# THE ULTIMATE WARDROBE GUIDE

By (Group 24 of group 1)

|  |  |  |
| --- | --- | --- |
| No. | MEMBERS | MATRICULE |
| 1. | NDASSA NJOYA FILS FAYSSAL (Group leader) | ICTU20233915 |
| 2. | FUTSEU KENMOGNE JUNIOR ERWAN | ICTU20234174 |
| 3. | MBEY JOSEPH JUNIOR | ICTU20233903 |
| 4. | MAKUETE LEKOGNIA MARIE MICHELLE | ICTU20234486 |
| 5. | MPODE NGWESE AMY-COURTNEY | ICTU20233731 |

**Table of Contents**

**1. Executive Summary**

**2. Introduction**

**2.1. Purpose and Scope**

**2.2. Intended Audience**

**2.3. Document Conventions**

**2.4. References**

**3. System Overview**

**3.1. Business Context**

**3.2. Key Stakeholders**

**3.3. System Goals and Objectives**

**4. Architectural Design**

**4.1. System Architecture**

**4.2. Component Overview**

**4.3. Data Model**

**4.4. Technology Stack**

**4.5. Security Considerations**

**5. Functional Specifications**

**5.1. Use Case Descriptions**

**5.2. User Interface (UI) Overview**

**5.3. Functional Modules**

**5.3.1. Wardrobe Management Module**

**5.3.2. Outfit Generation Module**

**5.3.3. Styling Advice Module**

**5.3.4. Image Management Module**

**5.4. Data Flow Diagrams**

**6. Non-Functional Requirements**

**7. Future Enhancements**

**8. Conclusion**

**1. Executive Summary**

WardrobeApp is a desktop application designed to revolutionize personal wardrobe management through digital innovation. This comprehensive solution enables users to catalog their clothing inventory, generate intelligent outfit combinations, and receive personalized style recommendations based on seasonal trends and personal preferences.

The application features:

- Intuitive clothing item management (add/edit/delete)

- Smart outfit generation algorithms

- Virtual mannequin visualization

- Seasonal fashion advice engine

- Responsive user interface

Built with Java Swing, the application employs a modular architecture that ensures maintainability and future scalability. This document serves as the definitive reference for technical managers, developers, and stakeholders, providing complete system specifications from high-level architecture to detailed functional requirements.

**2. Introduction**

**2.1 Purpose and Scope**

**Purpose:**

This document provides a complete technical blueprint for WardrobeApp, detailing system architecture, functional specifications, and design considerations to guide development and facilitate stakeholder alignment.

**Scope:**

- Comprehensive system architecture and component interactions

- Detailed functional specifications for all modules

- Data model and storage architecture

- User interface design principles

- Security and performance considerations

**Exclusions:**

- Implementation-level code details

- Third-party API specifications

- Graphic design assets

**2.2 Intended Audience**

Primary stakeholders include:

1. **Technical Managers:** Overseeing development and implementation

2. **Software Developers:** Implementing system components

3. **Product Owners:** Defining feature priorities

4. **Business Stakeholders:** Evaluating business value

**2.3 Document Conventions**

- **Bold text:** UI elements (e.g., **Upload Panel**)

- `**Monospace**`: Code references (e.g., `**ClothingItem`** class)

- **Italics**: Important notes or emphasis

- Numbered lists: Sequential processes

- Bullet points: Feature sets or options

**3. System Overview**

**3.1 Business Context**

The modern fashion landscape faces critical challenges:

- 92 million tons of annual textile waste (Ellen MacArthur Foundation)

- 60% increased clothing purchases compared to 2000 (World Resources Institute)

- 30% wardrobe underutilization (Fashion Revolution Report)

