

<<**Bloodline DNA Testing**

**Service Management System**>>

**Software Design Specification**

– HCM, June 2025 –

**Record of changeS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **A\* M, D** | **In charge** | **Change Description** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

\*A - Added M - Modified D - Deleted

**Table of Contents**

[I. Overview 4](#_heading=h.gjdgxs)

[1. Code Packages 4](#_heading=h.30j0zll)

[2. Architecture Diagram 5](#_heading=h.1fob9te)

[3. Database Diagram 5](#_heading=h.1fob9te)

[a. Database Schema 5](#_heading=h.3znysh7)

[b. Table Description 6](#_heading=h.2et92p0)

[II. Code Designs 6](#_heading=h.tyjcwt)

[1. Class Diagram 7](#_heading=h.1t3h5sf)

[2. Activity Diagram(s) 8](#_heading=h.4d34og8)

[3. Sequence Diagram(s) 11](#_heading=h.2s8eyo1)

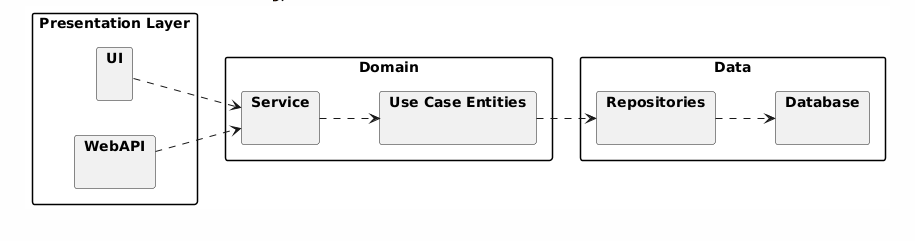
[4. Deloyment Diagram 13](#_heading=h.1t3h5sf)

[5. Requirement Diagram 14](#_heading=h.2s8eyo1)

[6. Use Case Diagram 15](#_heading=h.1t3h5sf)

# I. Overview

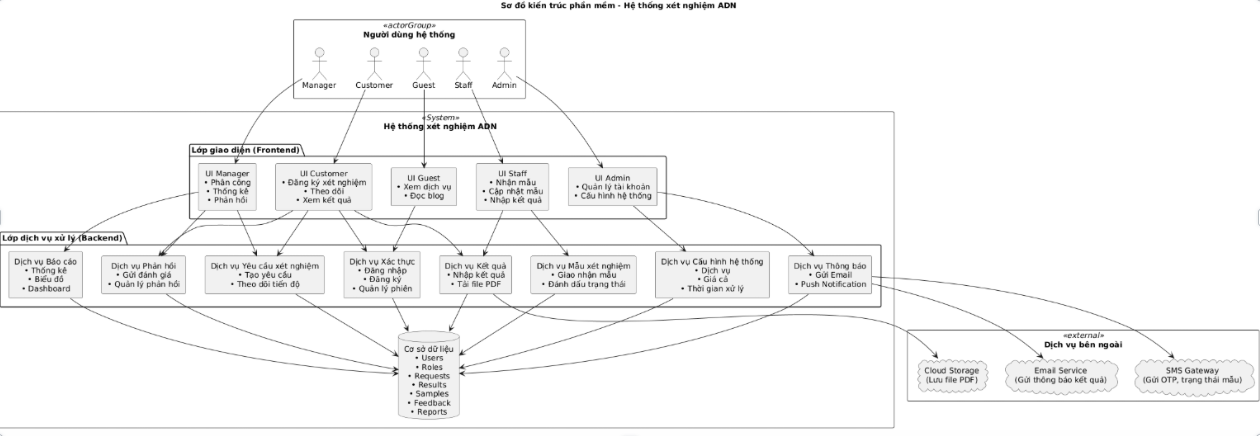
## 1. Code Packages



***Package descriptions***

|  |  |  |
| --- | --- | --- |
| **No** | **Package** | **Description** |
| *01* | *UI* | *Contains all user interface components, including screens and interactive pages allowing users to log in, view results, and submit feedback.* |
| *02* | *WebAPI* | *Manages the communication layer between client and server, handling incoming HTTP requests and routing them to the appropriate service logic.* |
| *03* | *Service* | *Implements business rules and coordinates operations between the domain layer and data access. Processes requests, authentication, reporting, etc.* |
| *04* | *Use case entities* | *Defines domain models and core entities (e.g., User, TestRequest, Sample) reflecting business use cases and functionalities.* |
| *05* | *Repositories* | *Provides data access logic, offering methods for CRUD operations with underlying storage using SQL or ORM frameworks.* |
| *06* | *Database* | *Manages schema definitions, connection configs, migration scripts, and stores persistent application data including users, requests, and results.* |

