Software Design Specification

Smart Designing Assistant

Project Code: CS-IDA-2025

Internal Advisor: Dr.Faisal Shehzad

Project Manager: Dr. Muhammad Ilyas

Project Team:

Badar Munir - BSCS51F21S095
Hafsa Hadi - BSCS51F22S001
Sana Jabbar- BSCS51F22S030

Submission Date: 15th December 2025

Faisal

Dr.Hafiz Muhammad Faisal Shehezad
Associate Professor
Unversity of Sargodha

_____

Project Manager's Signature

Document Information

Category	Information
Customer	Department of Computer Science
Project	Smart Designing Assistant
Document	Software Design Specification
Document Version	1.0
Identifier	IDA-CS-2025
Status	Draft
Author(s)	Badar Munir Hafsa Hadi Sana Jabbar
Approver(s)	PM
Issue Date	Sept. 15, 2025
Document Location	
Distribution	1. Dr. Faisal Shehzad 2. Dr. Muhammad Ilyas 3. Project Office

Definition of Terms, Acronyms and Abbreviations

Terms/ Acronym	Description Term
AI	Artificial Intelligence enables the system to analyze images and generate smart interior design recommendations.
ML	Machine Learning: used to train models that detect interior design styles from images.
MobileNetV2	A lightweight convolutional neural network used for interior style classification.
OpenCV	Open-Source Computer Vision Library is used for image preprocessing and analysis.
API	Application Programming Interface; allows communication between the front-end and back-end services.
GUI	Graphical User Interface: the visual interface through which users upload images and view results.
UML	Unified Modeling Language; used to design and document system architecture and workflows.

Table of Contents

`

1.1 Purpose of Document

This document defines the detailed design for the Interior Design Advisor system. It is intended for developers, testers, students and the project advisor. It specifies architecture, module decomposition, class and sequence designs, data models, GUI specifications, and design strategies driving implementation. The project follows an Object-Oriented Design methodology using UML diagrams.

1.2 Project Overview

Interior Design Advisor is a smart application that helps users get interior design suggestions by uploading a room image. The system analyzes the image using AI and image processing, detects room style and colors, and provides design recommendations such as matching styles and color palettes.

1.3 Scope

System Will:

- Allow users to upload room images
- Detect interior design style using ML
- Extract dominant colors from images
- Provide smart design recommendations

System Will Not:

- Create 3D room models
- Provide real-time AR visualization

2. Design Considerations:

Before producing detailed modules, we address issues that affect the design:

- **Performance vs. Cost:**

Use MobileNetV2 for reasonable accuracy and fast
inference on CPU. GPU optional for training.

- **Extensibility:**

The system must permit new styles, palette heuristics, and sample images without changing core code (data-driven rules).

- **Failure Handling:**

Error reporting and retriable job architecture for long-running tasks.

Usability: Minimal UI steps → upload image → view results. Provide clear explanations for suggestions.

2.1 Assumptions and Dependencies:

- Users have internet access
- Images are clear and supported formats (JPG, PNG)
- ML model (MobileNetV2) is pre-trained
- System depends on Python, ML libraries, and database services