**WardrobeApp addresses these issues by**:

- Digitalizing wardrobe inventory

- Maximizing existing clothing utilization

- Reducing impulse purchases through visualization

**3.2 Key Stakeholders**

| Stakeholder | Interest |

|-------------|----------|

| End Users | Wardrobe organization, outfit suggestions |

| Fashion Retailers | Potential integration for smart shopping |

| Sustainability Advocates | Waste reduction metrics |

| Developers | System maintainability |

**3.3 System Goals and Objectives**

**Strategic Objectives:**

1. Achieve 90% wardrobe utilization through smart suggestions

2. Reduce clothing purchases by 30% through visualization

3. Maintain sub-second response time for core operations

**Key Performance Indicators:**

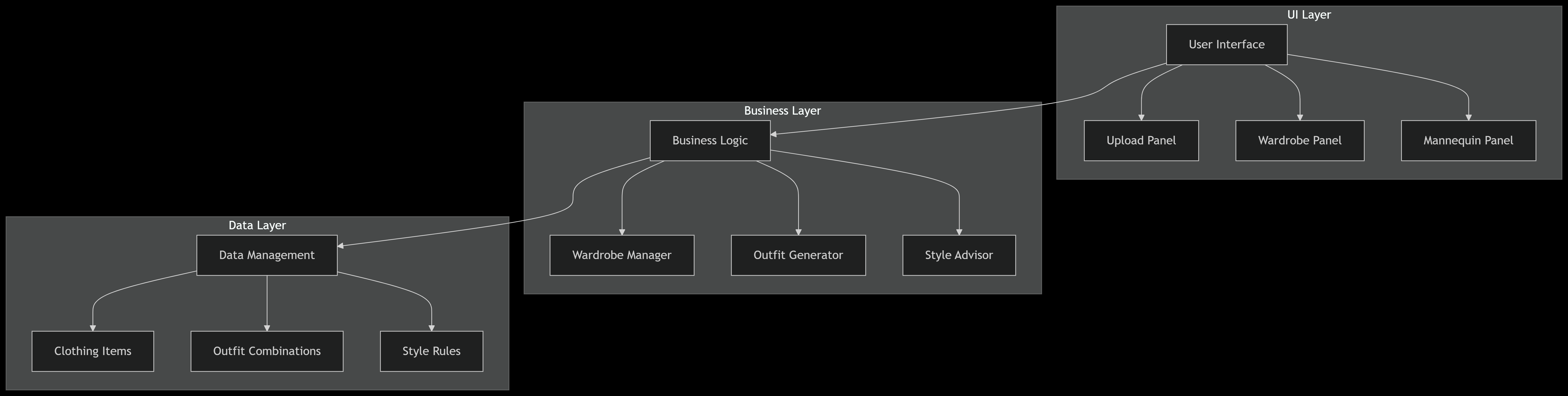
- 95% successful outfit generation rate

- <500ms image loading time

- 90% user satisfaction in outfit recommendations

**4. Architectural Design**

**4.1 System Architecture**



**Three-Tier Architecture:**

1. **Presentation Layer:** Java Swing UI components

2. **Business Logic Layer:** Outfit generation algorithms

3. **Data Layer:** In-memory collections with JSON serialization

**4.2 Component Overview**

**Core Components:**

1. **Wardrobe Manager:** CRUD operations for clothing items

2. **Outfit Engine:** Combination generator with season filters

3. **Style Advisor:** Color theory and trend analysis

4. **Image Processor:**Thumbnail generation and caching

**4.3 Data Model**

```

classDiagram

class ClothingItem {

+String name

+String category

+String size

+String color

+String season

+ImageIcon image

+toString()

}

class Outfit {

+List<ClothingItem> items

+Date createdDate

+saveOutfit()

}

```

**4.4 Technology Stack**

| Component | Technology |

|-----------|------------|

| Language | Java 11 |

| UI Framework | Swing |

| Image Processing | Java AWT |

| Data Storage | JSON (future SQLite) |

| Build Tool | Maven |

**4.5 Security Considerations**

**Security Measures:**

- Local data storage only

- Future encryption for sensitive data

- Secure image handling protocols

**5. Functional Specifications**

**5.1 Use Case Descriptions**

**Use Case UC-01: Add Clothing Item**

1. **Actor:** User

2. **Preconditions:** Application running

3. **Main Flow:**

a. Navigate to Upload Panel

b. Enter item details

c. Upload image

d. Confirm addition

**4.Postconditions:** Item appears in wardrobe

**Use Case UC-02:** **Generate Outfit**

1. **Actor:** System

2. **Trigger:**User request

3. **Algorithm:**

```

java

public List<ClothingItem> generateOutfit(String season) {

return wardrobe.stream()

.filter(item -> item.season.equals(season))

.collect(Collectors.toList());

}

