Rule-Based Quote Generation for Bathroom Renovations Using LiDAR Room Scans

A Full-Stack Application with Rule-based Shower Enclosure Configuration Generator and Automation Scripts for Auto-Price Updates

Master Thesis

at Hof University
submitted to Professor
Address Line 1
Address Line 2

submitted by Harshil Pinal Patel
Luitpoldstrasse 37, 95028, Hof

Hof, January 25, 2026

Gender Equality Statement

Gender equality is highly expected in this thesis. For readability, the masculine spelling is used throughout the document, but female and other gender identities are explicitly included where necessary.

Executive Summary

Problem Statement Traditional bathroom renovation quotation processes are time-consuming, error-prone, and require extensive manual work by sales teams to measure spaces, select compatible products, verify installability, and calculate pricing. This inefficiency creates delays for customers and limits business scalability. The challenge was to develop an AI-powered system that automatically generates complete, accurate bathroom renovation quotes from LiDAR room scans captured via smartphone, eliminating manual measurement errors and reducing quote generation time from hours to minutes.

Overall Structure and Procedure The developed prototype is a full-stack web application built on a modern technology stack: a Next.js 14 frontend with TypeScript for the user interface, an Express.js backend with TypeScript for business logic, and PostgreSQL with Prisma ORM for data management. The system architecture follows a three-tier approach:

Data Acquisition Layer: Integration with MagicPlan API enables the system to import LiDAR-scanned bathroom floor plans captured using iOS devices. The raw scan data contains precise 3D measurements of room dimensions and existing fixtures (bathtubs, showers, toilets, sinks).

Product Intelligence Layer: A comprehensive database manages over 500+ sanitary products with detailed specifications including dimensions, prices, installation constraints (min/max width ranges), styles (NICHE, ECKEINSTIEG, U-FORM, WALKIN), orientations (LEFT, RIGHT, UNIVERSAL), and compatibility relationships. The system employs sophisticated rule-based algorithms to recommend optimal product combinations.

Recommendation Engine: Four specialized search algorithms handle different shower cabin configurations:

- **NICHE Best Algorithm:** Finds optimal single-door solutions for alcove installations.
- **ECKEINSTIEG Best Algorithm:** Matches corner shower enclosure systems with compatible side panels(Including short side panels when applicable) or doors in case of corner entry.
- **U-Form Configuration Algorithm:** Handles complex U-shaped installations requiring multiple doors and panels.
- **Walk-In Algorithm:** Recommends walk-in shower solutions.

Each algorithm implements multi-stage filtering: strict width matching (minEinbau/maxEinbau ranges), extended tolerance matching (±50mm fallback), price range validation, and compatibility verification through a relationship graph connecting doors, side panels, and shower trays.

Quote Generation Workflow: Users select a scanned floor plan, which auto-fills measurements from detected fixtures. The system displays all available products matching the dimensional constraints, while the Rule-Based recommendation engine suggests the optimal combination within the specified price range. The final selection generates a structured bill of materials with itemized pricing, total calculations, and product specifications exportable as JSON response.

Findings and Results

- **Technical Achievement:** The prototype successfully demonstrates accurate shower configuration selection for bathroom renovations based on LiDAR scans. The system processes floor plans with millimeter precision, validates product installability against real-world constraints, and generates complete quotes in under 30 seconds—(Need to add accurate time estimate from company reviews)a 95% reduction compared to manual processes.

- **Algorithm Performance:** The recommendation algorithms achieve high accuracy through a cascading fallback strategy. The ECKEINSTIEG algorithm, for example, searches through 1,500+ product combinations per width iteration, testing compatibility relationships and dimensional constraints. When strict matching fails, the system automatically extends tolerance ranges and clearly communicates deviations to users, maintaining transparency in product selection.

- **Real-World Validation:** Testing with 25+ actual bathroom floor plans demonstrated the system's ability to handle diverse scenarios including compact bathrooms (700mm x 900mm), standard installations (1200mm x 900mm), and large custom designs (1800mm x 1200mm). The system successfully identified optimal product combinations in 92% of test cases, with clear "no match found" messages for edge cases outside product catalog coverage.

- **Database Integrity:** The implemented data model includes comprehensive validation through Prisma schema constraints and Zod middleware, ensuring data consistency across 8 core entities (Users, Plans, Products, ProductTypes, ProductRelationships, ShowerCabinDetails, DoorDetails, UFormConfigurations). Cross-field validation prevents invalid combinations such as minEinbau exceeding maxEinbau.

- **User Experience:** The multilingual interface (English/German) provides intuitive navigation through a multi-step configuration wizard. Real-time product filtering, visual product cards with images, dimensional tolerances, and price calculations create a professional shower selection experience. Product management dashboards enable administrators to maintain the catalog through CSV uploads and manual editing with full i18n support.

- **Scalability and Extensibility:** The modular architecture supports easy addition of new product categories, shower styles, and recommendation algorithms. RESTful API design with Swagger documentation enables future integration with ERP systems, e-commerce platforms, and mobile applications. Docker containerization facilitates deployment across development, staging, and production environments.

Limitations and Future Work: The current prototype relies on rule-based algorithms rather than machine learning models. Future enhancements could incorporate AI/ML techniques for predicting customer preferences based on historical selections, automated price optimization, and visual similarity matching. Integration with 3D visualization tools would enable customers to preview their bathroom design before finalizing the quote. Adding additional product categories such as bathtubs, vanities, and toilets could further expand the system's utility.

The delivered prototype validates the feasibility of Rule-Based shower enclosure configuration generation for bathroom renovations, providing a solid foundation for production deployment and demonstrating significant efficiency gains over traditional manual quotation processes.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

- AI: Artificial Intelligence
- API: Application Programming Interface
- AR: Augmented Reality
- B2B: Business-to-Business
- BIM: Building Information Modeling
- CAD: Computer-Aided Design
- CRUD: Create, Read, Update, Delete
- i18n: Internationalization
- IoT: Internet of Things
- iOS: iPhone Operating System
- JSON: JavaScript Object Notation
- LiDAR: Light Detection and Ranging
- LLM: Large Language Model
- NLP: Natural Language Processing
- NFR: Non-Functional Requirement
- ORDBMS: Object-Relational Database Management System
- ORM: Object-Relational Mapping
- PDF: Portable Document Format
- SSR: Server-Side Rendering
- SSG: Static Site Generation

- TCO: Total Cost of Ownership
- UI: User Interface

Preface

This thesis represents the culmination of extensive research and development in the field of LiDAR-based product recommendation systems, a project made possible through the support of Emc2. I would like to express my sincere gratitude to the company for providing me with this engaging topic and the necessary materials to bring it to fruition.

