FIT3164 Data Science Project Part 2

Test Report

Automated Health Information System

Team MDS2

Foo Kai Yan | 33085625 | kfoo0012@student.monash.edu

Alicia Quek Chik Wen | 33045240 | aque0004@student.monash.edu

Eunice Lee Wen Jing | 33250979 | elee0075@student.monash.edu

Jesse Yow San Gene | 32794649 | jyow0001@student.monash.edu

Supervised by:

Dr Muhammad Fermi Pasha

Word Count: 8058

Date: 11th October 2024

Table of Contents

1. Introduction

The Automated Health Information System (AHIS) project is a web-based health information system that aims to enhance data management by utilizing advanced automated data input technologies. This approach reduces the need for manual input from healthcare staff, particularly nurses. The web-based system incorporates an Image Cropping (ICROP) model and a Handwritten Text Recognition (HTR) model to improve the accuracy and efficiency of patient data entry during registration. By incorporating the ICROP model, the system achieves greater precision in recognizing the handwritten text, which significantly improves the performance of the HTR model.

The system enables users to upload handwritten patient registration forms, which are first processed by the ICROP model that cropped the form into individual form fields. Each field image is then analyzed by the HTR model to extract the handwritten text. The extracted text is then displayed on a digital form for users to review and make edits if necessary before submission, as adjustments may be needed due to potential inaccuracies in the HTR model's output.

The system is composed of four main software components: the ICROP model, HTR model, frontend and backend of the web-based system. Therefore, software testing is essential in the development process, as it helps detect bugs and errors that could affect the performance of the web application. Detecting and resolving these issues enables the team to continuously improve and optimize the web application. With that, this report outlines the testing process and results, including methodologies such as black-box, integration, and usability testing, to ensure that the application meets quality standards.

2. Testing Plan

In the previous semester, the team gathered to discuss and document the project requirements in a Requirements Traceability Matrix (RTM). However during the implementation phase, some requirements proved to be impractical or unfeasible. Hence, adjustments were made to the RTM as shown in Table 1 below.

Req. ID	Requirement Description	Category	Requirement Type	Source	Completion Status	Tests Conducted
1	External devices to capture form images	Feature	Functional	Dr. Fermi	Completed	4.1.2.
2	Scalable database design to store date	Feature	Non-Functional	Dr. Fermi	Completed	All
3	Data privacy and confidentiality	Security	Non-Functional	Dr. Fermi	Completed	4.1.1.

4	Integrated CROP and HTR model to web app	Feature	Functional	Dr. Fermi	Completed	5.
5	User Authentication	Security	Functional	Dr. Fermi	Completed	4.1.1.
6	Profile Management	Feature	Functional	Dr. Fermi	Completed	4.1.1.
7	Contains required features for a complete healthcare web app	Feature	Functional	Dr. Fermi	Completed	All
8	Appointment and Encounter Feature	Feature	Functional	Dr. Fermi	Completed	4.1.3.
9	Review patient past medical records	Feature	Functional	Dr. Fermi	Completed	4.1.2.
10	Results displayed on web app is easily to comprehend	Quality	Non-Functional	Dr. Fermi	Completed	6.
11	Satisfaction on the HTR model accuracy	Perform ance	Non-Functional	Dr. Fermi	Completed	4.2.1. - 4.2.8.
12	User Friendly User Interface	Quality	Non-Functional	Dr. Fermi	Completed	6.

Table 2.1. Requirement Traceability Matrix (RTM)

The final column on the left displays the testing done for each requirement, with further elaboration on test procedure provided in upcoming sections.

3. Testing Approach Methodology

The models and web application sections are the two primary divisions, which the entire project is divided into. The web application can also be further separated into two sections, which are the frontend and the backend. It has been planned to conduct 3 different testing approaches on the whole project from the web application to the ICROP and HTR models as arranged in Table 2 below.

Testing Approach	Tests Conducted	Description
Black-Box Testing	State Transition	State transition testing validates the behaviour of the web application based

		on how one page of the web application navigates to another page of the web application.
	Equivalence Partitioning	Equivalence partitioning will be conducted to divide the input data into similar classes, reducing the amount of test cases and enhancing testing efficiency.
	Boundary Value Analysis	Boundary value analysis testing will be done to test the edges of input ranges where errors most likely will occur at.
	Negative Testing	Negative testing is conducted to ensure the web application handles invalid, unexpected, and unusual inputs correctly.
	Performance Testing	Performance testing is done to evaluate how well the web application performs under certain possible scenarios.
	Compatibility Testing	Compatibility testing is done for the HTR model in which different pen configurations and environments will be tested out to ensure results are produced appropriately.
	Functional Testing	Functional testing is done to ensure that the HTR model is functioning as intended under different environments such as italics and/or cursive texts.
	Stress Testing	Stress testing is done to assess the stability of the HTR model under extreme conditions such as with the inclusion of symbols and punctuation.
Integration Testing	The core project pipeline is tested, from patient registration using the smart data entry functionality in the	Integration testing is done during the integration of both ICROP and HTR models into the web application. This specific testing is performed mostly to

	“Register patient” web page till the new patient details has been saved into the database.	ensure that no problems or issues arise when people use the web application.
Usability Testing	Manual testing involves approaching and asking three anonymous individuals of the public to test out the web application. The results and input from these individuals will be gathered and analyzed to identify potential project improvements.	Usability testing is done by allowing end-users to test the web application to obtain conclusive evaluations on the web application. This specific testing is conducted mainly for the team to identify user experience issues and judge the product satisfactions from end-users.

Table 3.1. Test Approach Methodology

4. Black-Box Testing

4.1. Test 1 : Web Application

Black box testing was conducted on three key features of the web application, including **User Login**, **Patient Registration**, and **the flow of Appointment Scheduling, Diagnosis, and Prescription**. Black box testing is a form of testing performed without knowledge of a system’s internal structure (What Is Black Box Testing?, n.d.). This testing method helps evaluate an application’s functionality, security, and performance by focusing on the expected output of given inputs (What Is Black Box Testing?, n.d.). Different black box testing methods were chosen for each feature based on their input types and the desired testing outcomes.

The following table provides an overview of the features tested, the corresponding black box testing methods:

Feature	Testing Method	Description
User Login	Equivalence Partitioning	Used to classify the email and password fields into equivalence classes (valid, invalid, empty).
Register New Patient	Boundary Value Analysis	Focus on testing the boundaries of input fields.
Appointment Scheduling, Diagnosis, and	State Transition	Evaluate the system's behavior as users navigate through various states in the appointment and diagnosis processes.

Prescription System		

Table 4.1 Web Application Black Box Testing Methods

4.1.1. User Login Feature

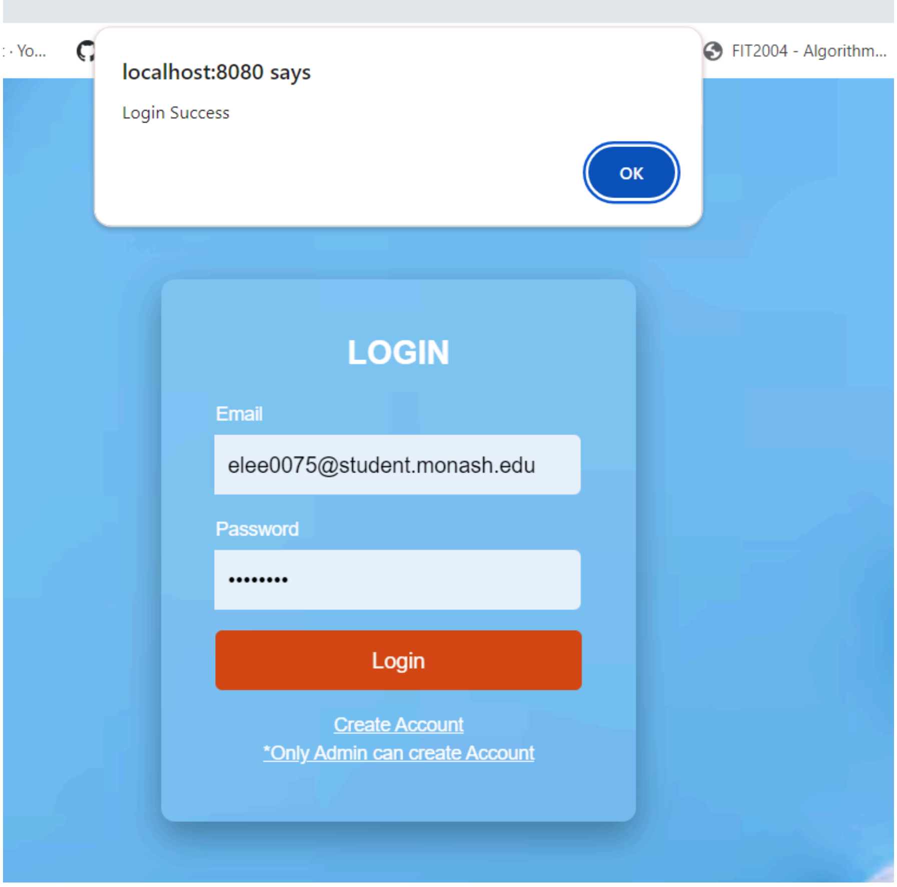
The User Login feature of the web application was tested using equivalence partitioning, a method that categorizes input combinations for the email and password fields into three distinct equivalence classes: valid inputs, invalid inputs, and empty inputs. The objective of this testing is to verify that the web application grants access only to users with valid credentials while appropriately handling invalid or missing inputs by displaying relevant error messages.

By categorizing the inputs into equivalence classes, the testing aims to cover all possible input scenarios, including entering a valid email with an incorrect password, leaving either the email or password field empty, or entering invalid data. These tests ensure that users are granted access only when both the email and password are valid, and that any invalid input triggers a corresponding error message.

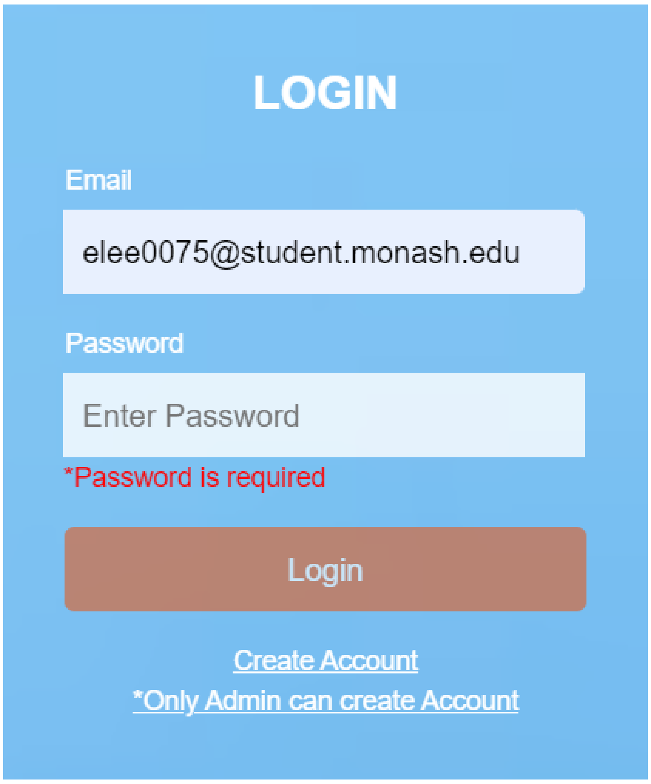
When a user provides valid credentials, they are successfully logged in and redirected to the main dashboard of the application. However, if the provided credentials are invalid or the input fields are left empty, an error message is displayed, prompting the user to re-enter the correct information.