**2.Architecture Diagram**

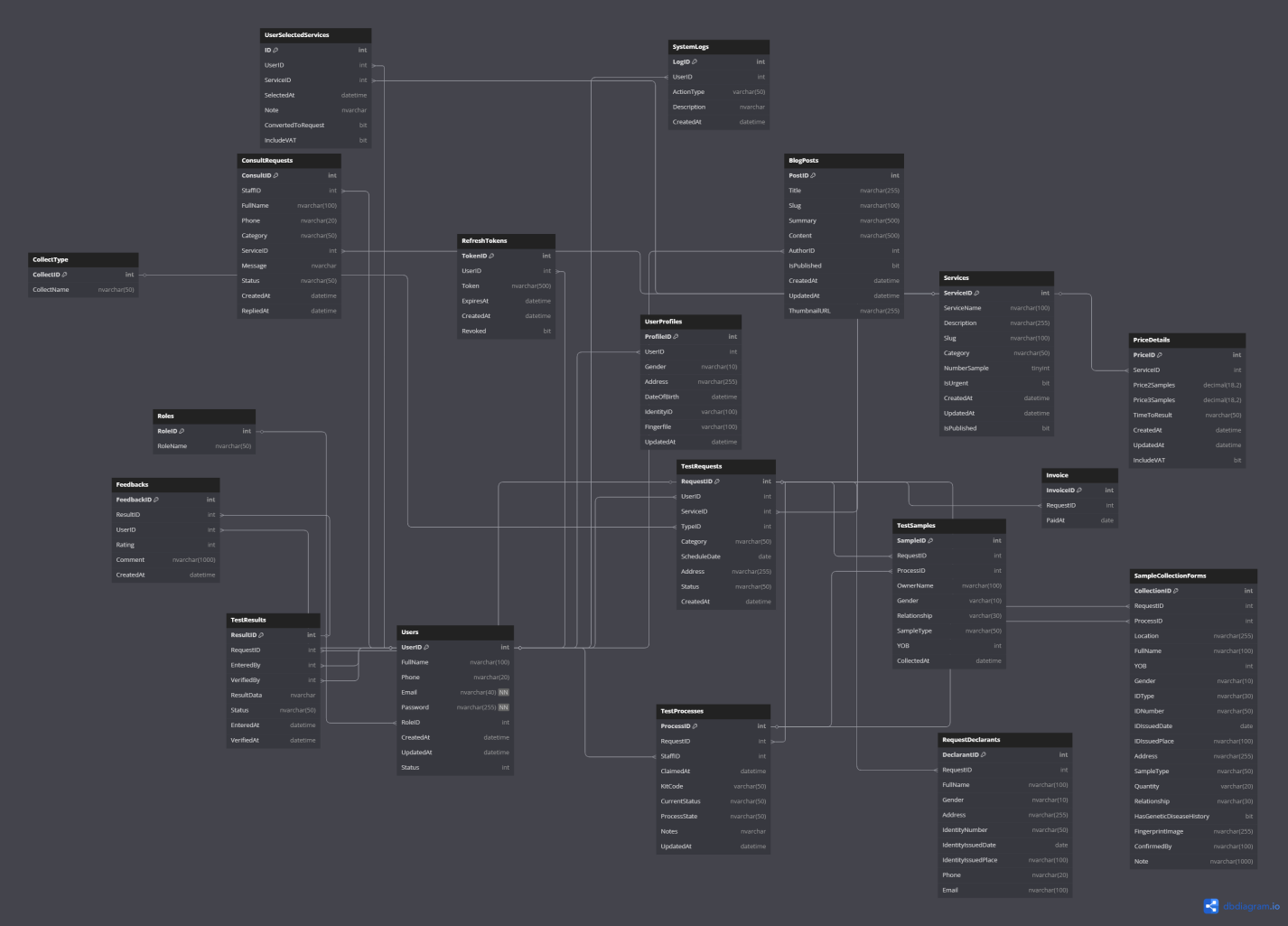
******

***Architecture descriptions***

|  |  |  |
| --- | --- | --- |
| **No** | **Layer** | **Description** |
| *01* | *System users* | *Includes roles: Manager, Customer, Guest, Staff and Admin. Each role interacts with its own interface and performs corresponding functions in the system.* |
| *02* | *Frontend* | *The user interface is divided by role: Guest (view services, blog), Customer (register, track, view results), Staff (receive samples, enter results), Manager (statistics, validate results), Admin (manage accounts, services).* |
| *03* | *Backend Service Layer* | *Contains key services such as: test request processing, user role authentication, sample processing, saving and exporting results, feedback management, service configuration, statistics and sending notifications.* |
| *04* | *Database* | *Where the entire system data is stored: users, roles, requests, test samples, results, feedback and reports.* |
| *05* | *External Services* | *Including Cloud Storage (save result files), Email Service (send results), SMS Gateway (send OTP, sample status notification).* |

## 3. Database Diagram

### a. Database Schema

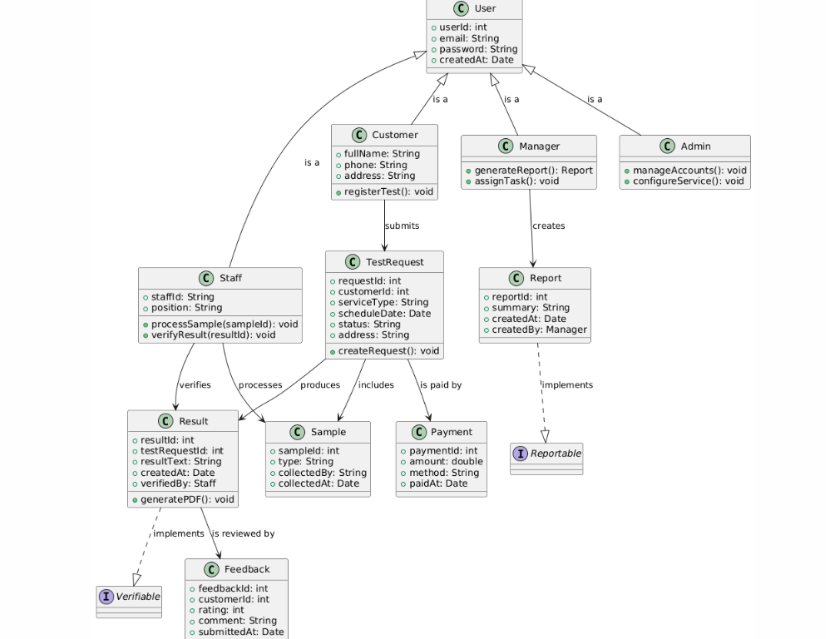


### b. Table Description

|  |  |  |
| --- | --- | --- |
| **No** | **Table** | **Description** |
| *01* | *Role* | *Contains system role types such as guest, customer, employee, manager and admin. Helps to assign access rights according to function.* |
| *02* | *Users* | *Stores user account information including name, email, encrypted password, and system role. It is the central table that identifies users in the entire system.* |
| *03* | *UserProfiles* | *Contains system role types such as guest, customer, employee, manager and admin. Helps to assign access rights according to function.* |
| *04* | *Services* | *Describes the types of testing services available, including service name, description, number of samples required, and display status.* |
| *05* | *PriceDetails* | *Set prices for each service based on the number of people giving samples and processing time. Helps calculate costs automatically.* |
