FINAL PRESENTATION

TEAM 13

Smart Mirror System for **Skin health care**

INTRODUCTION TO SOFTWARE ENGINEERING TEAM 13

조재훈, VINCENT PAN, 박민서 , 설채은 , 이재혁 , 정민석 , 백송현

SYSTEM REQUIREMENT SPECIFICATION

INDEX - SRS

INTRODUCTION

SRS

INTRODUCTION

Software **R**equirements **S**pecification for Smart Mirror System for **Skin health care**

- **Scope**: Smart mirror H/W + S/W + Mobile Application
- Uses the **built-in camera** in the smart mirror to objectively check the condition of an individual's skin
- Serves as an **IoT device** that displays simple information through the built-in display
- With the mobile application, it is possible to inquire the **skin diagnosis record** and check it in various forms

OVERALL DESCRIPTION

SRS

OVERALL DESCRIPTION – System Interface

- Server
- Database Management
- Data format
- Communication

AWS Instance

mongoDB

JSON

AWS Instance

OVERALL DESCRIPTION – SW Interface

- Application will be compatible with smart phone with at least **Android OS 7.0** or **iOS 15.0**

OVERALL DESCRIPTION - User Characteristics

- **System manager**

Identify and manage the overall structure and data flow of the system

- **User**

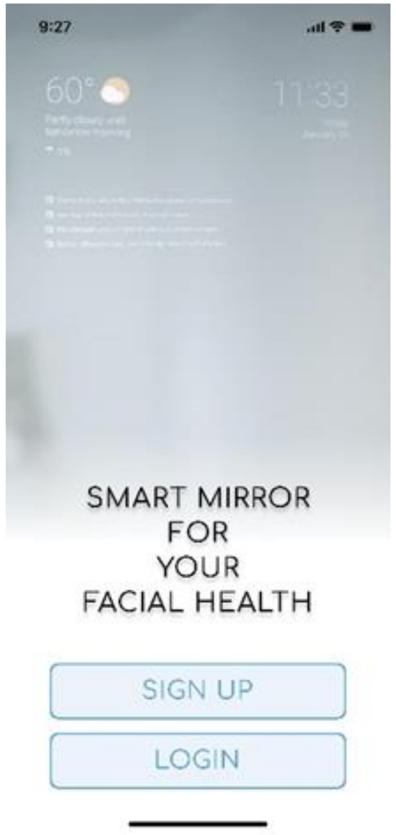
Users with accounts using Smart Mirror

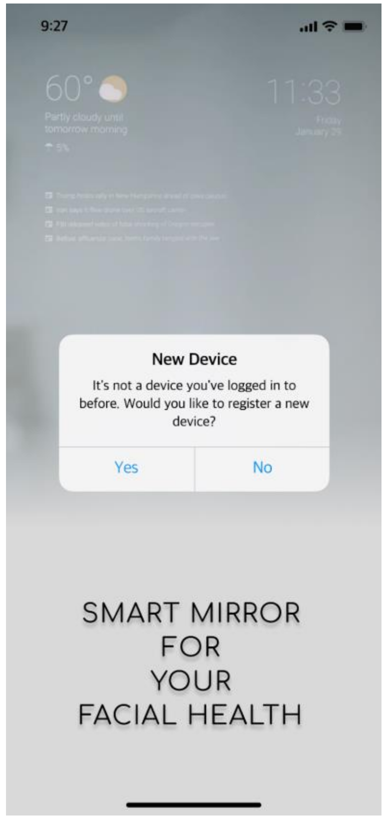
SPECIFIC REQUIREMENTS

SRS

SPECIFIC REQUIREMENTS – EXTERNAL INTERFACE REQUIREMENTS

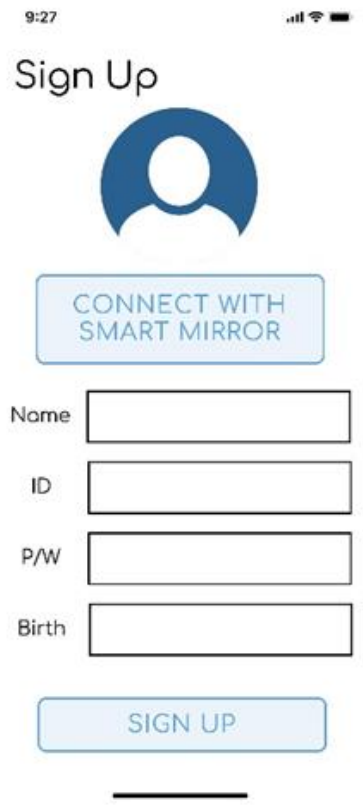
- **User Interfaces**(Application)

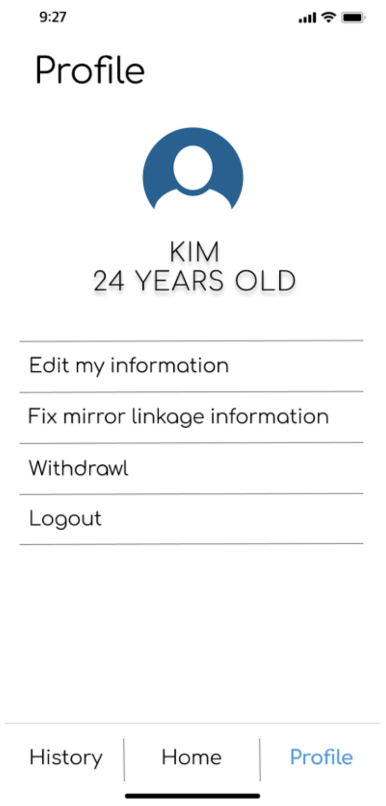


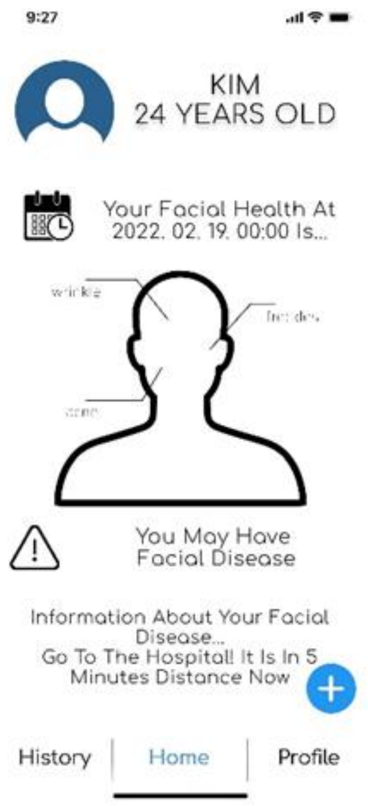


New Device

It's not a device you've logged in to before. Would you like to register a new device?





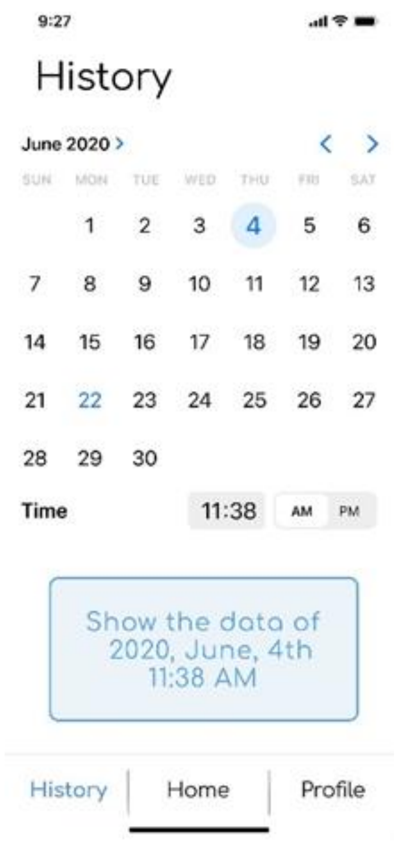


Your Facial Health At 2022. 02. 19. 00:00 Is...

Information About Your Facial Disease...
Go To The Hospital! It Is In 5 Minutes Distance Now

SPECIFIC REQUIREMENTS – EXTERNAL INTERFACE REQUIREMENTS

- **User Interfaces**(Application)



9:27

History

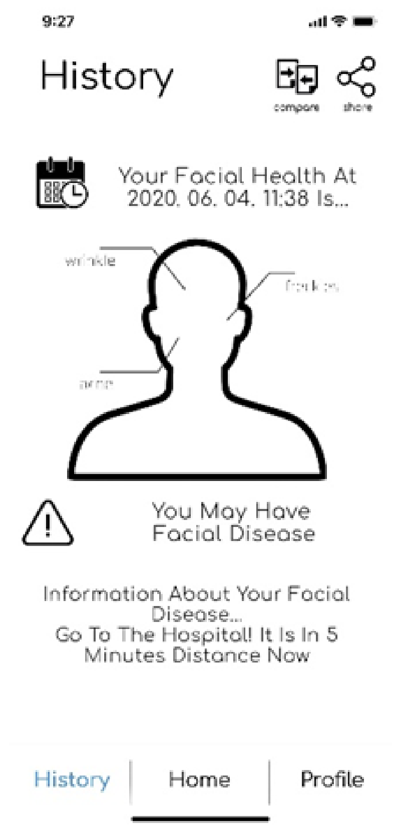
June 2020 >

SUN	MON	TUE	WED	THU	FRI	SAT
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

Time 11:38 AM PM

Show the data of 2020, June, 4th 11:38 AM

History | Home | Profile



9:27

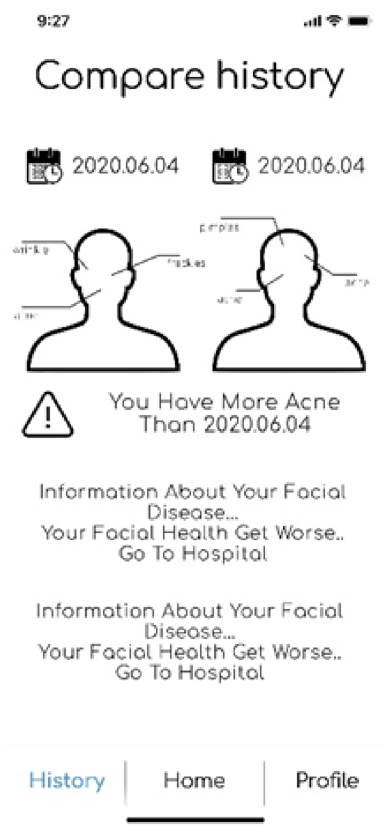
History

Your Facial Health At 2020. 06. 04. 11:38 Is...

You May Have Facial Disease

Information About Your Facial Disease...
Go To The Hospital! It Is In 5 Minutes Distance Now

History | Home | Profile



9:27

Compare history

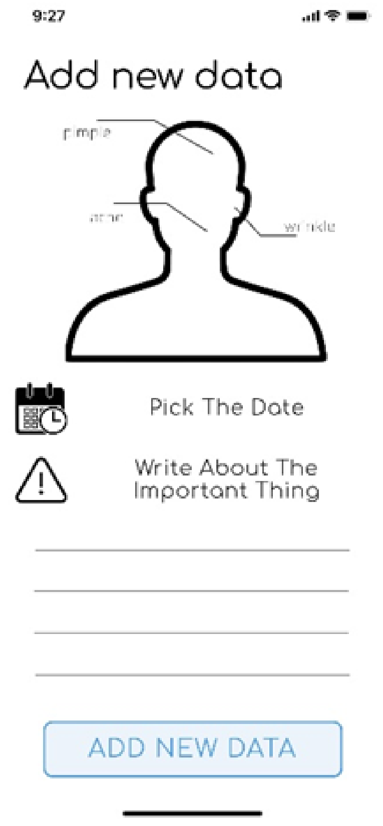
2020.06.04 2020.06.04

You Have More Acne Than 2020.06.04

Information About Your Facial Disease...
Your Facial Health Get Worse..
Go To Hospital

Information About Your Facial Disease...
Your Facial Health Get Worse..
Go To Hospital

History | Home | Profile



9:27

Add new data

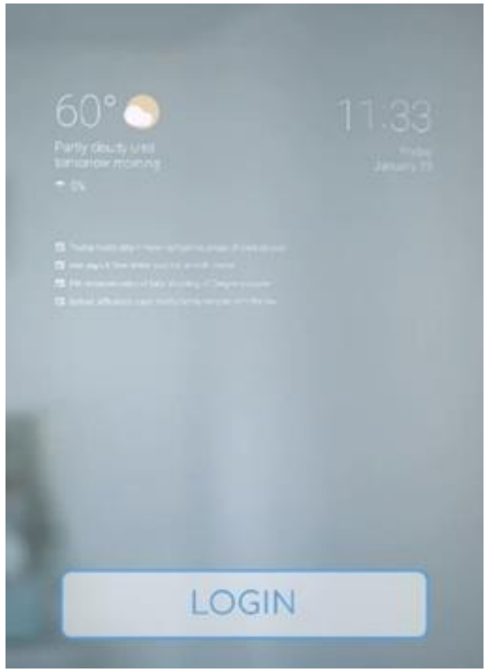
Pick The Date

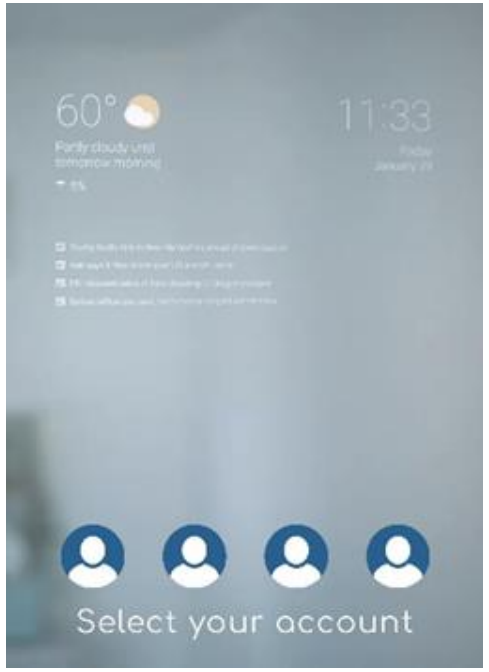
Write About The Important Thing

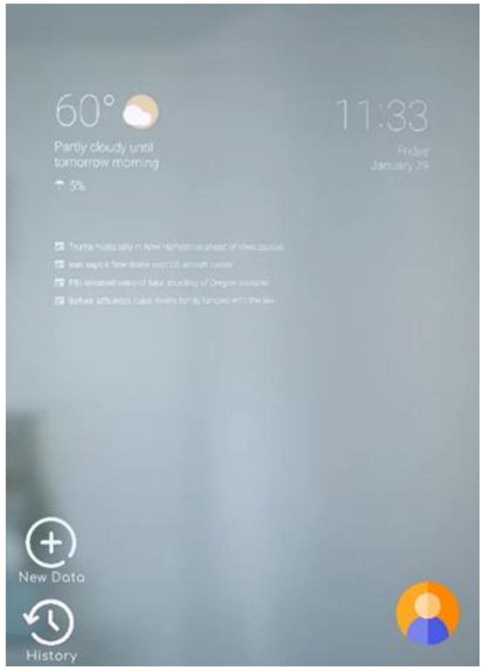
ADD NEW DATA

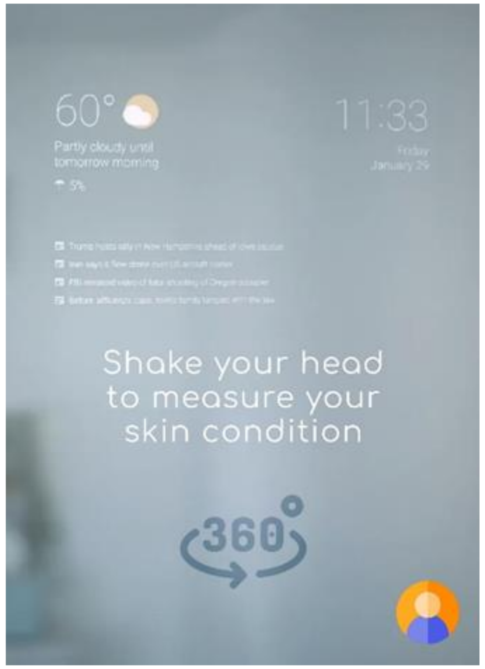
SPECIFIC REQUIREMENTS – EXTERNAL INTERFACE REQUIREMENTS

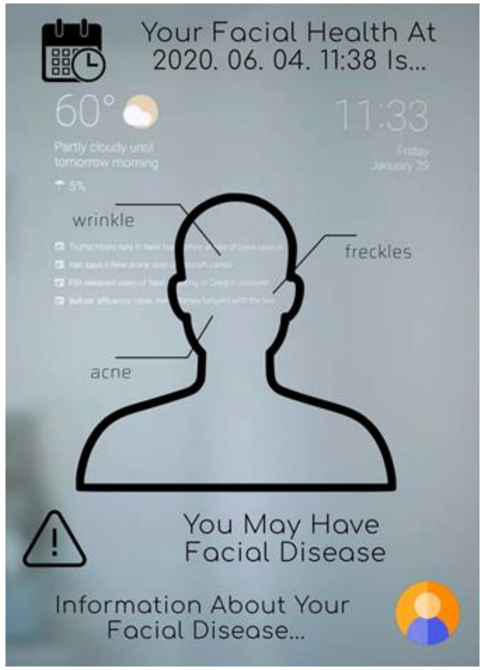
- **User Interfaces**(Mirror)



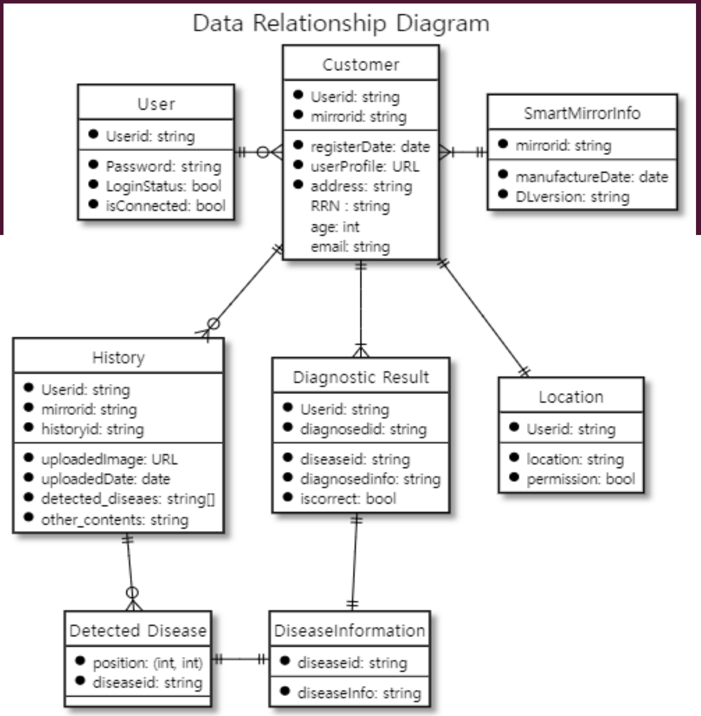





60°
11:33
Partly cloudy until tomorrow morning
Friday January 29
Shake your head to measure your skin condition
360°


Your Facial Health At 2020. 06. 04. 11:38 Is...
60°
11:33
Partly cloudy until tomorrow morning
wrinkle
freckles
acne
You May Have Facial Disease
Information About Your Facial Disease...

