

# Institute of Technology, Carlow

## Computing

**Course:** BSc (Hons) Software Development Year 4

**Author:** Eamonn Gaynor

**Student ID:** C00197458

**Tutor:** Mr. Nigel Whyte

**Document:** Design Document

## *Plant Disease Identification Application*

Institiúid Teicneolaíochta Cheatharlach



INSTITUTE *of*  
TECHNOLOGY  

---

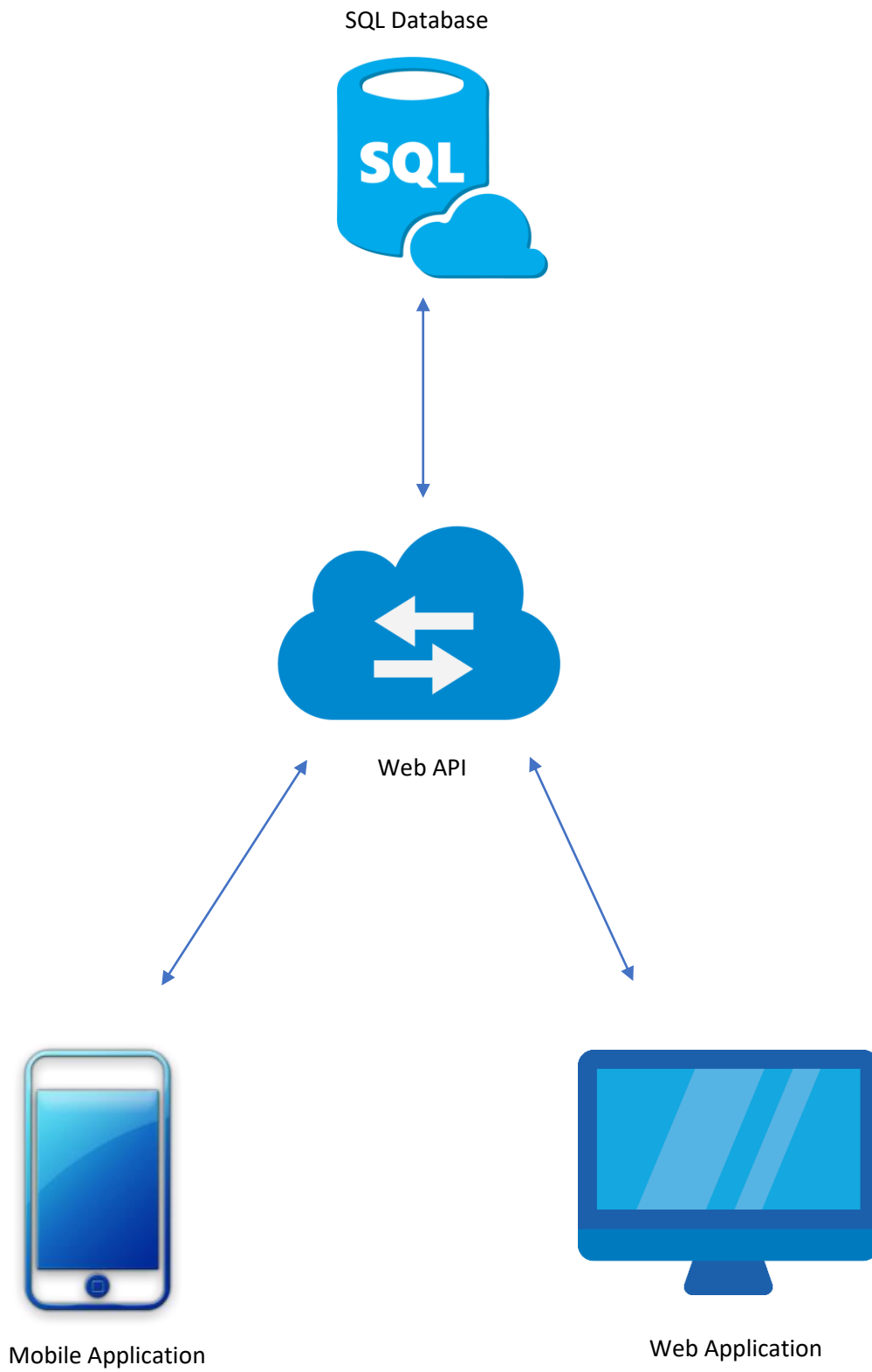
CARLOW

At the heart of South Leinster

## Contents

System Architecture.....	3
Use Cases.....	4
Use Case Diagram.....	4
Detailed Use Cases .....	5
1. Login.....	5
2. Change Password.....	5
3. Logout.....	6
4. Upload Image.....	6
5. Save Result.....	7
6. Add New User.....	7
7. Manage Users.....	8
8. Add New Disease.....	8
9. Manage Diseases.....	9
10. Create Report .....	9
System Sequence Diagrams .....	10
1. Login.....	10
2. Change Password.....	11
3. Log out.....	12
4. Upload Image.....	13
5. Save Result.....	14
6. Add New User.....	15
7. Manage Users.....	16
8. Add New Disease.....	17
9. Manage Disease .....	18
10. Generate Reports.....	19