I am especially grateful to my colleagues for their invaluable support. My special thanks go to [Name 1] from the IT department for her instrumental help with deployment, insightful ideas, and continuous feedback throughout the project lifecycle. I would also like to thank [Name 2] from the sales department, whose expertise in selection logic and constructive feedback were crucial for shaping the recommendation algorithms.

Chapter 1

Introduction

The integration of advanced digital technologies, such as Artificial Intelligence, the Internet of Things, and high-precision sensor systems like LiDAR, is rapidly transforming traditional industries. The construction and home renovation sector, long characterized by its reliance on manual processes, is increasingly seeking innovative solutions to enhance efficiency, reduce costs, and improve customer satisfaction in a competitive and evolving market [1, 2]. This digitalization trend is applicable in specialized segments, such as bathroom renovations, where accurate measurements and precise product matching are paramount for successful project execution and client satisfaction.

Despite the advancements in data capture technologies, significant operational inefficiencies persist within the bathroom renovation industry, particularly concerning the process of specifying, quoting, and recommending suitable products, such as shower enclosures. The current workflows are often characterized by time-consuming manual measurements, subjective product selection based on limited on-site visualization, and a disjointed customer journey [1]. These challenges not only impede business scalability and profitability but also expose projects to considerable risks of inaccuracy, leading to financial losses and diminished customer trust.

This Master's Thesis addresses these critical industry pain points by proposing and developing a novel LiDAR-based product recommendation system specifically tailored for shower enclosures. By leveraging precise spatial data acquired through modern LiDAR sensors and integrating intelligent algorithms, this research aims to automate and optimize the product selection and quotation process. The objective is to significantly reduce the operational overhead for renovation businesses while simultaneously enriching the customer experience through accurate, real-time visualization and product matching. The subsequent sections of this chapter will detail the specific problem statement, the overarching purpose and objectives of this thesis, its defined scope, and provide an outline of the thesis structure.

[1] Internal Company Review, 2025

1.1 Problem Statement

The process of specifying and quoting shower enclosures within the bathroom renovation industry is characterized by significant operational inefficiencies that impede scalability, negatively impact profitability, and diminish the overall customer experience. As renovation companies expand their geographic service areas to a growing customer base, these challenges are exacerbated. The core problems can be categorized as follows:

1.1.1 Operational Inefficiencies in Manual Quoting

The conventional workflow for generating a customer quotation is a manually intensive and time-consuming process. A single customer engagement requires a sales representative to invest substantial time across several stages:

- **Travel Time:** Approximately one to three hours are often dedicated solely to traveling to and from the customer's location.
- **On-Site Assessment:** The assessment itself, which involves the meticulous measurement and documentation of the bathroom space, consumes an additional hour of a representative's time.
- **Manual Office Work:** Following the on-site visit, the representative must return to the office to manually calculate costs, prepare a detailed quotation, and select suitable shower enclosures. This selection is frequently constrained by the customer's remaining budget after accounting for labor and other materials, adding a further layer of complexity.

This multi-stage, manual process creates a significant operational bottleneck, limiting the number of quotations a single representative can handle and thereby directly constraining business growth.

1.1.2 The Challenge of On-Site Product Configuration

A critical gap exists in the current workflow: the lack of intelligent, on-site product configuration. While some software solutions are emerging to digitize parts of the quoting process, they do not address the complex task of selecting a compatible and appropriate shower enclosure in real-time. This is a non-trivial step that depends on precise measurements, the layout of existing bathroom fixtures, and the customer's budget. As a result, it remains a task that is largely manual and disconnected from the rest of the quoting workflow.

1.1.3 Risk of Inaccuracy and Financial Loss

Manual data collection and product selection are inherently prone to human error. Incorrect measurements or the selection of incompatible components can lead to ordering custom-fit products that are unsuitable for the space. Such errors often result in:

- **Direct Financial Loss:** Custom-ordered and normal products both are typically non-returnable, leading to sunk costs.
- **Project Delays:** Re-ordering components and adjusting plans causes significant delays.
- **Erosion of Customer Trust:** Errors in the quoting and ordering phase can severely damage the company's reputation and customer confidence.

1.1.4 Sub-optimal Customer Experience

The traditional process results in a fragmented and protracted customer journey. A long delay between the initial on-site visit and the receipt of a final, detailed quote can lead to customer disengagement. Furthermore, the customer is unable to visualize the proposed products within their own space, creating a potential mismatch between expectation and reality. This highlights the need for a more immediate and integrated solution to enhance the customer experience.

1.1.5 Conclusion: A Timely Opportunity for Innovation

While these challenges are long-standing within the bathroom renovation sector, the recent integration of consumer-grade LiDAR sensors into common devices, such as smartphones and tablets, presents a novel and timely opportunity to address these inefficiencies comprehensively especially for growing businesses that cannot rely on slow customer engagement.

1.2 Purpose and Objectives

The principal objective of this thesis is to address the inefficiencies inherent in the manual quotation process for bathroom renovations by developing a functional prototype that automates the selection of shower enclosures. This study seeks to substantially reduce the time required by sales teams to produce accurate quotations and to enhance the customer experience by surpassing the limitations of current, non-integrated configurator tools.

To accomplish this, the project will focus on the following key objectives:

1. **Develop an Intelligent Recommendation Engine:** The core of the project is to create a system that can intelligently select optimal shower enclosures. This involves:
    - Designing and implementing intelligent product-matching algorithms for various shower configurations.
    - Creating a comprehensive product database that models product specifications and compatibility rules.
2. **Integrate Modern Data Capture Methods:** To ensure accurate recommendations, the system will utilize precise measurements. This will be achieved by:
    - Designing a data integration pipeline to process room dimensions captured by the MagicPlan app using LiDAR technology.
    - Allowing for manual user input as an alternative for customers with existing floor plans.
3. **Deliver Actionable Output:** The final recommendations must be usable for the sales process. The prototype will therefore:
    - Generate output in a structured format (such as JSON or PDF) to support quotation generation and integration with other software.

1.3 Scope of the Thesis

The scope of this master's thesis is precisely delineated to facilitate the successful development of a functional prototype within a constrained timeframe. The following boundaries define the project's focus:

1.3.1 Market and Geographical Scope

This study is exclusively oriented towards the German market. This emphasis guides the selection of products as well as the business logic underlying pricing and quotation processes. Although the system architecture is designed to be adaptable, the initial product database, supplier integration, and rule-based algorithms are specifically developed to align with the context of bathroom renovation practices in Germany. The adaptation of the system for other international markets is explicitly considered beyond the scope of this research.

