BABEŞ-BOLYAI UNIVERSITY CLUJ-NAPOCA
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
SPECIALIZATION COMPUTER SCIENCE

DIPLOMA THESIS

Tailored Crochet: AI-Powered Body Measurement Estimation and Custom Pattern Generation

Supervisor
Lect. PhD Mircea Ioan-Gabriel

Author
Calistru Diana-Maria

2025

UNIVERSITATEA BABEŞ-BOLYAI CLUJ-NAPOCA
FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ
SPECIALIZAREA INFORMATICĂ

LUCRARE DE LICENŢĂ

Tailored Crochet: AI-Powered Body Measurement Estimation and Custom Pattern Generation

Conducător științific
Lect. Dr. Mircea Ioan-Gabriel

Absolvent
Calistru Diana-Maria

2025

ABSTRACT

Crochet, a traditional textile craft, faces significant challenges in pattern customisation due to the diversity of body shapes, material properties, and crafter techniques. Existing tools often lack integration between body measurement extraction and pattern adaptation, leaving users to manually adjust designs with limited guidance. This thesis introduces Tailored Crochet, a digital system that bridges this gap by combining AI-driven body measurement estimation with customisable pattern generation. Using a simple neural network, the system extracts key anthropometric dimensions from front-view body images, eliminating the need for specialised 3D scanning hardware. These measurements, along with user-provided crochet swatch details (e.g., width, height, row count, and stitch count), feed into an application that generates personalised crochet patterns. The end-to-end solution includes a web-based platform built with React, Spring Boot, and Python, enabling users to create, save, and download the customised patterns. The evaluation results demonstrate that the system achieves an average mean absolute error of 1.64 cm across measurements, offering sufficient precision for garment adaptation in crochet. This work advances the accessibility and scalability of customised crochet design, providing both novice and experienced crocheters with a practical tool for creating garments tailored to individual bodies.

Keywords: customizable crochet, body measurement estimation, pattern generation, artificial intelligence, web application

Contents

1 Introduction **1**

1.1 Declaration of Generative AI and AI-Assisted Technologies in the Writing Process . . . 2

2 Literature Review **3**

2.1 Computational Approaches to Pattern Generation . . . 4

2.1.1 Pattern Representation Systems . . . 4

2.1.2 Automated Pattern Generation . . . 7

2.2 Body Measurement Extraction Techniques . . . 10

2.2.1 3D Scanning Approaches and Human Body Reconstruction . . . 10

2.2.2 Image-Based Measurement Estimation . . . 12

2.3 Digital Tools for Textile Crafts . . . 15

2.4 Synthesis and Research Gap . . . 17

3 Proposed Method **19**

3.1 Motivation for the Approach . . . 19

3.2 Dataset and Preprocessing . . . 20

3.3 Model Architecture . . . 20

3.4 Evaluation . . . 22

4 Application Development **24**

4.1 Requirements and Specifications . . . 24

4.1.1 Functional Requirements . . . 24

4.1.2 Non-functional Requirements . . . 26

4.1.3 Technology Stack . . . 27

4.1.4 Data Model . . . 27

4.1.5 Key UI Specifications . . . 28

4.2 Design analysis . . . 29

4.3 Implementation . . . 32

4.4 Testing . . . 34

4.5 User Experience 34

5 Conclusion **40**

Bibliography **42**

Chapter 1

Introduction

Crochet, in the form we now have, is a textile craft that has been around since the nineteenth century. It uses a single uninterrupted yarn strand and a hook to build up a garment or object, one loop at a time. Compared to knitting, which leaves the stitches open while working, each stitch in crochet is finished before starting the next one, which is why it has not been automated yet.

With the extensive reach of the Internet and endless tutorials, anyone can learn to crochet, but finding the right pattern is not always a trivial task. That could be complicated by something like wanting to combine the sleeves from one pattern with the torso from another. Sizing is also an issue since most of the tutorials and patterns available do not account for the wide variety of shapes and sizes of the human body. They provide one size and anyone that might want to crochet the object will have to figure out the size adaptation on their own in order for the garment to fit properly. This is, most often, a shot in the dark, especially for beginners. Even more so, crafter-specific variables (i.e. yarn thickness, crochet hook size, and tension in the crafter's grip) can alter the outcome of a project. The same pattern, when followed by two individuals with different working styles, can yield strikingly dissimilar products.

This thesis present Tailored Crochet, a digital framework designed to assist crafters of all levels in generating custom crochet patterns based on individual body measurements, swatch data, and style preferences. The work begins in Chapter 2, where existing research and digital tools related to pattern generation, body measurement extraction, and textile crafts are analysed. This review highlights current gaps, particularly the lack of integrated solutions that bridge measurement estimation and pattern customisation. In Chapter 3, I introduce an approach that combines MediaPipe-based key-point extraction and a neural network for body measurement estimation, using only front-view images and user provided height as input. This chapter details the data preparation, model architecture, and evaluation results that validate the method's accuracy for the intended purpose. Chapter 4 describes the design, implementation, and testing of the full-stack web application built to apply the proposed method. This includes an overview of requirements, technology choices, backend logic and integration of the AI measurement module. Finally, in the Conclusion (Chapter 5) I summarize the contributions, discuss limitations, and propose future directions for enhancing the system.

1.1 Declaration of Generative AI and AI-Assisted Technologies in the Writing Process

During the preparation of this work, the author used DeepSeek and ChatGPT to reformulate the text for increased clarity and to ensure a proper academic style. In addition,these tools were used to analyser messages and provide technical guidance during the software implementation phase when troubleshooting challenges arose. After using these tools, the author reviewed and edited the content as needed and takes full responsibility for the content of the thesis.

Chapter 2

Literature Review

Adapting and resizing a crochet pattern by hand can be a difficult task, especially for beginners. There are numerous patterns online for virtually anything that one might want to create, yet the chances of the finished product fitting properly without any modifications remain quite small. This is a prevalent problem among all skill levels, but especially for novices.

Furthermore, in the case of wanting to follow a design using a technique or yarn other than the recommended one, the stitch and row counts given in the original instructions would yield different measurements. This is where adapting the pattern based on a test sample that uses the desired techniques comes into play. In this way, some computations that involve the measurements of the crochet swatch, the values given in the pattern, and the actual body measurements solve the sizing and customization issues. This process needs to be streamlined so that any user, regardless of their knowledge related to mathematics or computer science, can follow it and adapt it to their needs.

The purpose of this chapter is to review and compare existing research and technologies related to computational pattern generation, body measurement extraction, and other digital tools for textile crafts. The papers found were organized by category, based on what their general topic was, and evaluated, mostly keeping in mind relevance to the theme and objectives of this thesis. Each paper was assigned a score based on subjective criteria, including rigour of its structural organization. By synthesizing these research threads, we identify both the opportunities and limitations in current solutions, which will aid in the development of the proposed pattern generation tool.

2.1 Computational Approaches to Pattern Generation

This section will dive into some fundamental concepts from the world of crochet, mostly related to visual and written representation systems and to automated approaches to the generation of crochet patterns. Both limitations and advancements will be explored, with the intention of becoming more familiar with the craft that is at the centre of this thesis.

2.1.1 Pattern Representation Systems

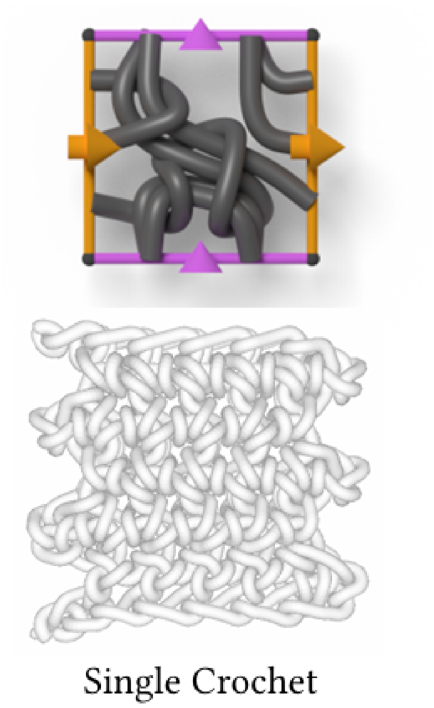
Even though all types of stitches are represented in sets of symbols and abbreviations that are well known within the crochet community, those can be overwhelming for someone just starting out. In this way, patterns written using those specific symbols are hard to understand at first glance and require a lot of extra attention.

There is also a need for machine representation methods in the context of digitally generating crochet patterns. Something that a machine could be easily programmed to understand and follow, whilst maintaining feasibility of the pattern and integrity of the garment that will result from it.

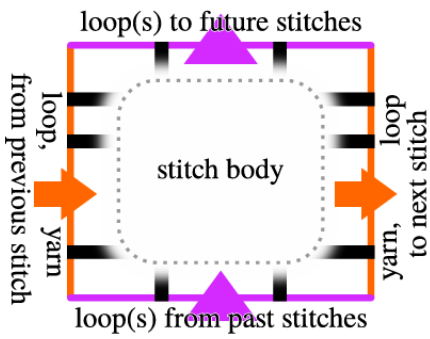
The following articles have explored various approaches to represent crochet patterns, both for human readability and machine interpretation. The key contributions refer to stitch meshes, graph-based languages, and the use of specific data structures.

Guo et al. [GLNM20] propose a ”stitch mesh” paradigm for representing crochet patterns, in which each individual stitch corresponds to a tile in a larger model (Figure 2.1a). The tiles themselves each represent a 3D rendered stitch as the current working loop and its connections to previous and subsequent stitches and rows (Figure 2.1b). Since crochet is highly dependent on where in the previous stitch or row the loop of yarn is pulled through, all possible situations need to be considered. They took into account all the little variations that would affect the way the stitches connect and interact with each other. Thus, they came up with an entire library of tiles that, when attached to each other, would behave like their real-life counterpart. This library does not encompass the entire complexity of this craft, as crochet allows for a lot of flexibility in the construction of different stitches.

Another approach that is quite interesting is a graph-based visual language proposed by Seitz et al. in [SRLH22]. This is a slightly more straightforward method compared to the ”Stitch-Mesh” technique. They address ambiguities in traditional crochet notation by introducing the use of graphs as a way to map out connections



(a) Single crochet tile and model



(b) Stitch dependency example

Figure 2.1: Stitch-Mesh Tile[GLNM20]

between stitches. The nodes represent the insertion points (for example holes or loops in the previously crocheted fabric), while the edges encode the dependencies between them (previous, insertion, slip stitch) (Figure 2.2). They analysed crochet charts that use standard notations and found problems that ultimately make the pattern unclear (Figure 2.3). The language supports representation of advanced techniques like increases/decreases, magic rings (the usual name for the technique that allows crocheting in the round) and insertions into specific parts of the available loops, since this can alter the aspect of the final structure. Those would otherwise be difficult to represent in the standard crochet charts or written patterns. With the aim of enabling interactive editing of the visual language, they also created an editor prototype for visualising and customising the crochet charts in both 2D and 3D, while also allowing for auto-completion of the charts. The downside of this tool is that it is limited to linear editing.

The work of Elena Zaharieva-Stoyanova and Damyan Beshevliev [ZSB18] focuses on the importance of a standardized set of crochet symbols for software applications that would eliminate platform dependency. Building on their earlier Portable Knitting Format (PKF), they extend this framework to crochet, enabling the creation of advanced stitches (e.g., popcorn stitches, shells) via combination of basic stitch primitives (e.g., chain, single crochet). Users can modify parameters such as

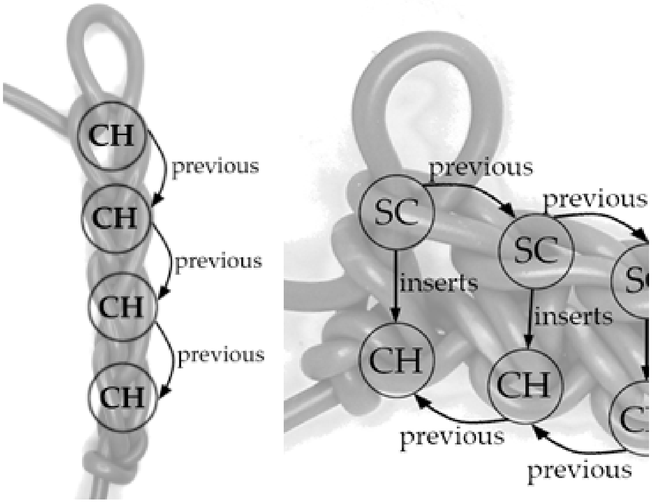


Figure 2.2: "Example graph structure mapped onto crochet fabric. The pattern on the left consists of a simple string of chain stitches (ch), which only depend on the previous chain stitch. The pattern on the right is a two-row pattern, with the first row consisting of chain stitches (h) and the second row of single crochets (sc), which are created by inserting into corresponding chain stitches from the first row." [SRLH22]

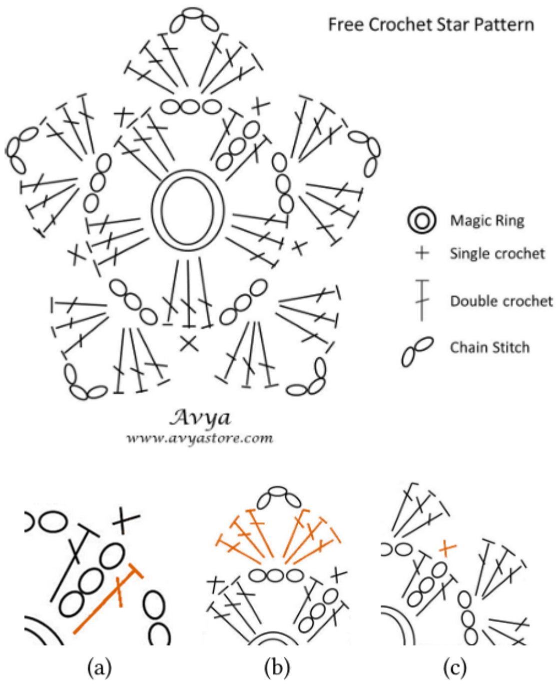


Figure 2.3: Chart Pattern using standard notations. "(a) unspecified closing method for the first round, (b) insertion point missing for the six double crochets on each point of the star, (c) unspecified closing method for the last round "[SRLH22]

stitch count or insertion points to generate new symbols programmatically. Their approach standardizes notation mainly with cultural heritage preservation in mind, such that it remains constrained by platform-specific implementations, limiting cross-software interoperability. However, their XML schema provides a foundation for custom stitch libraries that would adhere to already existing standards.

Karmon et al. present KNITIT[KSS+18], a computational pipeline for industrial knitting that bridges design intent and manufacturability through a matrix-based data structure. Their tool translates diverse inputs (e.g., images) into a 2D character matrix, where each symbol corresponds to a predefined knitting structure (e.g., 'A' for dense stitches, 'B' for net-like patterns)(Figure 2.4). This matrix is then compiled into machine-readable Jacquard files, mapping symbols to needle operations (knit, tuck, miss) while preserving geometric proportions through parametric compensation strategies. The key to their approach is the decomposition of patterns into repeatable blocks and a library of structures with measured physical properties , enabling dynamic adjustment to lessen fabric deformation. For my tool, this matrix logic is directly adaptable: replacing knitting structures with crochet stitches (e.g., 'sc' for single crochet, 'dc' for double crochet, etc.) would allow seamless translation from an internal symbolic representation to user-friendly crochet instructions. KNITIT's framework, though created specifically for knitting, underscores the viability of matrix-driven systems for textile customization, offering a model for automating pattern generation in crochet.