- **Valid Input**: Both email and password exist in the system and are correctly formatted.
- **Invalid Input**: Either the email or password is incorrect or does not exist.
- **Empty Input**: Either the email or password field (or both) is left empty.

Test ID	Test Description	Expected Result	Actual Result	Status
B01	Valid email and valid password	User is successfully logged in, redirected to the dashboard.	Successfully logged in and redirected to the main page.	PASS

<table>
<tr><td colspan="5">
localhost:8080 says
Login Success

Email: elee0075@student.monash.edu
Password: ••••••••
Figure 4.1.1.1 The screenshot above shows a message "Login Success" with an "OK" button after the user enters the correct credentials. The user can proceed to the main page by clicking the "OK" button.</td></tr>
<tr><td>B02</td><td>Valid email and invalid password</td><td>Error message: "Login Failed! Incorrect password or Invalid email! Try Again!". User must re-enter the correct password.</td><td>Error message displayed. User prompted to re-enter credentials.</td><td>PASS</td></tr>
<tr><td>B03</td><td>Valid email and empty password</td><td>Error message shown below the password field: "Password is required". User must enter the missing password.</td><td>Error message displayed. User prompted to re-enter credentials.</td><td>PASS</td></tr>
</table>



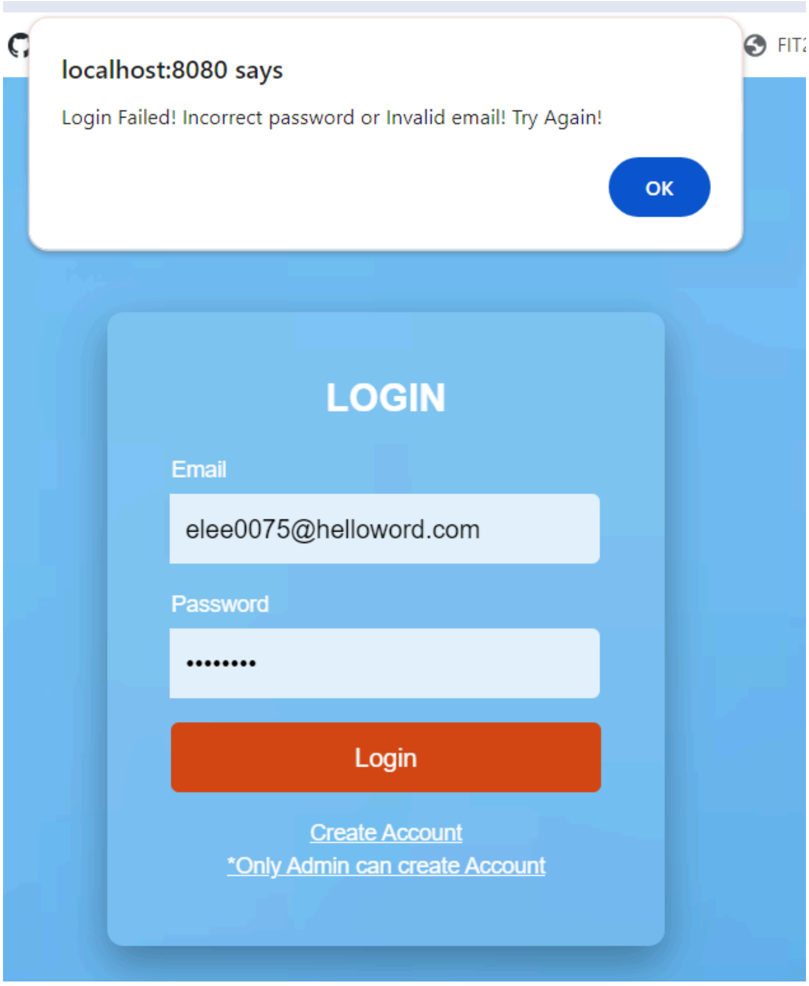
elee0075@student.monash.edu

Enter Password

*Password is required

Figure 4.1.1.2 The error message "Password is required" is shown when the user leaves the password field empty.

B04	Invalid email and valid password	Error message: "Login Failed! Incorrect password or Invalid email! Try Again!". User must re-enter the correct email.	Error message displayed. User prompted to re-enter credentials.	PASS



localhost:8080 says

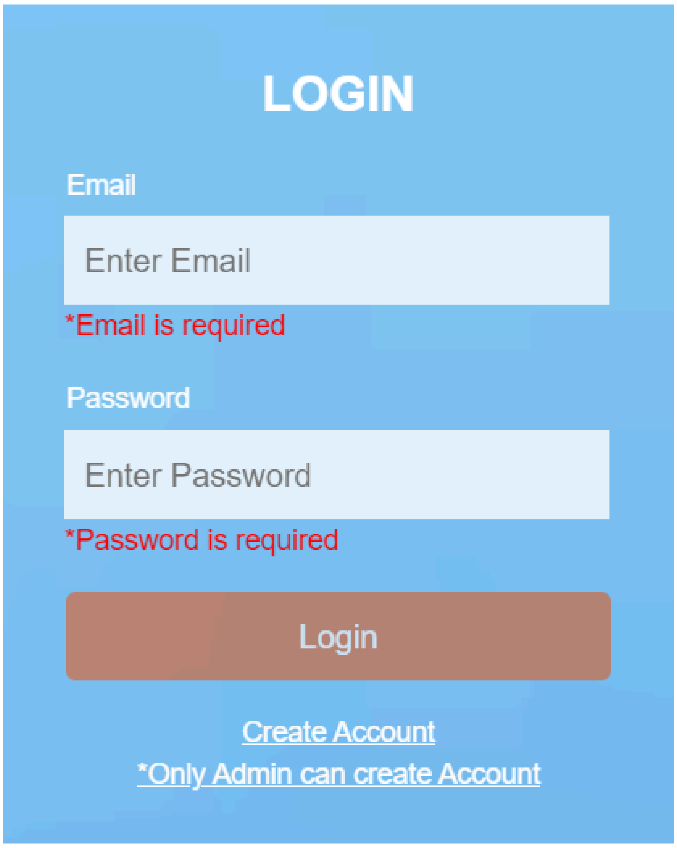
Login Failed! Incorrect password or Invalid email! Try Again!

Email: elee0075@helloword.com

Password: ••••••••

Figure 4.1.1.3 The error message "Login Failed! Incorrect password or Invalid email! Try Again!" is shown when the user enters an incorrect email.

B05	Invalid email and invalid password	Error message: "Login Failed! Incorrect password or Invalid email! Try Again!". User must re-enter the correct credentials.	Error message displayed. User prompted to re-enter credentials.	PASS
B06	Invalid email and empty password	Error message: "Password is required". User must re-enter both fields.	Error message displayed. User prompted to re-enter credentials.	PASS
B07	Empty email and valid password	Error message: "Email is required". Users must re-enter both fields.	Error message displayed. User prompted to re-enter credentials.	PASS

B08	Empty email and invalid password	Error message: "Email is required". Users must re-enter both fields.	Error message displayed. User prompted to re-enter credentials.	PASS
B09	Empty email and empty password	Error message: "Email is required" and "Password is required". Users must enter both fields.	Error message displayed. Users are prompted to re-enter both fields.	PASS
 *Email is required *Password is required LOGIN Email Enter Email Password Enter Password Login Create Account *Only Admin can create Account Figure 4.1.1.4 The error messages "Email is required" and "Password is required" are shown below the fields when the user leaves both fields empty.				

4.1.2. New Patient Registration Feature

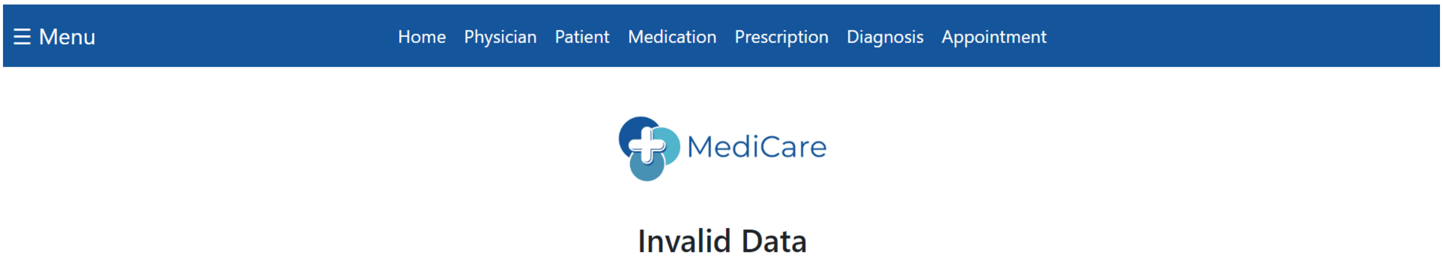
The New Patient Registration feature was tested using Boundary Value Testing. This testing approach ensures that the system behaves correctly at the critical limits of input fields. This technique provides efficient and effective validation for edge cases where the system is most likely to encounter input-related errors.

We examined key input fields such as patient names, nationality, contact number, and identification number to ensure the system accepts only valid characters and enforce appropriate field constraints.

- **Patient Name**: The name field must contain only word characters and spaces.
- **Nationality**: The nationality field must also accept only word characters and spaces. The invalid input characters such as numbers and special characters are rejected.
- **Contact Number**: The contact number field must only accept numeric input.
- **Identification Number**: The identification number field must accept only numeric input and reject any non-numeric characters.

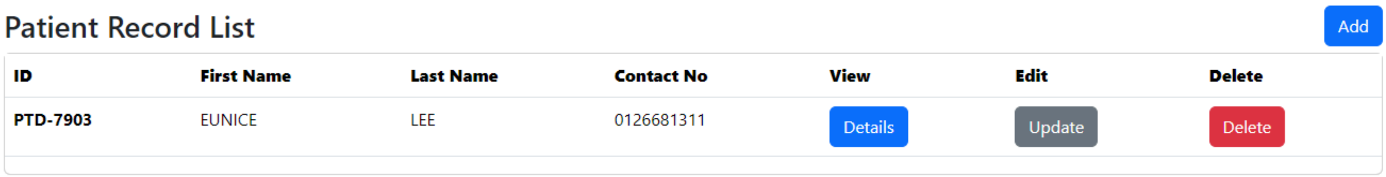
Test ID	Test Case	Test Description	Expected Result	Actual Result	Status
B10	Valid Patient Name Input	The user enters a valid patient name consisting of letters and spaces.	The system successfully registers the patient	The patient was successfully registered and added to the patient list.	PASS
B11	Invalid Patient Name Input	The user enters a patient name containing numbers or special characters.	Error message: “Invalid Data” shown.	Error message: “Invalid Data” shown.The system prompts the user to enter a valid name.	PASS
B12	Valid Identification No Input	The user enters an identification number with digits only.	The system successfully registers the patient.	The patient was successfully registered and added to the patient list.	PASS
B13	Invalid Identification No Input	The user enters an identification number with	Error message: “Invalid Data” shown.	Error message: “Invalid Data” shown.The system prompts the user to	PASS

		letters.		enter a valid identification number.	🟩



Invalid Data

Figure 4.1.2.1 The system displays an error message stating "Invalid data," indicating that the input provided does not meet the required criteria.



