints are carried, while context-aware intent classifiers rely on exact phrasing and LLM interpretations to determine whether a user is refining or starting a new search, leading to divergent results. Especially in cases where users might be modifying their search, for example shifting from a dress to a skirt, the system must balance carrying or dropping the set filters as well as any irrelevant text, sometimes leading to incorrect results compared to the user's intention. This means that user satisfaction completely depends on the type of conversation had, where minor refinements seem to be handled more smoothly and struggling with larger refinements, for example when shifting categories but expecting the same style. This highlights user expectations for AI to possess human-like understanding of what counts as structural parameters, for example size or budget, versus more fluid stylistic desires, vibes or styles. These results can be interpreted as a need for a flexible continuity system where different cases must be individually handled, with no strictly general solution.

As previously discussed, many users expressed frustration regarding the high system response time, which contributed to dissatisfaction with the system. This indicates that while the overall user perception of the system regarding conversational quality was good, technical performance limitations such as response time lowered the user satisfaction with the system. Ultimately, these findings reveal that the user satisfaction of the Style Assistant is complex as it successfully met the linguistic expectations but underlying retrieval and memory mechanisms occasionally failed to sustain the consistency required for rigorous task support.

6.3.2 Comparison of Technical and Perceived Quality

To evaluate the RAG pipeline of the Style Assistant in the design cycle, an automated RAGAs evaluation suggested by Es et al. (2024) was identified in the rigor cycle as a suitable method to provide methodological rigor in the evaluation. This section discusses the results from the RAGAs evaluation in relation to the perceived quality from the UX evaluation. The results from the RAGAs evaluation were rather varying for the three different metrics used.

Answer relevancy received the highest scores among the three RAGAs metrics used, as seen in the results. This indicated that the generated answer answered the user queries well. Since the Style Assistant is not a traditional question-answering system, the retrieved content is not fully synthesized into the generated message. In this case, this means that the generated message did not actually say anything about the retrieved products but instead only evaluated the generated message in relation to the query. During the UX evaluation, several users expressed that they liked that the Style Assistant repeated their request in the message, creating a feeling of it understanding what the user meant. Also, users generally believed that the system's language was easy to understand. This indicates coherence and success for both the RAGAs evaluation and the UX evaluation regarding the generated messages.

Answer faithfulness stood out with a very low mean score. This indicated that RAGAs could not find that the generated answer was supported by the retrieved content, which could indicate hallucinations (Es et al., 2024). However, for the Style Assistant, the low scores can likely be explained by limitations in the retrieval context. Only certain metadata attributes are included in the context used by RAGAs. Since RAGAs cannot handle the product image, visual attributes not stated in metadata are not used by the RAGAs evaluation. A generated message including words expressing for example a style, vibe, occasion, fit or specific detail that cannot be retrieved in product metadata will therefore result in an answer faithfulness failure. This also explains why the scenario groups related to explicit filters scored a bit higher regarding answer faithfulness, as filters can be retrieved in the product context.

The Style Assistant artifact was built to handle visual attributes in fashion items semantically through multimodal vector embeddings, meaning that images and text exist in the same vector space. That meant that desired attributes such as style or fit were likely represented by the retrieved products, but since these attributes are only visually captured, RAGAs could not make that conclusion. Analyzing the results from the UX evaluation, users generally expressed that they felt like the system understood what style, fit or vibe they were looking for, especially compared to traditional keyword and filter-based search. This also supports that the low answer faithfulness score is a result of poor RAGAs data combability rather than a system performance issue. The UX evaluation showed that the generated message in some cases makes promises that were not fully true in relation to the retrieved products. However, this was mainly a problem for specific attributes, rather than broader exploratory terms. This problem was often more related to specific functionalities such as negations or reference items, instead of the full RAG pipeline.

For the retrieval part, RAGAs evaluated *context relevance*, which means how much of the retrieved content was useful for the generated answer (Es et al., 2024). The results were rather varied, where a clear score difference could be observed between different scenario groups. Scenario groups involving many specific metadata details that exist in the retrieval context, such as filters and specific product types and attributes, received higher scores, while scenarios with little to no explicit requirements, such as the explore and negation scenarios, receive lower scores. Again, as discussed for the answer faithfulness metric, this is a consequence of the textual nature of RAGAs evaluation. While image attributes have a significant role in the multimodal vector retrieval, this is not captured by RAGAs, resulting in low scores for scenarios where existing textual metadata has low relevance. This meant that low scores did not necessarily indicate a system failure but were instead a consequence of RAGAs not handling a multimodal knowledge base.