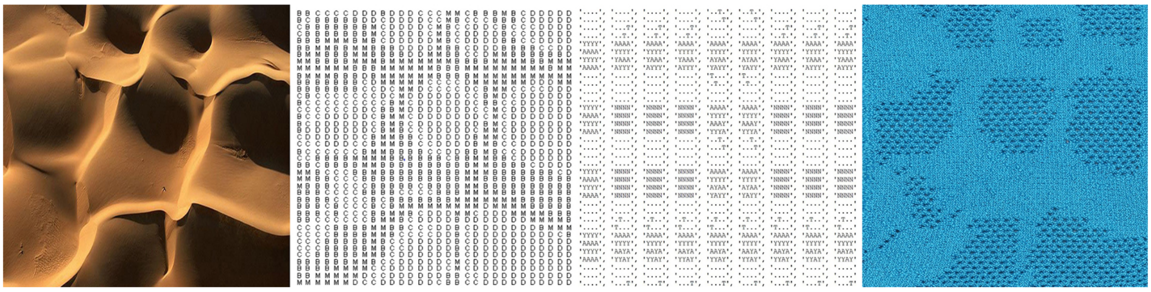


Figure 2.4: "The KNITIT pipeline: input data is translated into strings, which are transformed into an array of machine commands for the fabrication of multi-structured knits"[KSS+18]

2.1.2 Automated Pattern Generation

Even though it is constantly studied and attempted, automation of the crochet process has not been successfully done yet. Certain attempts, like the work of Storck et al. in "Design tool for automated crocheting of fabrics" [SFK23], mimic the aspect of crochet, but since the process they used is very similar to automatic knitting, the resulting fabric exhibits a "technical front and back". This is not the case for hand crocheted pieces. Now, even though the process itself cannot be automated, the pattern generation can.

In ”Computing Stitches and Crocheting Geometry”, Capuman et al.[CBG17] introduce a computational framework for generating crochet patterns from 3D CAD models. They analysed both what they refer to as determinate variables (material properties like elasticity and thickness as well as hook size) and indeterminate variables (tension in the crafter′s grip) in order to establish their effects on the size and structure of the resulting crochet piece. Following this, they came up with a set of ”crocheting rules” that would be used as reference points in crafting a computer algorithm. Their main focus was to develop a method that would be able to produce a pattern for virtually any 3D shape that would then be crocheted using three simple crochet rules: single crochet, increase, and decrease(Figure 2.5). To calibrate individualized patterns, the authors introduce a 10x10 tension swatch that captures the unique movements of the crafter and the material choices, effectively translating indeterminate variables into quantifiable inputs(Figure 2.6). This process allows crafters to generate unique crochet patterns for the same digital design. Just as adjusting the number of polygons in a 3D model changes its level of detail, small changes in the swatch, like using thicker yarn or pulling stitches tighter, directly affect how dense or sparse the final crochet pattern needs to be. By testing their method on complex shapes like branching structures, the authors showed how computers can turn hands-on crochet skills into clear, repeatable rules, linking tactile nature with digital accuracy. Their approach maps how stitches should be spaced across curved surfaces and refines the process through group workshops. This shifts crocheting from a traditional copy-based craft to a flexible way of making personalized items at a scale. Here, computer-generated rules ensure reliable results while still leaving room for creative tweaks.

Building on their earlier work, which was presented in the previous paragraph, the authors expanded their computational framework to address branching geometries: complex shapes that split into multiple arms or components [CBG22]. They kept the approach of a test swatch, which ensured that even when multiple people collaborate, the final pieces fit together seamlessly, matching the digital model’s dimensions. This was tested by two students who collectively made a 14-part branching structure. The system outputs step-by-step text instructions (similar to a recipe), making it possible to crochet intricate forms without expensive machines. This work also highlights crocheting’s potential for sustainable, large-scale projects, since yarn can be reused, and components can be made separately and assembled later.

Nakjak et al. [NRP18] focus on a different branch of crochet: Amigurumi dolls, a popular style of small crochet plushies. They developed a system that turns hand-drawn 2D sketches into 3D models and crochet patterns. The users sketch basic shapes that are matched by an algorithm with one of three predefined structures:

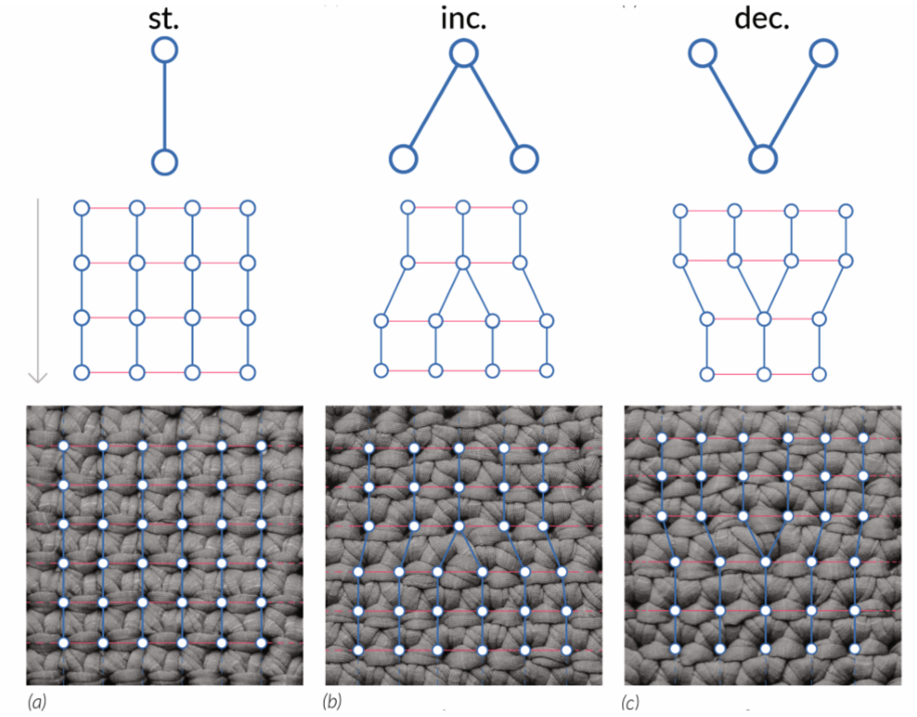


Figure 2.5: "(a) single crochet: plain stitch; (b) increase: diverging two stitches from one stitch; (c) decrease: converging two stitches into one stitch"[CBG17]

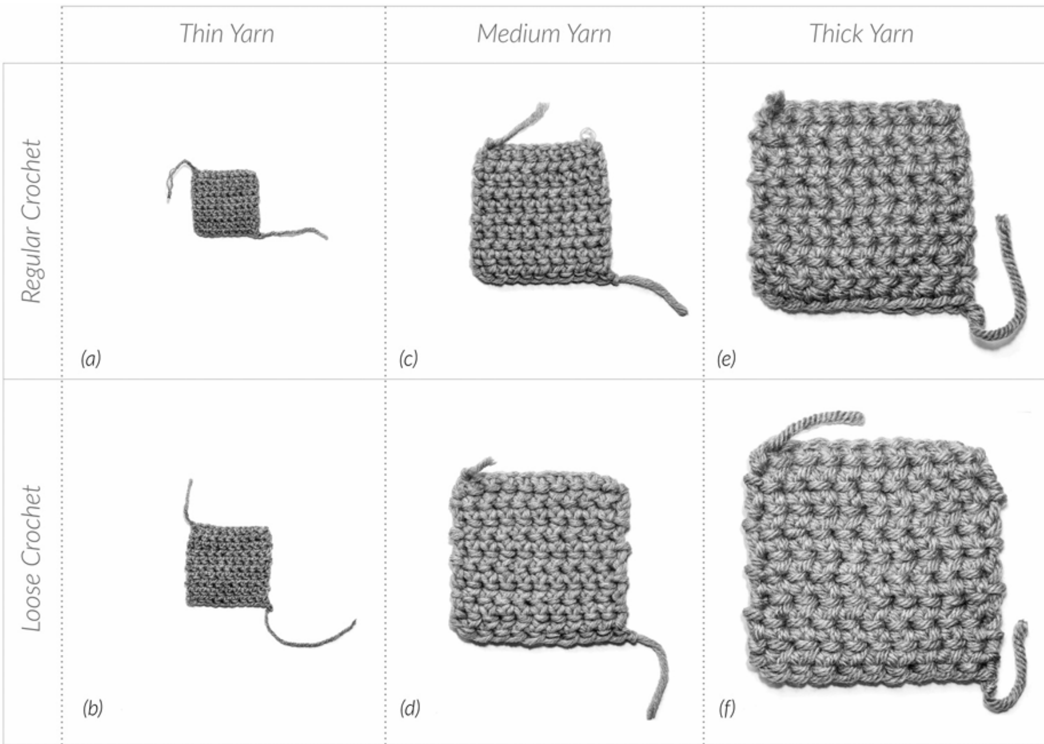


Figure 2.6: "Six different 10-by-10-stitch swatches to capture the crafter's grip on the yarn"[CBG17]

sphere, tear drop, or cylinder, since those are the main shapes of which the dolls are made up. To match the ”cute” Amigurumi aesthetic, the system adjusts proportions (e.g., enlarging the head to twice the body size) and ensures symmetry by mirroring parts such as the ears, arms, or legs and then automatically converts them into patterns. Mathematical formulas calculate stitch counts based on the model’s dimensions, guaranteeing that patterns are feasible for hand crocheting. In tests, users only needed 2-7 minutes to create doll designs, and the generated patterns produced objects closely matching their sketches.

2.2 Body Measurement Extraction Techniques

Having in mind the need for a simplified and accessible process for users of all backgrounds, the purpose of this chapter is to explore available techniques for extracting all necessary body measurements that will later be used in the pattern customisation step. The approaches studied range from the use of specialised scanners to the use of neural networks to estimate measurements from various inputs.

2.2.1 3D Scanning Approaches and Human Body Reconstruction

Even though specialized scanners exist and would yield the most accurate results, they are not accessible to the wider public, nor are they needed in this context. For customizing sizing in crochet pieces, the level of detail necessary is quite low and a simple convolutional neural network for silhouette extraction would be enough. Pose variability, viewpoint ambiguity, and the need for detailed surface geometry, are all problems that benefit from the advancements in 3D human body reconstruction that make use of deep-learning and parametric models.

A common foundation across several works is the Skinned Multi-Person Linear model (SMPL), a parametric representation of human body shape and pose. For instance, Suh et al. [PTSZL22] integrated a U-Net segmentation model for extracting the silhouette from frontal and side-view images of the human body. For their specific implementation, the task of the U-Net was to label each pixel in the provided images as either part of the background or part of the silhouette. This produced images in which the background was black and the silhouette of the body was white, isolating the two main components (Figure 4.1). Their method simplified industry adaptation, since U-Net is an easy to implement and train, and it requires less computational power in comparison to other architectures, while maintaining accuracy. Similarly, Zhu et al. [JQW+19] and Zhang et al. [ZX21] extended SMPL by incorporating mesh deformation techniques. The first introduced a hierarchical deformation strategy, refining the initial SMPL estimate using joint, anchor, and vertex-level adjustments, guided by 2D projections and shading cues. The latter, employed Graph Convolutional Networks (GraphCNN) to deform SMPL meshes in a coarse-to-fine manner, recovering surface details. All three of these approaches demonstrate how parametric models can serve as starting points that can be later enhanced by free-form deformations to capture intricate geometries.

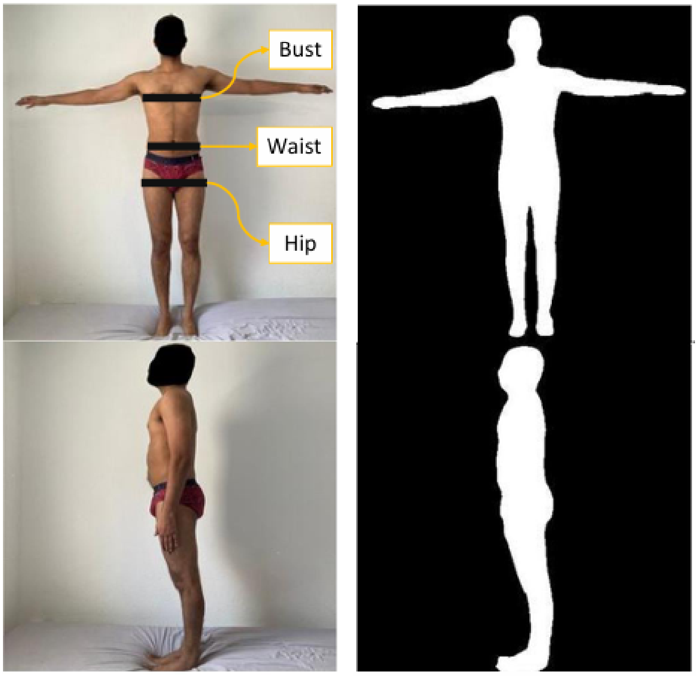


Figure 2.7: Frontal and side-view input images with the extracted silhouettes [PTSZL22]

The choice of input data, whether single-view images, multi-view silhouettes, or segmented masks, significantly influences the reconstruction quality of different CNN based approaches, balancing trade-offs between geometric detail, robustness to occlusions, and practicality in real-world applications. Multi-view methods, which leverage frontal and side silhouettes or binary masks, reduce ambiguity by incorporating explicit geometric information. For instance, frameworks employing two-branch CNNs with PCA-based shape descriptors have demonstrated that weight-sharing between branches enhances accuracy when processing front and lateral views ([CZY20]). In contrast, single-view approaches prioritize scalability, reconstructing detailed shapes directly from RGB images by fusing features into 3D mesh topologies ([ZX21] and [JQW+19]). While multi-view techniques benefit from structured geometric cues, single-view solutions face inherent challenges in resolving occlusions and depth ambiguities, trading precision for broader applicability.

Idrees et al. [IGa24] provide a review of mobile apps that allow contactless scanning of the human body. Eighteen 3D scanning mobile apps that provide a detailed analysis of the body were examined and it resulted that the scanning time is relatively fast, taking anywhere from 30 seconds to 5 minutes. Approaches make use of AI and computer vision to extract measurements from simple smartphone photos. Different methods use either two images (front and side view or back, and side view), three images (front, back, and side view) or a full 360-degree scan, achieved via video, in order to generate the 3D model of the body and a list of precise measurements. Most of the reviewed apps use this for features like body tracking, virtual try-on, size and fit visualization and other things relevant to personalized online shopping. This study provides a lot of organized information pertaining to each app, the approach used, details related to the resources available, the types of data required from the user and the data given as a result (time required, list of measurements, accuracy etc.) (Example of such a table in Figure 2.8).

