ELEC5620 Personal Cosmetic Assistant

Project Stage 1: Requirements Documentation, Architecture Design and System Modeling

Team Members:

Yue Kang (540574397)
Ethan Ma (540169665)
Sean Pattana (540906048)
Evangeline Zhang (540498196)
Yiming Shen (540139598)

Table of Contents

1. Introduction **4**
2. Identifying and Solving the Problems **4**
2.1 Information overload 4
2.2 Knowledge gap 4
2.3 Health risks 4
3. Importance of Agentic AI **5**
3.1 Continuous and autonomous monitoring 5
3.2 Real-time, complex data processing 5
3.3 Data-driven decisions 5
4. Primary User Groups and Roles **5**
4.1 Users 5
4.2 Data Analytic Team 5
4.3 Technical staff 5
5. Key Capabilities and Core Features **6**
6. Ad-Hoc Approach **8**
6.1 Upload facial images (Ethan) 8
6.2 Facial analysis (Ethan) 8
6.3 Database for cosmetic products (Sean) 8
6.4 Automatically get cosmetic products and put them into the database (Sean) 8
6.5 User preference input (Yue) 8
6.6 Ingredient conflict analysis (Yue) 8
6.7 Product recommendations (Evangeline) 8
6.8 Skincare routine tracking and feedback (Yiming) 9
7. Feature-Oriented Approach **9**
8. Use case overall diagram **10**
9. Use Case Approach **11**
9.1 Use Case 1 (Yue) 11
9.2 Use Case 2 (Ethan) 11
9.3 Use Case 3 (Sean) 12
9.4 Use Case 4 (Yiming) 12
9.5 Use Case 5 (Evangeline) 13
9.6 Use Case 6 (Yiming) 14
10. Architecture Analysis and Design(s) **15**
10.1 Model of Computation (MoC) 15
10.2 The 4+1 Viewpoint 15
11. Elementary Structure Modelling **16**
11.1 Class Diagram 16
12. Complex Structure Modelling **17**
12.1 Object diagram 17
12.2 Collaboration diagram 18
12.3 Structure Class 19
14. Activity Diagrams **20**

14.1 Data analysis team manages the database (Yue) 20
14.2 User access product details (Ethan) 21
14.3 Cosmetic product ingestion system (Sean) 22
14.4 Skincare Routine Tracking & Feedback (Yiming) 23
14.5 User upload the image (Evangeline) 24
15. Interaction Diagrams **25**
15.1 Data analysis team manages the database (Yue) 25
15.2 User access product details (Ethan) 26
15.3 Data analyst analyze the recommendation system (Sean) 27
15.4 Skincare Routine Tracking & Feedback — Interaction Diagram Description (Yiming) 28
15.5 User upload the image (Evangeline) 29
16. State Machine Diagrams **30**
16.1 Data analysis team manages the database (Yue) 30
16.2 User access product details (Ethan) 31
16.3 Data crawling system (Sean) 32
16.4 Skincare Routine Tracking & Feedback (Yiming Shen) 33
16.5 User upload the image (Evangeline) 34
17. Package Diagram **35**
18. Deployment Diagram **36**
19. Contribution Table **37**

1. Introduction

We've all been there. Sifting through hundreds and thousands of websites, trying to find the best option for ourselves. Cosmetic products are no different. With the rapid development of technology, every brand is growing rapidly, each striving to improve quality, efficiency and features. As companies focus on improvements, cosmetic products are becoming less comparable as their quality standards are starting to match. More importantly, the internet allows everyone to access almost any product in the world, resulting in an unprecedented variety of choices and convenience for consumers.

This is where our Personal Cosmetic Assistant comes in. By leveraging the power of Agentic AI, we are able to collect facial data and offer unique insights. Using this analysis, it can then provide personalised cosmetic product recommendations. Our product aims to eliminate information overload from manual labour or lack of knowledge. On top of that, we can also automatically detect ingredient conflicts between cosmetics to alert users of potential allergies. All of this combined is to create a smooth and effortless experience for users who want to save time but still invest in cosmetics.

2. Identifying and Solving the Problems

2.1 Information overload

The rapid proliferation of cosmetic brands and products has backfired on the consumers. The sheer amount of options makes it harder for consumers without extensive interest and knowledge to make well-informed purchasing decisions. The easy accessibility of the internet exposes consumers to a wide variety of options, leading to confusion and decision fatigue.

Our project consolidates all the available data, breaks it down into smaller pieces and makes it more digestible for the consumers. This can eliminate any fear of being overwhelmed with too many options and making the wrong decisions.

2.2 Knowledge gap

Most consumers lack the expertise or resources to explore all the options. Sometimes they may be able to make good decisions based on word of mouth but not everyone is lucky enough to get good recommendations. Therefore, finding good cosmetic products proves to be a difficult task without the right knowledge.

Our project utilizes Agentic AI which has extensive knowledge about cosmetic product quality. Consumers do not have to spend time doing research before making a purchase. It also serves as a learning experience as consumers can gain knowledge on the go.

2.3 Health risks

Besides from finding the best product, another problem consumers might not even know they face is allergies. There is a chance that a mixture of different product ingredients could trigger skin reactions. Identifying these ingredient conflicts requires extremely in-depth knowledge which most consumers do not possess, marking a significant and often unrecognized threat.

Our project uses LLM reasoning to identify conflicts among selected product ingredients. Not only does it lower the risks of having allergies, it can potentially educate the users in case of making future purchasing decisions.

3. Importance of Agentic AI

3.1 Continuous and autonomous monitoring

Agentic AI provides continuous, hands-off tracking of facial condition changes through regular image capture and analysis. This removes the need for users to manually log changes or manually input data. Over time, the system builds a profile of the user's skin condition, notifying them of any significant changes or conditions that need attention. Agentic AI's role in this is crucial as they help maintain consistency without the hassle of manually logging, monitoring and analyzing the data.

3.2 Real-time, complex data processing

After collecting data, Agentic AI can process a vast array of data streams simultaneously. This includes data like facial images, user histories, and product details. Agentic AI analyzes facial images in seconds, identifies specific skin conditions and problems, filters out the necessary product categories, and identifies any product combinations that may cause allergies. All of these are completed in mere minutes, easily outperforming any average manual review process.

3.3 Data-driven decisions

With the consolidated data, our Agentic AI takes different factors into consideration, such as user skin profile, history, and preferences. It will learn from each user interaction to understand what the user needs and wants. Instead of plain and monotone recommendations, our product responds to user desires and needs to provide more custom-made solutions.

4. Primary User Groups and Roles

4.1 Users

They are the primary stakeholders of our Personal Cosmetic Assistant project. They interact with our LLM-powered agents to receive personalized cosmetic recommendations through facial analysis, and product comparisons. The system takes users of different age groups into consideration, including multiple respective factors like skincare needs and concerns.

4.2 Data Analytic Team

The team utilizes our system's LLM capabilities and collects data to generate insights. Data can be used to improve recommendation algorithms. They also work with technical staff to continuously enhance the agent's performance through data-driven analysis.

4.3 Technical staff

They maintain, update, and continuously improve our Agentic AI. They manage the system infrastructure, update the AI models, maintain the product database, and implement new features. They are the core user group other than the users, that makes sure the system works smoothly and remains accurate, secure and reliable.

5. Key Capabilities and Core Features

Core Features

These core features form the foundation of our Personal Cosmetic Assistant. Each feature works independently and integrates with each other to create a pipeline. The system cannot function without any of these features. It also demonstrates our software's capability as an Agentic AI to help our users select the optimal cosmetic products.

Core Feature 1: **Upload Facial Images**
[Users capture or upload facial images for analysis.]
Agentic AI Responsibilities:

- Guides the users with preferred language instructions
- Provides real-time feedback on image quality (retake if subpar)
- Camera with AR guidance (for aligning)
- Image quality assurance using AI models
- Encrypted storage (cybersecurity related)

Core Feature 2: **Facial Analysis**
[Automatic AI analysis of skin conditions without manual input.]
Agentic AI Responsibilities:

- Analyze using multi-modal LLM + computer vision
- Identify skin type and concerns
- Generate preferred language reports
- Keeps track of facial analysis records and identify changes over time

Core Feature 3: **Database for Cosmetic Products**
[Well-organised and secure database for storing all products with their respective details.]
Agentic AI Responsibilities:

- Query using natural language
- Identify and compare products
- Consolidate and explain product details and differences

Core Feature 4: **Automatically Update Database**
[Agentic AI continuously updates product database.]
Agentic AI Responsibilities:

- Scans websites for new products
- Uses LLM to extract product data (prices, ingredients, etc.)
- Categorizes products and verifies data accuracy
- Insertion into database

Core Feature 5: **User Preference Input**
[Conversation to capture preferences using preferred language.]
Agentic AI Responsibilities:

- Intelligent follow-ups
- Identify conflicts or confusions in preferences
- Learn user preferences by analyzing patterns and behaviors

Core Feature 6: **Ingredient Conflict Analysis**
[Real-time conflict detection with LLM reasoning.]
Agentic AI Responsibilities:

- Identify ingredient conflicts to prevent allergy issues
- Provide detailed explanation on why these conflicts occur
- Provide alternative solutions
- Answer any user questions

Core Feature 7: **Product Recommendations**
[AI-powered personalised recommendations.]
Agentic AI Responsibilities:

- Takes analysis and preferences into consideration
- Query database with product comparison
- Provide explanation for its decisions

Core Feature 8: **Skincare Routine Tracking and Feedback**
[Record and analyze daily skincare activities to provide personalized improvement suggestions.]
Agentic AI Responsibilities:

- Monitor user skincare logs and detect changes in skin condition over time
- Generate visualized trend reports for progress tracking
- Identify ineffective or harmful product combinations
- Offer adaptive suggestions for improving skincare routines

Optional Features

These features would enhance user experience but are not a must–have for the basic functionalities of our system. The function can function without any of these features. They are potential enhancements for when we have extra budget and resources.

Optional Feature 1: **Price Comparison**
[Compare prices across retailers in real-time.]
Agentic AI Responsibilities:

- Query multiple retailers
- Show comparisons
- Explain comparisons
- Recommend best overall deal

Optional Feature 2: **Budget Management**
[Tracks spending and optimises costs.]
Agentic AI Responsibilities:

- Analyze spending patterns
- Provide insights on spending habits
- Notify users when spending reaches budget limits

Optional Feature 3: **Product Filter**
[Advanced filtering.]
Agentic AI Responsibilities:

- Apply multiple filters when querying
- Provide suggestions when filtered results are limited

6. Ad-Hoc Approach

6.1 **Upload facial images** (Ethan)

When the camera is powered on within the app, the facial image capture system is enabled. The system will provide the users with an overlay guide of how to align their faces. Once properly aligned, the user has to press a button in order to capture a photo of their face. Next to the capture button is an option to toggle the camera flash for low light environments.

6.2 **Facial analysis** (Ethan)

Once the facial image is captured, AI analyzes the photo and presents a summary of the user's face condition. The overview screen includes multiple tabs with detailed results. Such tabs may include face shape, skin assessment, facial features, color analysis etc. This layout ensures users can easily access detailed feedback on each major aspect of their facial condition after AI analysis.

6.3 **Database for cosmetic products** (Sean)

The NoSQL database stores cosmetic products from various sources which will be used to simulate a virtual facial image when the customer applies these products. The database includes - price, quantity, cosmetic description, brand, and other important cosmetic information.

6.4 **Automatically get cosmetic products and put them into the database** (Sean)

The application intelligently crawls cosmetic products from various websites into the database automatically. The system focuses on 1 to 2 e-commerce platforms to get the data from due to simplicity. Whenever a new product has been released or an old product is out of stock, the system will be updated real time.

6.5 **User preference input** (Yue)

Users can enter key information based on their daily skin care and makeup habits, such as allergy history, preferred skin care quality, makeup needs, ingredient preferences, and special conditions such as pregnancy. The system builds user portraits based on information analysis, which is convenient for the system to recommend more accurate products for users in the future.

6.6 **Ingredient conflict analysis** (Yue)

Based on the user's allergy history, analyze the cosmetic ingredients to determine whether there is a conflict between different ingredients, such as retinol and fruit acids which cannot be used at the same time. The system can give scientific usage suggestions or replacement product plans to provide users with safety guarantees.