| *06* | *TestRequests* | *Represents a user-submitted test request, containing service type, appointment time, collection address, and processing status.* |
| *07* | *RequestedDonators* | *Save details about the person who requested the test for each request, including age, contact information.* |
| *08* | *TestSamples* | *Represents specific biological samples such as hair, blood, etc. associated with each tested person, including owner information and sample type.* |
| *09* | *SampleCollectionForm* | *Record details of the sample collection process, including collector information, location, time and sample storage conditions. This form is for administrative testing.* |
| *10* | *TestProcesses* | *Track test processing steps for each request, including test code, current status, and responsible staff.* |
| *11* | *TestResults* | *Save DNA analysis results, including who performed the test, reliability, verification status, and feedback if available.* |
| *12* | *Feedbacks* | *This table stores the information of customers who submitted reviews after receiving results, including number of stars, comments and submission time.* |
| *13* | *SystemLogs* | *This table records system activities such as logins, information changes, and user actions for monitoring purposes.* |
| *14* | *RefreshTokens* | *The table supports secure authentication by storing tokens used to refresh user login sessions.* |
| *15* | *CollectType* | *List of sample collection methods such as collection at the center, at home, etc. to serve sample classification and processing.* |
| *16* | *ConsultingRequests* | *A table that records customer consultation requests, including contact information, content, interested services, and time of request.* |
| *17* | *Invoices* | *The table records the billing information for each test request, including the billing code, payment date, and related request.* |
| *18* | *UserSelectedServices* | *A temporary table of services that the user has selected before officially submitting a test request.* |
| *19* | *BlogPosts* | *This table will store articles, news, and professional knowledge related to testing that are publicly displayed on the system.* |

