

KINGDOM OF SAUDI ARABIA
Ministry of Higher Education
Taibah University
College of Computer Science and Engineering



المملكة العربية السعودية
وزارة التعليم العالي
جامعة طيبة
كلية علوم و هندسة الحاسبات

Beauty Application

A Project Report Submitted to Fulfill the Requirements of CS289

By Group (5)

Alshaima Hritani (SID: 4471227)
Shatha Ghabban (SID: 4456055)
Aeshah Almubarak (SID: 4455886)
Bushra Almutrafi (SID: 4456347)
Lujain Alahmadi (SID: 4455858)

Supervised By Dr.Ohood Alrohili

30 November, 2025

Abstract

Many people struggle with diagnosing their skin problems and choosing the right products to address them. Others also face difficulty in accurately identifying where to apply makeup. Although there are beauty clinics and salons available, they can be expensive, time-consuming, and not always accessible.

Therefore, and in line with modern technology and the era of advancement, we have developed an application that aims to solve these issues. The app is powered by artificial intelligence, specifically using deep learning, machine learning, and computer vision techniques. It has been trained on a vast amount of data, which allows it to provide highly accurate diagnoses and personalized product recommendations based on each user's skin condition , potentially reducing the need to visit beauty clinics or salons.

In addition, the application helps users identify the correct areas to apply makeup on the face, and suggests products that are suitable for their specific skin concerns.

Keywords: Beauty ; Makeup ; Skincare ;AI

Table of Content

Abstract.....	2
Chapter 1: Introduction	4
1.1 Overview	4
1.2 Client Context	4
1.3 Similar Projects	4
1.4 Problem Statement	4
Chapter 2: Requirements Elicitation	4
2.1 Methodology	5
2.2 Requirements Listing	6
Chapter 3: System Models	9
3.1 Use Case Model	9
3.2 Object Model	2
3.3 Dynamic Model	3
Chapter 4: Architecture Models & Design Patterns	5
4.1 Architecture Models.....	5
4.2 Design Patterns	6
Chapter 5: Test Cases and Develop Test Plans	Error! Bookmark not defined.
5.1 Test Cases	Error! Bookmark not defined.
5.2 Test Plans	Error! Bookmark not defined.
Chapter 6: Conclusion	9
6.1 Conclusion	9
6.2 Lessons Learnt	9
References	10

Chapter 1: Introduction

1.1 Overview

- Our client, a startup beauty company, is suffering from a severe lack of customers on its online platform. Therefore, they are seeking a solution that boosts sales and keeps up with the latest technology.
- The App we will develop will capture an image of the customer's face, analyze the skin, and provide a detailed report highlighting skin issues along with recommended products to help address them. Additionally, it will offer suggestions for suitable makeup colors based on the user's skin tone and indicate the appropriate areas of application. The use of AI is essential in this App, as the model is trained on large number of samples, resulting in highly accurate analysis.
- The AI model has been trained on large datasets covering a wide range of skin types and conditions. As a result, the AI system can automate the diagnostic process, enhance decision-making, and offer personalized product recommendations.

2.1 Client Context

- Our client is a startup beauty company.
- Integrating AI into the project will significantly boost productivity, as the model's analysis is highly accurate and provides direct, practical solutions. It also recommends products available in the company's market that match the specific skin issues.

1.3 Similar Projects

- In Korea, similar projects have been developed by companies such as NEWAGINE and MEDI SILK, featuring smart mirror and camera systems. [1]
- It identifies facial features, recommends accurate products, and offers solutions for skin problems. It combines capturing, analyzing, and displaying, but it is expensive and not suitable for everyone.
- Our project includes a feature that shows where to apply makeup on the face. It's simple, easy to use, and available anytime.

1.4 Problem Statement

- This project aims to help our client to boost their online sales, also helps the company get more online customers.
- The company is not famous and lacks online customers.
- Therefore, they want a product that can attract customers and help increase their online sales.

