

# Assignment 1 (E1800176 & E1800182)

*by I Made Siva Aditya Surya*

---

**Submission date:** 11-Mar-2021 07:35PM (UTC-0800)

**Submission ID:** 1530849888

**File name:** Assignment\_1\_Report\_-\_E1800176\_E1800182.docx (2.57M)

**Word count:** 8010

**Character count:** 42759



## Task 1 - Project Plan

## Table of Contents

Table of Contents .....	i
1. Overview .....	Error! Bookmark not defined.
1.1. Project Background and Motivation.....	Error! Bookmark not defined.
1.2. Project Customer/User .....	Error! Bookmark not defined.
1.3. Project Deliverables .....	Error! Bookmark not defined.
1.4. Project Estimate Cost .....	Error! Bookmark not defined.
1.5. Project Duration.....	Error! Bookmark not defined.
2. Project Aims .....	Error! Bookmark not defined.
3. Project Objectives .....	Error! Bookmark not defined.
4. Project Scope .....	Error! Bookmark not defined.
5. Project Schedule.....	Error! Bookmark not defined.
5.1. Work Breakdown Structure .....	Error! Bookmark not defined.
5.2. Milestones/Deliverable.....	Error! Bookmark not defined.
5.3. Baseline Gantt Chart .....	Error! Bookmark not defined.
6. Technical Description .....	Error! Bookmark not defined.
6.1. Development Platform .....	Error! Bookmark not defined.
6.2. Demonstration Platform.....	Error! Bookmark not defined.
7. Risk Management Plan.....	Error! Bookmark not defined.
Bibliography .....	Error! Bookmark not defined.

## **1. Overview**

### **1.1. Project Background and Motivation**

COVID-19 is an infectious virus disease that was first recognized in Wuhan, China, in December 2019 and has since spread worldwide (Wikipedia, 2021). Globally, as of 23 February 2021, a total of 111,419,939 COVID-19 confirmed cases and 2,470,772 deaths have reported to WHO (WHO, 2021). In Malaysia itself, 32,076 cases and 1,076 deaths have been reported as of 23 February 2021 (Malaysia MOH, 2021).

WHO has recommended robust diagnostic testing ever since the COVID 19 turned from an outbreak into a global pandemic. The testing is developed in order to differentiate the SARS-CoV-2 virus that causes COVID-19 disease from other respiratory infections as well as to develop a guidance of appropriate clinical management. In response to the WHO guidance, Malaysia has increased the capacity for weekly viral detection aside from intensive contract tracing and nationwide social distancing movement (Lim et al., 2020). This is an effort to enable Malaysia as a country to resume normal day-to-day activities as soon as possible.

The COVID-19 diagnostic test that are most commonly used in Malaysia are RT-PCR test, RTK-Antigen test, and antibody test. RT-PCR and RTK-Antigen are able to show whether a person has an active infection of COVID-19, while the antibody test shows if a person had a past infection of COVID-19. The public can get tested for COVID-19 in test centre registered at clinics or hospitals that was already approved by the Ministry of Health, Malaysia (MOH).

Joe Nollar stated that (Bonislawski, 2020), COVID-19 pandemic increases the demand for a direct-to-patient reporting of test results. The COVID-19 outbreak has accelerated the trend of developing an information system to have the capability of handling patient-initiated testing and reporting the result of the test directly to the patient. Suren Avunjian added (Bonislawski, 2020), an information system will enable the patient to provide all relevant information electronically instead of going through a cumbersome manual process.

Therefore, this project is initiated to develop a website information system to administer tests and keep track of the test result of COVID-19 patients under the name of HELP Covid-19 Testing Information System (CTIS) Website. HELP CTIS website is developed in hope to aid the health ministry by replacing the outdated and not thoroughly secured system that is used by the hospital and medical centre across the country.

## **1.2. Project Customer/User**

The HELP CTIS website is developed to be used by these following users:

1. Patient
2. Test Centre Officer
3. Test Centre Manager
4. Tester

The system will be managed by the Test Centre Manager which act as the representative of the test centre. Manager will be able register the test centre on the system, manage test kit stock, and record the Officer and Tester. Tester will be the one who record the test data of the Patient and update the test result. The Test Centre Officer (which also include the Manager and the Tester) can generate test report, view test kit, and view a list of all Officer that are employed in their Test Centre. Patient will be able to log in into the system and view their testing history.

## **1.3. Project Deliverables**

The end product of this project is a website information system under the name of HELP CTIS (Covid-19 Testing Information System) that enable test centre to manage test result and test kit, and enable patient to view their testing history.

## **1.4. Project Estimate Cost**

The project starts on 18<sup>th</sup> February 2021 and ends on 30<sup>th</sup> April 2021, with a total of 496 working hours. The project consisted of two members, a project leader and a project member which both have a labor cost of \$ 2.45/hour (RM 10/hour). The total of labor cost of this project is estimated to be \$ 2,430. The cost of internet bills and electricity throughout the whole project is estimated to be \$ 1,000. Therefore, the total of cost estimate for this project is \$ 3,430.

## **1.5. Project Duration**

Start Date : Thursday, 18<sup>th</sup> February 2021

End Date : Tuesday, 30<sup>th</sup> April 2021

## **2. Project Aims**

1. Provide solutions for a new COVID-19 testing system used by the health ministry, health workers, and the public
2. Provide a system that can enable direct-to-patient reporting of COVID-19 test results
3. Provide a system that can encourage appropriate clinical management of COVID-19 diagnostic testing

## **3. Project Objectives**

1. Create a project plan as the initiation of the project.
2. Create a requirement specification and design specification documents to state all the requirements and define the UML modelling of the website project.
3. Develop a website information system which can administer COVID-19 test to the patient and record the result.
4. Develop a website system that gives test centre manager the ability to do COVID-19 test kit management through the system
5. Develop a website information system that can be used by COVID-19 patient to check on their test result and view their testing history
6. Develop a website system that can be easily accessed through PC and smartphone devices
7. Develop a test plan for the system and compile the result as testing document
8. Create a final report to indicates the end of project

## 4. Project Scope

<b>Project Scope Statement</b>
<b>Project Title:</b> HELP COVID-19 Testing Information System (CTIS) Website
<b>Date:</b> 27 <sup>th</sup> February 2021
<b>Prepared by:</b> I Made Siva Aditya Surya (Team Leader/mdsivaaditya@gmail.com)
<b>Project Summary:</b> <p>This project is initiated to aid the health ministry by developing a website information system to administer tests and keep track of the test result of COVID-19 patients. Test centre will be able to manage test kit stock, generate test report, and do test result management. Patient will be able to log in to the system and view their test result history. This project is aimed to enable a direct-to-patient reporting system of COVID-19 test results.</p>
<b>High Level Requirements:</b> <ol style="list-style-type: none"><li>1. Patient can log in to the system to check on their test results</li><li>2. Tester can record the test that are already administered and the obtained results</li><li>3. Manager can do test kit management on the system and record officer/tester</li><li>4. Officer can generate report of patient test which are already administered in the test centre</li></ol>
<b>Summary of Project Deliverables</b> <b>Project management-related deliverables:</b> Project overview, project aims, project objectives, project scope statement, WBS, project schedule, project proposal/project plan, risk management plan, requirement specification document, gantt chart, final report, evaluation report, and any other documents that is required to manage the project.
<b>Product-related deliverables:</b> <ol style="list-style-type: none"><li>1. Testing report</li><li>2. UML design diagrams</li><li>3. A website system to record test result and generate test report.</li><li>4. Ability to update test result in the website by the tester.</li><li>5. Test kit stock management in the website by the test centre manager.</li></ol>

- |   |
|---|
| <ul style="list-style-type: none"><li>6. Test centre registration in the website by the test centre manager.</li><li>7. Patient can website and view their own testing history.</li><li>8. Website system can be accessed through browser in smartphones and PC device.</li></ul> |
|---|

**Project Out of Scope:**

- |  |
|--|
| <ul style="list-style-type: none"><li>1. Support a consultation system for the patient before taking the test.</li><li>2. A test result viewing system by entering the ID of a patient.</li><li>3. A map live view of the test centre.</li></ul> |
|--|

**Project Success Criteria:**

<p>We will consider our project a success if we are able to fulfil all the requirements and deliverables that have been stated in this project scope statement within the project duration that is 3 months which ended in 30<sup>th</sup> April 2021 and has already been stated in the Gantt Chart.</p>
---

## **5. Project Schedule**

### **5.1. Work Breakdown Structure**

1. Initiating Tasks
  - 1.1. Selecting Project Leader
  - 1.2. Conducting Research
  - 1.3. Identifying Project Background
  - 1.4. Identifying Functional and Non-functional Requirements
2. Planning Tasks
  - 2.1. Determining Project Scope
  - 2.2. Establishing WBS
  - 2.3. Milestones/Deliverable
  - 2.4. Project Schedule and Baseline Gantt Chart
  - 2.5. Development and Demonstration Platform
  - 2.6. Risk Management Plan
  - 2.7. Modelling Use Case Diagram and Class Diagram
  - 2.8. Expanded Use Cases and System Sequence Diagrams
  - 2.9. Analysis Class Diagram
3. Executing Tasks
  - 3.1. Designing the Web Page
  - 3.2. Developing Prototype
  - 3.3. System Finishing
4. Monitoring and Controlling Tasks
  - 4.1. Update Gantt Chart
  - 4.2. Update Requirements Specification
  - 4.3. Developing & Update Design Specification
  - 4.4. Testing the Prototype
  - 4.5. System Testing
5. Closing
  - 5.1. Final Report

## 5.2. Milestones/Deliverable

Schedule	Start Date	End Date	Estimate Days	Status	Responsible
<b>Initiating Tasks</b>					
<b>Selecting Project Leader</b>	Thu, 2/18/2021	Thu, 2/27/2021	1 day	Complete	All
<b>Conducting Research</b>	Fri, 2/19/2021	Mon, 2/22/2021	3 days	Complete	All
<b>Identifying Project Background</b>	Tue, 2/23/2021	Wed, 2/24/2021	2 days	Complete	Siva Aditya
<b>Determining Project Scope</b>	Thu, 2/25/2021	Sat, 2/27/2021	3 days	Complete	All
<b>Finish Project Initiation</b>	Mon, 3/1/2021	Mon, 3/1/2021	0 day	Complete	All
<b>Planning Tasks</b>					
<b>Identifying Functional and Non-functional Requirements</b>	Fri, 2/19/2021	Sat, 2/20/2021	2 days	Complete	All
<b>Establishing WBS</b>	Sat, 2/20/2021	Sat, 2/20/2021	1 day	Complete	Siva Aditya
<b>Milestones/Deliverable</b>	Sat, 2/20/2021	Sat, 2/20/2021	1 day	Complete	Siva Aditya
<b>Project Schedule and Baseline Gantt Chart</b>	Sat, 2/20/2021	Sat, 2/20/2021	1 day	Complete	Siva Aditya
<b>Development and Demonstration Platform</b>	Mon, 2/22/2021	Tue, 2/23/2021	2 days	Complete	Awidya Andika
<b>Risk Management Plan</b>	Wed, 2/24/2021	Thu, 2/25/2021	2 days	Complete	All
<b>Modelling Use Case Diagram and Class Diagram</b>	Fri, 2/26/2021	Mon, 3/1/2021	3 days	Complete	Awidya Andika
<b>Expanded Use Cases and System Sequence Diagrams</b>	Tue, 3/2/2021	Thu, 3/4/2021	3 days	Complete	All
<b>Analysis Class Diagram</b>	Fri, 3/5/2021	Sat, 3/6/2021	2 days	Complete	Awidya Andika

<b>Finish Project Requirements Specification</b>	Mon, 3/8/2021	Mon, 3/8/2021	0 day	Complete	All
<b>Executing Tasks</b>					
<b>Designing the Web Page</b>	Mon, 3/8/2021	Sat, 4/10/2021	30 days	Ongoing	All
<b>Developing Prototype</b>	Mon, 3/8/2021	Sat, 4/10/2021	30 days	Ongoing	All
<b>System Finishing</b>	Mon, 4/12/2021	Tue, 4/27/2021	14 days	-	All
<b>Final Product Deliverables</b>	Wed, 4/28/2021	Wed, 4/28/2021	0 day	-	All
<b>Monitoring and Controlling Tasks</b>					
<b>Update Gantt Chart</b>	Mon, 2/22/2021	Fri, 4/30/2021	59 days	Ongoing	Siva Aditya
<b>Update Requirements Specification</b>	Thu, 3/4/2021	Tue, 4/27/2021	47 days	Ongoing	Siva Aditya
<b>Develop &amp; Update Design Specification</b>	Thu, 3/4/2021	Tue, 4/27/2021	47 days	Ongoing	Siva Aditya
<b>Testing the Prototype</b>	Mon, 3/8/2021	Sat, 4/10/2021	30 days	Ongoing	All
<b>System Testing</b>	Mon, 4/12/2021	Tue, 4/27/2021	14 days	-	All
<b>Test Documents Compiled</b>	Fri, 4/28/2021	Fri, 4/28/2021	0 day	-	All
<b>Final Design Specification Document</b>	Wed, 4/28/2021	Wed, 4/28/2021	0 day	-	All
<b>Closing</b>					
<b>Final Report</b>	Wed, 4/28/2021	Thu, 4/29/2021	2 days	-	All
<b>End of Project</b>	Fri, 4/30/2021	Fri, 4/30/2021	0 day	-	All