SPECIFIC REQUIREMENTS – FUNCTIONAL REQUIREMENTS



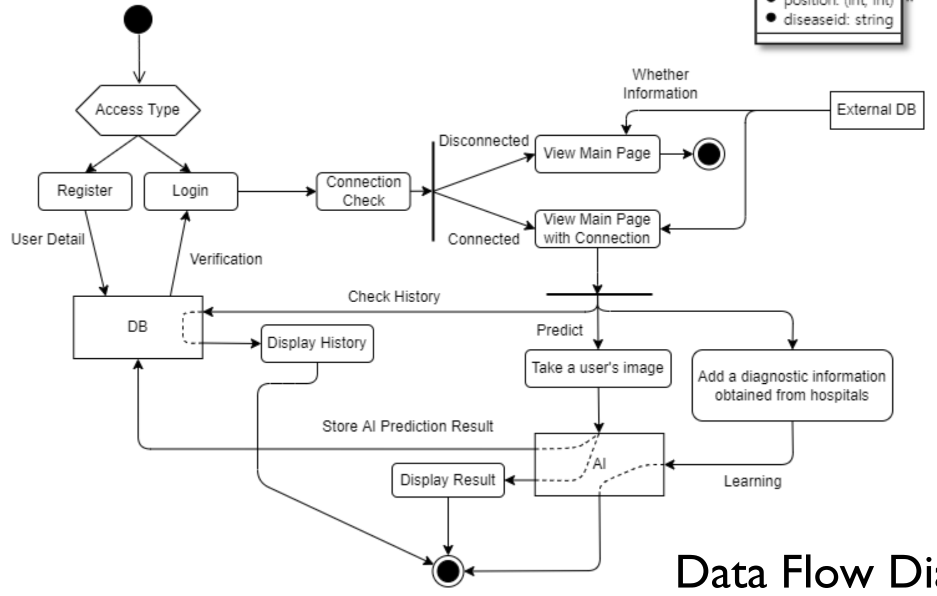
UseCase

- Based on the functions of login, sign up, record inquiry, past record storage, and diagnosis result.

Data Flow

- Database Schema
- Data Flow Diagram

Data Relationship Diagram



Data Flow Diagram

SPECIFIC REQUIREMENTS

Product requirements

- Usability Requirement
- Performance Requirement
- Security Requirement

Organizational requirements

- Environmental Requirement
- Operation Requirement
- Development Requirement

External requirements

- Regulatory Requirement
- Ethical Requirement
- Safety/Security Requirement

Logical Database requirements

mongoDB

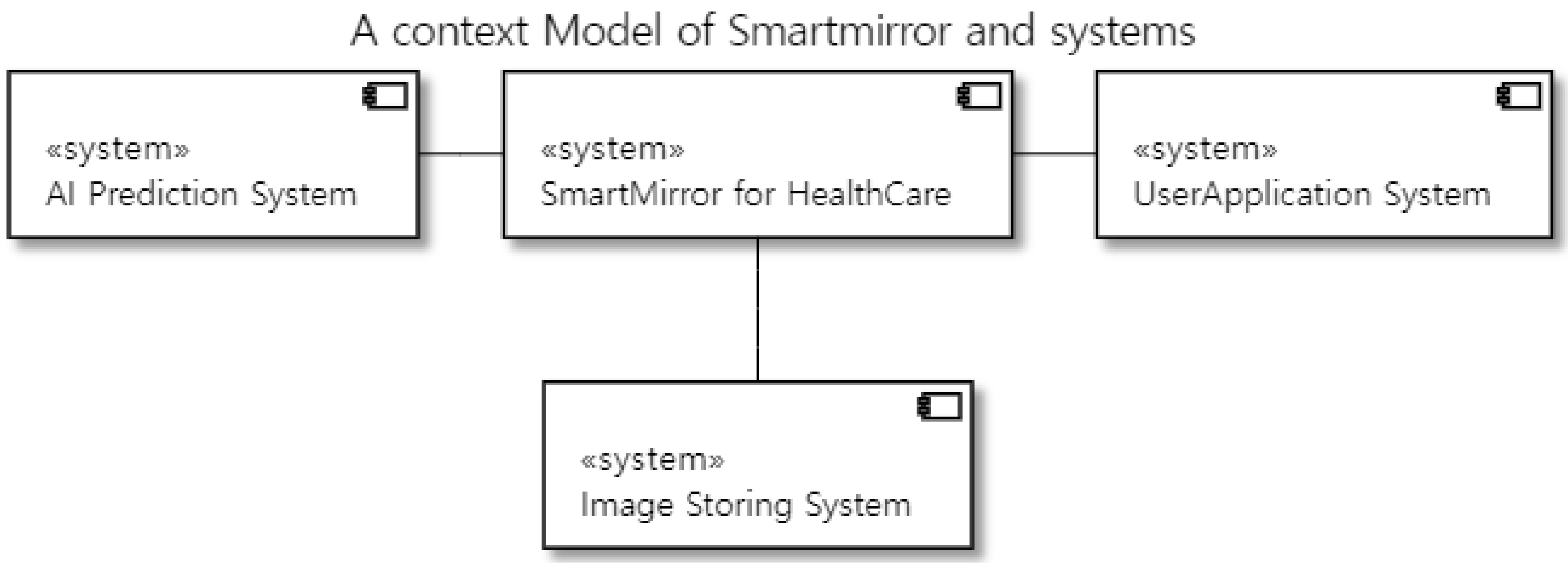
- Store and manage necessary data using **mongoDB** service

SYSTEM MODEL

SRS

SYSTEM MODEL – INTERACTION MODEL

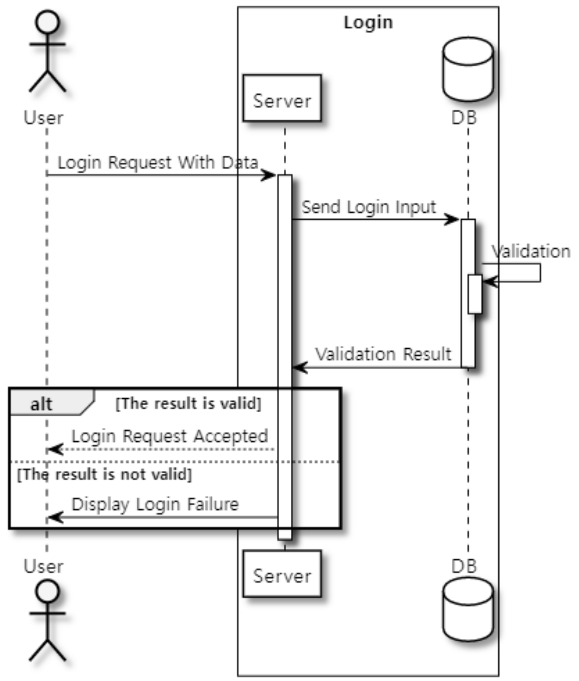
- **Context model diagram**



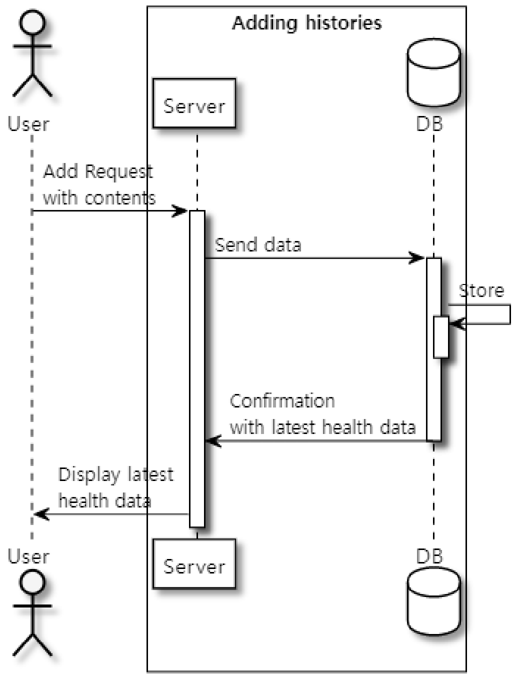
SYSTEM MODEL – INTERACTION MODEL

- **Sequence model diagram**

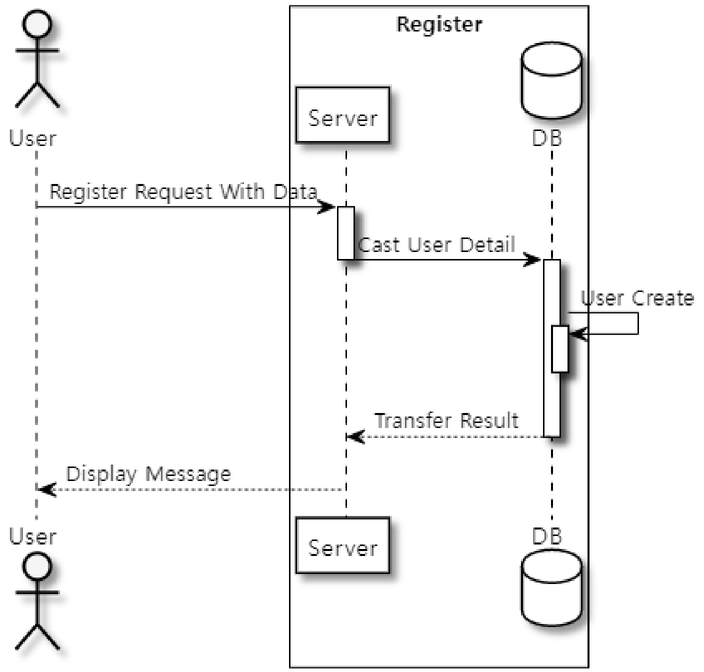
Login Sequence



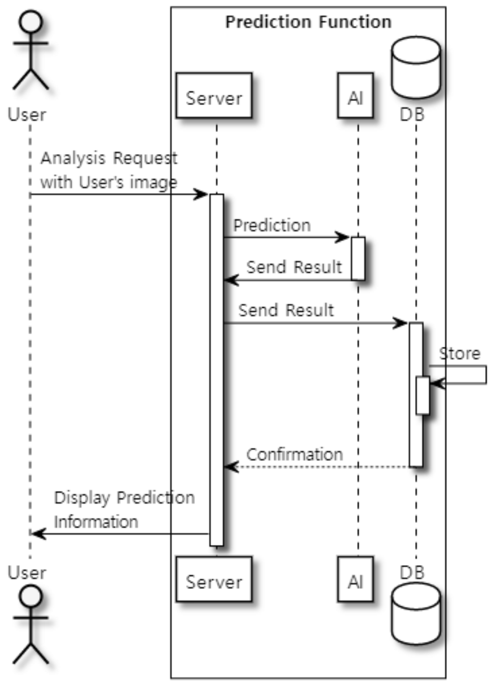
Adding a new history Sequence


Adding histories
User
Server
DB
Add Request with contents
Send data
Store
Confirmation with latest health data
Display latest health data