Chapter 2: Requirements Elicitation

2.1 Methodology

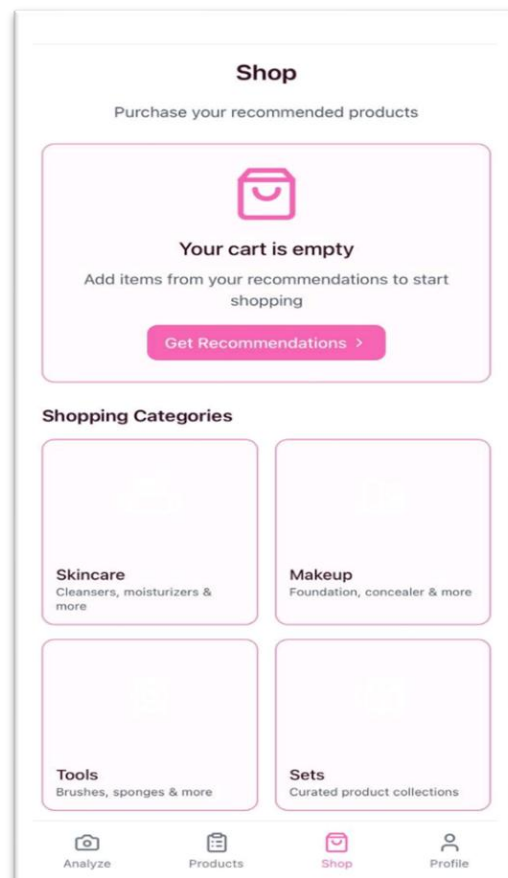
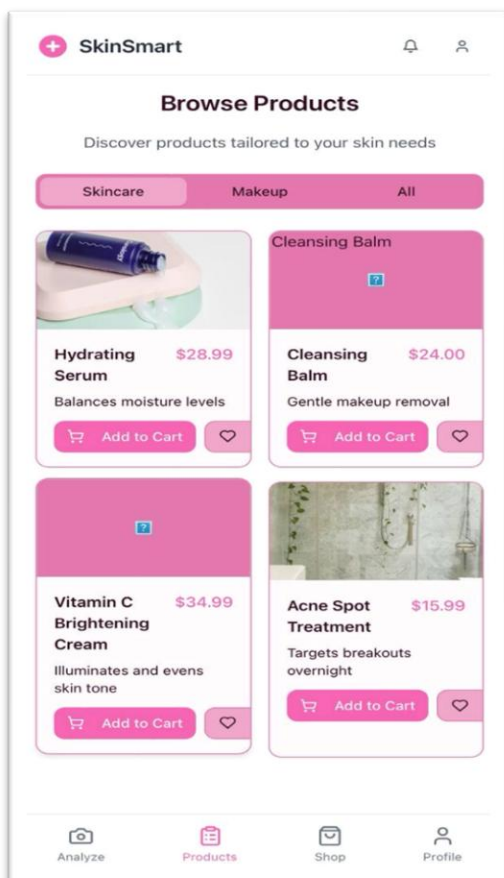
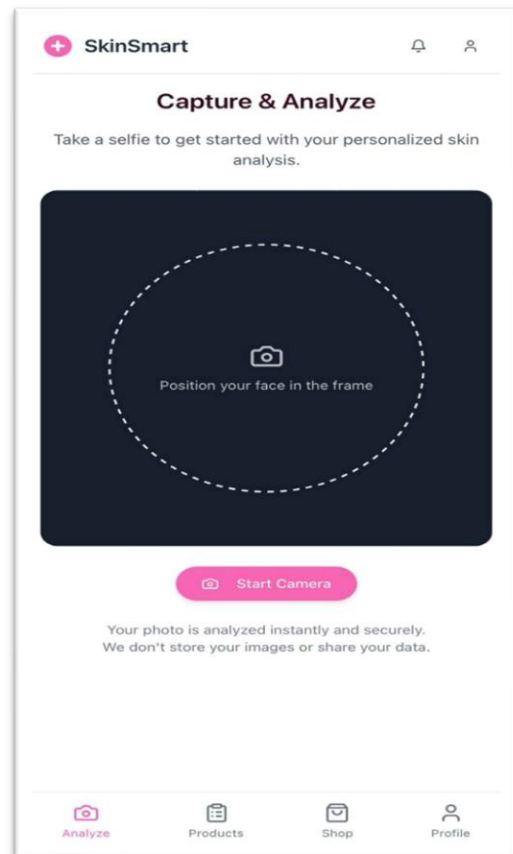
We have used two methods since our project is complex and requires continuous feedback:

- We use Questionnaire method to gather requirements from stakeholders. This questionnaire will help us to understand the stakeholder's requirements, needs, and expectations regarding the development of the beauty application

The questionnaire is titled "Stakeholder feedback questionnaire" and is divided into three main sections:

- Section 1: General information**
 - Introduction: "This questionnaire will help us to understand the stakeholder's requirements, needs, and expectations regarding the development of the beauty application. Your feedback is important to us. Thank you for your participation."
 - Field: "الوصف (اختياري)" (Description - Optional)
 - Field: "Name:" with a text input field and a "تصنيف الإجابة" (Answer classification) dropdown.
 - Field: "Position/Role:" with a text input field and a "تصنيف الإجابة" (Answer classification) dropdown.
 - Field: "company:" with a text input field and a "تصنيف الإجابة" (Answer classification) dropdown.
- Section 2: Questions**
 - Question 1: "What specific problem do you hope this app will solve?" with a text input field and a "تصنيف الإجابة" (Answer classification) dropdown.
 - Question 2: "What is your vision for the beauty app?" with a text input field and a "تصنيف الإجابة" (Answer classification) dropdown.
 - Question 3: "What are the essential features you want for the app?" with a text input field and a "تصنيف الإجابة" (Answer classification) dropdown.
 - Question 4: "Have you used similar apps? What did you like or dislike about them?" with a text input field and a "تصنيف الإجابة" (Answer classification) dropdown.
 - Question 5: "Are there any key milestones or phases we should aim to achieve?" with a text input field and a "تصنيف الإجابة" (Answer classification) dropdown.
- Section 3: Additional questions**
 - Question 6: "How much do you expect the application cost will be?" with a text input field and a "تصنيف الإجابة" (Answer classification) dropdown.
 - Question 7: "What additional feature would you like to see in the app?" with a text input field and a "تصنيف الإجابة" (Answer classification) dropdown.
 - Question 8: "Any other comments or suggestions?" with a text input field and a "تصنيف الإجابة" (Answer classification) dropdown.