### 5.3. Baseline Gantt Chart

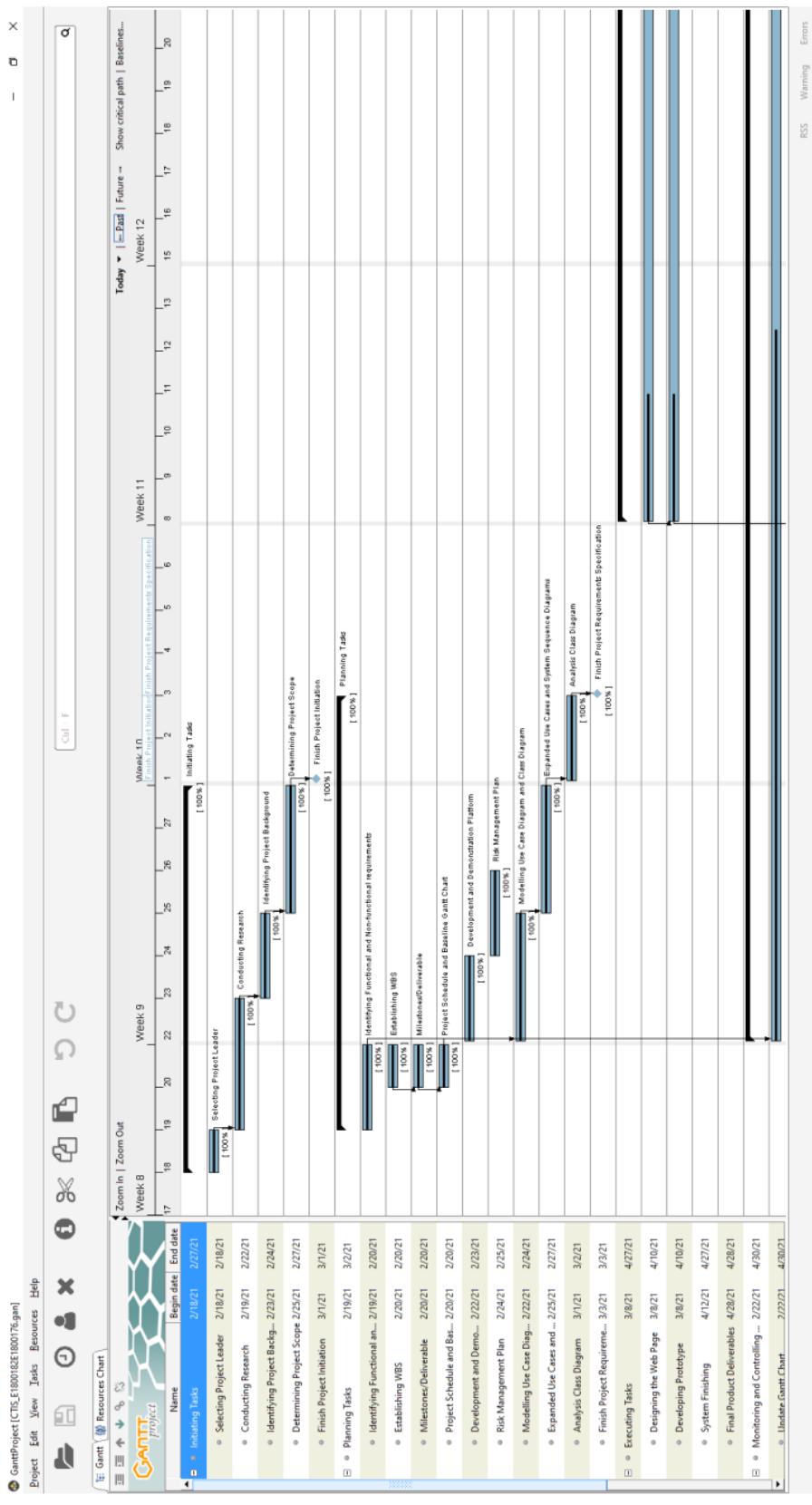


Figure 1. Gantt Chart (part. 1)

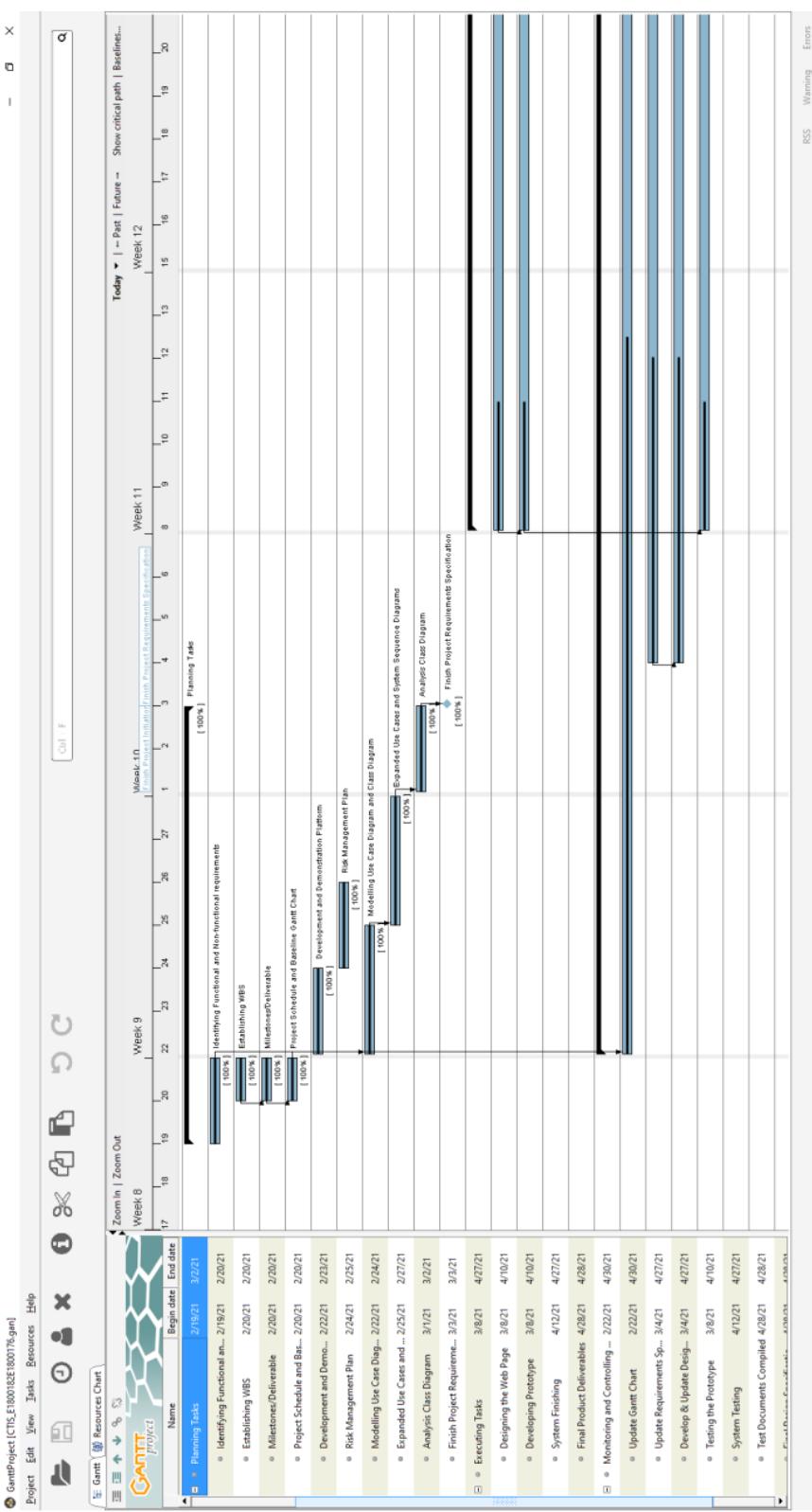


Figure 2. Gantt Chart (part. 2)

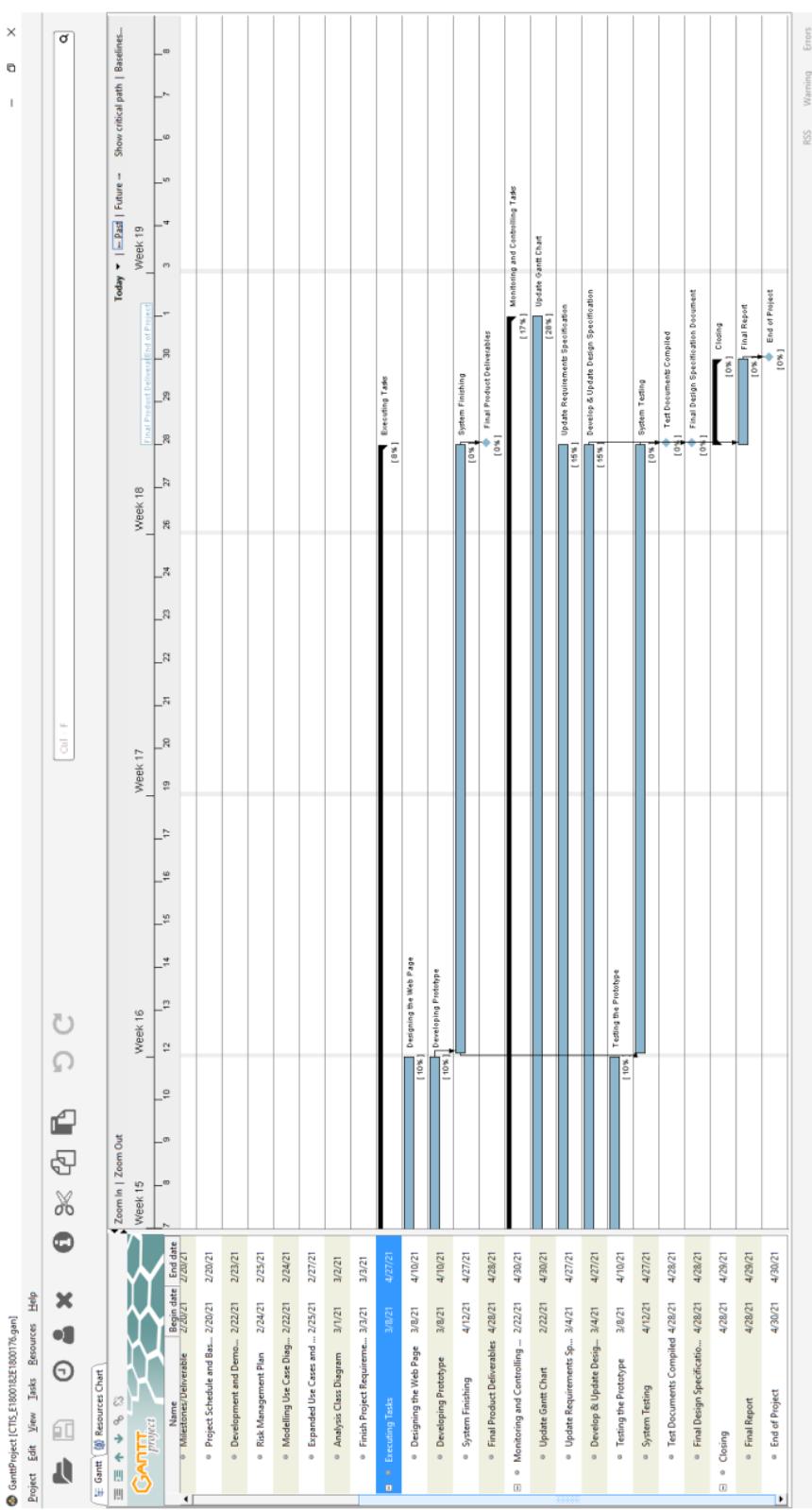


Figure 3. Gantt Chart (part. 3)

## **6. Technical Description**

### **6.1. Development Platform**

#### **Software/Tools**

##### **1. Microsoft Word**

We will use Microsoft Word to create all documents and reports related to the project.

##### **2. Microsoft Powerpoint**

We will use Microsoft Powerpoint to create a presentation file of the project.