2.2 Risks and Volatile Areas

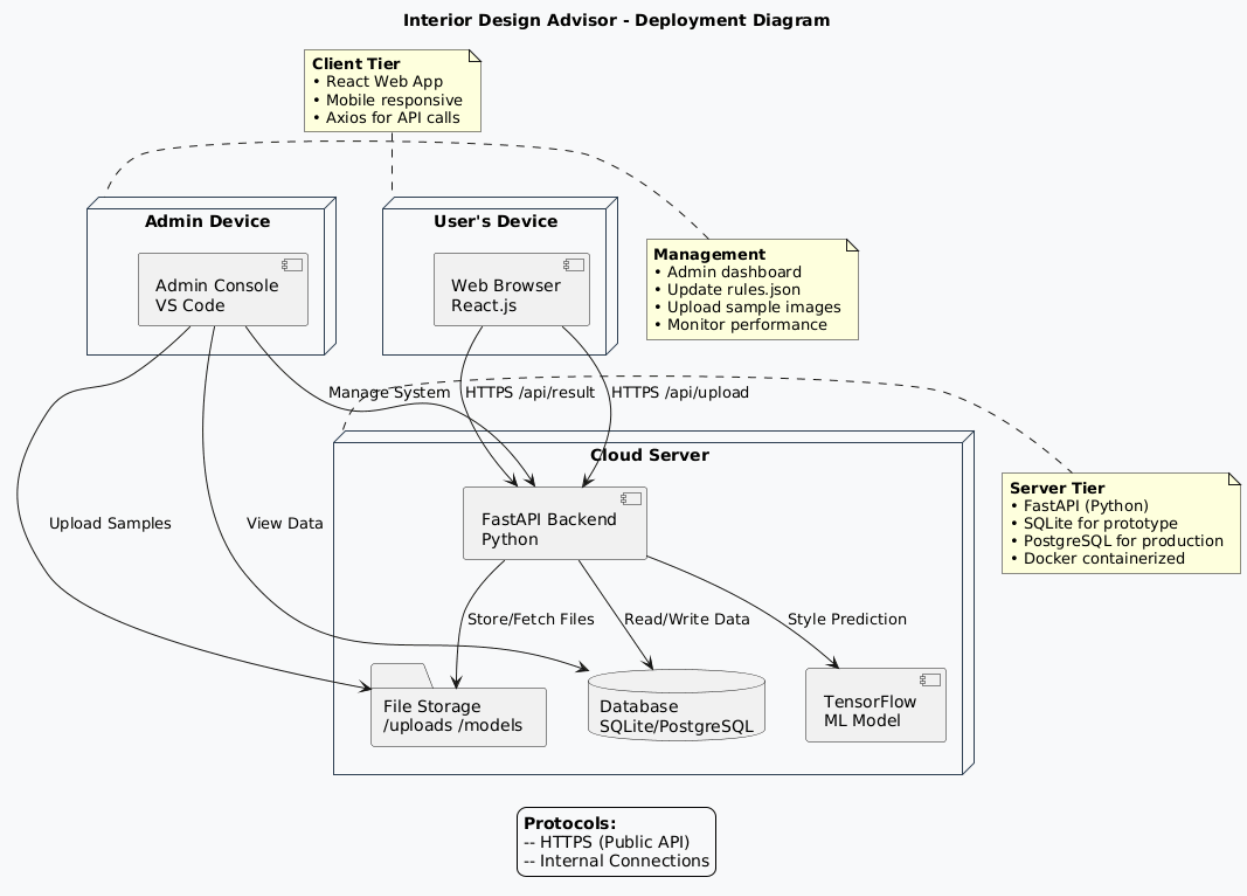
- ML model accuracy may vary
- Poor image quality affects results
- Small datasets may produce lower accuracy
- Poorly angled photos reduce performance

3. System Architecture:

3.1 System Level Architecture

The system follows a layered architecture:

- User Interface Layer
- Back-end API Layer
- Image Processing & AI Layer
- Recommendation Layer



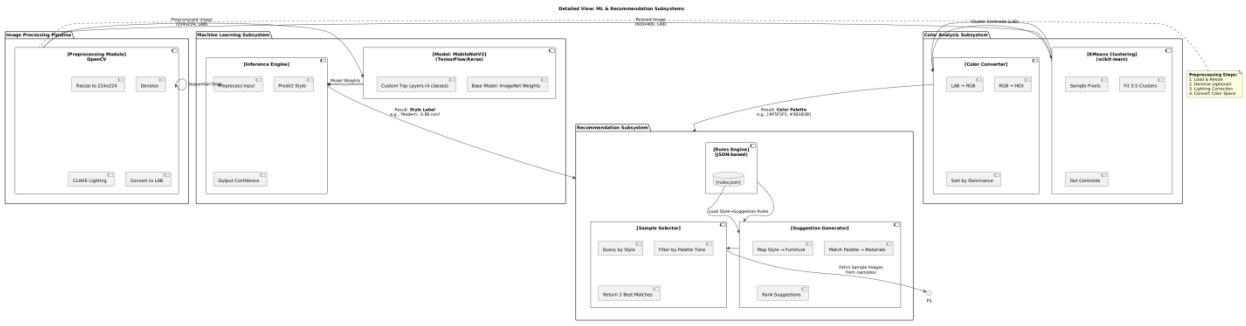
- Front-End: React Web App on User/Admin devices.
- Back-End: Fast-API (Python) on a Cloud Server.
- Storage: PostgreSQL Database and separate File Storage.
- AI: Uses a TensorFlow ML Model for predictions.
- Protocol: All external communication is via HTTPS.