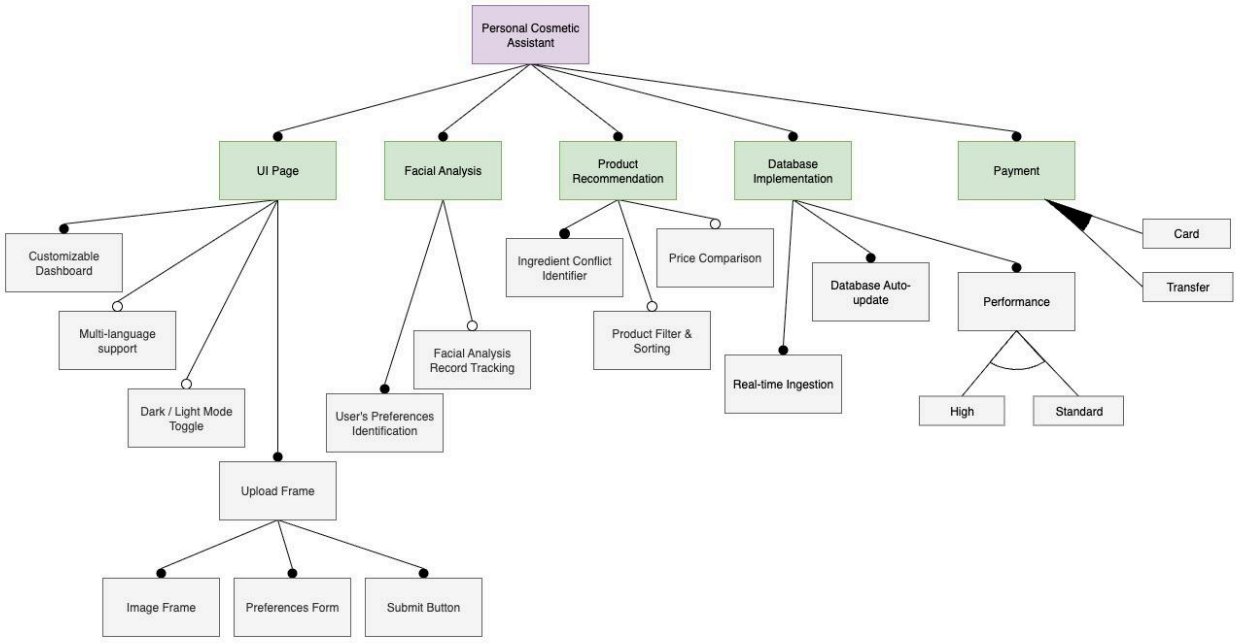
6.7 **Product recommendations** (Evangeline)

The system provides personalized, ranked cosmetic product suggestions that are tailored to each user’s unique profile and preferences. Unlike generic online shopping recommendations, this feature integrates facial analysis, user input, and product database intelligence to generate precise, evidence-based recommendations.

6.8 **Skincare routine tracking and feedback** (Yiming)

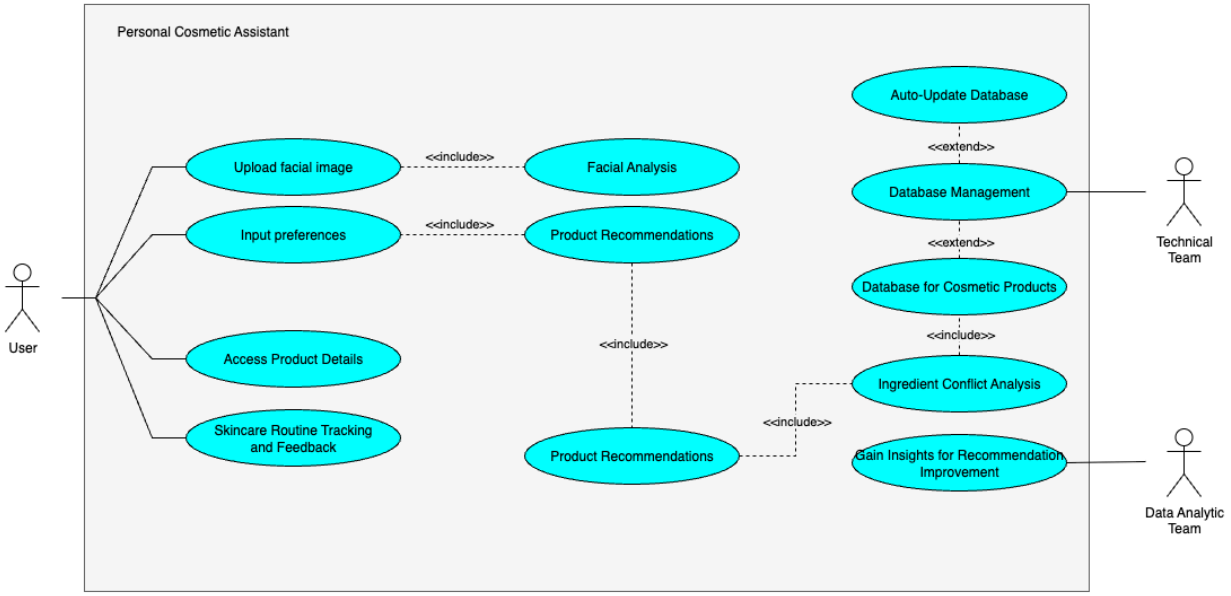
Users can log their daily skincare activities and skin conditions within the app. They can record products used, rate skin conditions (such as dryness or irritation), and upload facial images for progress tracking. The system analyzes these entries to generate personalized trend reports and provides improvement or adjustment suggestions for the skincare plan.

7. Feature-Oriented Approach



The feature diagram of Personal Cosmetic Assistant consists of 5 primary components: UI page, Facial Analysis, Product Recommendation, Database Implementation, and Payment. The UI Page supports customizable dashboards, multi-language options, dark/light mode toggling, and an upload frame for user images and preferences. The Facial Analysis component identifies user preferences and maintains analysis records. Product Recommendation includes ingredient conflict identification, price comparison, and product filtering. The Database Implementation component supports real-time ingestion, automatic updates, and performance control. Finally, the Payment module provides options for card and transfer payments.

8. Use case overall diagram



This use case diagram illustrates how different actors interact with the Personal Cosmetic Assistant system. The User can upload facial images, input preferences, access product details, and track skincare routines to receive personalized product recommendations. The Technical Team manages the system's backend by maintaining and auto-updating the database for cosmetic products. Meanwhile, the Data Analytic Team performs data mining to gain insights about user behavior, helping to identify ways to improve the recommendation system in the future. Together, these interactions ensure the system provides accurate, personalized, and continuously updated cosmetic recommendations.

9. Use Case Approach

9.1 Use Case 1 (Yue)

Name	Goal	Level	Precondition	Success condition	Failure condition	Trigger
Upload Facial Image	Provide a clear facial image for skin analysis	User	User has access to camera or image file	Image successfully uploaded and recognized	Image not uploaded or not recognizable	User selects "Upload Image" function

Scenario Steps:

1. User selects “Upload Image” option.
2. System prompts user to choose image source (camera or file).
3. User provides image.
4. System checks image clarity and recognizability.
5. System accepts and stores image.

Extension scenario: System requests re-upload.

9.2 Use Case 2 (Ethan)

Name	Goal	Level	Precondition	Success condition	Failure condition	Trigger
Input Preferences	Capture user’s skincare/makeup	User	User has logged into the system	Preferences saved successfully	Preferences not stored or input rejected	User opens reference input page

Scenario Steps:

1. User navigates to “Preferences” page.
2. System displays available preference categories (e.g., skin type, product types, ingredients to avoid).
3. User selects and inputs preferences.
4. System validates input format.
5. System saves preferences, confirmation message is shown.

9.3 Use Case 3 (Sean)

Name	Goal	Level	Precondition	Success condition	Failure condition	Trigger
Access Product Details	View information about a recommended product	User	Product recommendation list is generated	Product details displayed to user	Product details not accessible	User clicks on product from recommendation list

Scenario Steps:

1. User views recommendation list.
2. User clicks on a product.
3. System retrieves product details from database.
4. System displays details (ingredients, reviews, price, usage instructions).
5. User browses details.

9.4 Use Case 4 (Yiming)

Name	Goal	Level	Precondition	Success condition	Failure condition	Trigger
Manage Database	Maintain integrity and accuracy of product / ingredient database	Technical Staff	Admin authentication successful	Database updated without errors	Database update fails or data inconsistency occurs	Technical staff use database management function

Scenario Steps:

1. Technical staff logs into admin portal.
2. System verifies authentication.
3. Staff selects database management function.
4. Staff performs database operation.
5. System validates and applies changes.

9.5 Use Case 5 (Evangeline)

Name	Goal	Level	Precondition	Success condition	Failure condition	Trigger
Gain Insights for Recommendation Improvement	Understand user behavior and product interactions to identify factors influencing purchasing decisions	Data Analytic Team	Historical user data and feedback available	Meaningful insights about user behavior and recommendation performance are obtained and documented	Data analysis yields insufficient or unclear insights for improving the recommendation system	The Data Analyst initiates a data mining process to analyze user behavior

Scenario Steps:

1. The Data Analyst gathers user data such as product views, purchases, and feedback from the database..
2. The analyst performs data mining and pattern analysis to explore correlations between recommendations and user purchasing behavior.
3. The analyst identifies key factors that influence why users accept or reject recommended products.
4. The findings are summarized into actionable insights for improving future recommendation models.
5. The insights are reported to the development for later implementation.

Extension scenario: If no significant patterns are found, the analyst records observations and recommends additional data collection for future analysis.

9.6 Use Case 6 (Yiming)

Name	Goal	Level	Precondition	Success condition	Failure condition	Trigger
Skincare Routine Tracking and Feedback	Allow users to record their skincare product usage and receive AI-generated improvement feedback over time	User	User has a registered skincare plan and completed initial profile setup	Skincare usage and condition data successfully recorded; system generates a visible progress report	Data submission fails or incomplete; system unable to process trend analysis	User opens the "Daily Skincare Log" and submits their skincare record

Scenario Steps:

1. User opens "Daily Skincare Log."
2. System displays form for product selection, condition rating, and photo upload.
3. User fills in products used, rates skin condition, and optionally uploads photos.
4. System validates inputs and checks photo quality.
5. System saves the log and generates a trend update.

Extension scenario: If required fields are missing or photo quality is poor, the system prompts user to correct and resubmit.

10. Architecture Analysis and Design(s)

10.1 Model of Computation (MoC)

Our project will adopt a **Structure-Dominant** MoC. Personal Cosmetic Assistant has multiple independent components that act as containers of information (state) and behavior, which react in response to stimuli, such as facial analysis, database query, conflict identification, product recommendation. These agents communicate asynchronously to form the basic functionality of the system. Therefore, the stateful components include user profiles and preferences, facial analysis records, product database and user interaction history.

In our project, the system does not follow a fixed procedural flow. Instead, it is dependent on the user's actions. For example, the system will respond when the user uploads an image or enters their preferences. It has an event-driven nature. And most importantly, since our agentic AI needs continuous and autonomous monitoring, this aligns perfectly with the actor-based structure-dominant model.

10.2 The 4+1 Viewpoint

Logical View

The Personal Cosmetic Assistant consists of multiple components that work together. This includes facial image processing, product recommendation, automatically updating products in the database, and a payment system. Each component reacts on user or system events such as when a user uploads their images or a user orders a product.

Implementation View

The system is organized into separated functional modules, allowing easier to maintain the system. The presentation layer consists of Web Applications that users interact with. The security layer manages the account to login to the system. The application layer contains modules for recommendation system, payment system, and repositories accessing products in the database. This modular design promotes scalability, reusability, and flexibility, enabling developers to modify or upgrade individual components without affecting the rest of the system.

Process View

Each agent processes images and suggests a product asynchronously and concurrently. When a user uploads an image, the facial analysis begins and sends the result to the recommendation process. This event retrieves relevant products from databases, checks the constraint, and gives suggestions. This asynchronous process model ensures responsiveness, efficient resource usage, and real-time updates while allowing multiple operations to run simultaneously

Deployment View

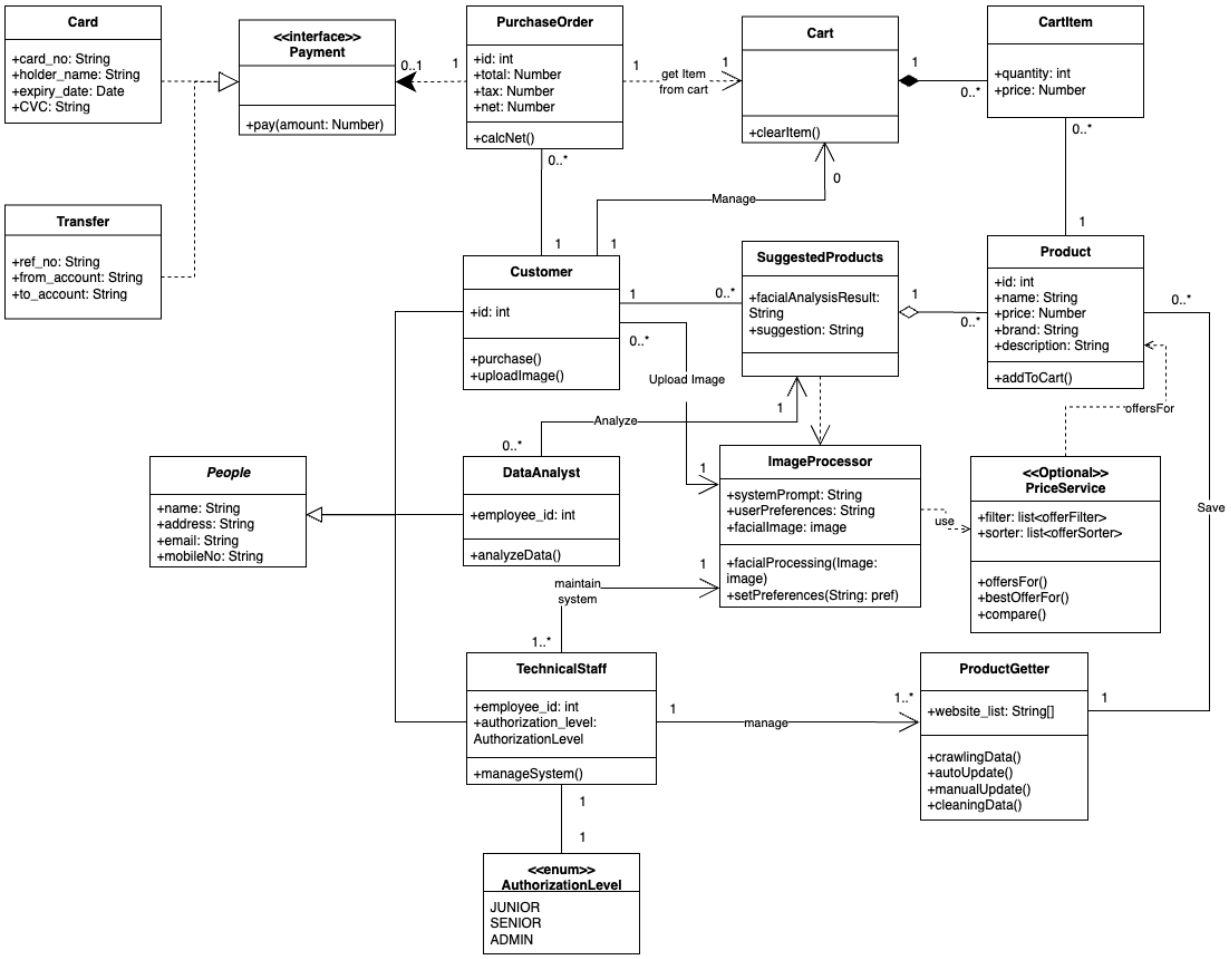
The Personal Cosmetic Assistant is deployed on the cloud. All main services, such as facial analysis, recommendation, and product data management, run on cloud servers. The system stores information in online databases and allows users to access it through a web or mobile interface. This setup makes the system easy to scale, reliable, and available anytime from anywhere.