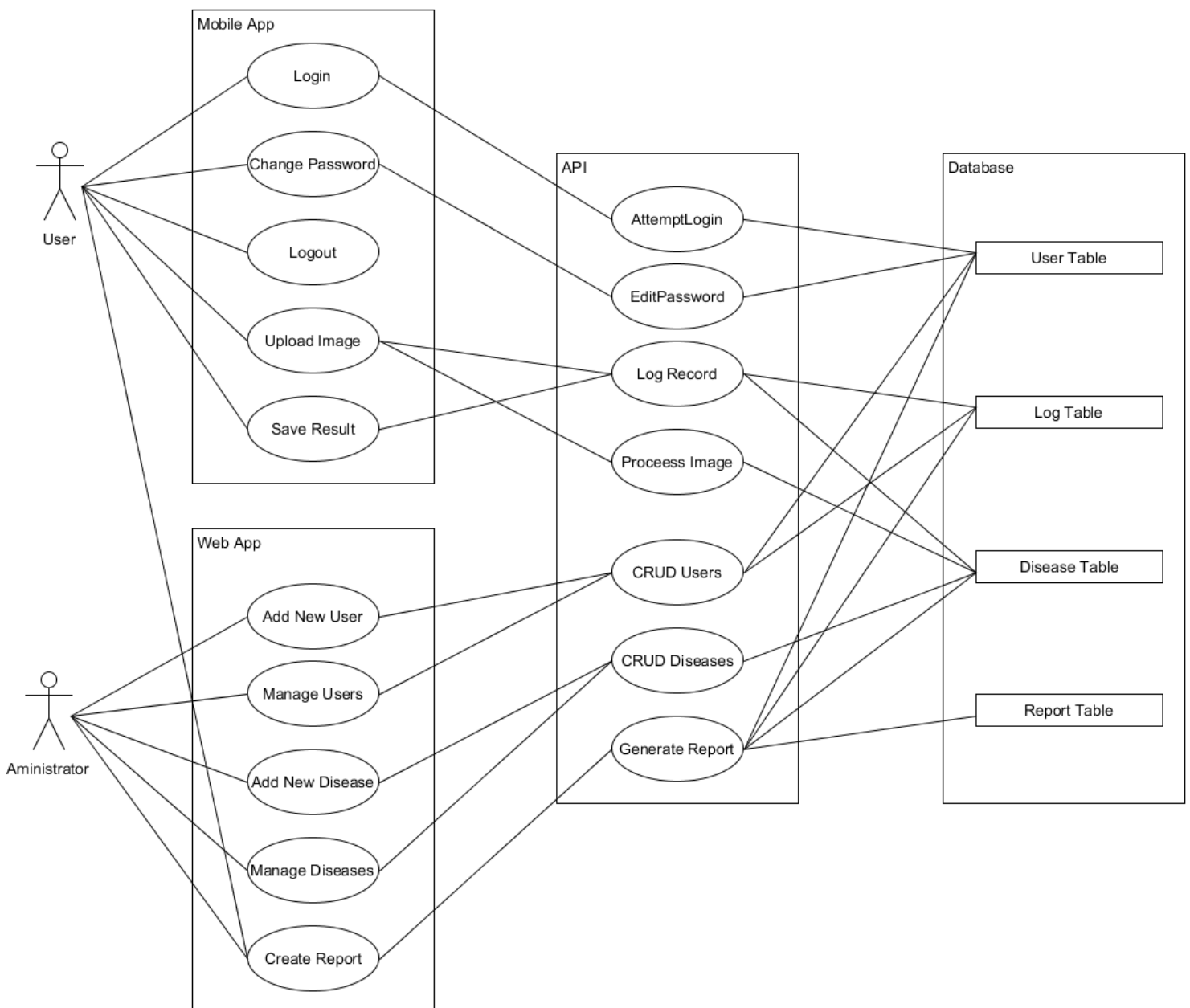
## System Architecture



# Use Cases

## Use Case Diagram

\*Note: Diagram taken from Functional Specification document.



## Detailed Use Cases

### 1. Login

**Use Case:** Login

**Actors:** User, Mobile App, API, Database

**Description:** This use case occurs when a user attempts to login into the system. The user enters a username and password, in which the system will validate to ensure the credentials are correct.

**Main Success Scenario:**

1. User enter their username and password.
2. Existing users enter their username and password.
3. New users are presented with a link to sign up.
4. Username and password fields are checked for minimum requirements.
5. Network connection is checked for availability.
6. Credentials are sent to the web API.
7. Credentials are sent to the database from the API for validation against the User Table records.
8. Result of validation, true or false, is sent back to the API.
9. API sends result back to mobile app.
10. Mobile app outputs message with either successful, or unsuccessful login attempt.

### 2. Change Password

**Use Case:** Change Password

**Actors:** User, Mobile App, API, Database

**Description:** This use case occurs when a user attempts to change their login password. The username must first login with their credentials to access this feature.

**Main Success Scenario:**

1. User selects change password function.
2. User enters current username and password.
3. User enters new password, twice.
4. Validation is processed to ensure current password is correct, and new password is not the same as current password.
5. Network connection is checked for availability.
6. New credentials are sent to the web API.
7. New credentials are sent to the database from the API for validation against the User Table records and to update.
8. User table is updated with the password.
9. Database sends confirmation back to API

10. API sends result back to mobile app.
11. Mobile app outputs confirmation message with either successful, or unsuccessful change password attempt.

### 3. Logout

**Use Case:** Logout

**Actors:** User

**Description:** This use case occurs when a user wishes to log themselves out of the system. A user will simply select the logout option provided.

**Main Success Scenario:**

1. User selects logout option on main menu.
2. The mobile app logs the user out of the system.

### 4. Upload Image

**Use Case:** Upload Image

**Actors:** User, Mobile App, API, Database

**Description:** This use case occurs when a user wishes to upload an image to the system, for plant disease processing. The user must capture an image or select previously stored image in memory. The system will process the image using computer vision techniques, to successfully deliver a diagnosis of the plant back to the user.

**Main Success Scenario:**

1. User selects upload image feature.
2. The camera is accessed and displayed on screen with an option to upload a stored image.
3. User takes a picture from the camera.
4. The image is saved to the device.
5. The image is sent to the API, where it is sent to the database to be stored.
6. The saved image is sent to the API, where computer vision techniques are performed to determine result.
7. Result is sent back to the mobile app, and a description of the disease is presented to the user.

## 5. Save Result

**Use Case:** Save Result

**Actors:** User, Mobile App, API, Database

**Description:** This use case occurs when a user wishes to save the result of the diagnosis which was gathered by the upload image function. Once the user is presented with the diagnosis, they may select the save button which will save this information to database. The user must be logged in to perform this action.

**Main Success Scenario:**

1. User is presented with the diagnosis from uploading the image.
2. The user selects the save result option.
3. The result of the diagnosis is sent to the log record function in the API.
4. The API sends this record to the database, to be stored on Log Table.
5. A confirmation is sent from the database back to the API.
6. The API sends this confirmation back to the Mobile App.
7. The mobile app outputs this confirmation message to the user that the result has been saved.

## 6. Add New User

**Use Case:** Add New User

**Actors:** Administrator, Web App, API, Database

**Description:** This use case occurs when the administrator wishes to create a new user. The administrator accesses this function via the web application.

**Main Success Scenario:**

1. The admin selects the Add New User option from the main menu.
2. The admin enters the credentials of the user, including first name, surname, email and password.
3. The admin selects send.
4. The information entered is validated for minimum requirements.
5. The information is sent to the API where the CRUD operation is performed.
6. The API sends this data to the database, where is compared to the User Table for duplicates.
7. The information is saved in the User Table.
8. The database sends the result of successful or unsuccessful save back to the API.
9. The API sends this result back to the Web App.
10. The Web App outputs the successful or unsuccessful confirmation message to the admin.

