**COVID-19 Vaccination Record Software**

Test Plan

By

Yanhang Yang 622115513

Jun Zhang 632115505

Haizhao Li 622115504

Zhuoyuan Li 622115515

Suchanun sirijanya 622115040

Department of Software Engineering

College of Arts, Media and Technology

Chiang Mai University

Project Advisor

Jayakrit Hirisajja

**Document History**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **History** | **Status** | **Date** | **Viewable** | **Editable** | **Responsible** |
| Add   * Introduction * Test Plan and Test Procedure | Draft | March 2nd2023 | YYH, JZ, HZL,ZYL,SS JH | YYH, JZ, HZL,ZYL,SS | YYH |
| Add   * Unit test * System test | Draft | March 3th 2023 | YYH, JZ, HZL,ZYL,SS JH | YYH, JZ, HZL,ZYL,SS | YYH, JZ, HZL,ZYL,SS |
| Update   * Unit test * System test | Draft | March 4th 2023 | YYH, JZ, HZL,ZYL,SS JH | YYH, JZ, HZL,ZYL,SS | YYH, JZ, HZL,ZYL,SS |
| Update   * Unit test * System test | Draft | March 5th 2023 | YYH, JZ, HZL,ZYL,SS JH | YYH, JZ, HZL,ZYL,SS | YYH, JZ, HZL,ZYL,SS |
| Add   * Integration test | Draft | March 6th 2023 | YYH, JZ, HZL,ZYL,SS JH | YYH, JZ, HZL,ZYL,SS | YYH, JZ, HZL,ZYL,SS |
| Update   * Integration test | Draft | March 8th 2023 | YYH, JZ, HZL,ZYL,SS JH | YYH, JZ, HZL,ZYL,SS | YYH, JZ, HZL,ZYL,SS |

\* YYH= Yanhang Yang

\* JZ= Jun Zhang

\* HZL= Haizhao Li

\* ZYL= Zhuoyuan Li

\* SS=Suchanun sirijanya

\* JH= Jayakrit Hirisajja

**Contents**

[**Introduction** 4](#_Toc129109958)

[1. Objective 4](#_Toc129109959)

[2. Testing method 4](#_Toc129109960)

[3. Intended Audience 4](#_Toc129109961)

[4. Project Scope 4](#_Toc129109962)

[5. Acronyms 4](#_Toc129109963)

[6. Definition 5](#_Toc129109964)

[**Test Plan and Test Procedure** 6](#_Toc129109965)

[1. Test Objectives 6](#_Toc129109966)

[2. Result of Testing 7](#_Toc129109967)

[3. Test Environment 7](#_Toc129109968)

**Introduction**

1. Objective

This document is the Test Plan. It helps us determine the effort needed to validate the quality of the application under test. The test plan serves as a blueprint to conduct software testing activities as a defined process, minutely monitored and controlled by the test manager.

In addition, it should also have the following objectives:

* To find defects that may get created by the
* To make sure that the result meets the business and user requirements.
* To ensure that it satisfies the SRS that is System Requirement Specifications.

1. Testing method

|  |  |  |
| --- | --- | --- |
| Unit Test | White-box testing | |
| Automatic test (Jest framework) | |
| System Test | Blackbox testing | Function test |
| GUI test |

1. Intended Audience

The mainly intended audience of this document is all significant stakeholders, which include the development team, the project owner, testers, the senior project advisor (Dr. Jayakrit Hirisajja), and anyone evaluating the project.

1. Project Scope

develop a web-based COVID-19 vaccination tracking application include a front-end user interface that displays a list of vaccinated individuals, their names, surnames, and vaccine status. When a user clicks on a name, they can view the individual's personal information, including name, surname, age, and hometown, along with their vaccination details. The back-end system will handle user authentication and access permissions, with admin users having full access to patient data and the ability to add vaccine information, while regular users can only view their own data. Doctors can add comments to their patients' profiles, and the system includes a registration page for new users. Admins can designate users as doctors or regular users, and doctors can view and add comments to their patients' profiles.

1. Acronyms

SRS - Software Requirement Specification

URS - User Requirement Specification

UC - Use Case

UI - User Interface

UTC - Unit Test Case  
STC - System Test Case

IT- Integration test

## Definition

|  |  |
| --- | --- |
| Name | Definition |
| Feature | Transformation of input parameters to output parameters based on a specified algorithm. It describes the functionality of the product in the language of the product. Used for requirements analysis, design, coding, testing or maintenance. [IEEE90] |
| IEEE | Institute for Electrical and Electronics Engineers. Biggest global interest group for engineers of different branches and computer scientists. [IEEE90] |
| Requirement | 1. A condition or capability needed by the user to solve a problem or achieve an objective. 2. A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document. 3. A documented representation of a condition or capability as in definition (1) or (2). [IEEE90] |
| Specification | Precise description of an activity or work product that serves as the basic or input for further activities or work product. A specification can comprise requirements for a product and how they will be solved. Different parts of a specification (e.g., what is to be done, how it will be done) must not be mixed. [IEEE90] |
| Use case | 1. Concept to describe a system based on usage of system resources by its environment. Characterized by an objective-set of interactions within and at the borders of that system. 2. Notation from UML for describing a scenario (Usage approach, operational scenario) from the perspective of this user. [IEEE90] |

**Test Plan and Test Procedure**

1. Test Objectives

The objectives of Vaccination Record Software System are:

1. All bugs or defects are detected

2. Those bugs or defects are fixed.

3. Functionalities and user interface covered the requirements.

4. All functionalities and features must be there.

**Scope of Testing**

Vaccination Record Software System will be tested the unit testing and system testing and record the test results in the test record.