Use-Case View

A key use case driving architecture starts when a user uploads their facial image into the system. The facial analysis system processes the image to identify skin type and features, after which a recommendation system retrieves suitable products from the database. Finally, the system presents suitable cosmetic products to a user.

11. Elementary Structure Modelling

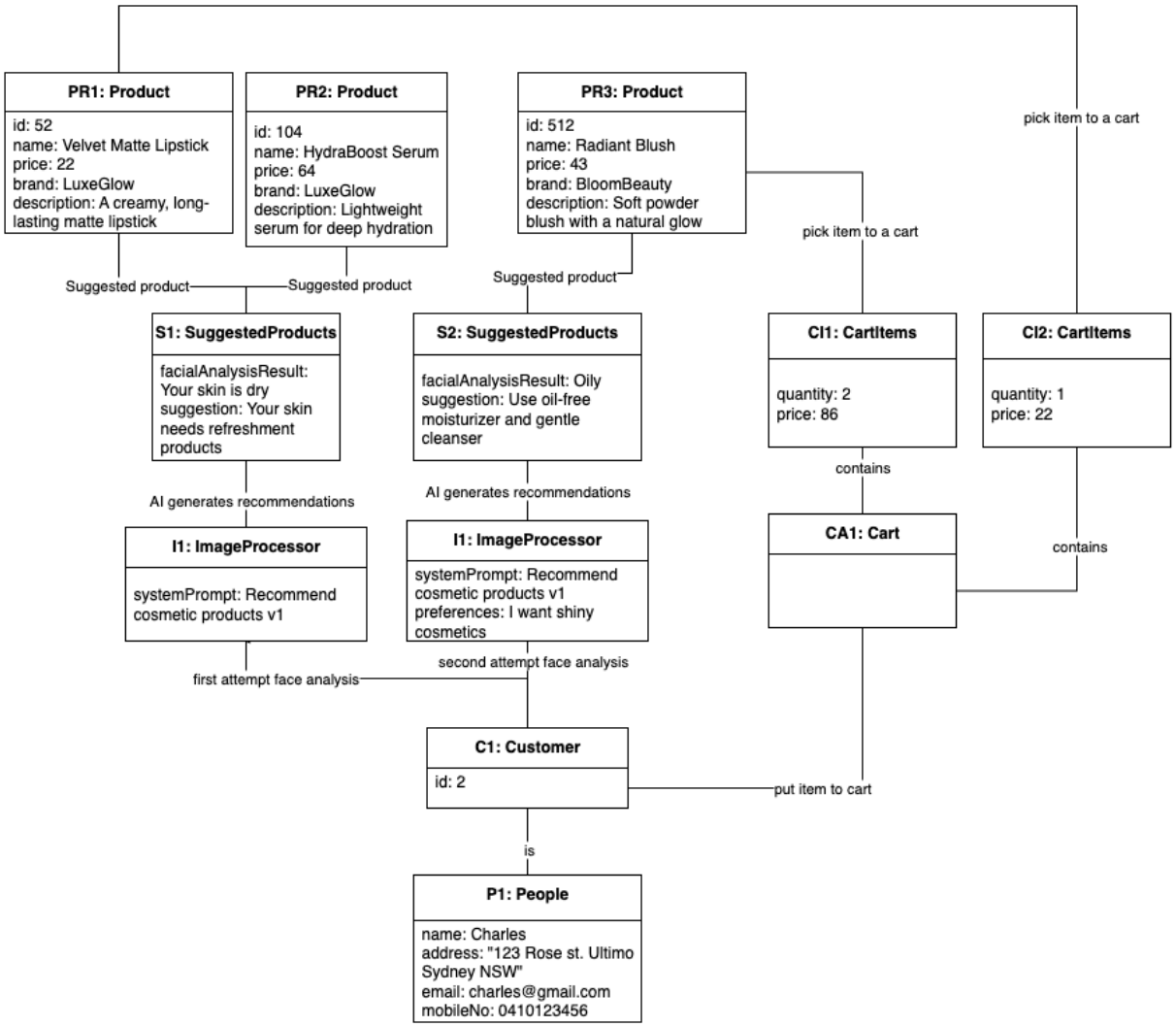
11.1 Class Diagram



This class diagram describes the static structure of a cosmetic recommendation system. The Customer, DataAnalyst, and TechnicalStaff classes inherit similar behaviors from the People class. The Customer class has associations with PurchaseOrder and Cart, which are related to the Payment service. The SuggestedProducts and ImageProcessor classes are the core components of the system, where facial images are processed and products are recommended to the user. The DataAnalyst class evaluates the SuggestedProducts to determine whether the product has been purchased, checking the performance of the recommendation system. The TechnicalStaff class manages the retrieval of cosmetic products from multiple websites by using ProductGetter to handle product information, and maintain the ImageProcessor.

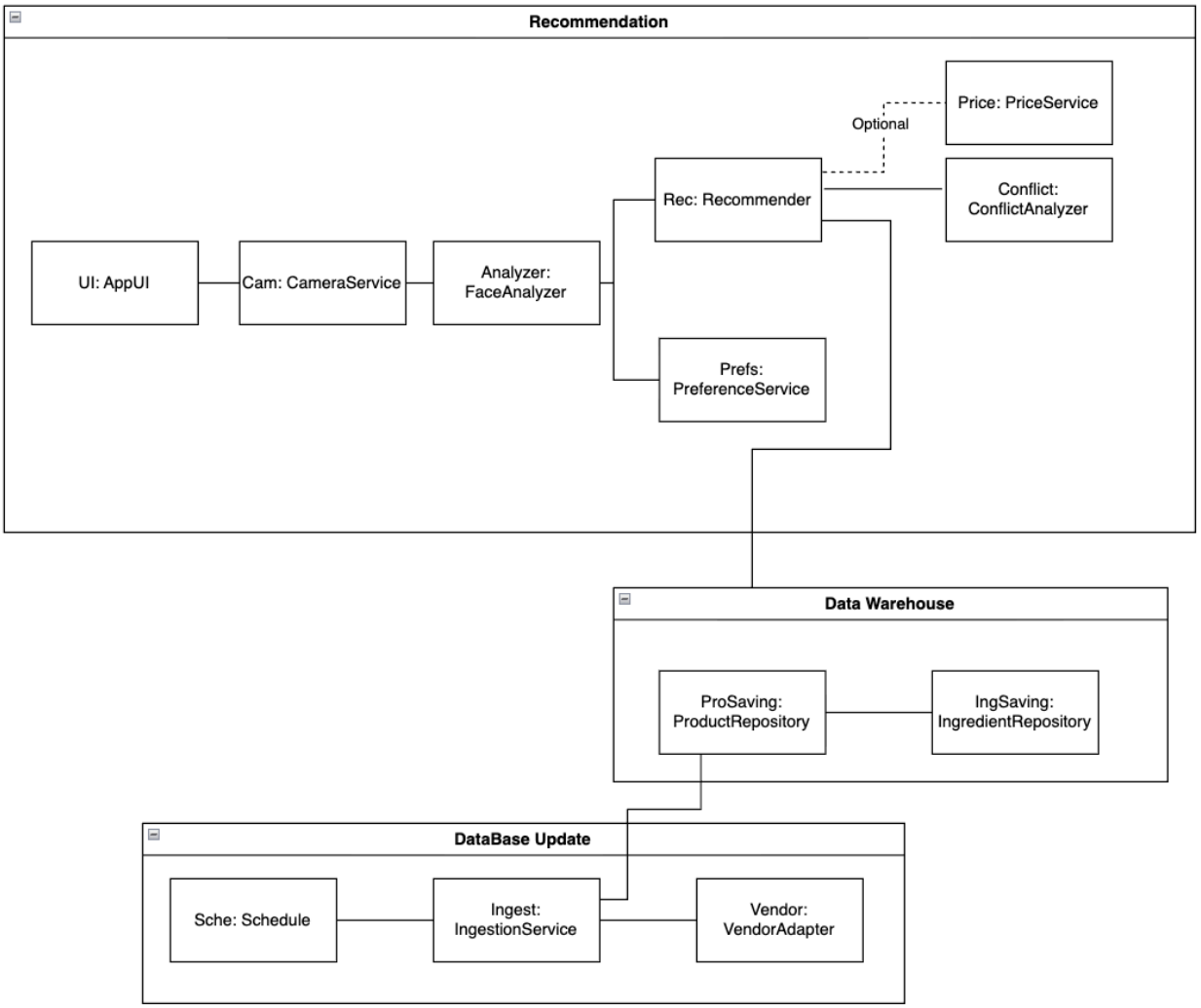
12. Complex Structure Modelling

12.1 Object diagram



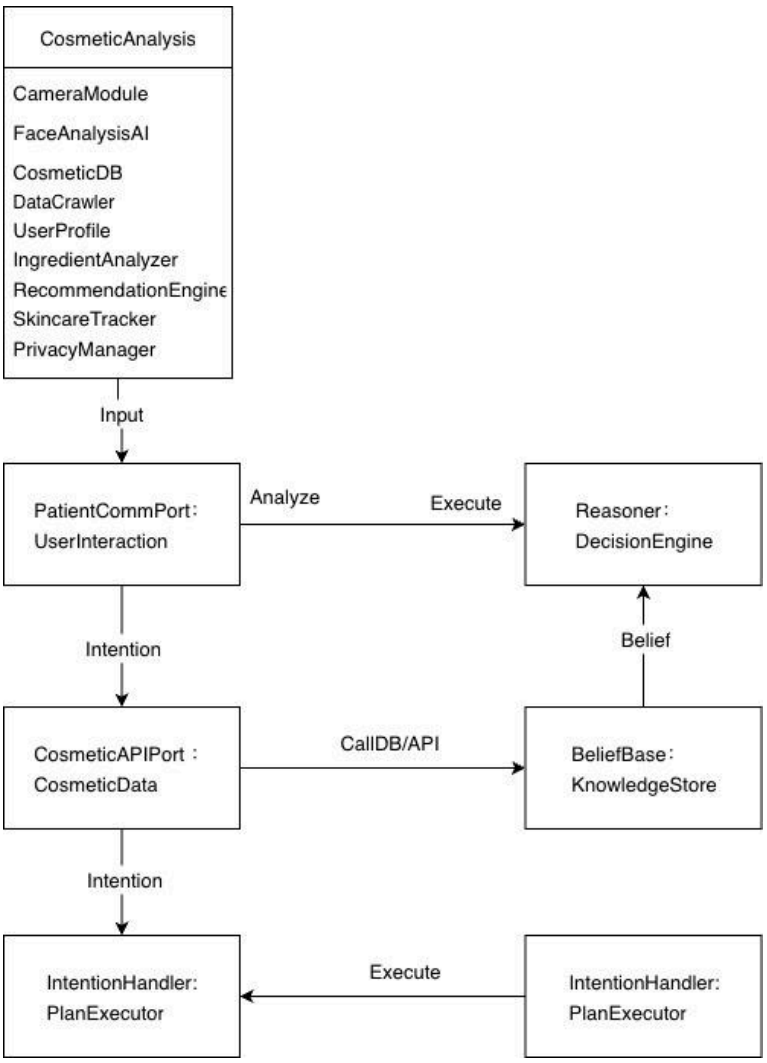
The object diagram represents a snapshot of the system's behavior when a user uploads their facial image to the cosmetic recommendation system. The user can upload or update their preferences multiple times, and the ImageProcessor analyzes the facial image and retrieves relevant data from the database. Based on the analysis, the system generates product suggestions, and the user may add items of interest to the Cart. In this diagram, the payment components are not shown, as the user has not yet completed the payment process.