- These prototype screens were developed after conducting questionnaires to gather insights from stakeholders, ensuring the interface meets user needs and expectations before final implementation. [2]



2.2 Requirements Listing

2.2.1 Functional Requirements:

- A) The app must accurately analyze the skin tone and type.
- B) The app must point out the areas of the skin issues
- C) The app must detect facial features and recommend precise areas for makeup application.
- D) The app must suggest suitable makeup shades based on the skin tone.
- E) The app must generate a report with suitable skincare products for specific skin issues.
- F) The app must recommend the products from the company's website only.
- G) The app must offer multiple options of payment. [3]

2.2.2 Non-Functional Requirements:

Performance Guidelines:

- Optimize AI model efficiency
- improve response time
- Automatic filtering of blurry images

Security Guidelines:

- Delete images automatically
- Avoid sending sensitive data to the server

Accuracy Guidelines:

- Training the model on realistic and low-quality data
- Using advanced and reliable AI libraries

Usability Guidelines:

- Displaying multiple suggestions with priority ranking
- Displaying accuracy percentage for each suggestion
- Responsive design for all devices

	Performance	Security	Accuracy	Usability
<i>Optimize AI model efficiency</i>	Complementary	Conflicting	Complementary	Complementary
<i>improve response time</i>	Complementary	Independent	Complementary	Complementary
<i>Automatic filtering of blurry images</i>	Complementary	Independent	Complementary	Complementary
<i>Delete images automatically</i>	Independent	Complementary	Independent	Independent
<i>Avoid sending sensitive data to the server</i>	Conflicting	Complementary	Overlapping	Overlapping
<i>Training the model on realistic and low-quality data</i>	Complementary	Independent	Complementary	Overlapping
<i>Using advanced and reliable AI libraries</i>	Complementary	Independent	Complementary	Overlapping
<i>Displaying multiple suggestions with priority ranking</i>	Overlapping	Independent	Overlapping	Complementary
<i>Displaying accuracy percentage for each suggestion</i>	Overlapping	Independent	Complementary	Complementary
<i>Responsive design for all devices</i>	Independent	Independent	Independent	Complementary

Chapter 3: System Models

3.1 Use Case Model

3.1.1 Use Case Descriptions

User stories (scenarios):

A) As a client, I want the app to accurately analyze my skin tone and type so that I can receive personalized skincare and makeup recommendations tailored to my unique needs.

B) As a client, I want the app to highlight my skin issues (e.g., acne, dryness) so that I can use treatments effectively.

C) As a makeup lover, I want the app to detect my facial features and recommend precise areas for makeup application (e.g., contouring, eyeshadow) so that I can confidently improve my appearance.

D) As a client, I want the app to suggest makeup shades that complement my skin tone so that I may select products that naturally complement my complexion.

E) As a user with some skin issues, I want the app to generate a report of suitable skincare products so that I can address my concerns with targeted solutions.

F) As a loyal customer, I want the app to recommend skincare and makeup products exclusively from the company's website so that I can easily purchase authentic items directly from the company.

G) As a user, I want the app to offer multiple payment options (e.g., credit card, PayPal) so that I can complete purchases using my preferred method.

Use cases:

Use case	Analyze skin tone and type
Goal	Accurately analyze my skin tone and type
Actors	Primary actor: customer, secondary actor: camera
Preconditions	- The customer is logged into the mobile app.
Postconditions	- Skin tone and type are successfully analyzed.
Main flow	1. The customer takes a selfie. 2. The system analyzes the image. 3. The system provides skin tone and type.
Alternative flow	-
Exceptions	1. If the image is not clear, the system displays a message to retake a photo. 2. If there is more than one face in the image, the system displays a message to retake a photo with only one face.

Use case	Highlighting skin issues
Goal	The app should highlight all skin's issues
Actors	Primary actor: customer, secondary actor: -
Preconditions	- Skin analysis has been completed
Postconditions	- Area of skin issues are highlighted
Main flow	1. The system processes the image 2. the system identifies and highlights areas with skin issues
Alternative flow	-
Exceptions	If there is no issue found, the app notifies the customer

Use case	Recommend precise areas for makeup application
Goal	Detect facial features and recommend precise areas for makeup application
Actors	Primary actor: customer, secondary actor: -
Preconditions	<ul style="list-style-type: none"> - The customer logs in. - The customer takes a selfie. - The customer confirms the photo.
Postconditions	<ul style="list-style-type: none"> - System analyzes the face. - System highlights the areas. - System writes on each area what to put on.
Main flow	<ol style="list-style-type: none"> 1.The system highlights each area. 2. The system numbers each area. 3. The system writes what product will be used next to each number.
Alternative flow	-
Exceptions	If the tool shows wrong places of makeup, the user should notify it, and it will reload the process again