3.2 Sub-System / Component / Module Level Architecture

Main components include:

- Image Upload & Validation Module

- Image Processing Module
- Style Detection Module
- Color Analysis Module
- Recommendation Engine
- Database Management Module

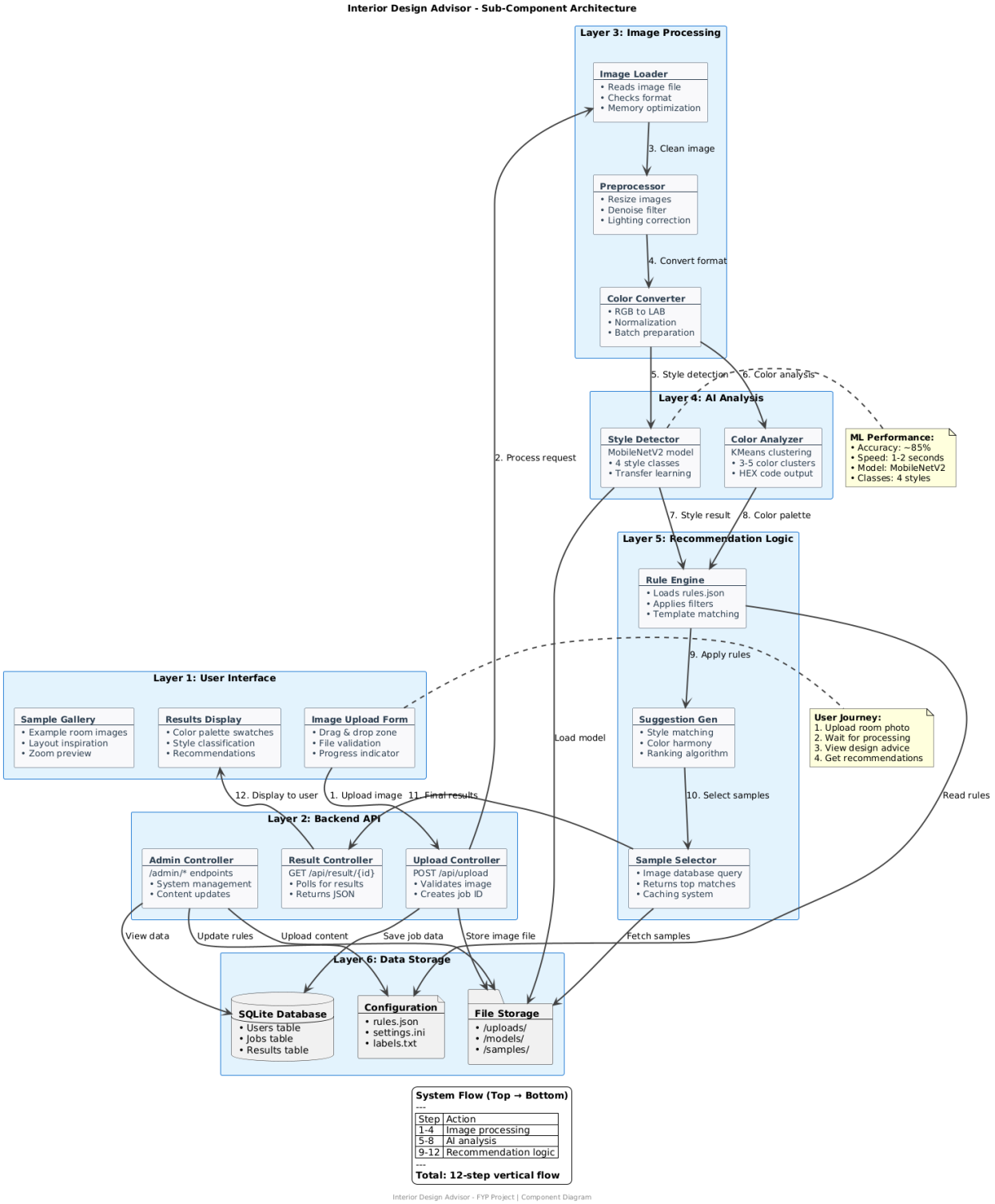


- The main actors are the Home User (for design advice) and the Administrator (for system management).
- Users upload photos to the system to Get Design Advice, which is the core AI-driven feature.
- The system uses a Fast-API (Python) Back-end running on a Cloud Server.
- Images are first processed by OpenCV and then analyzed by a MobileNetV2 (TensorFlow) ML model.
- Data is stored in both a PostgreSQL Database and a dedicated File Storage.
- A Rules Engine combines predicted style and clustered color data to generate the final design recommendations.

3.3 Sub-Component / Sub-Module Level Architecture

- Image Processor: resize, reprocess, convert
- Style Detector: ML prediction, confidence score

- Color Analyzer: k-means clustering, palette extraction
- Recommender: rule-based JSON logic



- User uploads a photo via the Front-end to the Back end for processing.
- The Back end (Controller) sends the image to the ML Engine for style and the Color Engine for the color palette.
- The Recommend-er uses a Rule Engine to combine the style and color results to generate design suggestions.
- The final advice is returned to the user and saved to the SQLite/PostgreSQL Database.
- An Administrator can Update Rules and Manage Content through dedicated controllers.

4. Design Strategies:

This section explains the key design decisions used in developing the Interior Design Advisor system. The goal is to keep the system simple, flexible, and easy to enhance in the future.

4.1 Modular Design Strategy

The system is divided into independent modules such as image processing, style detection, color analysis, recommendation engine, and database. Each module performs a specific task. This approach makes the system easier to understand, test, maintain, and update without affecting other parts.

4.2 Scalability and Extensibility Strategy

The design allows future enhancements such as adding more interior styles, improving ML models, or supporting new image formats. New modules can be added with minimal changes to existing components.

4.3 Re-usability Strategy

Common functionalities like image reprocessing, ML prediction, and rule-based recommendations are designed as reusable components. These can be reused in other AI-based or image-processing projects.

4.4 User Interface Design Strategy

The user interface is kept simple and user-friendly. The system uses clear buttons, progress indicators, and result screens so that even non-technical users can easily upload images and understand the recommendations.