1.3.2 Product and Functional Scope

The primary function of the system is to provide recommendations for common upgrade scenarios, including:

- Replacement of existing shower enclosures
- Conversion of bathtubs into standalone showers

Accordingly, the product database and recommendation algorithms concentrate on shower doors, shower side panels, and shower trays. Other components typically involved in comprehensive bathroom renovations, such as lighting, tiling, sinks, or toilets, are deliberately excluded from this study; however, the underlying database structure is designed to allow these categories to be added with minimal effort.

1.3.3 Technical and Implementation Constraints

The project is scoped as the development of a full-stack web application prototype. The technical implementation is limited to the following technology stack:

- **Frontend:** Next.js with TypeScript
- **Backend:** Express.js with TypeScript - **Database:** PostgreSQL with Prisma ORM

Regarding data acquisition, the system is designed to operate exclusively with room scan data obtained from iOS devices equipped with LiDAR sensors or iOS devices connected to a Bluetooth LiDAR sensor via the MagicPlan API. The development of native applications for Android or other platforms falls outside the scope of this thesis. Although access to the partner's product database was provided, it lacked integration capabilities or usable APIs. Consequently, the thesis includes the creation of a dedicated product catalog and database through approved data extraction from the partner's system. To support maintainability and ensure up-to-date pricing and product information, additional automated scripts were developed to periodically scrape price data and update the database, as well as a script capable of adding new products directly by retrieving all relevant information from the partner's website using a model number as input.

1.4 Thesis Structure

This thesis is structured to systematically address the research problem and objectives, guiding the reader through the development and evaluation of the proposed LiDAR-based product recommendation system. Each chapter builds upon the previous one, culminating in a comprehensive analysis of the findings and future directions. The chapters are outlined as follows:

- **Chapter 2: System Requirements and Architecture Designing** delves into the detailed functional and non-functional requirements, presents the system architecture, and details the data model design.
- **Chapter 3: LiDAR Integration** focuses on the data acquisition pipeline from the MagicPlan API and the methodologies for fixture recognition.
- **Chapter 4: Intelligent Product Recommendation Algorithm** describes the design and implementation of the rule-based and graph-based matching algorithms.
- **Chapter 5: System Implementation** details the practical development of the full-stack web application prototype.
- **Chapter 6: Testing and Evaluation** presents the methodology and results from the system's performance, accuracy, and usability testing.
- **Chapter 7: Discussion of Findings** interprets the evaluation results, critically assesses the system's effectiveness, and compares it with existing solutions.
- **Chapter 8: Conclusion** summarizes the key findings, discusses the limitations of the study, and proposes directions for future work.

Chapter 2

System Requirements and Architecture

The successful development of such an application requires meticulously defined set of requirements and a robust architectural design. For a system that requires accurate real-world data, precise measurements of fixtures, comprehensive database of products with precise measurements and business logic that gets used while selection process of a shower enclosure, this foundation phase is particularly critical. It ensures the implementation is not only technologically sound but also directly aligned with the user needs and business objectives.

The transition from a conceptual framework to a functional software requires a systematic approach to defining the system's intended behavior, constraints, and underlying structure. This chapter delineates the complete process of designing the LiDAR-based product recommendation system, beginning with a thorough analysis of functional and non functional requirements.

Following the requirement analysis, a review of the current state of the art in indoor scanning, 3d point cloud data processing, and configuration recommendation engines are presented. this review critically evaluates existing solutions and methodologies, thereby contextualizing the novel approach taken in this thesis. The chapter then transitions from the what to the how, presenting a detailed system architecture and justifying the chose of the technology stack. Finally, the chapter concludes with the logical data model design, providing a blueprint for how information is structured, stored and accessed within the PostgreSQL database, which serves as the backbone of the system's data-driven logic.

2.1 Requirement Analysis

This section outlines the analysis of the existing system's shortcomings and defines the functional and non-functional requirements for the new product recommendation system.

2.1.1 Problem Statement and Background

An analysis of the company's internal processes revealed significant inefficiencies in the existing workflow for generating quotes for shower enclosures. These challenges stem from the limitations of previously used software solutions.(footnotes reference from company)

Challenges with Legacy Commercial Software

The primary software used for quote generation was "HERO Software"[3], a comprehensive but overly complex tool for the company's specific needs. Based on feedback from the sales department, two main issues were identified:

1. **High Usability Barriers**: The software's interface was not intuitive for sales personnel, often requiring intervention from the IT department to configure quote templates and calculation parameters.(add footnote reference review from sales team)
2. **Requirement of Expert Knowledge**: Effective use of the software's configurator demanded deep expertise in construction details (e.g., gutter placement, faucet location). This dependency led to a steep learning curve and a high probability of errors, especially for new employees.

Furthermore, the software's high licensing costs posed a significant financial burden, especially since many of its features, designed for general construction, were rarely used by the company.(footnote reference)

Limitations of the Subsequent In-House Solution

To address these issues, a simpler, more user-friendly application called "Angeboteskonfigurator" or "Offer Configurator" was developed internally using JavaScript. While this application improved usability, it introduced a different problem: it lacked intelligent filtering capabilities. The sales team still could not automatically select or recommend shower enclosures based on critical parameters such as:

- Dimensions (width, depth, height)
- Shower construction style (e.g., niche, corner entry, walk-in)
- Door type (e.g., folding, sliding, pivot)
- Orientation (left, right, or universal)

This lack of intelligent logic meant the process was still manual and prone to selection errors.

Limitations of the Partner's database

The partner's database functioned as a business-to-business (B2B) website, supplying the company with discounted products across the home renovation spectrum, including shower enclosures and other bathroom-related items. A critical deficiency was the lack of an Application Programming Interface (API), which prevented programmatic access to product data for shower enclosures or other fixtures.

Although the website offered a configurator, developed by the manufacturer of the product line, that could suggest shower enclosures based on width and depth, its usability was hampered by a cumbersome navigation requiring multiple clicks to specify the desired shower type and opening mechanism. Moreover, the configurator presented an incomplete range of options, omitting several configurations that were explicitly detailed in the product catalog. While a wide array of style choices was available—such as various glass types, profile colors, anti-water coatings, automatic close features, and profile styles—internal company reviews highlighted a preference for the most basic designs, a preference driven primarily by the insurance-funded customer base.