# II. Code Designs

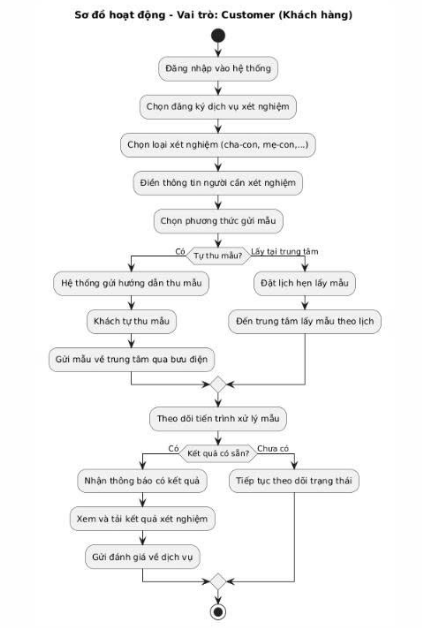
### 1. Class Diagram

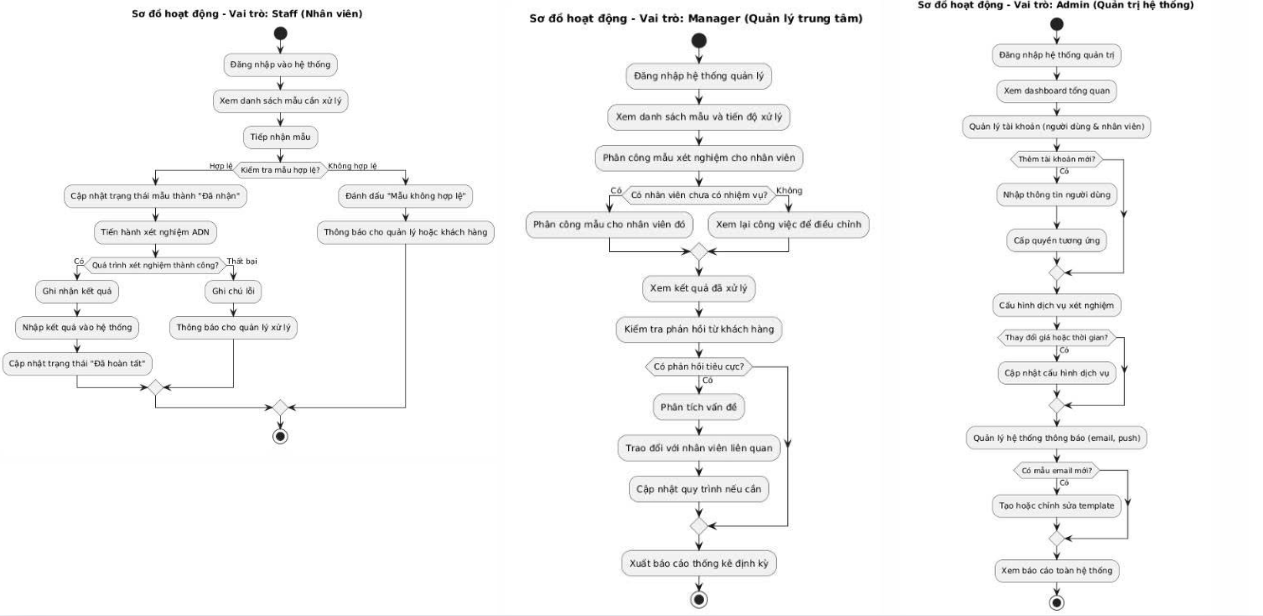