## 7. Manage Users

**Use Case:** Manage Users

**Actors:** Administrator, Web App, API, Database

**Description:** This use case occurs when the administrator wishes to modify a user's account details. The administrator performs this action from the web application.

**Main Success Scenario:**

1. The admin selects the Manage User's option from the main menu.
2. The admin selects a user from a drop-down menu.
3. Once a user is selected, the admin can modify all credentials, such as email, names and password.
4. Once modified, the admin selects send.
5. The information entered is validated for minimum requirements.
6. The information is sent to the API where the CRUD operation is performed.
7. The API sends this data to the database, where is compared to the User Table for duplicates.
8. The information is saved in the User Table.
9. The database sends the result of successful or unsuccessful save back to the API.
10. The API sends this result back to the Web App.
11. The Web App outputs the successful or unsuccessful confirmation message to the admin.

## 8. Add New Disease

**Use Case:** Add New Disease

**Actors:** Administrator, Web App, API, Database

**Description:** This use case occurs when the administrator wishes to enter a new disease. The administrator accesses this function via the web application.

**Main Success Scenario:**

1. The admin selects the Add New Disease option from the main menu.
2. The admin enters the information regarding the new disease.
3. The admin selects send.
4. The information entered is validated for minimum requirements.
5. The information is sent to the API where the CRUD operation is performed.
6. The API sends this data to the database, where is compared to the Disease Table for duplicates.
7. The information is saved in the Disease Table.
8. The database sends the result of successful or unsuccessful save back to the API.
9. The API sends this result back to the Web App.
10. The Web App outputs the successful or unsuccessful confirmation message to the admin.



## 9. Manage Diseases

**Use Case:** Manage Diseases

**Actors:** Administrator, Web App, API, Database

**Description:** This use case occurs when the administrator wishes to modify the disease library. The administrator performs this action from the web application.

**Main Success Scenario:**

1. The admin selects the Manage Disease option from the main menu.
2. The admin selects a disease from a drop-down menu.
3. Once a user is selected, the admin can modify all information regarding the disease.
4. Once modified, the admin selects send.
5. The information entered is validated for minimum requirements.
6. The information is sent to the API where the CRUD operation is performed.
7. The API sends this data to the database, where is compared to the Disease Table for duplicates.
8. The information is saved in the Disease Table.
9. The database sends the result of successful or unsuccessful save back to the API.
10. The API sends this result back to the Web App.
11. The Web Application outputs the successful or unsuccessful confirmation message to the admin.

## 10. Create Report

**Use Case:** Create Report

**Actors:** Administrator, Web App, API, Database

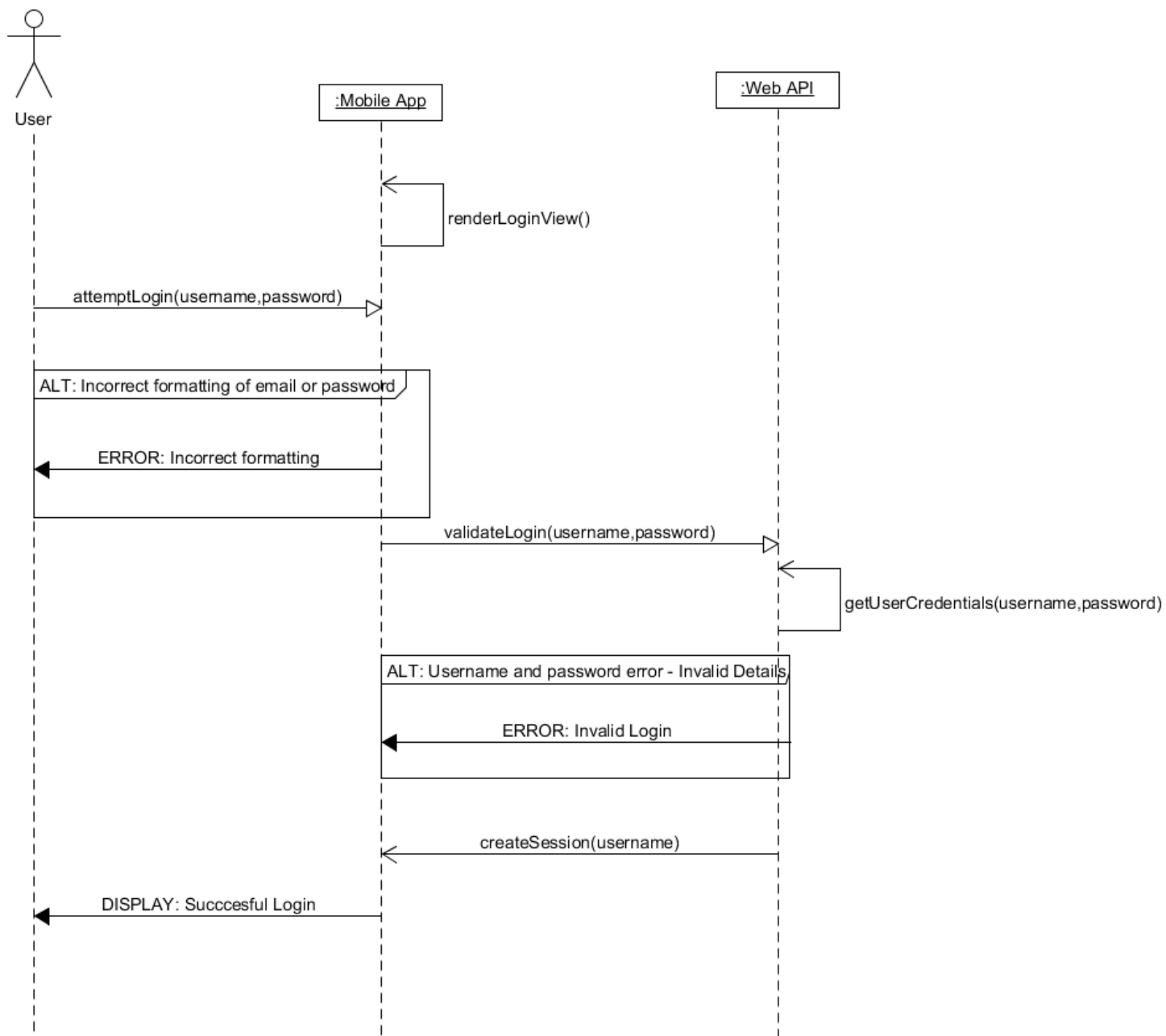
**Description:** This use case occurs when the administrator wishes to create a report from the information stored in the database. The administrator performs this action from the web application.