Patient Record List

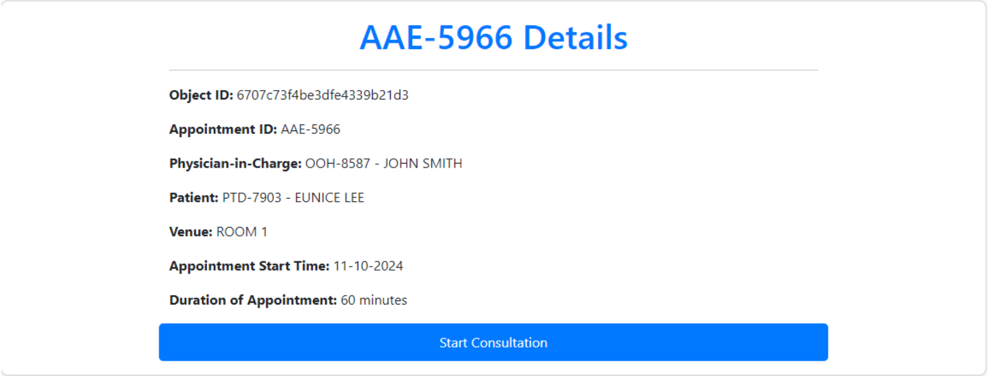
ID	First Name	Last Name	Contact No	View	Edit	Delete
PTD-7903	EUNICE	LEE	0126681311	Details	Update	Delete

Figure 4.1.2.2 The patient has been successfully registered and added to the patient records list.

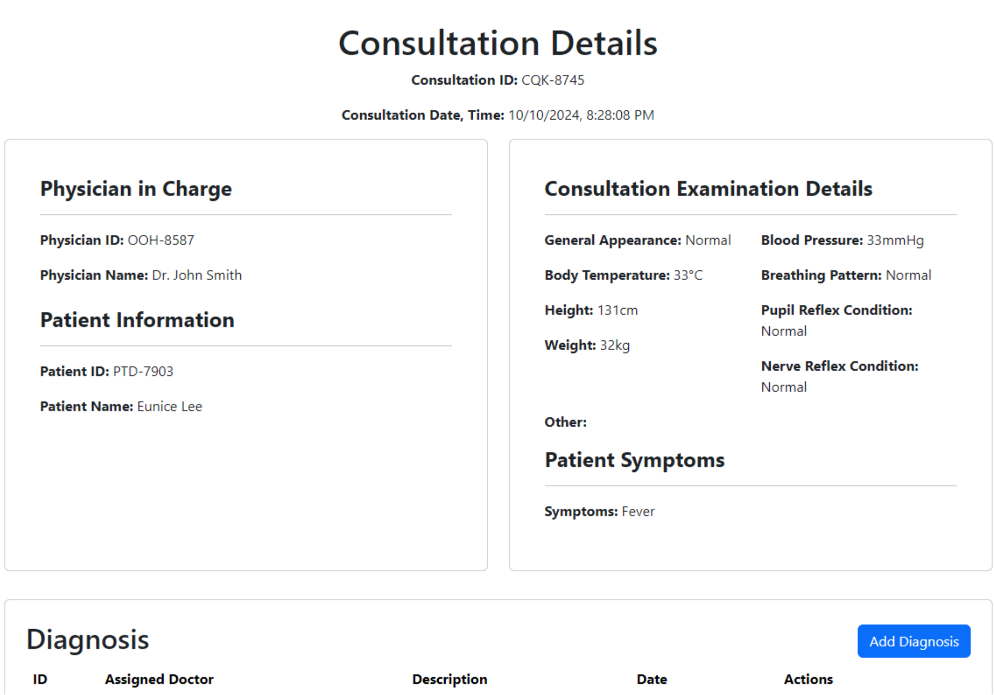
4.1.3. Appointment Scheduling, Diagnosis, and Prescription System

State Transition Testing was used to assess the system's behavior during the appointment scheduling, consultation, diagnosis, and prescription processes. This testing approach is to ensure the system accurately transitions between different states as users navigate these functions.

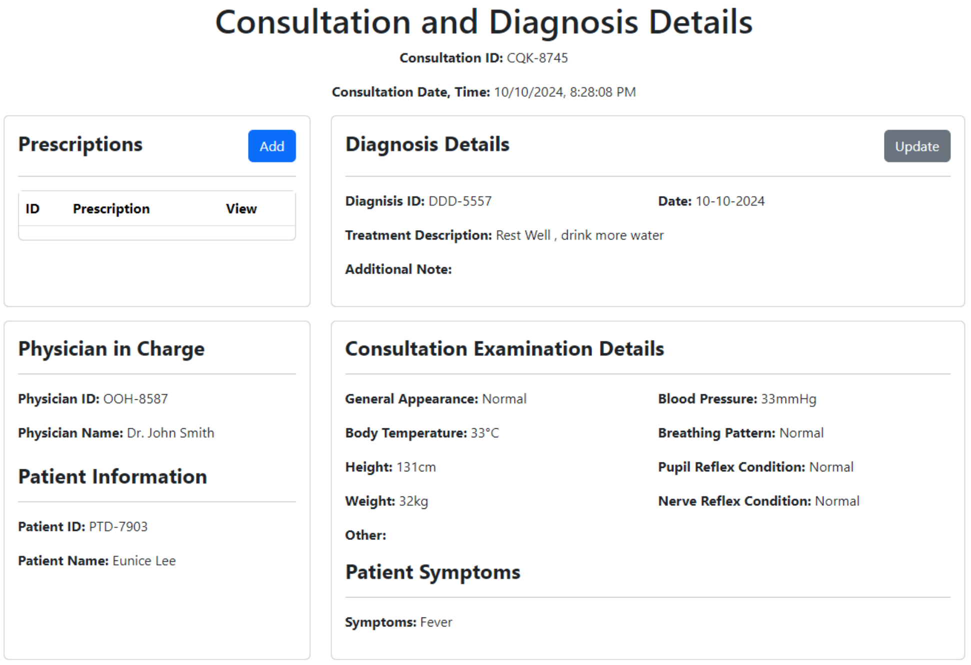
Test ID	B14
Test Case	Scheduling an Appointment
Test Description	1. Navigate to the homepage and click on the "Schedule Appointment" button. 2. Fill in the required appointment details and click "Add Appointment." 3. Click on "View Appointment" to see the appointment details. 4. Update the appointment by clicking "Update Appointment," modifying the details, and saving the changes.

	5. Delete the appointment by clicking the "Delete Appointment" button.
Expected Result	● The system redirects to the Add Appointment page. ● The appointment is successfully added to the database. ● The appointment details are correctly displayed. ● The updated details are saved and displayed. ● The appointment is removed from the system.
Actual Result	PASS
Screenshot	 AAE-5966 Details **Object ID:** 6707c73f4be3dfe4339b21d3 **Appointment ID:** AAE-5966 **Physician-in-Charge:** OOH-8587 - JOHN SMITH **Patient:** PTD-7903 - EUNICE LEE **Venue:** ROOM 1 **Appointment Start Time:** 11-10-2024 **Duration of Appointment:** 60 minutes Figure 4.1.3.1 Show the details of the scheduled appointment after the user successfully added an appointment .

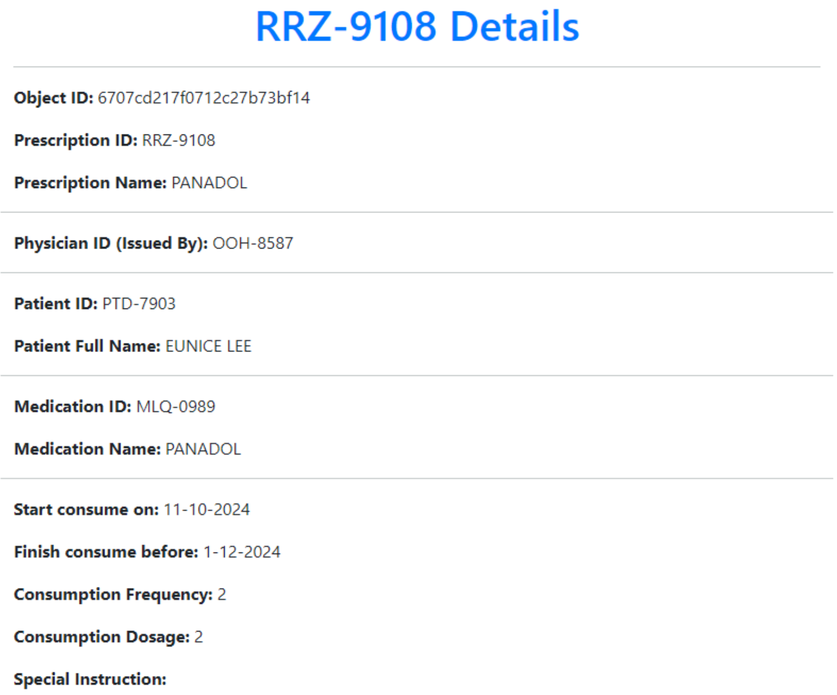
Test ID	B15
Test Case	Initiating Encounter
Test Description	1. Click on the "Start" button in the appointment records page. 2. Fill out the medical consultation form with relevant details and click "Submit."
Expected Result	● The system redirects to the consultation feature once the user clicks on the start button. ● The consultation is saved, and the appointment is removed from the list.
Actual Result	PASS

Test ID	B16
Test Case	Managing Consultation Details
Test Description	1. Click on "View Consultation Detail" to see the details of the consultation. 2. Update consultation details by clicking "Update Consultation Detail," modify the information, and save the changes. 3. Delete a consultation record by clicking "Delete Consultation."
Expected Result	● Consultation details are correctly displayed. ● Updated details are saved in the database and displayed. ● The consultation record is removed from the system.
Actual Result	PASS
Screenshot	 Consultation Details **Consultation ID:** CQK-8745 **Consultation Date, Time:** 10/10/2024, 8:28:08 PM Physician in Charge **Physician ID:** OOH-8587 **Physician Name:** Dr. John Smith Patient Information **Patient ID:** PTD-7903 **Patient Name:** Eunice Lee Consultation Examination Details **General Appearance:** Normal **Blood Pressure:** 33mmHg **Body Temperature:** 33°C **Breathing Pattern:** Normal **Height:** 131cm **Pupil Reflex Condition:** Normal **Weight:** 32kg **Nerve Reflex Condition:** Normal **Other:** Patient Symptoms **Symptoms:** Fever Diagnosis ID | Assigned Doctor | Description | Date | Actions Figure 4.1.3.2 Display of the consultation details.

Test ID	B17
Test Case	Adding Diagnosis

Test Description	1. Click on "Add Diagnosis" to navigate to the Add Diagnosis page. 2. Enter the diagnosis details and click "Submit Diagnosis." 3. View diagnosis details by clicking "View Diagnosis Detail." 4. Update the diagnosis by clicking "Update Diagnosis Detail" and modifying the details. 5. Delete a diagnosis by clicking "Delete Diagnosis."
Expected Result	● The system redirects to the Add Diagnosis page. ● The diagnosis is saved in the database. ● Diagnosis details are displayed correctly. ● Updated details are saved and displayed. ● The diagnosis is removed from the system
Actual Result	PASS
Screenshot	 Consultation and Diagnosis Details **Consultation ID:** CQK-8745 **Consultation Date, Time:** 10/10/2024, 8:28:08 PM Prescriptions ID Prescription View Diagnosis Details **Diagnisis ID:** DDD-5557 **Date:** 10-10-2024 **Treatment Description:** Rest Well , drink more water **Additional Note:** Physician in Charge **Physician ID:** OOH-8587 **Physician Name:** Dr. John Smith Patient Information **Patient ID:** PTD-7903 **Patient Name:** Eunice Lee Consultation Examination Details **General Appearance:** Normal **Blood Pressure:** 33mmHg **Body Temperature:** 33°C **Breathing Pattern:** Normal **Height:** 131cm **Pupil Reflex Condition:** Normal **Weight:** 32kg **Nerve Reflex Condition:** Normal **Other:** Patient Symptoms **Symptoms:** Fever Figure 4.1.3.3 Display of consultation and diagnosis details after submitting the diagnosis.