Sr No.	*Application Name*	*3D Scan, Import and Export Files in Format*	*Customer Dimensions*	*PAAS*	*OEM*	*API*	*SDK*
1	**3DsizeME**	IMED, VRML, STL, AOP, PLY, or OBJ.	CSV	No	No	Yes	Yes
2	**3D Look**	OBJ	CSV	No	No	Yes	No
3	**Nettelo**	OBJ and STL customer dimensions in CSV	CSV	No	No	Yes	Yes
4	**Me-Three-Sixty**	OBJ and STL customer dimensions in CSV	CSV	No	No	Yes	Yes
5	**Mirrorsize**	OBJ and STL customer dimensions in CSV	CSV	No	No	Yes	Yes
6	**Verifyt Body**	OBJ and STL	CSV	No	No	Yes	Yes
7	**3D Avatar Body (IBV)**	OBJ and STL	CSV	No	No	Yes	Yes
8	**Sizer Me/ Sizer M2M/ Sizer Pro**	OBJ and STL	CSV	No	No	Yes	Yes
9	**SizeYou / Size You Pro**	OBJ and STL	CSV	No	No	Yes	Yes
10	**SizeMeRight**	No	No	No	No	No	No
11	**ASizer: Virtual Fitting Room**	No	No	No	No	Yes	No
12	**1Measure**	OBJ and STL	CSV	No	No	No	Yes
13	**Fashion Tech**	No	No	No	No	Yes	No
14	**Right Fit**	No	No	No	No	Yes	No
15	**ZoZo app**	OBJ and STL	CSV	No	No	Yes	Yes
16	**3D Measure Up**	OBJ and STL	CSV,PDF,HTML	Yes	Yes	Yes	Yes
17	**Presize.ai**	No	CSV/JSON *via* URL	Yes	Yes	Yes	Yes
18	**Meepl**	No	No	No	No	Yes	No

Figure 2.8: This table provides details about scan and file formats, OEM (Original Equipment Manufacturer), PaaS (Platform as a Service), API (Application Programming Interface) and SDK (Software Development Kit) for each app. This information can "aid software developers to use specific platforms to develop and personalize applications"[IGa24]

2.2.2 Image-Based Measurement Estimation

Image-based measurement estimation (IBME) has emerged as a scalable and accessible alternative to traditional 3D scanning for acquiring body dimensions, especially in the personalised garment design. Unlike specialized hardware-dependent methods, IBME leverages widely available tools, the most common being smartphone cameras and computer vision algorithms that extract anthropometric data from images.

Tejeda and Mayer studied how certain factors (gender, pose and camera distance) affect the accuracy of the results given by a convolutional neural network (CNN) in the context of Human Body Dimension Estimation (HBDE) from images. They trained and evaluated the CNN focusing on four different scenarios: "(1) training with subjects of a specific gender, (2) in a specific pose, (3) sparse camera distance and (4) dense camera distance" [TM22]. Since the cost and effort of collecting real-world data would be too high, they decided to use a synthetically augmented dataset based on the SMPL model (Skinned Multi-Person Linear model), which is derived from real humans and it is the most employed model in the industry because of its realism and simplicity. Their experiments revealed three key findings:

1. Gender bias, with male HBDs estimated more accurately than female ones. The highest relative error for females regarding shoulder width (7.37% vs. 3.93%). (Figure 2.9)

2. Pose dependency, where shoulder width estimation unexpectedly performed worse in lowered-arm poses. (Figure 2.10)

3. Camera distance robustness, demonstrating that CNNs maintained consistent accuracy across varying distances, challenging prior claims that fixed distances are necessary for reliable estimation. (Figure 2.11)

The study highlights the complexity of HBD estimation, particularly for shoulder width, and underscores the need for diverse training data to mitigate biases. Future work, as noted by the authors, should explore real-world generalization and minimal data requirements for practical deployment.

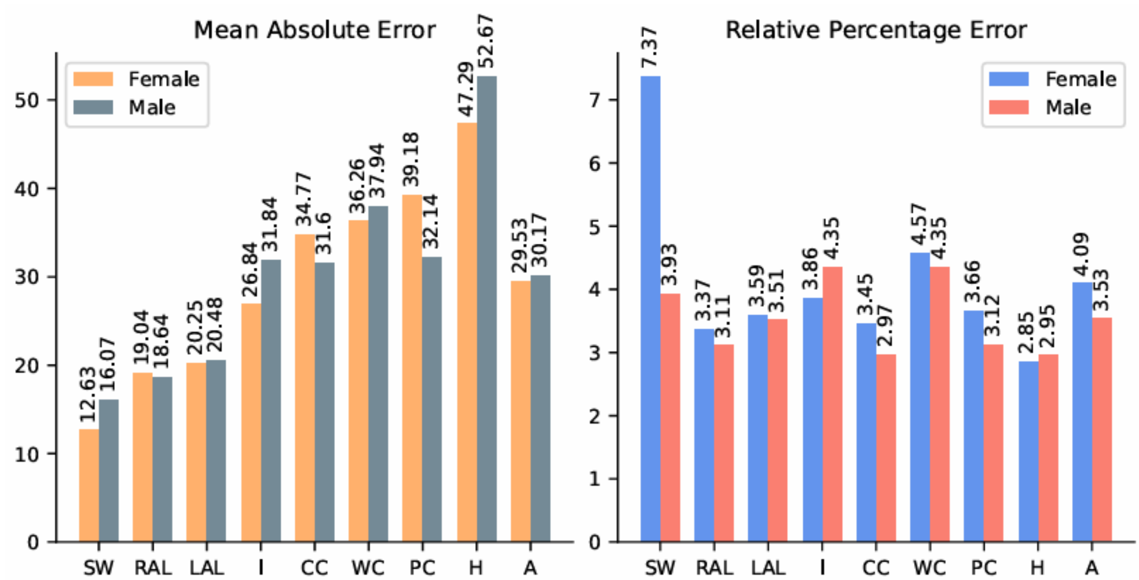


Figure 2.9: Effect of gender on HBDE[IGa24]

While both Skorvankova et al. [DS21] and Zhang [Zha24] address the challenge of non-contact human body measurement estimation using deep learning, their approaches differ significantly. Skorvankova's team focuses on overcoming data

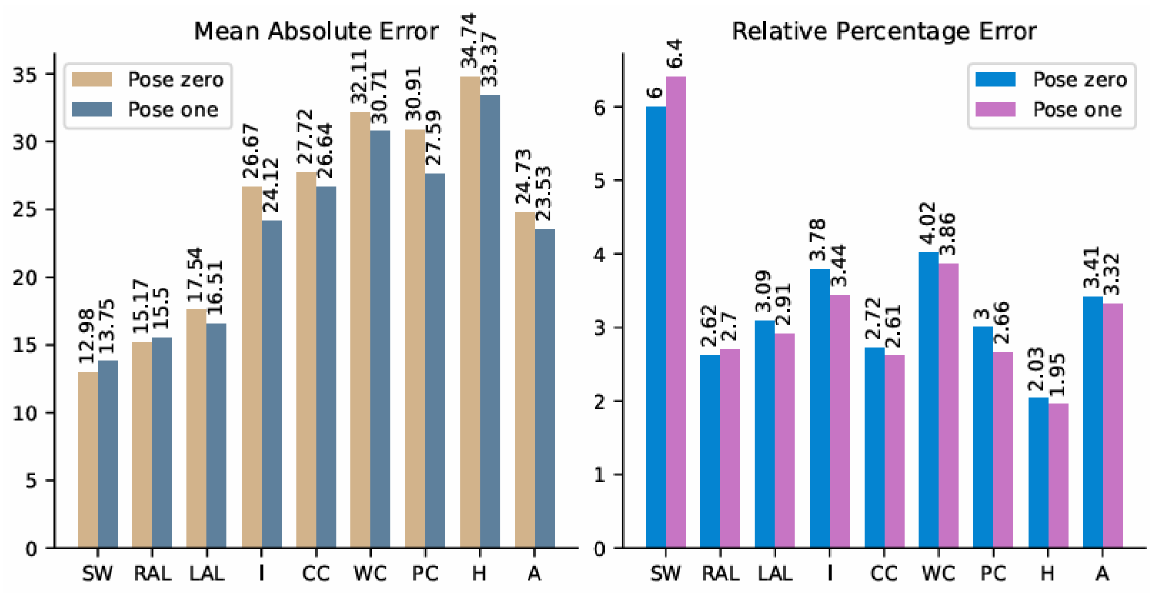


Figure 2.10: Effect of pose on HBDE[IGa24]

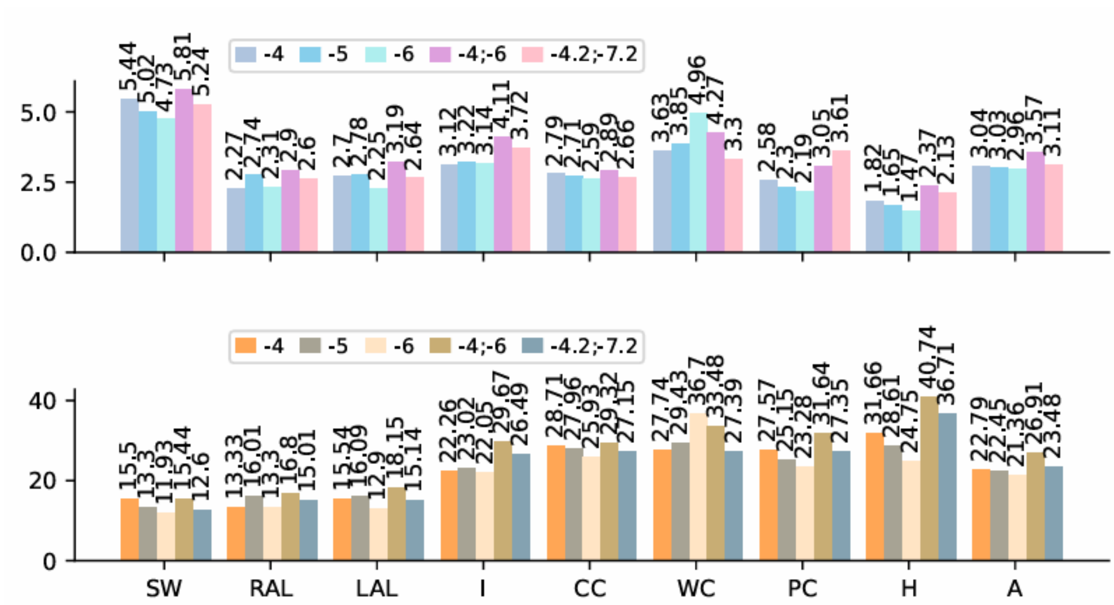


Figure 2.11: Effect of camera distance on HBDE[IGa24]

scarcity by generating a synthetic dataset of 100k 3D body models with annotated measurements, providing detailed definitions for 16 key dimensions (Table 2.1). Their work stands out for its practical pipeline, comparing 2D grayscale images and 3D point clouds as inputs, with clear visualizations of measurement protocols. In contrast, Zhang's review casts a wider net, analysing diverse methods (from CNN-based image processing to ABSS-algorithm hybrids) and benchmarking their performance across datasets like CEASAR and ANSUR II. Notably, Zhang highlights trade-offs: while Neural Anthropometer, inspired by Skorvankova's work, achieves

25.22mm MAE for chest measurements, optimization-heavy models like GBWO-ENN reduce errors to 8.4mm for waist circumferences. Both papers converge on critical gaps: Skorvankova identifies the need for real-world validation of synthetic data, whereas Zhang stresses dataset quality issues and computational constraints in deployment. For practical applications, the annotated dataset and measurement definitions from the first piece of research offer concrete implementation values, while the comparative analysis of model architectures from the latter provides a strategic roadmap for selecting approaches. Together, these studies highlight a key insight: obtaining accurate measurements requires not only high-quality data, but also innovative modelling techniques. Striking this balance is especially vital for precision-dependent applications.

A specific use of estimating body sizes from images was presented by Jin et al. [JGZ+23] in the context of automating pants pattern design. By extracting silhouettes from front and side view images of the body, their system is able to identify key body landmarks (e.g., waist, hip, thigh) and estimate measurements such as heights, widths, and depths. A set of rules link these measurements to pattern lines, enabling the generation of customized designs through adjustments to certain features. The approach combines basic pattern templates with style parameters, allowing modifications like high-waisted cuts or tapered legs. Testing showed predicted measurements had average errors under 2 cm for most body regions. Sample pants produced in this manner were evaluated both by experts for fit at key points and by wearers for overall comfort of the garment. As a result, the method demonstrates potential for reducing manual effort in clothing customization while maintaining fit accuracy.

2.3 Digital Tools for Textile Crafts

"Automatic Crochet Pattern Generation from 2D Sketching" by Nakjan et al. [NRP18] presents a system tailored for novice crafters to design Amigurumi dolls through 2D sketching. The tool detects hand-drawn shapes (spheres, cylinders, tear drops) and converts them into 3D models aligned with Amigurumi aesthetics, emphasizing symmetry and proportional adjustments such as enlarging the head-to-body ratio to achieve a "cute" style. Users sketch components (e.g., head, limbs), and the system automatically generates symmetrical counterparts, reducing manual effort. The 3D model is translated into a crochet pattern using stitch formulas that account for shape type and size, though limitations arise with small components (e.g., tight magic rings). Evaluations with 12 participants revealed rapid design times (2–7 minutes) and general alignment between generated patterns and final crocheted dolls.

Body measurement	Definition
Head circumference	circumference taken on the Y-axis at the level in the middle between the head skeleton joint and the top of the head (the intersection plane is slightly rotated along X-axis to match the natural head posture)
Neck circumference	circumference taken at the Y-axis level in 1/3 distance between the neck joint and the head joint (the intersection plane is slightly rotated along X-axis to match the natural posture)
Shoulder-to-shoulder	distance between left and right shoulder skeleton joint
Arm span	distance between the left and right fingertip in T-pose (the X-axis range of the model)
Shoulder-to-wrist	distance between the shoulder and the wrist joint (sleeve length)
Torso length	distance between the neck and the pelvis joint
Bicep circumference	circumference taken using an intersection plane which normal is perpendicular to X-axis, at the X coordinate in the middle between the shoulder and the elbow joint
Wrist circumference	circumference taken using an intersection plane which normal is perpendicular to X-axis, at the X coordinate of the wrist joint
Chest circumference	circumference taken at the Y-axis level of the maximal intersection of a model and the mesh signature within the chest region, constrained by axilla and the chest (upper spine) joint
Waist circumference	circumference taken at the Y-axis level of the minimal intersection of a model and the mesh signature within the waist region – around the natural waist line (mid-spine joint); the region is scaled relative to the model stature
Pelvis circumference	circumference taken at the Y-axis level of the maximal intersection of a model and the mesh signature within the pelvis region, constrained by the pelvis joint and hip joint
Leg length	distance between the pelvis and ankle joint
Inner leg length	distance between the crotch and the ankle joint (crotch height); while the Y coordinate being incremented, the crotch is detected in the first iteration after having a single intersection with the mesh signature, instead of two distinct intersections (the first intersection above legs)
Thigh circumference	circumference taken at the Y-axis level in the middle between the hip and the knee joint
Knee circumference	circumference taken at the Y coordinate of the knee joint
Calf length	distance between the knee joint and the ankle joint

Table 2.1: "Definition of annotated anthropometric body measurements."[DS21]

However, user feedback highlighted challenges in sketch recognition precision and a preference for drag-and-drop features over freehand drawing. The work demonstrates the feasibility of bridging 2D design intent with machine-readable crochet instructions but underscores gaps in accommodating advanced techniques (e.g., overlapping stitches) and user-centric flexibility. This aligns with broader efforts to democratize pattern creation while highlighting unresolved usability trade-offs.