##### **3. Microsoft Visio and Visual Paradigm**

We will use Microsoft Visio and Visual Paradigm to create UML model such as use case diagram, class diagram, system sequence diagram, and entity relationship diagram of this project.

##### **4. Bootstrap, Laravel, and jQuery**

We will use Laravel as our PHP web framework because it is easy to connect with MySQL and can facilitate rapid web application development. We use Bootstrap as our CSS framework and jQuery as our JavaScript framework.

##### **5. MySQL**

We will use MySQL, a relational database management system based on SQL, because it is open source, reliable, and easy to manage.

##### **6. XAMPP**

We will use XAMPP to run a web server and database on the computer localhost for developing and testing purposes of the website.

##### **7. PhpStorm**

We will use PhpStorm which is a cross-platform IDE for PHP built by the company JetBrains as the tools. We choose to use this IDE because it provides an editor for PHP, HTML, CSS, and JavaScript with auto-completion and error debugging.

##### **8. Adobe Photoshop**

We will use Photoshop to edit images that will be used in our website design.

##### **9. GanttProject**

We will use GanttProject to create the Gantt chart of our project.

## 10. Trello

We will use Trello to manage and keep track of the tasks in this project.

## 11. Git and GitHub

We will use the combination of Git and GitHub to do the version control system of our project. We also use GitHub as the collaboration tool between the project members during the development of the system.

## 12. Google Chrome

We will use Google Chrome browser for testing purposes of our project. We choose Chrome because it is the most commonly used browser and we only need to do minor adjustment in Chrome in order for the website page to appear nicely in another browser.

### **Hardware**

#### 1. Laptop and PC

The development of this system will be done using laptop and PC running Windows operating system.

## **6.2. Demonstration Platform**

### **Software**

#### 1. Web browser

We choose Google Chrome, Mozilla Firefox, and Microsoft Edge as the demonstration platform for our website because those three are among the most common browsers used now days. We are also considering Microsoft Edge since we are developing the website in PC that run Windows.

#### 2. Microsoft Powerpoint

We will use Microsoft Powerpoint to present all the project-management related documents.

### **Hardware**

#### 1. Laptop and PC

The website is intended to be opened from a laptop or a PC that have one of the browsers mentioned in software platform installed.

#### 2. Smartphone

Our website is also intended to be opened from a smartphone device browser.

## 7. Risk Management Plan

Risk Management Plan for HELP Covid-19 Testing Information System Website Project									Date: 25 <sup>th</sup> February 2021
NO	Risk Type	Risk	Risk Description	Risk Probabilty	Risk Impact	Risk Assess-ent Grade	Trigger	Mitigation Plan	Contingency Plan
R1	Operational Risk	Failure to fully understand all requirements in the project	If requirements are not fully grasped by the project team members, the system that will be delivered may fail to match the user needs and expectations	M	M	4	Poorly executed brainstorming of the project requirement and project scope or lack of research conducted on the project	Allocate more time to do research and brainstorming, communicate found similar system/event	Conduct a thorough research on the requirements
R2	Schedule Risk	Failure to meet the task deadlines in the project/schedule/exceeding working time	Team members are overwhelmed with another assignment/project/event that led to the project to be put aside and left unworked	H	M	4	Busy schedule or sudden personal agenda from team members that is unavoidable (accident, family occasion,etc)	All	Allocate more time on the task that is already past the deadline thus it can be finished quickly, and replanning the schedule/deadline of the remaining task
R3	Management Risk	Conflict between project members	Lack of communication/Misunderstanding across project team members may spark conflict or unclear task delegation	H	H	5	Busy schedule that results in poor communication between project team members, poorly delivered thoughts and ideas across team members	Siva Aditya	Even in this pandemic time, try to conduct a physical meeting occasionally to avoid miscommunication
R4	Technical Risk	Bugs, crashes, and errors in system	The system may fail to meet the expectation outcome due to logical/integration error	M	H	4	Bad coding implementation, not enough testing	Awidya Andika	Do a more thorough checking and conducting testing after every code implementation
									Checking the part of system that is causing the error and solve the error by redo the faulty features or implement another method

## Risk Assessment Matrix

		Severity of Harm (Impact)		
		Low (L)	Medium (M)	High (H)
Probability	High (H)	3	4	5
	Medium (M)	2	3	4
	Low (L)	1	2	3

Risk Mitigation Based Upon Grade	
Grade	Possible Action
5	Priority risk, mitigation actions are to be identified and implemented at the start of the project
4	Mitigation actions are to be identified and implemented throughout the course of the project
3	Mitigation actions are implemented should the time and cost permit execution
2	Risk to be noted and actions are only required should risk raise in grade over time
1	Risk to be noted and actions are only required should risk raise in grade over time

## Bibliography

## **Task 2 – Requirements Specification Documents**

## Table of Contents

<b>Table of Contents .....</b>	i
<b>1. Functional and Non-functional Requirements .....</b>	1
<b>2. Use Case Diagram .....</b>	4
<b>3. Analysis Class Diagram .....</b>	6
<b>4. Expanded Use Cases, System Sequence Diagrams, &amp; Contracts .....</b>	8
<b>4.1. Login.....</b>	8
<b>4.2. View Test Kit .....</b>	10
<b>4.3. Manage Test Kit Stock .....</b>	12
<b>4.4. Register Test Centre .....</b>	15
<b>4.5. View Test Centre Officer.....</b>	17
<b>4.6. Record Tester .....</b>	19
<b>4.7. Record Officer .....</b>	23
<b>4.8. Record New Test .....</b>	26
<b>4.9. Update Test Result.....</b>	29
<b>4.10. Generate Test Report.....</b>	31
<b>4.11. View Testing History .....</b>	33
<b>4.12. Log Out .....</b>	35
<b>5. Task Division .....</b>	38
<b>Bibliography .....</b>	39

## **1. Functional and Non-functional Requirements**

### **Functional Requirements**

#### **Patient Requirements:**

1. A menu to login to the system
2. Username and password verification in login menu
3. Home page menu after patient successfully login to system
4. A button in home page menu to navigate to view testing history menu
5. View testing history menu that shows the patient's Covid-19 test data and result in chronological order
6. A button to log out from the system

#### **Test Centre Officer Requirements:**

1. A menu to login to the system
2. Username and password verification in login menu
3. Test Centre Officer dashboard after successfully login to system
4. A sidebar in dashboard to navigate to test report menu, test centre officer menu, and test kit menu
5. Test report menu that shows a list of all test that have been administered in the test centre that the officer is employed at are shown in chronological order
6. Test centre officer menu that shows a list of all test centre officer (regardless of position) that is employed in the test centre
7. Test kit menu that shows a list of all test kit that the test centre has and the stock

#### **Tester Requirements:**

1. A menu to login to the system
2. Username and password verification in login menu
3. Tester dashboard after successfully login to system
4. A sidebar in dashboard to navigate to patient test menu, test report menu, test centre officer menu, and test kit menu

5. A form in patient test menu to record new test data with appropriate label and column to fill out patient data, patient type, and symptoms and a button to record the data
6. Update test result function in patient test menu that shows a list of test data in chronological order with a button to edit test result, the button will navigate to update test result page with appropriate form to fill out the result data and a button to update the test result
7. Test report menu that shows a list of all test that have been administered in the test centre that the officer is employed at are shown in chronological order
8. Test centre officer menu that shows a list of all test centre officer (regardless of position) that is employed in the test centre
9. Test kit menu that shows a list of all test kit that the test centre has and its stock

#### **Test Centre Manager Requirements:**

1. A menu to login to the system
2. Username and password verification in login menu
3. Test Centre Manager dashboard after successfully login to system
4. Register test centre page when the manager first login to system, to verify and record the manager's test centre into database
5. A sidebar in dashboard to navigate to test report menu, test kit menu, and test centre officer menu
6. Test kit menu that includes a list of all test kit available with its stock, a button to input new test kit, and a button to update test kit details (stock)
7. Test centre officer menu that shows a list of all test centre officer (regardless of position) that is employed in the test centre
8. Record officer function in test centre officer menu that includes a form with appropriate label and column to fill out new officer data, a username verification, and a button to record the data
9. Record tester function in test centre officer menu that includes a button to update the officer position to tester if the manager wants to change the position of an existing officer to tester, or a form to fill out officer data if the manager wants to record new officer with the position of tester
10. Test report menu that shows a list of all test that have been administered in the test centre that the officer is employed at are shown in chronological order

### **Non-functional Requirements**

1. Correctness – all the information that is recorded to the system and provided by the system must be accurate, complete, and up-to-date. To support this requirement, the system must have a data verification every time a data is about to be recorded to the database.
2. Efficiency – the system should be lightweight and does not require extensive amount of hardware resources to perform a function.
3. Privacy of information – personal patient and officer data that is stored in the database must be properly consent and regulated.
4. Integrity – the system must control unauthorized person from accessing and changing the data. The only one who can record and update test centre officer data and position is the manager. The only who can record and update patient's test data and test result is the tester.
5. Usability – the system should be easy to learn, understand, and operate all the features. To support this requirement, a good design for the website with appropriate icon for buttons and menu should be implemented.
6. Reliability – the system should perform all its functions as intended without failure.

## 2. Use Case Diagram

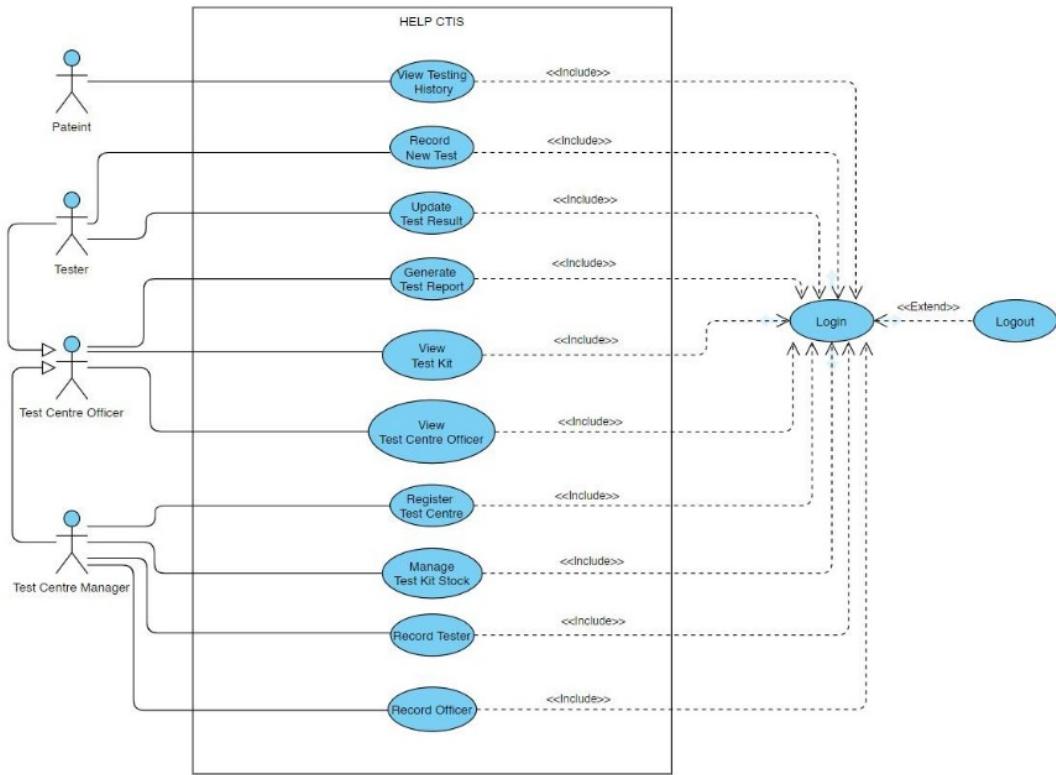


Figure 4. Use Case Diagram

This project has four actors (Patient, Test Centre Officer, Tester, Manager) and 12 use cases. Patient has three use cases: View Testing History, Login, and Logout. Test Centre Officer has five use cases: Generate Test Report, View Test Kit, View Test Centre Officer, Login, and Logout. Tester has a generalization relation with Test Centre Officer therefore it inherits all the use cases of Test Centre Officer. Aside from the inherited use cases, Tester has two other use cases: Record New Test and Update Test Result. Test Centre Manager also has a generalization with Test Centre Officer and inherits all the use cases of Test Centre Officer. Test Centre Manager, aside from the inherited use cases, has four use cases: Register Test Centre, Manage Test Kit Stock, Record Tester, and Record Officer.

Some use cases use “include” and “extend” dependency relation. Every use case (except Logout) has an “include” dependency relation with use case Login, it indicates that to do the use cases the actor must done the Login use case first. The Logout use case, on the other hand, has an “extend”

dependency relation with use case Login. The “extend” dependency indicates that the actor that have done Login do not essentially have to Logout, but in order to Logout they have to Login first.