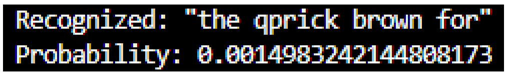
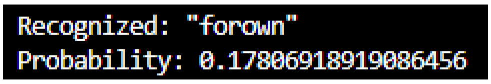
Test ID	B18
Test Case	Adding Prescription

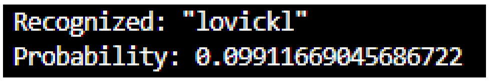
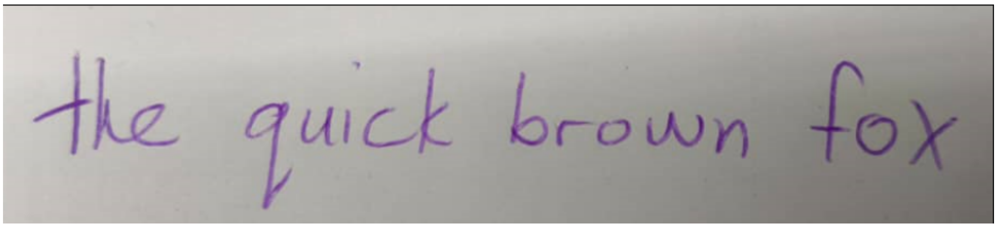
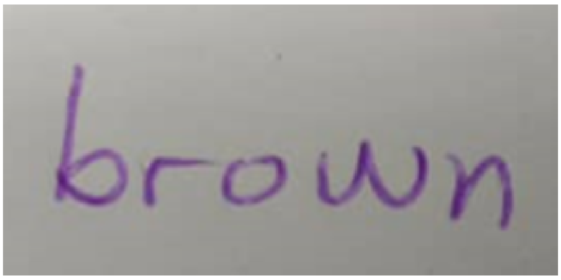
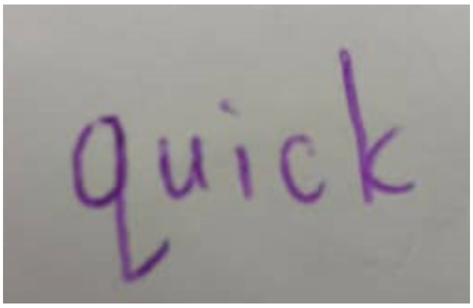
Test Description	1. Click on "Add Prescription" to navigate to the Add Prescription page. 2. Enter prescription details and click "Submit Prescription." 3. View prescription details by clicking "View Prescription Detail."
Expected Result	● The system redirects to the Add Prescription page. ● Prescription details are saved in the database. ● Prescription details are displayed correctly.
Actual Result	PASS
Screenshot	 RRZ-9108 Details **Object ID:** 6707cd217f0712c27b73bf14 **Prescription ID:** RRZ-9108 **Prescription Name:** PANADOL **Physician ID (Issued By):** OOH-8587 **Patient ID:** PTD-7903 **Patient Full Name:** EUNICE LEE **Medication ID:** MLQ-0989 **Medication Name:** PANADOL **Start consume on:** 11-10-2024 **Finish consume before:** 1-12-2024 **Consumption Frequency:** 2 **Consumption Dosage:** 2 **Special Instruction:** Figure 4.1.3.4 Show the details of the prescription after the user has successfully added the prescription.

4.2. Test 2: HTR model

In this black-box testing phase for the HTR model, a comprehensive evaluation is conducted to assess the system's ability to accurately recognize and interpret handwritten input under various conditions. The test cases include a diverse set of scenarios, such as recognizing text written with different colored pens, cursive handwriting, uppercase letters, thicker/bolder pens, symbols and backgrounds with varying shadow conditions. Some test cases consist of complete sentences while others focus on single words. However, the distinction between these formats is less significant for the overall testing objectives. Ultimately, we will not be testing full sentences as the primary focus is on specific fields within our patient form. A test case will be considered a pass if the model achieves an accuracy rate of 80% or above, which is calculated based on the number of correctly recognized characters compared to the total characters in the input. Additionally, the model provides a probability score that reflects its confidence in the recognized text. This probability helps us understand how likely the recognized text is correct that gives a clearer picture of the model's reliability even beyond the raw accuracy percentage.

4.2.1. With Shadows in Backgrounds Test

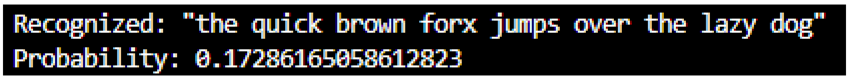
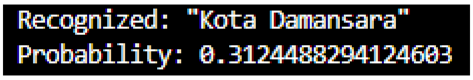
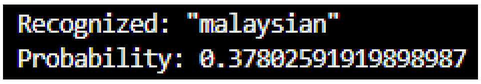
Test ID	T1
Test Case	Verify the ability of the HTR model to accurately extract text from handwritten forms with shadows present in the background.
Test Method	Negative Testing
Test Process Description	1. Write down a sentence or single word on a piece of white paper using a pen. 2. Take a photo of the handwritten sentence or word ensuring the image is well-lit and clear. 3. Upload the photo to the HTR model for processing, and observe the output text generated by the model.
Input	A photo of handwritten text (sentence or word) with visible shadows in the background uploaded by the user.
Expected Output	The HTR model should correctly recognize and transcribe the handwritten text, despite the presence of shadows and output the correct sentence or word.
Actual Output	Figure 4.2.1.1 results:  Recognized: "the qprick brown for" Probability: 0.0014983242144808173 Accuracy: 95% Figure 4.2.1.2 results:  Recognized: "forown" Probability: 0.17806918919086456 Accuracy: 50% Figure 4.2.1.3 results:

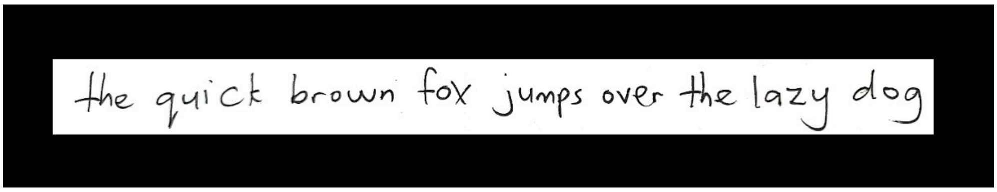
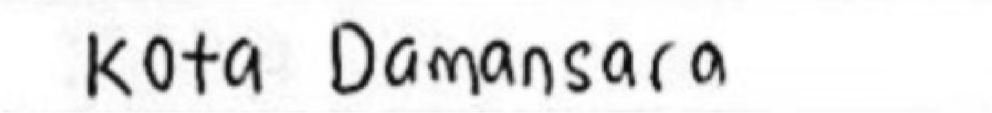
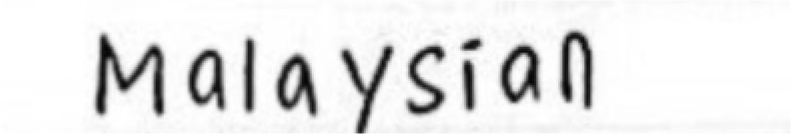
	 Recognized: "lovickl" Probability: 0.09911669045686722 Accuracy: 0% The results demonstrate significant inconsistencies in the HTR model's performance with shadows present making it unsuitable to pass this test.
Result	FAIL
Screenshots	 Figure 4.2.1.1 Image of a handwritten sentence tested using the HTR model  Figure 4.2.1.2 Image of a handwritten word tested using the HTR model  Figure 4.2.1.3 Image of a handwritten word tested using the HTR model

4.2.2. Removing Shadow from Backgrounds Test

Test ID	T2
Test Case	Verify the ability of the HTR model to recognize text accurately when

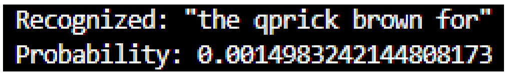
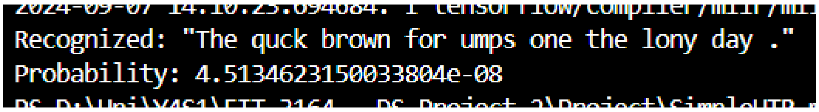
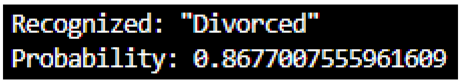
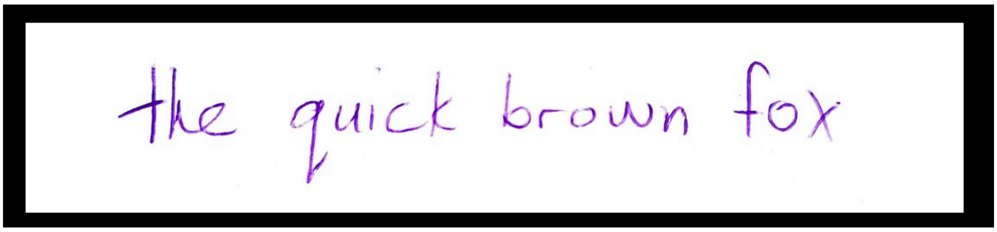
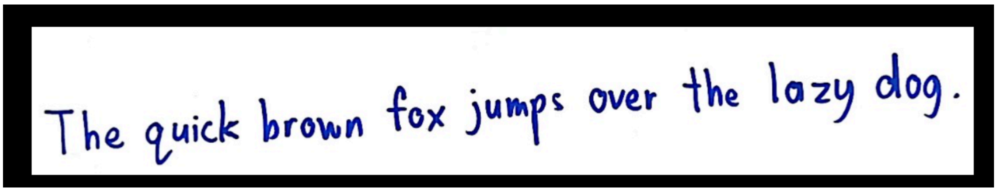
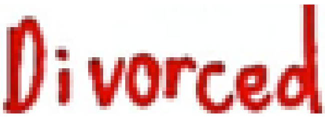
	shadows are removed from the background of the handwritten image.
Test Method	Performance Testing
Test Process Description	1. Write down a sentence or single word on a piece of white paper using a pen. 2. Take a photo of the sentence/word in a well-lit environment ensuring no shadows are present on the paper. 3. Used a cropping model to remove more shadows from the backgrounds. 4. Upload the photo to the HTR model for processing, and observe the output text generated by the model.
Input	A photo of a handwritten sentence/word without any shadows in the background uploaded by the user.
Expected Output	The HTR model successfully processes the image, recognizing and extracting the text with an accuracy of at least 80%. The extracted text matches the handwritten input with minor or no errors in character recognition.

Actual Output	Figure 4.2.2.1 results:
	
	`Recognized: "the quick brown forx jumps over the lazy dog"`
	`Probability: 0.17286165058612823`
	Accuracy: 97%
	
	Figure 4.2.2.2 results:
	
	`Recognized: "Kota Damansara"`
	`Probability: 0.3124488294124603`
	Accuracy: 100%
	
	Figure 4.2.2.3 results:
	
	`Recognized: "malaysian"`
	`Probability: 0.37802591919898987`
	Accuracy: 89%
	
	The results show that the HTR model performs well when shadows are removed with high accuracy rates across different test cases. From here, all tests are going to be using images without shadows in the background.