Register Sequence

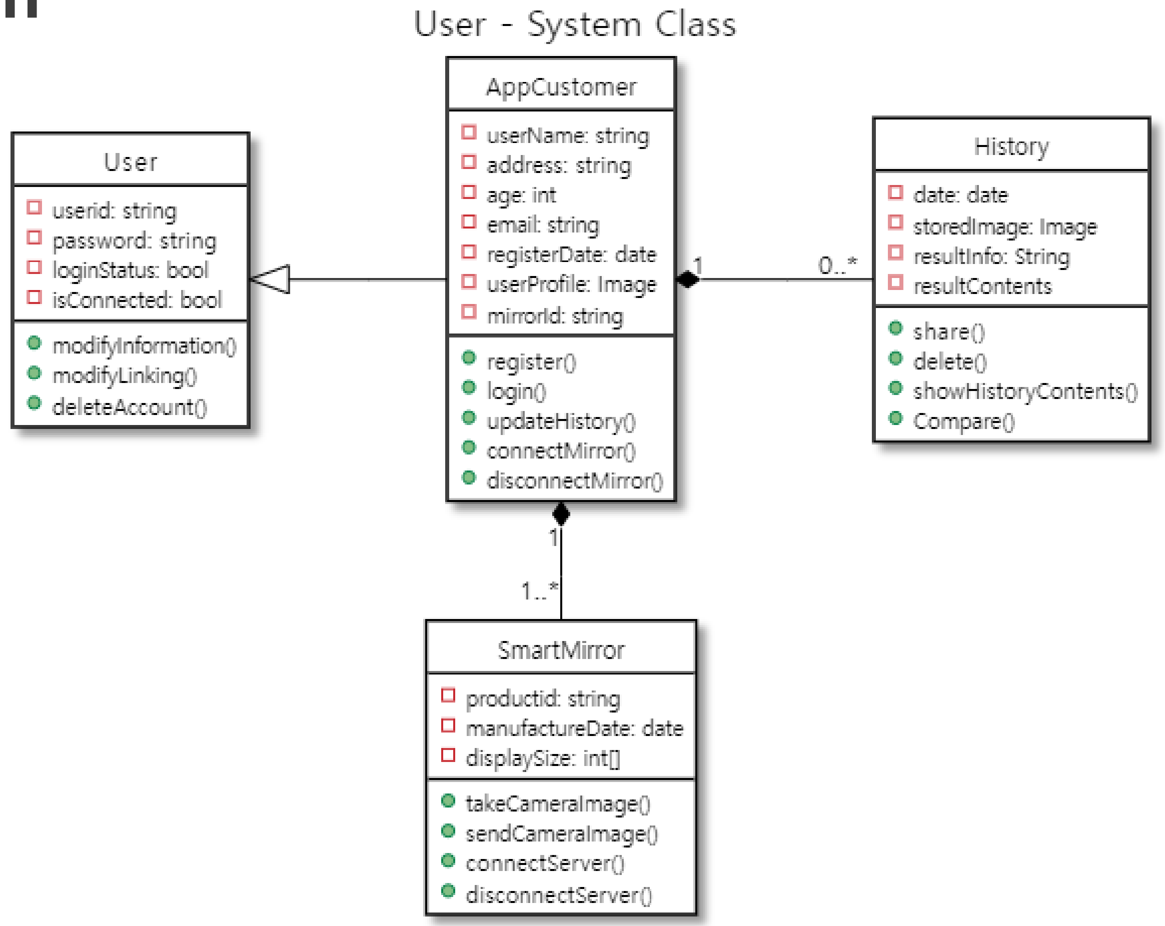


Using AI Model Sequence

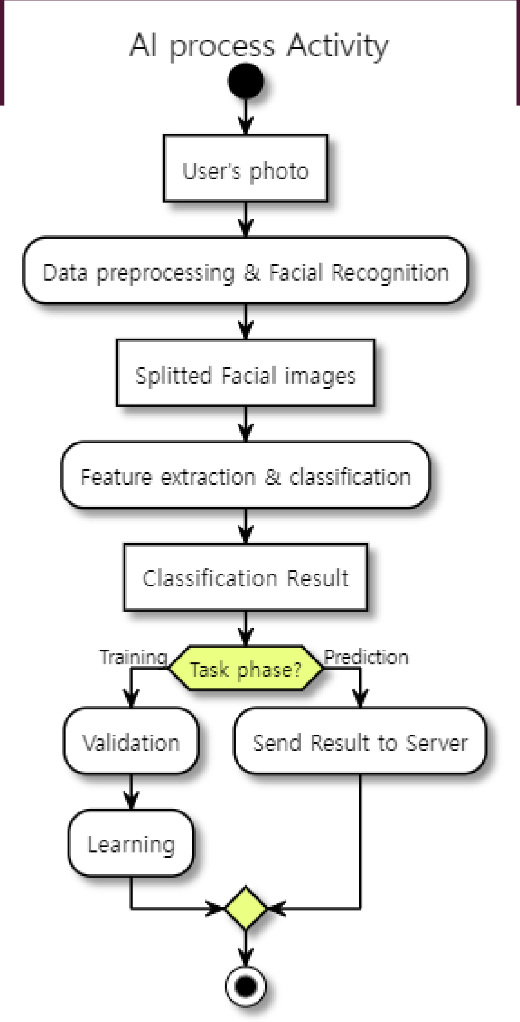


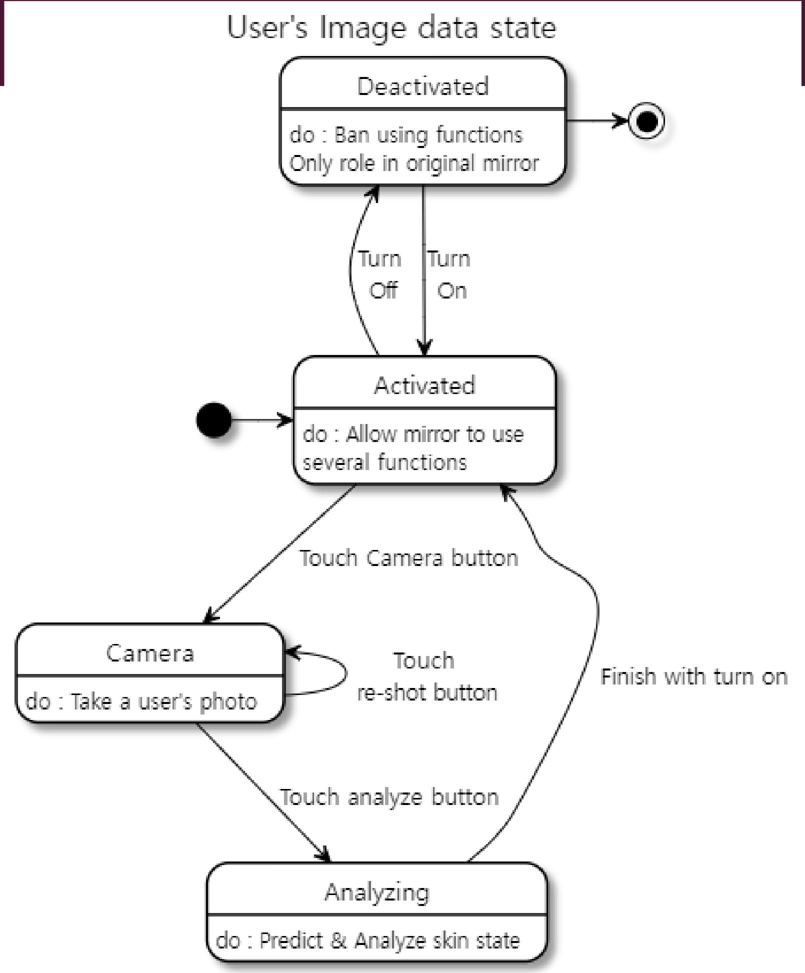
SYSTEM MODEL – STRUCTURAL MODEL

- **Class Diagram**



SYSTEM MODEL – BEHAVIORAL MODEL





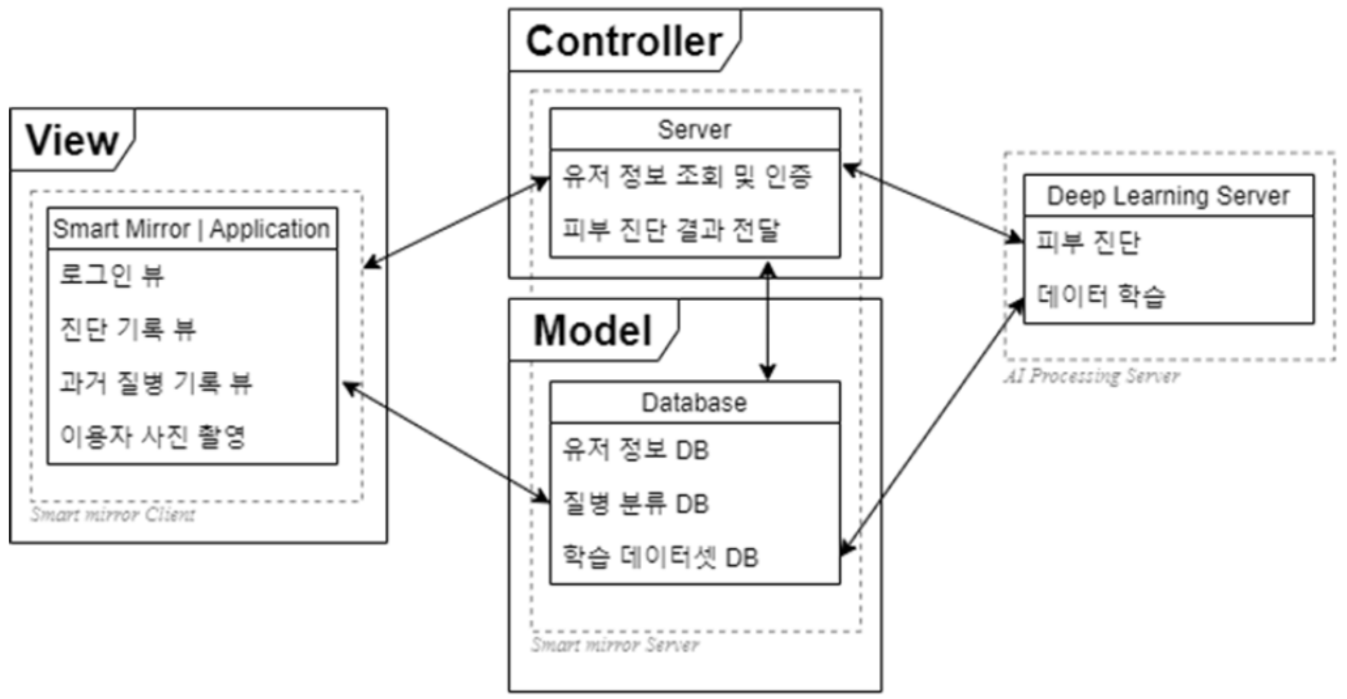
- **Activity Diagram**
- **State Diagram**

ARCHITECTURE

SRS

ARCHITECTURE

- **MVC Pattern**



SYSTEM EVOLUTION

SRS

SYSTEM EVOLUTION

- **Gesture recognition** instead of touching display
- Hardware evolution: **Deep learning server hardware** improvements
- Hardware evolution: **Camera performance** improvements
- **Emotion Analysis** & Mental illness through facial expression recognition

SYSTEM DESIGN SPECIFICATION

INDEX - SDS

INTRODUCTION

SDS

INTRODUCTION

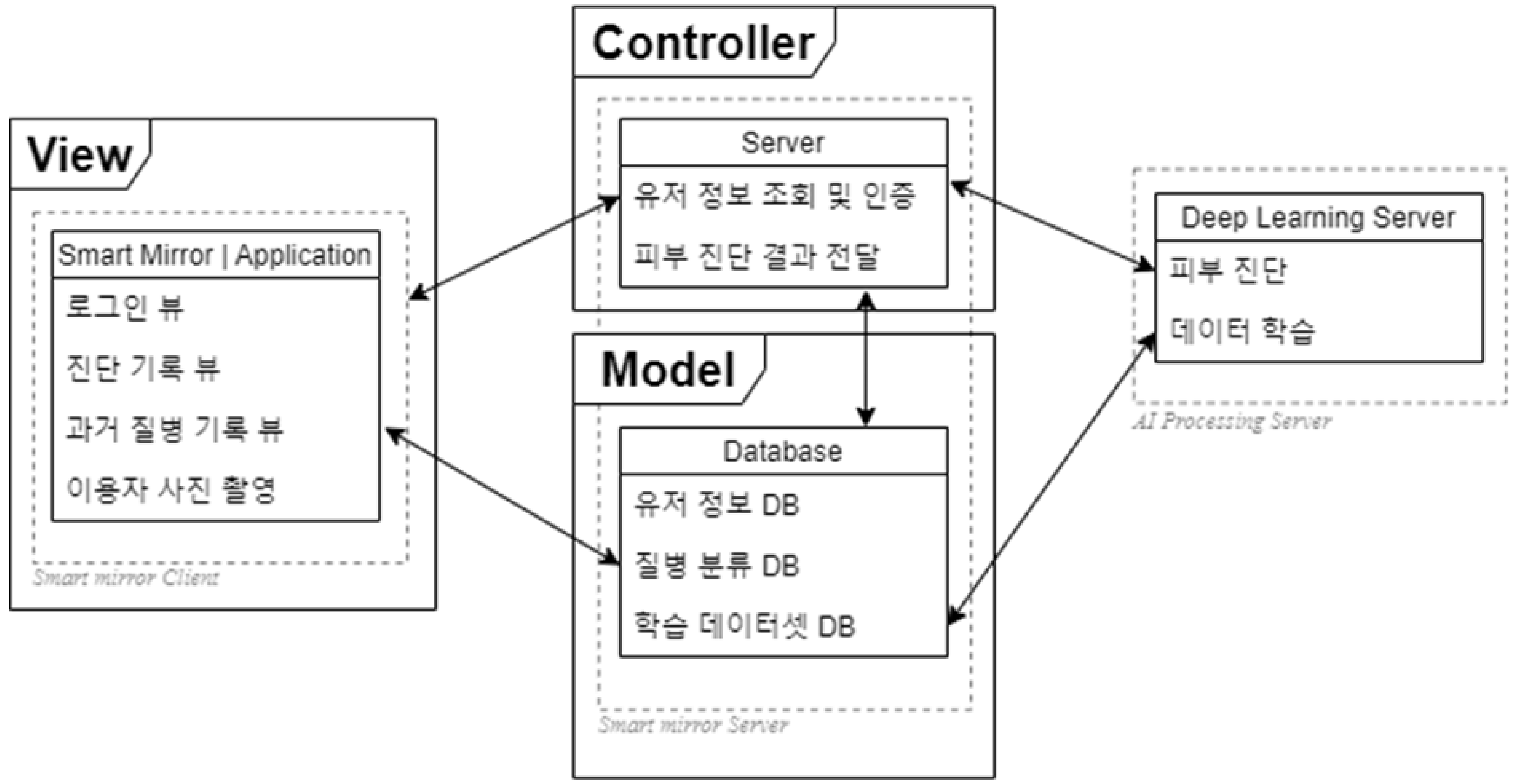
- **Overall design and function** of the software described in SRS to implement the skin diagnosis function of the smart mirror.
- **Diagram**: Class, Sequence, Activity, State, Context
- **Applied Tools**: Word, Draw.io