No	Requirement	Actor	Use Case
1	Allow Patient to view their testing history	Patient	View Testing History
2	Allow Tester to record Patient and their test data	Tester	Record New Test
3	Allow Tester to update Patient's test result and status	Tester	Update Test Result
4	Allow Officer to generate a list of test report that are administered in their Test Centre	Test Centre Officer	Generate Test Report
5	Allow Officer to view a list of test kit and stock	Test Centre Officer	View Test Kit
6	Allow Officer to view a list of all employed Officer in their Test Centre	Test Centre Officer	View Test Centre Officer
7	Allow Test Centre to be created and recorded for the Manager	Test Centre Manager	Register Test Centre
8	Allow Manager to add new test kit and update stock	Test Centre Manager	Manage Test Kit Stock
9	Allow Manager to record Officer as Tester	Test Centre Manager	Record Tester
10	Allow Manager to record new Officer	Test Centre Manager	Record Officer
11	Allow User to login to the system	Patient, Tester, Test Centre Officer, Test Centre Manager	Login
12	Allow User to log out from the system	Patient, Tester, Test Centre Officer, Test Centre Manager	Logout

### 3. Analysis Class Diagram

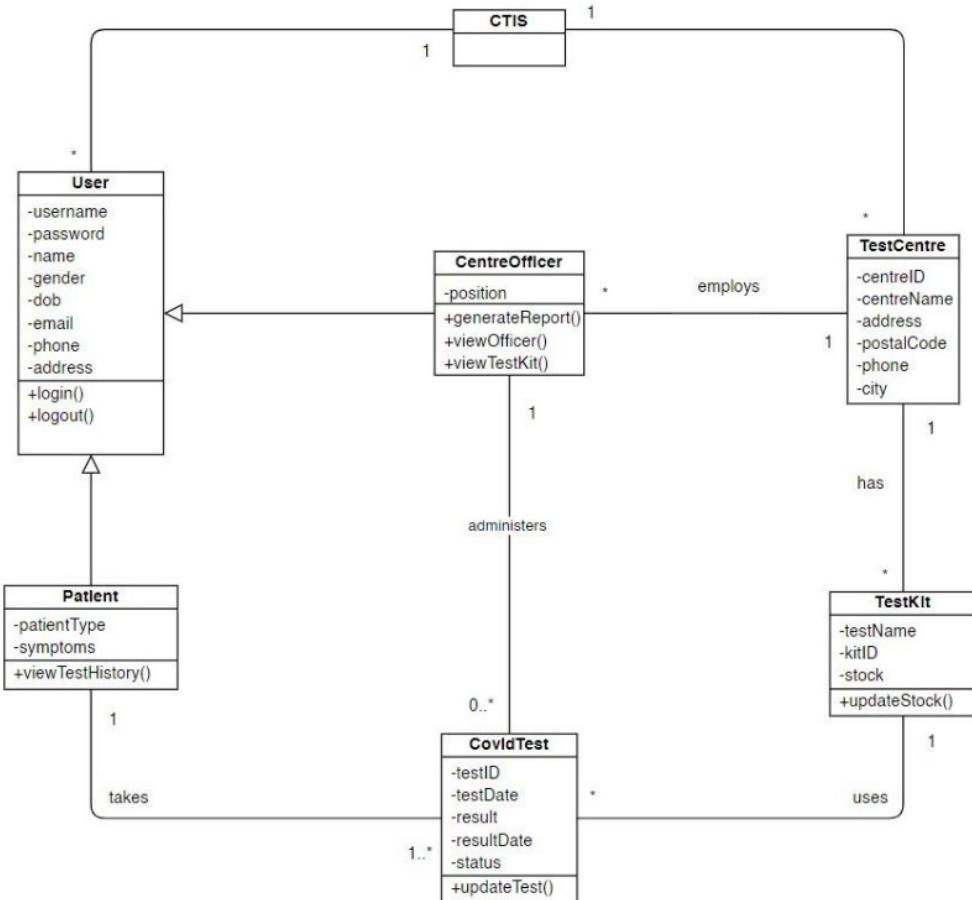


Figure 5. Class Diagram

This project has seven classes: CTIS, TestCentre, TestKit, User, CentreOfficer, Patient, and CovidTest. CTIS class represents the system. User class act as the parent class for CentreOfficer and Patient, it has eight attributes and two operations. CentreOfficer class inherits to User class and represents the Test Centre Officer, Test Centre Manager, and Tester actor from the use case diagram. The attributes “position” in CentreOfficer class will differentiate the three actors in the system, with each actor have its own distinctive “position”. The CentreOfficer is employed in only one Test Centre and has three operations which generate data from the Test Centre. The TestCentre class represents the Test Centre which employs many Officer and has many Test Kit. The attributes in TestCentre class relates to the location of the Test Centre. Patient class inherits to User class and represents the Patient

actor from the use case diagram, it has “symptoms” as its attribute and also “patientType” which is one of the five types (returnee, quarantined, close contact, infected, or suspected). Patient is able to take one or more tests, and will be able to view their testing history via the operations `viewTestHistory()`. TestKit class represents the Test kit that will be used in patient covid test. Test Kit will be used in many tests therefore it has “stock” attribute to indicates the availability of the test kit stock. CovidTest class represents the Covid Test that will be taken by the Patient and is administered by the Centre Officer (specifically the Tester). The CovidTest has the attributes “result” which is the result of the test, “status”, and date attributes (“testDate” and “resultDate”). The status of Covid Test will be updated according to the stages of the test.

## 4. Expanded Use Cases, System Sequence Diagrams, & Contracts

### 4.1. Login

Prepared by: Ida Bagus Gede Awidya Andika

- Expanded Use Case

Use Case	Log in
Goal in Context	Allow User to login to the system
Primary Actor	Test Centre Officer and Patient
Secondary Actor	-
Typical Course of Events	
Actor Actions	System Response
1. Process begins when the Test Centre Officer/Patient input their username and password and click the login button	2. System will verify the input data, whether it matches the data in database or not
	3. System will navigate the user to their dashboard
Alternative Course	
1a: If Test Centre Officer or Patient inputs incorrect username or password, then the system will display a warning notification that includes an instruction to try log in again	

- System Sequence Diagram

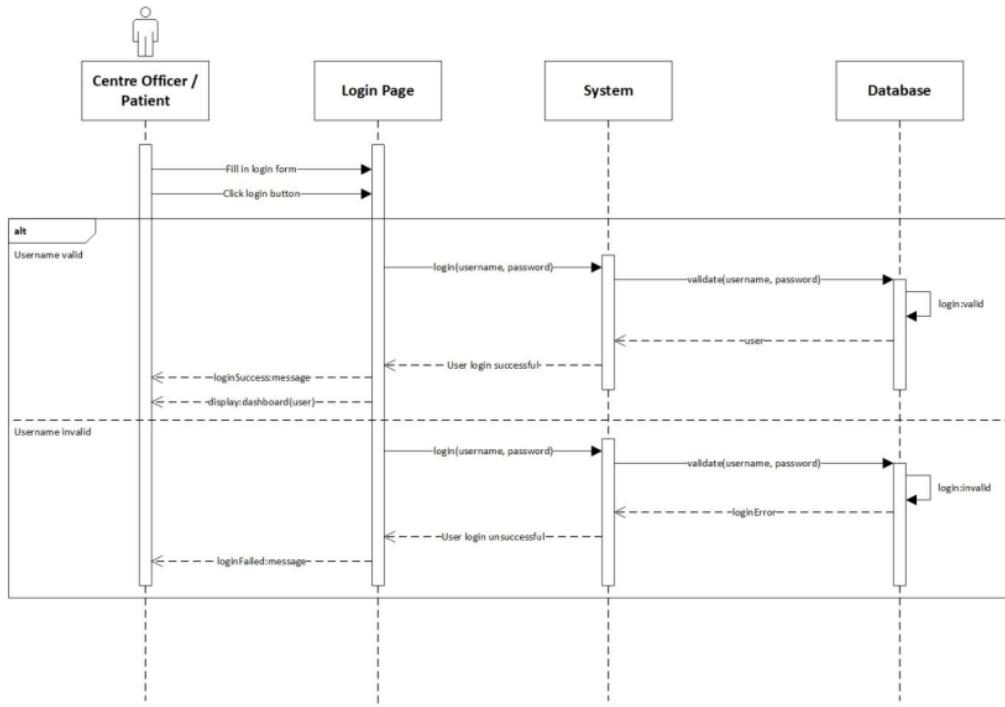


Figure 6. Login System Sequence Diagram

- Contract

Cross References	Login
Operation	login(username, password)
Responsible	To access the user's dashboard.
Pre-conditions	<ul style="list-style-type: none"> <li>User object must be available</li> <li>A valid username and password</li> </ul>
Post-conditions	<ul style="list-style-type: none"> <li>Session was created (instance creation)</li> <li>Session was associated with the User (association formed)</li> </ul>
Cross References	Login
Operation	login(username, password)
Responsible	To access the user's dashboard.

Pre-conditions	<ul style="list-style-type: none"> <li>User object must be available</li> <li>Invalid username and/or password</li> </ul>
Post-conditions	<ul style="list-style-type: none"> <li>Session was not created</li> </ul>

#### 4.2. View Test Kit

**Prepared by: Ida Bagus Gede Awidya Andika**

- Expanded Use Case

Use Case	View Test Kit
<b>Goal in Context</b>	Allow Officer to view a list of test kit and stock
<b>Primary Actor</b>	Test Centre Manager
<b>Secondary Actor</b>	-
<b>Typical Course of Events</b>	
Actor Actions	System Response
1. Process begins after Test Centre Manager has successfully login to system	2. System will navigate Test Centre Manager to their dashboard
3. Test Centre Manager clicks test kit menu	4. System displays a list of test kit data
<b>Alternative Course</b>	
-	

- System Sequence Diagram

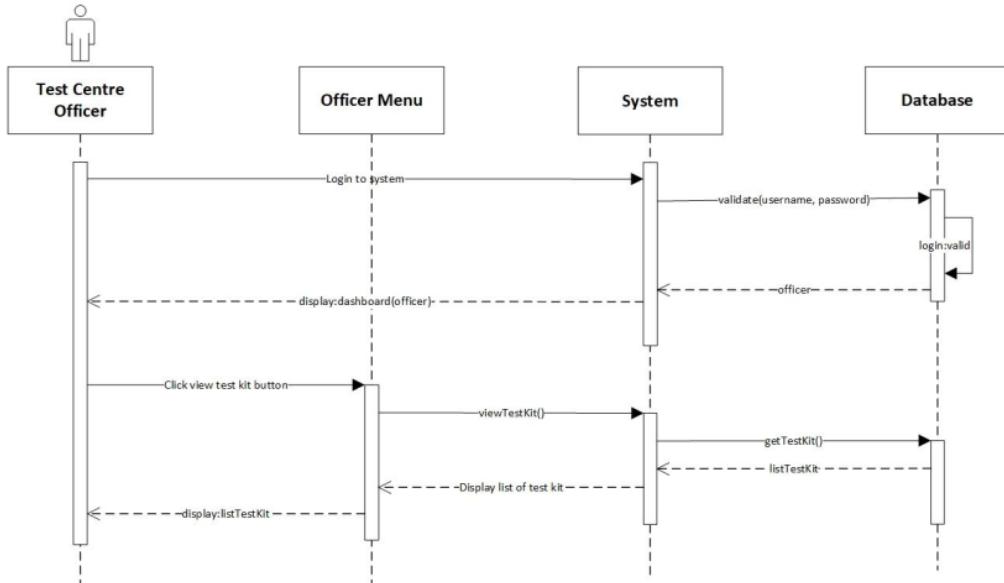


Figure 7. View Test Kit System Sequence Diagram

- Contract

Cross References	View Test Kit
Operation	login(username, password)
Responsible	To access the officer's dashboard.
Pre-conditions	<ul style="list-style-type: none"> <li>• Officer object must be available</li> <li>• A valid username and password</li> </ul>
Post-conditions	<ul style="list-style-type: none"> <li>• Session was created (instance creation)</li> <li>• Session was associated with the Officer (association formed)</li> </ul>
Cross References	View Test Kit
Operation	viewTestKit()
Responsible	To view a list of all test kit data.
Pre-conditions	testKit object must be available
Post-conditions	<ul style="list-style-type: none"> <li>• ListTestKit was created (instance creation)</li> </ul>

	<ul style="list-style-type: none"> <li>• ListTestKit was associated with viewMenu (association formed)</li> </ul>
--	---