4.5 Data Management Strategy

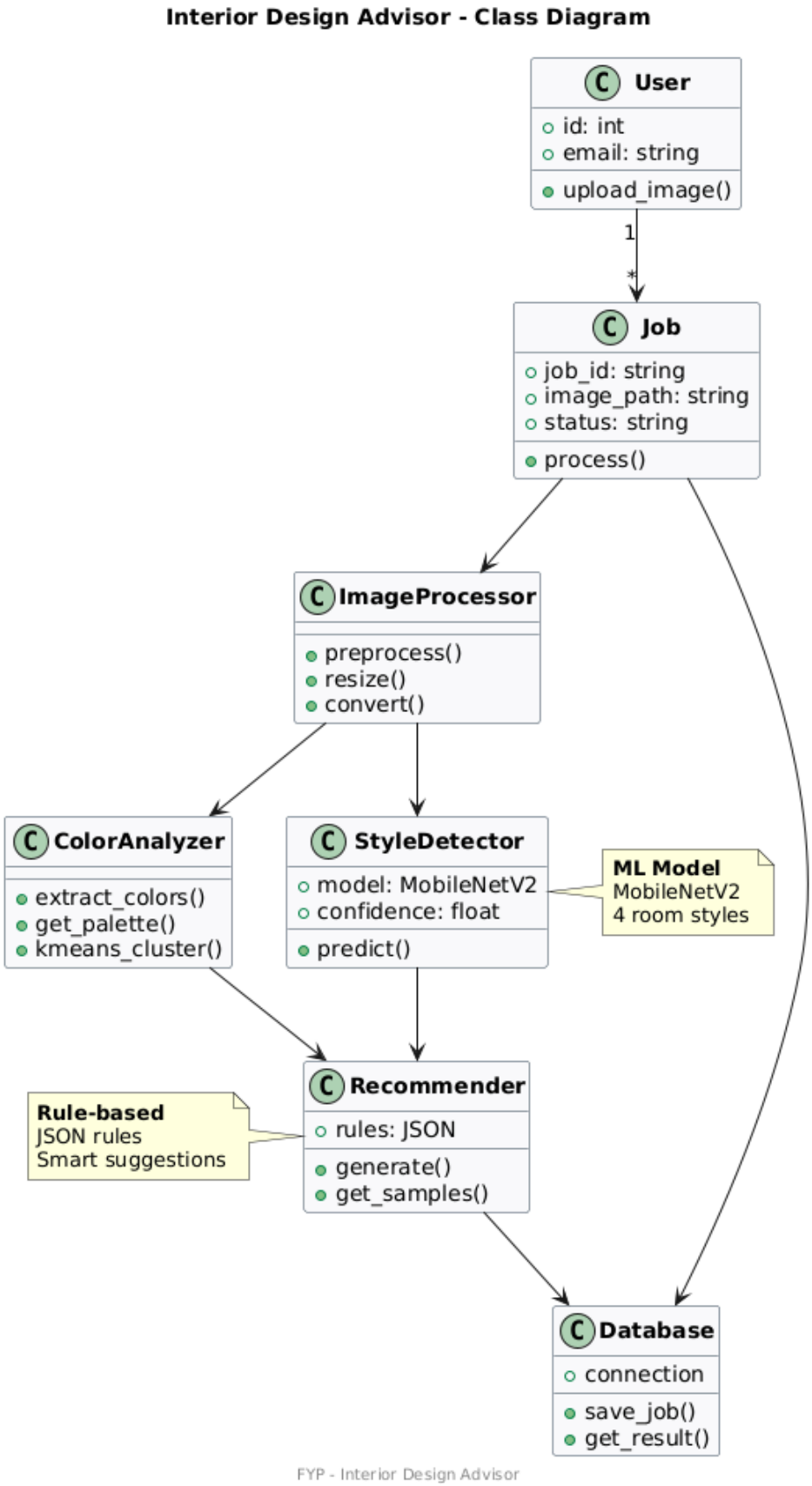
- Images are stored in the file system to reduce database load.
- SQLite database is used for storing users, jobs, and results.
- JSON files are used for storing recommendation rules, making them easy to update.

5. Detailed System Design:

This section provides a detailed explanation of how the system works internally using different design models.