OVERALL ARCHITECTURE

SDS

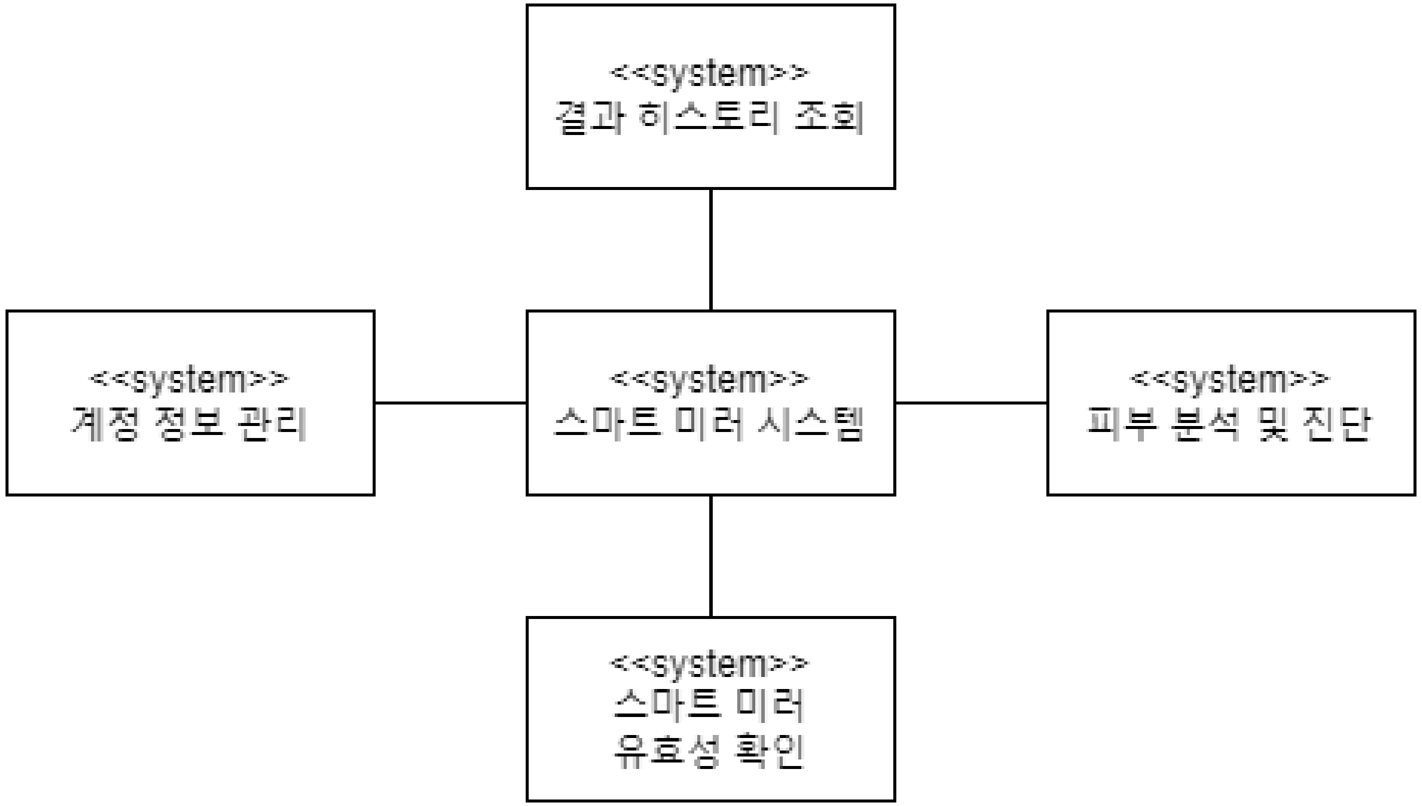
OVERALL ARCHITECTURE

- System Organization – **Overall system Architecture**



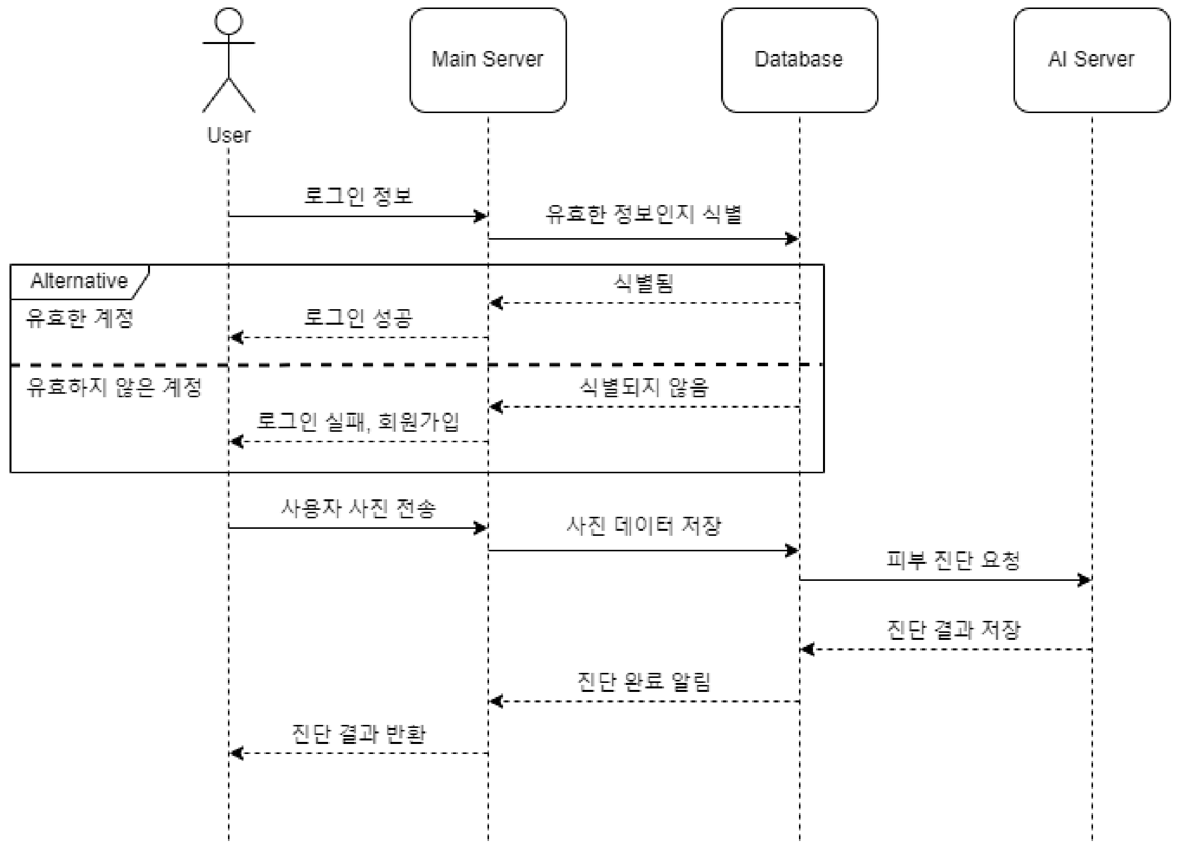
OVERALL ARCHITECTURE

- System Organization – **Context Diagram**



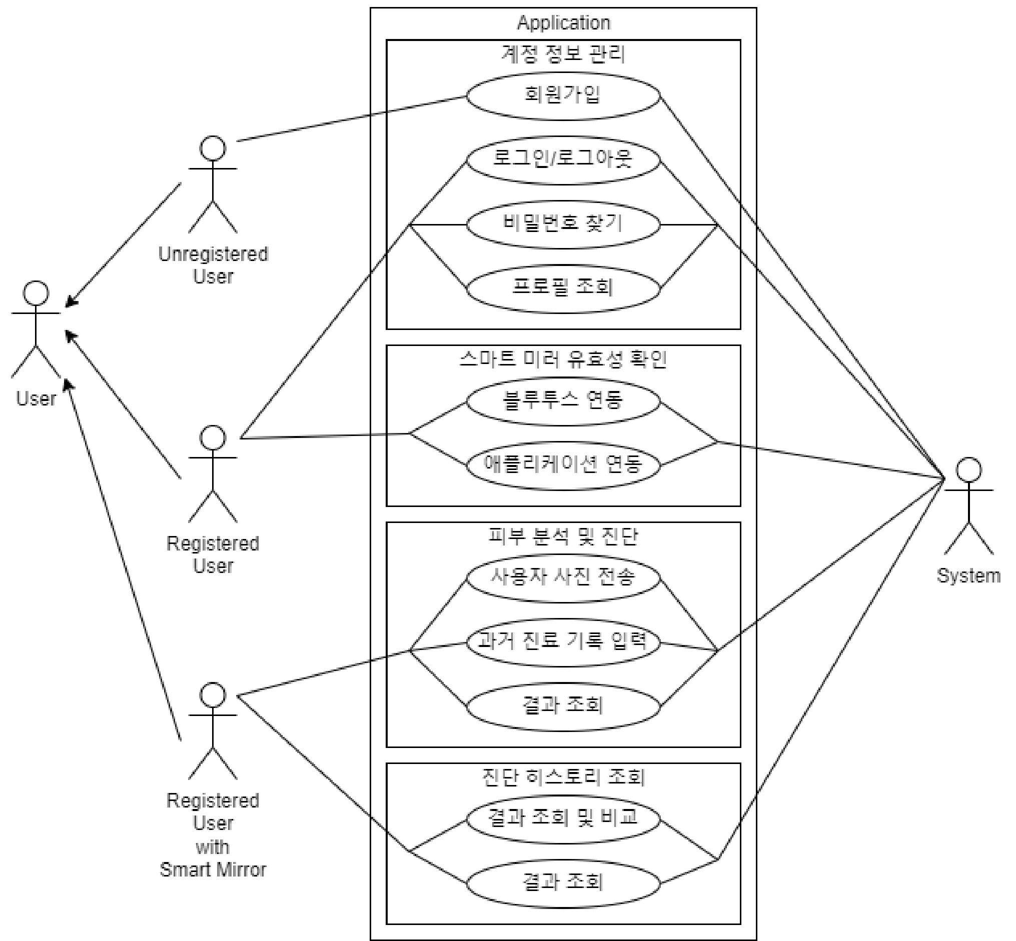
OVERALL ARCHITECTURE

- System Organization – **Sequence Diagram**



OVERALL ARCHITECTURE

- System Organization – **Use Case Diagram**

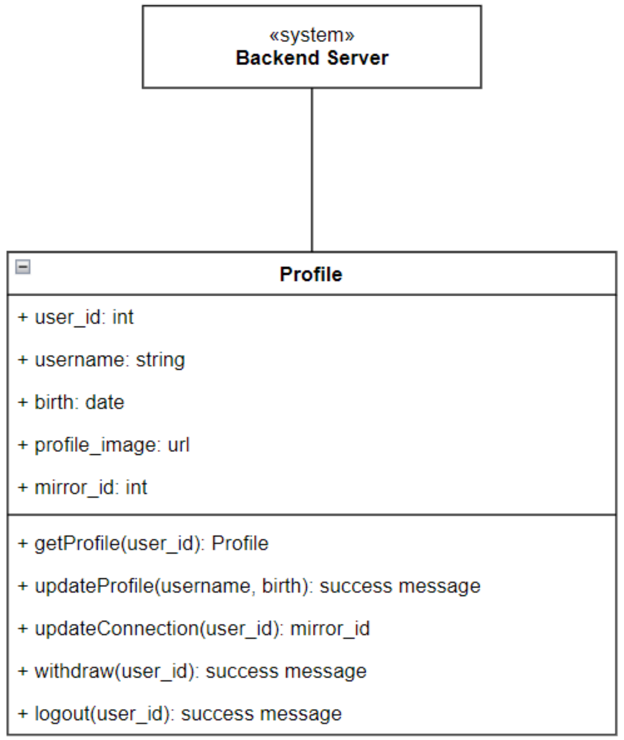


SYSTEM ARCHITECTURE - FRONTEND

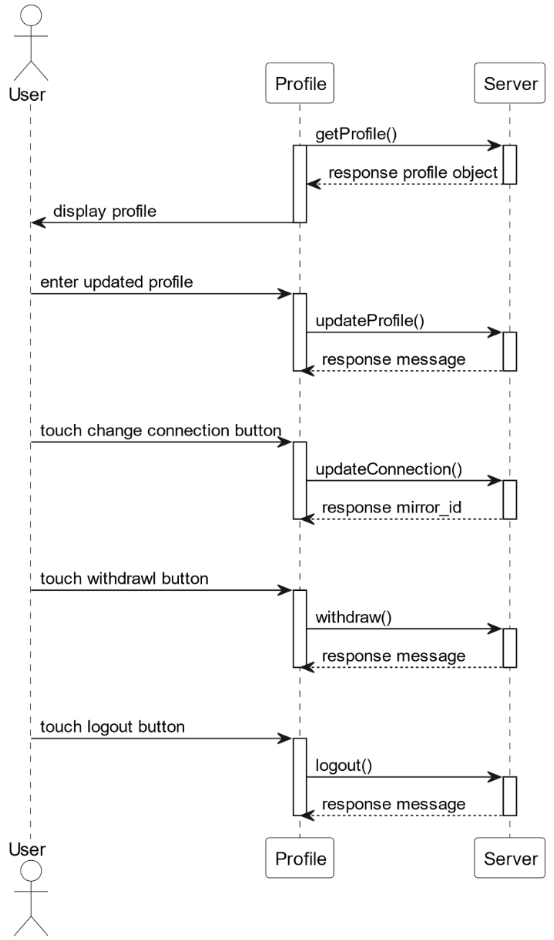
SDS

SYSTEM ARCHITECTURE – FRONTEND(MOBILE)

- System Component – **Profile**


«system»
Backend Server
Profile
+ user_id: int
+ username: string
+ birth: date
+ profile_image: url
+ mirror_id: int
+ getProfile(user_id): Profile
+ updateProfile(username, birth): success message
+ updateConnection(user_id): mirror_id
+ withdraw(user_id): success message
+ logout(user_id): success message

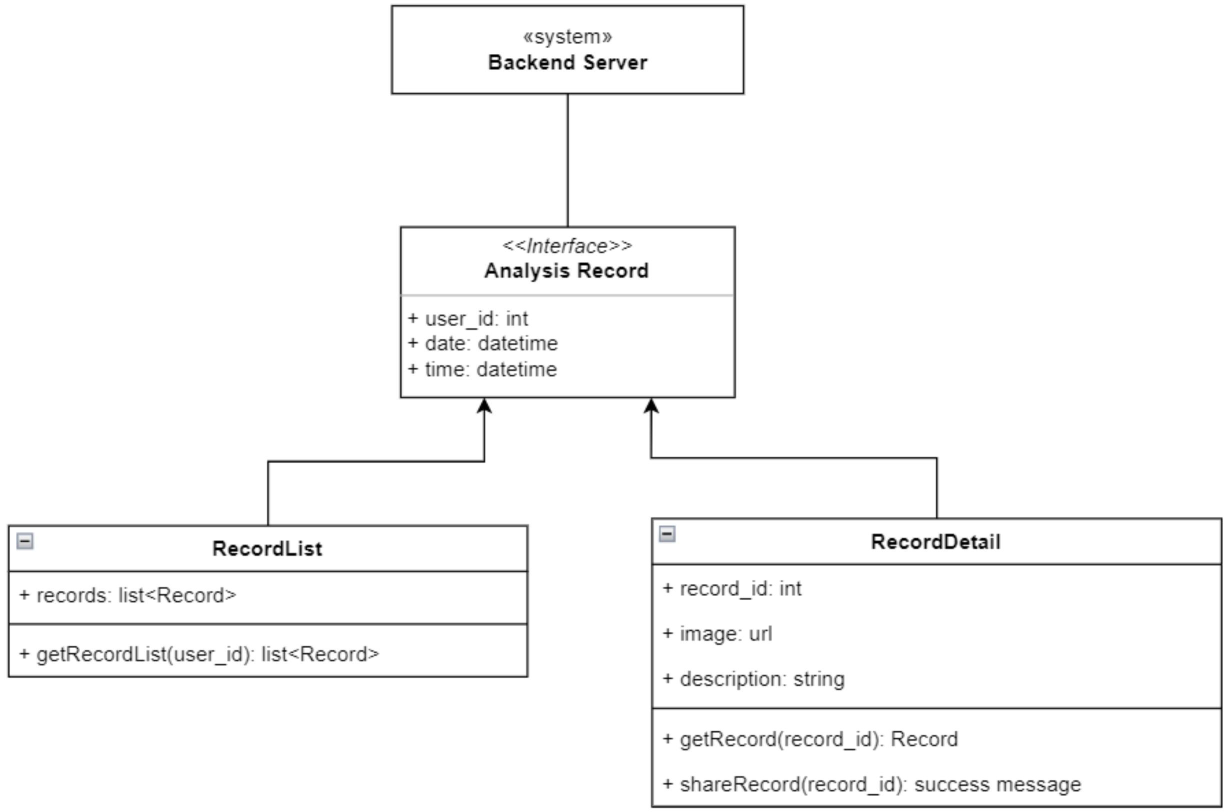
Class Diagram



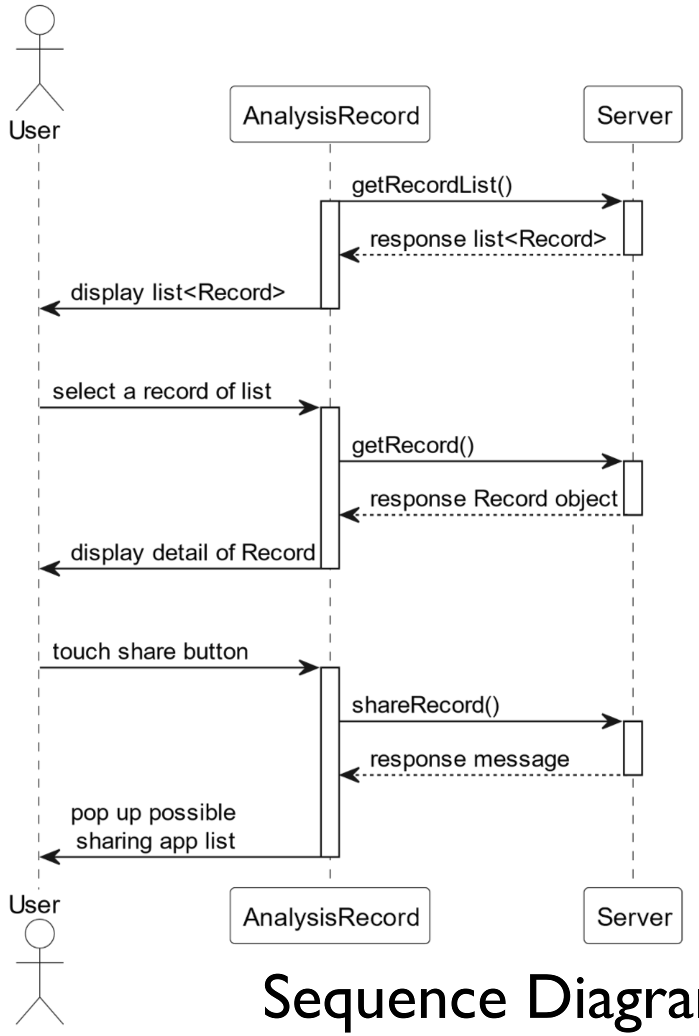
Sequence Diagram

SYSTEM ARCHITECTURE – FRONTEND(MOBILE)

- System Component – **AnalysisRecord**



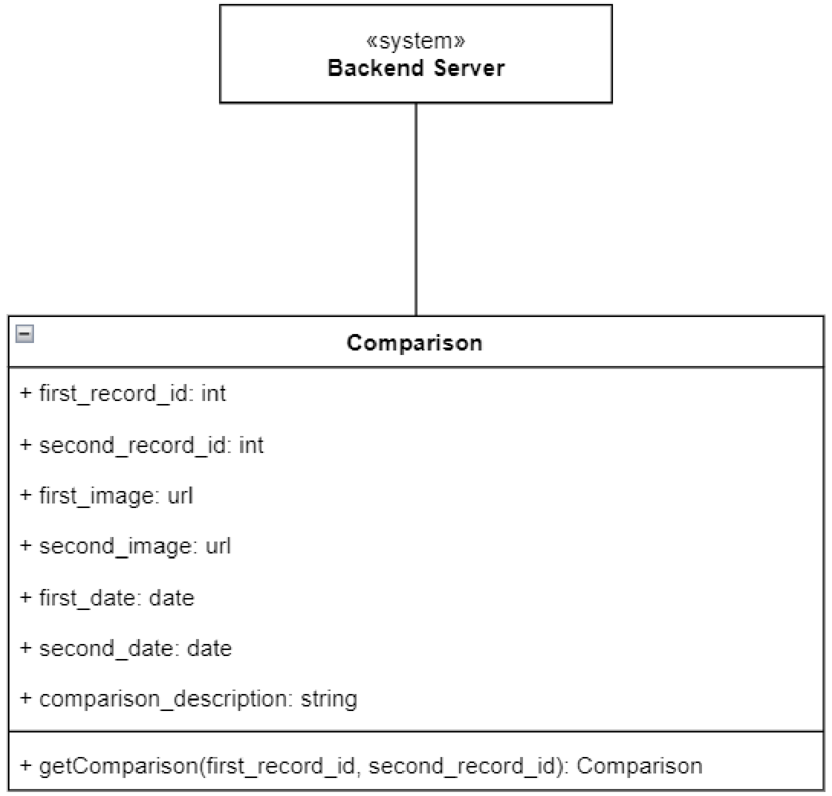
Class Diagram



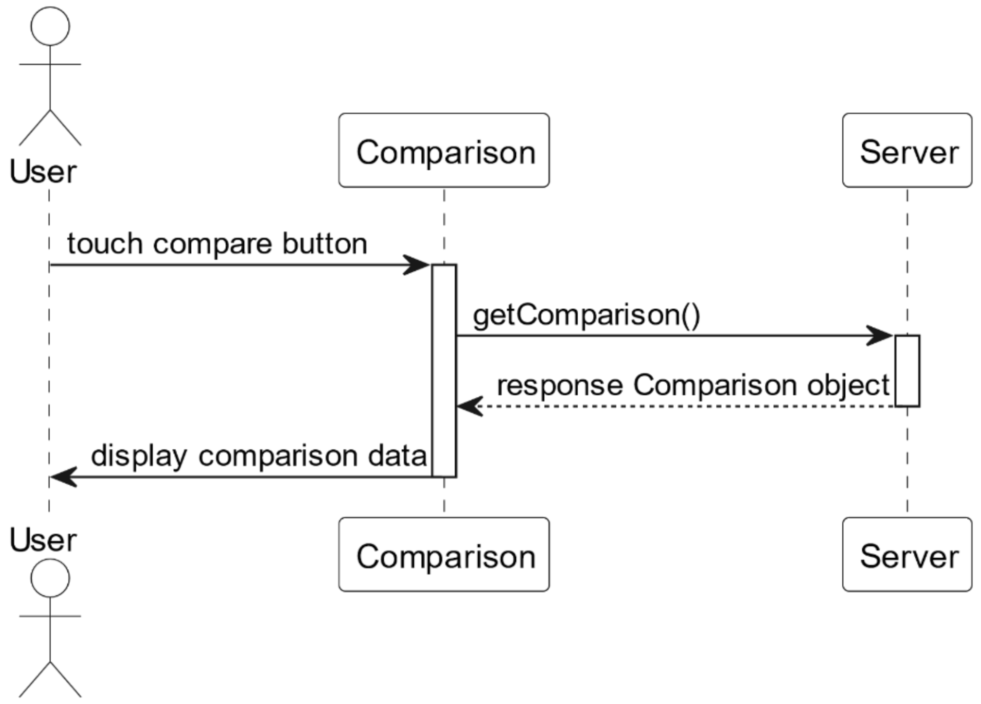
Sequence Diagram

SYSTEM ARCHITECTURE – FRONTEND(MOBILE)