12.2 Collaboration diagram



The system is divided into **three collaboration domains:**
Recommendation: mainly responsible for facial image analysis, user preference acquisition, conflict component verification, and result generation
Data warehouse: primarily stores product and ingredient data
Database update: Store supplier data and update the database regularly.
In the **collaboration section**:
Recommendation: UI → Camera → Face Analysis → Recommender, the recommender simultaneously obtains user preferences and component conflict analysis, and can choose whether to connect to the price comparison service.
Data warehouse: Contains product and ingredient data, providing information about candidate products and ingredients to the recommender.
Database update: The scheduler triggers the warehousing service to fetch data from external suppliers and write it into the product database.
The three parts do not directly depend on each other, only share data through the data warehouse.

12.3 Structure Class



The **CosmeticAnalysis** structured class diagram illustrates the internal architecture of the **Intelligent Skincare Agent**.

Externally, the system includes two main ports:

- **UserInteraction Port** — receives user inputs such as facial images and skin profile data.
- **CosmeticData Port** — communicates with external product databases and AI interfaces.

Internally, the system consists of several key subcomponents working together to perform data analysis and personalized recommendation tasks.

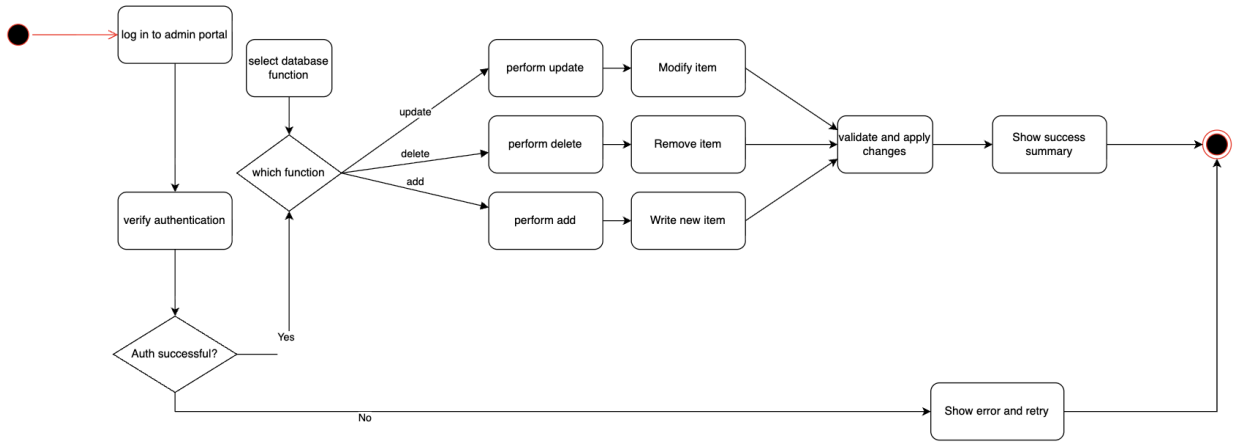
- The **DecisionEngine** analyzes user input and generates skincare plans based on reasoning rules. It retrieves historical data and product information from the **KnowledgeStore** to support decision-making and recommendation generation.
- Once a plan is created, the **PlanExecutor** calls external databases or APIs via the **CosmeticAPIPort**, executes the recommendation plan, and returns results to the user.

The diagram presents the system's complete runtime workflow — **Input → Analyze → Belief → Intention → Execute**, clearly showing the collaboration among reasoning, memory, and communication modules.

This structured class modeling effectively demonstrates how internal components are organized and interact during execution, reflecting the overall architectural logic of the intelligent skincare recommendation system.

14. Activity Diagrams

14.1 Data analysis team manages the database (Yue)



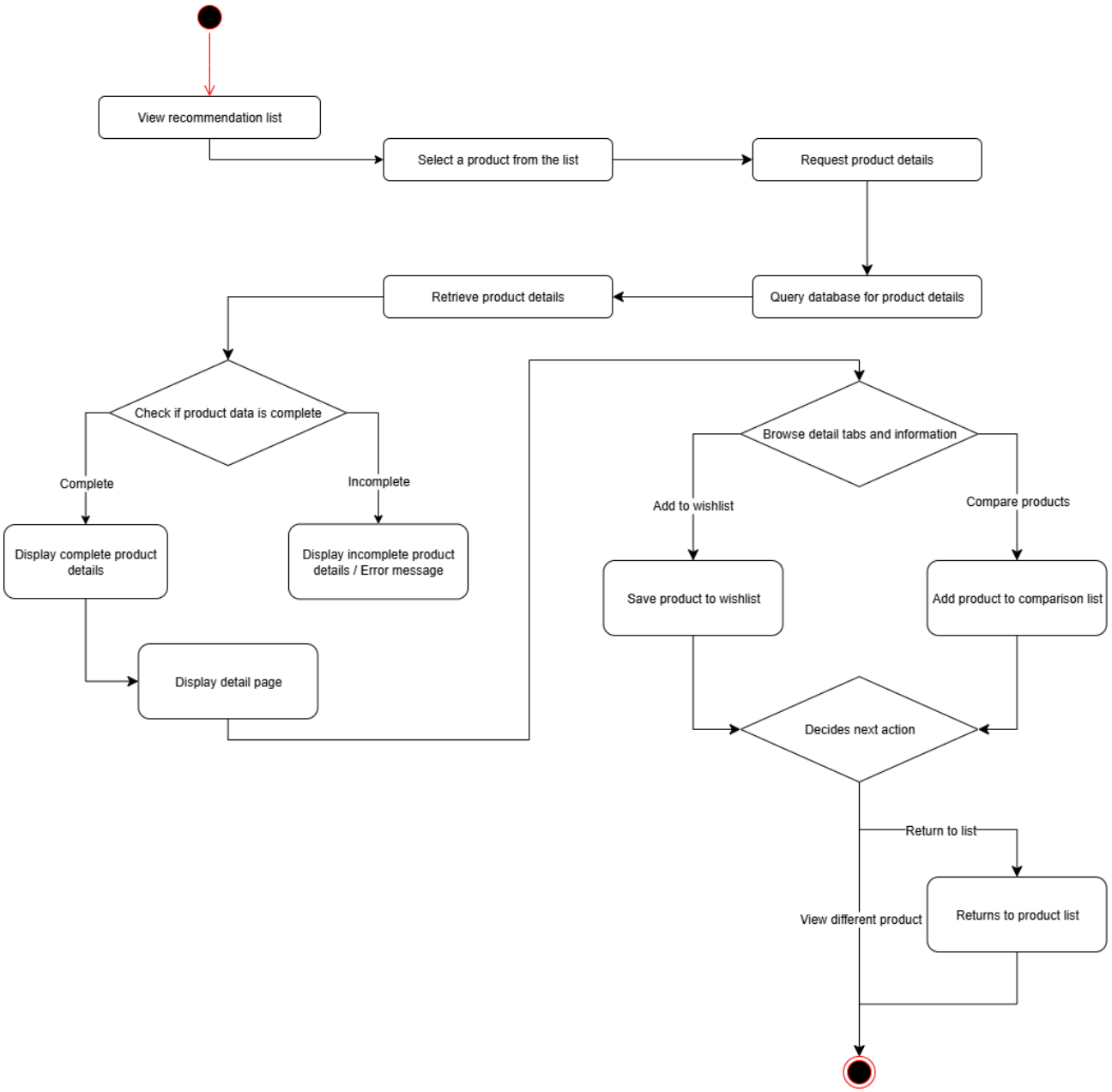
This active diagram shows the entire process of the data analysis team managing a database. The data analysis team logs in to the management portal and performs identity verification; if verification fails, enter "Error and Retry" until completion.

After verification, select the database function (Add/ Delete/ Updata).

Branch execution:
Update → Modify entry; Delete → Remove entry; Add → Write new entry.

All branches converge to verify and apply changes; if successful, display the "Success Summary" and end, if failed, enter "Error and Retry" and return to the end of the process.

14.2 User access product details (Ethan)



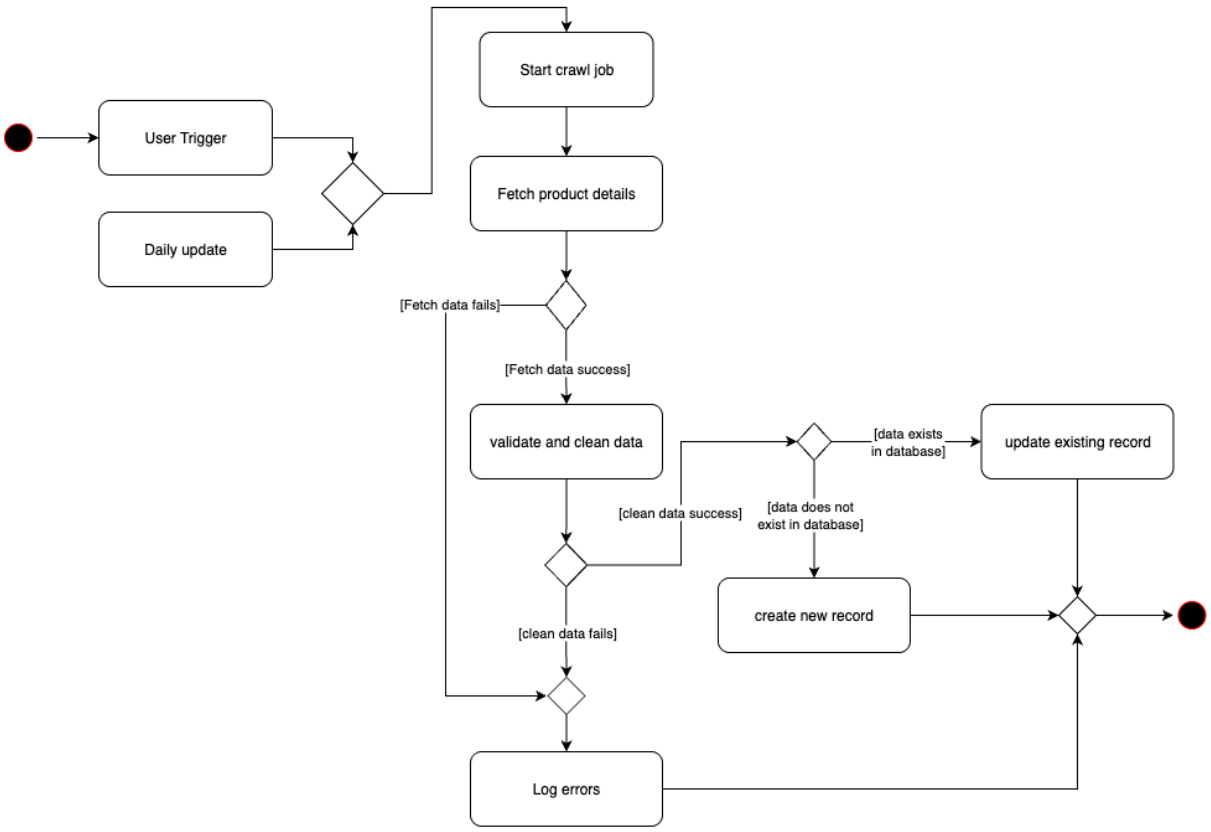
This diagram visualizes the activity stages when “user access product details”.

First, the prerequisite is that the user is at the product recommendation list. The user can then select one product from the list. This will initiate a query process for retrieving product details. If the product data is complete, it will display the details but if the data is incomplete, an error message will be displayed.

Moving on, the detail page has different tabs where the user can navigate in and out of. Depending on the user’s course of action, products can either be saved to the wishlist, or compared between each other.

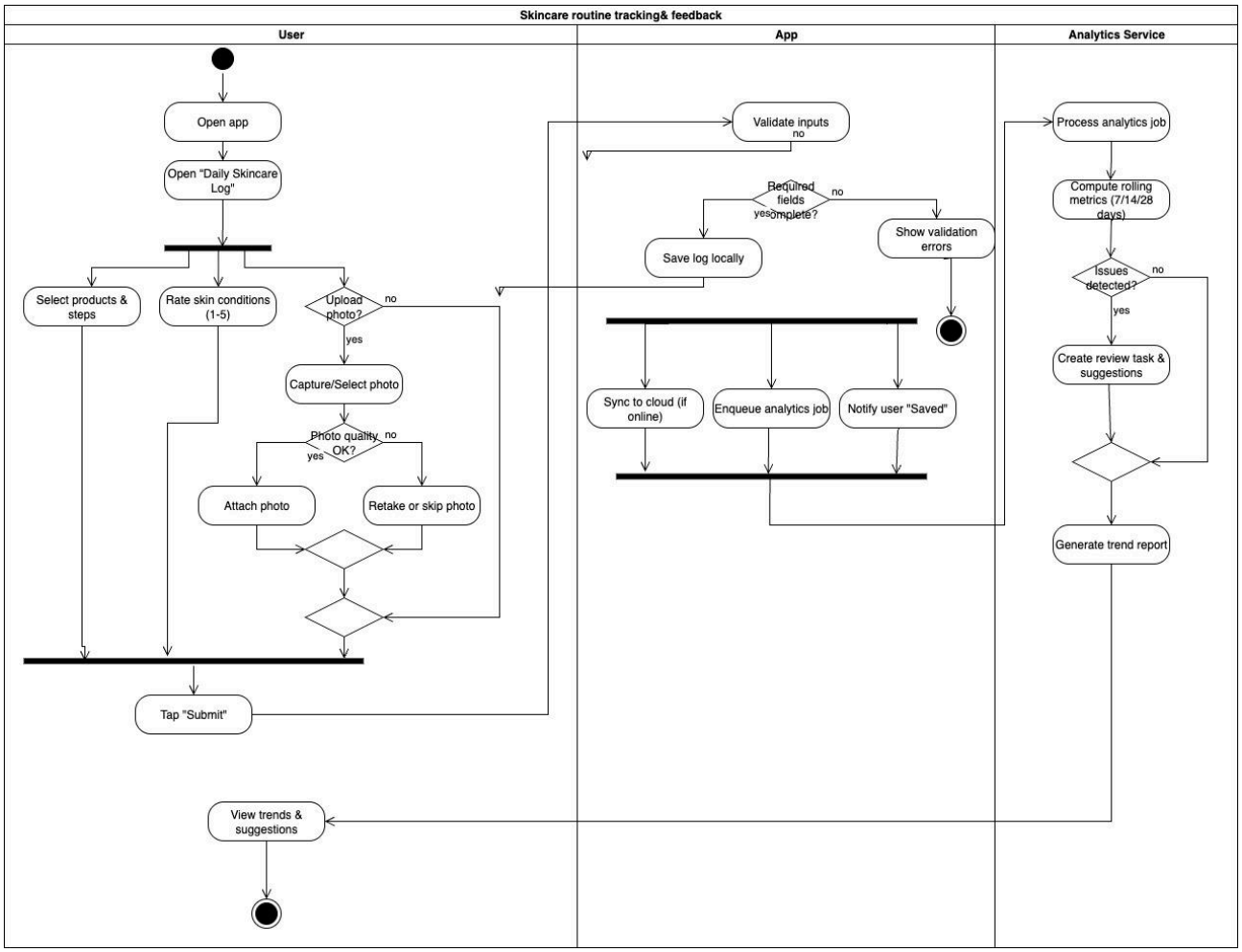
Eventually, the user can choose to look at another product or go back to the list.

