# 1.0 Introduction

## 1.1 Purpose

* To outline the approach, scope, objectives, and resources required for testing a software system or application.
* To serve as a guide for the testing team, ensuring that the testing process is organised, systematic, and comprehensive.
* To identify potential risks and issues that may arise during testing and provide a framework for managing these risks.
* To improve the overall quality of the software and increase confidence in its reliability and performance.
* To provide a reference document for future testing efforts, which can be updated as needed to reflect changes in testing requirements or project scope.

## 1.2 Scope

The test plan focuses solely on black box functional testing for features F001 to F013 in the MyCovidView system. It does not include unit testing, integration testing, user acceptance testing, regression testing, confirmation testing, or other forms of testing. The scope of testing is limited to system-level testing, and its purpose is to verify that the system meets its requirements and is suitable for use. By concentrating on system-level testing, the test plan aims to ensure that the system meets its functional requirements and is fit for purpose.

## 1.3 References

The test plan is based on the following documents:

* Fantasy\_UML Use Case Diagram.pdf
* Link: [Fantasy\_UML Use Case Diagram.pdf](https://drive.google.com/file/d/1a7KVPc5SOeFN8XS3uQDbTG4RSZpzrTsJ/view?usp=share_link)
* Fantasy&LastMinute\_SRS.pdf
* Link: [Fantasy&LastMinute\_SRS.pdf](https://drive.google.com/file/d/12pMOOhSH5E1NrB2bTWifNTF1nWj397Hk/view?usp=drive_link)

# 2.0 Details of Test Plan

## 2.1 Test Items

The operation will be determined based on the information provided in documents as below:

* Fantasy\_UML Use Case Diagram.pdf
* Link: [Fantasy\_UML Use Case Diagram.pdf](https://drive.google.com/file/d/1a7KVPc5SOeFN8XS3uQDbTG4RSZpzrTsJ/view?usp=share_link)
* Fantasy&LastMinute\_SRS.pdf
* Link: [Fantasy&LastMinute\_SRS.pdf](https://drive.google.com/file/d/12pMOOhSH5E1NrB2bTWifNTF1nWj397Hk/view?usp=drive_link)

## 2.2 Test Traceability Matrix

The functions in MyCovidView and its traceability are shown as the table below:

Table 2.2: Test Traceability Matrix

| Function ID | Function Description | Risk Level | Source of Function |
| --- | --- | --- | --- |
| F001 | Import data | Medium | UC-01 import data |
| F002 | Export data | Medium | UC-02 export data |
| F003 | Clean and process data | High | UC-03 clean and process data |
| F004 | View visualized data in web app | Low | UC-04 view visualized data in web app |
| F005 | Compare visualized data | Low | UC-05 compare visualized data |
| F006 | Delete data to be visualized | Medium | UC-06 delete data to be visualized |
| F007 | Update data to be visualized | Medium | UC-07 update data to visualized |
| F008 | Zoom in and out data visualization | Low | UC-08 zoom in and out data visualization |
| F009 | Filter and sort data | Low | UC-09 filter and sort data |
| F010 | Change visualization technique | Low | UC-10 change visualization technique |
| F011 | Change data to be visualized | Medium | UC-11 change data to be visualized |
| F012 | Change data visualization colour | Low | UC-12 change data visualization colour |
| F013 | View visualized data in iOS and Android apps | Low | UC-13 view visualized data in iOS and Android apps |

## 

## 2.3 Features to be Tested

The following table contains the features to be tested based on Fantasy\_UML Use Case Diagram.pdf. Listed together are the function ID and its corresponding functions and estimated risk level.