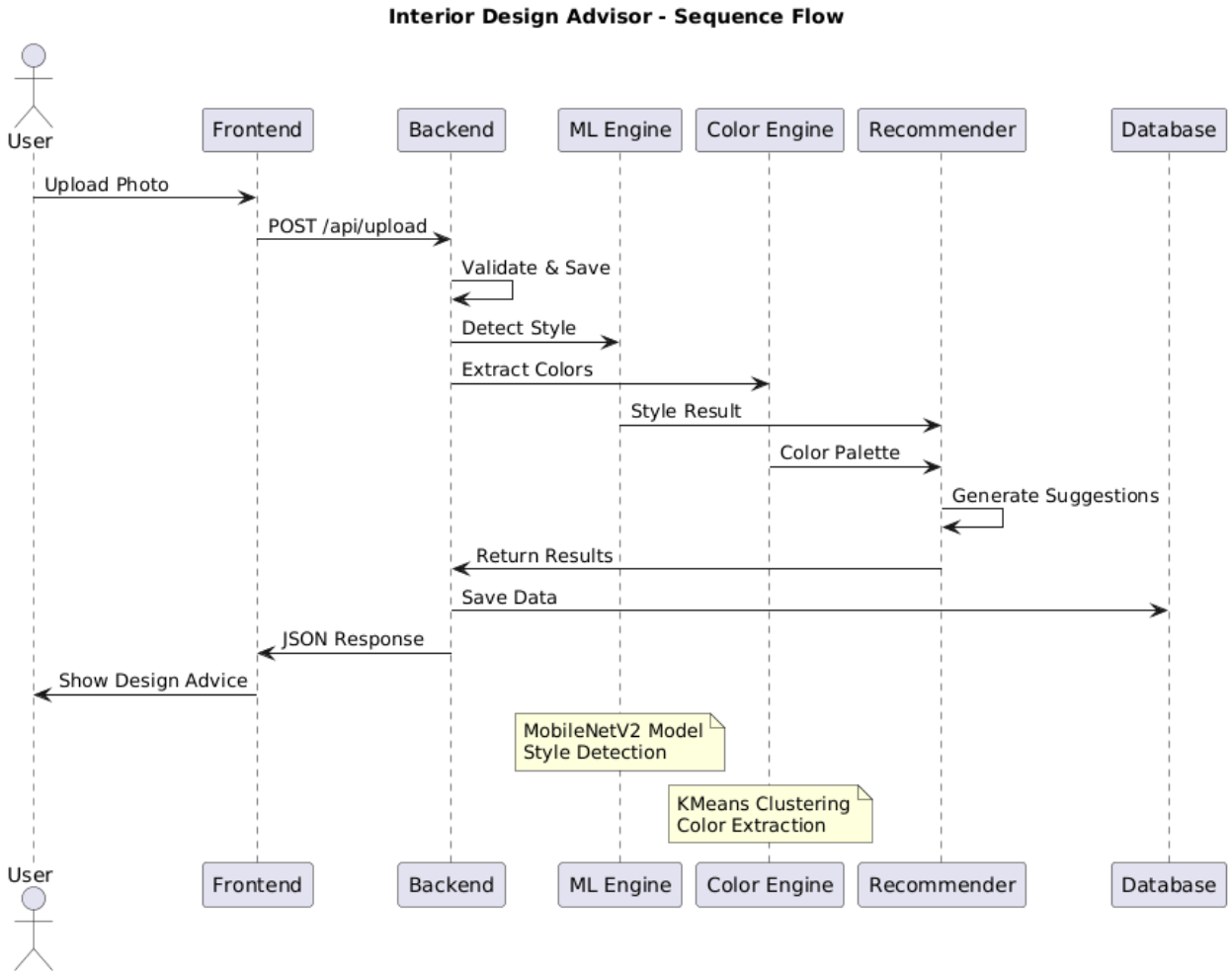
5.1 Class Design

The system follows an object-oriented approach. Each class represents a real-world entity or system component:


Interior Design Advisor - Class Diagram
User
id: int
email: string
upload_image()
1
*
Job
job_id: string
image_path: string
status: string
process()
ImageProcessor
preprocess()
resize()
convert()
ColorAnalyzer
extract_colors()
get_palette()
kmeans_cluster()
StyleDetector
model: MobileNetV2
confidence: float
predict()
ML Model
MobileNetV2
4 room styles
Recommender
rules: JSON
generate()
get_samples()
Rule-based
JSON rules
Smart suggestions
Database
connection
save_job()
get_result()
FYP - Interior Design Advisor

- **User**: Uploads images and views results.
- **Job**: Stores image details, processing status, and controls workflow.
- **Image-processor**: Handles image re-sizing, prepossessing, and format conversion.
- **Style-detector**: Uses the ML model to detect interior design style.
- **Color Analyzer**: Extracts dominant colors and generates a color palette.
- **Recommendation**: Applies rule-based logic to generate design suggestions.
- **Database**: Stores user data, job details, and results.