Result	PASS
Screenshots	 the quick brown fox jumps over the lazy dog Figure 4.2.2.1 Image of a handwritten sentence tested using the HTR model  Kota Damansara Figure 4.2.2.2 Image of a word used to be tested by HTR model  Malaysian Figure 4.2.2.3 Image of a word used to be tested by HTR model

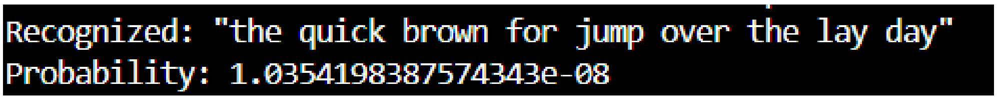
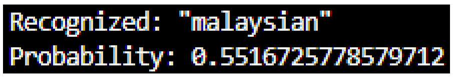
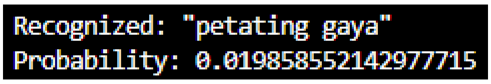
4.2.3. Different Colored Pens (Blue, Red, Pink) Test

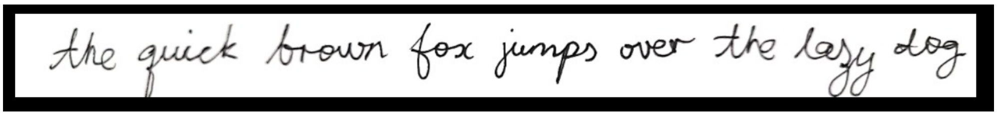
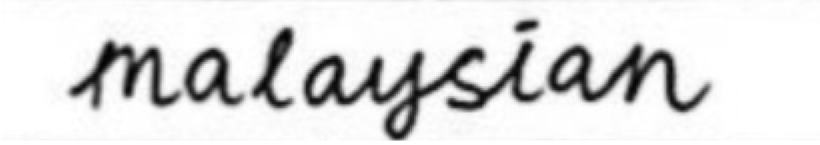
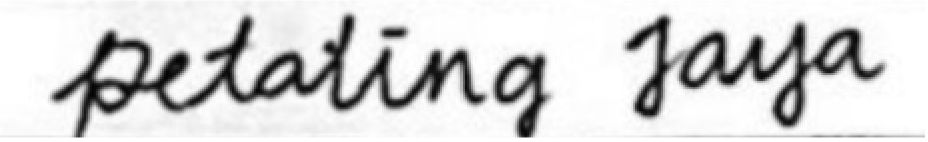
Test ID	T3
Test Case	Verify the ability of the HTR model to accurately recognize text written with different colored pens.
Test Method	Compatibility Testing
Test Process Description	1. Write a word or sentence on a piece of white paper using different colored pens (e.g., black, blue, and red). 2. Take a photo of the text written with each pen. 3. Used a cropping model to remove more shadows from the backgrounds. 4. Upload the photo to the HTR model for processing, and observe the output text generated by the model.
Input	A photo of a handwritten sentence/word with different coloured pens uploaded by the user.
Expected Output	The HTR model processes the images and recognizes the text accurately with an expected accuracy of 80% or higher for each color tested.

Actual Output	Figure 4.2.3.1 results:  Recognized: "the qprick brown for" Probability: 0.0014983242144808173 Accuracy: 89% Figure 4.2.3.2 results:  2024-09-07 14:10:25.694684: I tensorflow/compiler/mlir/mli Recognized: "The quck brown for umps one the lony day ." Probability: 4.5134623150033804e-08 PS D:\Uni\Y4S1\FIT 3164 - DS Project 2\Project\SimpleHTR Accuracy: 77% Figure 4.2.3.3 results:  Recognized: "Divorced" Probability: 0.8677007555961609 Accuracy: 100% The results indicate that the HTR model exhibits varying levels of performance when recognizing text written with different colored pens with accuracy rates showing both successful recognition and areas needing improvement. Overall, these test cases achieved high accuracy.
Result	PASS
Screenshots	 the quick brown fox Figure 4.2.3.1 Image of a handwritten sentence using a pink pen tested using the HTR model  The quick brown fox jumps over the lazy dog. Figure 4.2.3.2 Image of a handwritten sentence using a blue pen tested using the HTR model  Divorced Figure 4.2.3.3 Image of a word used to be tested using HTR model

4.2.4. Cursive Writing Test

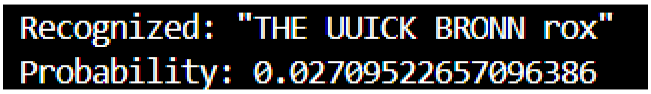
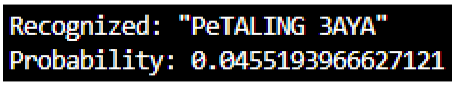
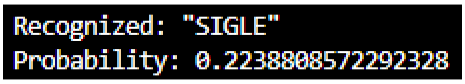
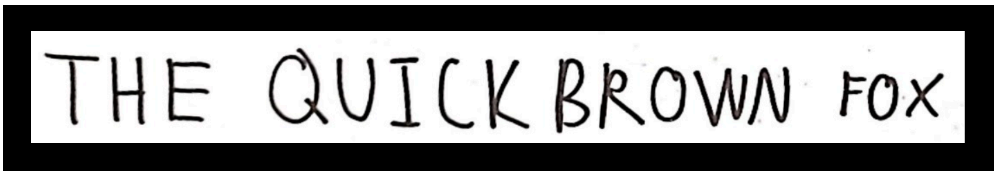
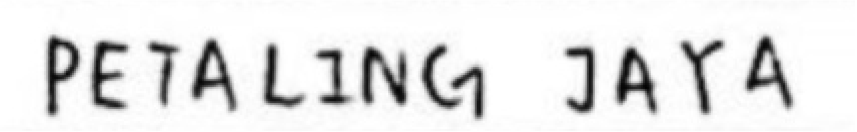
Test ID	T4
Test Case	Verify the HTR model's ability to recognize and interpret cursive handwriting.
Test Method	Functional Testing
Test Process Description	1. Write a sentence or a single word in cursive on a piece of white paper using a pen. 2. Take a photo of the text written. 3. Used a cropping model to remove more shadows from the backgrounds. 4. Upload the photo to the HTR model for processing, and observe the output text generated by the model.
Input	A photo of a cursively written sentence or word uploaded by the user.
Expected Output	The HTR model should accurately recognize the cursive handwriting returning the correct characters with high precision. The accuracy should be 80% or above.

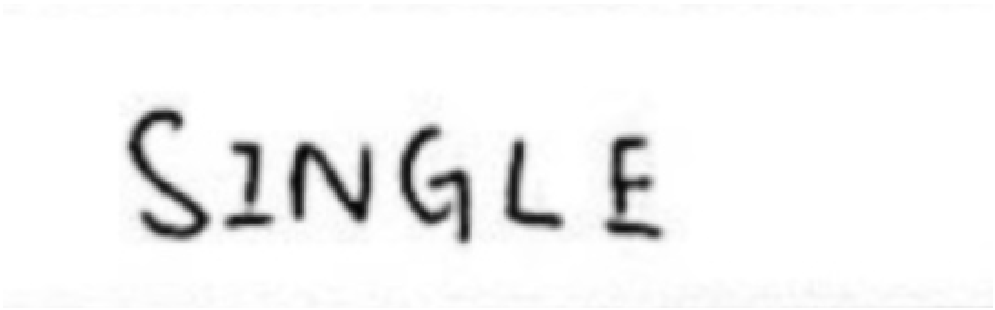
Actual Output	Figure 4.2.4.1 results:
	
	`Recognized: "the quick brown for jump over the lay day"`
	`Probability: 1.0354198387574343e-08`
	Accuracy: 88%
	
	Figure 4.2.4.2 results:
	
	`Recognized: "malaysian"`
	`Probability: 0.5516725778579712`
	Accuracy: 100%
	
	Figure 4.2.4.3 results:
	
	`Recognized: "petating gaya"`
	`Probability: 0.019858552142977715`
	Accuracy: 85%
	
	The results indicate that the HTR model demonstrates varying levels of performance when recognizing text written in cursive. Overall, these test cases achieved high accuracy.

Result	PASS
Screenshots	 the quick brown fox jumps over the lazy dog Figure 4.1.4.1 Image of a handwritten sentence in cursive tested using the HTR model  malaysian Figure 4.1.4.2 Image of a handwritten word in cursive tested using the HTR model  petaling Jaya Figure 4.1.4.3 Image of a word in cursive used to be tested usingHTR model

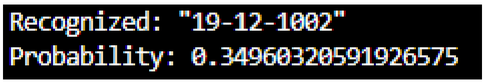
4.2.5. All Capital Letters Test

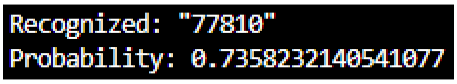
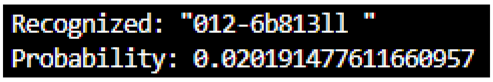
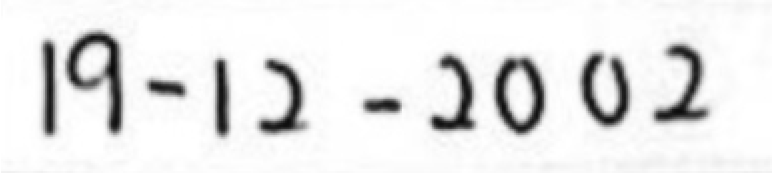
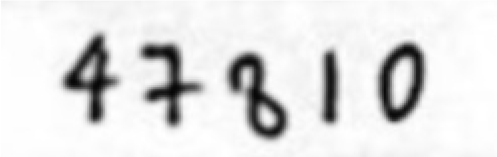
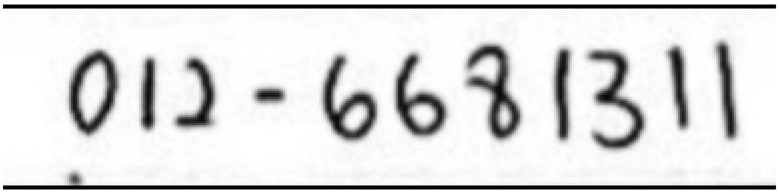
Test ID	T5
Test Case	Verify the ability of the HTR model to accurately recognize handwritten text written entirely in capital letters.
Test Method	Boundary Value Testing
Test Process Description	1. Write a sentence or a single word in capital letters on a piece of white paper using a pen. 2. Take a photo of the text written. 3. Used a cropping model to remove more shadows from the backgrounds. 4. Upload the photo to the HTR model for processing, and observe the output text generated by the model.
Input	A photo of uppercase words written in a sentence or word uploaded by the user.
Expected Output	The HTR model should accurately recognize the capital handwriting, returning the correct characters with high precision. The accuracy

	should be 80% or above.
Actual Output	Figure 4.2.5.1 results:  Recognized: "THE UUICK BRONN rox" Probability: 0.027095522657096386 Accuracy: 79% Figure 4.2.5.2 results:  Recognized: "PeTALING 3AYA" Probability: 0.0455193966627121 Accuracy: 85% Figure 4.2.5.3 results:  Recognized: "SIGLE" Probability: 0.2238808572292328 Accuracy: 83% The results show that the HTR model exhibits inconsistent performance in recognizing text written in capital letters. These test cases reveal areas for improvement in the model's recognition of uppercase text. However, these test cases are still of high accuracy.
Result	PASS
Screenshots	 Figure 4.2.5.1 Image of a handwritten sentence in uppercase tested using the HTR model  Figure 4.2.5.2 Image of a handwritten word in uppercase tested using the HTR model