**Main Success Scenario:**

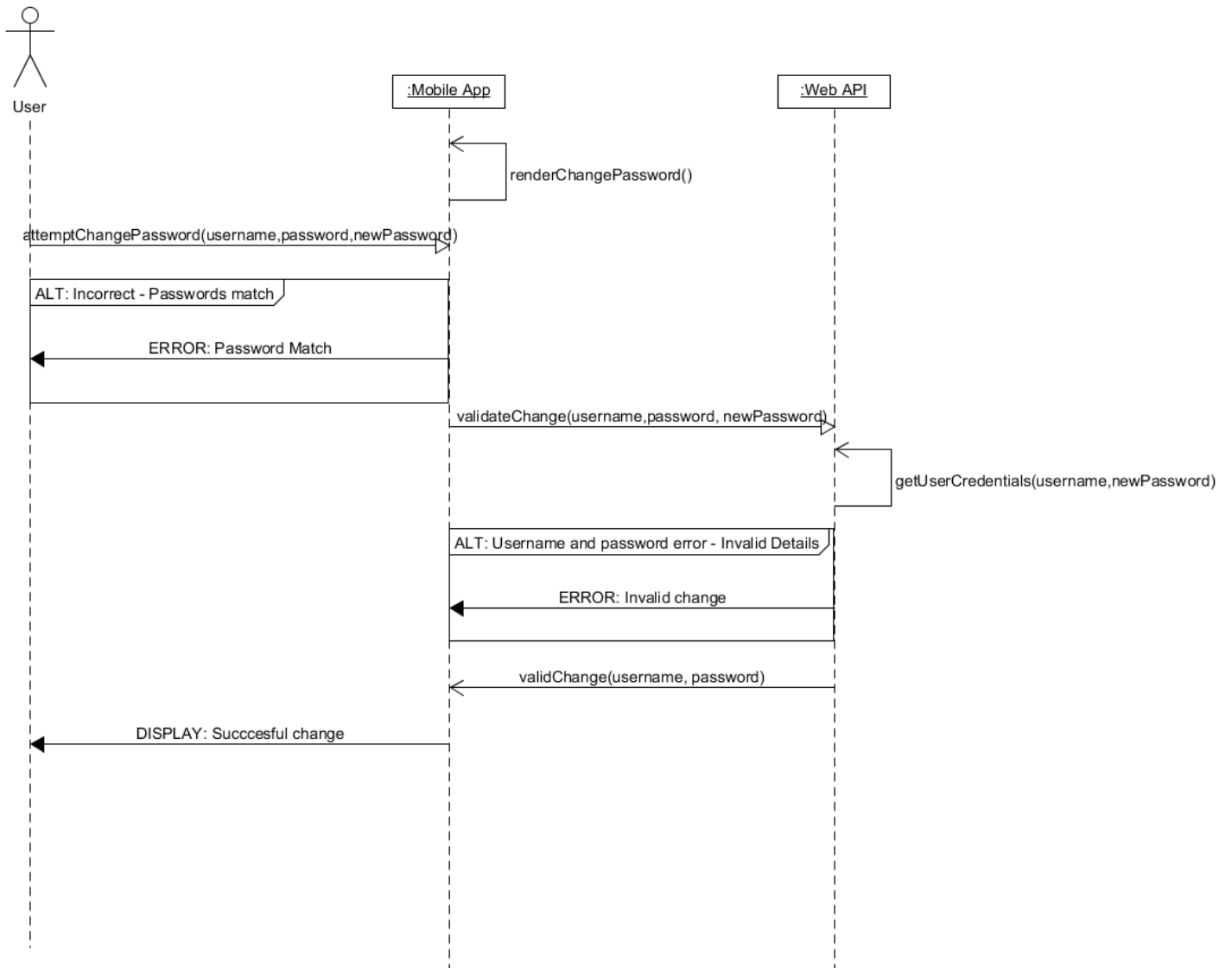
1. The admin selects the Create Report option from the main menu.
2. The admin can select multiple options from multiple drop-downs, including users, diseases, times and dates.
3. The admin selects the appropriate criteria they wish to create a report from.
4. The selected criteria is sent to the API, where a report is generated.
5. The API uses data from the database, including User Table, Log Table and Disease Table.
6. The API collects the required data and transforms it into the information in the form of a report.
7. The API sends this report back to the Mobile App.
8. The Web Application outputs this report to the admin.