Analysis of the results from the automated evaluation showed that the RAGAs framework proposed by Es et al. (2024) is insufficient for capturing the visual image aspects of the implemented MM-RAG pipeline. Because fashion is a highly visual domain (Deldjoo et al., 2025), text-only metrics fail to capture important visual attributes that have a significant impact

on the Style Assistant's recommendation logic. This scenario highlights a central challenge in multimodal evaluation regarding the paradox between text-based automated judges and the visual context of second-hand fashion. While RAGAs was identified as a rigorous evaluation method for RAG pipelines in the rigor cycle, these findings from this thesis demonstrated that RAGAs framework would have to be extended in order to contribute to research rigor in multimodal settings such as the Style Assistant. This created a scientific contribution regarding how automated evaluation must adapt to visual domains. Future implementations are recommended to either textually represent image attributes in the evaluation context or to employ multimodal models acting directly as visual judges. Consequently, the evaluation did not only assess the performance of the Style Assistant's recommendations, but also generated new contributions to the broader field of MM-RAG evaluation, especially within the fashion domain.

6.4 Limitations and Future Research

Findings from the previous sections demonstrated the potential of the Style Assistant as a MM-RAG-based CRS in addressing the product discovery bottleneck in second-hand e-commerce, while also identifying several limitations in the current implementation. Addressing the identified constraints is essential to achieve a robust system in order to fulfill the utility goal of Design Science Research (Hevner et al., 2004). Consequently, these limitations point toward necessary enhancements for future work for the next design cycle iteration.

6.4.1 Technical Constraints

One identified technical limitation was the system's ability to handle complex user constraints. Occasionally, the system struggled to ensure that the user's requirements were fulfilled by the retrieved products, for example regarding negations or specific attribute requirements. The importance of natural language comprehension, including negations, and controlled retrieval based on this is emphasized as a core functionality in CRS literature (Nawara & Kashef, 2025; Mamun et al., 2025; An et al., 2025; Gao et al., 2021). Ultimately, this tension exposes a central design trade-off between semantic flexibility and strict payload constraints. This highlights a functional design lesson suggesting that the current *means* for handling these *ends* requires future refinement to ensure that the results actually follow strict constraints. Additional methods for ensuring these results could also be researched, for example implementing higher reliance on strict filters. The system needs to respect strict boundaries to prevent unrelated results which in other cases can break user trust.

Furthermore, the UX evaluation demonstrated scenarios that have no implemented *means* in the current version of the Style Assistant, leading to low recommendation relevance in these scenarios. First, several users requested complete outfits with several item types specified. This functionality is not handled by the Style Assistant, but it would be an interesting extension to, for example, implement several retrieval legs for the different specified item types. Second, several users used the reference pin functionality to find matching items. Again, no technical *means* for

this is implemented in the current version, since the reference item functionality only aims at making smaller refinements to the referenced product. Third, a recurring scenario was users trying to use the reference item functionality to pin multiple items, aiming to capture the overall style that they liked as feedback to the system. As stated by Deldjoo et al. (2025), it is often difficult to express stylistic requests, and being able to reference multiple items could therefore be a good solution to capturing the desired style or vibe that would be valuable to implement in the Style Assistant. Since these three scenarios were recurring in the UX evaluation, these would be interesting areas to research during future studies to enable the Style Assistant to handle these desired use cases.

Finally, the UX evaluation revealed latency to be one of the most critical constraints, highlighting a vital design lesson regarding the balance between pipeline complexity and conversational flow. The complex architecture introduces an extreme latency serving as an operational bottleneck caused by many sequential LLM-based tasks mainly during the pre-retrieval phase. Because this response time broke the natural flow of human-computer interaction and directly drove user frustration and abandonment, a primary recommendation for future implementation is to focus on pipeline optimization via architectural compression.

6.4.2 Personalization

The results showed that some users experienced the results to be a bit random, or not fully aligned with their expectations. Since the Style Assistant currently operates on a cold-start basis, it lacks knowledge about the user beyond the current session. This means that recommendations are solely based on the implemented pipeline, which treats all users equivalently.