	 Figure 4.2.5.3 Image of a word in uppercase used to be tested usingHTR model

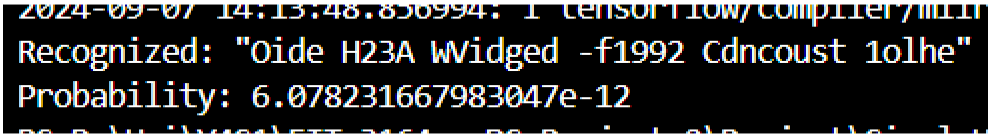
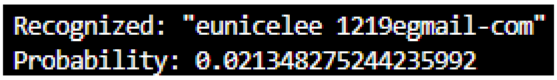
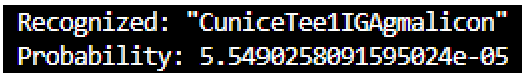
4.2.6. All Numbers Test

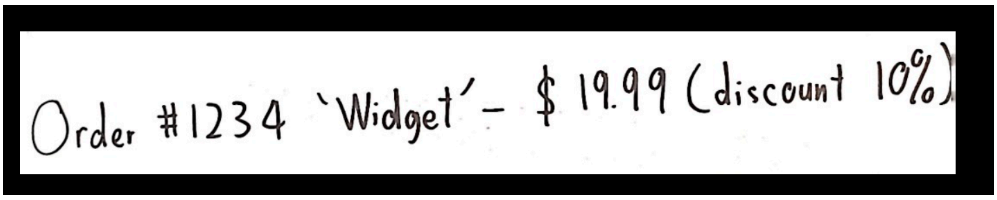
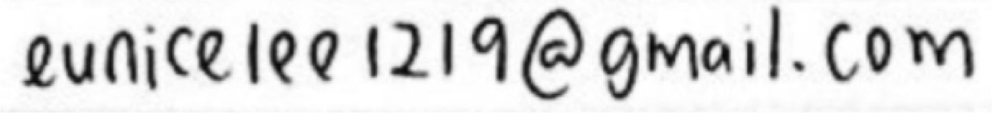
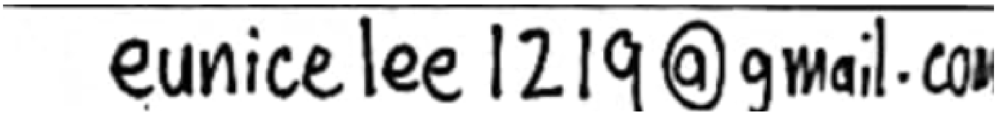
Test ID	T6
Test Case	Verify the ability of the HTR model to accurately recognize handwritten text consisting entirely of numerical values.
Test Method	Boundary Value Testing
Test Process Description	1. Write a sequence of numbers on a piece of white paper using a pen. 2. Take a photo of the sequence of numbers written. 3. Used a cropping model to remove more shadows from the backgrounds. 4. Upload the photo to the HTR model for processing, and observe the output text generated by the model.
Input	A photo of handwritten numbers (e.g., dates or numerical sequences) uploaded by the user.
Expected Output	The HTR model should accurately recognize the handwritten numbers, returning the correct characters with high precision. The accuracy should be 80% or above.
Actual Output	Figure 4.2.6.1 results:  Recognized: "19-12-1002" Probability: 0.34960320591926575 Accuracy: 92% Figure 4.2.6.2 results:

	 Recognized: "77810" Probability: 0.7358232140541077 Accuracy: 80% Figure 4.2.6.3 results:  Recognized: "012-6b813ll " Probability: 0.020191477611660957 Accuracy: 72% The results demonstrate that the HTR model displays inconsistent performance in recognizing handwritten numbers. Overall, these test cases reflect a mixed performance, with some achieving high accuracy while others highlight the need for further improvement. However, since the majority of the test cases meet or exceed the minimum accuracy threshold of 80%, the HTR model can still be considered to pass the test.
Result	PASS
Screenshots	 Figure 4.2.6.1 Image of a handwritten date tested using the HTR model  Figure 4.2.6.2 Image of a handwritten postcode tested using the HTR model  Figure 4.2.6.3 Image of a handwritten number used to be tested usingHTR model

4.2.7. Symbols Test

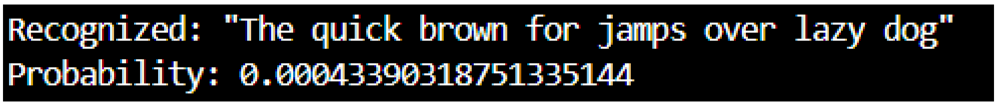
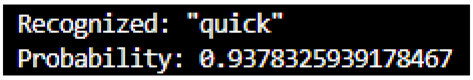
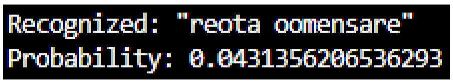
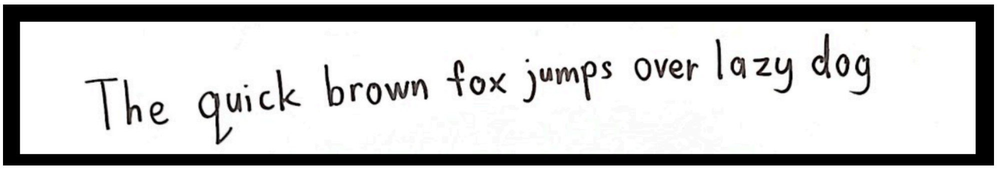
Test ID	T7
Test Case	Verify the ability of the HTR model to accurately recognize handwritten symbols.
Test Method	Stress Testing
Test Process Description	1. Write a sequence of words and symbols on a piece of white paper using a pen. 2. Take a photo of the words and symbols written. 3. Used a cropping model to remove more shadows from the backgrounds. 4. Upload the photo to the HTR model for processing, and observe the output text generated by the model.
Input	A photo of a sequence of words and symbols (e.g., @, #, $, &, *) uploaded by the user.
Expected Output	The HTR model should accurately recognize the symbols, returning the correct characters with high precision. The accuracy should be 80% or above.

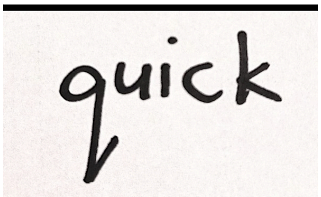
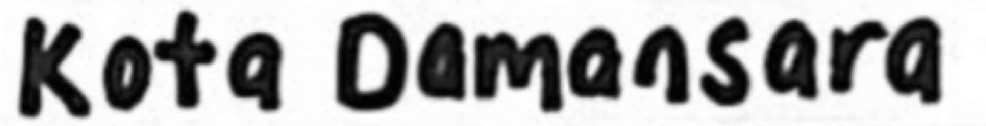
Actual Output	Figure 4.2.7.1 results:
	
	`2024-09-07 14:13:48.836994: I tensorflow/compiler/mlir`
	`Recognized: "Oide H23A WVidged -f1992 Cdncoust 1olhe"`
	`Probability: 6.078231667983047e-12`
	Accuracy: 52%
	
	Figure 4.2.7.2 results:
	
	`Recognized: "eunicelee 1219egmail-com"`
	`Probability: 0.021348275244235992`
	Accuracy: 91%
	
	Figure 4.2.7.3 results:
	
	`Recognized: "CuniceTee1IGAgmalicon"`
	`Probability: 5.5490258091595024e-05`
	Accuracy: 62%
	
	The results demonstrate that the HTR model exhibits inconsistent performance in recognizing handwritten symbols. Although accuracy

	is high, the majority of the symbols were recognized incorrectly. However, since the forms primarily require recognition of a limited number of symbols such as commas (,), periods (.) and hyphens (-), this limitation can be mitigated.
Result	FAIL
Screenshots	 Figure 4.2.7.1 Image of a handwritten sequence of words and symbols tested using the HTR model  Figure 4.2.7.2 Image of a handwritten sequence of words and symbols tested using the HTR model  Figure 4.2.7.3 Image of a handwritten sequence of words and symbols used to be tested usingHTR model

4.2.8. Thicker Pens Test

Test ID	T8
Test Case	Verify the ability of the HTR model to accurately recognize handwritten text written with thicker pens.
Test Method	Stress Testing
Test Process Description	1. Write a sequence of numbers on a piece of white paper using a thicker pen (0.7mm and above, sharpie). 2. Take a photo of the text written. 3. Used a cropping model to remove more shadows from the backgrounds. 4. Upload the photo to the HTR model for processing, and observe the output text generated by the model.

Input	A photo of handwritten text (sentence or word) uploaded by the user.
Expected Output	The HTR model should accurately recognize the handwritten numbers, returning the correct characters with high precision. The accuracy should be 80% or above.
Actual Output	Figure 4.2.6.1 results:  Recognized: "The quick brown for jamps over lazy dog" Probability: 0.00043390318751335144 Accuracy: 97% Figure 4.2.6.2 results:  Recognized: "quick" Probability: 0.9378325939178467 Accuracy: 100% Figure 4.2.6.3 results:  Recognized: "reota oomensare" Probability: 0.0431356206536293 Accuracy: 67% The results indicate that the HTR model exhibits varying performance when recognizing handwritten text, with thicker pens generally improving accuracy. However, excessively thick pens, such as sharpies, can negatively impact recognition as seen in the results. This issue can be mitigated as no one typically uses sharpies to fill in the forms. Despite this inconsistency, the majority of test cases meet the minimum accuracy threshold of 80% that allows the results to be considered a pass.
Result	PASS
Screenshots	 The quick brown fox jumps over lazy dog Figure 4.2.6.1 Image of a handwritten sentence using 0.7mm pen tested using the HTR model

	 quick Figure 4.2.6.2 Image of a handwritten word using 0.9mm pen tested using the HTR model  Kota Damansara Figure 4.2.6.3 Image of a handwritten sentence using Sharpie tested using HTR model

5. Integration Testing

Integration testing is conducted to ensure that various software components function cohesively and interact with each other as intended after integration, especially in terms of functionality and performance between the models and the web-based system. This approach is essential for uncovering and resolving any defects that may arise from component integration and those that may not be apparent when testing individual components (Testlio, 2024). This includes compatibility issues, database defects, data flow defects, performance defects and API communication error between the frontend and backend of the web-based system, including the ICROP and HTR models (Tasnim, 2024). By detecting these incompatibilities early in the development cycle, integration testing enhances overall system reliability and quality (Testlio, 2024).

Through testing, the teams validate component interactions, assess overall system performance, ensure accurate data flow, and maintain data integrity. This testing is essential in achieving software quality and robustness, as it ensures that the final product system meets specified requirements and that the interface is bug-free (Testlio, 2024).

The Automated Health Information System (AHIS) mainly consists of four main components:

1. Frontend of the web-based system built using Angular.
2. Backend of the web-based system built using Express, MongoDB and Node.JS.
3. Image Cropping (ICROP) Model
4. Handwritten Text Recognition (HTR) Model

The following tables outline the integration tests and testing strategies used, describing what is being tested, how it is being tested, what are the inputs to the code, what are the expected outputs and what are the actual outputs being observed.