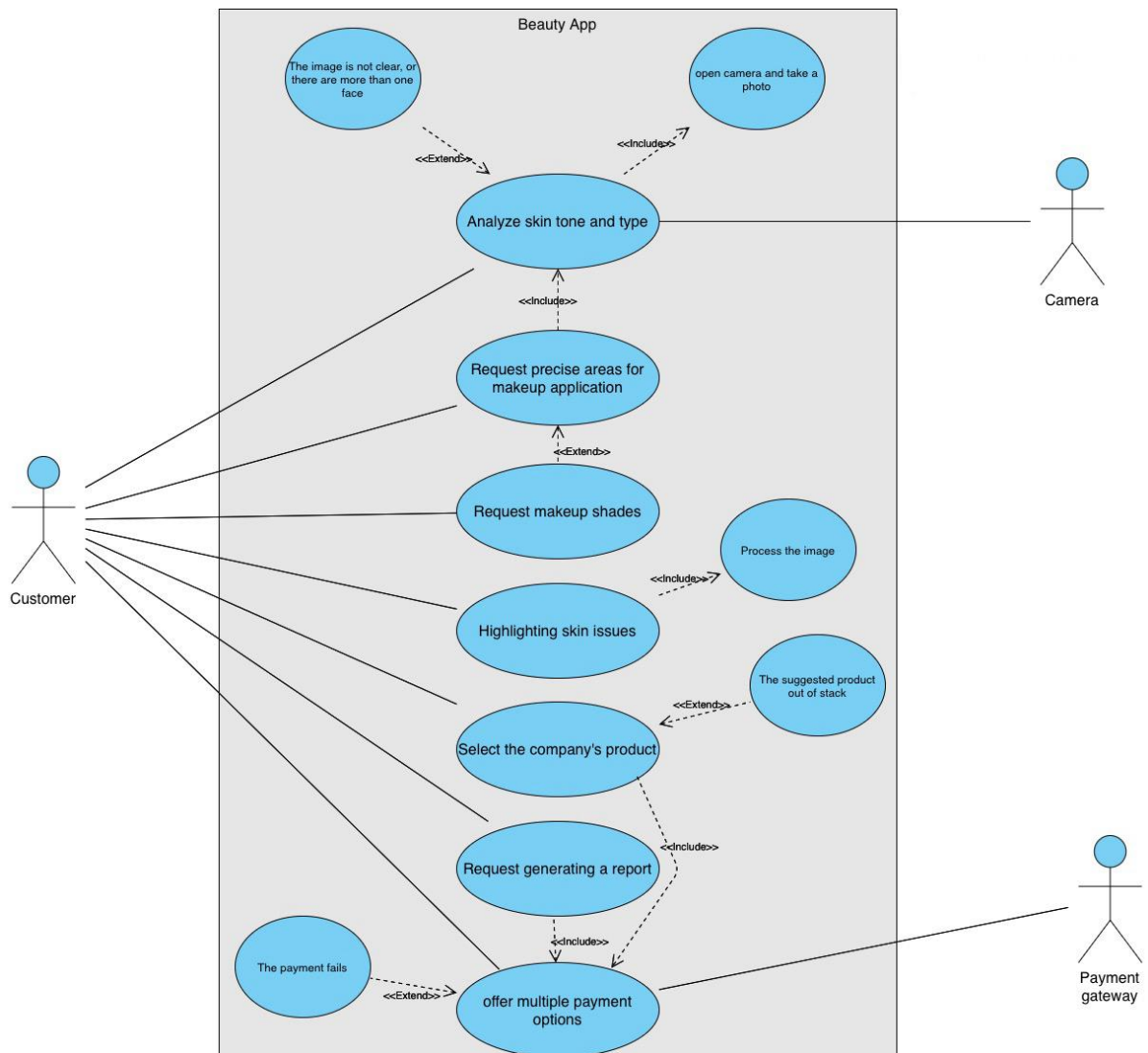
Use case	Suggest makeup shade
Goal	The app suggests makeup shades that complement the skin tone
Actors	Primary actor: customer, secondary actor: -
Preconditions	<ul style="list-style-type: none"> - The customer is logged into the mobile app. - The customer takes a selfie. - System analyzes the image.
Postconditions	The system suggests the best makeup shades
Main flow	<ol style="list-style-type: none"> 1.System determines the color of skin. 2. System suggests the best colors for each product (e.g.: rose shades for eyeshadow, etc.)
Alternative flow	<p>Change suggestion:</p> <p>If the customer doesn't like the shades, the customer will ask the tool to change the recommendations</p>
Exceptions	-

Use case	Generate a report of problems and suitable skincare products
Goal	The app will eventually generate a report on the suggested products approved by the user, including usage instructions for each product
Actors	Primary actor: customer, secondary actor: -
Preconditions	<ul style="list-style-type: none"> - The customer is logged into the mobile app. - The customer takes a selfie. - System analyzes the image.
Postconditions	<ul style="list-style-type: none"> - System determines the problems. - System suggests suitable company products and sends a report to the user. - The user chooses the products. - The order is confirmed. - The payment is processed successfully.
Main flow	<ol style="list-style-type: none"> 1.The system gives solutions for skin problems and suggests some Products from the company. 2. The system sends a full description report. 3. The customer provides payment details. 4. The system confirms the payment and displays a confirmation screen.
Alternative flow	-
Exceptions	-