**Test Duration**

|  |  |
| --- | --- |
| Progress | Date and duration |
| **Term project Progress Report** | **Perform date:** March 1 – 10 2023  **Duration:** 10days |

**Test Responsibility**

|  |  |
| --- | --- |
| Item | Responsibility |
| Test unit test of web application | YYH |
| Record unit test of web application | YYH |
| Test system test of web application | ZYL,SS |
| Record system test of web application | ZYL,SS |
| Test Integration test of web application | JZ,HZL |
| Record Integration test of web application | JZ,HZL |

**Test Strategy**

COVID-19 Vaccination Record Software System test will be following by:

1. Design test case for each feature.

2. Prepare test data for each feature.

3. Determine expected result.

4 . Perform testing on individual features.

5. Result of testing will be recorded.

1. Result of Testing

In the test record, the test result will separate into two parts, which are:

1. Actual output: The actual outputs that are performed by each test case.

2. Pass/Fail criteria:

* Pass: the result of actual result is the same as expected result
* Fail: the result of actual result is not the same as expected result.

1. Test Environment

**Laptops**

**MacBook Pro (M1, 2022)**

Processor: Apple M1 chip with 8-core CPU and 8-core GPU

Memory: 8GB

Graphics: 8-core GPU

Operating System: macOS Big Sur

**ROG Q533Q**

Processor: Intel Core i7-1165G7 Processor, 2.8 GHz

Memory: 32 GB DDR4 3200MHz SDRAM

Graphics: NVIDIA GeForce RTX2070 , with 8GB GDDR6 VRAM

Operating System: Windows 10 Home

**Internet**

# Unit test

Unit Test Case 001 (UTC-001):

Method name: getVaccine(@PathVariable("id") Long id)

Description: This unit test is for the getVaccine method in the VaccineController class. It tests whether the method returns the expected result when given a valid or invalid ID.

Data for testing​:

|  |  |
| --- | --- |
| ID | Valid or Invalid |
| 1 | Valid |
| 100 | Invalid |

Test Case：

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expected Result |
| 1. | Enter vaccine with ID to test whether the ID can return the correct vaccine | 1 | The method should return a ResponseEntity with the VaccineDTO for the ID 1 |
| 2. | Enter vaccine with ID to test whether the ID can return the correct vaccine | 100 | The method should return a "The given id is not found" |

Unit Test Case 002 (UTC-002):

Method name: getDoctor(@PathVariable("id") Long id)

Description: This code is a method for getting a doctor by their ID in the system. This unit test is to ensure that the method returns the expected result when given a valid or invalid ID.

Data for testing​:

|  |  |
| --- | --- |
| ID | Valid or Invalid |
| 1 | Valid |
| 32 | Invalid |

Test Case：

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expected Result |
| 1. | Enter doctor with ID to test whether the ID can return the correct doctor | 1 | The method should return a ResponseEntity with the DoctorDTO for the ID 1 |
| 2. | Enter doctor with ID to test whether the ID can return the correct doctor | 32 | The method should return a "The given id is not found" |

Unit Test Case 003 (UTC-003):

Method name: getPatient(@PathVariable("id") Long id)

Description: This code is a method for getting a patient by their ID in the system. This unit test is to ensure that the method returns the expected result when given a valid or invalid ID.

Data for testing​:

|  |  |
| --- | --- |
| ID | Valid or Invalid |
| 1 | Valid |
| 88 | Invalid |

Test Case：

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expected Result |
| 1. | Enter patient with ID to test whether the ID can return the correct patient | 1 | The method should return a ResponseEntity with the PatientDTO for the ID 1 |
| 2. | Enter patient with ID to test whether the ID can return the correct patient | 88 | The method should return a "The given id is not found" |

Unit Test Case 004 (UTC-004):

Method name: Authority.builder().name( ).build()

Description: It is used to detect different login user identities, thus giving users different permissions

Data for testing​:

|  |  |
| --- | --- |
| Email | User identity |
| admin@admin.com | admin |
| enabled@user.com | user |
| doctor@doctor.com | doctor |

Test Case：

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expected Result |
| 1. | After entering the email, the system will detect and assign different user permissions | .email("admin@admin.com") | .name(AuthorityName.ROLE\_ADMIN) |
| 2. | After entering the email, the system will detect and assign different user permissions | .email("enabled@user.com ") | .name(AuthorityName.ROLE\_DOCTOR) |
| 3 | After entering the email, the system will detect and assign different user permissions | .email("doctor@doctor.com ") | .name(AuthorityName.ROLE\_USER) |