# System Sequence Diagrams

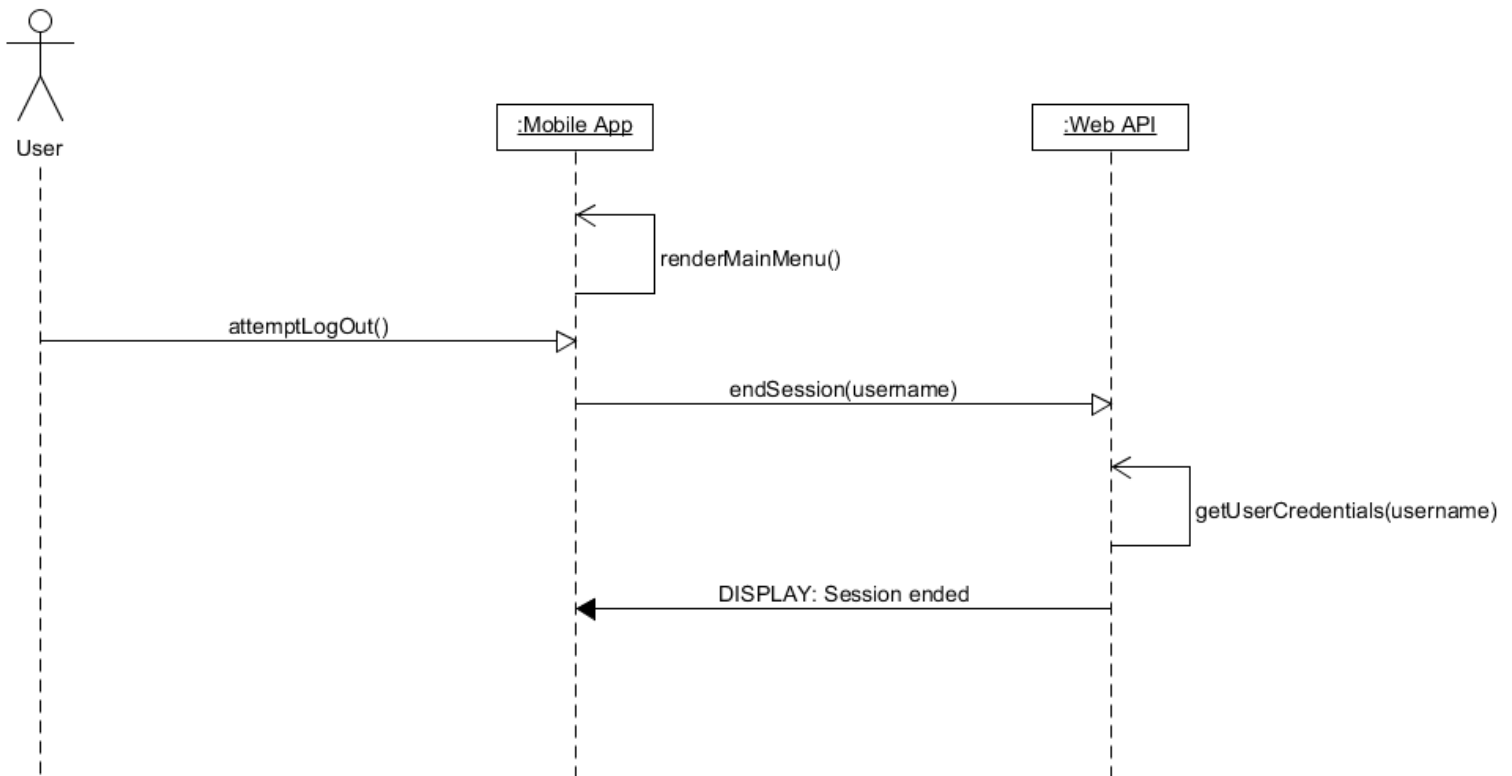
## 1. Login



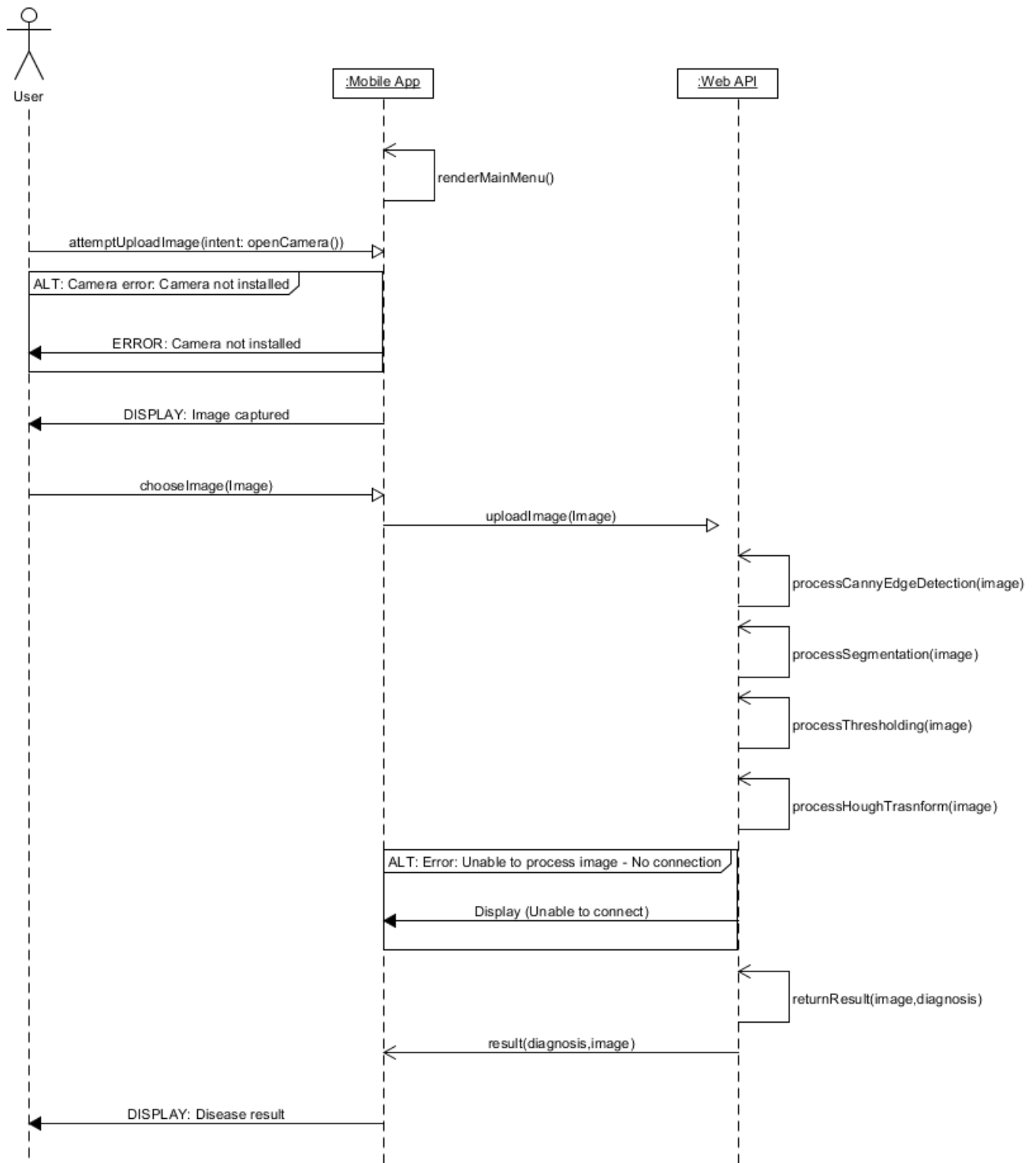