Use case	Suggest the company's products
Goal	the app will only suggest the company's products for the customer
Actors	Primary actor: customer, secondary actor: -
Preconditions	<ul style="list-style-type: none"> - The tool has analysis the face image. - The skin issues have been highlighted. -The makeup recommendation has been suggested.
Postconditions	<ul style="list-style-type: none"> - The customer chooses the company's products. - The order is confirmed. - The payment is processed successfully
Main flow	<ol style="list-style-type: none"> 1-The customer selects the company's products. 2-The customer provides payment details. 3-The system confirms the order and displays a confirmation screen
Alternative flow	-
Exceptions	If the product out of stock, the app suggests similar products if possible

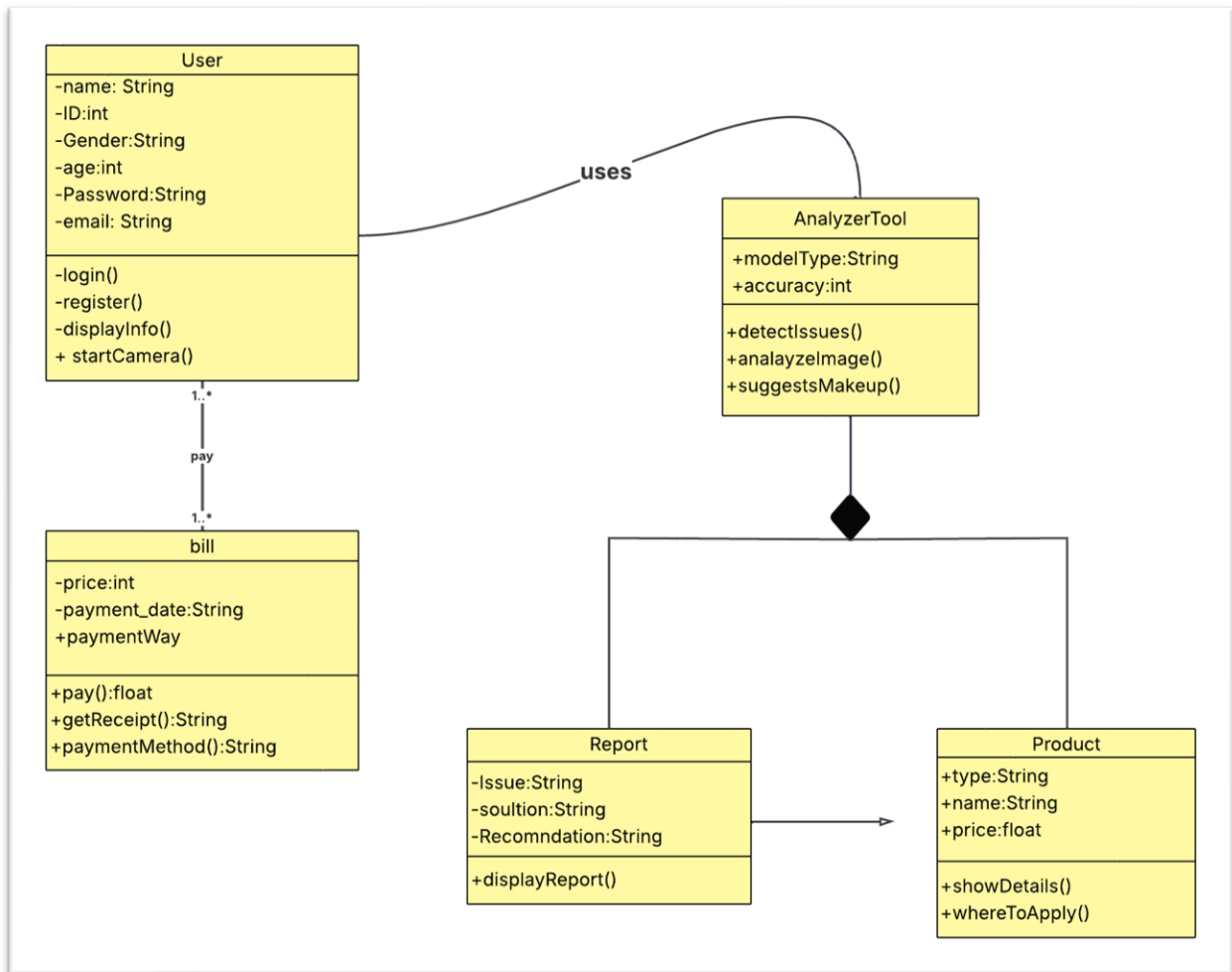
Use case	Offers multiple payment options
Goal	Enable customers to choose their preferred method and offer various payment alternatives
Actors	Primary actor: customer, secondary actor: payment gateway
Preconditions	<ul style="list-style-type: none"> -The customer takes a selfie. - AI analyzes the image. -The report generated or product selected.
Postconditions	<ul style="list-style-type: none"> - The customer chooses the products. - The order is confirmed. - The payment is processed successfully
Main flow	<ol style="list-style-type: none"> 1. The customer provides payment details. 2-The customer chooses the suitable payment method. 3. The system confirms the order and displays a confirmation screen
Alternative flow	<ol style="list-style-type: none"> 1-If the customer cancels, the system returns to the cart. 2-If the customer modified the order, the system updated the cart and the payment details.
Exceptions	If the payment fails, the system asks the customer to try again or change to another way to pay.