14.3 Cosmetic product ingestion system (Sean)



The activity diagram shows the process of how a product is updated in the database. Firstly, the system retrieves the data from websites when a user triggers or it is automatically updated daily. The system fetches product details, validates, and cleans the data before updating new products into the database. When the error occurs during these steps, the system logs the errors and terminates the system. For successfully validated data, the system checks whether the product already exists in the database - if it does, the existing record is updated; if not, a new record is created

14.4 Skincare Routine Tracking & Feedback (Yiming)



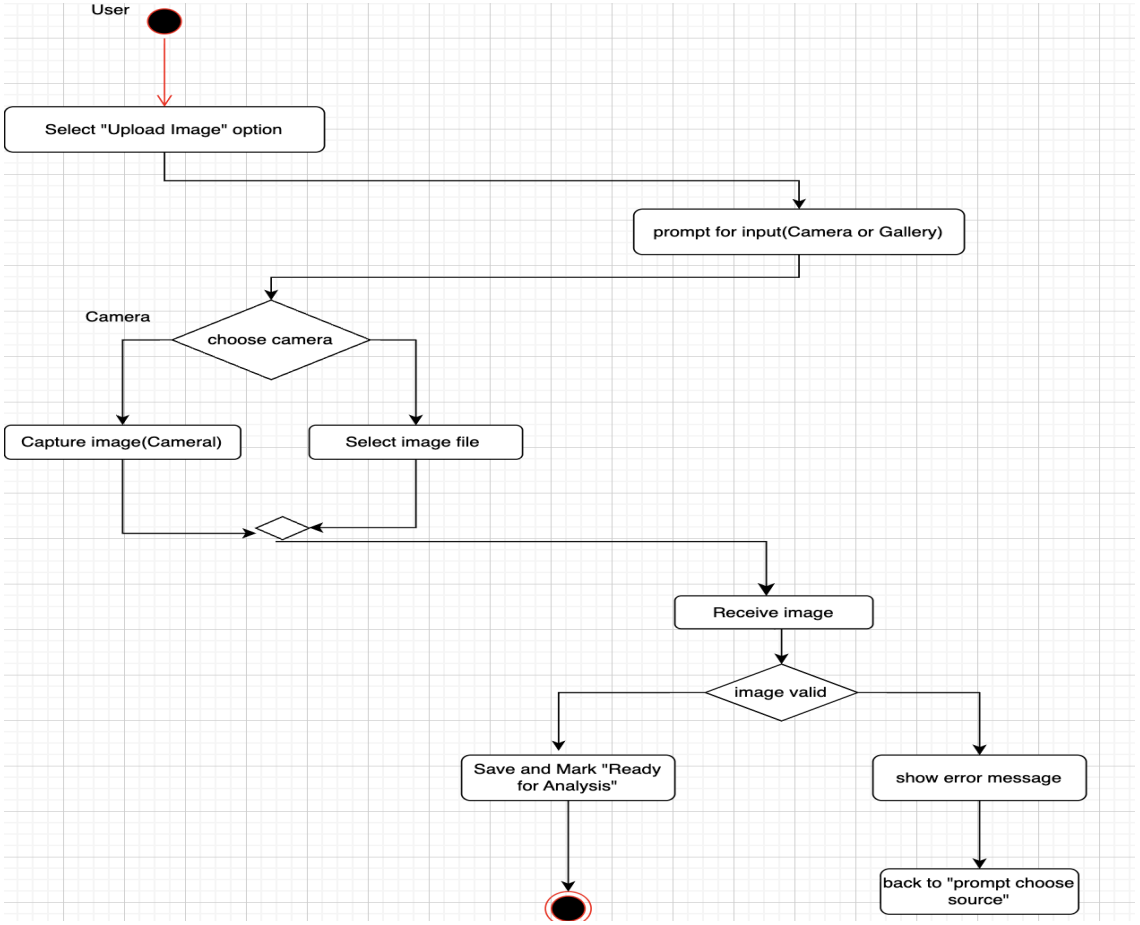
This activity diagram shows the main flow of the **"Skincare Log & Feedback"** feature.
It records users' daily skincare routines and skin conditions, generates trend reports, and provides personalized suggestions.

Roles:

- **User:** Selects products, rates skin condition, uploads photos, views feedback.
- **App:** Validates input, saves locally, syncs to cloud, sends for analysis.
- **Analytics Service:** Analyzes data, detects trends/anomalies, returns suggestions.

Flow:
User fills the log → App saves & submits → Service analyzes → Results return → User views feedback.
Offline data is cached and synced later. Photo upload is optional. All data is encrypted and privacy protected.

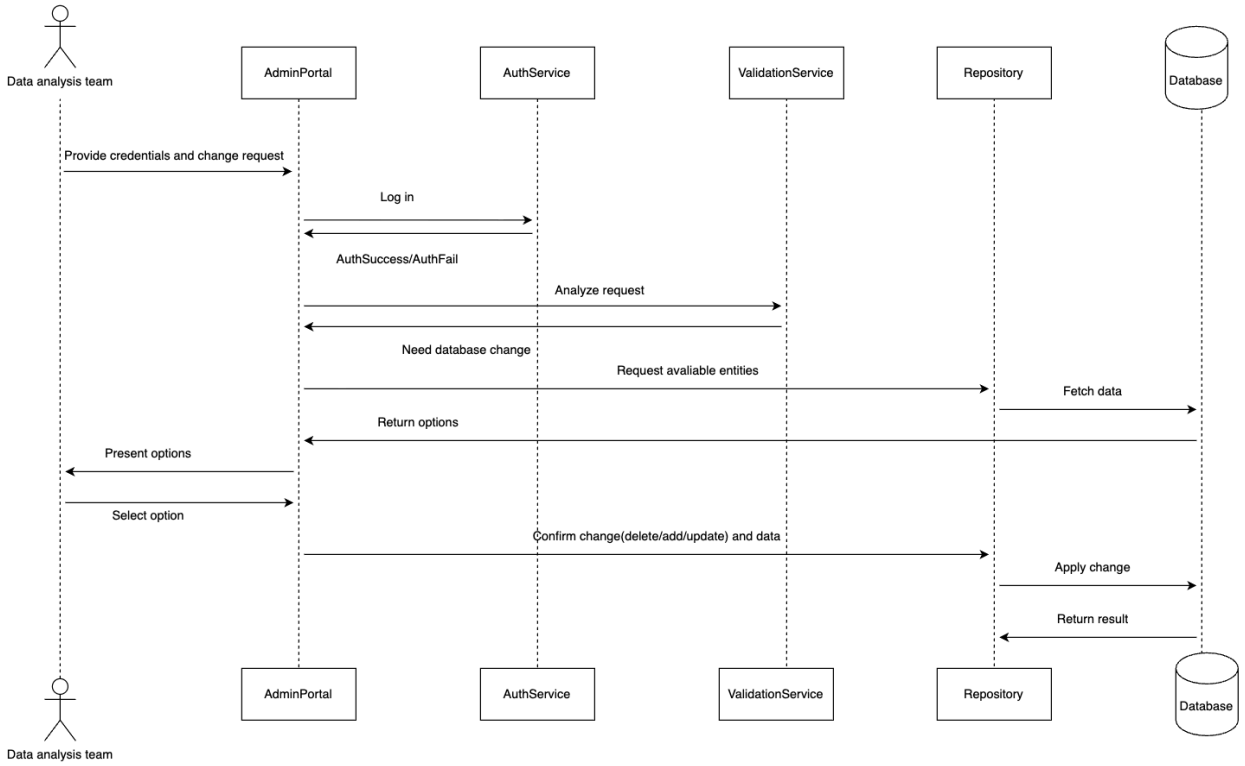
14.5 User upload the image (Evangeline)



This Activity Diagram describes the process of how the user uploads a facial image in the *Personal Cosmetic Assistant* system.
The flow begins when the User selects the “Upload Image” option. The system then prompts for input, asking whether the user wants to use the Camera or choose an existing file from the Gallery.
If the user selects the camera, the system captures a new image; if the gallery is chosen, the user selects an existing image file.
Once the image is received, the system checks whether it is valid.
If the image passes validation, it is saved and marked “Ready for Analysis.”
If the image is invalid, the system shows an error message and returns to the prompt for input.
This diagram outlines the main user actions and system responses during the image upload process, ensuring a clear workflow for user interaction and input validation.