**

### 

### 2. Activity Diagram

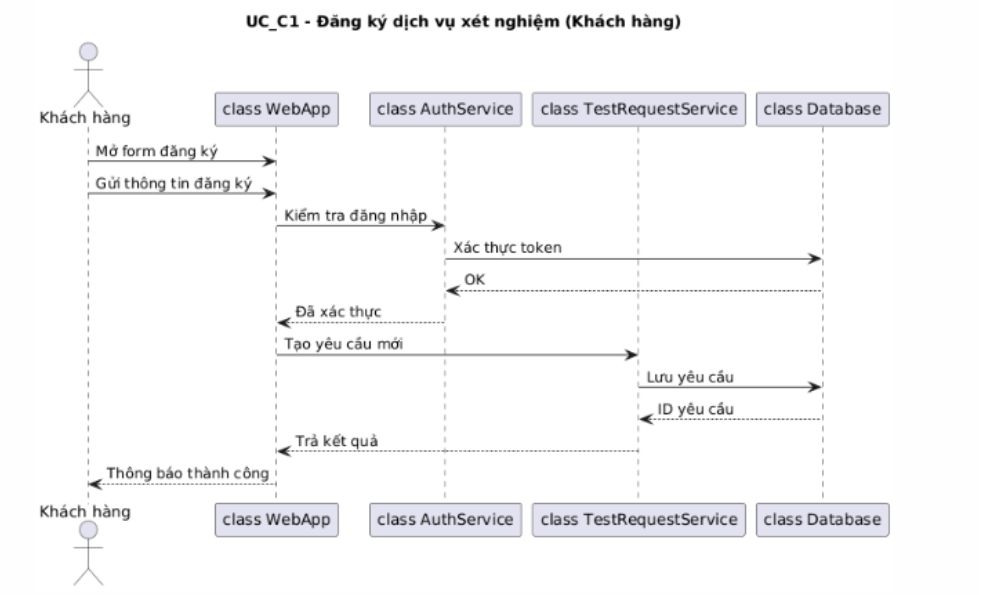
### 

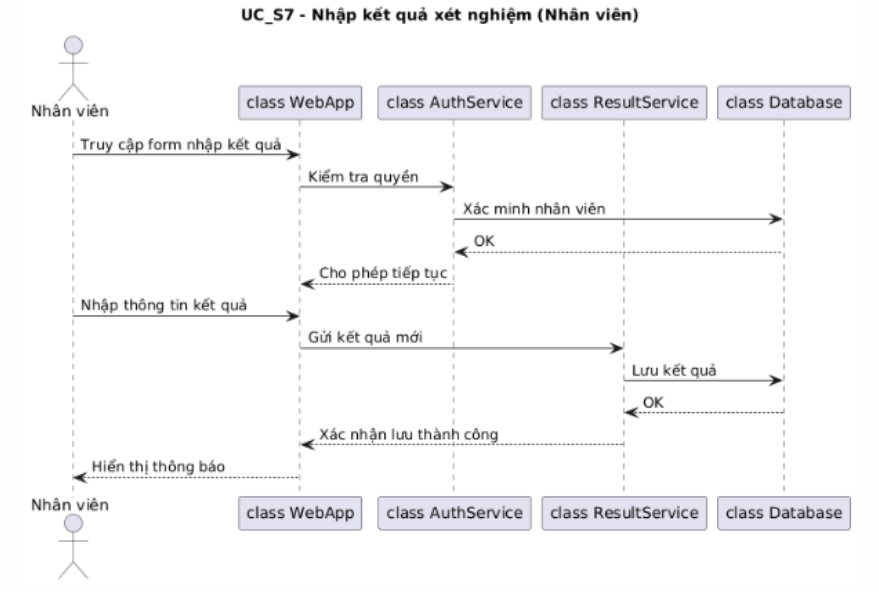