3.1.2 Use Case Diagram [4]



3.2 Object Model

3.2.1 Class Diagram



User class: Represent who uses the app, it shows an **association** relationship with the AnalyzerTool class, the user **uses** the Analyzer tool (no dependency will exist without it)

Analyzer tool class: Represent the AI model inside the app, it shows a **composition** relationship with the product and issue classes (it is a part of the app)

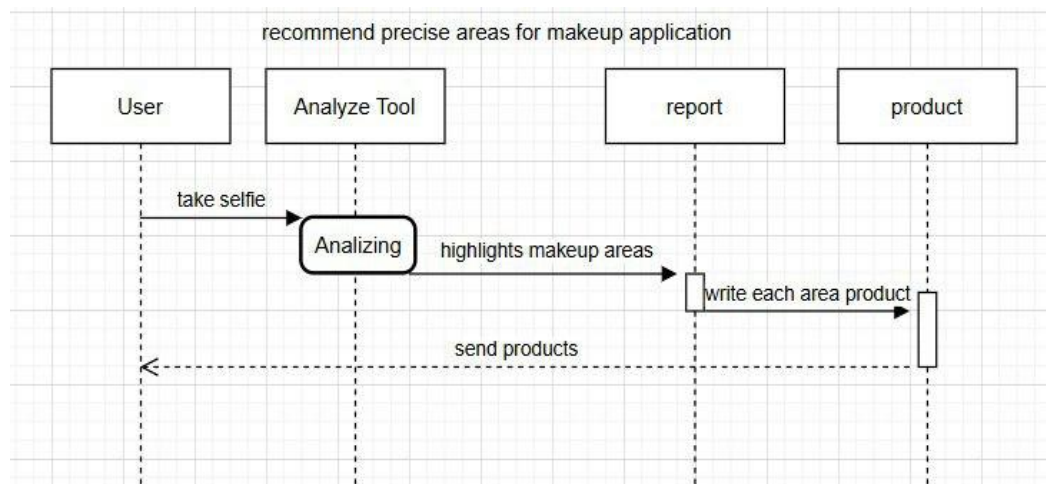
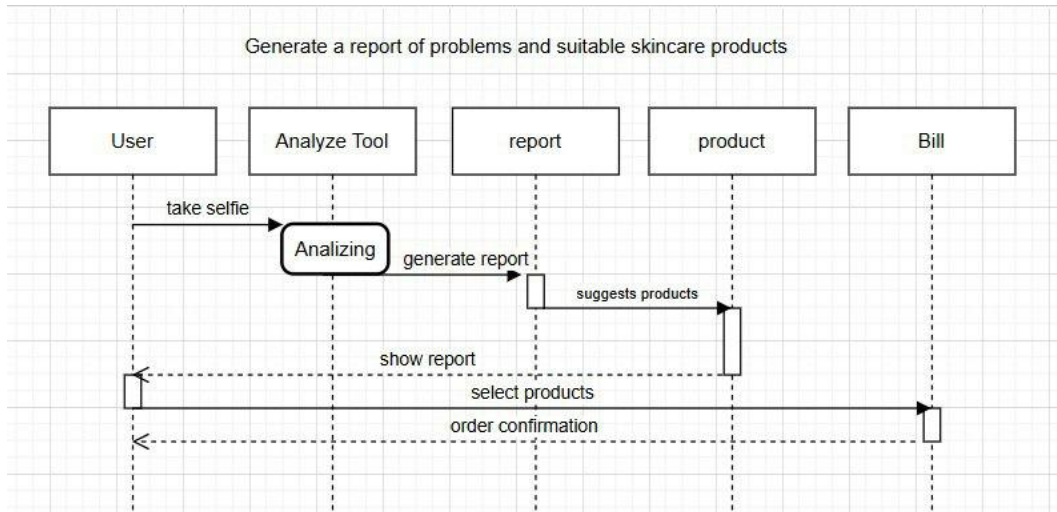
Report class: Represent the report generated from the analyzer tool, it shows a **composition** relationship with the analyzer tool class (won't exist without it) and **inherit** from the product class

The product class: Represent the recommended product, it shows a **composition** relationship with the analyzer tool class (won't exist without it)

The bill class: Represent the bill, it shows an **association** relationship with user class (. At least one user or more will pay by payment gateway once or more) [5]