#### 4.3. Manage Test Kit Stock

**Prepared by: Ida Bagus Gede Awidya Andika**

- Expanded Use Case

Use Case	Manage Test Kit Stock
<b>Goal in Context</b>	Allow Test Centre Manager to add test kit or update test kit stock
<b>Primary Actor</b>	Test Centre Manager
<b>Secondary Actor</b>	-
<b>Typical Course of Events</b>	
Actor Actions	System Response
1. Process begins after Test Centre Manager has successfully login to system	2. System will navigate Test Centre Manager to their dashboard
3. Test Centre Manager clicks test kit menu	4. System will display a list of test kit data
5. Test Centre Manager clicks add test kit	6. System will navigate Test Centre Manager to add test kit page and display test kit data form
7. Test Centre Manager fills in the data for the name of the test kit and the amount of stock on the form provided, then clicks the submit button	8. System will store the data that has been entered by the Test Centre Manager into the database then Test Centre Manager will be directed to the test kit page and a message that notifies success in adding the test kit is displayed

## Alternative Course

**5a:** if test kit is already recorded in the database, Test Centre Manager will be able to update the stock amount of the available test kit

- System Sequence Diagram

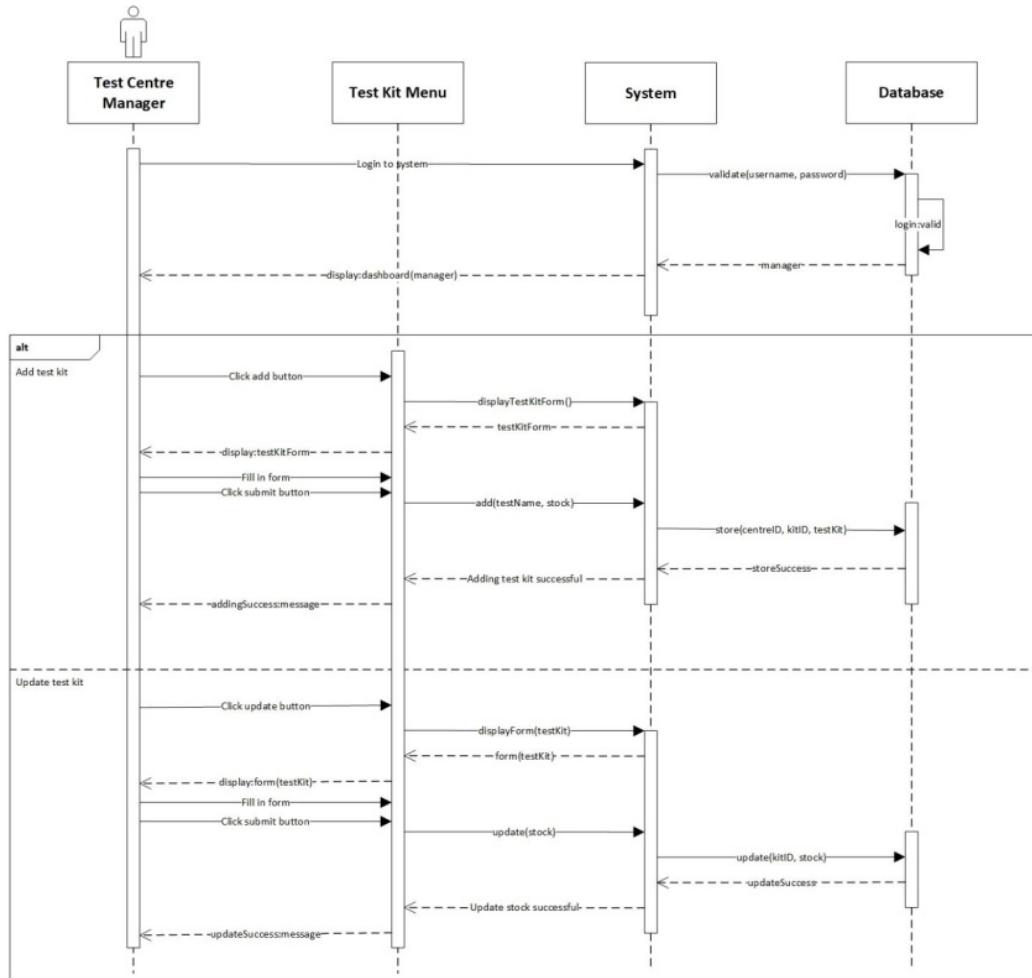


Figure 8. Manage Test Kit Stock System Sequence Diagram

- Contract

<b>Cross References</b>	<b>Manage Test Kit Stock</b>
Operation	login(username, password)
Responsible	To access the manager's dashboard.
Pre-conditions	<ul style="list-style-type: none"> <li>• Manager object must be available</li> <li>• A valid username and password</li> </ul>
Post-conditions	<ul style="list-style-type: none"> <li>• Session was created (instance creation)</li> <li>• Session was associated with the Manager (association formed)</li> </ul>
<b>Cross References</b>	<b>Manage Test Kit Stock</b>
Operation	add(testName, stock)
Responsible	To add new test kit with name and available stock.
Pre-conditions	<ul style="list-style-type: none"> <li>• Manager object must be available</li> <li>• TestCentre object must be available</li> <li>• Session must already be created</li> <li>• Test name and stock data is available</li> </ul>
Post-conditions	<ul style="list-style-type: none"> <li>• TestKit was created (instance creation)</li> <li>• TestDate was generated (attribute modification)</li> <li>• KitID was generated (attribute modification)</li> <li>• TestKit was associated with the Test Centre (association formed)</li> </ul>
<b>Cross References</b>	<b>Manage Test Kit Stock</b>
Operation	update(stock)
Responsible	To update available stock from test kit.
Pre-conditions	<ul style="list-style-type: none"> <li>• Manager object must be available</li> <li>• Session must already be created</li> <li>• TestKit object must be available</li> <li>• Stock data is available</li> </ul>
Post-conditions	Stock was updated (attribute modification)

#### **4.4. Register Test Centre**

**Prepared by: Ida Bagus Gede Awidya Andika**

- Expanded Use Case

Use Case	Register Test Centre
Goal in Context	Allow Test Centre to be created and recorded for the Manager
Primary Actor	Test Centre Manager
Secondary Actor	-
<b>Typical Course of Events</b>	
Actor Actions	System Response
1. Process begins when Test Centre Manager first login to the system	2. System will navigate Test Centre Manager to register test centre page and display test centre data form
3. Test Centre Manager fills in the name and address of the test centre on the form provided and then clicks the register button	4. System will store the data that has been inputted by the Test Centre Manager into the database, then Test Centre Manager will be directed to the test centre page and a message that notifies success in registering the test centre is displayed
<b>Alternative Course</b>	
-	

- System Sequence Diagram

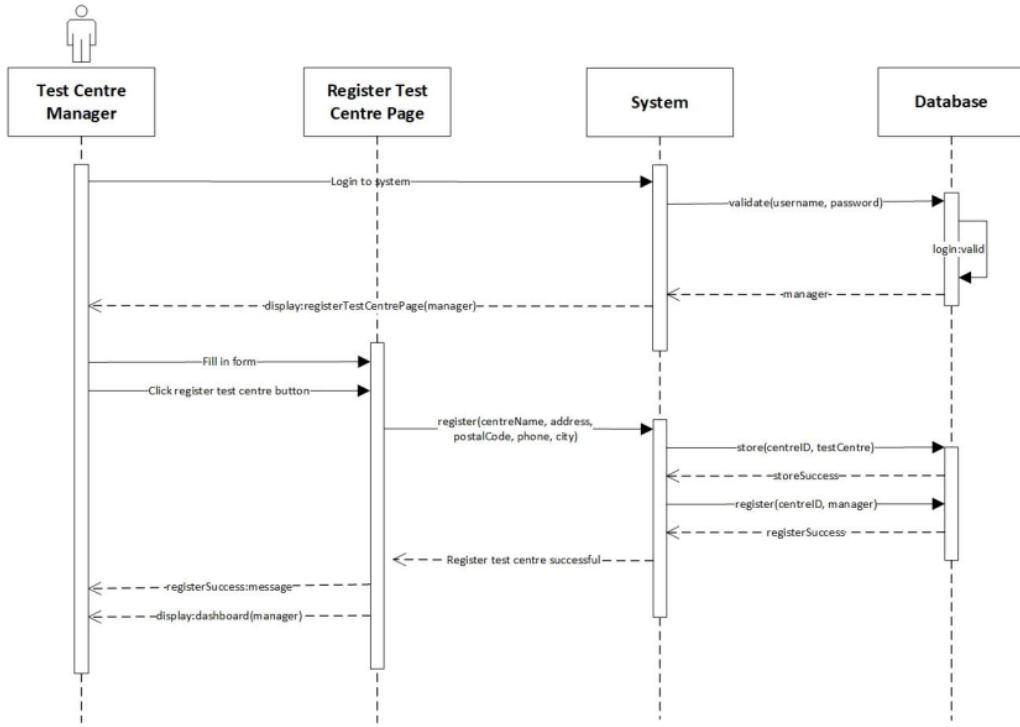


Figure 9. Register Test Centre System Sequence Diagram

- Contract

Cross References	Register Test Centre
Operation	login(username, password)
Responsible	To access the manager's dashboard.
Pre-conditions	<ul style="list-style-type: none"> <li>Manager object must be available</li> <li>A valid username and password</li> </ul>
Post-conditions	<ul style="list-style-type: none"> <li>Session was created (instance creation)</li> <li>Session was associated with the Manager (association formed)</li> </ul>
Cross References	Register Test Centre
Operation	register(centreName, address, postalCode, phone, city)

Responsible	To register test centre and record for the manager.
Pre-conditions	<ul style="list-style-type: none"> <li>Manager object must be available</li> <li>Session must already be created</li> <li>Test centre is not registered for Manager</li> <li>Test centre data is available</li> </ul>
Post-conditions	<ul style="list-style-type: none"> <li>TestCentre was created (instance creation)</li> <li>centreId was generated (attribute modification)</li> <li>Manager was associated with the TestCentre (association formed)</li> </ul>

#### 4.5. View Test Centre Officer

Prepared by: Ida Bagus Gede Awidya Andika

- Expanded Use Case

Use Case	View Test Centre Officer
Goal in Context	Allow Officer to view a list of all employed Officer in their Test Centre
Primary Actor	Test Centre Officer
Secondary Actor	-
Typical Course of Events	
Actor Actions	System Response
1. Process begins after Test Centre Officer has successfully login to system	2. System will navigate Test Centre Officer to their dashboard
3. Test Centre Officer clicks test centre officer menu	4. System will display a list of test centre officer data
Alternative Course	
-	

- System Sequence Diagram

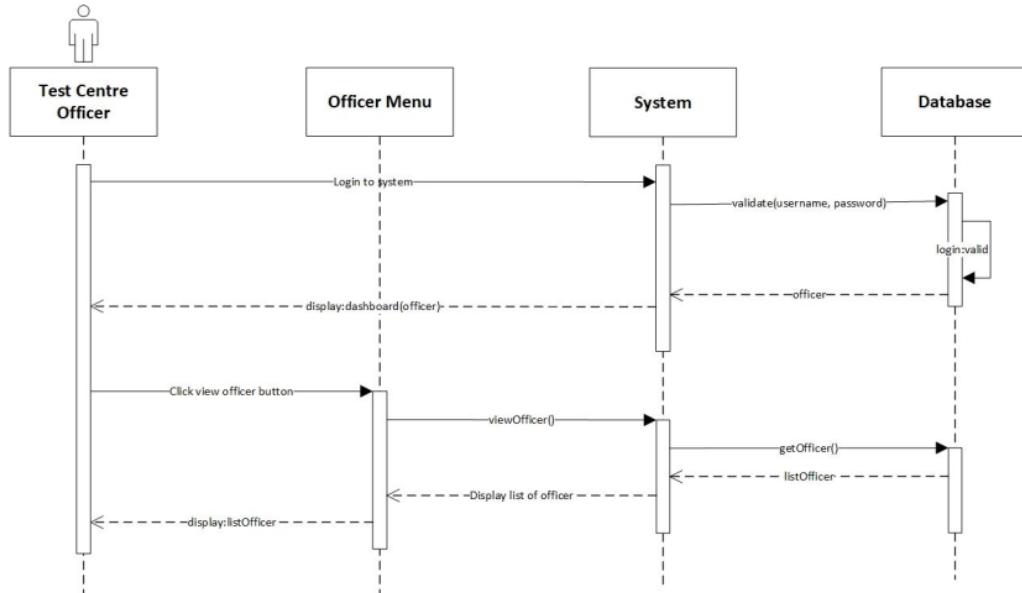


Figure 10. View Test Centre Officer System Sequence Diagram

- Contract

Cross References	View Test Centre Officer
Operation	login(username, password)
Responsible	To access the officer's dashboard.
Pre-conditions	<ul style="list-style-type: none"> <li>• Officer object must be available</li> <li>• A valid username and password</li> </ul>
Post-conditions	<ul style="list-style-type: none"> <li>• Session was created (instance creation)</li> <li>• Session was associated with the Officer (association formed)</li> </ul>
Cross References	View Test Centre Officer
Operation	viewOfficer()
Responsible	To view a list of test centre data.
Pre-conditions	<ul style="list-style-type: none"> <li>• Officer object must be available</li> <li>• Session must already be created</li> </ul>
Post-conditions	<ul style="list-style-type: none"> <li>• ListOfficer was created (instance creation)</li> </ul>