In order to make the recommendations more adapted to the user, future research could explore integration of personalization into the Style Assistant. This could include using historical user data, such as past item interactions, preferred brands, sizes, stylistic preferences, aiming to provide higher relevance in the recommendations based on a specific user. Enhancing the recommendation relevance would also potentially lower the number of iterations needed to find relevant products. Furthermore, the problem analysis identified that users often find it difficult to articulate their specific preferences in written language, especially when specifying their style or certain aesthetics. The implementation of personalization could also reduce the cognitive burden of using the system, as the need for extensive natural language specifications as well as iteration effort would decrease. This would allow the Style Assistant to go from purely reactive to a proactive discovery tool that understands the user’s unique style preferences.

6.4.3 Adoption and Continuance

While this thesis focused on the technical performance and perceived quality of the RAG-based CRS, a broader question remains regarding the long-term willingness of users to adopt such an interface. It is yet to be determined if a conversational approach truly solves the identified discovery bottleneck.

The user interface design was, as previously stated, not the primary focus within the research scope. Evaluation feedback showed that certain UI limitations hindered the overall experience to a certain extent. Future research should treat the UI not just as a display layer, but as a critical component of the Style Assistant artifact. Refining as well as evaluating UI functionalities is a necessary step to transform the Style Assistant from a functional prototype into a high-utility tool, ready for production in relation to DSR (Hevner et al., 2004).

Future studies should investigate the continuance intention of users by researching whether they would return to the Style Assistant in real-world scenarios. Research should explore effective paths forward in second-hand e-commerce, or similar domains, to obtain the optimal solution. Understanding the transition from trying the Style Assistant to relying on it is a vital next step for both design science and e-commerce research.

As stated by Hevner (2007), in order to successfully meet the requirements from the relevance cycle, it is necessary to continue iterating on the design cycle based on identified limitations, until the Style Assistant actually serves as a solution to the identified design problem of product discovery in second-hand e-commerce.

7. Conclusion

The aim of this study was to address the product discovery challenge in second-hand fashion e-commerce by designing, developing and evaluating the Style Assistant, a Multimodal Retrieval-Augmented Generation (MM-RAG) based Conversational Recommender System (CRS). A Design Science Research (DSR) approach based on the three cycles presented by Hevner (2007) was used to address the problem.

The thesis identified second-hand fashion e-commerce as a unique and complex context, in which traditional search, filtering and recommendations systems are often insufficient due to data sparsity and the unique and constantly changing product inventory. Therefore, this domain requires a shift towards dynamic, context-aware and multi-turn recommendation systems. To address this, the Style Assistant was designed and developed to support multi-turn dialogue and multimodal semantic retrieval, enabling users to express visual fashion preferences in natural language. The resulting artifact pipeline implemented several pre-retrieval, retrieval, post-retrieval and generation strategies, aiming to enhance recommendation relevance while using multimodal embeddings to map images and text into the same semantic space.

Findings from this study showed that a MM-RAG-based CRS, such as the Style Assistant, has the potential to address the product discovery challenge by bridging the gap between natural language user requests and the visual nature of fashion products. The mixed-method evaluation confirmed that the primary aim of handling the product discovery bottleneck is successfully achieved since all participants managed to discover relevant items they would consider purchasing. Participants particularly appreciated features such as multi-turn refinements and nuanced natural language understanding. On a general level, these results showed that the artifact can serve as a solution to make sustainable fashion choices more accessible to users by simplifying product discovery.

Furthermore, the UX evaluation revealed that the main limitations of the system being latency and reliance on purely semantic retrieval that occasionally compromises explicit user requirements. To transform the Style Assistant from a functional prototype into a high-utility tool, future iterations within the DSR cycles must prioritize three core improvements. First, pipeline optimization concerning LLM subtasks and prompt handling is required to reduce latency and maintain a conversational flow. Second, exploration and implementation of methods to ensure correctness in the results, such as stronger reliance on hard filters. Finally, the system should incorporate personalization to avoid the static cold-start baseline by also including historical data and individual user preferences to deliver relevant recommendations.

Additionally, certain limitations regarding the evaluation method were identified. A critical insight is the shortcoming of automated textual frameworks such as RAGAs for evaluation of multimodal systems with strong visual importance. While the artifact yielded high perceived

satisfaction, some automated RAGAs metrics received low scores since the method did not capture visual features. This demonstrated that text-based judges are insufficient in multimodal settings, and future evaluation of multimodal systems therefore requires an extension of the RAGAs evaluation framework to incorporate visual features, especially in the fashion domain.