## 2. Change Password



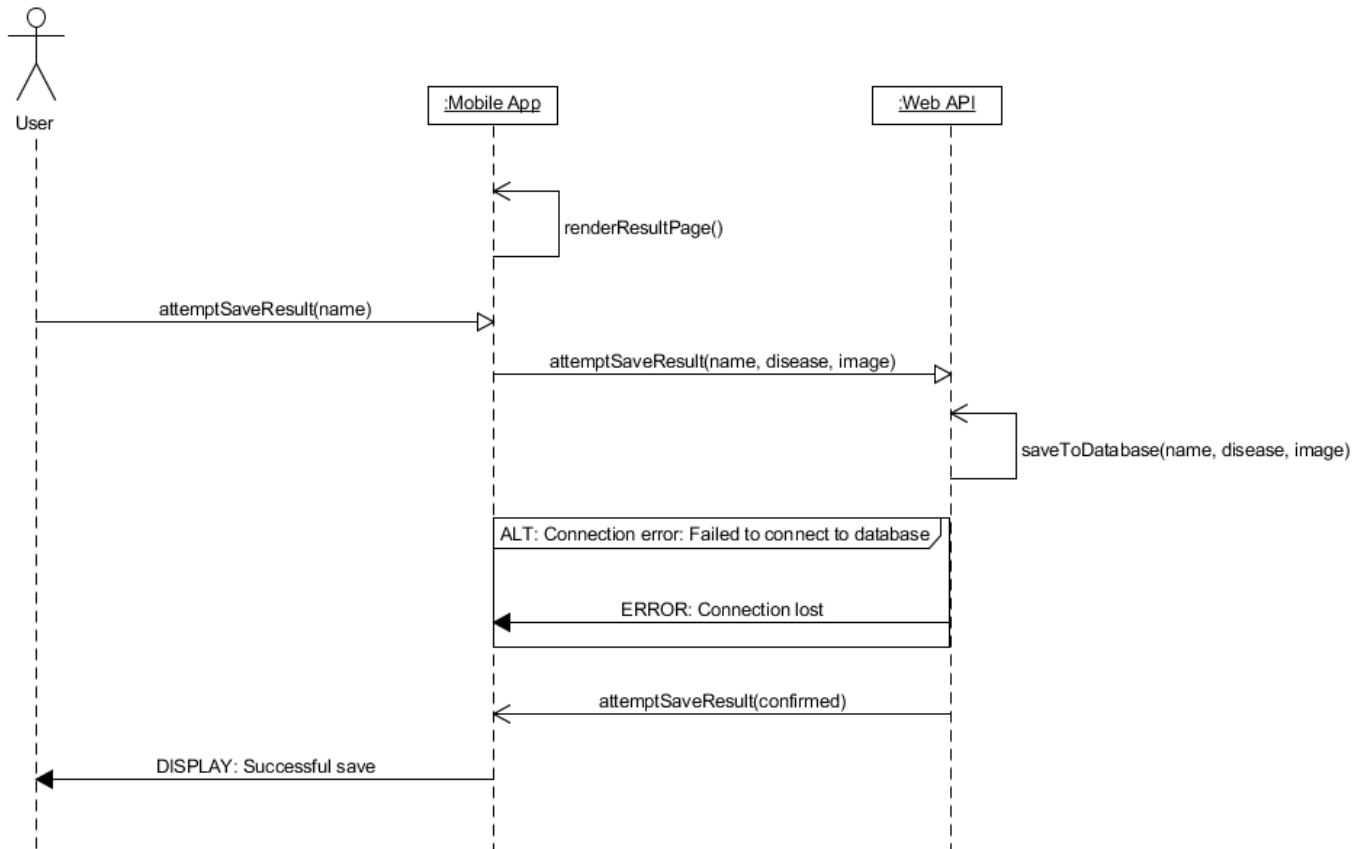
## 3. Log out



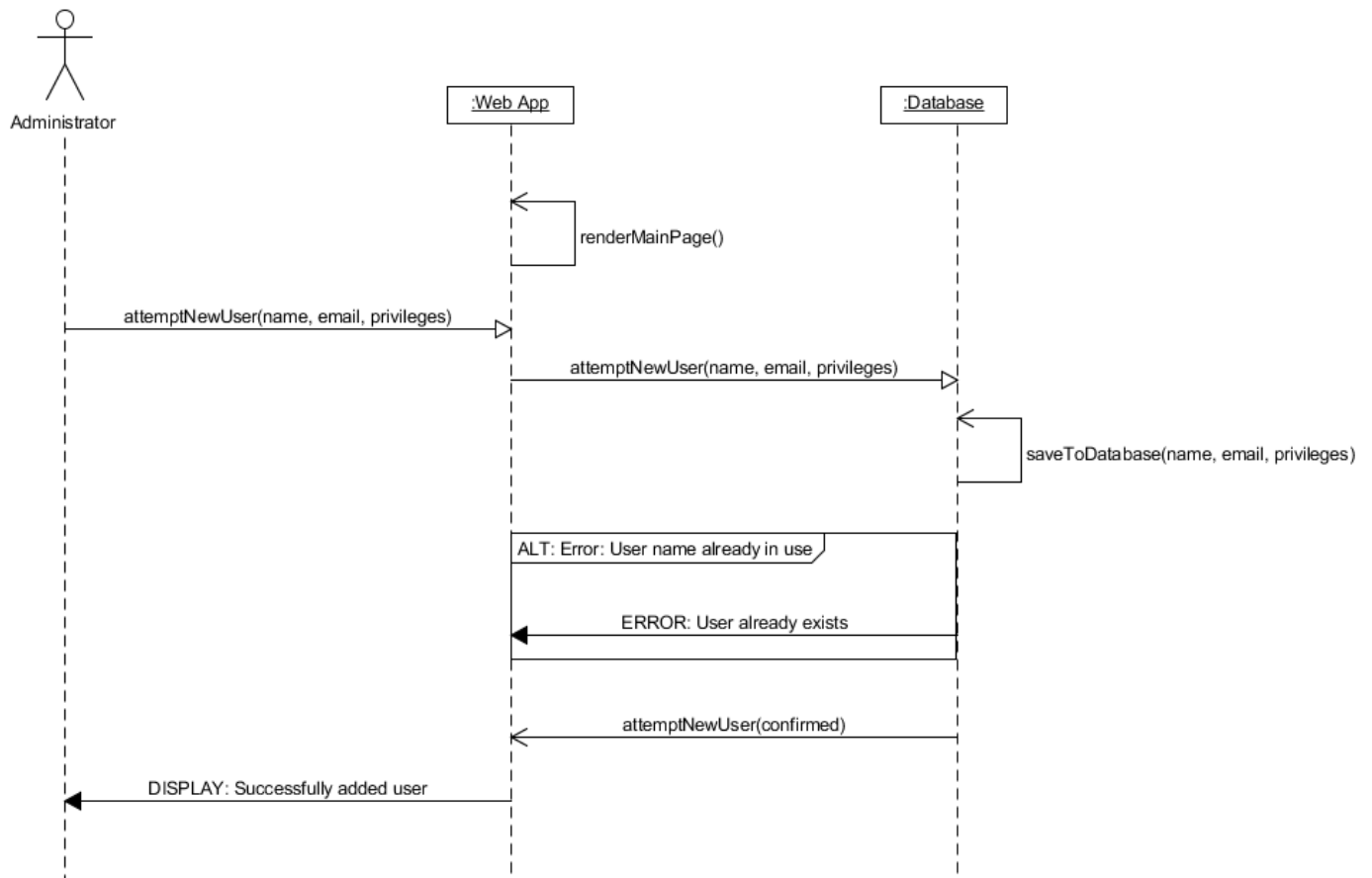