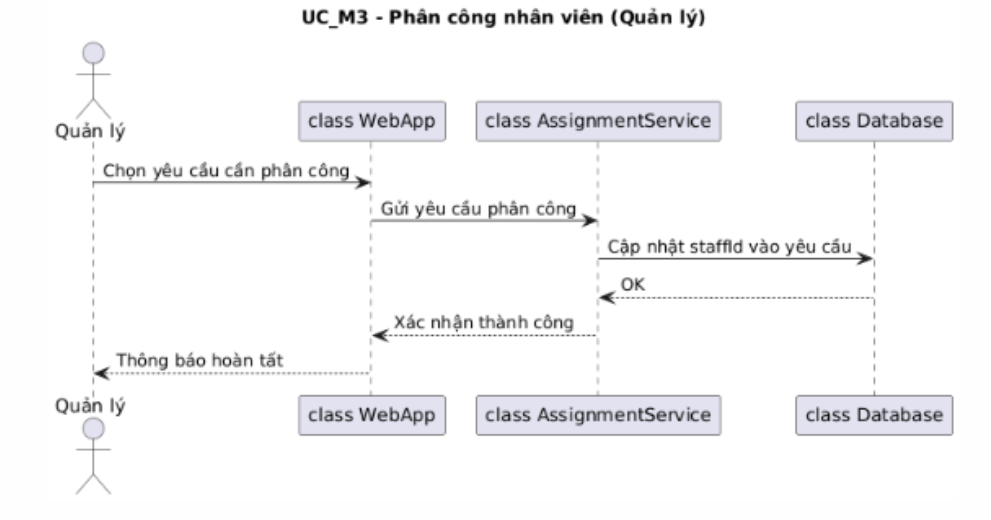


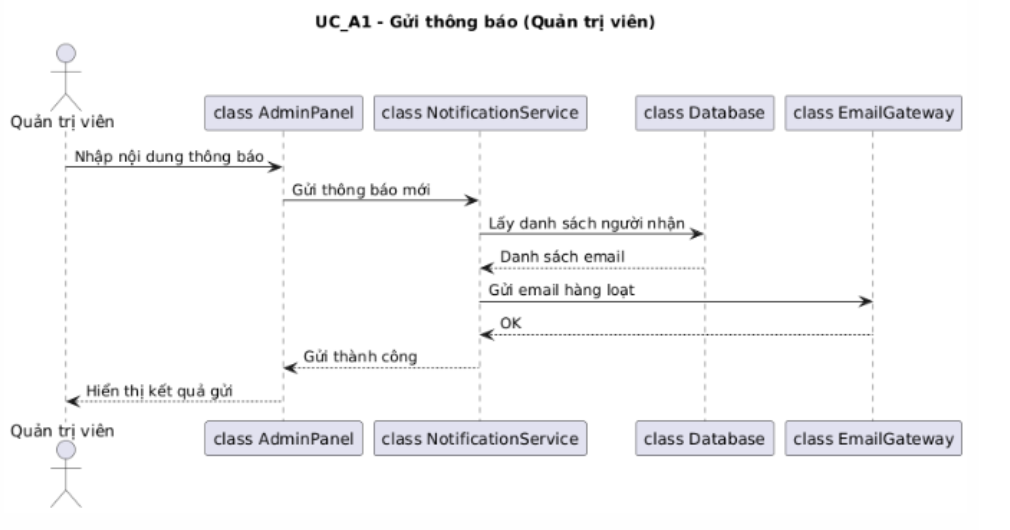
### 3. Sequence Diagram(s)