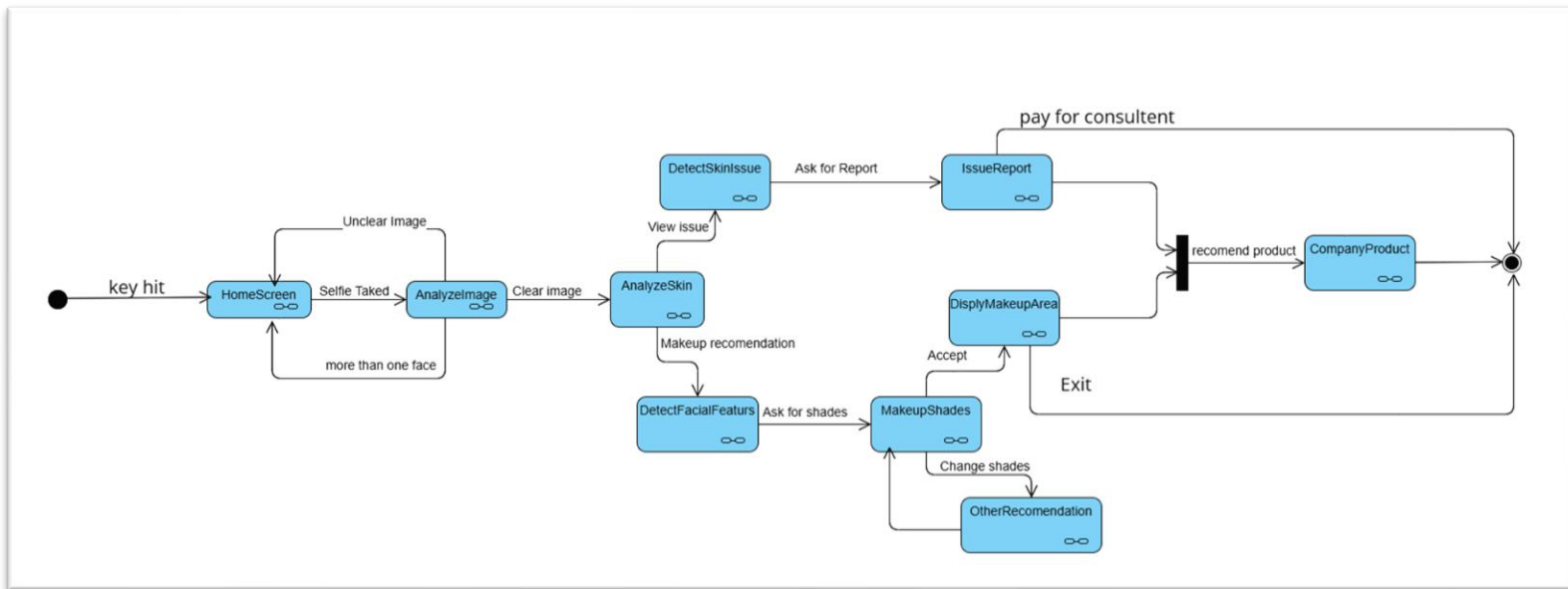
3.3 Dynamic Model

3.3.1 Sequence Diagrams [6]



3.3.2 State Diagram

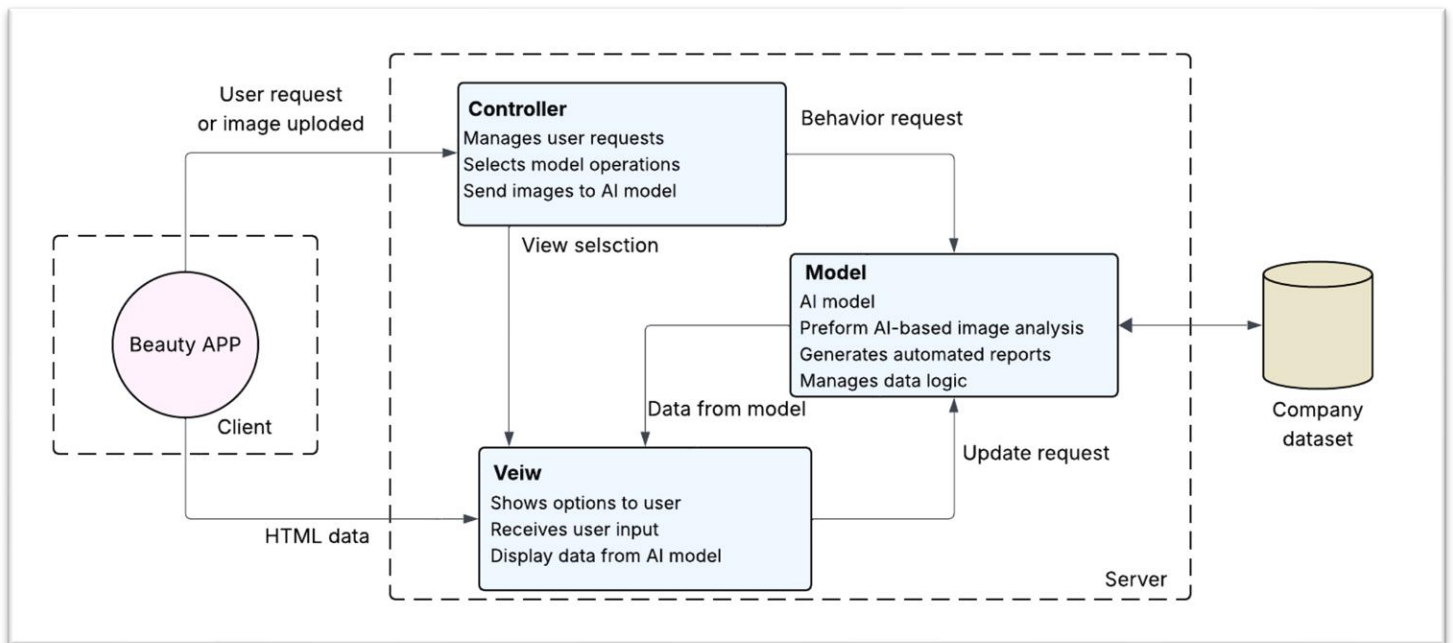
State diagram for the AnlyzerTool lifecycle.