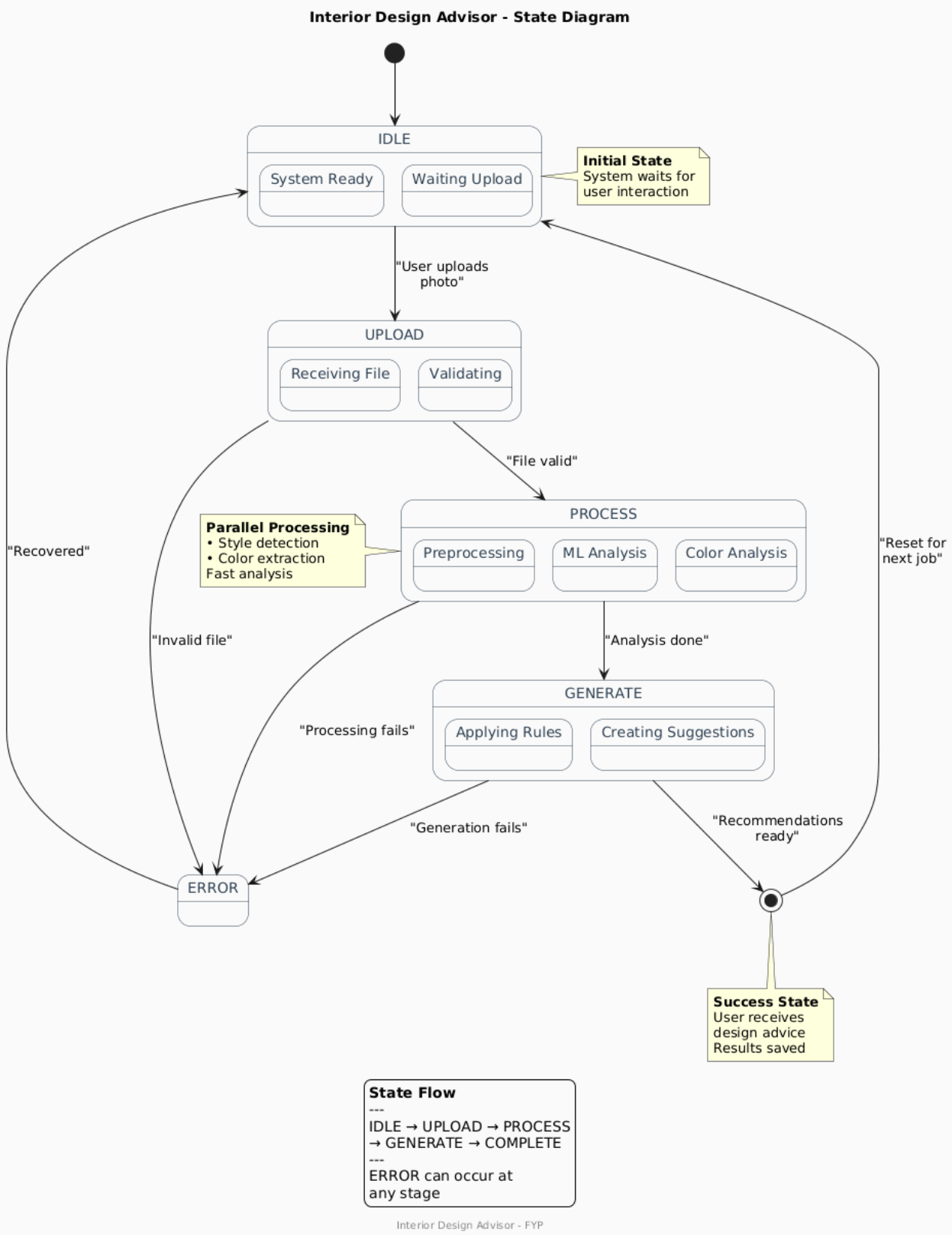
5.2 Sequence Diagram (Working Flow)



- User uploads an image.
- Back-end creates a job and stores the image.
- Image Processor prepossesses the image.
- Style Detector predicts the interior style.

- Color Analyzer extracts dominant colors.
- Recommender generates suggestions.
- Results are saved and displayed to the user.

5.3 State Transition Diagram



A job moves through the following states:

- Uploaded
- Processing
- Analyzing
- Recommendation Generated
- Completed

This ensures proper tracking of each request.