# 

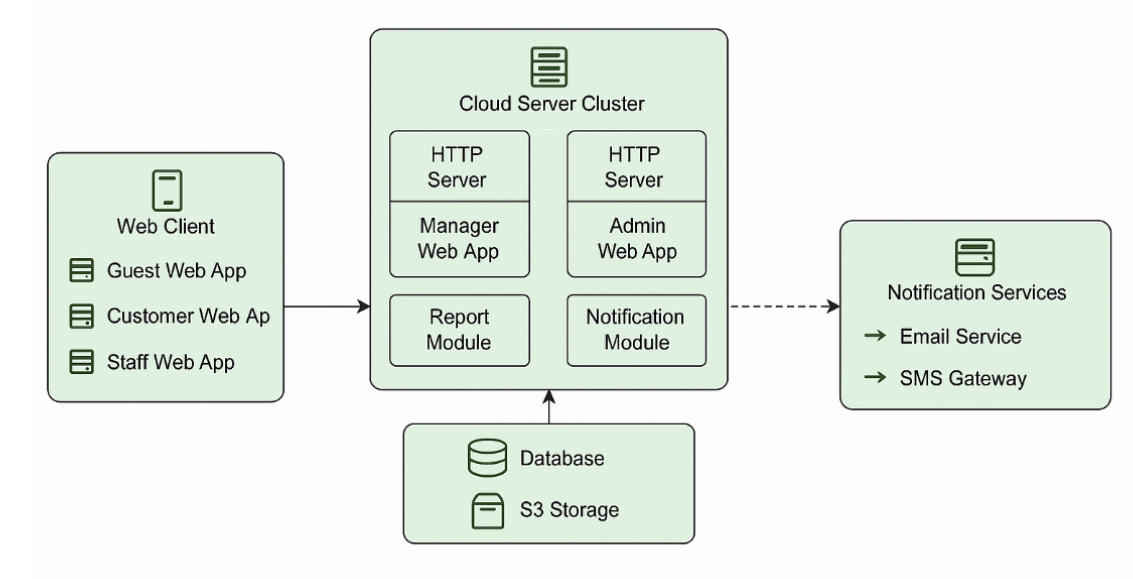








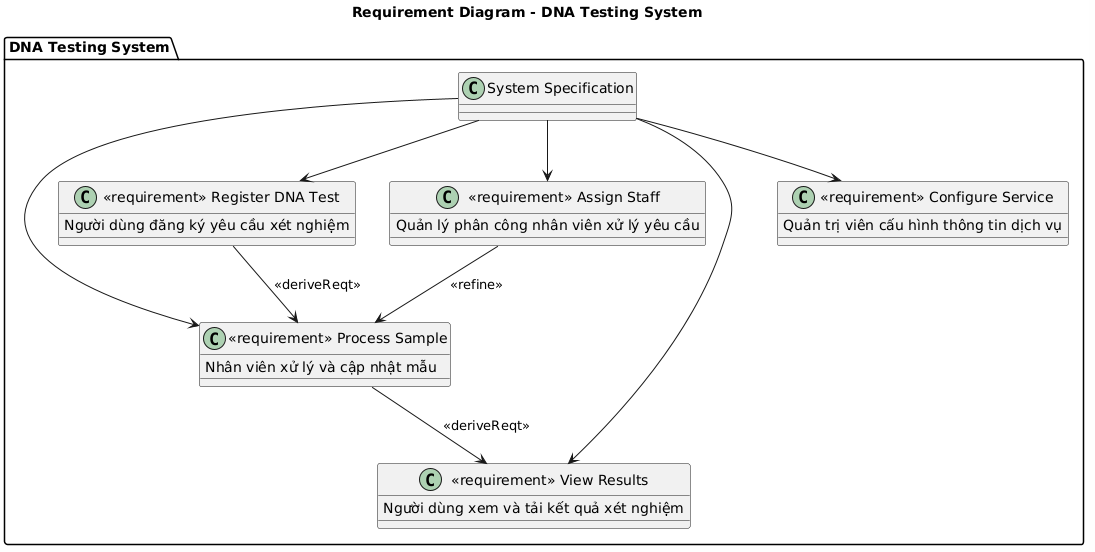
***4. Deployment Diagram***



***Deployment descriptions***

|  |  |  |
| --- | --- | --- |
| **No** | **Component** | **Description** |
| *01* | *Web Client* | *Includes browser-based apps such as Guest Web App, Customer Web App, and Staff Web App. Allows users to view services, send samples, and check results.* |
| *02* | *Cloud Server Cluster* | *Hosts backend services, including two HTTP Servers running internal apps like Manager Web App and Admin Web App. Also includes business modules such as reports and notifications.* |
| *03* | *Report Module* | *Responsible for aggregating data, generating system operation statistics, and producing visual reports for management use.* |
| *04* | *Notification Module* | *Handles internal system notifications to users via email and SMS, integrated with Notification Services.* |
| *05* | *Database & S3 Storage* | *Stores relational data about users and DNA tests; S3 Storage is used to store result files (PDFs, images) for flexible access.* |
| *06* | *Notification Services* | *Includes Email Service and SMS Gateway – allowing the system to send result notifications, OTPs, or sample status updates to end-users.* |

***5. Requirement Diagram***

******

***6. Use Case Diagram***