## 4. Upload Image



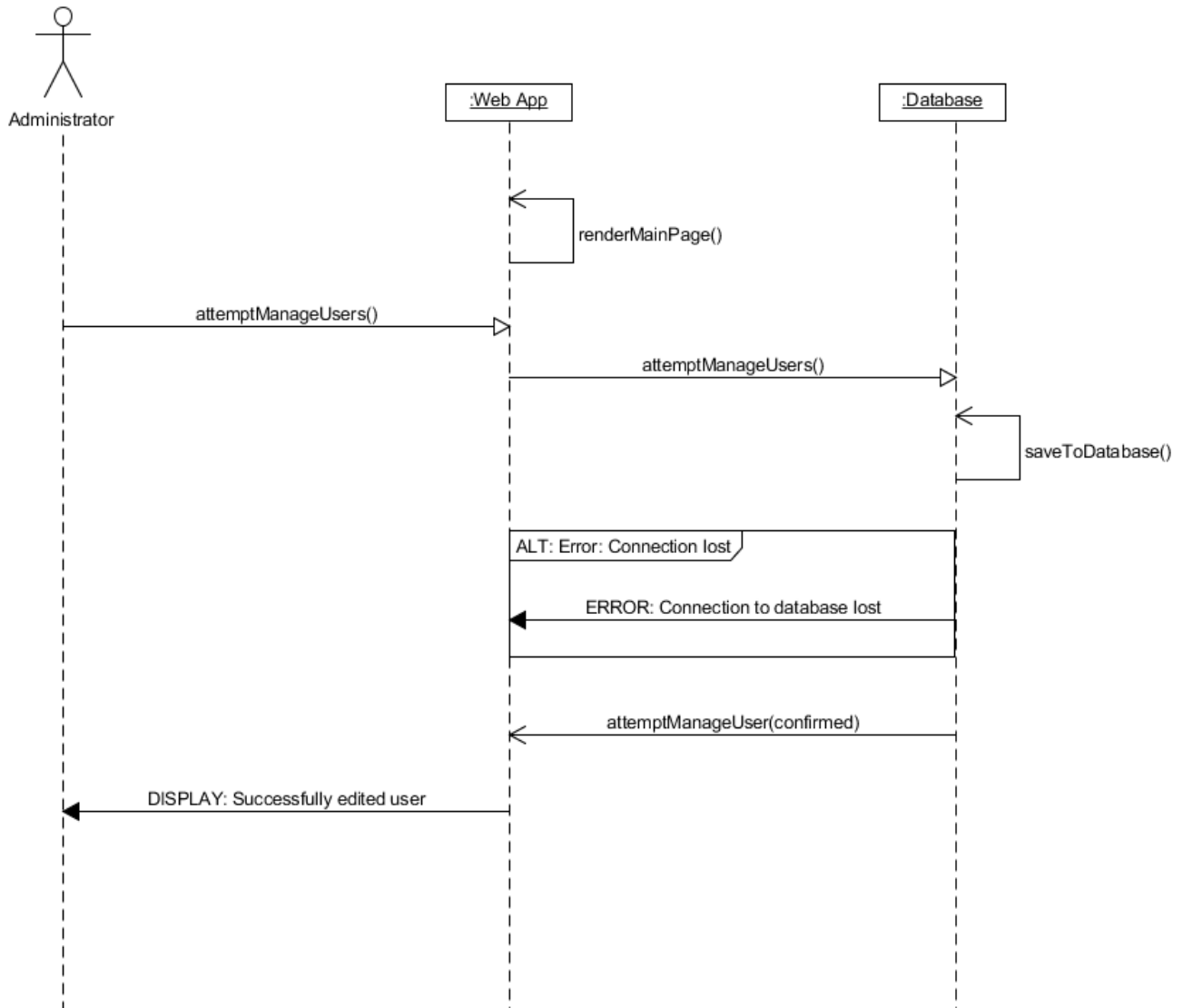
## 5. Save Result



## 6. Add New User

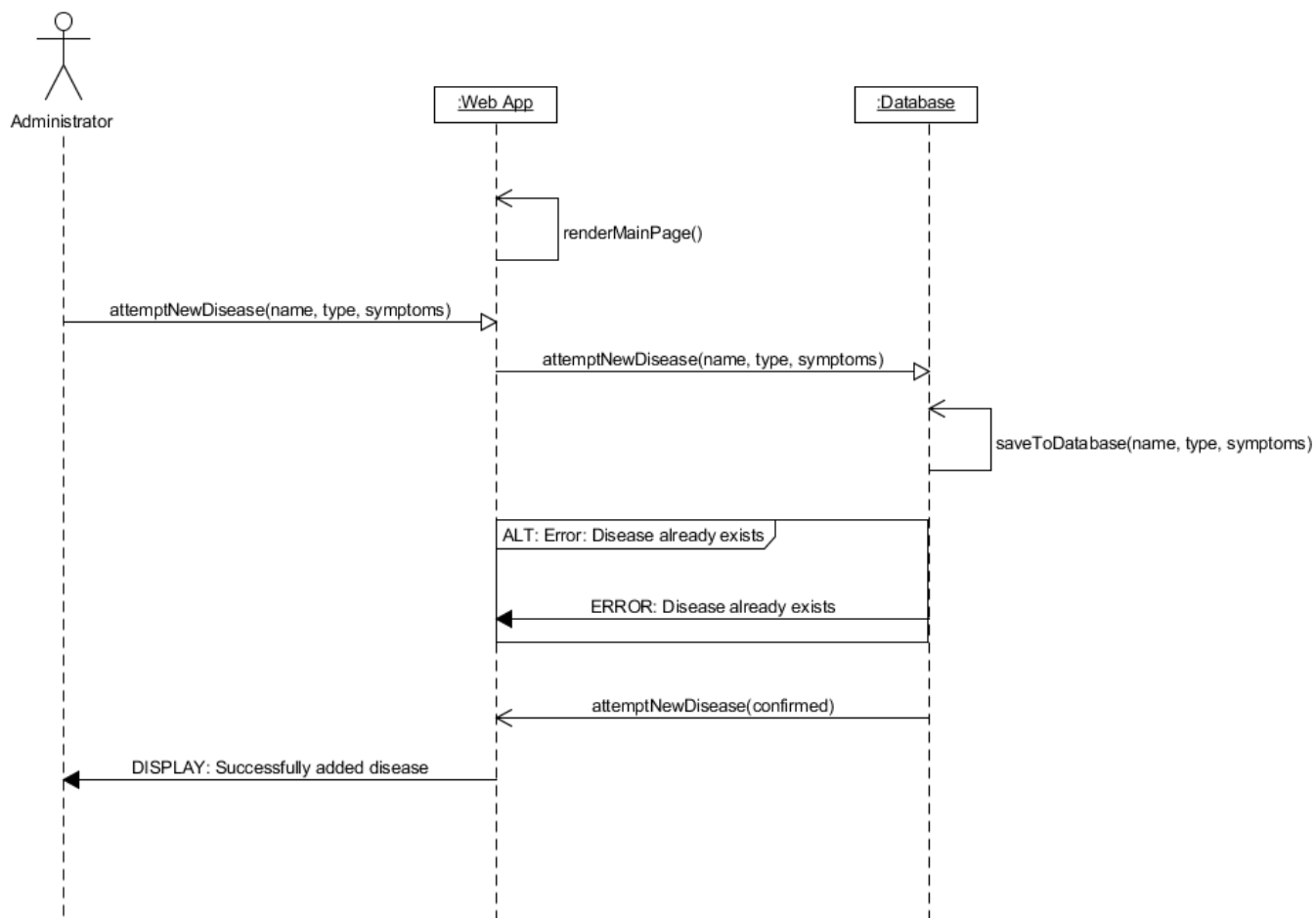