- System Component – **Comparison**



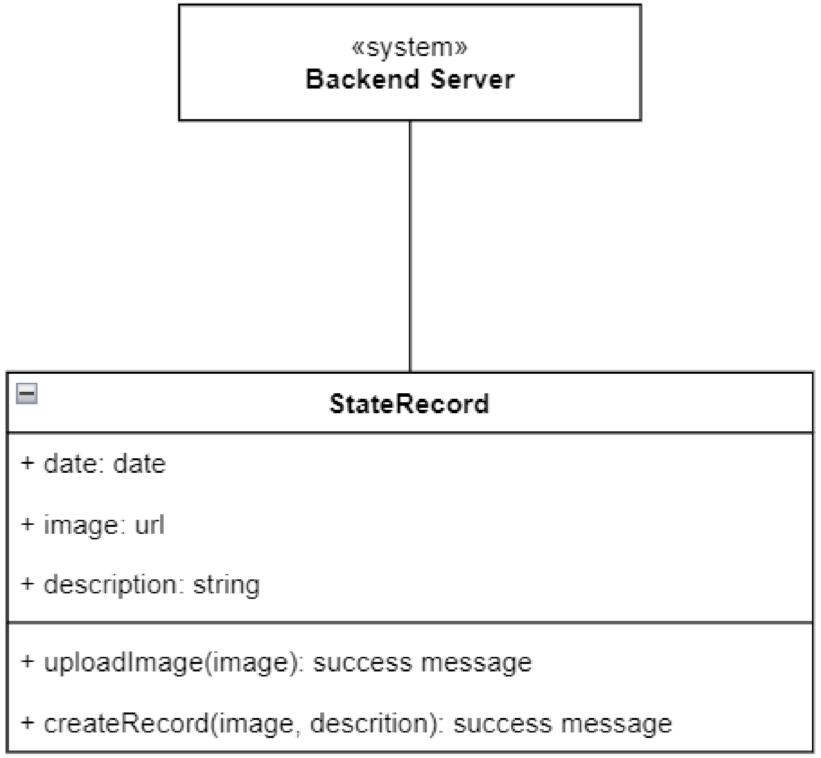
Class Diagram



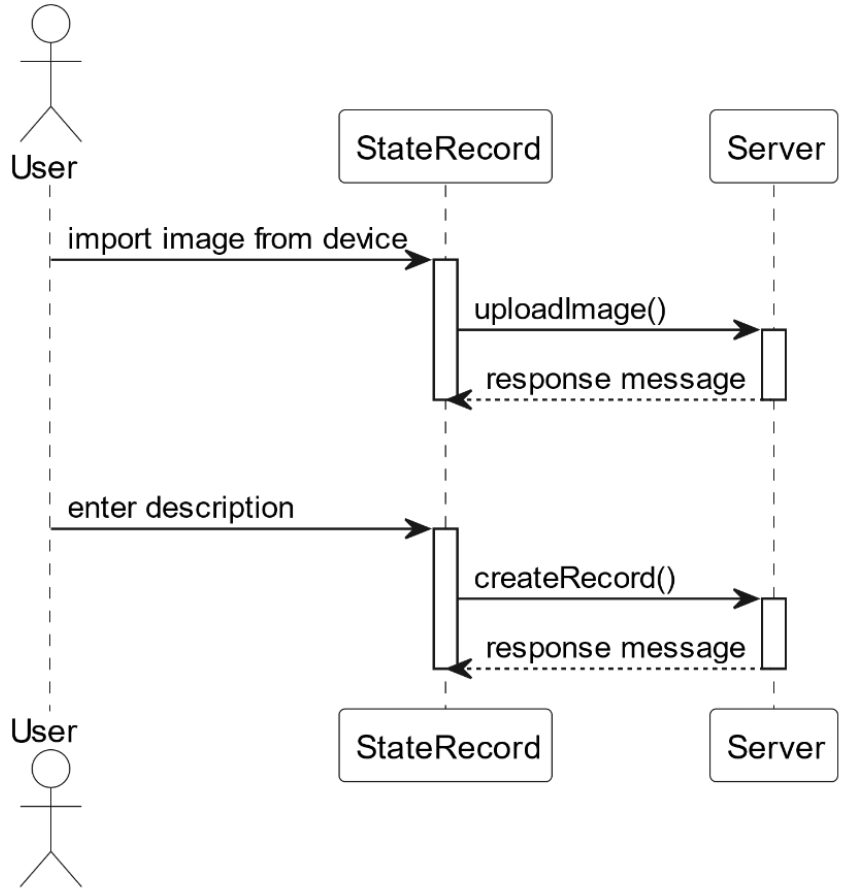
Sequence Diagram

SYSTEM ARCHITECTURE – FRONTEND(MOBILE)

- System Component – **StateRecord**


«system»
Backend Server
StateRecord
+ date: date
+ image: url
+ description: string
+ uploadImage(image): success message
+ createRecord(image, descrition): success message

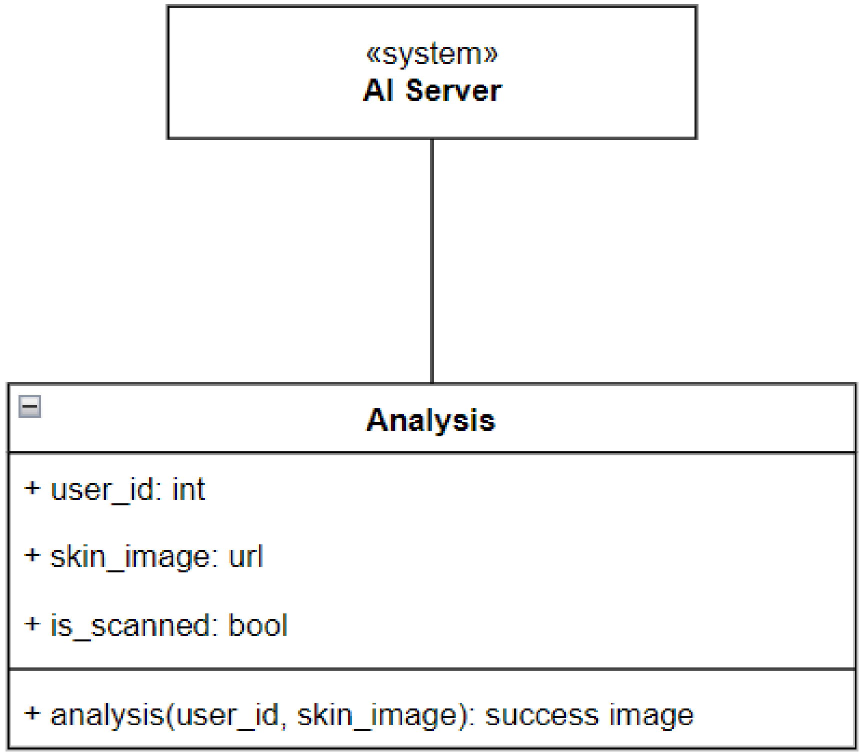
Class Diagram



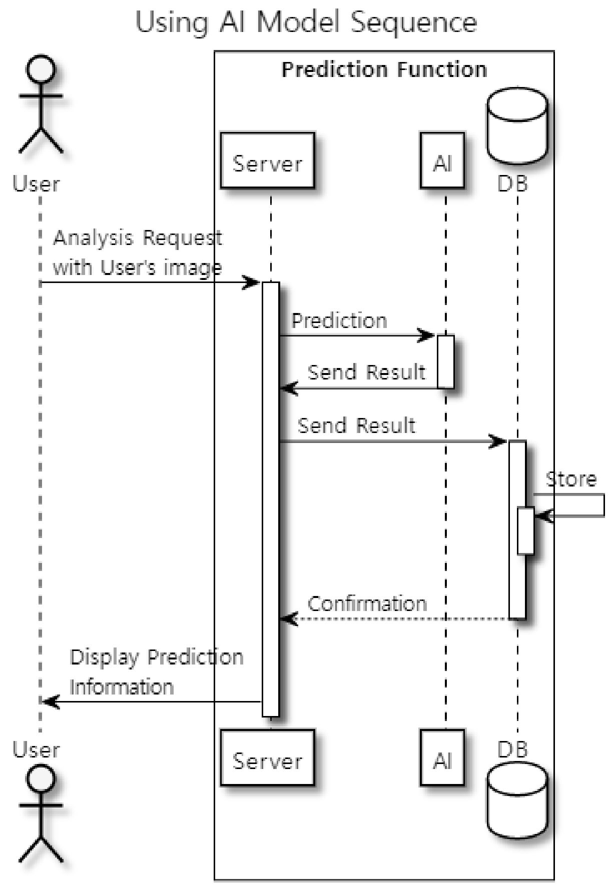
Sequence Diagram

SYSTEM ARCHITECTURE – FRONTEND(MIRROR)

- System Component – **Analysis**



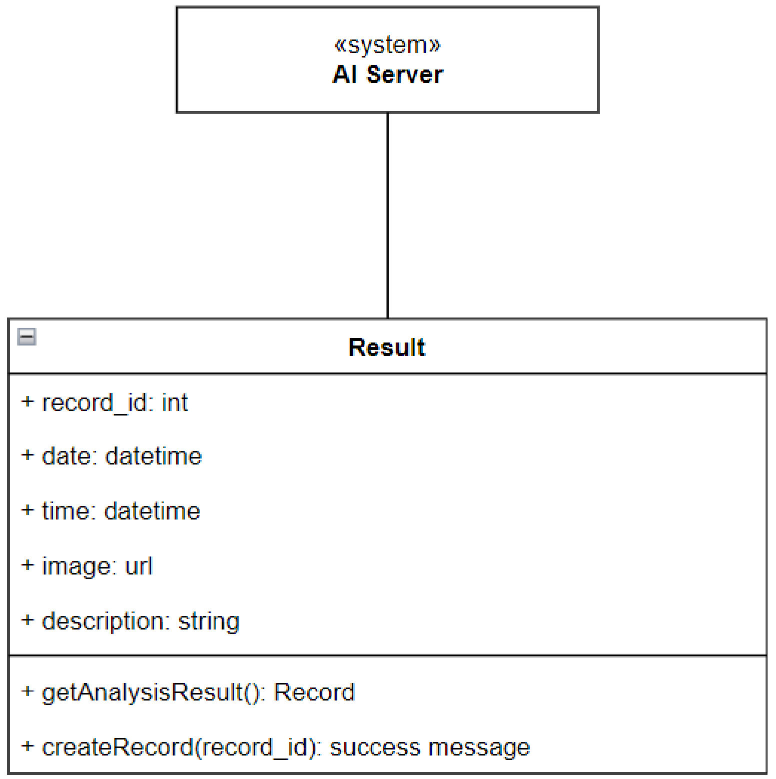
Class Diagram



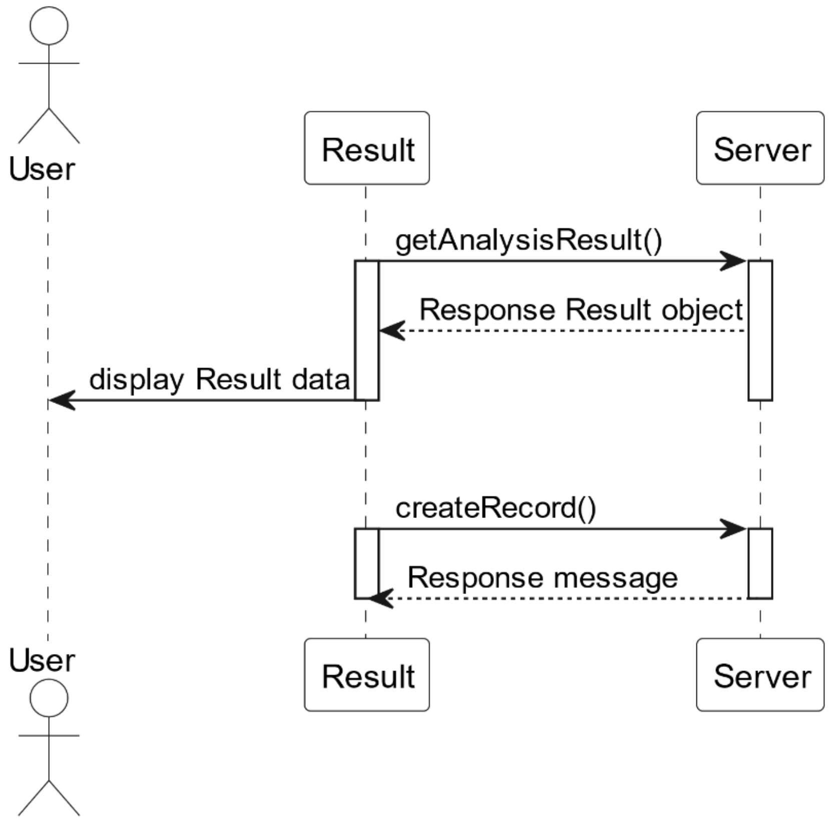
Sequence Diagram

SYSTEM ARCHITECTURE – FRONTEND(MIRROR)

- System Component – **Result**



Class Diagram



Sequence Diagram

SYSTEM ARCHITECTURE – FRONTEND

- **Protocol**

Frontend system of mobile application and smart mirror communicates with backend server and AI system through **HTTP**.

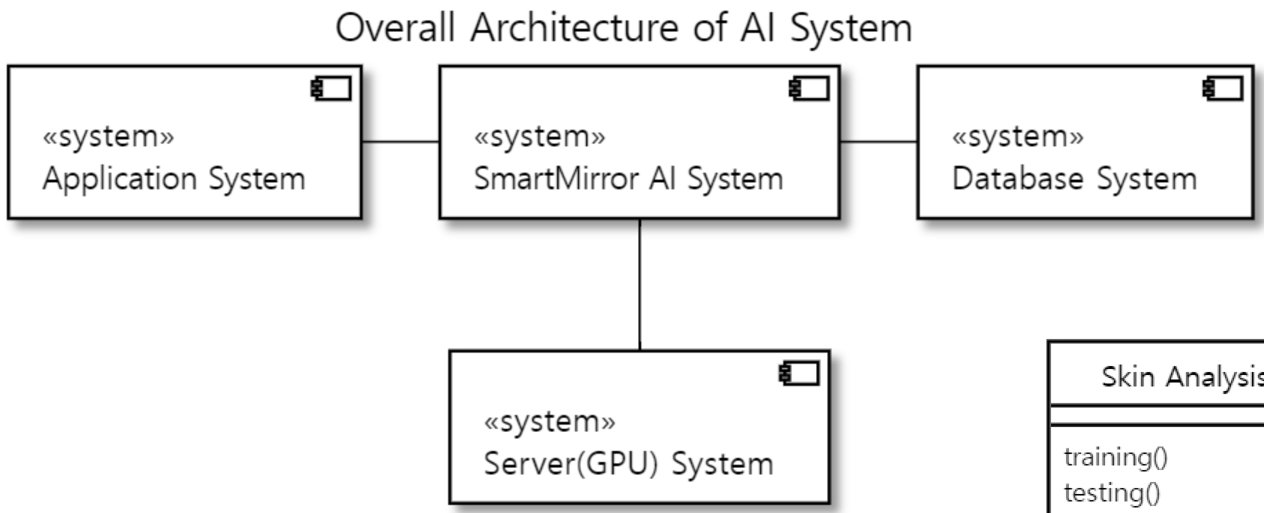
Both Request and Response use **JSON** format.