For Risk Level **High** : Conduct **3** testings

For Risk Level **Medium** : Conduct **2** testings

For Risk Level **Low** : Conduct **1** testings

| **Function ID** | **Function Description** | **Risk Level** |
| --- | --- | --- |
| F001 | Import data | Medium |
| F002 | Export data | Medium |
| F003 | Clean and process data | High |
| F004 | View visualized data in web app | Low |
| F005 | Compare visualized data | Low |
| F006 | Delete data to be visualized | Medium |
| F007 | Update data to be visualized | Medium |
| F008 | Zoom in and out data visualization | Low |
| F009 | Filter and sort data | Low |
| F010 | Change visualization technique | Low |
| F011 | Change data to be visualized | Medium |
| F012 | Change data visualization colour | Low |
| F013 | View visualized data in iOS and Android apps | Low |

**Table 2.3:** Features To Be Tested

## 2.4 Features Not to be Tested

Other aspects of the system are not covered in this test plan. This includes, but is not limited to:

1. Operation Procedure

This test plan is for static, integration and system level test, operation procedure is not covered here.

1. Network Security

This test plan is designed for functional tests, security is not covered here.

1. Performance

The test plan is for functional testing. Hence, performance testing is not covered.

## 2.5 Approach

The test on MyCovidView is a system level functional test that focuses only on the functional part of the system. Hence, the testing will be using a risk-based strategy. This test will also be a black-box test where test cases are derived using black box testing techniques such as:

1. State transition testing
2. Use case testing
3. GUI testing

## 2.6 Item Pass/Fail Criteria

The system must satisfy the following criteria in order to pass:

1. All test cases must be passed
2. No deadlock during the operation

## 2.7 Suspension Criteria and Resumption Requirements

Suspension Criteria:

In cases where a Javascript-enabled environment is not available, all tests will be suspended.

Resumption Requirement:

A Javascript-enabled environment is prepared.

## 2.8 Test Deliverables

The following documents will be generated by the system test group and will be delivered during the system testing.

Test documentation:

1. Test Plan
2. Test Template
3. Test Completion Report

## 2.9 Entry Criteria

The following items are needed before the testing can begin:

* Requirements documents for MyCovidView
* Software Under Test: MyCovidView
* Source code for MyCovidView

## 2.10 Exit Criteria

The following items are needed before the testing can end:

* Test execution has been completed.
* All Test Deliverables in Section 2.8 Test Deliverables have been delivered to the client.
* Zero major defects.
* A maximum of three minor defects.

# 3.0 Test Management

## 3.1 Planned Activities

**Table 3.1:** Planned Activities

| No. | Task | Activity | Target Date |
| --- | --- | --- | --- |
| 1 | Test Planning and control | 1. Define test project scope and objectives 2. Requirement view 3. Identified risks, staffing and scheduling 4. Documenting Test Plan 5. Technical review 6. Submission and approval of Test Plan 7. Develop Test Template 8. Submission and approval of Test Template | Week 11 |
| 2 | Test Implementation and Execution | 1. Perform functional test 2. Record test result 3. Analyse test result | Week 12 |
| 3 | Documenting Test Closure | 1. Collecting test process data from completed activities to consolidate experience, testwares, facts and numbers. 2. Compile Test Completion Report | Week 13 |
| 4 | Test Complete | 1. Delivery of product to client. | Week 14 |

## 3.2 Environment and Infrastructure

This section describes the hardware, software and operating system that are required for this system testing.

### 3.2.1 Hardware Requirements

#### 3.2.1.1 Personal Computers

**Table 3.2.1.1:** PC Hardware Requirements

| Processor | Intel Core i5-4430 |
| --- | --- |
| Memory | 4GB |
| Storage | 128GB |

#### 

#### 3.2.1.2 Mobile Devices

**Table 3.2.1.2:** Mobile Device Hardware Requirements

| Processor | ARMv8-A architecture based CPU or Apple A11 Bionic, or greater |
| --- | --- |
| Memory | 2 GB |
| Storage | 32 GB |

### 3.2.2 Software Requirements

#### 3.2.2.1 Personal Computers

**Table 3.2.2.1:** PC Software Requirements

| Operating System | Windows 8.1 or greater, Mac OSX |
| --- | --- |
| Browser | Any modern browser that supports HTML5 and JavaScript. (e.g., Chrome, Edge, Safari, Firefox) |