Unit Test Case 005 (UTC-005):

Method name: addCommentPatient(@RequestBody Patients patient)

Description: Test whether the function that only doctors can leave messages runs successfully

Data for testing​:

|  |  |
| --- | --- |
| Email | User identity |
| admin@admin.com | admin |
| enabled@user.com | user |
| doctor@doctor.com | doctor |

Test Case：

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expected Result |
| 1. | Use the login account to enter the website respectively, and then check whether the patient can leave a message | .email("admin@admin.com") | null |
| 2. | Use the login account to enter the website respectively, and then check whether the patient can leave a message | .email("enabled@user.com ") | null |
| 3 | Use the login account to enter the website respectively, and then check whether the patient can leave a message s | .email("doctor@doctor.com ") | setDoctor\_comm(patient.getDoctor\_comm()); |

Unit Test Case 006 (UTC-006):

Method name: createAuthenticationToken() throws AuthenticationException

Description: This code is a method for creating an authentication token for a user in the system. It takes in a username and password as parameters and generates a token using JWT (JSON Web Token) for user authentication.

Data for testing​:

|  |  |  |  |
| --- | --- | --- | --- |
| ID | username | password | Valid or Invalid |
| 1 | testuser | testpassword | Valid |
| 2 | invaliduser | testpassword | Invalid |
| 3 | testuser | invalidpassword | Invalid |

Test Case：

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expected Result |
| 1. | Should generate an authentication token for a valid user | JwtAuthenticationRequest(username="testuser", password="testpassword"), Device | The method should return a ResponseEntity with a token and user information |
| 2. | Should throw AuthenticationException for an invalid user | JwtAuthenticationRequest(username="invaliduser", password="testpassword"), Device | The method should throw an AuthenticationException |
| 3 | Should throw AuthenticationException for an invalid password | JwtAuthenticationRequest(username="testuser", password="invalidpassword"), Device | The method should throw an AuthenticationException |
| 4 | Should throw IllegalArgumentException with null authenticationRequest | JwtAuthenticationRequest(username=null, password=null), Device | The method should throw an IllegalArgumentException |

Unit Test Case 006 (UTC-006):

Method name:getPatients()

Description: This method retrieves a list of all patients in the system if the user is an admin user, and retrieves only the data of the currently logged in user if the user is a normal user.

Data for testing​:

|  |  |
| --- | --- |
| Email | User identity |
| admin@admin.com | admin |
| enabled@user.com | user |

Test Case：

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expected Result |
| 1. | Should return a list of all patients when called by an admin user | .name(AuthorityName.ROLE\_ADMIN) | The method should return a ResponseEntity with a list of all PatientDTOs |
| 2. | Should return only the data of the currently logged in user | .name(AuthorityName.ROLE\_USER) | The method should return a ResponseEntity with the PatientDTO for the currently logged in user |

Unit Test Case 007 (UTC-007):

Method name: addVaccine()

Description: This method adds vaccine data to a user's profile in the system, but only if the user making the request has admin privileges.

Data for testing​:

|  |  |
| --- | --- |
| Email | User identity |
| admin@admin.com | admin |
| enabled@user.com | user |

Test Case：

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expected Result |
| 1. | Should add vaccine data to the user's profile when called by an admin user | .name(AuthorityName.ROLE\_ADMIN) | The method should return a ResponseEntity with a success message |
| 2. | Should throw ResponseStatusException when called by a normal user | .name(AuthorityName.ROLE\_USER) | The method should throw a ResponseStatusException And prompt that “you do not have permission to operate” |

Unit Test Case 007 (UTC-007):

Method name: register(username:String,password:String):String

Description: Test whether this method can be used to detect whether user account is registered.

Data for testing​:

|  |  |
| --- | --- |
| Username | Account registration status |
| enabled@user.com | Not registered |
| Enabled2@user.com | registered |

Test Case：

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expected Result |
| 1. | The user enters an unregistered user name, enters a password for the user name, and the format of the user name meets the requirements of the software. Then click the Registration button | “Username”:” enabled@user.com”, “password”:”123456” | "message":" Account created successfully ! " |
| 2. | The user enters an user name have already registered, enters a password for the user name. Then click the Registration button | “Username”:”Enabled2@user.com”,“password”:”123456” | "message":" Registration failed. The Email may have been registered" |

Unit Test Case 008 (UTC-008):

Method name: addDoctor(),addPat()

Description: Test whether the administrator correctly uses the permissions of the user group when assigning the registered account to different user permissions

Data for testing​:

|  |  |
| --- | --- |
| username | User identity assigned |
| 1 | doctor |
| 2 | user |

Test Case：

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expected Result |
| 1. | When a user is assigned to the doctor user group, check whether they have all the permissions of the doctor user group | “Username”:”1”, “User identity assigned”:” doctor” | User group permissions:  doctor |
| 2. | When a user is assigned to the user group, check whether they have all the permissions of the user group | “Username”:”2”  “User identity assigned”: “user” | User group permissions:  user |