SYSTEM ARCHITECTURE - AI

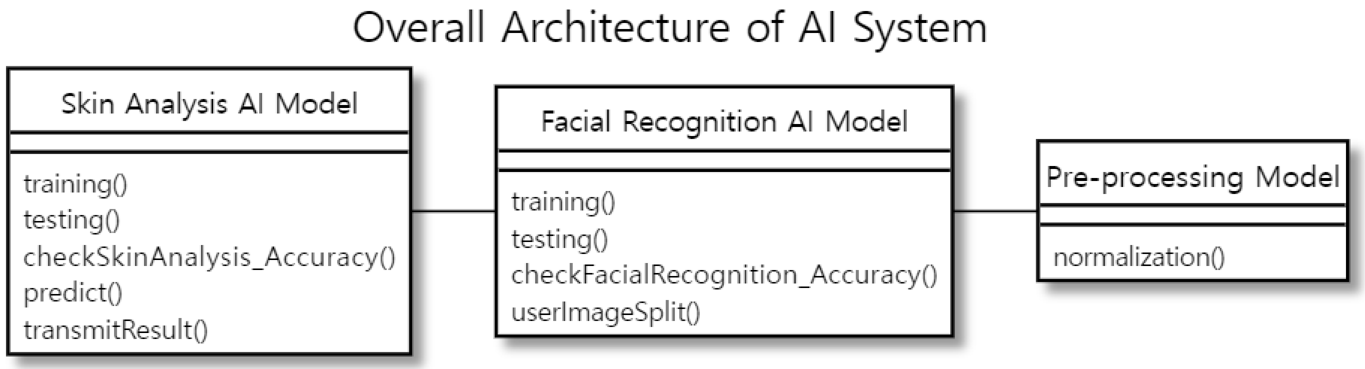
SDS

SYSTEM ARCHITECTURE - AI

- **Overall Architecture**



Context Model



Class Diagram

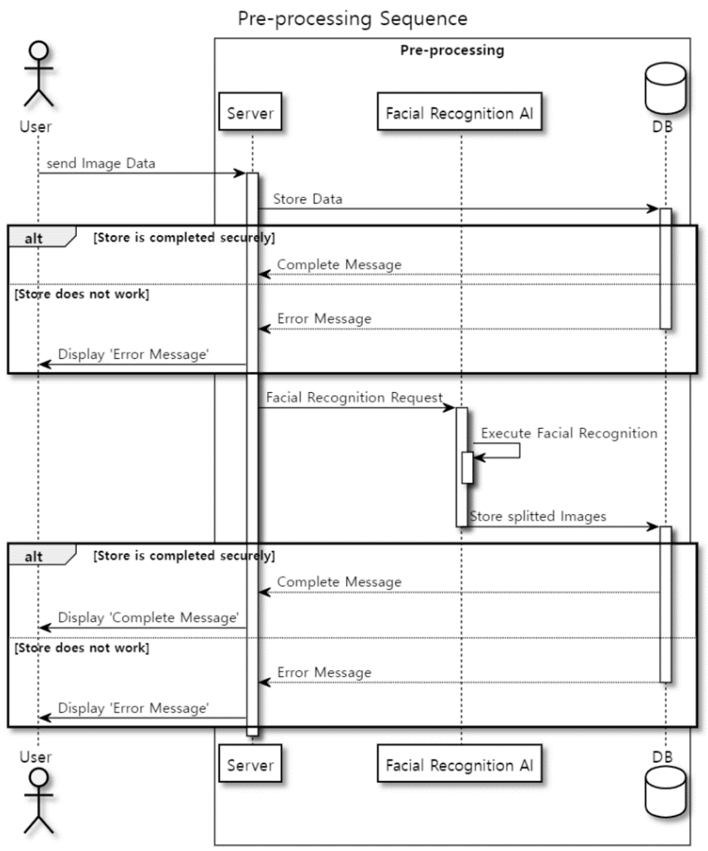
SYSTEM ARCHITECTURE - AI

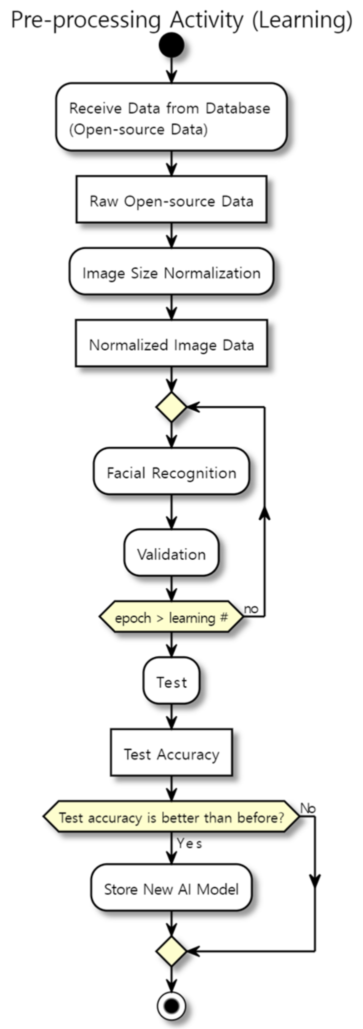
- Product information used in AI System

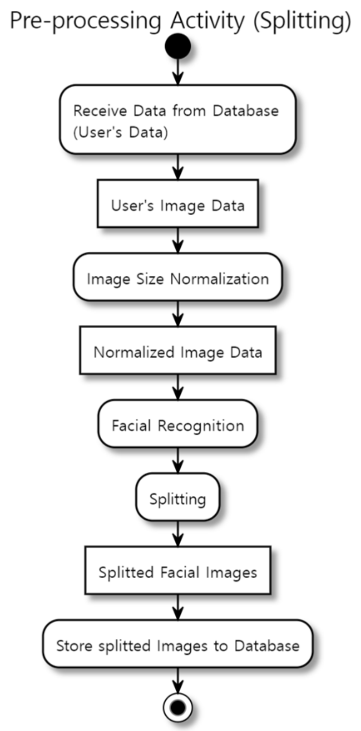
Name	Description
GPU	NVIDIA GeForce RTX 3060
Python	3.7.11 version, conda 4.10.3 version
Library	PyTorch 1.9.1 version, Numpy 1.21.2 version
CUDA	CUDA Toolkit 11.1.0

SYSTEM ARCHITECTURE - AI

- **Pre-processing System**



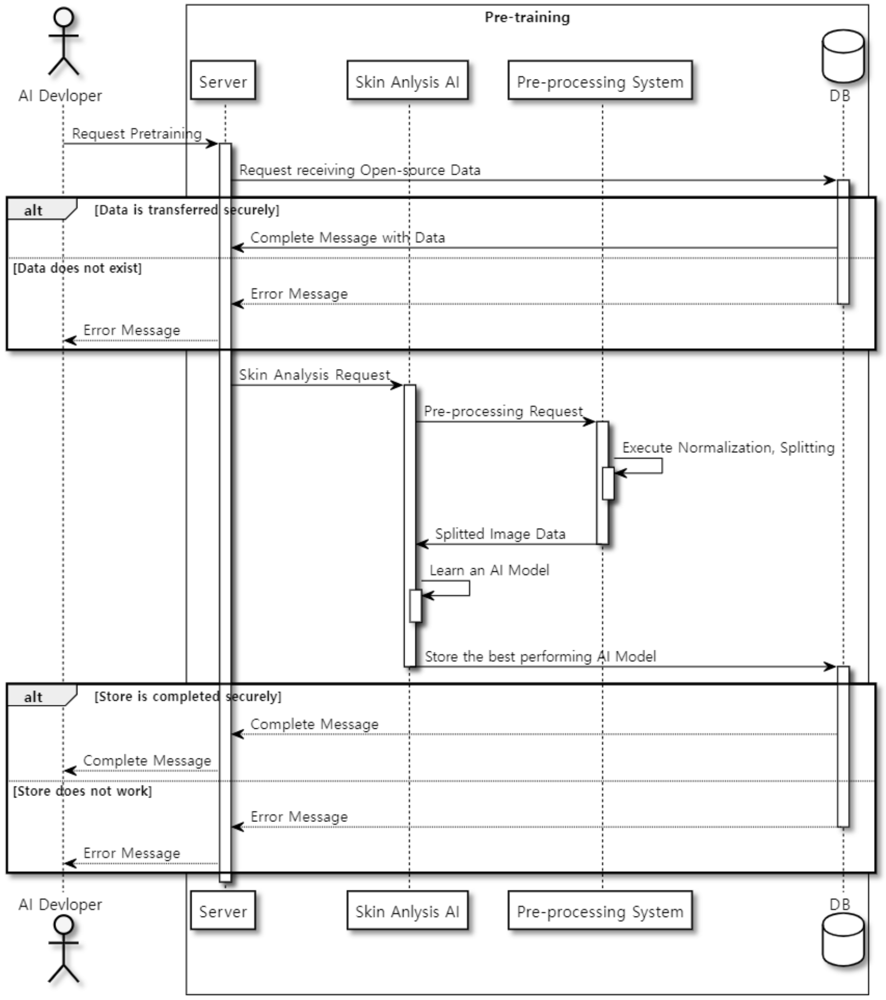




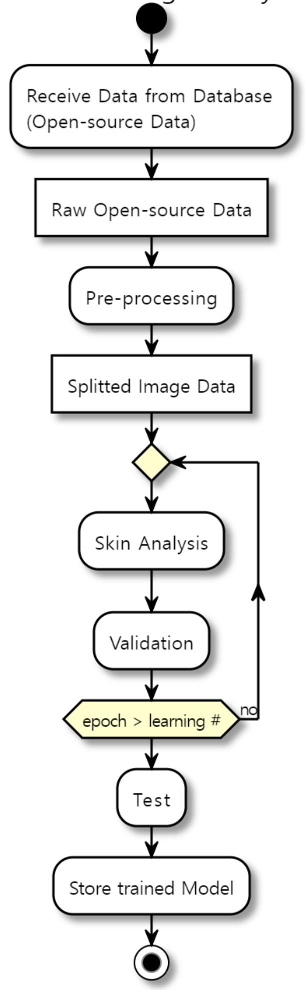
SYSTEM ARCHITECTURE - AI

- **Pre-training System**

Pre-training Sequence

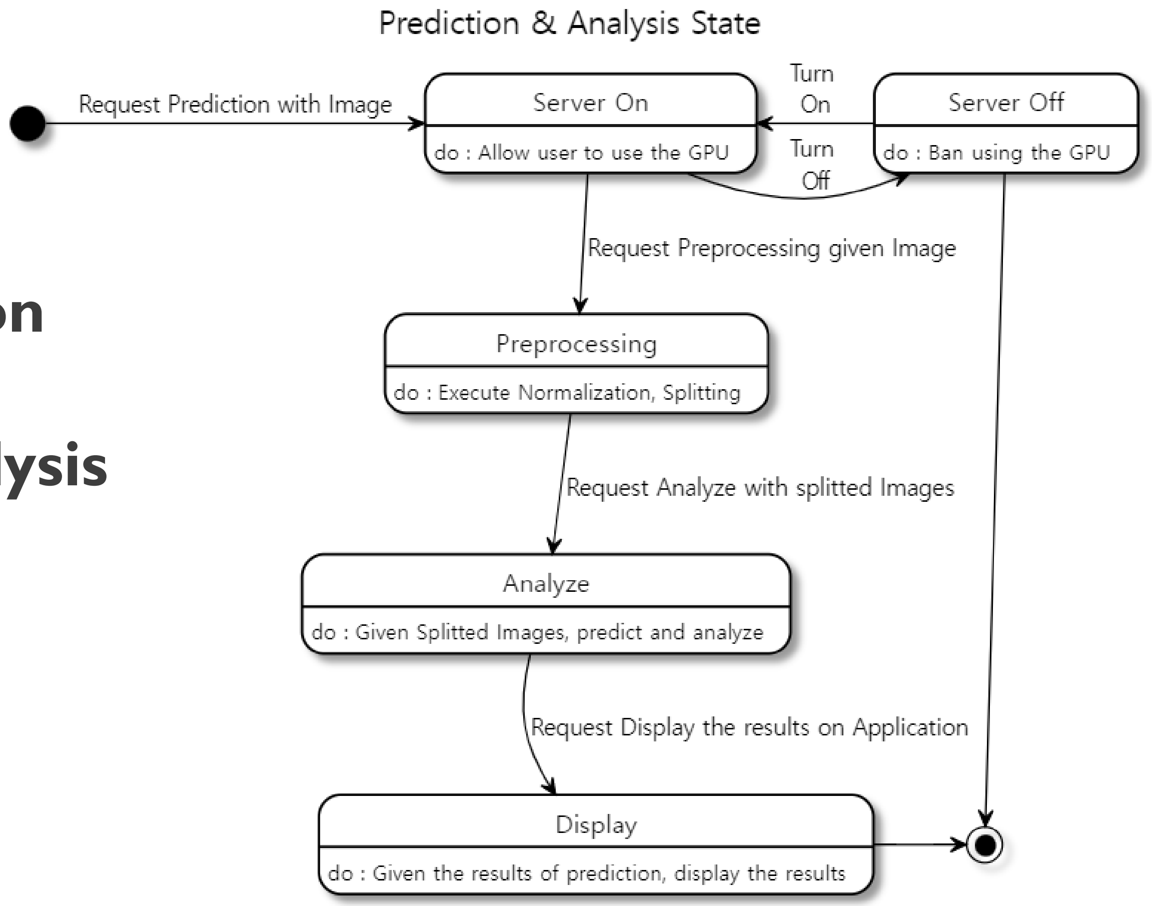


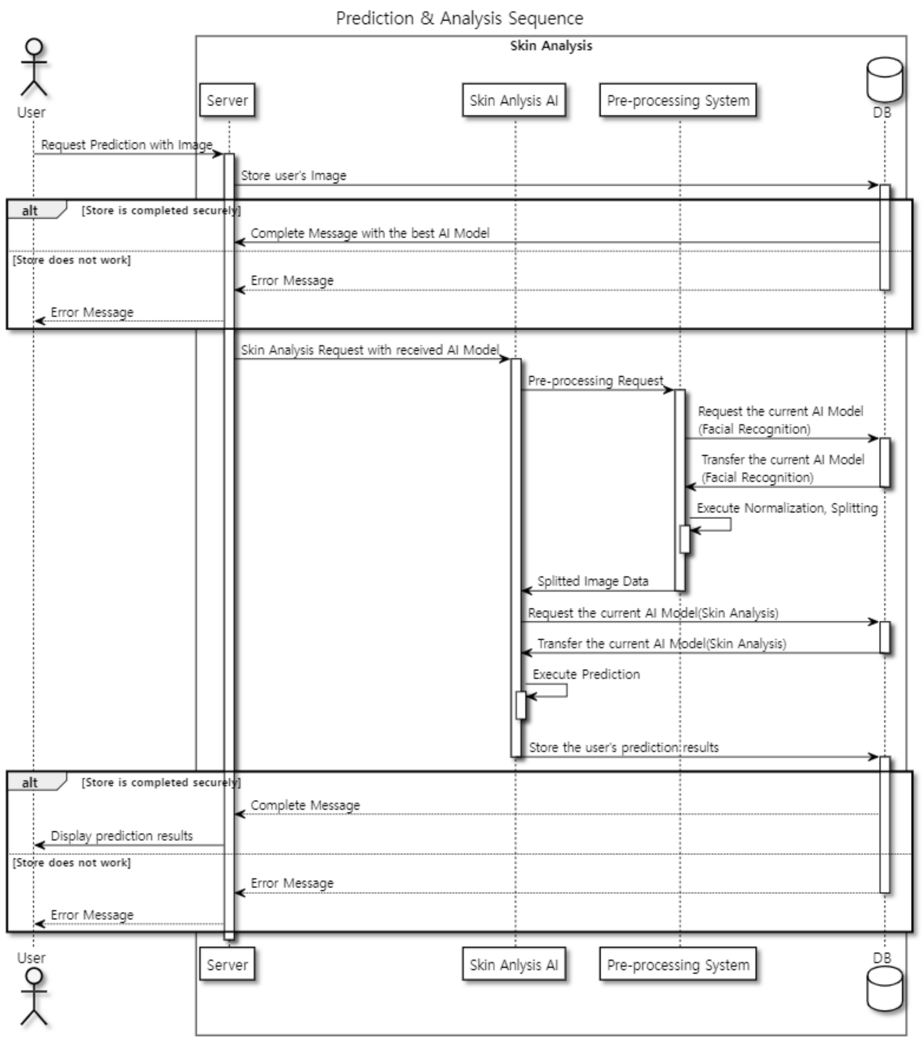
Pre-training Activity



SYSTEM ARCHITECTURE - AI

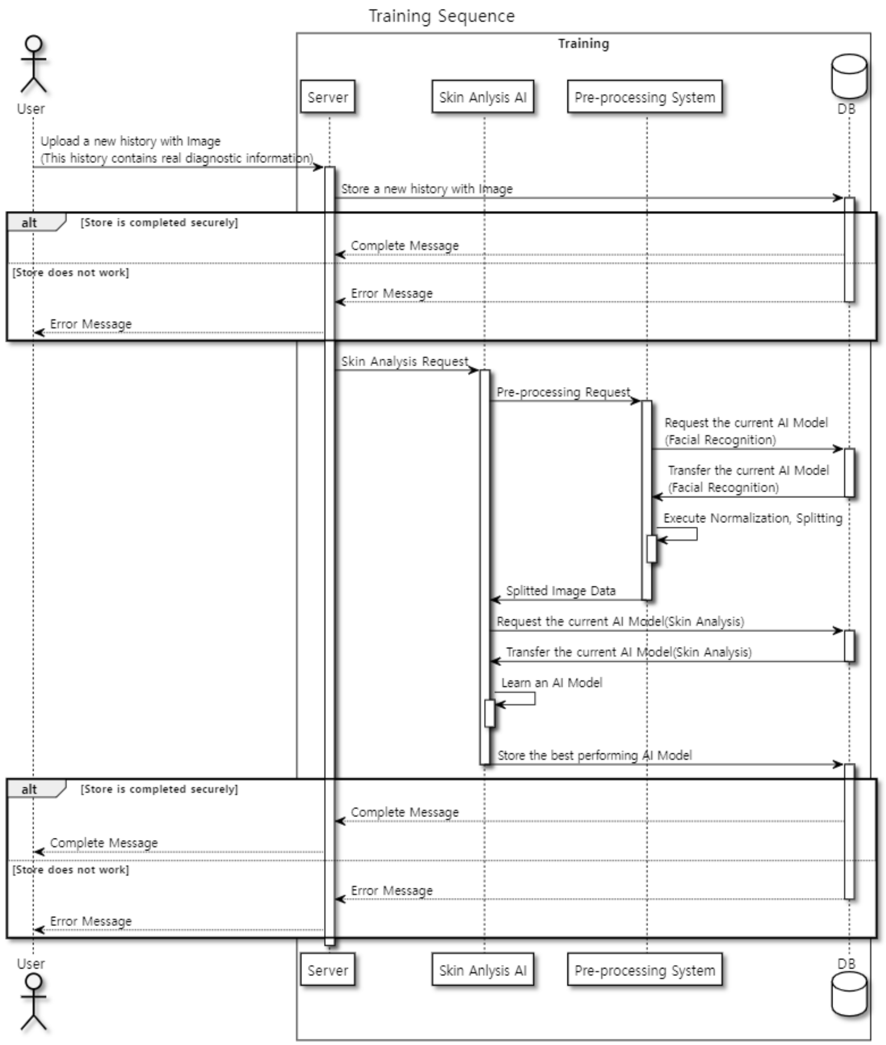
- **Prediction and Skin analysis**


Prediction & Analysis State
Request Prediction with Image
Server On
do : Allow user to use the GPU
Turn On
Turn Off
Server Off
do : Ban using the GPU
Request Preprocessing given Image
Preprocessing
do : Execute Normalization, Splitting
Request Analyze with splitted Images
Analyze
do : Given Splitted Images, predict and analyze
Request Display the results on Application
Display
do : Given the results of prediction, display the results