	<ul style="list-style-type: none"> <li>• ListOfficer was associated with viewMenu (association formed)</li> </ul>
--	---

#### 4.6. Record Tester

**Prepared by: I Made Siva Aditya Surya**

- Expanded Use Case

Use Case	Record Tester
<b>Goal in Context</b>	Allow Manager to record Officer as Tester
<b>Primary Actor</b>	Test Centre Manager
<b>Secondary Actor</b>	Test Centre Officer
<b>Typical Course of Events</b>	
Actor Actions	System Response
1. Process begins after Test Centre Manager successfully login to system	2. System will navigate Test Centre Manager to their dashboard
3. Test Centre Manager clicks test centre officer menu	4. System will display a list of test centre data
5. Test Centre Manager clicks edit button on the test centre officer whose position is going to be changed to tester	6. System will navigate Test Centre to the test centre officer edit page and display the test centre officer data form
7. Test Centre Manager changes the position of test centre officer from "officer" to "tester"	8. System will store the data that has been entered by the Test Centre Manager into the database then the Test Centre Manager will be navigated to the test centre officer page and a message appears that notify success in recording tester is displayed

### **Alternative Course**

**5a:** If there are no test centre officer that has been recorded in database or test centre manager wants to record new test centre officer with position “tester”, then the test centre manager will click add centre officer button and fill in all the required data in the form as well as setting the position to “tester”. Test centre manager will click the submit button to record the tester into database, then a message that notify success in recording tester is displayed

- System Sequence Diagram

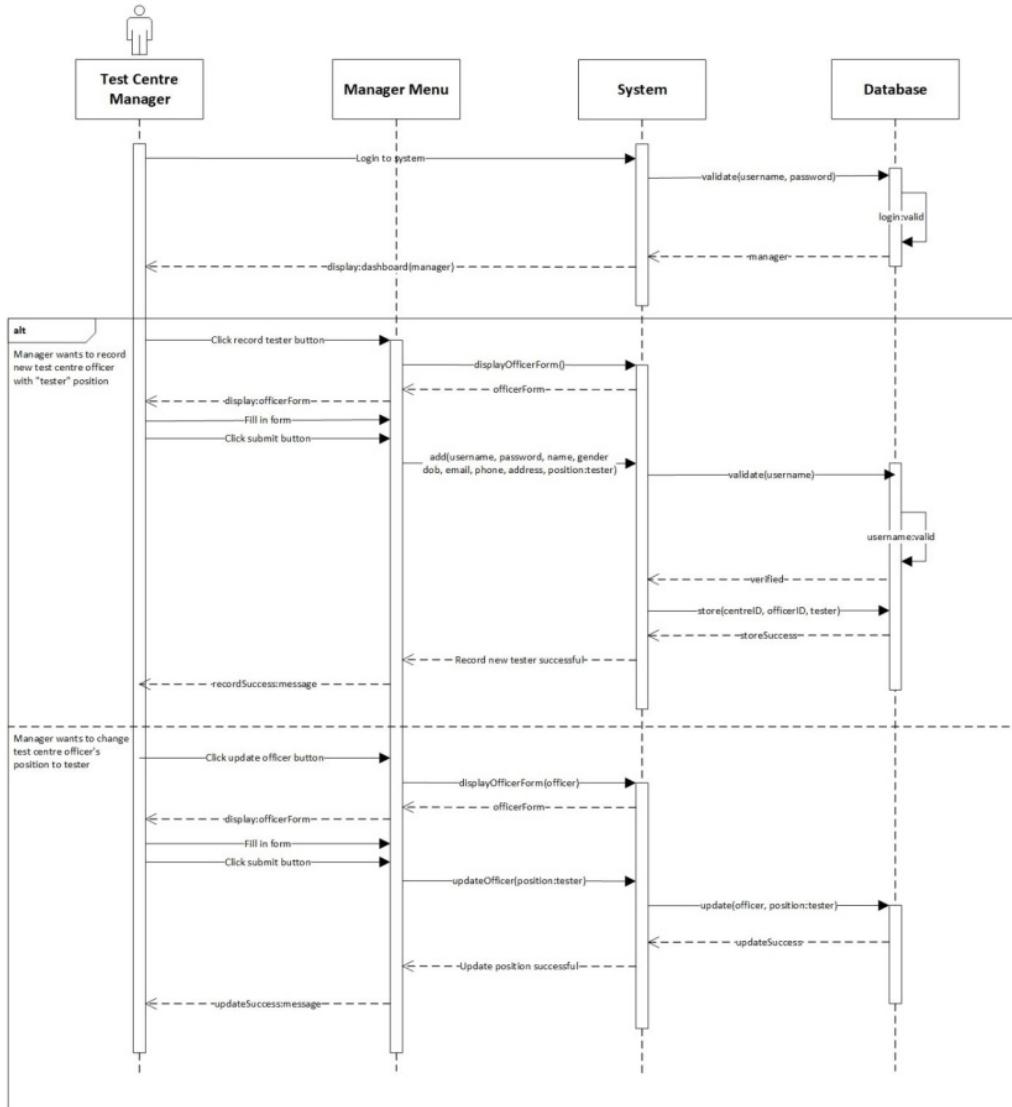


Figure 11. Record Tester System Sequence Diagram

- Contract

<b>Cross References</b>	<b>Record Tester</b>
Operation	login(username, password)
Responsible	To access the manager's dashboard.
Pre-conditions	<ul style="list-style-type: none"> <li>• Manager object must be available</li> <li>• A valid username and password</li> </ul>
Post-conditions	<ul style="list-style-type: none"> <li>• Session was created (instance creation)</li> <li>• Session was associated with the Manager (association formed)</li> </ul>
<b>Cross References</b>	<b>Record Tester</b>
Operation	updateOfficer(position:tester)
Responsible	To update the position of test centre officer to tester.
Pre-conditions	<ul style="list-style-type: none"> <li>• Manager object must be available</li> <li>• Session must already be created</li> <li>• Officer object must be available</li> </ul>
Post-conditions	Position was updated (attribute modification)
<b>Cross References</b>	<b>Record Tester</b>
Operation	add(username, password, name, gender, dob, email, phone, address, position:tester)
Responsible	To record Tester
Pre-Conditions	<ul style="list-style-type: none"> <li>• Manager object must be available</li> <li>• Session must already be created</li> <li>• TestCentre object must be available</li> <li>• Tester data must be available</li> <li>• A valid tester username</li> </ul>
Post-Conditions	<ul style="list-style-type: none"> <li>• Tester was created (instance creation)</li> <li>• OfficerID was generated (attribute modification)</li> <li>• Tester was associated with the TestCentre (association formed)</li> </ul>

#### 4.7. Record Officer

**Prepared by: I Made Siva Aditya Surya**

- Expanded Use Case

Use Case	Record Officer
<b>Goal in Context</b>	Allow Manager to record new Officer
<b>Primary Actor</b>	Test Centre Manager
<b>Secondary Actor</b>	Test Centre Officer
<b>Typical Course of Events</b>	
Actor Actions	System Response
1. Process begins after Test Centre Manager successfully login to system	2. System will navigate Test Centre Manager to their dashboard
3. Test Centre Manager clicks test centre officer menu	4. System will display a list of test centre officer data
5. Test Centre Manager clicks add test centre officer button	6. System will navigate Test Centre Manager to add test centre officer page and display test centre officer data form
7. Test Centre Manager fills in the data for username, password, and other test centre officer data on the form provided and then clicks the submit button	8. System will store the data that has been entered by Test Centre Manager into the database then Test Centre Manager will be navigated to the test centre officer page and a message that notifies success in recording test centre officer is displayed
<b>Alternative Course</b>	
-	

- System Sequence Diagram

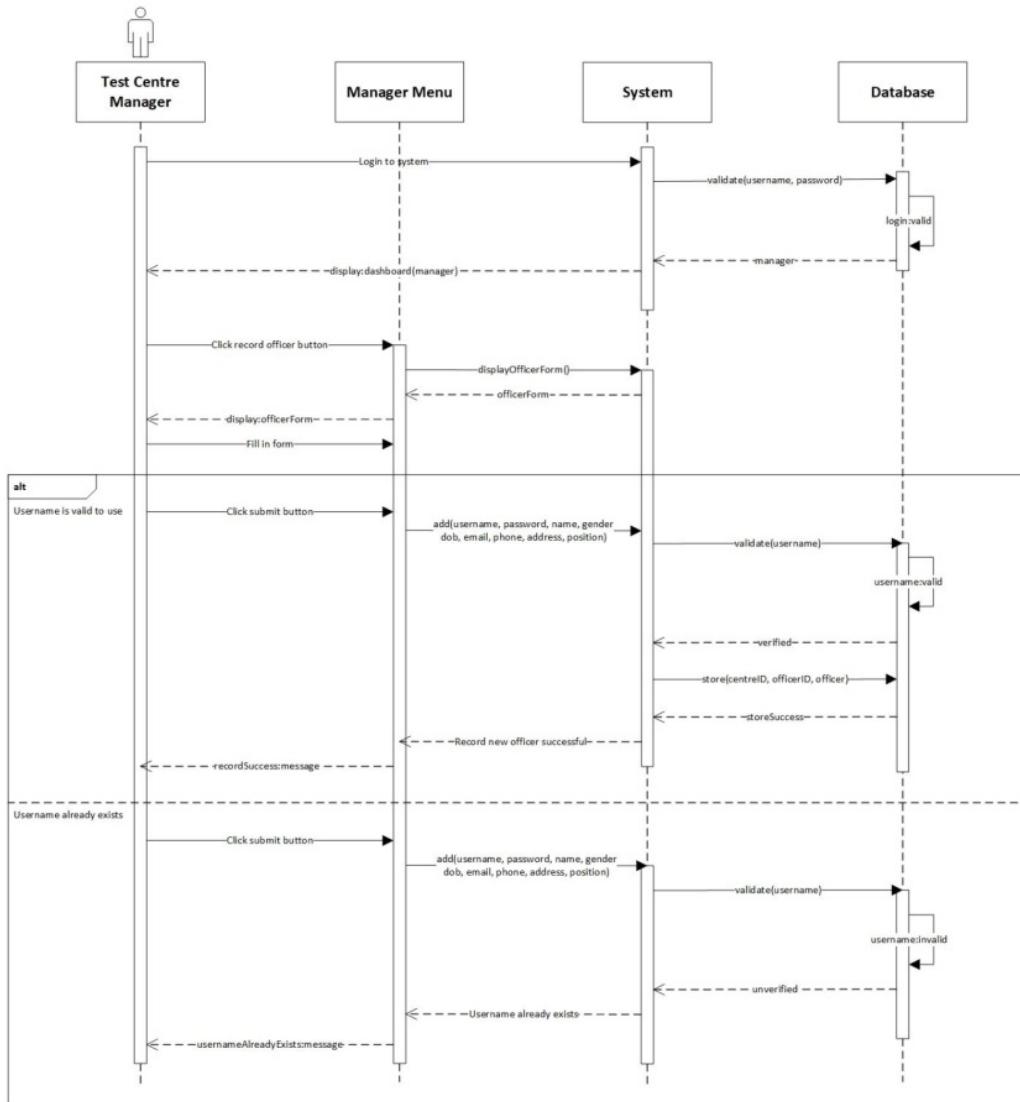


Figure 12. Record Officer System Sequence Diagram

- Contract

<b>Cross References</b>	<b>Record Officer</b>
Operation	login(username, password)
Responsible	To access the manager's dashboard.
Pre-conditions	<ul style="list-style-type: none"> <li>• Manager object must be available</li> <li>• A valid username and password</li> </ul>
Post-conditions	<ul style="list-style-type: none"> <li>• Session was created (instance creation)</li> <li>• Session was associated with the Manager (association formed)</li> </ul>
<b>Cross References</b>	<b>Record Officer</b>
Operation	add(username, password, name, gender, dob, email, phone, address, position)
Responsible	To record Officer
Pre-conditions	<ul style="list-style-type: none"> <li>• Manager object must be available</li> <li>• Session must already be created</li> <li>• TestCentre object must be available</li> <li>• Officer data must be available</li> <li>• A valid officer username</li> </ul>
Post-conditions	<ul style="list-style-type: none"> <li>• Officer was created (instance creation)</li> <li>• OfficerID was generated (attribute modification)</li> <li>• Officer was associated with the TestCentre (associate modification)</li> </ul>
<b>Cross References</b>	<b>Record Officer</b>
Operation	add(username, password, name, gender, dob, email, phone, address, position)
Responsible	To record Officer
Pre-conditions	<ul style="list-style-type: none"> <li>• Manager object must be available</li> <li>• Session must already be created</li> <li>• TestCentre object must be available</li> <li>• Officer data must be available</li> </ul>