## 7. Manage Users

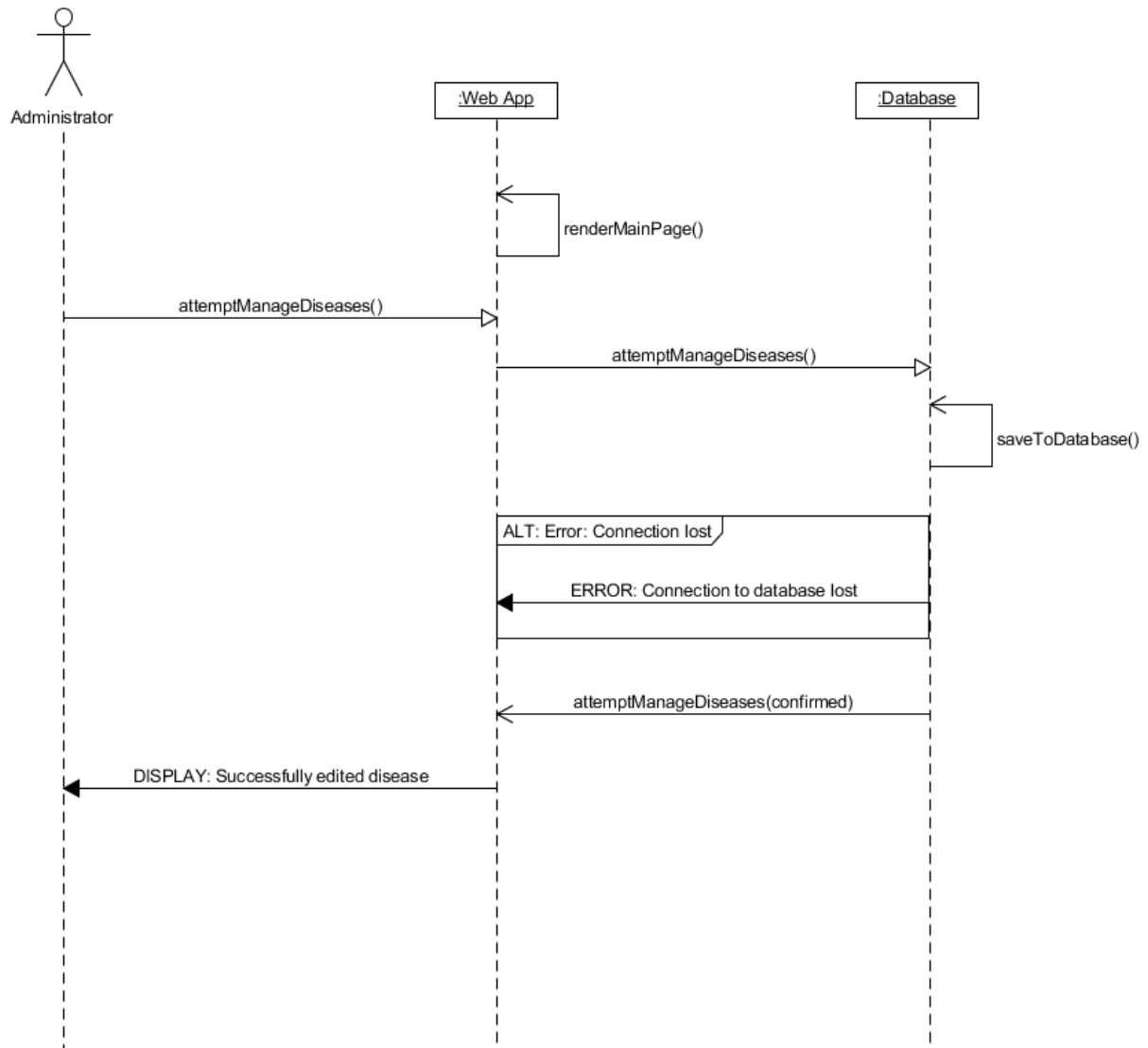




## 8. Add New Disease



## 9. Manage Disease



## 10. Generate Reports