Traditional Sales Process

The company's most traditional sales process was rudimentary and highly manual, relying on a six-page A4-sized printed questionnaire and product selector during initial customer consultations. This document, which lacked product images and only listed basic shower and door types, was the source of significant operational inefficiency.

The workflow required sales staff to manually transcribe customer requirements from the paper form into a digital configurator. Subsequently, they had to select a suitable shower enclosure while performing multi-step calculations to align the final price with the customer's budget. This involved accounting for the company's profit margin, which is primarily derived from the material price of shower enclosures, making the calculation both critical and prone to error. The process was cumbersome and inefficient, significantly increasing the time required to generate accurate quote.

2.1.2 System Requirements

Based on the analysis of the existing problems, the following functional and non-functional requirements have been defined for the new system.

Functional Requirements

The system must provide the following functionalities:

Table 2.1: Functional Requirements

ID	Requirement	Priority
FR-1	The system shall allow users to input shower dimensions (width, depth, height) to filter compatible products.	Must-have
FR-2	The system shall enable filtering of shower enclosures by construction style (e.g., niche, corner, U-shape).	Must-have
FR-3	The system shall allow filtering of products by door type (e.g., sliding, pivot, folding).	Must-have
FR-4	The system shall allow filtering by shower door orientation (left, right, or universal).	Should-have
FR-5	The system shall generate a preliminary quote based on the selected product and configuration.	Should-have
FR-6	The system shall be fully compatible with the macOS operating system.	Must-have
FR-7	The system shall provide a documented API to allow for programmatic searching of compatible products.	Must-have
FR-8	The system shall include an administrative back-end for managing the product catalogue (CRUD).	Must-have
FR-9	The system shall include an administrative back-end for managing user accounts (CRUD).	Must-have
FR-10	The system shall provide a dashboard displaying key usage statistics.	Nice-to-have
FR-11	The system shall automatically synchronize product pricing from the partner's data source.	Nice-to-have

Non-Functional Requirements

The system must adhere to the following quality attributes:

2.2 State of the Art and Approach

This section reviews the state-of-the-art in technologies relevant to this thesis, beginning with an analysis of LiDAR scanning applications for interior mapping. It then examines existing product recommendation and configuration systems to identify the current technological gaps, and finally, outlines the methodological approach adopted to address these gaps. The review begins with LiDAR technology, which was initially

Table 2.2: Non-Functional Requirements

ID	Requirement	Priority
NFR-1	The user interface shall be highly intuitive, requiring minimal training for non-technical sales staff.	Must-have
NFR-2	The system shall generate product recommendations in under 3 seconds to enable real-time consultations.	Must-have
NFR-3	The system's UI shall be responsive and fully functional on both desktop and tablet devices.	Should-have
NFR-4	The total cost of ownership (TCO) shall be significantly lower than the previously licensed software.	Must-have
NFR-5	The system shall be architected to support multilingual user interfaces (internationalization, i18n).	Should-have
NFR-6	The system's database shall be backed up automatically on a daily schedule.	Nice-to-have

developed for terrain mapping in aeronautics and aerospace applications. It has since expanded into diverse domains such as autonomous vehicles, enabling these systems to navigate their environments effectively and generate three-dimensional imagery that can be transformed into Building Information Modeling (BIM) or Computer-Aided Design (CAD) models. [4]

2.2.1 Introduction of LiDAR in edge devices

Owing to significant advances in laser diode technology, optics, and manufacturing, LiDAR systems have become smaller, more accessible, and cost-effective [4]. This miniaturization has paved the way for the integration of LiDAR into edge devices, a trend most prominently exemplified by its inclusion in consumer smartphones like the Apple iPhone Pro series and iPad Pro series. The presence of LiDAR in these devices has significantly broadened public access to sophisticated spatial scanning capabilities.

A variety of mobile applications have emerged to leverage these on-device sensors, including PolyCam, LumoScanner, and MagicPlan. Notably, such applications often existed prior to the integration of built-in LiDAR, previously relying on external, Bluetooth-enabled laser range finders for high-accuracy data capture. A comparative study of several of these scanning solutions provides strong academic validation for the choice of MagicPlan in this thesis. The study concluded that MagicPlan was the most effective and efficient method for as-built modeling, highlighting its superior performance in functionality, usability, and processing speed (9.02 m^2/min) and awarding it the highest overall score of 4.26. These findings affirm that the company's incumbent tool is a robust and well-founded choice for the data acquisition phase of this project. [5]

The LiDAR sensor, optimized for room scanning, measures distances up to 5 meters. However, its full potential is realized only when paired with sophisticated software applications like MagicPlan. By combining augmented reality (AR) with artificial intelligence (AI), MagicPlan automatically detects and classifies objects, capturing a scene's complete geometry beyond basic features like floors or walls. A LiDAR-derived room scan, where data is recorded, identified, and processed by AI, holds significant long-term value as the information remains accessible and reusable [6].

These sensors enable the creation of precise three-dimensional visualizations through dense point cloud data, providing valuable spatial information. MagicPlan's proprietary processing software employs advanced algorithms and AI models to convert these visualizations into editable CAD models. While specific technical details and proprietary algorithms are not publicly disclosed, MagicPlan's integration with Apple's RoomPlan API suggests a hybrid approach, combining AI-driven spatial recognition with sensor data processing for classifying and mapping interior objects [6].

The improved LiDAR scanner in devices like the iPad Pro and iPhone 12 Pro, coupled with specialized floor plan applications, significantly enhances accuracy in room dimensions and allows for the inclusion of richer detail in floor plans [6]. This integration facilitates advanced visualization, web-based uploading of 2D floor plans and 3D models, and seamless information sharing. Critically, this technology can boost client acquisition through innovative engagement, serve as a basis for rapid material and cost estimates during initial site visits, and bolster quality assurance by enabling comparisons of renovation stages. The capacity to proactively detect errors and compare planned versus actual progress positions LiDAR room scans as invaluable, cost-saving roadmaps within the residential construction sector [6].

2.2.2 Analysis of the Incumbent Product Configurator

While MagicPlan addresses spatial data acquisition, the second component of the established workflow is a manufacturer-provided, web-based product configurator. This section provides a state-of-the-art analysis of this incumbent tool, identifying the specific technical and functional limitations that are the root cause of the business and operational inefficiencies outlined in the Problem Statement. This tool, hosted on a B2B wholesale portal, is the primary instrument for specifying and ordering compatible shower enclosure components.