#### 3.2.2.2 Mobile Devices

**Table 3.2.2.2:** Mobile Device Software Requirements

| Operating System | Android 8.1 or greater, iOS 10 or greater |
| --- | --- |
| Type of Files(s) | .apk (Android), .ipa (iOS) |

### 3.2.3 Publications

The following document(s) are required to support system testing:

* Fantasy\_UML Use Case Diagram.pdf
* Link: [Fantasy\_UML Use Case Diagram.pdf](https://drive.google.com/file/d/1a7KVPc5SOeFN8XS3uQDbTG4RSZpzrTsJ/view?usp=share_link)
* Fantasy&LastMinute\_SRS.pdf
* Link: [Fantasy&LastMinute\_SRS.pdf](https://drive.google.com/file/d/12pMOOhSH5E1NrB2bTWifNTF1nWj397Hk/view?usp=drive_link)
* Test Template
* Link: [Test Template](https://docs.google.com/document/d/1BAgGtFGT9DibcmSYMr_bL0uJsp52-Dr6pNTB6RNJuaE/edit?usp=drive_link)

## 3.3 Responsibilities and Authority

This section only covers Responsibilities and Authority of Group Impact members which takes Software Tester as Role as a Team.

**Table 3.3:** Responsibilities and Authority of Group Impact Testing Procedures

| **Group Member** | **Responsibilities** |
| --- | --- |
| Ho Yi Liang (Leader) | * Audit and update Test Plan * Audit and update Test Template * Audit testing for own group’s visualisation development * Audit test results from other groups * Compile Test Completion Report |
| Elgene Ee Ding Ren | * Develop Test Plan * Develop Test Template * Perform testing for own group’s visualisation development * Audit test results from other groups * Compile Test Completion Report |
| Tan Wei Lok | * Develop Test Plan * Develop Test Template * Perform testing for own group’s visualisation development * Audit test results from other groups * Compile Test Completion Report |
| Goh Chee Lam | * Develop Test Plan * Develop Test Template * Perform testing for own group’s visualisation development * Audit test results from other groups * Compile Test Completion Report |
| Chai Yu Xuan | * Develop Test Plan * Develop Test Template * Perform testing for own group’s visualisation development * Audit test results from other groups * Compile Test Completion Report |

## 3.4 Resources and Allocation

### 3.4.1 Internal Resources

The internal quality assurance team should be responsible for reviewing the test deliverables for adherence to compliance.

### 3.4.2 External Resources

There is no external resource required in system testing.

## 3.5 Training

There is no additional training required in system testing.

# 4.0 General

## 4.1 Metrics

This section will be showing all the metrics collected to be used as monitoring of progress.

**Table 4.1:** Metrics for monitoring progress

| **Metric List** | **Metric Description** | **Goals** |
| --- | --- | --- |
| Actual duration vs plan | Metric to monitor the project progress compared to the plan | Allow up to a maximum 1 days delay. |
| Percentage of test cases run during test execution | Metric to monitor total number of test cases executed | 100% |
| Percentage of test cases passed against total of test cases executed | Metric to monitor total number of test cases executed with Pass criteria | All high-risk test cases pass. |
| Percentage of baseline requirement covered against total test cases | Metric to monitor total number of baseline requirement coverage | 100% |

## 4.2 Incident Classification

**Influence**

1. Mission critical: The application fails to run or the system fails.
2. Main problems: Serious problems, but may be solved.
3. Minor: It does not affect the function or availability of the process and does not meet the requirements / design specifications.

**Priorities**

1. Immediate: It must be repaired as soon as possible.
2. Delayed: The system is unstable, but the event must be fixed before the next level of test or shipment.
3. Deferred: Defects may be retained if necessary due to time or cost reasons.