2.4 Synthesis and Research Gap

Current advancements in computational pattern generation, body measurement extraction, and digital textile tools demonstrate significant progress in automating craft processes. Systems like stitch meshes [GLNM20] and graph-based languages [SRLH22] offer structured ways to encode crochet patterns, while parametric models (e.g., SMPL) and image-based measurement pipelines [JGZ+23] simplify sizing adaptation. However, these innovations remain fragmented. For instance, pattern generators like Capunaman et al.'s framework [CBG17] excel at translating 3D shapes into stitch rules but lack mechanisms to incorporate real-world variables such as yarn tension or user-preferred styles. Similarly, mobile body-scanning apps [IGa24] provide precise measurements but do not interface with pattern editors, leaving crafters to manually reconcile sizing data with design intent.

A critical gap lies in the disconnect between technical precision and practical usability. While graph-based editors [SRLH22] enable advanced customization, their reliance on abstract notation creates barriers for novices. Conversely, beginner-friendly tools often oversimplify pattern adaptation, neglecting nuanced factors like fabric drape or stretch. Furthermore, existing solutions rely heavily on synthetic or homogenous datasets (e.g., SMPL models [TM22]), risking poor generalization to diverse body types and materials. For example, Jin et al.'s pants pattern system [JGZ+23] achieves high accuracy in controlled settings but struggles with occluded poses common in user-submitted photos.

The absence of end-to-end integration is equally limiting. No current framework unifies measurement estimation, swatch-based calibration, and dynamic pattern resizing into a single workflow. This forces crafters to either do all the work and computations by hand, or to juggle disjointed tools—estimating measurements in one app, calculating gauge in another, and manually adjusting patterns—both processes prone to errors.

In summary, no comprehensive framework exists that integrates accessible body measurement estimation, user-friendly pattern editing, and dynamic adaptation to crafters' materials and techniques. Bridging these gaps requires a unified approach that balances technical precision with practical usability, validated across real-world scenarios.

Chapter 3

Proposed Method

Based on the analysis of existing research and software related to the customisation of crochet patterns, I concluded that the main objective of this work was to develop a system to help crochet artists of all levels customise clothing patterns. After reading about all kinds of specialised scanners or software that could extract body measurements from pictures or scans, I decided that the best way to include something similar in my application was to implement it myself, taking into account the requirements of the app. The proposed method is designed to be scalable and suitable for real-life use in a web-based application.

This chapter details the rationale behind the chosen approach, the dataset preparation and preprocessing pipeline, the architecture and training of the neural model, and how the system integrates into a broader application for personalised crochet pattern generation.

3.1 Motivation for the Approach

Crochet patterns often rely on a lot of body measurements and calculations for customisation. Manually measuring and then using these dimensions can be time consuming, error-prone, and for beginners, quite confusing. Existing 3D body scanning solutions are costly or require complex camera setups. Since crochet is a craft that doesn't require millimetre level precision, a light-weight method that balanced usability, accuracy, and technical feasibility was the best approach. I propose a 2D keypoint-based regression model, where body measurements are inferred from pose landmarks extracted using MediaPipe [LTN+19]. This approach would allow measurement extraction from a single image, which aligns with the goal of minimizing user input.

3.2 Dataset and Preprocessing

The dataset used was found via the Automatic Estimation of Anthropometric Human Body Measurements paper [DS21]. Skorvankova et al. created a large dataset with 100 000 t-pose body models (50 000 female and 50 000 male) in different formats including images, scans and meshes, with corresponding ground truth body measurements provided in CSV format. Out of the available formats, the front-view images made the most sense to use, since the end goal was for the users to upload only one image for the measurement extraction process. At first the preprocessed data consisted of a JSON file with the given measurements for each image, but when this was used for model training, the loss values that resulted from it were really high (going from Epoch 1/100 - Loss: 386567.4643 to Epoch 100/100 - Loss: 14208.9141). In order to try and fix this, before adding the data to the JSON file, each image was processed through MediaPipe's pose detection model to extract 33 key-points (in x, y coordinates), normalized to image dimensions, that were also added to the preprocessed data set. For the sake of speeding up the preprocessing of all 100 000 images, I used ProcessPoolExecutor from the concurrent.futures library to parallelise the process. The final dataset was normalized using min-max scaling, with the min/max values saved to a JSON file for use during inference.

3.3 Model Architecture

The model was implemented using PyTorch [PGM+19] and it is a fully connected feed-forward neural network. The input dimensions consist of 66 key-points and 1 normalized height, while the output dimensions are the 16 body measurements that can also be seen in Table 2.1. The model uses ReLU activations and a decreasing layer size to progressively compress and refine learned features (Figure 3.1).

```
class BodyMeasurementRegressor(nn.Module):
def __init__(self, input_dim=67, output_dim=16):
    super().__init__()
    self.model = nn.Sequential(
        nn.Linear(input_dim, 128),
        nn.ReLU(),
        nn.Linear(128, 64),
        nn.ReLU(),
        nn.Linear(64, 32),
        nn.ReLU(),
        nn.Linear(32, output_dim)
    )
```

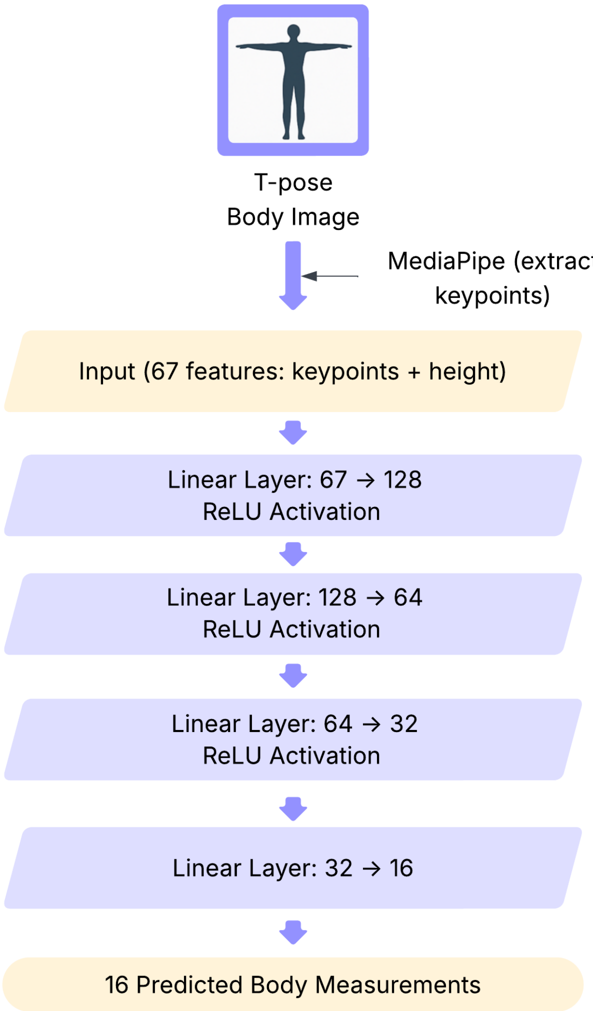


Figure 3.1: The architecture of the proposed body measurement extraction model.

The dataset was split into training and validations subsets in a ration of 80 to 20. The model was trained for 100 epochs using the Adam optimizer and MSE loss. A batch size of 64 was used to balance performance and memory efficiency. After training both the model weights and the normalization scalers were saved. Training loss consistently decreased across epochs and a learning curve was plotted to verify convergence. This can be seen in Figure 3.2.

During inference, a user uploads a front-facing image and inputs their actual height in centimetres. The image is processed via MediaPipe to extract normalised key-points. The height is appended to the flattened key-point vactor and scaled as during training. The model predicts the normalised body measurements, which are then unscaled using the saved min/max values. This process is orchestrated through a Python script callable by a Spring Boot backend, enabling real-time prediction via a React frontend.

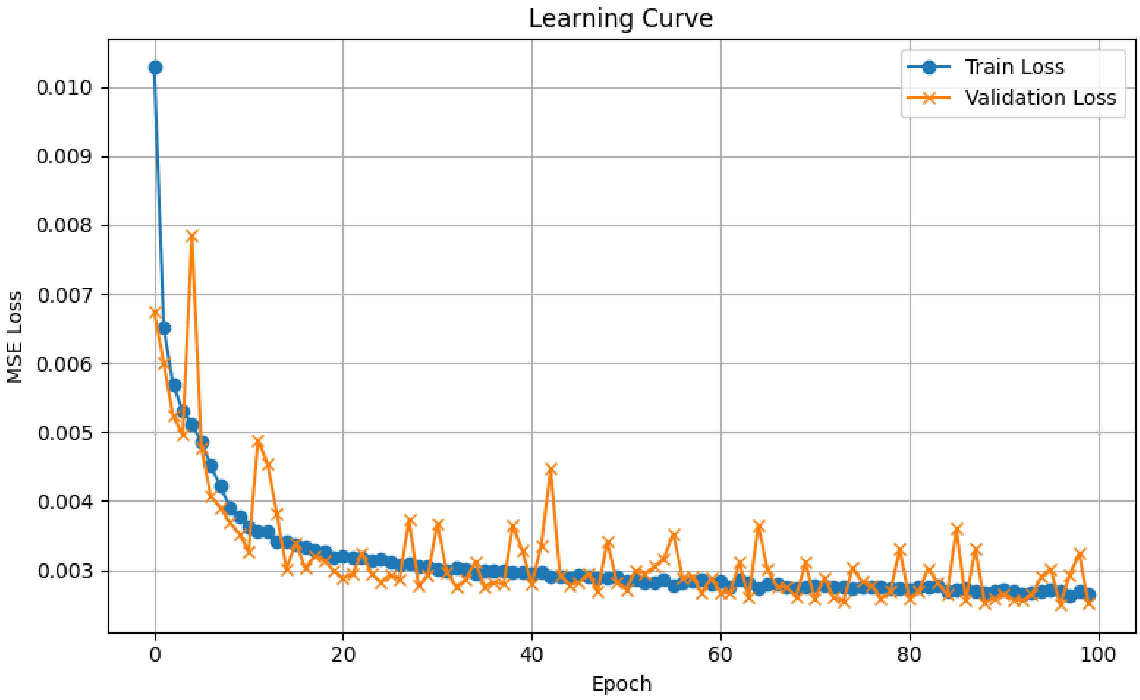


Figure 3.2: A learning curve showing how the model's training and validation loss change over 100 epochs.

3.4 Evaluation

The proposed model was evaluated using two key metrics: The mean absolute error (MAE) and the root mean squared error (RMSE), both measured in centimetres. Table 3.1 summarises the prediction performance for each of the 16 target body measurements. Overall, the model demonstrates strong predictive ability, achieving an average MAE of 1.64 cm and average RMSE of 2.07 cm across all measurements. This level of accuracy is particularly promising for the intended application of resizing crochet patterns, where even moderate errors (e.g., 2-3 cm) can typically be tolerated without impacting fir or wearability. The model performs best on linear measurements, such as arm, leg, or torso length. More complex predictions, like waist and chest circumference, show higher errors due to the challenges of inferring such measurements from only one 2D projection. This could be improved in the future with the use of a dataset that contains both front-view and side-view images of the same subjects.

In summary, the model generalises well across a wide range of measurements and it achieves an efficient balance between simplicity and utility. These measurements form the fooundation for dynamically adapting crochet patterns, which is the core functionality of the broader system. The rest of those features will be presented in the following chapter.

Measurement	**MAE (cm)**	**RMSE (cm)**
Chest circumference	4.14	5.22
Waist circumference	5.00	6.34
Pelvis circumference	3.16	4.00
Neck circumference	1.44	1.83
Biceps circumference	1.76	2.22
Thigh circumference	2.45	3.08
Knee circumference	1.34	1.69
Arm length	0.59	0.73
Leg length	0.69	0.87
Calf length	0.42	0.53
Head circumference	1.07	1.35
Wrist circumference	0.82	1.07
Arm span	0.79	0.91
Shoulders width	0.80	1.02
Torso length	0.76	0.97
Inner leg	1.06	1.35
Average	**1.64**	**2.07**

Table 3.1: Evaluation of body measurement predictions on the validation set. Errors are shown in centimetres.

Chapter 4

Application Development

This chapter documents the end-to-end process of developing the Tailored Crochet application, the core technical artefact of this thesis. Building upon the problem analysis and objectives established in the previous chapters, the structure of this chapter will progress through four critical phases: first, **Requirements and Specifications** define what the system must achieve and how it will be built; second, **Design Analysis** translates these specifications into blueprints and component models; third, **Implementation** details the practical construction of the system using selected technologies and methodologies; finally, **Testing** validates the implementation against initial requirements. Together, these sections provide a transparent account of the application's development lifecycle, demonstrating how theoretical research insights were translated into a functional system.

4.1 Requirements and Specifications

This section defines the functional and non-functional requirements for the Tailored Crochet application, followed by detailed technical specifications derived from these requirements. These came as a result of the analysis of the gap in current crochet and pattern generation related software, which was presented in the ”Synthesis and Research Gap” section of Chapter 2. The figure 4.1, which consists of a simple use case diagram, gives an overview of all the main features that will be available to the user. Each of them will be discussed in the following subsections.