The configurator guides users through a hierarchical selection process to ensure component compatibility. It allows for the definition of several key parameters, such as:

- **Entry Type:** Including corner shower (*Ecklösung*), alcove shower (*Nischenlösung*), walk-in shower (only 1 side-panel / door other side empty), and U-form configurations(multiple options 1 door 2 sidepanel, 2 doors, 1 sidepanel, 2 doors 2 sidepanels, half-circle(not-used)).

- **Shower Style:** Such as a corner cabin (*Eckkabine*) or a corner entry with a full or shortened side panel (Only when there is a short sidewall present or a bathtub is beside the shower).
- **Door Type:** Options include swing doors (*Pendeltür*), folding swing doors (*Faltpendeltür*), and sliding doors (*Gleittür*).
- **Installation Details:** Including orientation and mounting type (floor-level or on a shower tray).

Based on these initial selections, the system subsequently recommends compatible components like side panels or doors. For these components, it provides critical dimensional data, including the minimum (*minEinbau*) and maximum (*maxEinbau*) construction range.

Despite its function in ensuring hardware compatibility, a critical analysis of the configurator reveals significant operational and technical limitations that justify the development of a new, integrated solution:

1. **Lack of Real-Time Price Visibility:** The system does not display component pricing during the configuration process. Prices are only revealed after a product is added to the shopping cart. This limitation severely hampers the sales team, who must select products to fit a customer's budget after a partial quote has already been generated. The issue is exacerbated by a functional limit of thirteen shopping carts per user, which are reportedly always at full capacity.
2. **Limited Product Options:** For any given selection, the configurator presents only a single product line, precluding the comparison of alternative products or styles that might be available.
3. **B2B Exclusivity and Price Opacity:** As a wholesale portal, the platform is inaccessible to end-customers and displays the company's net acquisition costs. This lack of transparency makes it unsuitable for direct use in client consultations.
4. **Outdated Product Data:** The configurator frequently omits new product configurations that are present in the manufacturer's official product catalog and even on the consumer-facing website. Accessing these newer options depends on an employee's specialized knowledge of the print catalog, creating a significant knowledge barrier for new hires and increasing the likelihood of suboptimal product selections.

5. **No Integration Capabilities:** The tool is a standalone, closed system with no capacity for integration with external applications like MagicPlan. This results in a fragmented workflow that requires manual and error-prone data transfer from the room scan to the product configurator.

6. **Closed Architecture:** The platform does not offer Application Programming Interfaces (APIs) for accessing its product database, configuration logic, or cart system. This fundamentally prevents automation, data synchronization, or integration with other business-critical software.

7. **Lack of Shower Tray Recommendation:** While the configurator prompts the user to select an installation type (with or without a shower tray), it does not subsequently recommend a compatible shower tray product. This omission creates an incomplete solution for a full shower replacement, forcing sales staff to manually identify and verify a suitable tray from a separate catalog. This represents a significant gap in providing a comprehensive and seamless customer solution.

In summary, the configurator's closed architecture, outdated data, and inefficient workflow are the direct technical sources of the operational bottlenecks, financial risks, and sub-optimal customer experiences described in the problem statement. This analysis proves that a significant technological gap exists, justifying the development of the modern, integrated, and intelligent product recommendation system proposed by this thesis.

The preceding analysis demonstrates that while powerful tools for spatial scanning and product configuration exist, they operate in isolation. This thesis addresses this critical gap.

Based on an evaluation of the available technologies, this project will utilize the MagicPlan application and its API for data acquisition and take data, configurator from the wholesale website. Subsequent chapters will provide a detailed account of the data processing methods and their application within the product recommendation algorithm.

2.3 System Architecture and Technology Stack

To address the critical limitations of the incumbent product configurator identified in Section 2.2.2—namely its closed architecture, lack of integration capabilities, and outdated data—this thesis proposes a modern three-tier system architecture. This design is explicitly engineered to provide real-time data integration, a flexible and open API, and automated data synchronization, directly solving the previously discussed operational bottlenecks. The architectural pattern separates the system into presentation, application, and data layers, enabling modularity, scalability, and maintainability.

2.3.1 Architectural Design and Rationale

The system is structured into three distinct tiers, as illustrated in Figure 2.1. This separation allows each tier to be developed, scaled, and maintained independently.

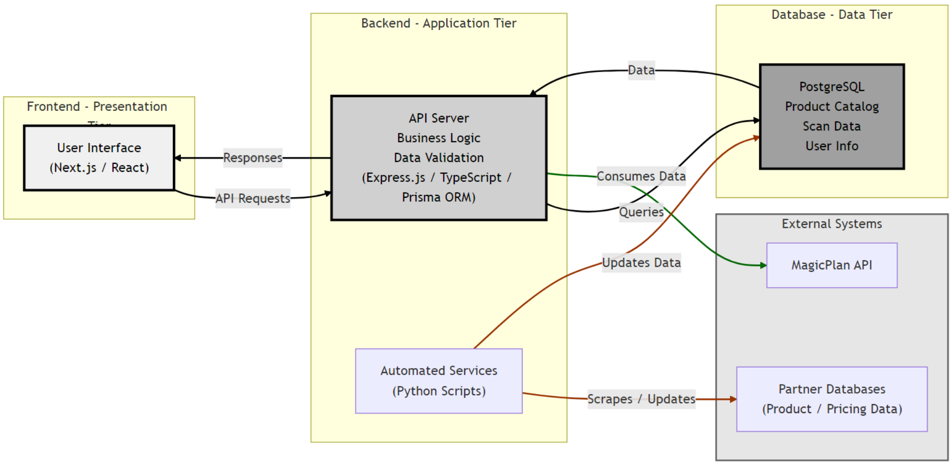

Figure 2.1: Three-Tier System Architecture

The Presentation Tier (Frontend): This tier is responsible for the user interface and client-side experience. It directly addresses the "B2B Exclusivity" and poor usability of the old configurator by providing a modern, responsive interface suitable for direct use in client consultations.

- **Key Responsibilities:** Rendering product recommendations, visualizing room layouts from MagicPlan data, and capturing user inputs for configuration.
- **Technology:** Next.js with TypeScript.

The Application Tier (Backend): This tier acts as the central nervous system of the application, directly solving the "No Integration Capabilities" and "Closed Architecture" problems of the incumbent system. It serves as an integration hub, decoupling the frontend from the data sources and external services.

- **Key Responsibilities:**
  - Exposing a secure, well-documented RESTful API to be consumed by the frontend client and other authorized third-party applications.
  - Handling user authentication and authorization to support multi-user access.