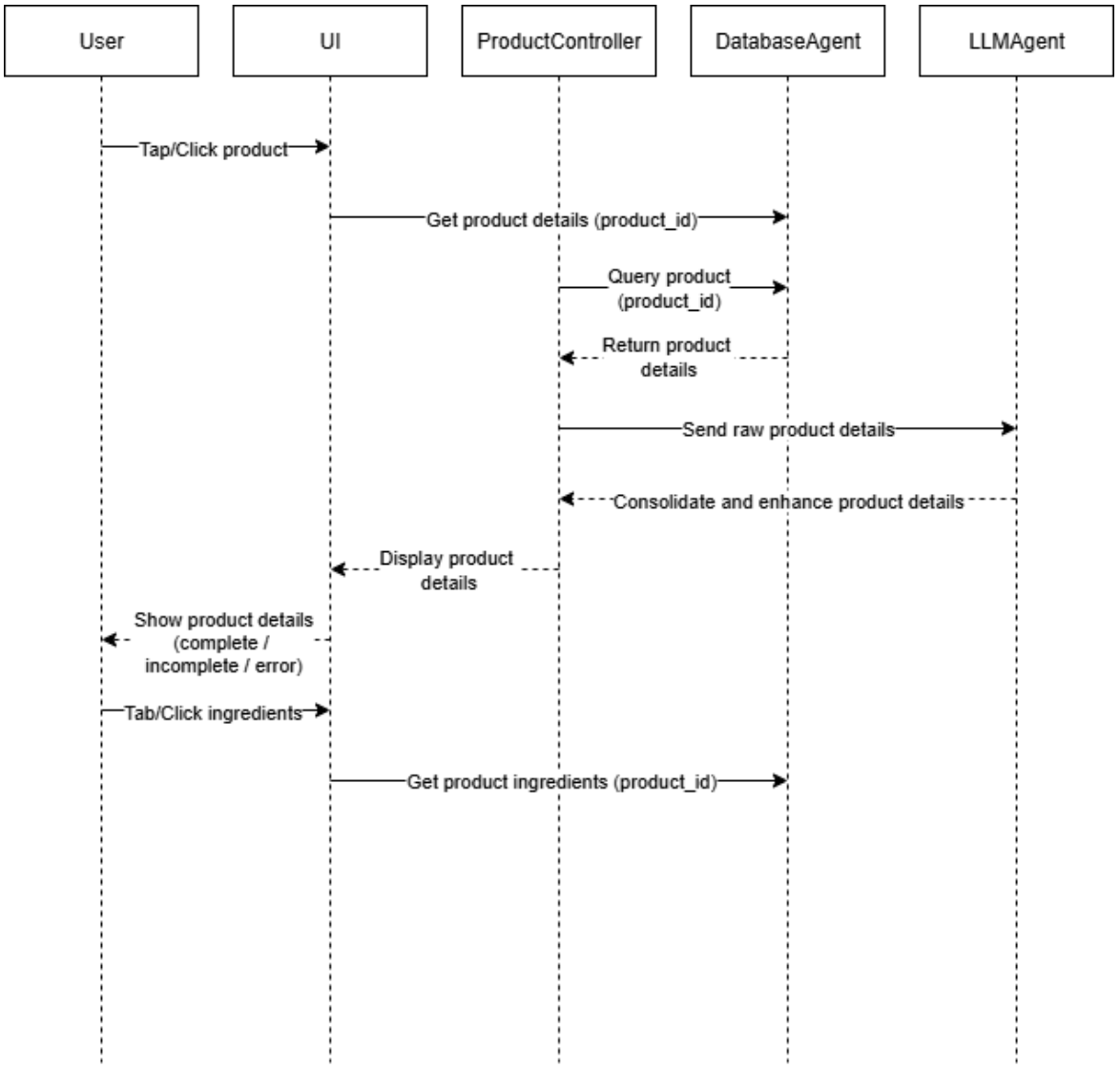
15. Interaction Diagrams

15.1 Data analysis team manages the database (Yue)



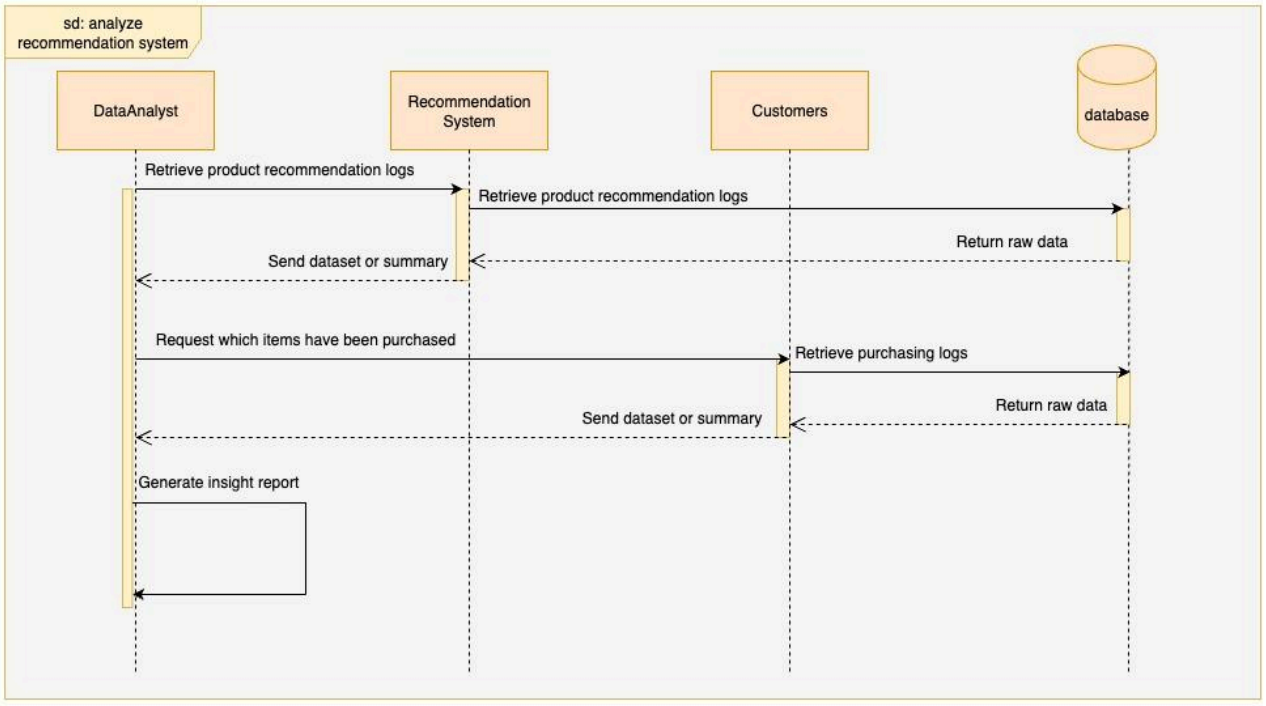
This interaction diagram shows the main process of “Data analysis team making changes to the database through the management portal".
Data analysis team members submit database change requests in the AdminPortal, AuthService authenticates them, and ValidationService analyzes the request and determines that database changes are needed. The portal then requests actionable entities from the Repository, which takes the number from the database and returns the options, and the portal returns the database change options to the person. The person selects and confirms the specific addition, deletion, or update operation, the Repository calls the database to execute the changes and returns the result, and the portal feeds back the execution result to the data analysis team.

15.2 User access product details (Ethan)



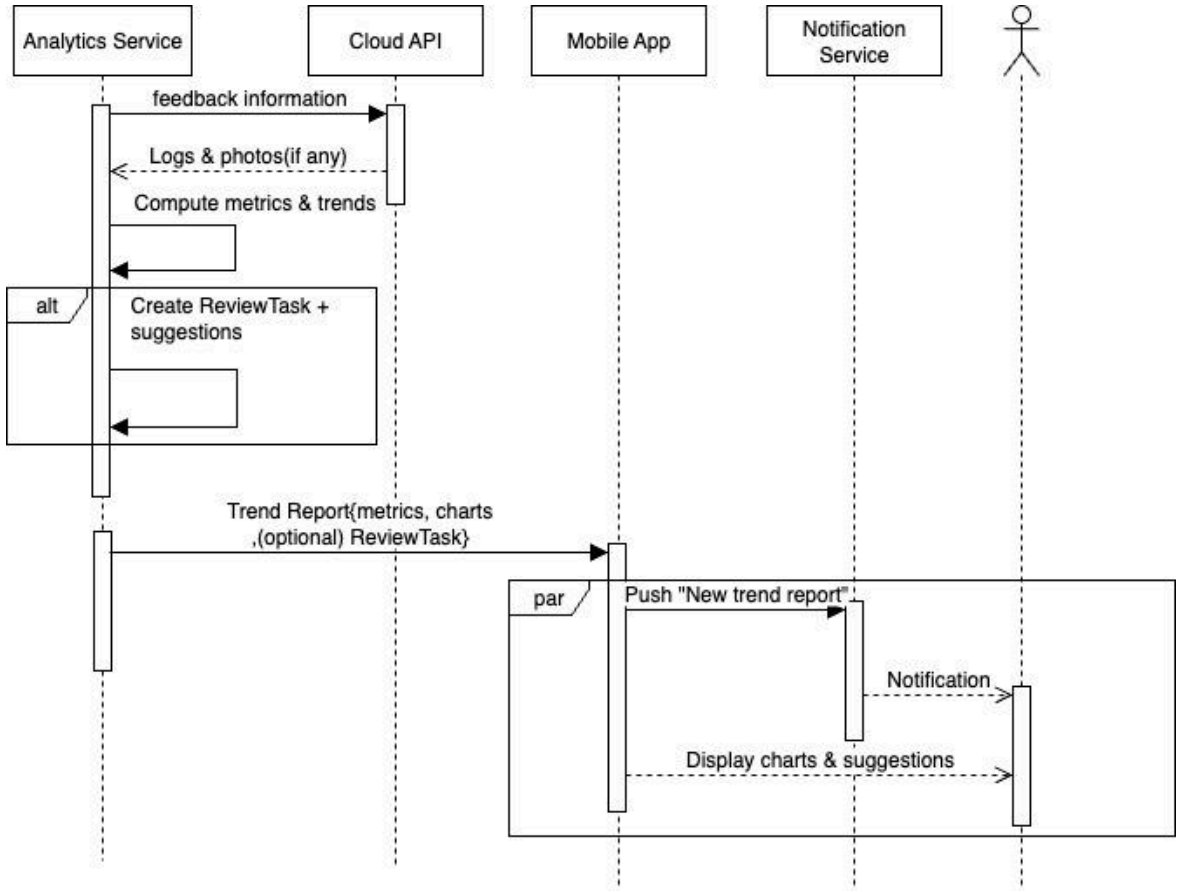
This diagram shows the interaction between different components. It represents the flow of the system when the user tries to access a product's details. The **User** is responsible for certain inputs like 'tap/click product' and 'tab/click ingredients'. The **UI** serves as the bridge between the user and the other components. It collects input and sends requests to other components. **ProductController** controls the product data before it's sent back to the UI for the user to see. **DatabaseAgent** controls what data is extracted and **LLMAgent** controls the logic and analysis for our services.

15.3 Data analyst analyze the recommendation system (Sean)



The interaction diagram illustrates the communication between the DataAnalyst, Recommendation System, Customers, and the Database. The DataAnalyst retrieves logs of product recommendations and purchase activities from the database through the recommendation system and customer modules. By comparing the recommended products with those actually purchased by customers, the analyst can evaluate the system's performance including the success and failure rates of product recommendations.

15.4 Skincare Routine Tracking & Feedback — Interaction Diagram Description (Yiming)

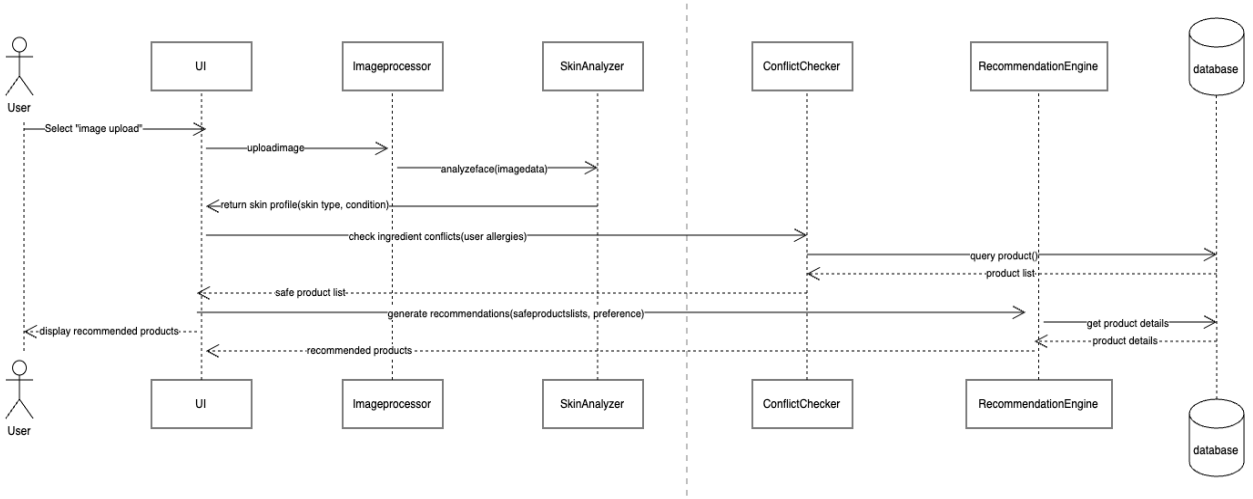


This interaction diagram illustrates the communication among components in the **skincare log feedback process**.

The **Analytics Service** receives log and photo data from the app, computes key metrics and trends, and generates review tasks and improvement suggestions when anomalies or product conflicts are detected.The system then sends the **trend report** (including metrics, charts, and recommendations) to the **Cloud API**, which delivers it to the **Mobile App**.The **mobile client** triggers the **Notification Service** to push a “New Trend Report” alert, allowing users to view visualized charts and personalized suggestions within the app.

This interaction process demonstrates the system’s **automation and real-time performance** in data analysis, report generation, and feedback delivery.

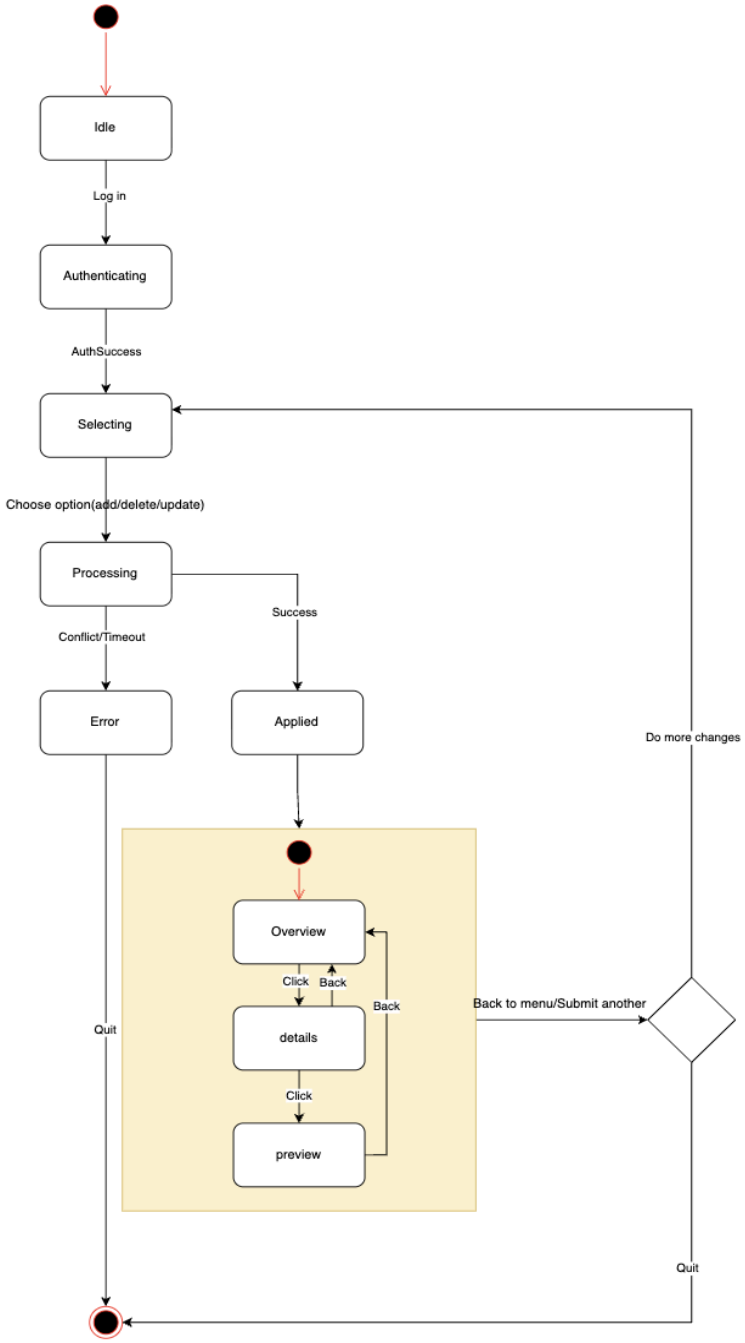
15.5 User upload the image (Evangeline)



This Diagram illustrates the main interaction flow in the *Personal Cosmetic Assistant* system. It shows how different components communicate when a user uploads a facial image to get personalized recommendations. The User initiates the process via the UI, which sends the image to the ImageProcessor and SkinAnalyzer for analysis. The analysis results are returned to the ConflictChecker to verify ingredient safety, which queries the Database for suitable products. The RecommendationEngine then generates personalized suggestions based on safe products and user preferences, and the UI presents the recommended cosmetic products back to the user.