5.1. Integration Test: ICROP Model with the Web-based System

Test ID	I1
Test Case	Verify the ability to upload a patient registration form from the frontend to the backend and seamlessly transfer it to the ICROP model for form field cropping.
Test Process Description	On the ‘Add Patient’ web page, the user uploads the scanned patient registration form and clicks the 'Upload' button The form is then sent to the ICROP model, which processes the image to crop each individual form field. The cropped images are saved in the backend folder with intuitive file names corresponding to each form field.
Input	A filled patient registration form uploaded by the user in a JPG extension format.
Expected Output	● The uploaded patient registration form is processed, with each form field cropped and saved in the backend folder named 'crop_images,' using intuitive file names that match the corresponding form fields. ● The 'Upload' button is disabled, the 'Extract' button is enabled, and a confirmation message 'Image uploaded successfully' is displayed on the form.
Actual Output	In the frontend: ● ‘Upload’ button is disabled. ● ‘Extract’ button is enabled. ● ‘Image Upload successfully’ message is displayed in the form. In the backend: ● Cropped image fields are stored in the ‘crop_images’ folder with intuitive file names.
Result	PASS

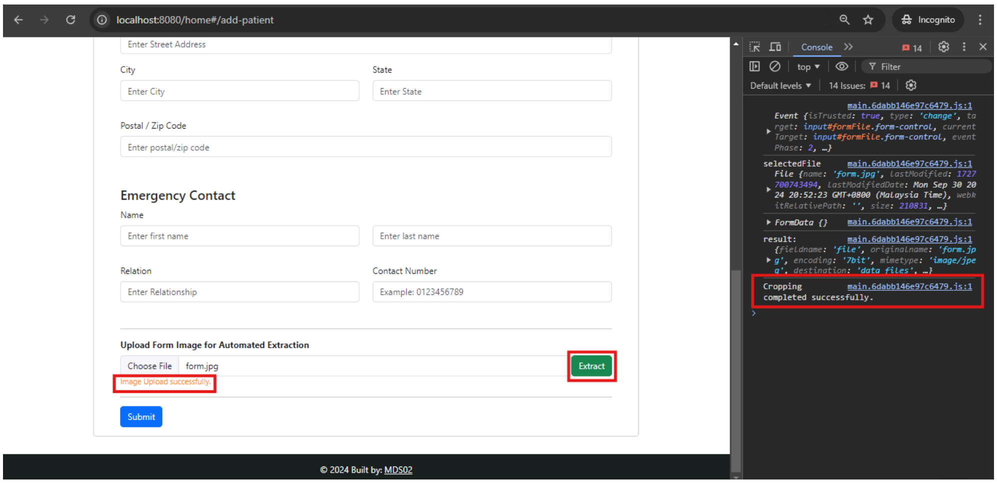
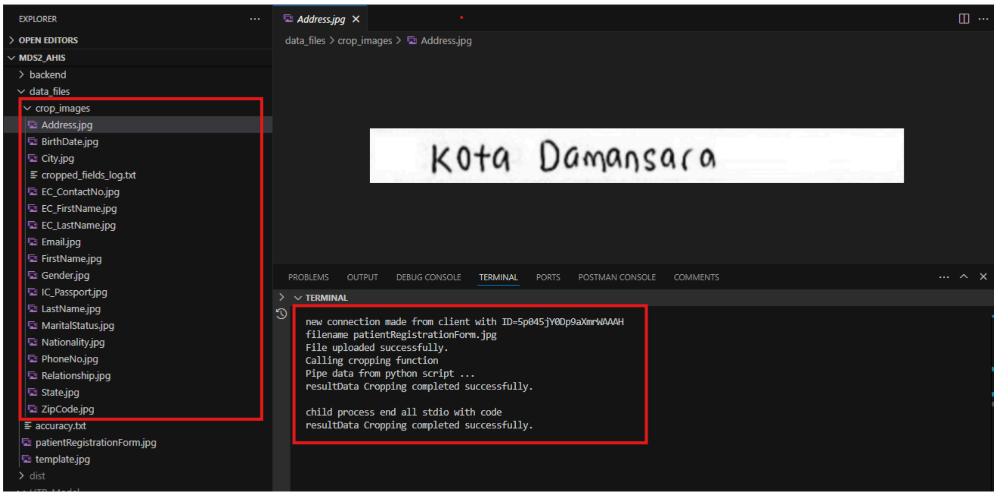
Screenshots	 localhost:8080/home#/add-patient Enter Street Address City: Enter City State: Enter State Postal / Zip Code: Enter postal/zip code Emergency Contact Name: Enter first name / Enter last name Relation: Enter Relationship Contact Number: Example: 0123456789 Upload Form Image for Automated Extraction Choose File form.jpg Extract Image Upload successfully. Submit © 2024 Built by: MDS02 Console main.6dabb146e97c6479.js:1 Event {isTrusted: true, type: 'change', target: input#formFile.form-control, currentTarget: input#formFile.form-control, eventPhase: 2, …} selectedFile main.6dabb146e97c6479.js:1 File {name: 'form.jpg', lastModified: 1727700743494, lastModifiedDate: Mon Sep 30 2024 20:52:23 GMT+0800 (Malaysia Time), webkitRelativePath: '', size: 210831, …} FormData {} main.6dabb146e97c6479.js:1 result: main.6dabb146e97c6479.js:1 {fieldname: 'file', originalname: 'form.jpg', encoding: '7bit', mimetype: 'image/jpeg', destination: 'data files', …} Cropping completed successfully. main.6dabb146e97c6479.js:1 Figure 5.1.1 Changes on the webpage and console log the ICROP model completion.  EXPLORER OPEN EDITORS MDS2_AHIS backend data_files crop_images Address.jpg BirthDate.jpg City.jpg cropped_fields_log.txt EC_ContactNo.jpg EC_FirstName.jpg EC_LastName.jpg Email.jpg FirstName.jpg Gender.jpg IC_Passport.jpg LastName.jpg MaritalStatus.jpg Nationality.jpg PhoneNo.jpg Relationship.jpg State.jpg ZipCode.jpg accuracy.txt patientRegistrationForm.jpg template.jpg dist Address.jpg data_files > crop_images > Address.jpg Kota Damansara PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE COMMENTS TERMINAL new connection made from client with ID=5p045jY0Dp9aXmrWAAAH filename patientRegistrationForm.jpg File uploaded successfully. Calling cropping function Pipe data from python script ... resultData Cropping completed successfully. child process end all stdio with code resultData Cropping completed successfully. Figure 5.1.2 Cropped image fields are saved in the 'crop_images' folder and the terminal displays console log updates, showing the progress of the ICROP model execution from start to completion

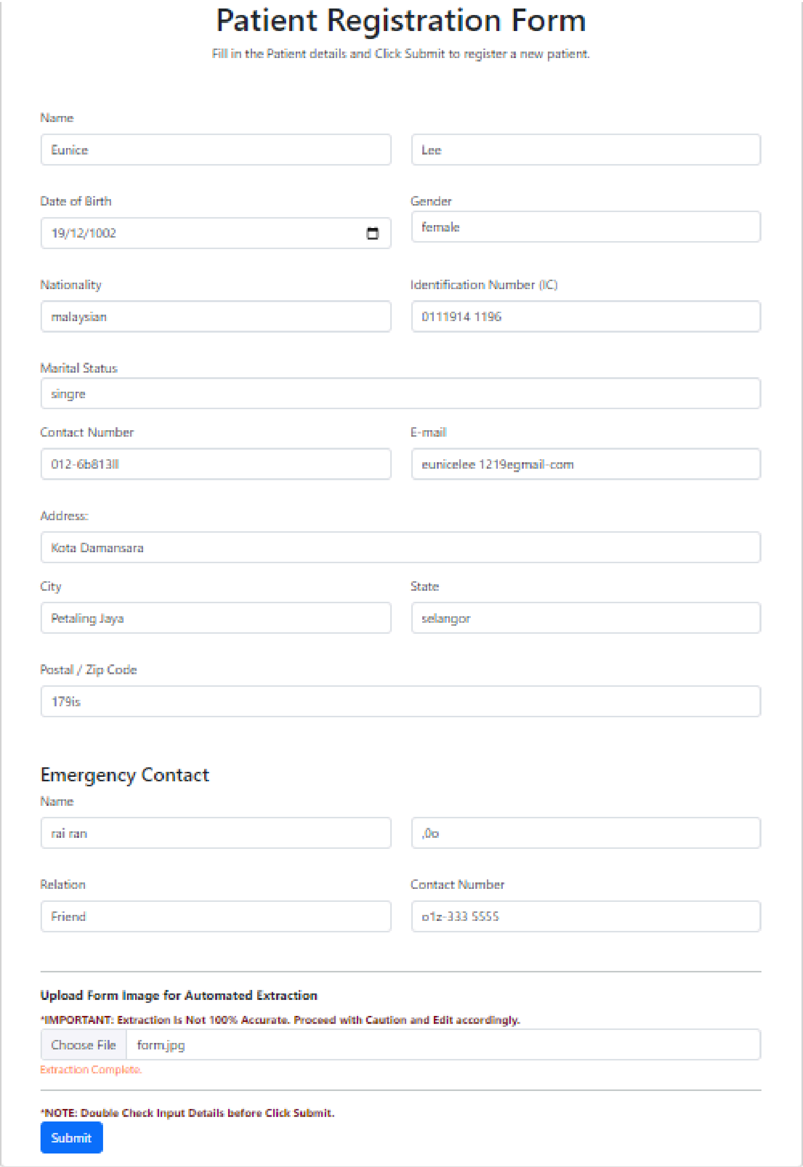
Table 5.1. Integration test of Image Cropping (ICROP) Model with the web-based system

5.2. Integration Test: HTR Model with the Web-based System

Test ID	I2
Test Case	Verify that the HTR model extracts text from each cropped image field and displays the corresponding results in the form.
Test Process Description	1. On the 'Add Patient' web page, after the image is successfully uploaded, click the 'Extract' button. 2. The HTR model processes the cropped image fields and generates the output in a JSON file. 3. The frontend then populates the corresponding form fields

	with the extracted text from the JSON output.
Input	Cropped image fields from ICROP Model.
Expected Output	● The handwritten text is extracted and stored in a JSON file named 'patient_data.'. ● The extracted text is displayed in the corresponding form fields on the webpage ● Upon completion, important notes and confirmation messages are shown on the form, indicating the completion of extraction and advising the user to review and modify patient details if necessary, as the HTR model may not be 100% accurate.
Actual Output	● The handwritten text is extracted and stored in a JSON file named 'patient_data.'. ● The extracted text is displayed in the corresponding form fields on the webpage ● On completion, important notes and confirmation messages are displayed on the form, indicating the completion of extraction and advising the user to review and modify patient details if necessary, as the HTR model may not be 100% accurate.
Result	PASS

Screenshots



Patient Registration Form

Fill in the Patient details and Click Submit to register a new patient.