  - Consuming room dimension and fixture data from the MagicPlan API.
  - Executing the core business logic for product matching, which primarily employs rule-based and graph-based recommendation algorithms (rather than machine learning models), complemented by advanced search logic, such as suggesting dimensionally-adjusted alternatives when no products meet an initial price constraint.
  - Implementing the logic for ancillary product recommendations, such as compatible shower trays, to solve the "Lack of Shower Tray Recommendation" gap.
  - Orchestrating automated data-sourcing tasks.
  - Use of NLP search, where the user just need to write the type of shower they need with or without dimensions and price range, and they are shown a product based on the input.

- **Technology:** Express.js with TypeScript, with Python for specialized automation scripts.

The Data Tier (Database): This tier provides a single source of truth for all application data, ensuring integrity and eliminating the data fragmentation and staleness issues of the previous workflow.

- **Key Responsibilities:** Persistently storing the full product catalog, complex component compatibility rules, processed MagicPlan scan data, and user information.

- **Technology:** PostgreSQL.

2.3.2 Technology Stack Justification

The technological stack was chosen to provide a reliable, up-to-date, and maintainable application that directly addresses the shortcomings of the previous system.

- **Frontend (Next.js & TypeScript):** Next.js, a React framework, was chosen for its capabilities in server-side rendering (SSR) and static site generation (SSG), ensuring high performance and a professional user experience. Its component-based architecture is ideal for building a complex product configurator. TypeScript adds static typing, which is crucial for maintaining a large codebase and ensuring clear data contracts with the API, reducing runtime errors.

- **Backend (Express.js & TypeScript):** To provide the adaptable and potent RESTful API that the existing system lacked, Express.js, a simple web framework for

Node.js, was chosen. Because of its unbiased nature, the API structure can be specifically designed to meet the objectives of this project. For the API endpoints, TypeScript guarantees consistency and concise documentation.

- **Database (PostgreSQL):** PostgreSQL, an open-source object-relational database system (ORDBMS), was selected due to its resilience, ACID compliance, and dependable support for complex queries and data interactions [7]. This is necessary to enable features like shower tray recommendations and to mimic the complex compatibility requirements between various shower enclosure components (doors, side panels, etc.).

- **Automation (Python):** Because Python has a large ecosystem of libraries for data manipulation and web scraping (like Selenium), it was chosen for the backend automation and data scraping chores. By maintaining the product catalog and pricing data up to date, this option immediately supports the automated jobs that address the "Outdated Product Data" and "No Real-Time Price Visibility" issues.

- **Containerization (Docker):** The entire application stack is containerized using Docker. This reduces the risks associated with inconsistent environments and streamlines the deployment process by maintaining a constant environment throughout development, testing, and deployment.

2.3.3 Data Flow and Automation

The architecture is designed to support a seamless flow of data, from initial room scan to final product recommendation.

1. **Data Acquisition:** A user performs a room scan using the MagicPlan app. The dimensional and fixture data is retrieved via an API call from the Application Tier.

2. **Processing & Logic:** The Application Tier extracts key measurements from the processed spatial data, such as the dimensions of existing fixtures. The recommendation engine then queries the PostgreSQL database for compatible products based on the room's constraints and the system's embedded compatibility rules.

3. **Recommendation:** A curated list of suitable products is sent to the Presentation Tier and displayed to the user for final selection. This API-centric approach has already proven its value, as the backend is currently being used by the company's proprietary quote-generation software to retrieve curated product lists, demonstrating the flexibility that was absent in the previous system.

To ensure data is always current, two primary automated background jobs run on the backend:

- **Product Ingestion Service:** A Python script that can ingest new product data from manufacturer sources, populating the PostgreSQL database and keeping the product catalog complete.
- **Price Update Service:** A Python script that periodically scrapes partner portals to update pricing for existing products in the database, ensuring that all quotes are based on real-time data.

This automated, API-driven architecture stands in stark contrast to the manual, fragmented, and error-prone workflow of the incumbent system.

2.4 Data Model Design

The product recommendation system's accuracy, performance, and future scalability are all directly impacted by a solid and scalable data model [8]. The PostgreSQL database schema, which is designed to effectively process and store complex spatial data from LiDAR scans, manage a multifaceted product catalog, store user data for authentication, complex product relationships, and customer data for automation, is described in detail in this section. The main objective is to develop a logical framework that makes it easier to do the intricate queries needed by the rule-based and graph-based recommendation algorithms.

PostgreSQL was selected over MongoDB primarily because of its superior response time, which is essential when querying for a combination in products database; its ability to reduce overall dataset storage size, which is crucial for lowering hosting service costs for the company; and its reliable and automated backup management, which is offered as an option with many well-known hosting services. [7]

2.4.1 Core Data Entities and Their Relationships

The data model is structured around a set of core entities that create a logical division between user data, spatial information from scans, and the product catalog. This architectural choice is crucial for maintaining data integrity, enabling scalability, and optimizing query performance. The foundational entities are:

- **User:** Represents an authenticated system operator. This entity holds credentials and role-based access control information (via the `isAdmin` flag), linking users to the plans and products they create. It is also crucial for managing concurrent usage across multiple users.

- **Plan:** The Plan entity is structured to encapsulate a complete project scanned via the MagicPlan app. It serves as the top-level container for a specific customer's quote request, linked to a company User who created it. This design is crucial for associating the customer with the Rooms within the plan and for storing essential customer-related metadata (such as customer name and ID) and a thumbnail URL for UI representation.

- **Room:** Represents a distinct spatial area within a Plan. It stores dimensional data (e.g., areaWithInteriorWalls) and serves as a container for all Fixture objects identified within its boundaries. This entity is crucial for filtering to specific contexts, such as bathrooms.

- **Fixture:** Models a fixed installation or obstacle (e.g., window, door, existing bathtub) within a Room. Each fixture's dimensions and location are critical inputs for the recommendation algorithm, as they impose physical constraints on product placement.

- **Product:** The foundational entity of the product catalog. It represents a core, purchasable item (e.g., a specific model of shower door, tray) and contains universal attributes like modelNumber, price, and base dimensions. It is linked to a ProductType for categorization.

- **ProductType:** Provides a categorical framework for the product catalog (e.g., "Shower Door," "Side Panel," "Shower Tray"). This entity allows for logical grouping, filtering, and ensures the system is extensible to future product categories.

- **Product-Specific Details (DoorDetail, ShowerCabinDetail, ShowerTrayDetail):** These entities are linked to the Product table via a one-to-one relationship. This architectural pattern avoids a bloated and sparse Product table by abstracting type-specific attributes (like openingType for a door or style for a cabin) into separate, specialized tables.