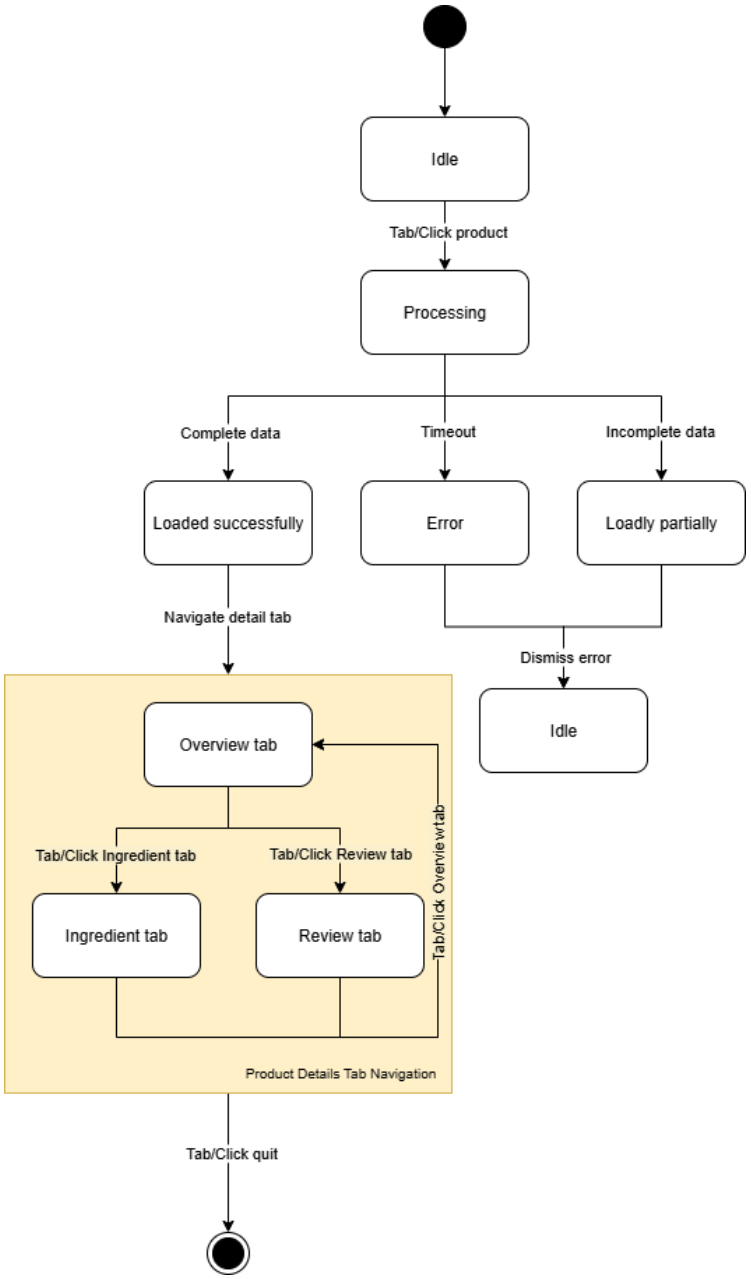
16. State Machine Diagrams

16.1 Data analysis team manages the database (Yue)



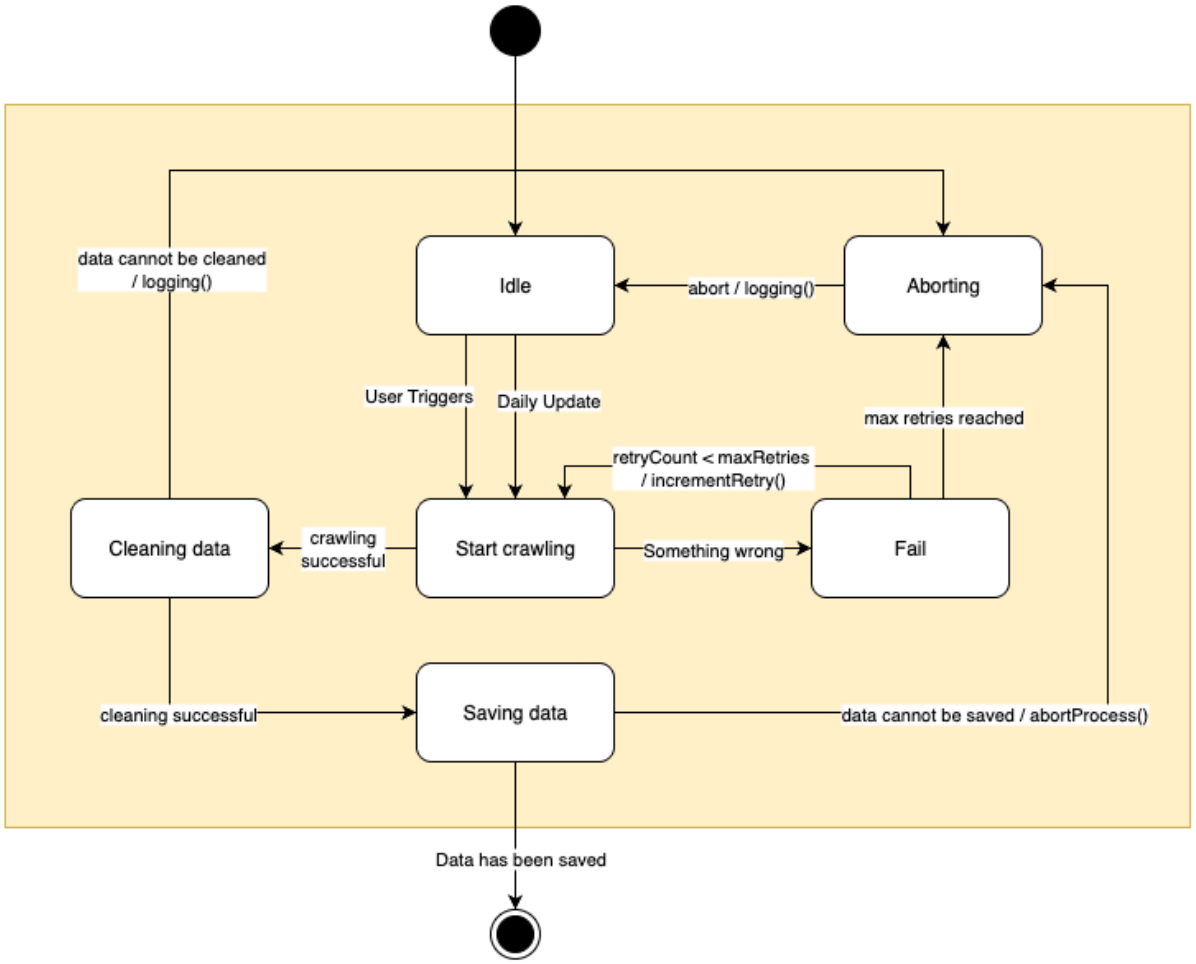
This state machine diagram describes the state transitions of a data analysts when managing a database:
From Idle to Authenticating to complete login, then to Selecting to perform operations (Add/ Delete/ Update); entering Processing to execute changes, successfully transitioning to Applied, or to Error and ending in case of conflict or timeout. After the application is successful, it enters a sub-state area, toggling between Overview/details/preview to view and preview. Finally, at a decision point, it chooses to continue submitting more changes or exit.

16.2 User access product details (Ethan)



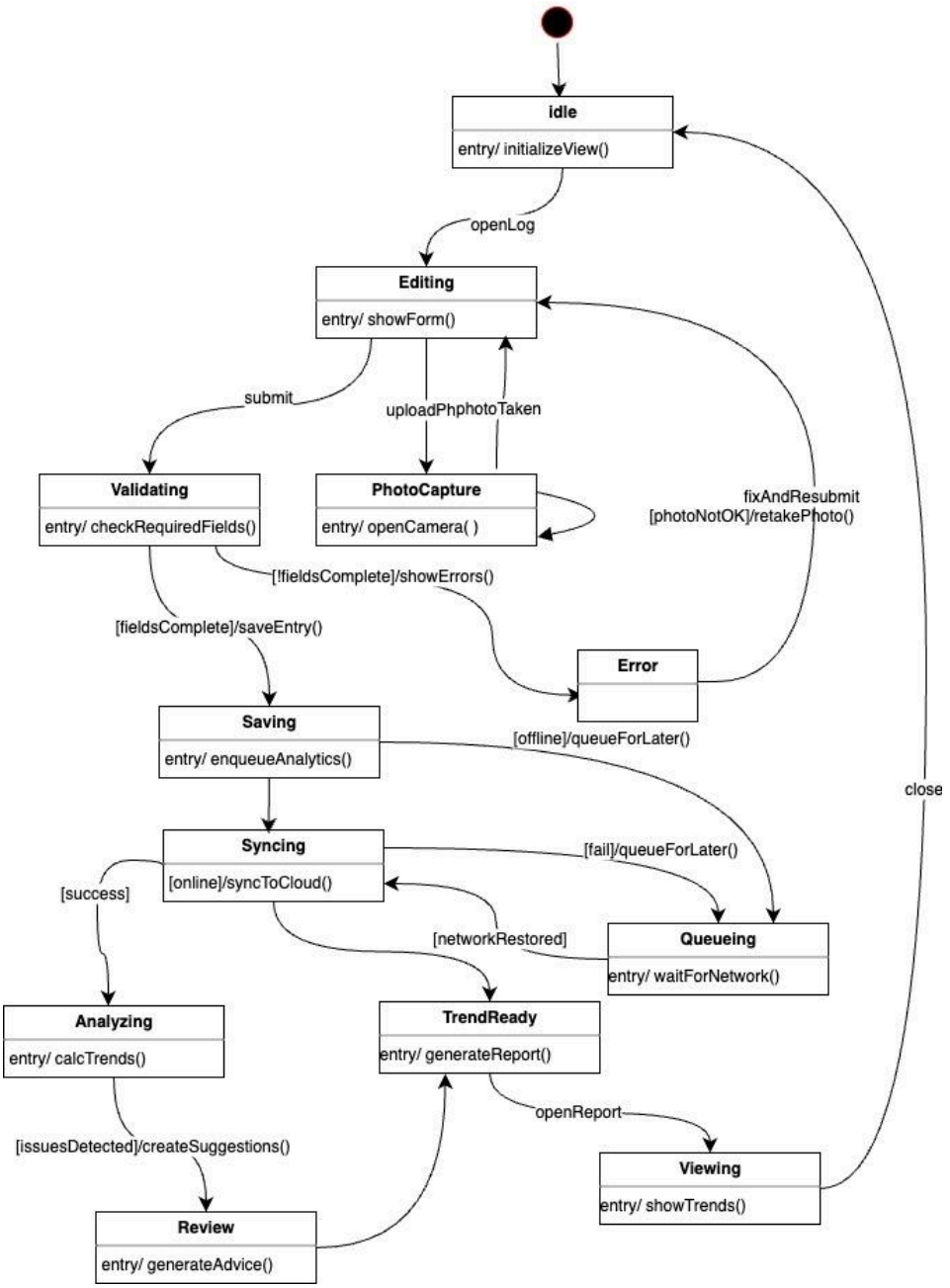
This state machine diagram shows the state changes when the user accesses a product details. Starting from an idle state, the user first chooses a product. It then transitions to 'Processing' state where there could be three outcomes: Complete data, Incomplete data and Error. If the data is not complete, the state will become idle after displaying an error message. Otherwise, the user will now be at the 'Overview tab' state where they can choose to check the ingredients or reviews of the product. Finally, they can quit to go back to the original state.

16.3 Automatically Update Database (Sean)



The state machine diagram shows the workflow of the data crawling process. The system starts in an Idle state and is triggered either by a user or a scheduled update to Start Crawling. If crawling is successful, it moves to Cleaning Data and then to Saving Data, where the cleaned information is stored. When the data is saved, the process ends successfully. If errors occur during crawling or saving, the system transitions to Fail, retries if possible, or moves to Aborting when the maximum retry limit is reached. Logging actions are performed whenever an error or abort occurs to record the issue.