4.1.1 Functional Requirements

Registration:

Before being able to access the main features of the app, each user must create an

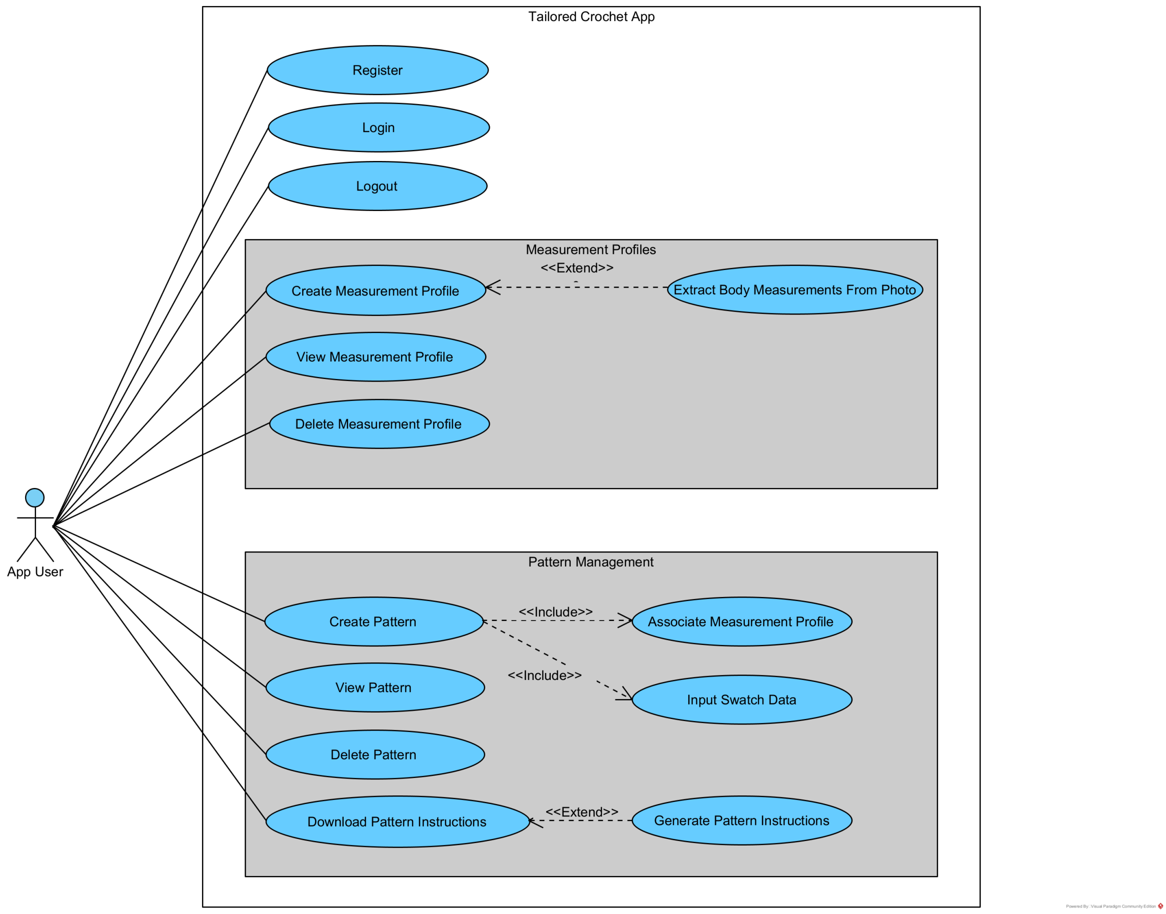


Figure 4.1: Use case diagram illustrating the main features of the application.

account. They will be able to do so by providing a username, an email that has not been used already for an account, and a password. If the information provided is valid, the account will be created and the user will be automatically logged in.

Log in / log out:

A user who has an account must log into the app before being able to access any of the other pages. This is done by providing the email and password. If those are valid, the authentication is done using JWT. The token returned by the server side is then stored in the browser cache memory until the session expires or the user logs out.

Measurement Estimation:

An authenticated user can create several ”measurement profiles” by uploading a front-view picture of themselves or someone else in a T-pose. The image gets processed and the server side returns estimated body measurements based on it. Alongside an image, each profile has a unique name, the height of the person in the image, and additional notes if the user chooses to give them. The user can also delete any of their existing measurement profiles at any time.

Pattern generation:

An authenticated user shall be able to create a custom crochet pattern for an item of clothing. They can choose from the available types of garments and from the different styles of parts that make up the chosen garment. As an example, if they choose to create a pattern for a sweater, they could pick out a simple sweater body, with a bell sleeve design and a ribbed neckline. After the parts of the pattern are decided and a measurement profile is associated to the current selection, the user also needs to crochet a small swatch with the yarn and stitch type they plan on using. Those details are then put into a form before the pattern can be saved. The swatch aids in the pattern measurement adaptation process, providing an even higher level of customisation. Once the user is satisfied with all the choices and details, the pattern can be saved to the personal gallery, which can be accessed on the "Pattern Gallery" page of the website. From that page a pdf with the instructions of the generated pattern can be downloaded.

4.1.2 Non-functional Requirements

Authentication Security:

The user authentication tokens (JWT) shall expire after 24 hours of inactivity.

Data Protection:

Passwords must be stored only after being encrypted. All API endpoints, with the exception of the login and register ones, shall require the request data as well as the generated JWT.

Input Validation:

All user inputs (email, image uploads, form fields) shall be sanitised and validated both on the front-end and the server side to prevent XSS, SQLi, and malicious file uploads.

System Responsiveness:

Core user actions (login, register, measurement profile creation, and pattern generation) shall have UI response times less than 2 seconds under normal load.

Error Handling:

User-facing errors shall provide clear guidance in plain language within the UI.

Data Integrity:

Deletion operations must require explicit confirmation.

4.1.3 Technology Stack

The Tailored Crochet application follows a modular three-tier architecture (Figure 4.2) with clear separation of concerns. This consists of the Presentation Layer - a React-based frontend handling user interactions, the Application Layer - Spring Boot backend managing business logic and API endpoints, and the Data Layer - PostgreSQL database for persistent storage. A unique aspect is the Python integration bridge for measurement estimation. When a user uploads an image, backend receives the image and height data via REST API and invokes the Python script via subprocess call. Python then processes the image using MediaPipe for pose detection and Torch for measurement regression. The results are returned as JSON and measurements are persisted to database. This architecture supports horizontal scalability, with potential to containerise Python components using Docker for production deployment.

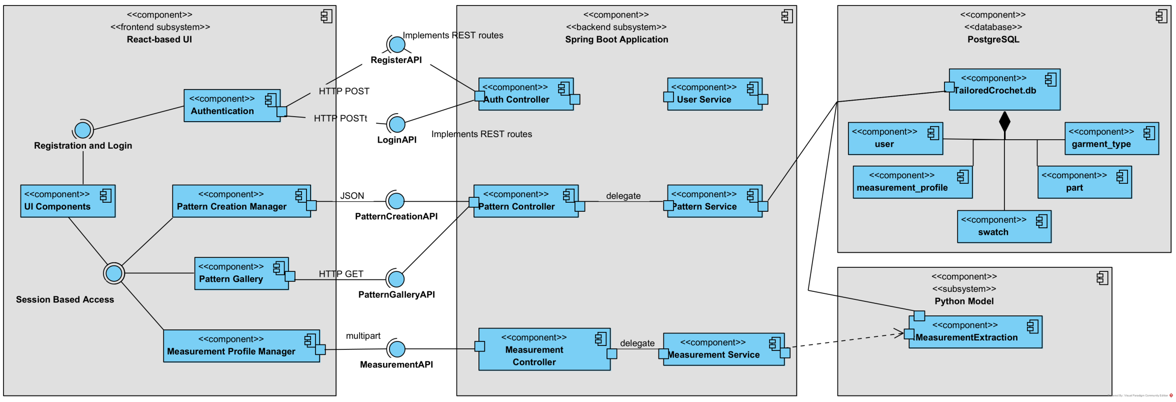


Figure 4.2: Component diagram with a simplified view of the main components of the implemented application.

4.1.4 Data Model

The database schema implements five core entities. These and their relationships are illustrated in Figure 4.3 .

Tables:

1. User: stores authentication credentials (id, username, email, and password after hashing)

2. Measurement Profile: contains body measurements linked to users (17 anthropometric fields)

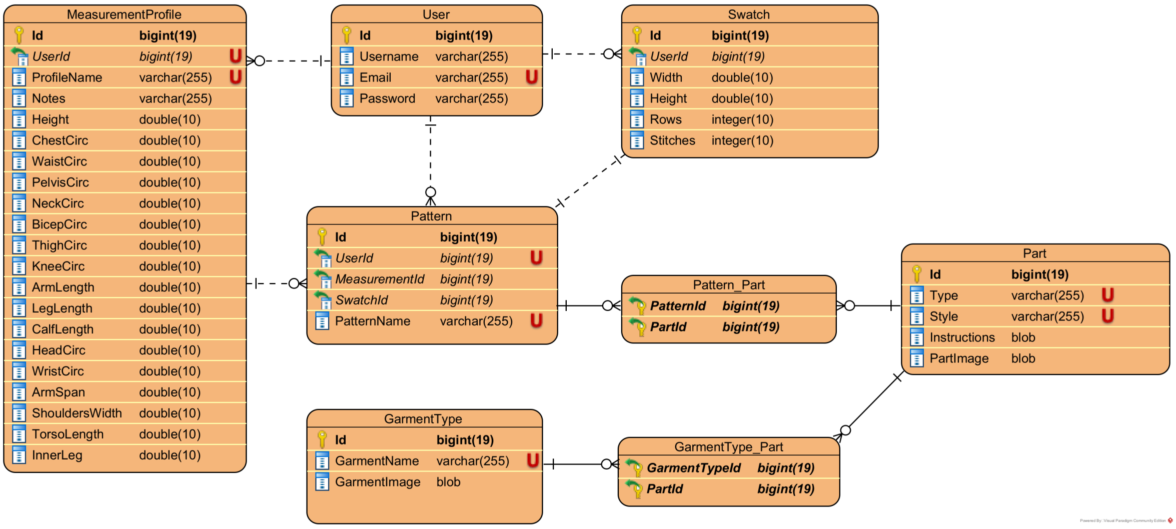


Figure 4.3: Entity-relationship diagram representing the core database schema of the Tailored Crochet application. It inlcudes entities for users, measurement profiles, and pattern components, with appropriate relationships and constraints ensuring data integrity and efficient linkage between the components.

3. Swatch: records user's gauge swatch details (width, height, rows, and stitches)
4. Garment Type: stores the available types for garment patterns (name and image)
5. Part: stores the available garment components, their variations, and the associated crochet instructions
6. Pattern: main pattern repository (garment type, foreign keys to swatches and measurements)
7. Pattern-Part and GarmentType-Part: the tables that map the existing many-to-many relationships

This structure required some constraints to establish an efficient and organised storage strategy. One of them was cascading deletes for user-owned entities to ensure that unnecessary entities are not taking up space in the database if they will not be accessed any more. Another constraint was that measurement fields should be stored as double precision for better accuracy. Lastly, to maintain referential integrity of the database, foreign keys are used whenever the linking of two or more tables was necessary.

4.1.5 Key UI Specifications

The app interface follows a task-oriented workflow with three main journeys, beginning with Measurement Profile Creation, where users encounter an image uploader with strict validation for file types (JPG/PNG) and size limits (¡ 5MB). Before being able to submit the image for processing, users need to provide the profile name, additional notes, and the subject's height through the appropriate input fields. For Pattern Generation, the workflow follows four stages: garment type selection, part configuration (both through selecting the desired options from a gallery-type view), association of a measurement profile, and submission of swatch details via a form. This journey concludes with a ”Save Pattern” action. The Pattern Gallery presents the current user's saved designs in a responsive grid layout, featuring immediate PDF download capabilities for the instructions associated to each pattern, with visual feedback during file generation. Throughout all interfaces, consistent form validation provides real-time error messaging, and success states trigger clear confirmation / rejection notifications to guide users.

4.2 Design analysis

The diagram in Figure 4.1 captures the core features of the software. The main categories of activities the user can perform are related to authentication, measurement estimation, and pattern creation. The user initiates all primary actions using the application interface and the system handles all automated processes.

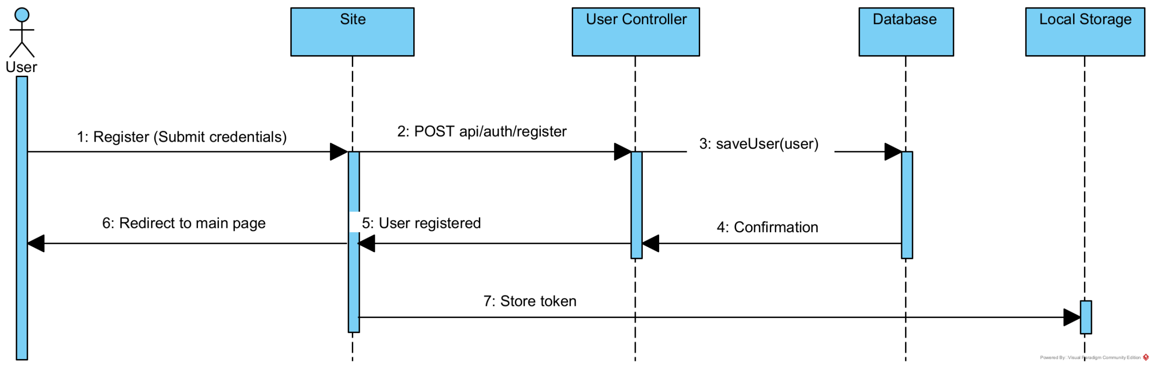


Figure 4.4: Sequence diagram illustrating the registration process, from user input through frontend validation, backend data processing, and secure credential storage in the database.

The registration sequence (Figure 4.4) begins when the user provides a username, email, and password through the frontend form. The frontend validates this input and sends a request to the backend, where the data is validated again. If the provided email is not already in use, the backend hashes the password and stores the credentials in the database. Upon successful registration, a JSON Web Token is generated and returned to the frontend, automatically logging in the user and redirecting them to the main interface.

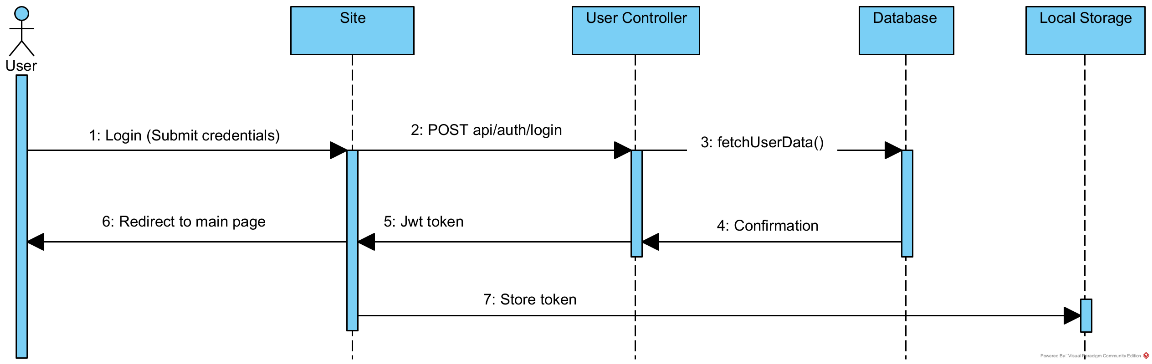


Figure 4.5: Sequence diagram depicting the login process, including credential validation against existing user profiles and token-based session authentication.

During login (Figure 4.5), the user submits their email and password, which are validated first by the frontend and then by the backend. The backend checks if the email exists and whether the provided password matches the hashed password stored in the database. If successful, a JWT is generated and sent back to the frontend, enabling secure management. The token is stored locally, allowing access to the protected areas of the application.