In conclusion, this thesis demonstrated that a MM-RAG-based CRS can effectively support product discovery in the complex context of second-hand e-commerce. However further research is needed to improve system efficiency and retrieval precision, as well as to incorporate personalization in order to allow the system to reach its full potential in enhancing product discovery and circular e-commerce.

References

Al Mamun, M., Ntsweng, O., David, A., Baah-Peprah, P., & Prybutok, V. R. (2025). What Drives User Intention to Continue Using Conversational AI? How Functional and Emotional Values Influence Continuance Intention. *AIS Transactions on Human-Computer Interaction*, 17(1), 1-34. https://doi.org/10.17705/1thci.00216 DOI: 10.17705/1thci.00216

An, G. T., Park, J. M., & Lee, K. S. (2025). NeGLU: Negation-Aware Sparse Retrieval With Negative Weights for Product Search. *IEEE Access*, 13, 144492-144504. doi: 10.1109/ACCESS.2025.3596549

Claudio, L., 2007. Waste couture: environmental impact of the clothing industry. *Environmental Health Perspectives*, 115(9), pp. A449–A454

Deldjoo, Y., Rafiee, N. and Ravanbakhsh, M. (2025) 'Agentic Personalized Fashion Recommendation in the Age of Generative AI: Challenges, Opportunities, and Evaluation', *ACM Transactions on Recommender Systems*, 1(1), 17 pages. Available at: https://doi.org/10.1145/nnnnnnn.nnnnnnn (Accessed: 4 May 2026).

Exploding Gradients (2026) RAGAs (version 0.4.3) [Computer program]. Available at: https://github.com/explodinggradients/ragas/tree/v0.4.3 (Accessed: 11 May 2026).

Gao, C., Lei, W., He, X., de Rijke, M. and Chua, T. S. (2021) 'Advances and challenges in conversational recommender systems: A survey', *arXiv preprint arXiv:2101.09459v7*.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M. and Wang, H. (2024) 'Retrieval-augmented generation for large language models: A survey', *arXiv preprint arXiv:2312.10997v5*.

Guiot, D. & Roux, D., 2010. A Second-hand Shoppers' Motivation Scale: Antecedents, Consequences, and Implications for Retailers. *Journal of Retailing*, 86(4), pp. 355–371.

Hevner, A. R., March, S. T., Park, J. and Ram, S., 2004. Design science in information systems research. *MIS Quarterly*, 28(1), pp. 75-105.

Hevner, A. R. (2007) 'A Three Cycle View of Design Science Research', *Scandinavian Journal of Information Systems*, 19(2), pp. 87-92.

Jing, Z., Su, Y., Han, Y., Yuan, B., Xu, H., Liu, C., Chen, K. and Zhang, M. (2024) 'When Large Language Models Meet Vector Databases: A Survey', *arXiv preprint arXiv:2403.01181*.

Khatwani, S. and Chandak, M.B. (2016) 'Building Personalized and Non Personalized Recommendation Systems', *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*. Pune, India, 9-10 September. IEEE, pp. 623-628

Laenen, K., Zoghbi, S. and Moens, M.F. (2018) 'Web Search of Fashion Items with Multimodal Querying', *Proceedings of the 11th ACM International Conference on Web Search and Data Mining (WSDM '18)*, Marina Del Rey, CA, USA, 5–9 February. New York: ACM, pp. 342–350. doi: 10.1145/3159652.3159716.

Lopez-Lopez, D. and Iniesta, M.B. (2025) 'The impact of conversational AI on consumer decision-making: A systematic review and cluster analysis', *International Journal of Engineering Business Management*, 17, pp. 1-15.

McInerney, J., Lacker, B., Hansen, S., Higley, K., Bouchard, H., Gruson, A. and Mehrotra, R. (2018). Explore, Exploit, and Explain: Personalizing Explainable Recommendations with Bandits. In: *Twelfth ACM Conference on Recommender Systems (RecSys '18)*. [online] Vancouver, BC, Canada: ACM, pp. 31–39. Available at: https://doi.org/10.1145/3240323.3240354.