Prediction & Analysis Sequence
Skin Analysis
User
Server
Skin Anlysis AI
Pre-processing System
DB
Request Prediction with Image
Store user's Image
alt
[Store is completed securely]
Complete Message with the best AI Model
[Store does not work]
Error Message
Error Message
Skin Analysis Request with received AI Model
Pre-processing Request
Request the current AI Model (Facial Recognition)
Transfer the current AI Model (Facial Recognition)
Execute Normalization, Splitting
Splitted Image Data
Request the current AI Model(Skin Analysis)
Transfer the current AI Model(Skin Analysis)
Execute Prediction
Store the user's prediction results
alt
[Store is completed securely]
Complete Message
Display prediction results
[Store does not work]
Error Message
Error Message
User
Server
Skin Anlysis AI
Pre-processing System
DB

SYSTEM ARCHITECTURE - AI

- **Training System using User's data System**


Training Sequence
Training
User
Server
Skin Anlysis AI
Pre-processing System
DB
Upload a new history with Image (This history contains real diagnostic information)
Store a new history with Image
alt [Store is completed securely]
Complete Message
[Store does not work]
Error Message
Error Message
Skin Analysis Request
Pre-processing Request
Request the current AI Model (Facial Recognition)
Transfer the current AI Model (Facial Recognition)
Execute Normalization, Splitting
Splitted Image Data
Request the current AI Model(Skin Analysis)
Transfer the current AI Model(Skin Analysis)
Learn an AI Model
Store the best performing AI Model
alt [Store is completed securely]
Complete Message
Complete Message
[Store does not work]
Error Message
Error Message
User
Server
Skin Anlysis AI
Pre-processing System
DB

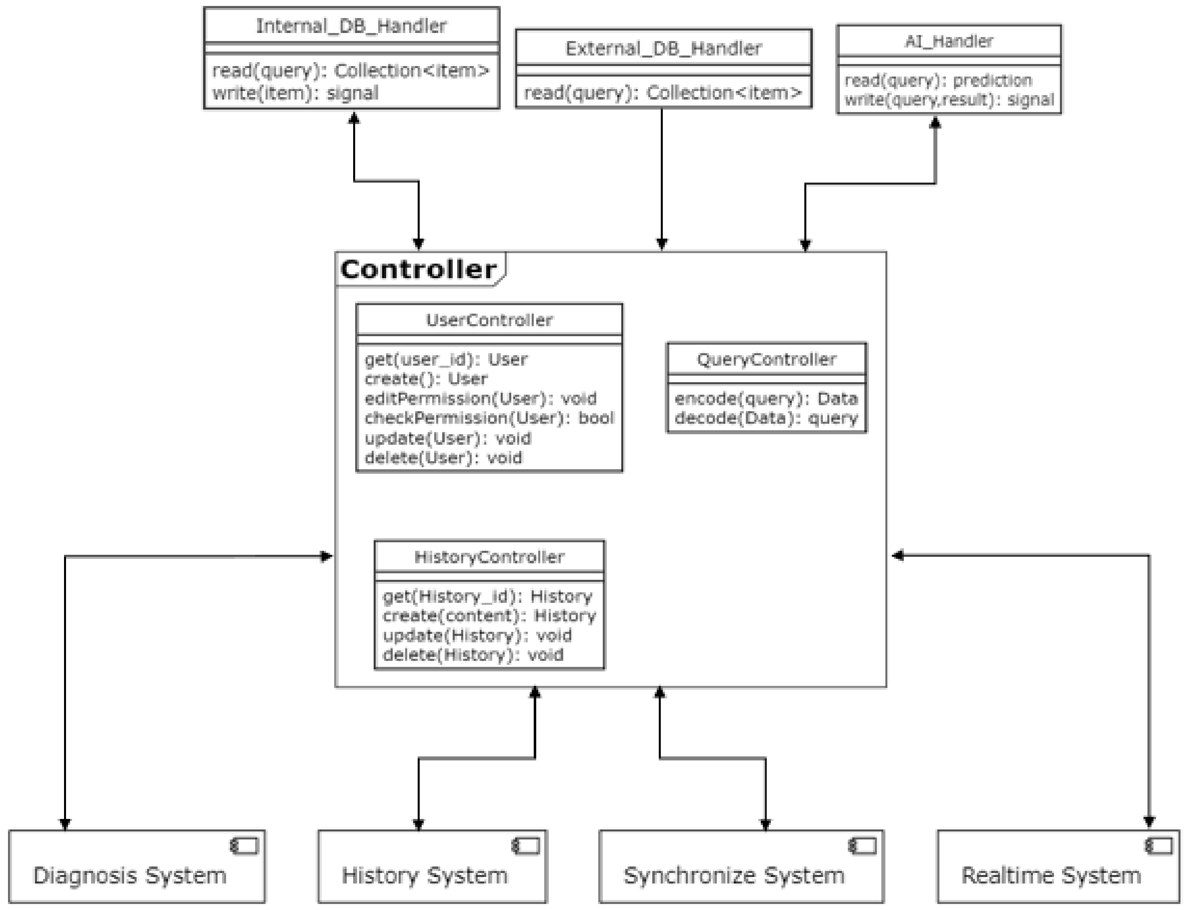
SYSTEM ARCHITECTURE - BACKEND

SDS

SYSTEM ARCHITECTURE - BACKEND

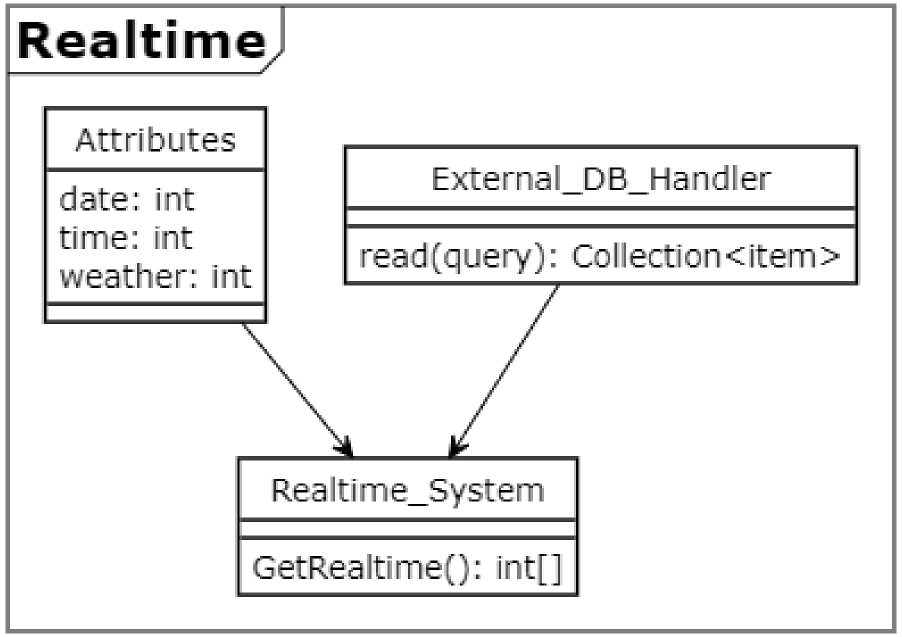
- **Overall Architecture of Backend System**