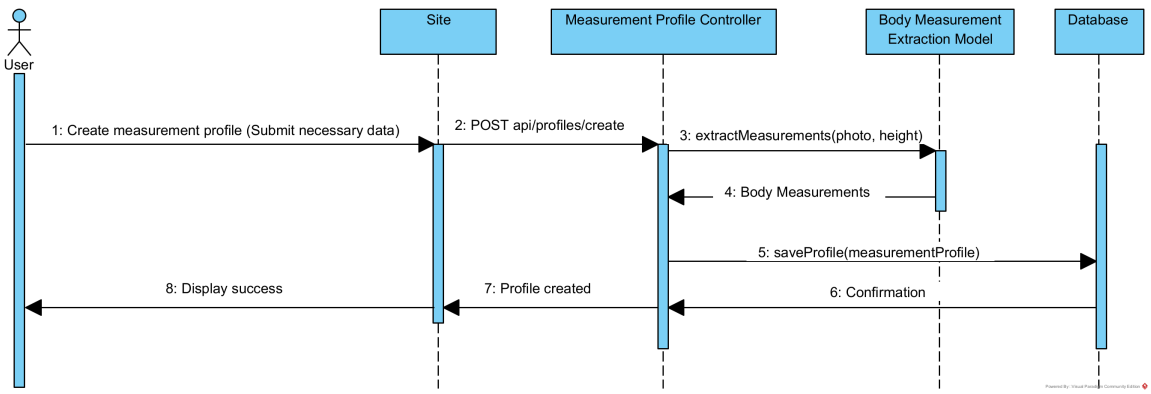


Figure 4.6: Sequence diagram showing the flow of measurement profile creation, including image upload, body measurement extraction, and data persistence.

The sequence diagram in Figure 4.6 illustrates the end-to-end workflow for the creation of measurement profiles. The user uploads a T-pose image and provides the subject's height, and other required information into the form provided in the React interface. Some initial validation is performed before it is transmitted to the backend. Here, the JWT is validated as well, and the service invokes the Python measurement script as a subprocess, passing the image path and height value as arguments. Once the image has been processed, the resulting body measurements are sent back to the frontend and also saved in the database.

This sequence (Figure 4.7) begins when a user chooses a garment, the parts that construct it, and submits pattern details, including swatch metrics and a selected

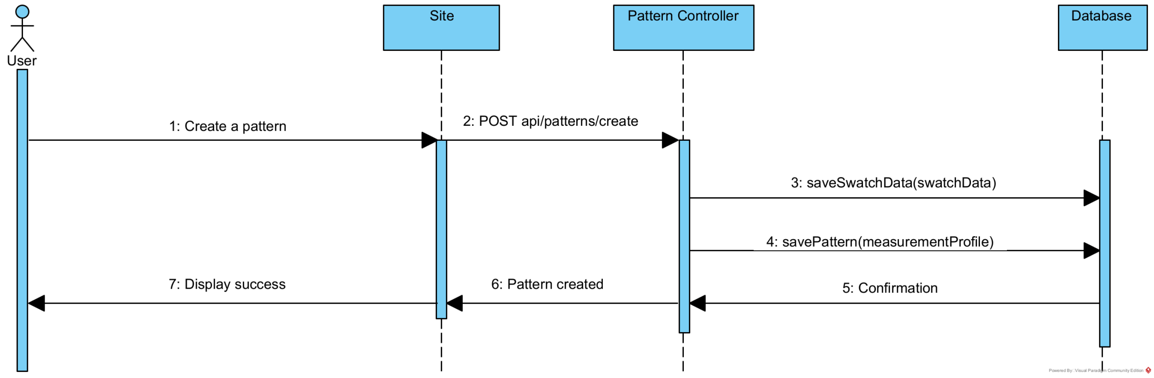


Figure 4.7: Sequence diagram representing the workflow for generating a customized crochet pattern by combining garment options, swatch data, and measurement profiles.

measurement profile. The frontend packages this data and sends it to the backend, where the service layer handles entity creation and association. Once all relationships are validated, the complete pattern is saved in the database and becomes available in the user's pattern gallery.

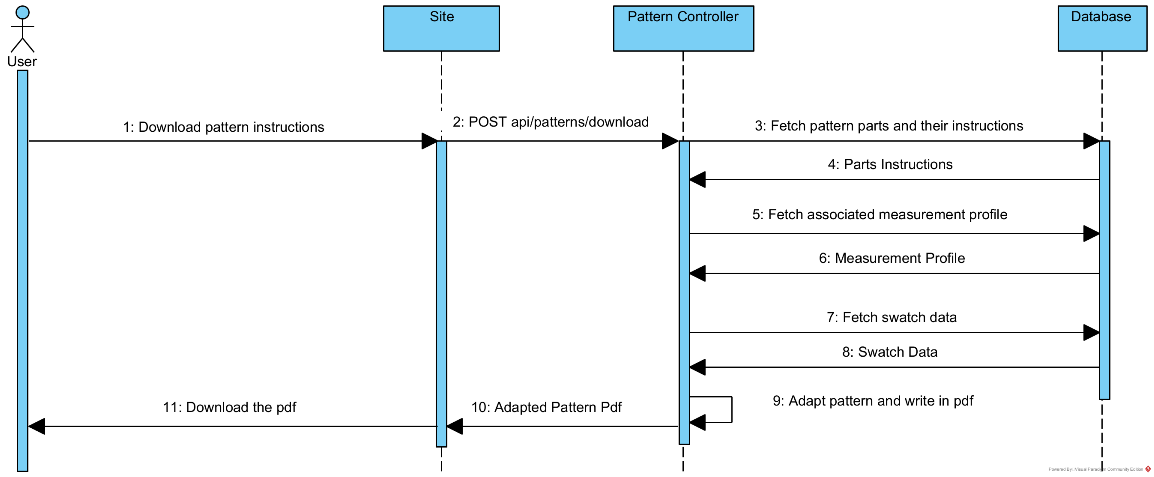


Figure 4.8: Sequence diagram illustrating the process of retrieving and downloading a PDF file containing the generated crochet pattern instructions.

When a user requests to download the instructions for a previously saved pattern(Figure 4.8), the frontend sends a request including the pattern ID. The backend verifies the user's token and retrieves the pattern data from the database. The instructions are then adapted to the associated measurements, taking into account the details of the crochet swatch. A PDF document is generated on-the-fly with the pattern's newly customised details and instructions, which is then sent back to the frontend for download. This enables users to retain offline copies of their customised crochet designs.

4.3 Implementation

This section will follow the step-by-step implementation of all the main features of the Tailored Crochet Web app. All the components referenced in this section can be visualised in the class diagram in Figure ... for an easier understanding of the connections between all of them.

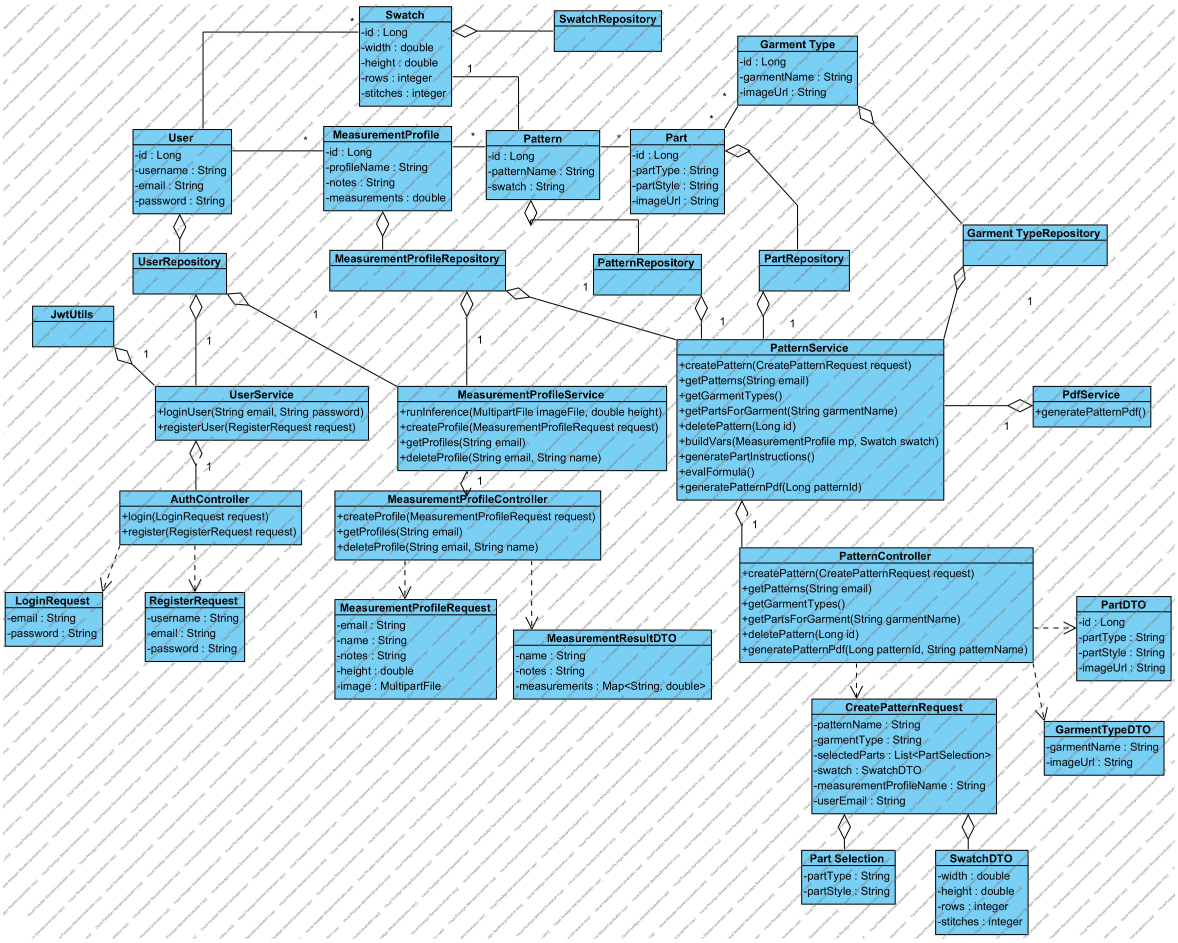


Figure 4.9: Class diagram that illustrates all the modules in the backend of the application.

The authentication system handles user registration and login through a secure Spring Boot backend. When a new user registers, the AuthController component receives their credentials and delegates processing to the UserService, which checks for duplicate emails using the UserRepository before hashing passwords with BCrypt. Successful registration immediately generates a JWT via JwtUtils using HMAC-SHA512 encryption with a 512-bit secret key, automatically logging in the user. For logins, credentials are validated against hashed passwords in the database, with failed attempts triggering specific exceptions (”User not found” or ”Invalid password”). The stateless architecture uses Spring Security’s session management configured in SecurityConfig, where JWTs expire after 24 hours of inactivity. All authentication endpoints (/api/auth/**) permit cross-origin requests from the React frontend while other routes require valid tokens. The process returns an AuthResponse DTO containing the JWT, email, and username for frontend storage in localStorage.

The measurement profile feature enables authenticated users to extract body dimensions from uploaded images. When a user submits a T-pose photo through the React frontend, the MeasurementProfileController receives the image, height, and metadata via multipart form data. The MeasurementProfileService first saves the image to a temporary directory with a UUID filename to prevent collisions, then invokes a Python script as a subprocess using ProcessBuilder, passing the image path and height as arguments. This Python component leverages MediaPipe for pose detection and Torch for regression modeling to compute 17 anthropometric measurements, returning the results as a JSON string through standard output. The Java service parses this JSON using Jackson's ObjectMapper, maps the measurements to the MeasurementProfile entity, and persists the complete profile to PostgreSQL with a foreign key linking it to the user. Error handling captures Python execution failures and invalid input, returning appropriate HTTP status codes.

The pattern creation feature enables users to assemble custom garments through a structured workflow. When a user submits a CreatePatternRequest via the React frontend, the PatternController delegates processing to the PaternService. The service first retrieves the authenticated user by email, then constructs the pattern entity through three key associations: swatch integration, measurement profile linking and part configuration. User-provided gauge data (width, height, row, and stitch numbers) is mapped to a Swatch entity, establishing a one-to-one relationship with the pattern. This captures the crafter's unique tension and material properties. The specified measurement profile is fetched by a name and user ID, ensuring profile ownership. This links anthropometric data to the pattern through a many-to-one relationship. Selected garment components (e.g., ″bell sleeve″, ″ribbed neckline″) are transformed into PatternPart entities with partType and option fields. These are stored as a child collection via many-to-many mapping. The resulting pattern is persisted as a single aggregate using JPA cascading.

The gallery feature provides access to saved patterns through a RESTful interface. When users request their patterns, the PatternService fetches all user-associated patterns from the PatternRepository. Before returning this information to the frontend, it transforms each Pattern entity into a PatternDTO containing: pattern ID, pattern name, garment type, list of parts with their associated options. The React frontend displays the patterns is a gallery type grid, where the instructions for each one can be downloaded in a pdf format.

4.4 Testing

Thorough testing was conducted to ensure the reliability and correctness of both the backend and frontend components of the application. All RESTful API endpoints developed in the Spring Boot backend were tested using Postman. It allowed for manual verification of each endpoint's behaviour under various conditions, including registering or logging in a user, creating and deleting measurement profiles, and creating and deleting patterns. In this way, error handling for invalid or incomplete requests was also tested. The tests confirmed that the backend correctly handles input validation, returns appropriate status codes, and maintains data integrity consistently.

For the frontend developed with React, manual exploratory testing was performed. This involved interactively navigating the user interface to identify bugs or usability issues in real-world scenarios. Key areas tested included navigation flow between view (e.g., pattern list, detail view, add screens), file attachment behaviour, integration with the backend through API calls, and form validation (e.g., login, register, add for measurement profiles and for patterns) to ensure that any input error was clearly displayed for the user. Manual testing was chosen due to the dynamic and visual nature of the application and its relatively small scale, which allowed efficient detection of user experience issues through direct interaction.

When it came to cross-component testing, to ensure full-stack integration, end-to-end tests were performed manually by simulating real user workflows. Future improvements may include the addition of automated tests (e.g. unit tests for React components or integration tests using tools JUnit) to improve the efficiency of regression testing.

4.5 User Experience

The user experience of Tailored Crochet was designed to be intuitive and accessible, guiding users through the entire process of creating custom crochet patterns. First, users must either register as a new user (Figure 4.10), or log in if they already have an account (Figure 4.11). Once authenticated, users can create a measurement profile by uploading a front-facing image of someone in a T-pose, entering the subject's height, a name for the measurement profile, and optionally adding notes (Figure 4.14). The profile is displayed to the user and saved for future use. Once a measurement profile is available, users proceed to the pattern creation page (Figure 4.16). They select a garment type and configure the desired parts using the gallery-style interface (Figure 4.17). After specifying the swatch details and associating the pattern with a measurement profile, users can save the pattern design. All saved patterns are accessible on the pattern gallery page (Figure 4.18), where users can download PDF instructions adapted to the associated measurements. Throughout this process, the interface provides real-time validation and visual feedback (Figure 4.19) to ensure clarity and prevent errors.