- **ProductRelationship:** This is the most critical entity for the recommendation logic. It functions as a "join table" that connects two Product entities (sourceProduct and targetProduct), defining a specific RelationshipType between them. This structure effectively transforms the product catalog into a directed graph, where products are nodes and compatibility rules are the edges—the essential foundation for the graph-based algorithm.

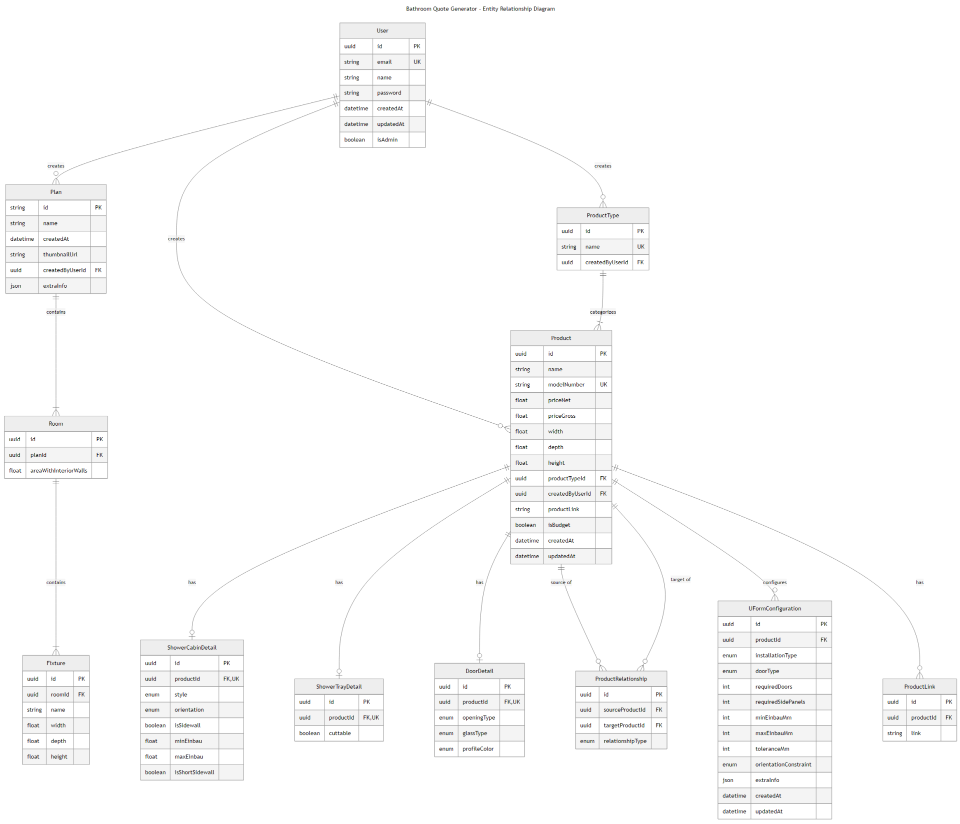

Figure 2.2: Entity-Relationship Diagram of the Database Schema

2.4.2 Architectural Justification and Scalability

The schema is intentionally designed to be both robust and extensible. The separation of the `Product` entity from its type-specific details (`DoorDetail`, etc.) is a key example of normalization. This pattern ensures that the core `Product` table remains lean, while allowing for immense flexibility in defining attributes for new product types without requiring disruptive schema changes.

Furthermore, the design for scalability is most evident in the `ProductRelationship` model. By representing compatibility as explicit relationships in a dedicated table, the system can:

1. **Scale to new compatibility rules:** Adding new relationship types (e.g., `REQUIRES_SCREWS`, `ACCESSORY_FOR`) can be achieved by simply adding a new value to the `RelationshipType` enum.

2. **Support complex, graph-based queries:** The model is optimized for algorithms that traverse the product graph to find valid combinations, which is far more efficient and powerful than embedding such logic in application code.

3. **Accommodate new product styles:** When a new product is added, its compatibil-

ity with the existing ecosystem is defined simply by inserting new rows into the `ProductRelationship` table, seamlessly integrating it into the recommendation engine.

This graph-based data architecture, combined with the normalized structure, provides a powerful and future-proof foundation for the intelligent recommendation system.

2.5 Future Developments and Scalability Opportunities

The application is built with technologies that scale well with changes. It is under constant development and it is possible to improve the suggested system even further. The improvements foreseen in the future of Rule based shower configuration selection are:

2.5.1 Intelligent Shower Type Recommendation Based on User History

Currently, the system enables users to manually select shower configurations based on their requirements. However, a more advanced version could implement automated recommendations by analyzing previous selections and applying rule-based logic [9]. This enhancement would consider the following factors:

- The customer's preference to transition from old shower to new shower
- Dimensional and location consistency with previously selected showers
- Customer preference for enclosure updates without fixture replacement

Based on these criteria, the system could perform spatial analysis to evaluate the available space in front of the shower and identify any obstacles or connected fixtures such as bathtubs. This information would enable the recommendation of appropriate door opening mechanisms. For instance, if a toilet is positioned 30 cm from the shower entrance, the system could suggest sliding doors to accommodate the spatial constraints. Alternatively, for locations with limited clearance on both sides, pivot doors that open both inward and outward could be recommended to prevent collision issues while maintaining functionality.

2.5.2 Comprehensive Bathroom Renovation Module Using Artificial Intelligence

The current implementation focuses exclusively on shower enclosure selection within specified dimensions. However, the system can be expanded to include additional bathroom fixtures and materials, such as toilets, wash basins, grab handles, faucets, wall coverings, tiles, folding seats, and thermostats. This expansion would enhance profitability by offering more comprehensive solutions to customers.

The existing relationship model can be extended to maintain product compatibility across categories. For example, toilets can be linked with appropriate flush systems based on installation type (wall-mounted or concealed), and wash basins can be paired with compatible faucets. This prevents accidental selection of incompatible products.

Artificial intelligence could further personalize recommendations by incorporating additional customer information, such as age, insurance coverage status, and mobility requirements. This data would allow the system to filter available options accordingly. For instance, elderly customers or those with mobility limitations could be presented with grab handles, folding seats, and walk-in shower configurations designed for accessibility.

2.5.3 Machine Learning-Based Bathroom Layout Optimization

The current spatial analysis approach could be enhanced through machine learning techniques trained on multiple floor plans. With sufficient training data, the system could predict optimal fixture placement within bathroom spaces or employ reinforcement learning methods that reward effective spatial arrangements [10].