Mohanty, I. (2023) 'Recommendation Systems in the Era of LLMs', *FIRE '23: Proceedings of the 15th Annual Meeting of the Forum for Information Retrieval Evaluation*. Panjim, India, 15-18 December. New York: ACM, pp. 142–144. doi: 10.1145/3632754.3632941.

Nawara, D. and Kashef, R. (2025) 'A Comprehensive Survey on LLM-Powered Recommender Systems: From Discriminative, Generative to Multi-Modal Paradigms', *IEEE Access*, 13, pp. 145772-145799.

Pradhan, L., Yu, L., Li, B., Simhadri, V. and Singarayar, J. (2023) 'Dynamic Filter Discovery and Ranking Framework for Search and Browse Experiences in E-Commerce', in *Proceedings of SIGIR e-commerce workshop 2023 (Conference SIGIReCom'23)*. Taipei, Taiwan, 27 July. New York: ACM, pp. 1–5.

Rubio, A., Yu, L., Simo-Serra, E. and Moreno-Noguer, F. (2017) *Multi-modal joint embedding for fashion product retrieval*. In: Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP). pp. 400–404. https://doi.org/10.1109/ICIP.2017.8296311

Shen, D., Ruvini, J-D., Mukherjee, R. and Sundaresan, N. (2012) 'A study of smoothing algorithms for item categorization on e-commerce sites', *Neurocomputing*, 92, pp. 54–60. doi:10.1016/j.neucom.2011.08.035.

Shu, Y. and Yu, Z. (2025) 'A Survey on RAG-Based Multimodal Embedding Techniques', in *2025 International Conference on Computers, Information Processing and Advanced Education (CIPAE)*. IEEE, pp. 372–379.

Sidlauskiene, J., Joye, Y. and Auruskeviciene, V. (2023) 'AI‐based chatbots in conversational commerce and their effects on product and price perceptions', Electronic Markets, 33(24)

Sinha, A.R. (2020) ‘Enhancing Global Search Functionality: A Comparative Analysis of E-commerce and Content Web Platforms’, *International Journal for Multidisciplinary Research (IJFMR)*, 2(4), pp. 1–13

Teixeira de Freitas, B.A. and de Alencar Lotufo, R. (2024) ‘Retail-GPT : leveraging Retrieval Augmented Generation (RAG) for building E-commerce Chat Assistants’, *XXXII Congresso de Iniciação Científica da UNICAMP*

Teubner, T., Flath, C.M., Weinhardt, C., van der Aalst, W. and Hinz, O. (2023) ‘Welcome to the Era of ChatGPT et al. The Prospects of Large Language Models’, *Business & Information Systems Engineering*, 65(2), pp. 95-101.

Vetenskapsrådet (2024) *Good Research Practice*. Stockholm: Vetenskapsrådet (Swedish Research Council). (VR2405).

Wang, H. and Na, T. (2023) 'Rethinking E-Commerce Search', *ACM SIGIR Forum*, 57(2), pp. 1-19.

Wang, Y., Dai, G., Ke, S. and Zheng, C. (2024) ‘Evaluating Sparse and Dense Retrieval in Retrieval-Augmented Generation Systems: A Study’, in *Proceedings of the 2024 10th International Conference on Communication and Information Processing (ICCIP '24)*. Lingshui, China, 21-24 November. New York: ACM, pp. 548–554. doi: 10.1145/3708657.3708747.

Winterflood, J. (2026) ‘The Recent Advances in Retrieval Augmented Generation (RAG) Systems’ in B. Stiller et al (Eds.) *Internet Economics XIX: IntEco Seminar*. University of Zurich, pp. 7-17.

Wu, W., Qi, Z., Tian, J., Wang, B., Tang, M. and Liu, X. (2025) 'An Enhanced Latent Factor Recommendation Approach for Sparse Datasets of E-Commerce Platforms', *Systems*, 13(5), p. 372. Available at: https://doi.org/10.3390/systems13050372.

Yang, S.C.-H., Rank, C., Whritner, J.A., Nasraoui, O. and Shafto, P. (2023) ‘Human variability and the explore–exploit trade-off in recommendation’, *Cognitive Science*, 47(4), e13279. Available at: https://doi.org/10.1111/cogs.13279.

Ye, F., Li, S., Fang, M. and Yilmaz, E. (2023) ‘Enhancing Conversational Search: Large Language Model-Aided Informative Query Rewriting’, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5985–6006.