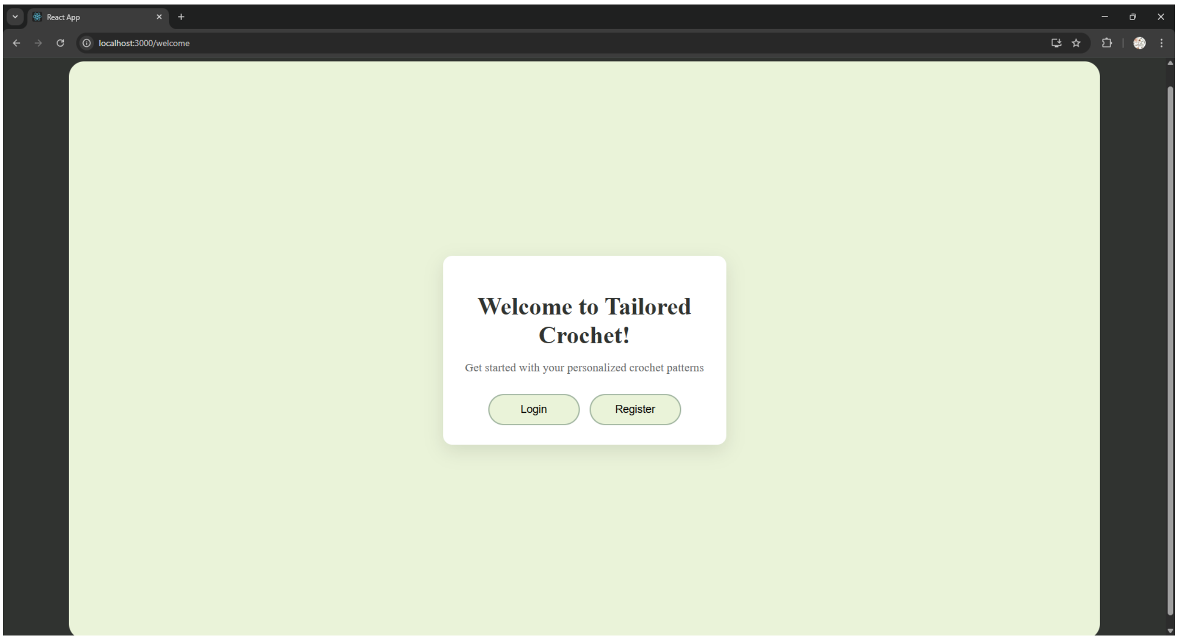


Figure 4.10: Main page that welcomes the user with the option to either register as a new user, or log into the application.

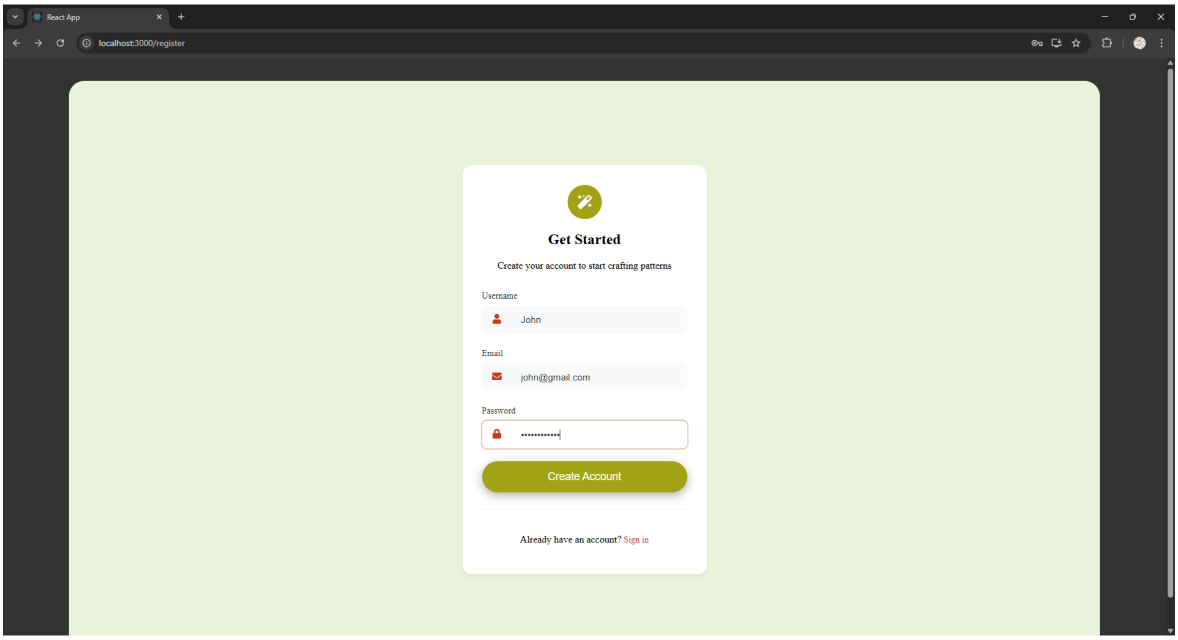


Figure 4.11: Register page with a valid input example.

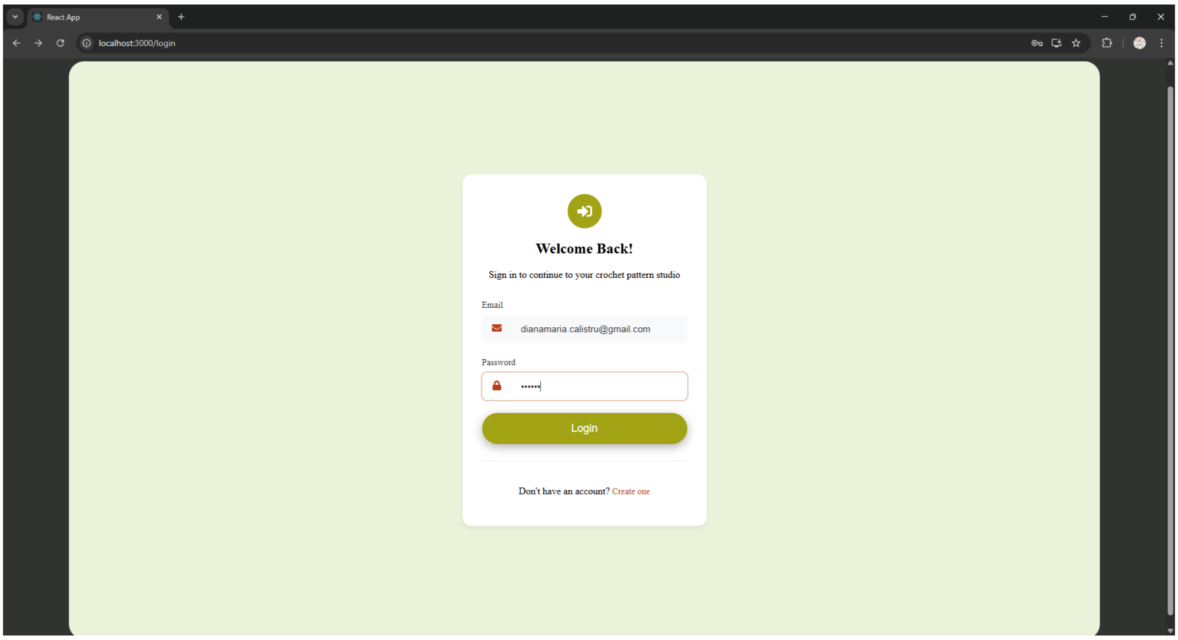


Figure 4.12: Login page with a valid input example.

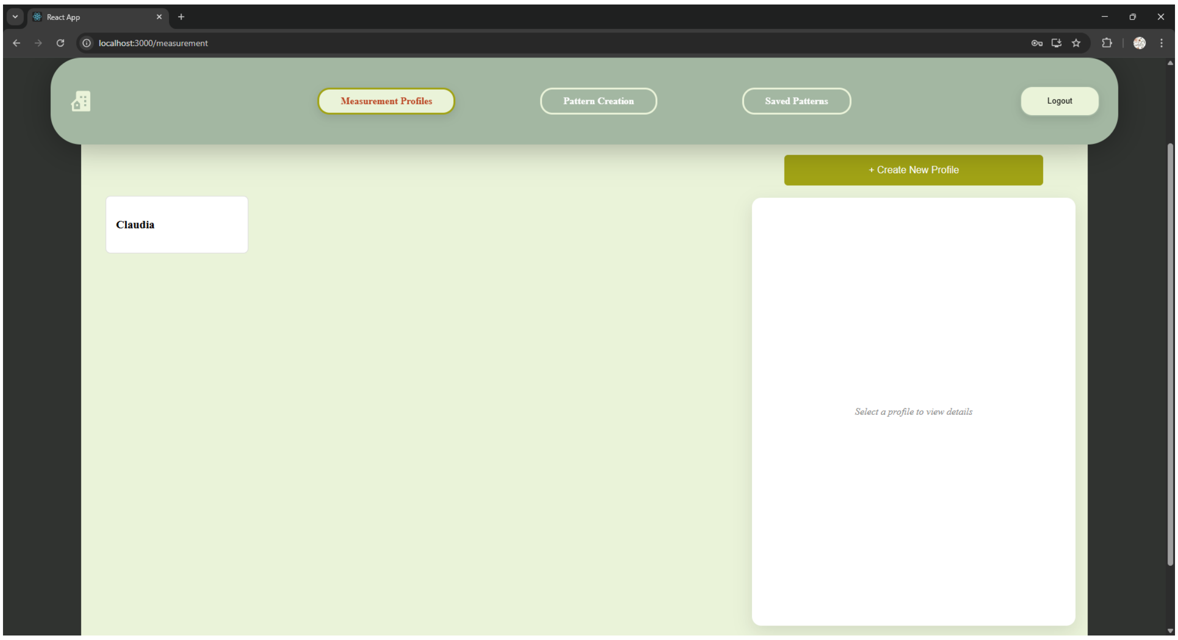


Figure 4.13: Measurement profiles page overview.

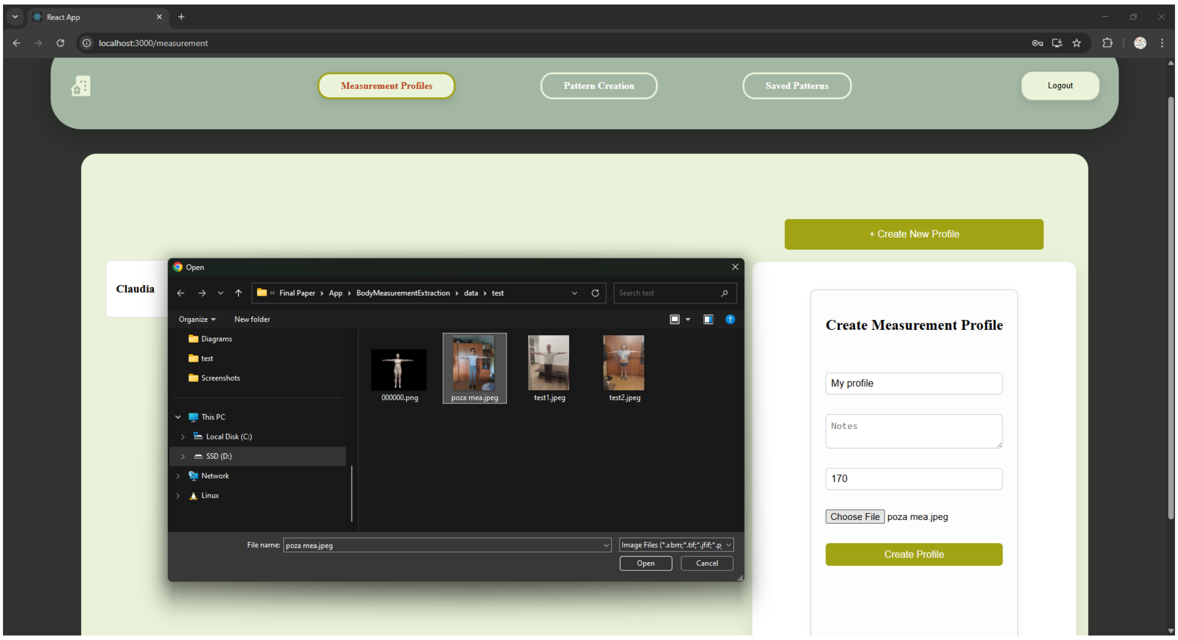


Figure 4.14: Measurement profile creation process with valid data.

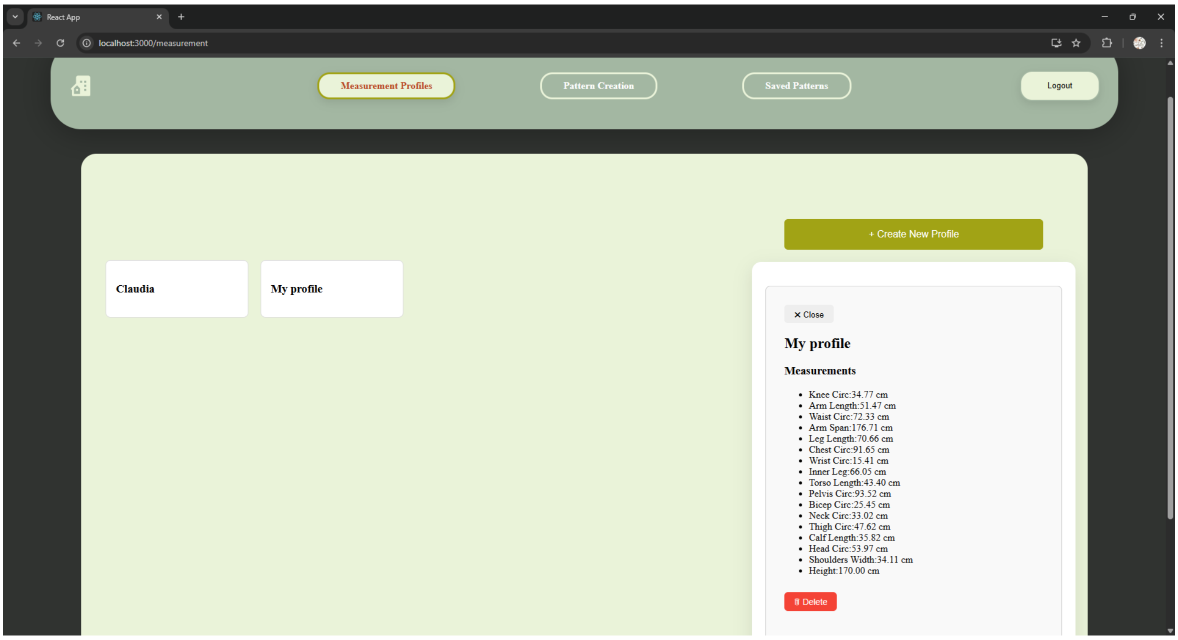


Figure 4.15: Measurement profile resulted from the processing of the image.