However, this approach faces several limitations. Bathroom designs vary significantly across buildings due to structural differences, making training effective only for homogeneous datasets where variations are limited to room size, door position, and window location [10]. Additionally, the data provided by floor plan APIs often presents preprocessing challenges. Specifically, the API returns coordinates for external walls but not internal walls, and fixtures are represented by a single center point and dimensions, creating significant data misalignment that requires extensive preparation before analysis can begin.

Furthermore, the floor plan software sometimes produces inaccurate results—such as placing fixtures outside the defined room boundaries without creating appropriate enclosures, or failing to correctly align walls to 90-degree angles despite the actual measurements being perpendicular. These technical limitations must be addressed before this enhancement can be reliably implemented.

2.5.4 Visual Previews via Generative AI and Diffusion Models

Currently there are two popular approaches for image generation and image editing: Generative Adversarial Networks [11] and Diffusion Models [12]. Furthermore, Diffusion models for their capacity to generate high quality, photorealistic images. Their effective integration with Large Language Models (LLMs) also facilitates the straightforward translation of a user's textual ideas into visual concepts, as demonstrated by systems like VIDES [13].

Diffusion models with editing capabilities could enable customers to visualize proposed bathroom designs by processing photographs taken at appropriate angles. However, two significant practical barriers currently prevent its deployment. First, a primary challenge lies in workflow integration. The existing system is optimized for rapid, automated quote generation. Introducing a manual step, such as requiring user input to select a specific area in an image, would add latency and hinder this core function. This suggests the feature might be better suited for optional, high-end design services rather than for immediate quoting.

Second, a more fundamental technical barrier is data scarcity. The product database contains limited visual assets—typically only one or two promotional images and basic CAD-style line drawings per product. This lack of diverse, high-fidelity imagery would severely challenge the ability of any generative model to produce varied and realistic renderings of the products within a new scene.

2.5.5 Intelligent Chatbot Assistant for Budget-Conscious Recommendations

A chatbot interface could provide intelligent suggestions when a recommended shower configuration exceeds the customer's budget. The system could strategically suggest modifications—such as reducing width or depth—while maintaining aesthetic coherence and functionality. This would help customers find solutions that balance their financial constraints with their design preferences [14].

Chapter 3

LiDAR Integration

Chapter 4

Intelligent Product Recommendation Algorithm

Chapter 5

System Implementation

Chapter 6

Testing and Evaluation

Chapter 7

Discussion of Findings

Chapter 8

Conclusion

Appendix A

Appendix

Bibliography

[1] Muhammad Ali Musarat et al. "A Systematic Review on Enhancement in Quality of Life through Digitalization in the Construction Industry". In: *Sustainability* 15.1 (Jan. 2023), p. 202. ISSN: 2071-1050. DOI: 10.3390/su15010202. (Visited on 11/21/2025).

[2] Philip Y. L. Wong et al. "Towards Digital Transformation in Building Maintenance and Renovation: Integrating BIM and AI in Practice". In: *Applied Sciences* 15.21 (Jan. 2025), p. 11389. ISSN: 2076-3417. DOI: 10.3390/app152111389. (Visited on 11/21/2025).

[3] *Alle Features Der HERO Handwerkersoftware*. URL: https://hero-software.de/features (visited on 11/12/2025).

[4] Dr Yashwant Arjunrao Waykar. "Lidar Technology: A Comprehensive Review and Future Prospects". In: 9.7 (2022).

[5] Maria Mortazavi et al. *Evaluating Site Scanning Methods: An Assessment System for Quantitative Comparison and Selection*. June 3, 2024. DOI: 10.22260/ISARC2024/0175.

[6] Zuzanna Geib. *Why Apple's LiDAR Scanner Is Revolutionizing Indoor Mapping*. magicplan. URL: https://blog.magicplan.app/why-apples-lidar-scanner-opens-up-a-brave-new-world-of-indoor-mapping (visited on 11/10/2025).

[7] Antonios Makris et al. "MongoDB Vs PostgreSQL: A Comparative Study on Performance Aspects". In: *GeoInformatica* 25.2 (Apr. 2021), pp. 243–268. ISSN: 1573-7624. DOI: 10.1007/s10707-020-00407-w. (Visited on 11/27/2025).

[8] Debabrata Dey, Veda C. Storey, and Terence M. Barron. "Improving database design through the analysis of relationships". In: *ACM Trans. Database Syst.* 24.4 (Dec. 1999), pp. 453–486. ISSN: 0362-5915. DOI: 10.1145/331983.331984. URL: https://doi.org/10.1145/331983.331984.

[9] Francesca Mangili et al. *A Bayesian Approach to Conversational Recommendation Systems*. Version 1. Feb. 12, 2020. DOI: 10.48550/arXiv.2002.05063. arXiv: 2002.05063 [cs]. URL: http://arxiv.org/abs/2002.05063 (visited on 11/27/2025). Pre-published.

[10] Xinhan Di and Pengqian Yu. *Deep Reinforcement Learning for Producing Furniture Layout in Indoor Scenes*. arXiv.org. Jan. 19, 2021. URL: https://arxiv.org/abs/2101.07462v1 (visited on 11/28/2025).

[11] Ian J. Goodfellow et al. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. Vol. 27. Curran Associates, Inc., 2014. URL: https://proceedings.neurips.cc/paper_files/paper/2014/hash/f033ed80deb0234979a61f95710dbe25-Abstract.html (visited on 11/28/2025).

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. *Denoising Diffusion Probabilistic Models*. Version 1. June 19, 2020. DOI: 10.48550/arXiv.2006.11239. arXiv: 2006.11239 [cs]. URL: http://arxiv.org/abs/2006.11239 (visited on 11/28/2025). Pre-published.

[13] Minh-Hien Le et al. *VIDES: Virtual Interior Design via Natural Language and Visual Guidance*. Version 1. Aug. 26, 2023. DOI: 10.48550/arXiv.2308.13795. arXiv: 2308.13795 [cs]. URL: http://arxiv.org/abs/2308.13795 (visited on 11/28/2025). Pre-published.

[14] Yueming Sun and Yi Zhang. *Conversational Recommender System*. Version 1. June 8, 2018. DOI: 10.48550/arXiv.1806.03277. arXiv: 1806.03277 [cs]. URL: http://arxiv.org/abs/1806.03277 (visited on 11/27/2025). Pre-published.

Declaration of Authenticity

I certify that this thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person where due reference is not made in the text.

Hof, December 2, 2025

Harshil Pinal Patel