```

**5.2 User Interface Overview**

**Main Window Components:**

1. **Navigation Panel:** Quick access to features

2. **Wardrobe Viewer:** Grid display with filtering

3.**Mannequin Canvas:** Outfit visualization area

**UI Specifications:**

- Minimum resolution: 1024×768

- Color scheme: Neutral tones with seasonal accents

- Font family: Sans-serif for readability

**5.3 Functional Modules**

**5.3.1 Wardrobe Management Module**

**Key Methods**:

- `addItem(ClothingItem item)`: Validates and stores new items

- `deleteItem(String itemId)`: Removes with confirmation

- `getByCategory(String category)`: Returns filtered lists

**5.3.2 Outfit Generation Module**

**Algorithm Features:**

- Color harmony scoring

- Seasonal appropriateness

- Occasion-based filtering

**5.3.3 Styling Advice Module**

**Advice Categories:**

1. Seasonal trends

2. Color combinations

3. Occasion-specific guidance

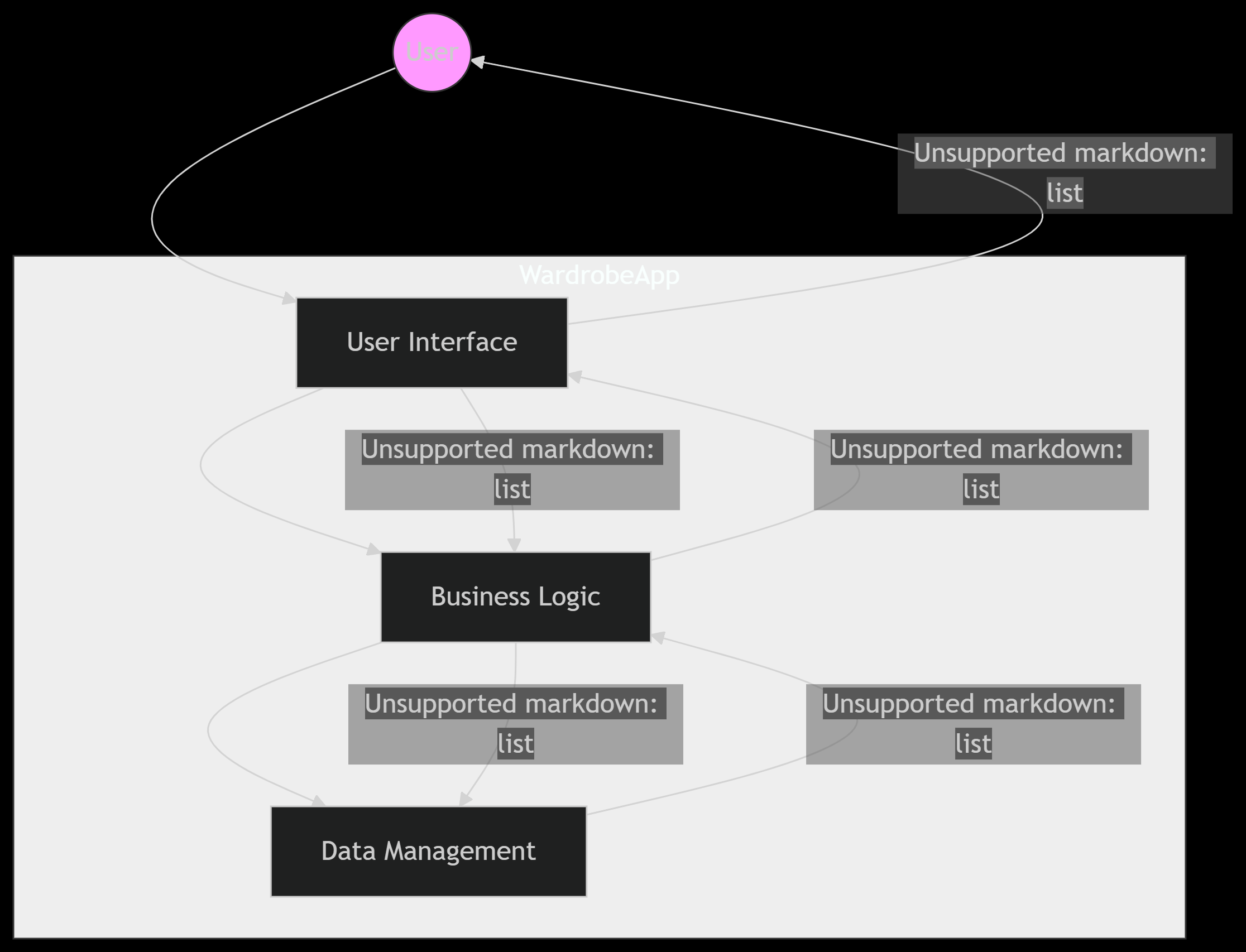
**5.3.4 Image Management Module**

**Features:**

- Auto-resizing to 300×300 thumbnails

- Local caching system

- Format support: JPG, PNG

**5.4 Data Flow Diagrams**

**Process Flow:**

1. User input → Validation → Storage

2. Generation request → Algorithm → Visualization

**6. Non-Functional Requirements**

**Performance:**

- <1s response time for wardrobe operations

- Support 500+ clothing items

**Reliability:**

- 99.9% uptime

- Auto-recovery from crashes

**Usability:**

- <5 minute learning curve

- ADA-compliant accessibility

1. **Future Enhancements**

1. **Mobile Integration:** Companion app synchronization

2. **AI Stylist:** Machine learning recommendations

3. **Retail Integration:** Smart shopping suggestions

**8. Conclusion**

This document presents a comprehensive design for WardrobeApp, establishing a robust foundation for development. The modular architecture ensures adaptability for future enhancements while meeting current user needs for digital wardrobe management. By combining technical precision with fashion intelligence, the system delivers sustainable value to users while promoting responsible consumption practices.