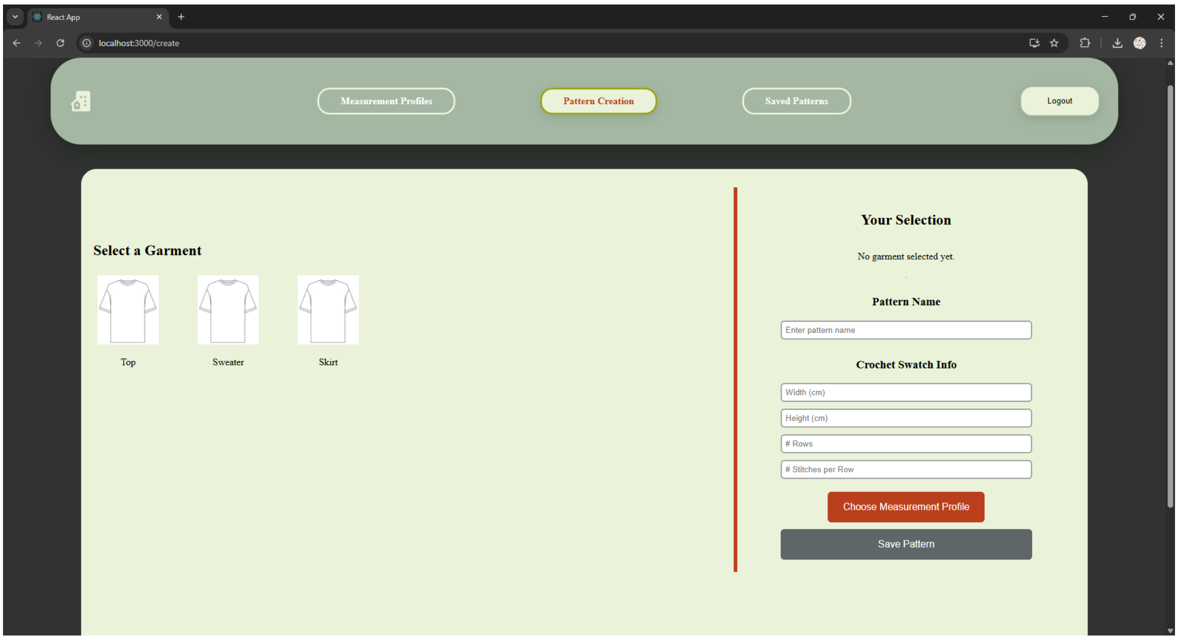


Figure 4.16: Empty pattern creation page.

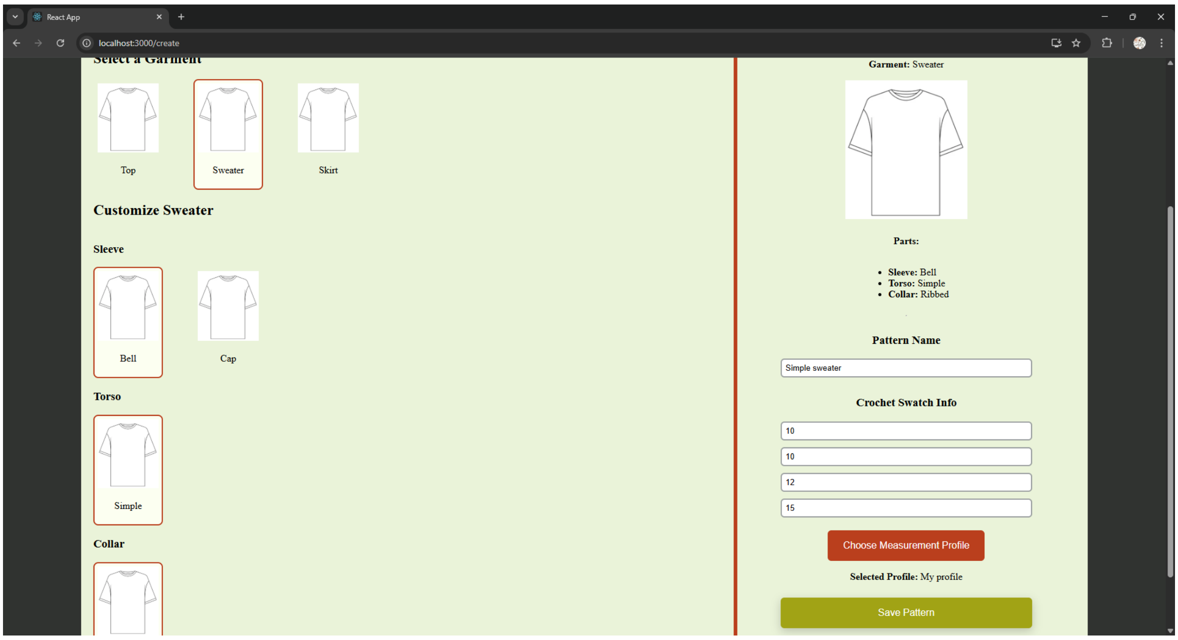


Figure 4.17: Pattern creation example with valid input.

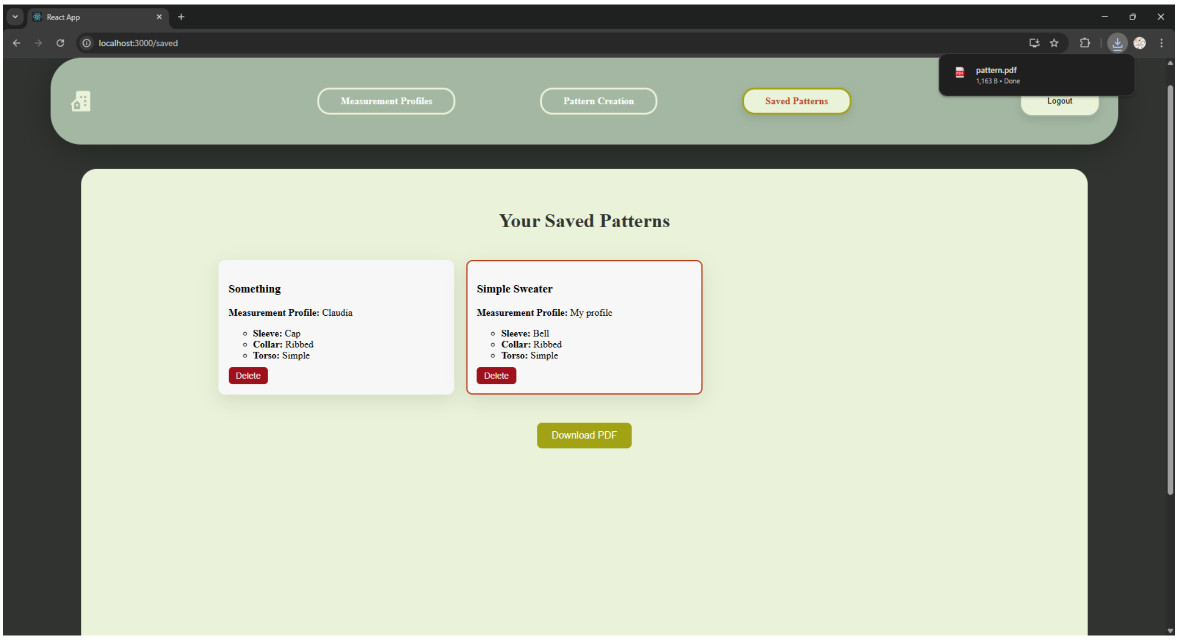


Figure 4.18: Saved patterns page with download example of the selected pattern.

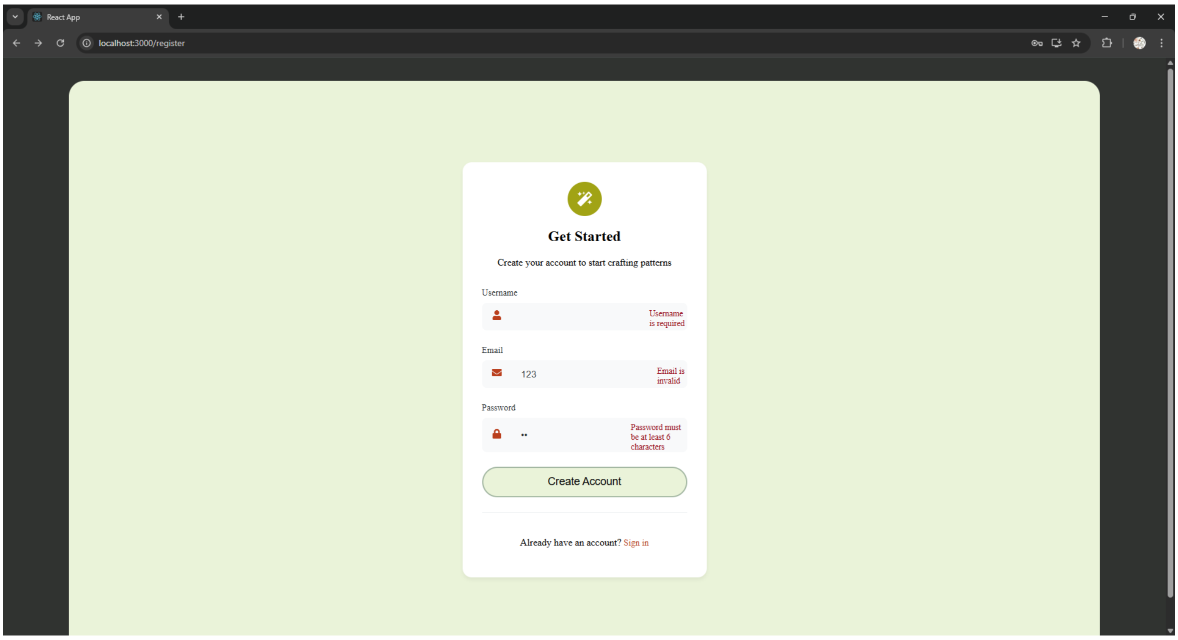


Figure 4.19: Example of an invalid input and the associated error messages.

Chapter 5

Conclusion

This thesis set out to address a persistent challenge in the world of crochet: the difficulty of generating size-customised patterns that consider individual body measurements, material properties, and design preferences. Through the development of this thesis I have presented a digital solution that combines artificial intelligence with pattern generation logic, that could offer both novice and experienced crafters a tool for creating personalised crochet designs.

The system integrates a lightweight 2D key-point-based body measurement estimation model with a dynamic pattern adaptation workflow. The evaluations demonstrated that the AI model achieves a level of precision suitable for crochet applications, with an average mean absolute error of 1.64 cm across 16 key body measurements. Furthermore, the implementation of a web-based application, built using React, Spring Boot, and Python, provided a practical interface for users to generate, save, and download customised patterns.

While the project has achieved its core objectives, several limitations still remain. The body measurement model, for instance relies only on front-view images and does not yet incorporate side-view images, which would add a great deal of accuracy for certain measurements of the body. Additionally, the pattern generation logic currently focuses on basic garment types and could be expanded to support more advanced techniques and styles.

Future work may explore the integration of an AI model that could translate crochet patterns found online to fit the specific template that the app currently uses, so that initial patterns will not have to be written manually. This would allow for a great expansion to the pattern library available in the app.

In conclusion, Tailored Crochet represents a step forward in bridging the gap between traditional textile crafts with and modern computational tools. By uniting AI-driven measurement estimation with dynamic pattern generation, this work contributes toward making personalised crochet design more accessible, efficient, and enjoyable for all levels of crafters.

Bibliography

[CBG17] Ozguc Bertug Capunaman, Cemal Koray Bingol, and Benay Gursoy. Computing stitches and crocheting geometry. In *Computer-Aided Architectural Design Futures*, 2017.

[CBG22] Ozguc Capunaman, C. Koray Bingol, and Benay Gursoy. From stitches to digits and back: Computational crocheting of branching geometries. *Eskişehir Technical University Journal of Science and Technology A - Applied Sciences and Engineering*, 23, 11 2022.

[CZY20] Jiayu Chen, Yueqi Zhong, and Zhicai Yu. Human body shape reconstruction from binary image using convolutional neural network. *Journal of Physics: Conference Series*, 1624(5):052002, oct 2020.

[DS21] Martin Madaras Dana Skorvankova, Adam Riecicky. Automatic estimation of anthropometric human body measurements. *CoRR*, abs/2112.11992, 2021.

[GLNM20] Runbo Guo, Jenny Lin, Vidya Narayanan, and James McCann. Representing crochet with stitch meshes. In *Proceedings of the 5th Annual ACM Symposium on Computational Fabrication*, SCF '20, New York, NY, USA, 2020. Association for Computing Machinery.

[IGa24] Sadia Idrees, Simeon Gill, and Gianpaolo Vignali and. Mobile 3d body scanning applications: a review of contact-free ai body measuring solutions for apparel. *The Journal of The Textile Institute*, 115(7):1161–1172, 2024.

[JGZ$^+$23] Shouning Jin, Bingfei Gu, Beibei Zhang, YUAN-PING Xia, and Huazhou He. Pants design and pattern generation based on body images. *Industria Textila*, 2023.

[JQW$^+$19] Zhongping Ji, Xiao Qi, Yigang Wang, Gang Xu, Peng Du, Xundong Wu, and Qing Wu. Human body shape reconstruction from binary silhouette images. *Computer Aided Geometric Design*, 71, 04 2019.

[KSS+18] Ayelet Karmon, Yoav Sterman, Tom Shaked, Eyal Sheffer, and Shoval Nir. Knitit: a computational tool for design, simulation, and fabrication of multiple structured knits. In *Proceedings of the 2nd Annual ACM Symposium on Computational Fabrication*, SCF '18, New York, NY, USA, 2018. Association for Computing Machinery.

[LTN+19] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. Mediapipe: A framework for building perception pipelines, 2019.

[NRP18] Pikanate Nakjan, Sukanya Ratanotayanon, and Natchayar Porwongsawang. Automatic crochet pattern generation from 2d sketching. In *2018 10th International Conference on Knowledge and Smart Technology (KST)*, pages 170–175, 2018.

[PGM+19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

[PTSZL22] Kundan Sai Prabhu Thota, Sungho Suh, Bo Zhou, and Paul Lukowicz. Estimation of 3d body shape and clothing measurements from frontal- and side-view images. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 2631–2635, 2022.

[SFK23] Jan Storck, Bjarte Feldmann, and Yordan Kyosev. Design tool for automated crocheting of fabrics. *Communications in Development and Assembling of Textile Products*, 4:254–272, 09 2023.

[SRLH22] Klara Seitz, Patrick Rein, Jens Lincke, and Robert Hirschfeld. Digital crochet: Toward a visual language for pattern description. In *Proceedings of the 2022 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*, Onward! 2022, page 48–62, New York, NY, USA, 2022. Association for Computing Machinery.

[TM22] Yansel Gonzalez Tejeda and Helmut A. Mayer. Effect of gender, pose and camera distance on human body dimensions estimation. In *Image Analysis and Processing. ICIAP 2022 Workshops: ICIAP International Workshops, Lecce, Italy, May 23–27, 2022, Revised Selected Papers, Part II*, page 179–190, Berlin, Heidelberg, 2022. Springer-Verlag.

[Zha24] Yichen Zhang. Non-contact human body dimension estimation methods based on deep learning: A critical analysis. *Theoretical and Natural Science*, 52:214–221, 09 2024.

[ZSB18] Elena Zaharieva-Stoyanova and Damyan Beshevliev. Digital representation of crochet symbols sets. *Digital Presentation and Preservation of Cultural and Scientific Heritage*, 8:159–164, 09 2018.

[ZX21] Sizhuo Zhang and Nanfeng Xiao. Detailed 3d human body reconstruction from a single image based on mesh deformation. *IEEE Access*, 9:8595–8603, 2021.