Post-conditions	<ul style="list-style-type: none"> <li>Officer was not created</li> </ul>
-----------------	---

#### 4.8. Record New Test

Prepared by: I Made Siva Aditya Surya

- Expanded Use Case

Use Case	Record New Test
Goal in Context	Allow Tester to record Patient and their test data
Primary Actor	Tester
Secondary Actor	Patient
Typical Course of Events	
Actor Actions	System Response
1. Process begins after Tester successfully login to system	2. System will navigate Tester to their dashboard
3. Tester clicks Patient Test menu	4. System will display a list of Patient Test data
5. Tester clicks add new test button	6. System will navigate Tester to add new test page and display patient data and test data form
7. Tester fills in the patient username, password, and other patient data, as well as test-related data which includes patient type and symptoms in the form provided and then clicks the submit button	8. System will store the data that has been entered by Tester into the database then the Tester will be directed to the Patient Test page and a message that notifies success in recording new test is displayed
Alternative Course	
5a: If patient has already done the test before, then Tester only needs to enter patient type and symptoms into the form provided	

- System Sequence Diagram

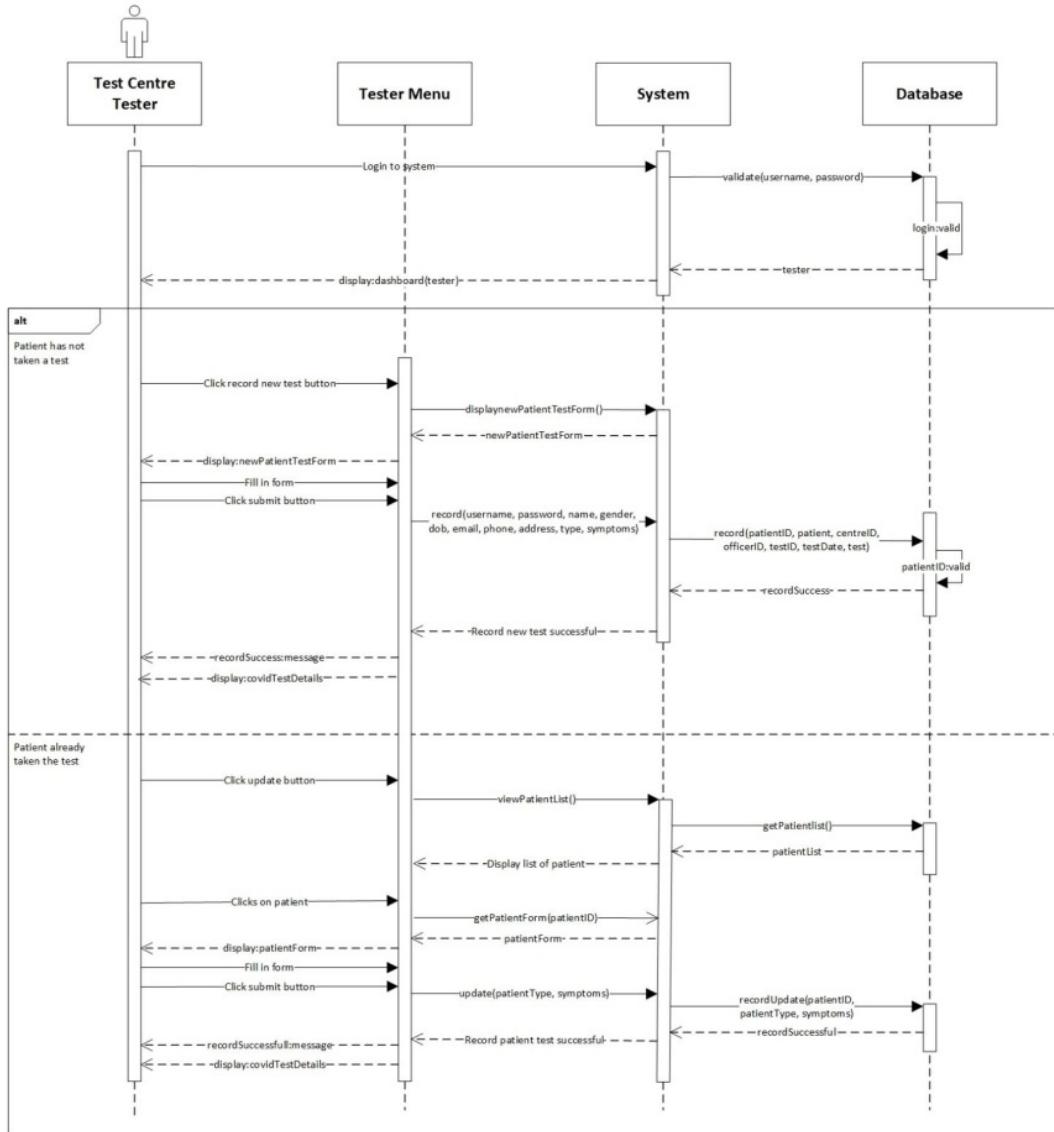


Figure 13. Record New Test System Sequence Diagram

- Contract

Cross References	Record New Test
Operation	login(username, password)
Responsible	To access the tester's dashboard.
Pre-conditions	<ul style="list-style-type: none"> <li>• Tester object must be available</li> <li>• A valid username and password</li> </ul>
Post-conditions	<ul style="list-style-type: none"> <li>• Session was created (instance creation)</li> <li>• Session was associated with the Tester (association formed)</li> </ul>
Cross References	Record New Test
Operation	record(username, password, name, gender, dob, email, phones, address, type, symptoms)
Responsible	To record patient data if patient has not taken the test.
Pre-conditions	<ul style="list-style-type: none"> <li>• Tester object must be available</li> <li>• Session must already be created</li> <li>• TestKit object must be available</li> <li>• Patient data is available</li> <li>• A valid patient username</li> </ul>
Post-conditions	<ul style="list-style-type: none"> <li>• Patient was created (instance creation)</li> <li>• PatientID was generated (attribute modification)</li> <li>• CovidTest was created (instance creation)</li> <li>• testID was generated (attribute modification)</li> <li>• testDate was generated (attribute modification)</li> <li>• Status was generated (attribute modification)</li> <li>• CovidTest was associated with the Tester (association formed)</li> <li>• CovidTest was associated with the TestKit (association formed)</li> <li>• CovidTest was associated with the Patient (association formed)</li> </ul>
Cross References	Record New Test

Operation	update(patientType, symptoms)
Responsible	To record test data after patient has taken the test.
Pre-conditions	<ul style="list-style-type: none"> <li>Tester object must be available</li> <li>Session must already be created</li> <li>CovidTest object is available</li> <li>Patient data is available</li> <li>Test data is available</li> </ul>
Post-conditions	<ul style="list-style-type: none"> <li>PatientType was updated (attribute modification)</li> <li>Symptoms was updated (attribute modification)</li> </ul>

#### 4.9. Update Test Result

Prepared by: I Made Siva Aditya Surya

- Expanded Use Case

Use Case	Update Test Result
Goal in Context	Allow Tester to update Patient's test result and status
Primary Actor	Tester
Secondary Actor	-
Typical Course of Events	
Actor Actions	System Response
1. Process begins after Tester successfully login to system	2. System will navigate Tester to their dashboard
3. Tester clicks Patient Test menu	4. System will display a list of patient test data
5. Tester clicks the edit button on the test patient data that needs to be updated	6. System will navigate Tester to edit test patient page and display patient test data form
7. Tester updates patient status data and result	8. System will store the data that has been entered by Tester into the database then

	Tester will be navigated to the Patient Test page and a message that notifies success in updating patient test data is displayed
<b>Alternative Course</b>	
-	

- System Sequence Diagram

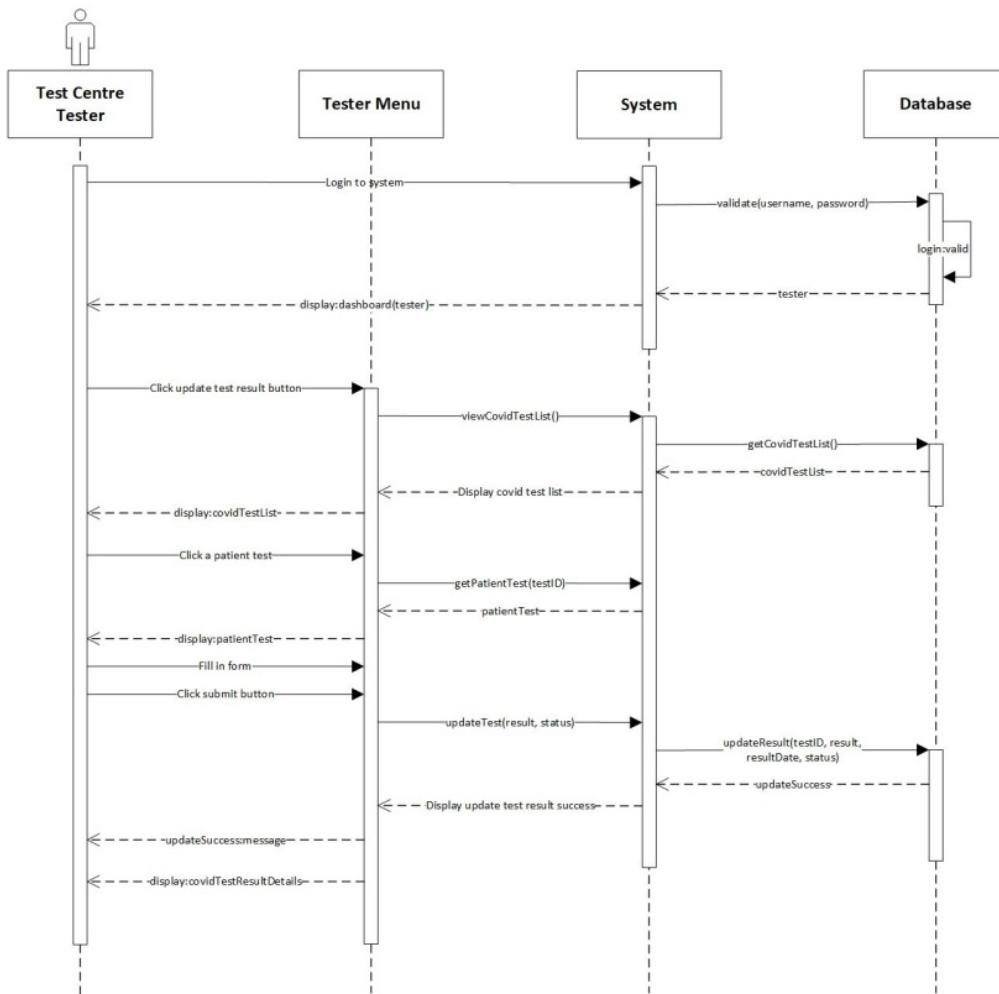


Figure 14. Update Test Result System Sequence Diagram

- Contract

Cross References	Update Test Result
Operation	login(username, password)
Responsible	To access the tester's dashboard.
Pre-conditions	<ul style="list-style-type: none"> <li>• Tester object must be available</li> <li>• A valid username and password</li> </ul>
Post-conditions	<ul style="list-style-type: none"> <li>• Session was created (instance creation)</li> <li>• Session was associated with the Tester (association formed)</li> </ul>
Cross References	Update Test Result
Operation	updateTest(result, status)
Responsible	To update the test result and status.
Pre-conditions	<ul style="list-style-type: none"> <li>• Tester object must be available</li> <li>• Session must already be created</li> <li>• CovidTest object must be available</li> <li>• Patient data must be available</li> <li>• Test result data must be available</li> </ul>
Post-conditions	<ul style="list-style-type: none"> <li>• Result was updated (attribute modification)</li> <li>• ResultDate was updated (attribute modification)</li> <li>• Status was updated (attribute modification)</li> </ul>

#### 4.10. Generate Test Report

Prepared by: I Made Siva Aditya Surya

- Expanded Use Case

Use Case	Generate Test Report
Goal in Context	Allow Officer to generate a list of test report that are administered in their Test Centre
Primary Actor	Test Centre Officer
Secondary Actor	-

### Typical Course of Events

Actor Actions	System Response
<b>1. Process begins after Test Centre Officer successfully login to system</b>	2. System will navigate Test Centre Officer to their homepage
<b>3. Test Centre Officer clicks Test Report menu</b>	4. System will display a list of all patient test report on the test centre which the officer is employed at
<b>Alternative Course</b>	
-	