Yu, X., Gan, T., Wei, Y., Cheng, Z., Nie, L. (2020) ‘Personalized Item Recommendations for Second-hand Trading Platform’, *In Proceedings of the 28th ACM International Conference on Multimedia (MM '20)*. Association for Computing Machinery, New York, NY, USA, 3478–3486. Available at: https://doi.org/10.1145/3394171.3413640

Zhang, R., Liu, C., Su, Y., Li, R., Huang, X., Li, X. and Yu, P.S. (2025) ‘A Comprehensive Survey on Multimodal RAG: All Combinations of Modalities as Input and Output’, *arXiv preprint arXiv:2511.17634v1*.

Appendix A. Survey on User Expectations

The following questions were included in a survey conducted to examine how users expect to interact with the artifact. A total of 11 participants responded to the questionnaire below:

Q1: What expectation would you have on a Sellpy chatbot?

Q2: In which cases would you use a Sellpy chatbot? (for example: searching for specific items, browsing for inspiration etc.)

Q3: How would you phrase a message to the chatbot? Please write an example (it can be anything, for example something you are currently interested in finding).

Appendix B. RAGAs Prompts

This appendix contains the specific prompt templates used by the judge LLM to evaluate the RAGAs metrics. These prompts define the instructions, constraints, and expected output formats based on the evaluation framework.

Context Relevance Prompt: Given a question, answer and context verify if the context was useful in arriving at the given answer. Give the verdict as "1" if useful and "0" if not with JSON output.

Answer Faithfulness Prompt 1: Given a question and an answer, analyze the complexity of each sentence in the answer. Break down each sentence into one or more fully understandable statements. Ensure that no pronouns are used in any statement. Format the outputs in JSON.

Answer Faithfulness Prompt 2: Your task is to judge the faithfulness of a series of statements based on a given context. For each statement you must return a verdict as 1 if the statement can be directly inferred based on the context or 0 if the statement can not be directly inferred based on the context.

Answer Relevance Prompt: Generate a question for the given answer and identify if the answer is noncommittal. Give noncommittal as 1 if the answer is noncommittal and 0 if the answer is committal. A noncommittal answer is one that is evasive, vague, or ambiguous. For example, "I don't know" or "I'm not sure" are noncommittal answers.

Appendix C. Post-evaluation Questionnaire

The following questionnaire and data metrics were used at the end of the UX evaluation of the Style Assistant. The evaluation is structured based on the ISO usability dimensions adapted from Jannach (2022). Questions are rated on a 7-point Likert scale (1 = Strongly disagree, 7 = Strongly agree) unless specified as Yes/No (Y/N). System-logged metrics that were internally collected are also listed at the end of this appendix.

Effectiveness

Basics

Q1: I found something that I liked. *(Y/N)*

Q2: I found something that I would consider buying. *(Y/N)*

Q3: The initial results felt relevant to me. *(1–7)*

Q4: I found the system helpful to find suitable items. *(1–7)*

Q5: I would use this system again. *(1–7)*

Search

Q6: The results fulfilled my request. *(1–7)*

Q7: It felt like the system understood what I was searching for. *(1–7)*

Explore

Q8: The results felt inspiring. *(1–7)*

Q9: The system helped me explore new items that I would not search for myself. *(1–7)*

Subtask

Q10: The system understood my intent. *(1–7)*

Q11: The results followed the applied filters. *(1–7)*

Q12: The system understood negations. *(1–7)*

Q13: The predefined prompts felt relevant. *(1–7)*

Efficiency

Q14: Much effort was needed to find something that I like. *(1–7)*

Q15: Much effort was needed to refine the search to find something that I like. *(1–7)*

Satisfaction

Q16: The system responded fast enough. *(1–7)*

Q17: The dialogue felt consistent throughout the dialogue. *(1–7)*

Q18: The system remembered my preferences throughout the conversation. *(1–7)*

Q19: The language was natural and easy to understand. *(1–7)*

Q20: I trusted the recommendations from the system. *(1–7)*

Open-Ended Questions

Q21: What did you like about the system?

Q22: What did you find difficult/frustrating with the system

Data

(Note: The following performance metrics were internally collected via the NoSQL database during the evaluation sessions and were not visible to the participants.)

- The system understood how much information was needed for searching. (data)
- Average time to find something they like. (data)
- Average number of iterations to find something they like. (data)
- Fraction of users that gave up before finding something they like. (data)
- Average response time. (data)