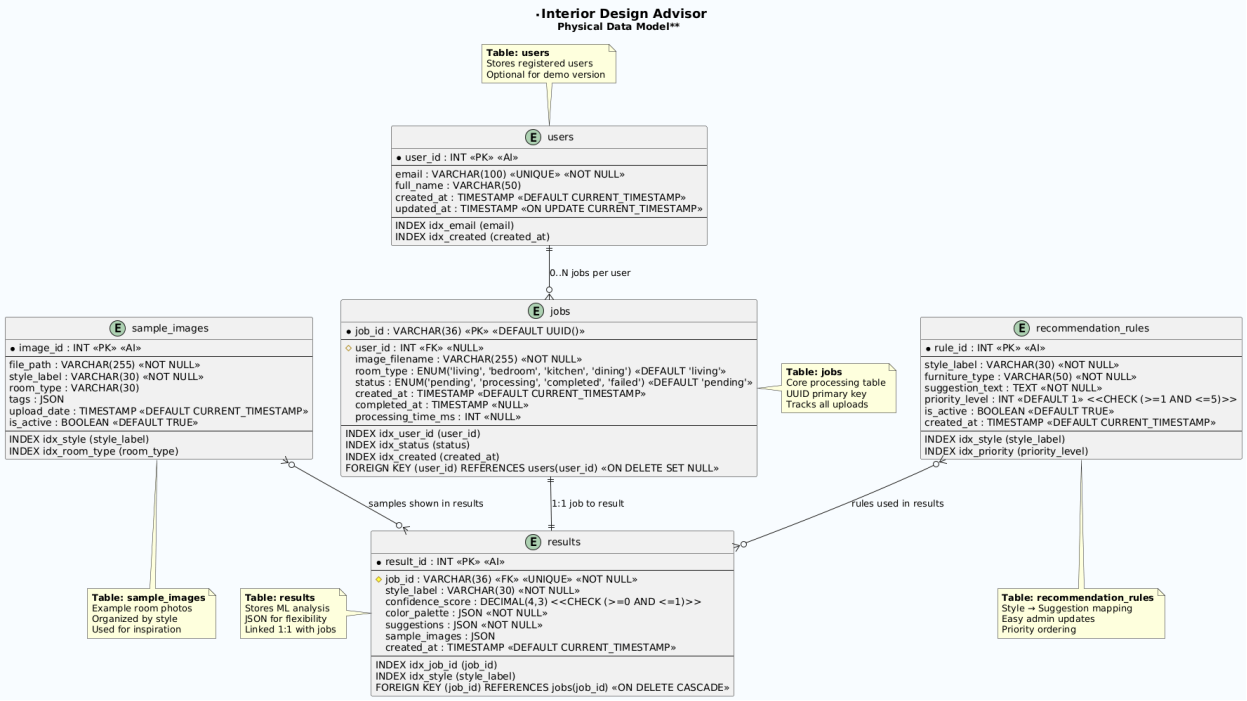
5.4 Logical Data Model

The logical data model includes:

- **Users Table**: Stores user information.
- **Jobs Table**: Stores image paths and job status.

Results Table: Stores style and color recommendations.

5.5 Physical Data Model



- SQLite database for structured data
- File storage for images and sample designs

5.6 GUI Design

The graphical user interface includes:

- Image upload screen
- Processing status indicator
- Results screen showing detected style, color palette, and design suggestions

6.References

Ref. No.	Document Title	Date of Release/ Publication	Document Source
[1]	Project Proposal	Oct 20, 2025	https://github.com/Badarmunir1/capstone-2025/blob/main/Hz_proposal.pdf
[2]	Functional Specification	Oct 20, 2025	https://github.com/Badarmunir1/capstone-2025/tree/main/SRS
[3]	OpenCV Documentation		https://docs.opencv.org
[4]	Open Images Dataset		https://storage.googleapis.com/openimages/web/index.html

7. Appendices

Include supporting details that would be too distracting to include in the main body of the document.

Appendix A: UML Diagrams

The project includes the following UML diagrams:

- System Architecture Diagram
- Component Diagram
- Class Diagram
- Sequence Diagram
- State Transition Diagram

Appendix B: Machine Learning Model

The system uses MobileNetV2 for interior style detection.

- Technique: Transfer Learning
- Output: Interior style with confidence score
- Supported styles: Modern, Classic, Minimal, Industrial

Appendix C: Image Processing

Basic image processing steps include

- resizing,
- noise removal,
- lighting correction,
- and color conversion to improve analysis accuracy.

Appendix D: Database Overview

The database stores:

- User information
- Job and image details
- Style and color analysis results

Appendix E: User Interface Screens

Main screens of the system:

- Image Upload Screen
- Processing Status Screen
- Results Display Screen

8. Diagrams

https://github.com/Badarmunir1/capstone-2025/tree/main