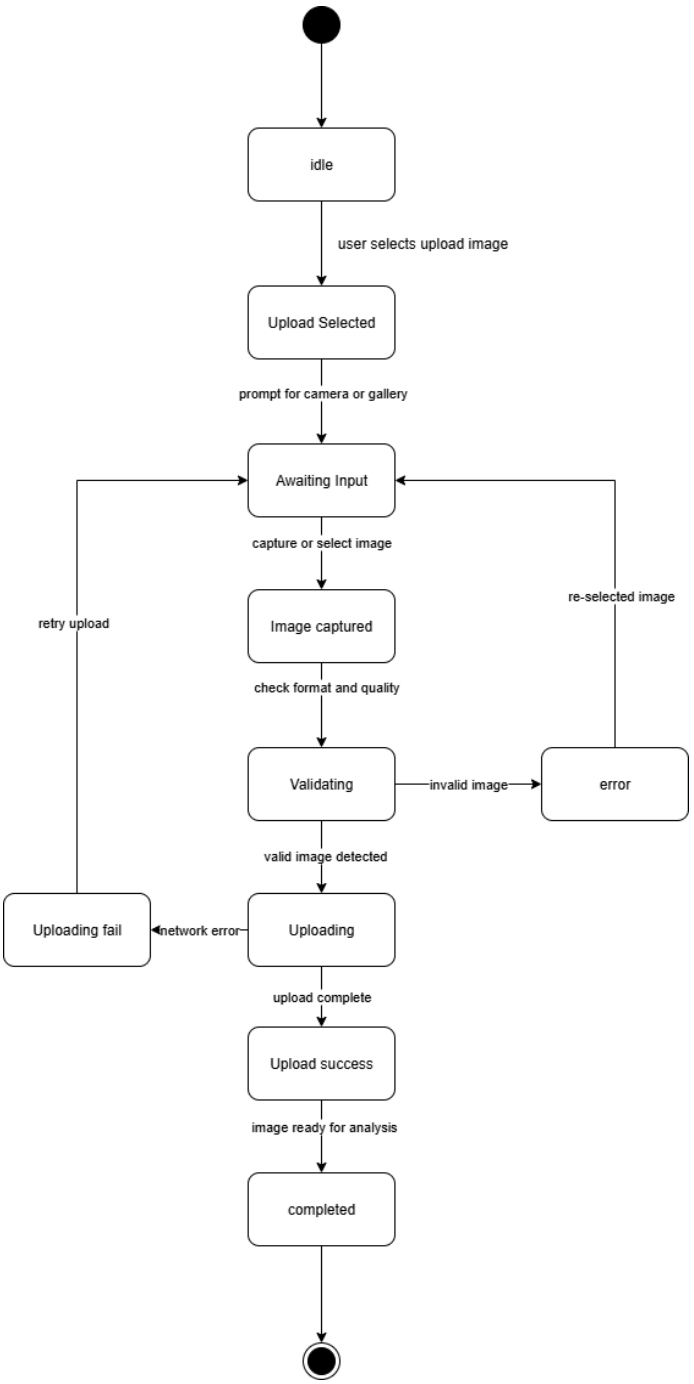
16.4 Skincare Routine Tracking & Feedback (Yiming Shen)



This state machine diagram shows the main state changes of the **skincare log** feature.

After initialization (**Idle**), the system enters **Editing** for users to fill logs and upload photos. Invalid input leads to **Error**; valid input moves to **Saving**. If online, data uploads in **Syncing**; if offline, it goes to **Queueing** until reconnected.After syncing, the system runs **Analyzing**, generates a report in **TrendReady**, and, if anomalies appear, moves to **Review** for suggestions.

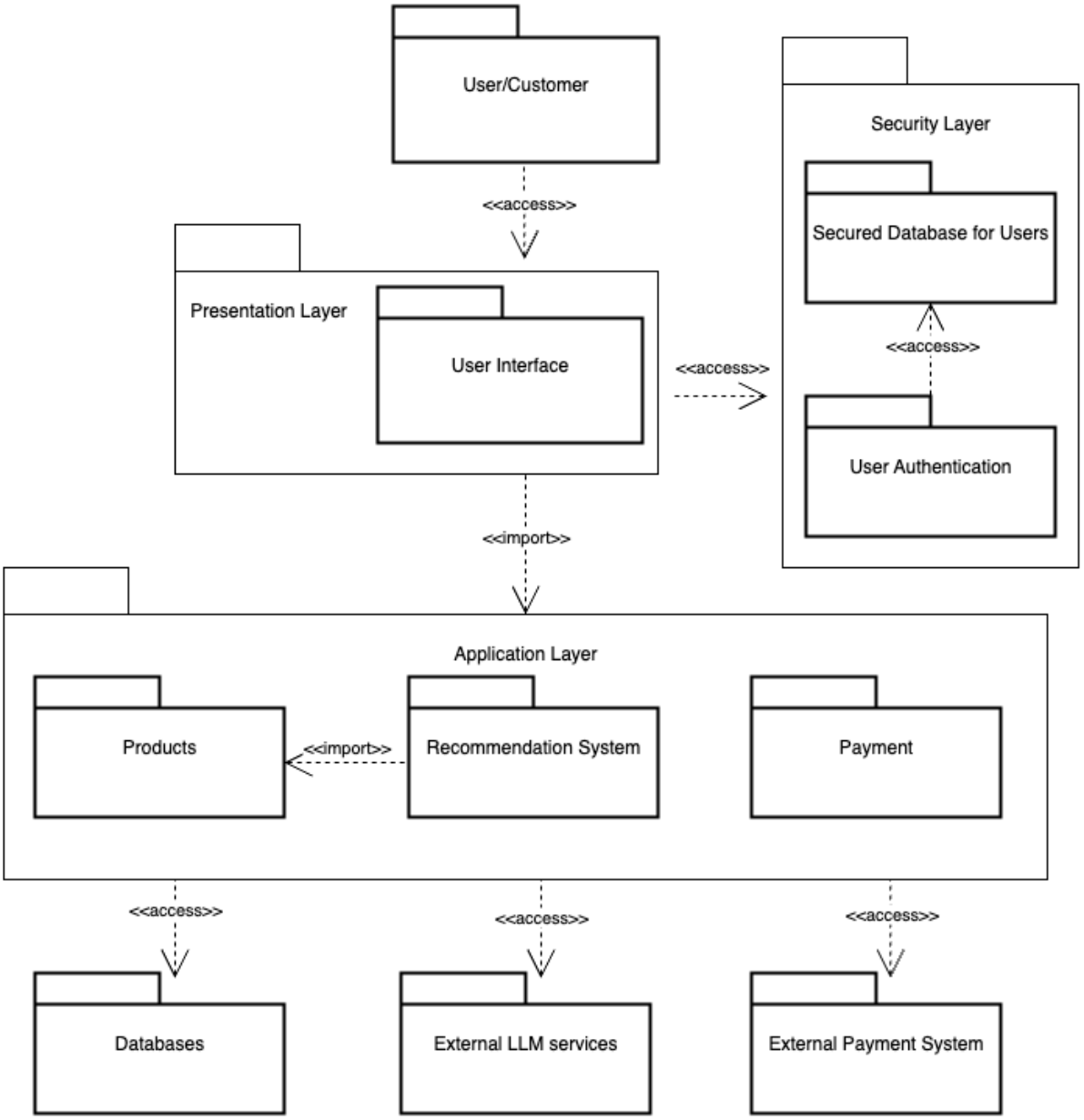
16.5 User upload the image (Evangeline)



This State Machine Diagram shows the overall process of facial image uploading in the Personal Cosmetic Assistant system. The process begins in the Idle state. When the user selects "Upload Image", the system moves to Upload Selected, then enters Awaiting Input, waiting for the user to capture or select a picture. After the image is captured, the system transitions to Validating to check its quality and format. If the image is valid, it proceeds to Uploading; otherwise, it goes to the Error state and returns to Awaiting Input. During uploading, if a network issue or system error occurs, the state changes to Uploading Fail; if successful, it transitions to Upload Success, and finally reaches the Completed state, meaning the image is ready for further analysis.

Finally, users view results in Viewing. The diagram outlines the workflow and error handling across input, sync, and analysis stages.

17. Package Diagram



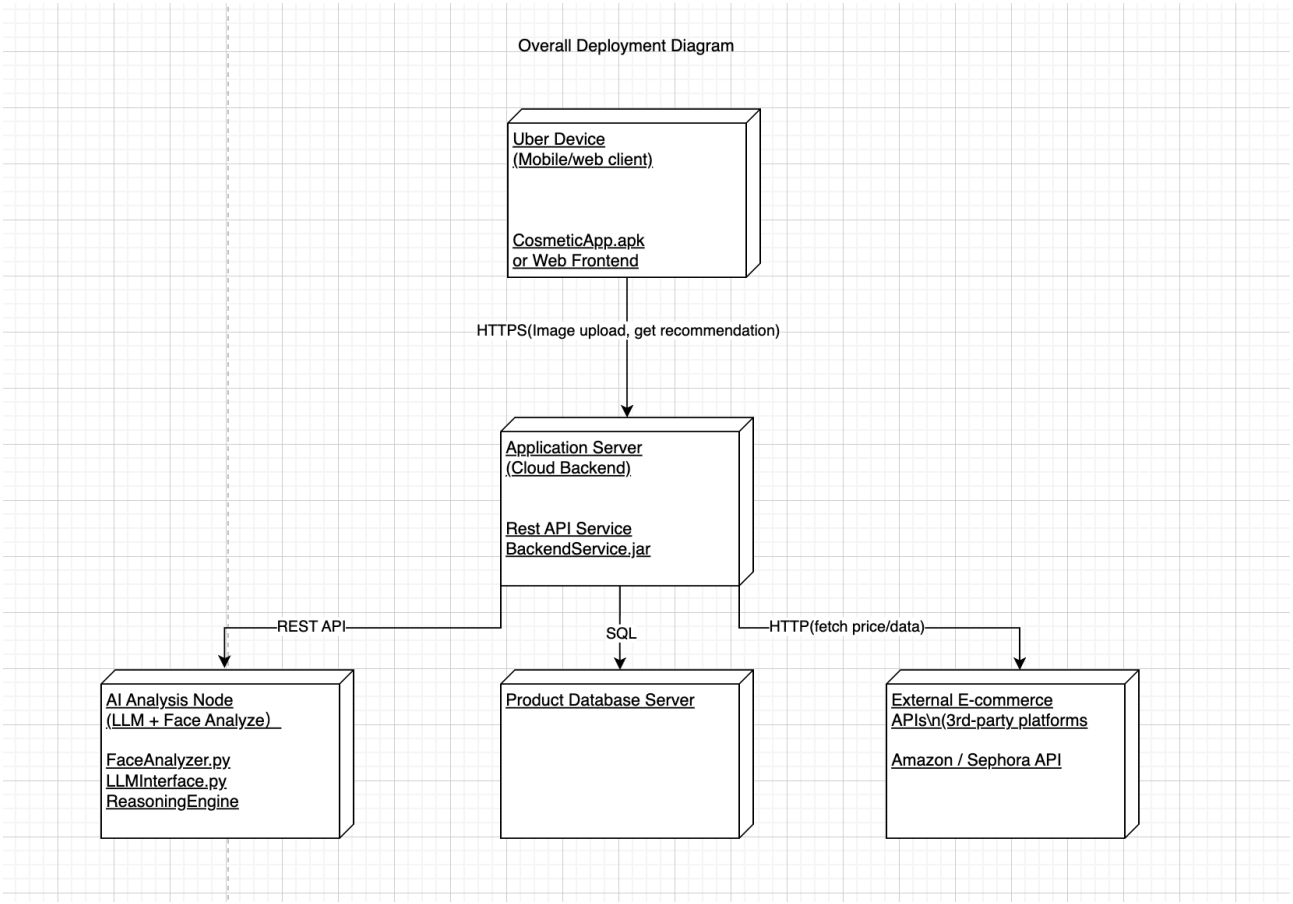
Presentation Layer: This layer contains user interface which can be accessed by a customer

Security Layer: Provides authentication system which is accessible by User Interface

Application Layer: This layer contains logics that can be imported to web applications. This layer can be imported into web applications and has access to both internal databases and external services such as large language model (LLM) APIs and external payment systems.

18. Deployment Diagram

The Deployment Diagram provides an overview of how the Personal Cosmetic Assistant system is physically deployed at runtime. It shows the main hardware and software nodes and how they communicate. The architecture is distributed: the User Device runs a mobile or web client for uploading facial images and viewing recommendations; the Application Server in the cloud manages business logic, interacting with the AI Analysis Node that performs facial and LLM-based analysis, and the Product Database Server that stores user and product data; finally, the server connects to External E-commerce APIs for real-time product and price information. All components communicate securely via HTTPS or REST APIs, ensuring scalability and reliable performance.



19. Contribution Table

Yue Kang (540574397)	- Conceptualized system functions - Defined key user group - Wrote the use cases - Designed the collaboration diagram - Created project demonstration slides
Ethan Ma (540169665)	- Set up the introduction - Identified the real-life problems - Understood the importance of Agentic AI - Designed the overall use case diagram - Formatted the report and edited the presentation
Sean Pattana (540906048)	- Initiated an ad-hoc requirement - Created a feature diagram - Designed an object diagram - Designed the overall of the class diagram - Designed the package diagram
Yiming Shen (540139598)	-Designed the structure class diagram -Created and presented the key features section in the presentation -Developed the implementation view -Assisted in integrating UML models and refining documentation -Collaborated in team coordination and discussion
Evangeline Zhang (540498196)	- Designed the deployment diagrams -Assisted in creating and formatting UML diagrams - Performed LLM integration research and use case alignment - Led requirement analysis and wrote Use Cases