- System Sequence Diagram

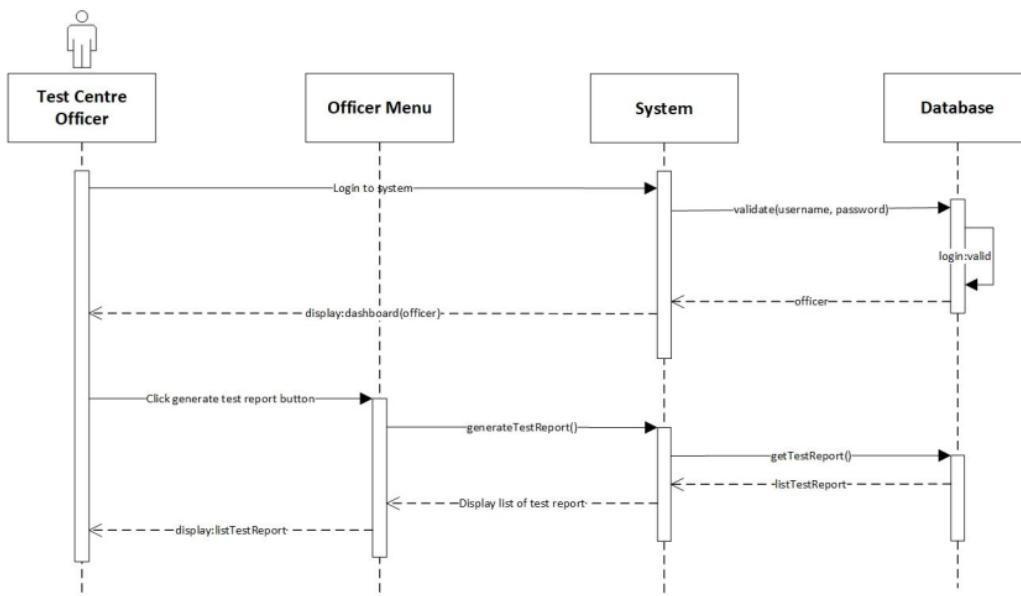


Figure 15. Generate Test Report System Sequence Diagram

- Contract

Cross References	Generate Test Report
Operation	login(username, password)
Responsible	To access the officer's dashboard.
Pre-conditions	<ul style="list-style-type: none"> <li>• Officer object must be available</li> <li>• A valid username and password</li> </ul>
Post-conditions	<ul style="list-style-type: none"> <li>• Session was created (instance creation)</li> <li>• Session was associated with the Officer (association formed)</li> </ul>
Cross References	Generate Test Report
Operation	generateTestReport()
Responsible	To view a list of all test report data.
Pre-conditions	<ul style="list-style-type: none"> <li>• Officer object must be available</li> <li>• Session must already be created</li> <li>• CovidTest object must be available</li> </ul>
Post-conditions	<ul style="list-style-type: none"> <li>• ListTestReport was created (instance creation)</li> <li>• ListTestReport was associated with viewMenu (association formed)</li> </ul>

#### 4.11. View Testing History

**Prepared by: I Made Siva Aditya Surya**

- Expanded Use Case

Use Case	View Testing History
Goal in Context	Allow Patient to view their testing history
Primary Actor	Test Centre Officer
Secondary Actor	-
<b>Typical Course of Events</b>	

Actor Actions	System Response
5. Process begins after Test Centre Officer has successfully login to system	6. System will navigate Test Centre Officer to their dashboard
7. Test Centre Officer clicks test centre officer menu	8. System will display a list of test centre officer data
<b>Alternative Course</b>	
-	

- System Sequence Diagram

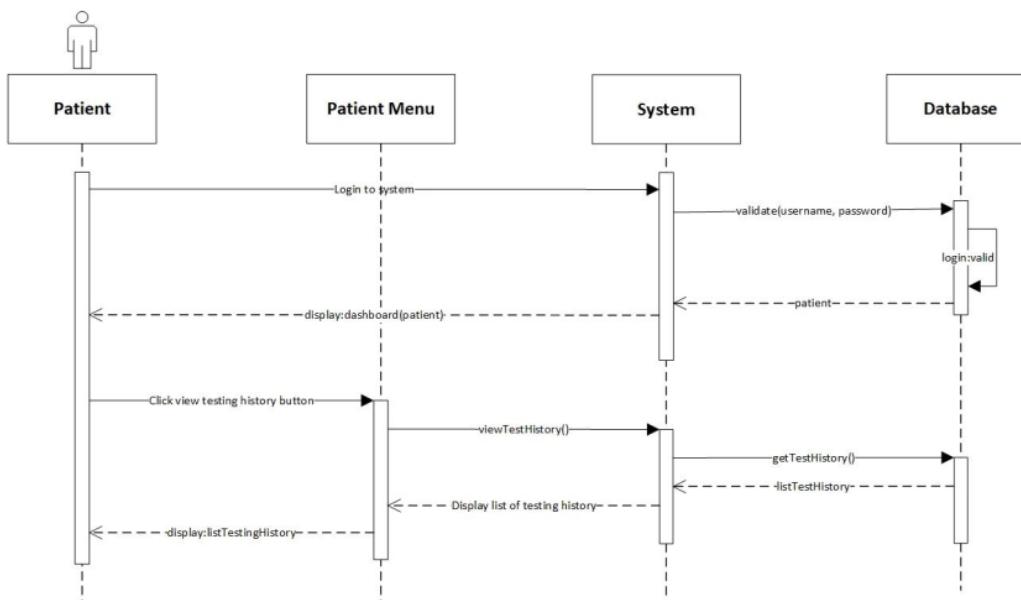


Figure 16. View Testing History System Sequence Diagram

- Contract

Cross References	View Testing History
Operation	login(username, password)
Responsible	To access the patient's dashboard.
Pre-conditions	<ul style="list-style-type: none"> <li>• Patient object must be available</li> <li>• A valid username and password</li> </ul>

Post-conditions	<ul style="list-style-type: none"> <li>Session was created (instance creation)</li> <li>Session was associated with the Patient (association formed)</li> </ul>
<b>Cross References</b>	<b>View Testing History</b>
Operation	viewTestHistory()
Responsible	To view a list of testing report history.
Pre-conditions	<ul style="list-style-type: none"> <li>Patient object must be available</li> <li>Session must already be created</li> <li>CovidTest object must be available</li> </ul>
Post-conditions	<ul style="list-style-type: none"> <li>ListTestHistory was created (instance creation)</li> <li>ListTestHistory was associated with viewMenu (association formed)</li> </ul>

#### 4.12. Log Out

**Prepared by: Ida Bagus Gede Awidya Andika**

- Expanded Use Case

Use Case	Log Out
Goal in Context	Allow User to log out from the system
Primary Actor	Test Centre Officer and Patient
Secondary Actor	-
<b>Typical Course of Events</b>	
Actor Actions	System Response
1. Process begins after Test Centre Officer/Patient has done everything they want to do on the system and wants to log out	
2. Test Centre Officer/Patient clicks logout button	3. System will display a message questioning whether Test Centre Officer/Patient really wants to log out from system

- |  |  |
|--|--|
| <p><b>4. Test Centre Officer/Patient clicks Yes on the message</b></p> | <p><b>5. System will end the login session of the Test Centre Officer/Patient and redirect to login page</b></p> |
|--|--|

**Alternative Course**

- 4a: If the officer/patient clicks on the No button, the log out process will be canceled**

- System Sequence Diagram

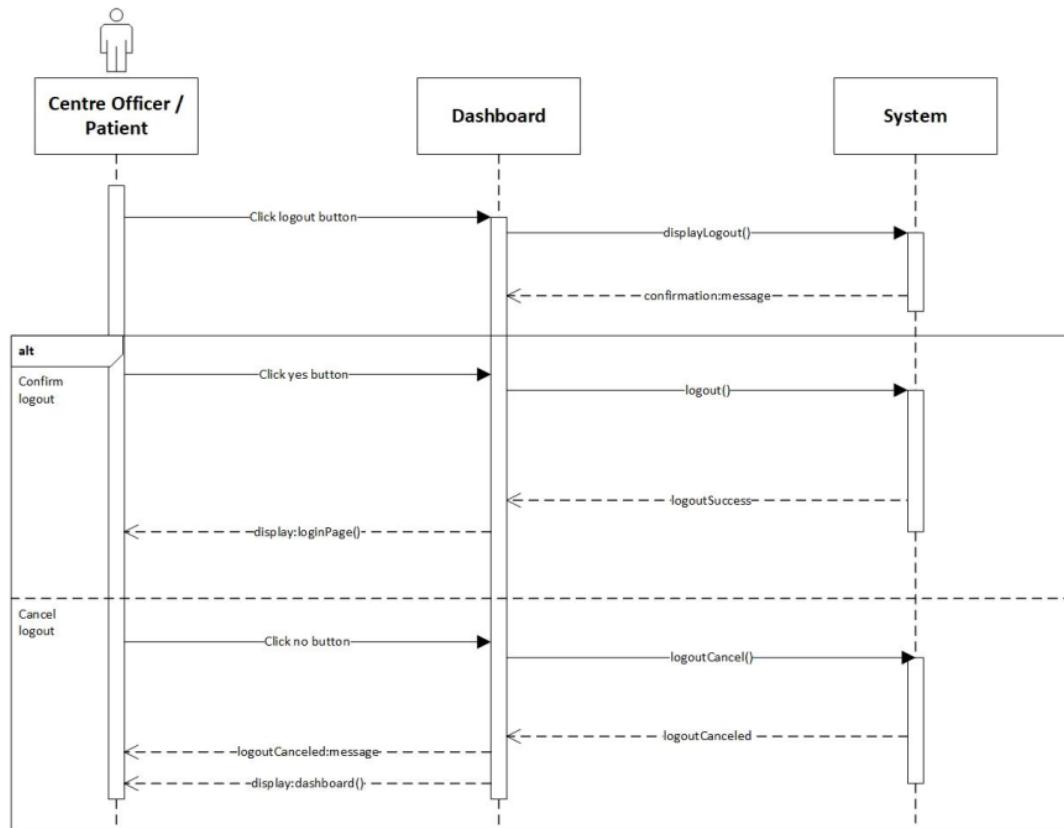


Figure 17. Log Out System Sequence Diagram

- Contract

<b>Cross References</b>	<b>Logout</b>
Operation	logout()
Responsible	To log out or exit from the system
Pre-conditions	<ul style="list-style-type: none"> <li>• User object must be available</li> <li>• Session must already be created</li> <li>• Confirm to logout</li> </ul>
Post-conditions	Session was destroyed (instance deletion)

<b>Cross References</b>	<b>Logout</b>
Operation	logoutCancel()
Responsible	To cancel log out or exit from the system
Pre-conditions	<ul style="list-style-type: none"> <li>• User object must be available</li> <li>• Session must already be created</li> <li>• Confirm to cancel logout</li> </ul>
Post-conditions	Session was not destroyed

## 5. Task Division

Required Behaviour	Use Case	Project Members
PHP script that enables Officer and Patient to login to the system	Login	Awidya Andika
PHP script that enables Officer to view a list of test kit and stock	View Test Kit	Awidya Andika
PHP script that enables Manager to add new test kit and update stock	Manage Test Kit Stock	Awidya Andika
PHP script that enables Test Centre to be created and recorded for the Manager	Register Test Centre	Awidya Andika
PHP script that enables Officer to view a list of all employed officer in their Test Centre	View Test Centre Officer	Awidya Andika
PHP script that enables Manager to record Officer as Tester	Record Tester	Siva Aditya
PHP script that enables Manager to record new Officer	Record Officer	Siva Aditya
PHP script that enables Tester to record Patient and their test data	Record New Test	Siva Aditya
PHP script that enables Tester to update Patient's test result and status	Update Test Result	Siva Aditya
PHP script that enables Officer to generate a list of test report that are administered in their Test Centre	Generate Test Report	Siva Aditya
PHP script that enables Patient to view their testing history	View Testing History	Siva Aditya
PHP script Officer and Patient to log out from the system	Log Out	Awidya Andika
HTML, CSS, and JavaScript to design the website menus and pages	All	Siva Aditya & Awidya Andika
Setting up MySQL database	All	Siva Aditya & Awidya Andika

## Bibliography

# Assignment 1 (E1800176 & E1800182)

---

## ORIGINALITY REPORT

---

**5%**

SIMILARITY INDEX

**5%**

INTERNET SOURCES

**1%**

PUBLICATIONS

**%**

STUDENT PAPERS

---

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

---

2%

★ [www.scribd.com](http://www.scribd.com)

Internet Source

---

Exclude quotes

Off

Exclude matches

Off

Exclude bibliography

Off