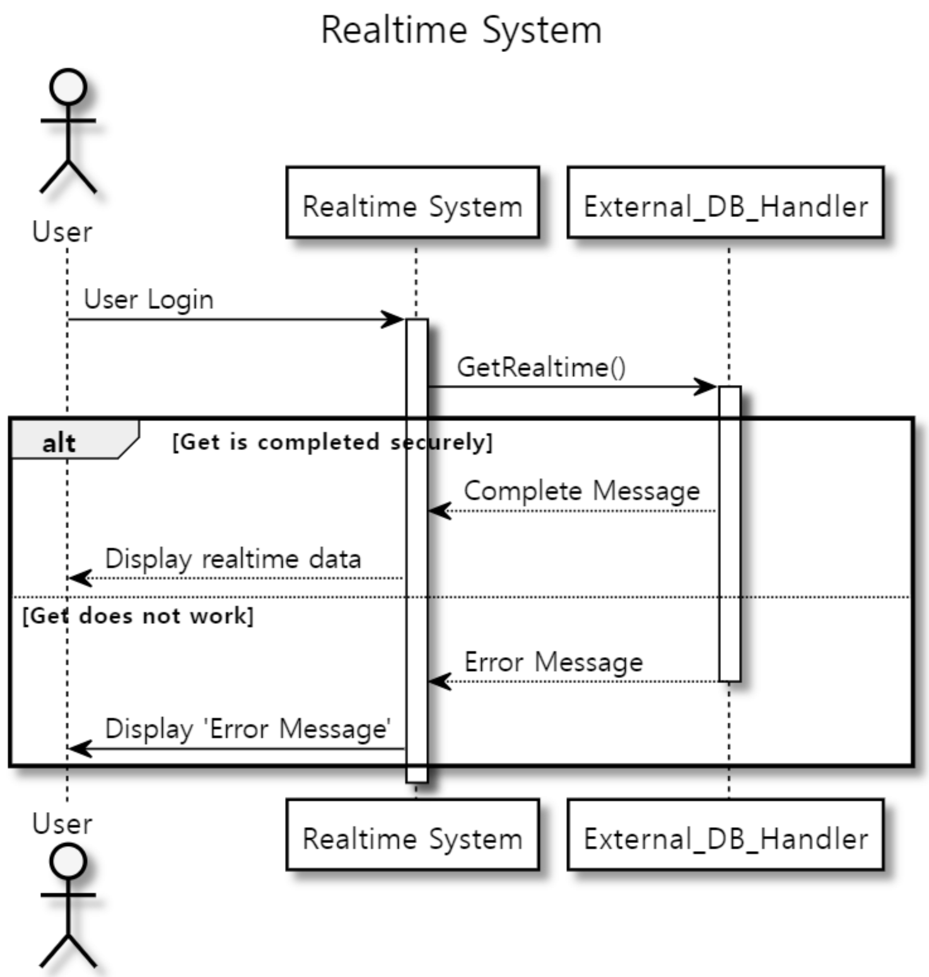
Overall Architecture of Backend System



SYSTEM ARCHITECTURE - BACKEND

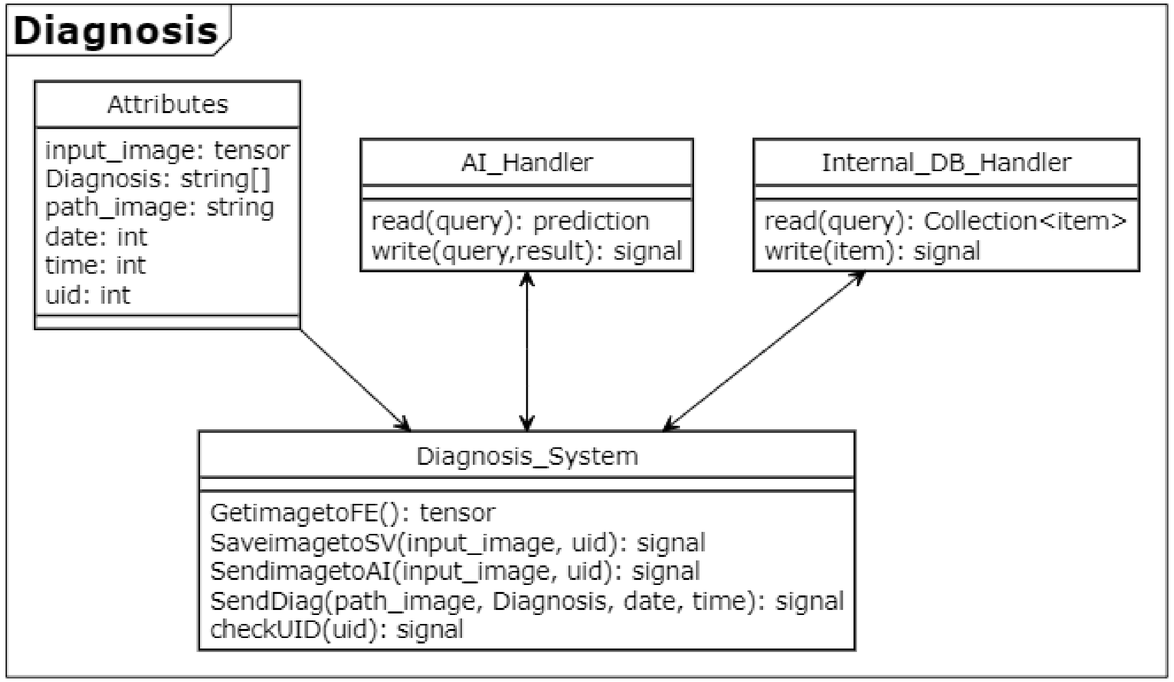
- **Realtime System**

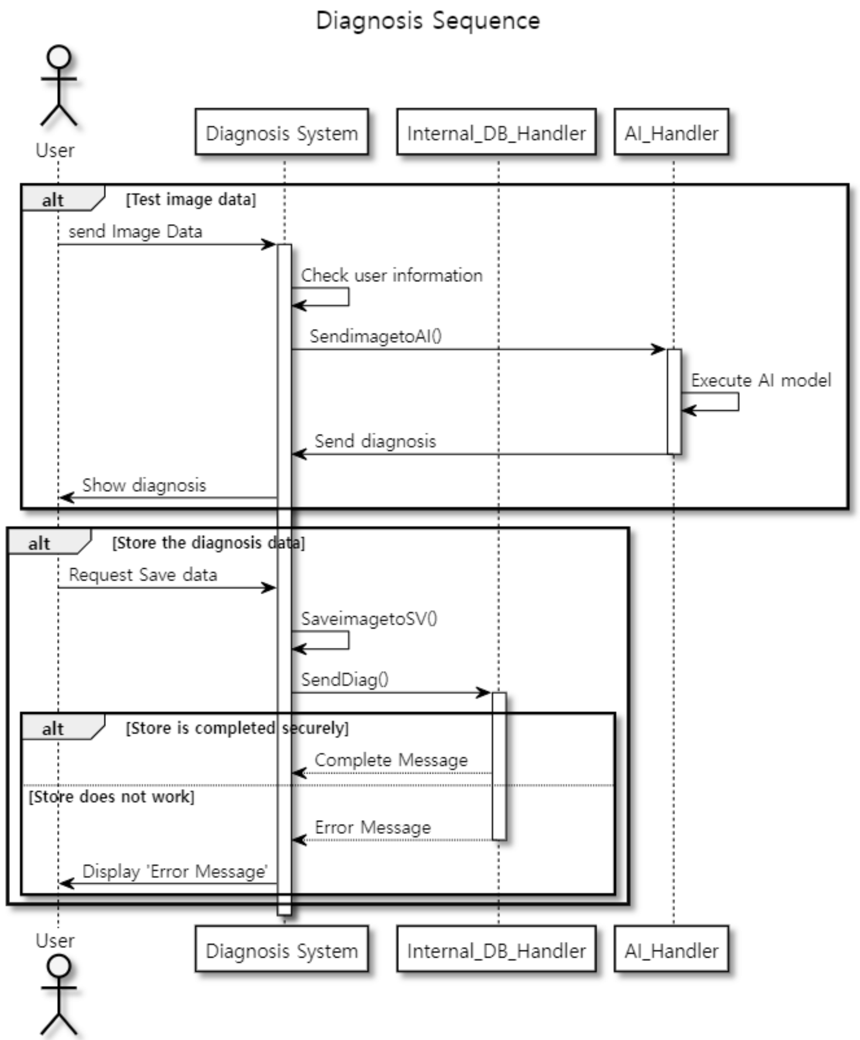




SYSTEM ARCHITECTURE - BACKEND

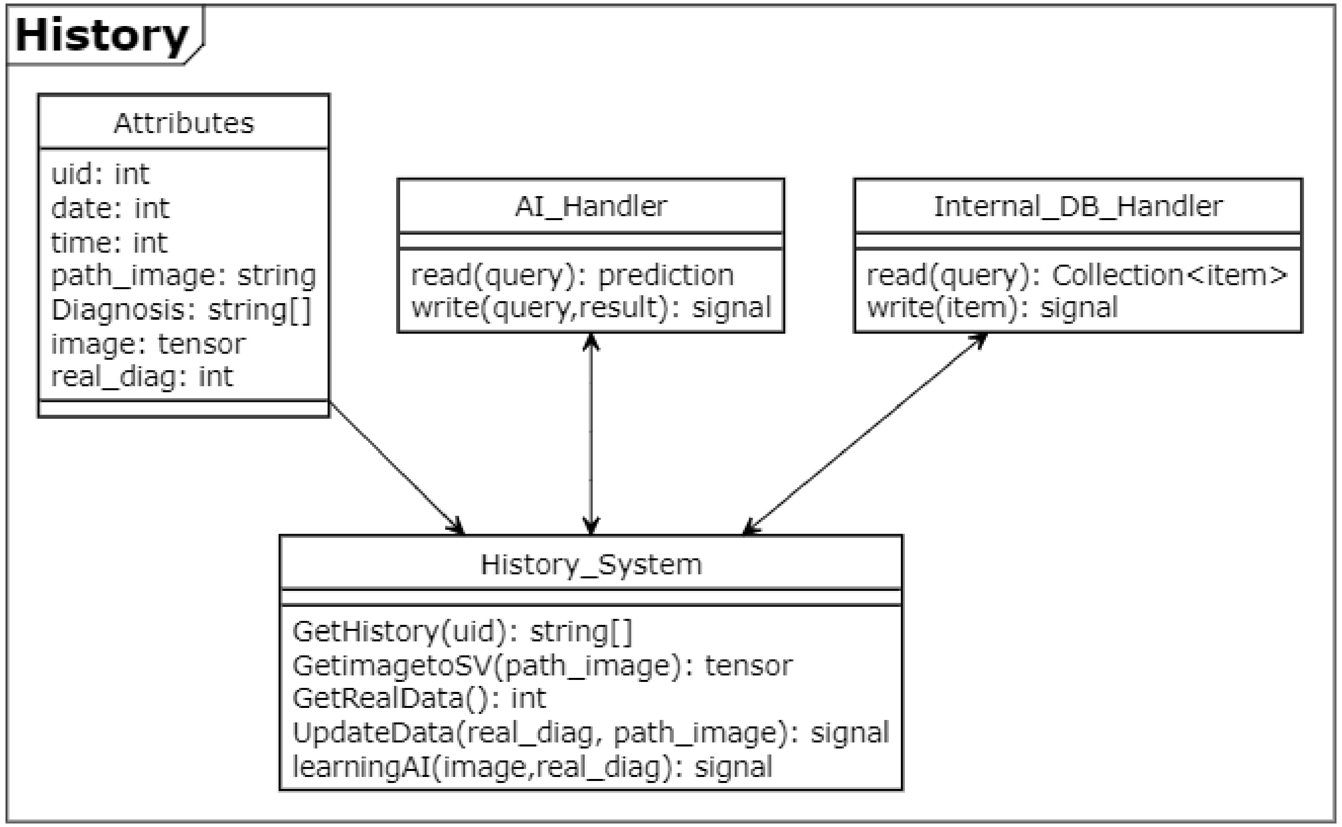
- **Diagnosis System**

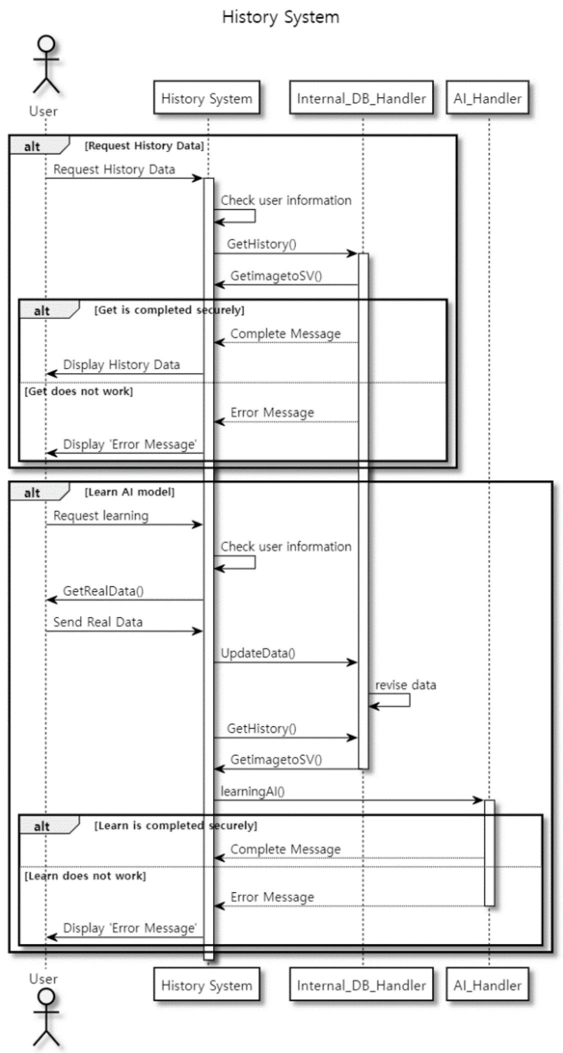




SYSTEM ARCHITECTURE - BACKEND

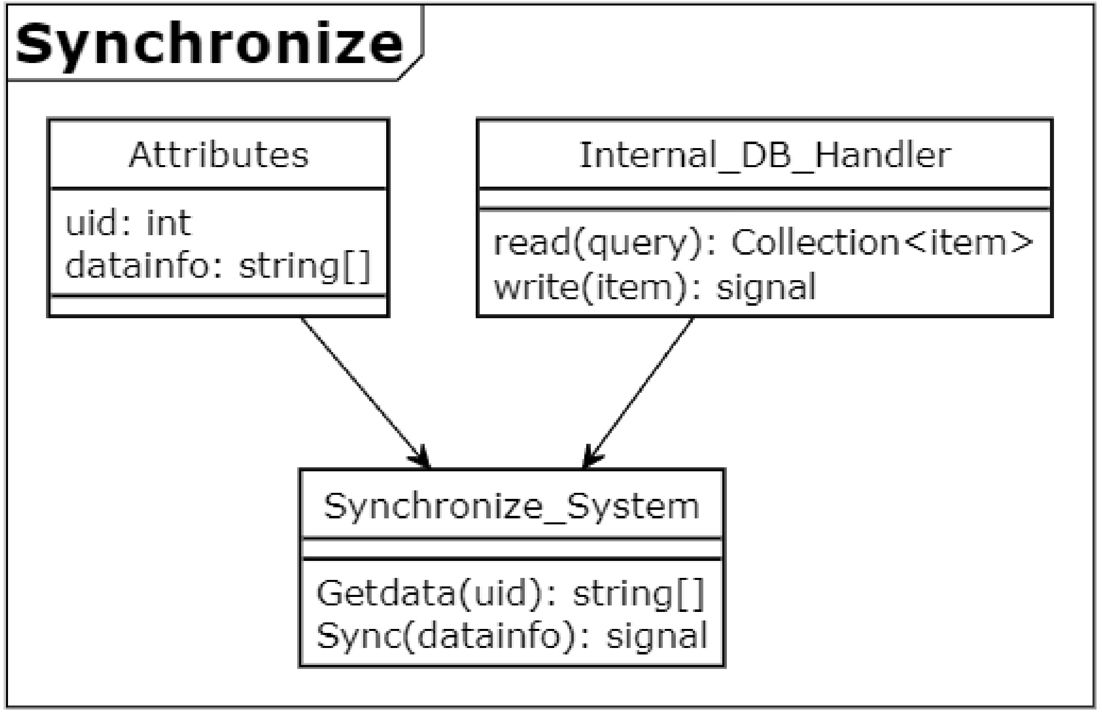
History System



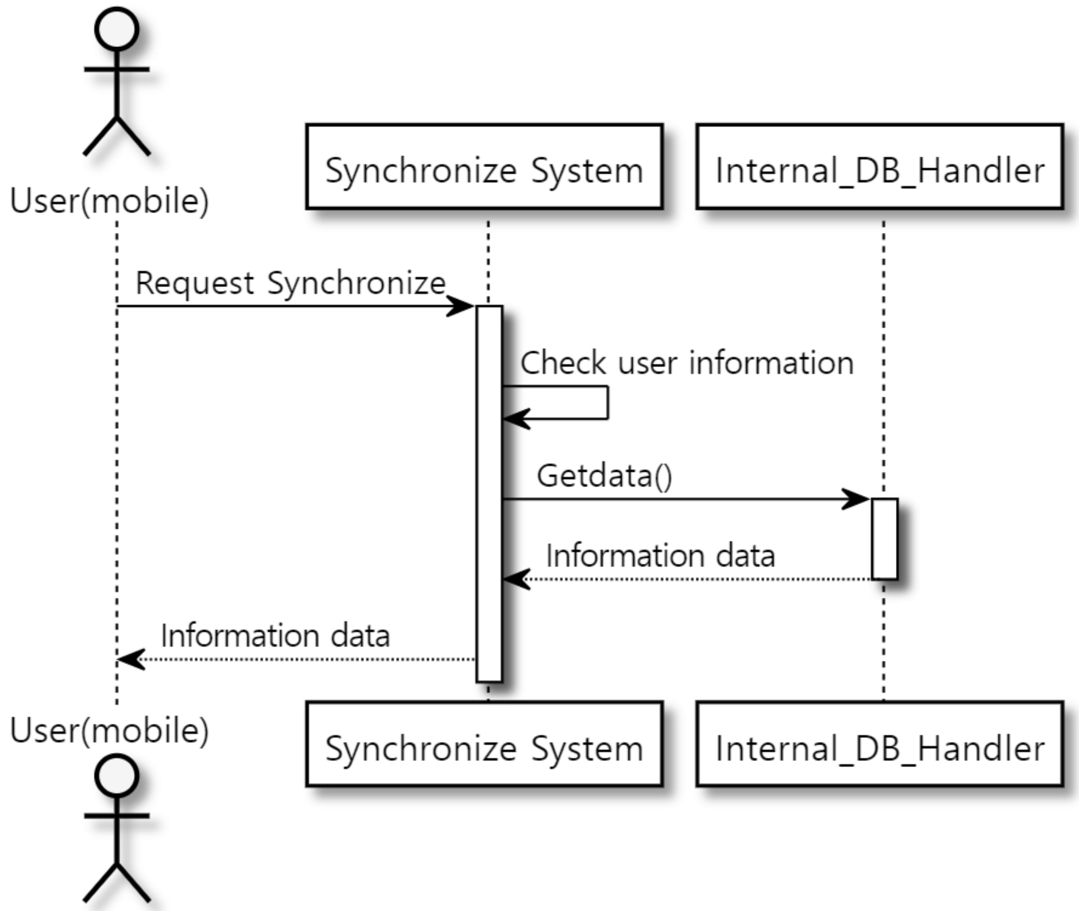


SYSTEM ARCHITECTURE - BACKEND

- **Synchronize System**



Synchronize System

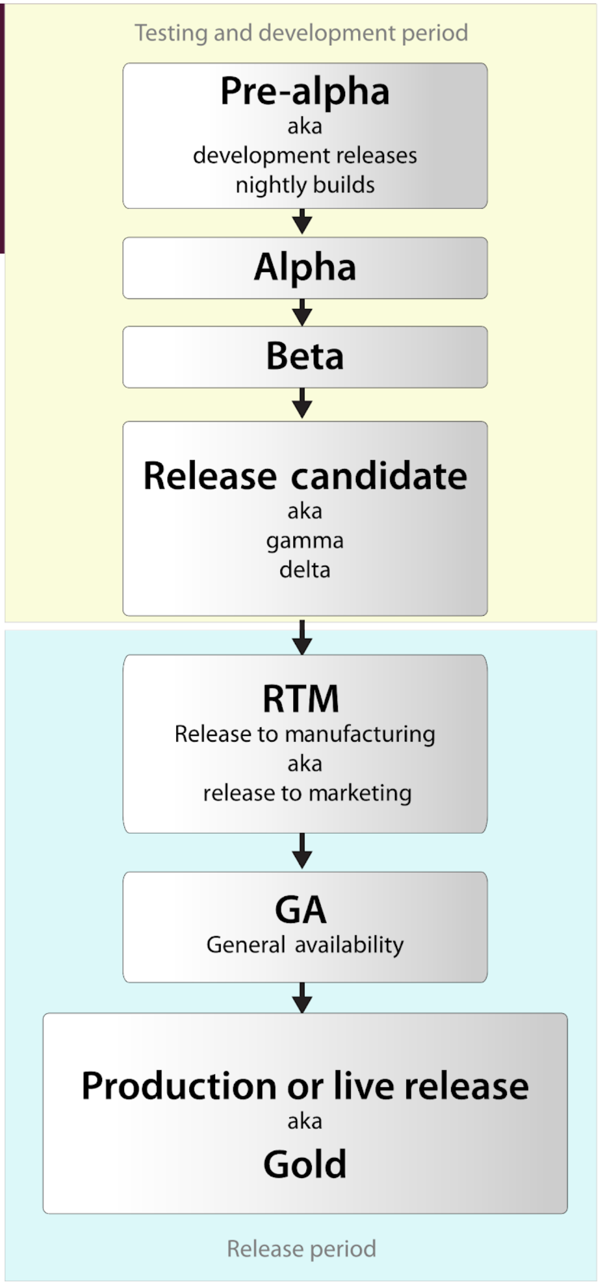


TESTING PLAN

SDS

TESTING PLAN – TESTING POLICY

- **Devlopment Test**: Performance, Reliability, Security
- **Release Test:** Test whether the implemented application can be used by users as planned or not
- **User Test**: About 30 test users will be selected and the test environment will be configured so that users can use the skin diagnosis function at their homes for 2 weeks.



TESTING PLAN – TESTING POLICY

- Test Case: Will be written so that developers can check the three tasks that they are targeting: functionality, performance, and unexpected access.
  - Testing for use in **unplanned directions**
  - Testing inducing **overload** of AI server with multi-user access
  - Testing in an **unstable network** environment
  - Testing in an environment with **smart mirror installed**

DEVELOPMENT PLAN

SDS

DEVELOPMENT PLAN - FRONTEND

- **Adobe Illustrator**: Used to create background images and icons that make up the Smart Mirror UI
- **Adobe Xd**: Used to design and visualize UI/UX for mobile applications
- **Xcode**: Used to create an iPhone version of a mobile application that assists the smart mirror system
- **Android Studio**: Used to create an Android version of a mobile application that assists the smart mirror system

DEVELOPMENT PLAN - BACKEND

- **mongoDB** - Used to store user information, photo data, and diagnosis result information of the smart mirror system
- **Node.js** - Used to configure Javascript based servers
- **Jenkins** - Used to build a continuous integration and continuous delivery environment for software development

DEVELOPMENT PLAN - AI

PyTorch

Python-based open source machine learning library

DEVELOPMENT PLAN - CONSTRAINTS

- **Access rights** must be required for smart mirrors and user devices such as camera and internal data access, and changes in rights must be possible within the device.
- The camera and display options of the smart mirror can be adjusted for uniformity of data, and **user consent** must be obtained for this.
- User's data is classified as a kind of medical data, so **security** needs to be paid more attention.
- The user's photo for skin diagnosis must be taken only through the smart mirror, and the data uniformity must be ensured by flexibly changing the smart mirror's settings.
- The skin diagnosis results of the smart mirror are for reference only and **have no legal effect.**
- The UI of the smart mirror and mobile application should be **intuitive** for easy use, and the arrangement, control method and overall process of objects should be considered for UX improvement.

THANK YOU

TEAM 13

INTRODUCTION TO SOFTWARE ENGINEERING TEAM 13

조재훈, VINCENT PAN, 박민서 , 설채은 , 이재혁 , 정민석 , 백송현