Chapter 4: Architecture Models & Design Patterns

4.1 Architecture Models

- We have chosen the model view controller (MVC) depending on the system's nonfunctional requirements, focusing on security, performance, usability and accuracy. The MVC style separates the system logic from the user interface, and each component is assigned distinct, non overlapping responsibilities that complement each other, leading to a streamlined and well-organized workflow also the user is separated from the internal system, ensuring high security and privacy and improving response time, for that choosing this style will be the best option for the AI model outputs. [7]



4.2 Design Patterns

Creational Patterns:

- **Abstract Factory:** to generate proper product recommendations based on different skin tones.
 - Abstract Factory: `makeupRecommendationsFactory` (defines makeup shades for each tone, such as foundation, blush, lipstick, etc.)
 - Concrete Factories:
 - `warmToneFactory`: Create warm tone products.
 - `coolToneFactory`: Create cool tone products.
 - `NeutralToneFactory`: Create neutral tone products.
- **Builder:** to build personalized makeup looks based on user analytics and preferences.
 - Builder: `MakeupRecommendationBuilder` (defines steps for adding makeup such as foundation, concealer, eyeshadow, highlighter, etc.)
 - Concrete Builders: `dailyLookBuilder`, `glamLookBuilder`, `artisticLookBuilder` (each provides custom makeup recommendations specific to the user's preference)

Structural Patterns:

- Adapter: To integrate multiple payment gateways (PayPal, Mada, etc.) with different interfaces to keep the app's payment processing simple and consistent.
 - Imagine the user wants to pay with Mada, the app will simply call the `PaymentMethod ()`. The `MadaAdapter` secretly handles currency conversion to SAR, Arabic API formatting and Mada's special authentication, Similarly as `PayPalAdaptor` would adapt the same call for PayPal's system.
- Container: To manage and organize different steps of the skin/makeup analysis process (image validation, feature detection, etc).
 - Each step of the analysis processes groped into one unified container object. Whether the user requests makeup shades suggestions or skin issue report, the `AnalysisContainer` will validate the image and detect skin issues and facial features then provide what the user wants.

Behavioral Patterns:

- Chain of Responsibility:

To handle the classification of skin issues where multiple classifiers with varying levels of specificity might attempt to categorize a skin issue.

A chain of classifiers handlers could be set up (e.g. AcneClassifier, DrynessClassifier, PigmentationClassifier, FallbackClassifier), Each classifier handles a specific issue type or passes it to the next

- Command:

To encapsulate different actions such as: productRecommendationsCommand, generateReportCommand and paymentProcessingCommand. [8]

Chapter 6: Conclusion

6.1 Conclusion

This project established an AI-powered beauty app using deep learning and computer vision principles to analyze skin conditions and recommend products, we gathered requirements through questionnaires and prototyping.

Following Agile techniques like user stories translated into functional requirements and non-functional requirements, including visual diagrams such as class diagram, state diagram, sequence and use case diagrams.

Also covering MVC architecture style to separate the AI processing (Model) from UI (View). Design patterns like Abstract Factory, Builder, Container, Chain of Responsibility, Adapter, and Command enhanced modularity.

6.2 Lessons Learnt

- Learned the importance of teamwork and effective task distribution.
- Understood how essential it is to gather and analyze user requirements early.
- Gained experience using Agile methodology.
- Realized the need to balance technical solutions with business goals.
- Improved our skills in applying design patterns and architecture models.

References

- [1] "5 Spectra Mode AI Facial Camera Skin Analysis Machine," NewAngie, [Online]. Available: <https://www.newangie.com/5-Spectra-Mode-AI-Facial-Camera-Skin-Analysis-Machine-SA15-pd560552068.html>.
- [2] "Turn your ideas into apps," Replit, [Online]. Available: <https://replit.com/>.
- [3] "A Guide to Functional Requirements (with Examples)," Nuclino, [Online]. Available: <https://www.nuclino.com/articles/functional-requirements>.
- [4] Visual paradigm, [Online]. Available: <https://online.visual-paradigm.com/drive/#proj=0&diagram=list>.
- [5] "Diagramming powered by intelligence," Lucid, [Online]. Available: <https://www.lucidchart.com/pages>.
- [6] draw.io, [Online]. Available: <https://www.drawio.com/>.
- [7] "Types of Software Architecture Patterns," GeeksforGeeks, 07 04 2025. [Online]. Available: <https://www.geeksforgeeks.org/types-of-software-architecture-patterns/>.
- [8] "The Catalog of Design Patterns," REFACTORING.GURU, [Online]. Available: <https://refactoring.guru/design-patterns/catalog>.