Name

Eunice

Lee

Date of Birth

19/12/1002

Gender

female

Nationality

malaysian

Identification Number (IC)

0111914 1196

Marital Status

singre

Contact Number

012-6b813ll

E-mail

eunicelee 1219egmail-com

Address:

Kota Damansara

City

Petaling Jaya

State

selangor

Postal / Zip Code

179is

Emergency Contact

Name

rai ran

,0o

Relation

Friend

Contact Number

o1z-333 5555

Upload Form Image for Automated Extraction

***IMPORTANT: Extraction Is Not 100% Accurate. Proceed with Caution and Edit accordingly.**

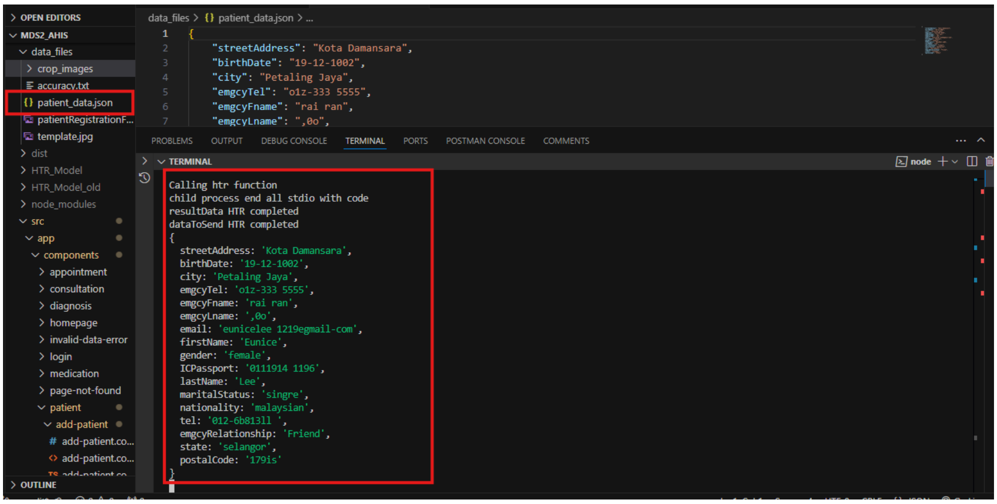
Choose File form.jpg

Extraction Complete.

***NOTE: Double Check Input Details before Click Submit.**

Submit

Figure 5.2.1 Display of the HTR model's output on the web form, along with important notes and confirmation messages to indicate users to review and modify details as needed.



```
{
    "streetAddress": "Kota Damansara",
    "birthDate": "19-12-1002",
    "city": "Petaling Jaya",
    "emgcyTel": "o1z-333 5555",
    "emgcyFname": "rai ran",
    "emgcyLname": ",0o",
```

```
Calling htr function
child process end all stdio with code
resultData HTR completed
dataToSend HTR completed
{
  streetAddress: 'Kota Damansara',
  birthDate: '19-12-1002',
  city: 'Petaling Jaya',
  emgcyTel: 'o1z-333 5555',
  emgcyFname: 'rai ran',
  emgcyLname: ',0o',
  email: 'eunicelee 1219egmail-com',
  firstName: 'Eunice',
  gender: 'female',
  ICPassport: '0111914 1196',
  lastName: 'Lee',
  maritalStatus: 'singre',
  nationality: 'malaysian',
  tel: '012-6b813ll ',
  emgcyRelationship: 'Friend',
  state: 'selangor',
  postalCode: '179is'
}
```

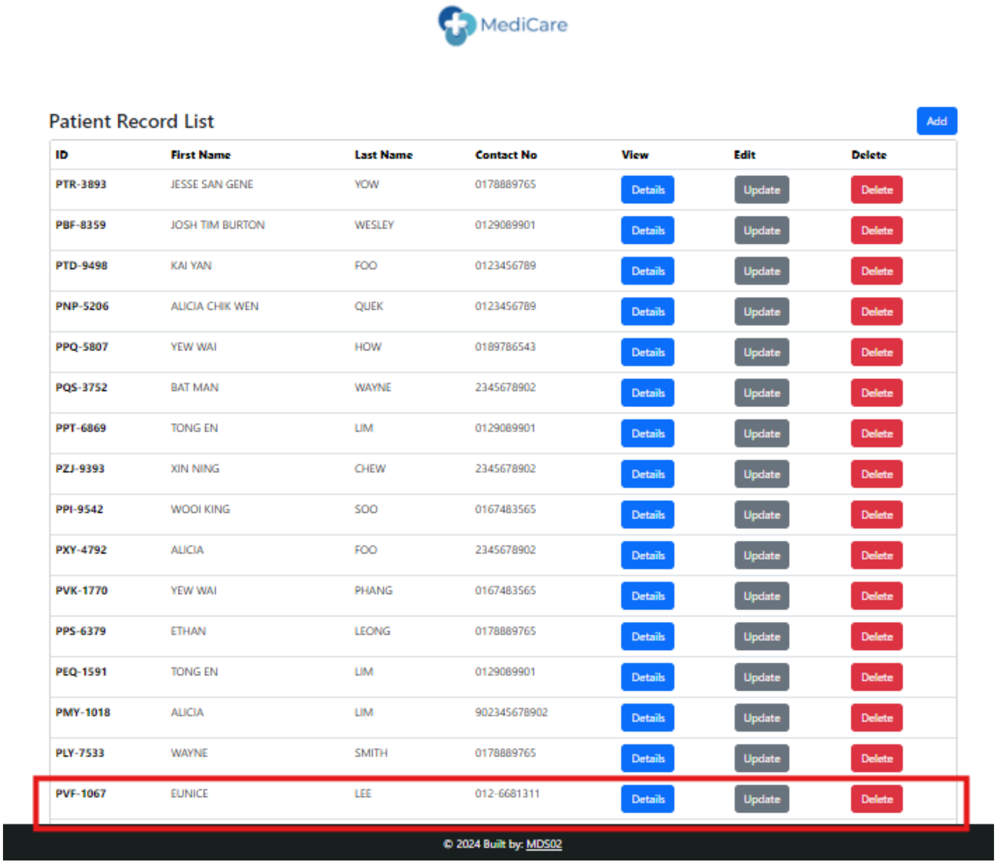
	Figure 5.2.2 Extracted handwritten text written to a json file named ‘patient_data’ and the terminal displays console log updates from initiation to completion and output of the HTR model.

Table 5.2. Integration test of Handwritten Text Recognition (HTR) Model with the web-based system

5.2. Integration Test: Web-based System with the MongoDB Database

Test ID	I3
Test Case	Verify that the patient details output by the HTR model can be successfully saved to the database after being modified and submitted.
Test Process Description	1. Review and correct any inaccuracies in the patient details generated by the HTR model. 2. Click the ‘Submit’ button once done.
Input	The patient details.
Expected Output	● New patient is added to the database with the accurate patient details entered and sent from the frontend to the backend. ● The web-based system redirects to the patient list page, where the newly added patient appears in the list, confirming that the operation completed without errors.
Actual Output	● New patient is added to the database with the accurate patient details entered and sent from the frontend to the backend. ● The web-based system redirects to the patient list page, where the newly added patient appears in the list, confirming that the operation completed without errors.
Result	PASS

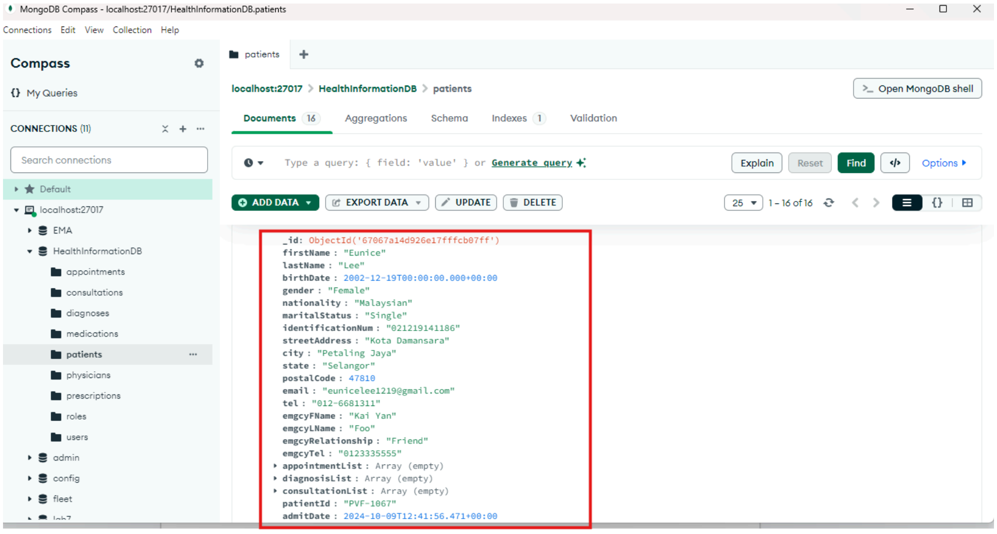
Screenshots



Patient Record List

ID	First Name	Last Name	Contact No	View	Edit	Delete
PTR-3893	JESSE SAN GENE	YOW	0178889765	Details	Update	Delete
PBF-8359	JOSH TIM BURTON	WESLEY	0129089901	Details	Update	Delete
PTD-9498	KAI YAN	FOO	0123456789	Details	Update	Delete
PNP-5206	ALICIA CHIK WEN	QUEK	0123456789	Details	Update	Delete
PPQ-5807	YEW WAI	HOW	0189786543	Details	Update	Delete
PQS-3752	BAT MAN	WAYNE	2345678902	Details	Update	Delete
PPT-6869	TONG EN	LIM	0129089901	Details	Update	Delete
PZJ-9393	XIN NING	CHEW	2345678902	Details	Update	Delete
PPI-9542	WOOI KING	SOO	0167483565	Details	Update	Delete
PXY-4792	ALICIA	FOO	2345678902	Details	Update	Delete
PVK-1770	YEW WAI	PHANG	0167483565	Details	Update	Delete
PPS-6379	ETHAN	LEONG	0178889765	Details	Update	Delete
PEQ-1591	TONG EN	LIM	0129089901	Details	Update	Delete
PMY-1018	ALICIA	LIM	902345678902	Details	Update	Delete
PLY-7533	WAYNE	SMITH	0178889765	Details	Update	Delete
PVF-1067	EUNICE	LEE	012-6681311	Details	Update	Delete

Figure 5.3.1 New patient is added successfully to the database and is being displayed in the patient list page.



```
_id: ObjectId('67067a14d926e17fffcb07ff')
firstName : "Eunice"
lastName : "Lee"
birthDate : 2002-12-19T00:00:00.000+00:00
gender : "Female"
nationality : "Malaysian"
maritalStatus : "Single"
identificationNum : "021219141186"
streetAddress : "Kota Damansara"
city : "Petaling Jaya"
state : "Selangor"
postalCode : 47810
email : "eunicelee1219@gmail.com"
tel : "012-6681311"
emgcyFName : "Kai Yan"
emgcyLName : "Foo"
emgcyRelationship : "Friend"
emgcyTel : "0123335555"
▸ appointmentList : Array (empty)
▸ diagnosisList : Array (empty)
▸ consultationList : Array (empty)
patientId : "PVF-1067"
admitDate : 2024-10-09T12:41:56.471+00:00
```

Figure 5.3.2 Newly added patient information successfully stored in the MongoDB database upon submission without errors.

Table 5.3. Integration test of the web-based system and the backend MongoDB database.

6. Usability Testing

Usability testing was chosen as one of the testing methodologies since it allows us to make sure that the web application satisfies end users' needs (Prescott & Crichton, 1999). The team can determine the web application's shortcomings and assess its efficiency by watching how end-users engage with it. With the information acquired by watching how end users engage with the application, the team will be able to improve it in response to feedback and suggestions, making it more user-friendly and pleasurable for the end users to use (Sfenrianto et al., 2018).

We approached and gathered a few individuals from both inside and outside Monash University to test our web application as part of our usability testing process. These individuals were given a Google Form to complete, which consists of a few simple questions so the team can assess the performance of our web application. The team believe the best way to test out the web application to identify potential bugs and errors is to allow the users who test the web application to conduct any action they want to during the testing process. Hence, during the whole testing process, no guidance will be provided by the project team member present on-site to maintain the integrity of the testing process as well as to guarantee the feedback received are unbiased. Before the present web application is finalized, all of the input and remarks from these individuals will be utilized to enhance the current web application. The table below contains the link to the Google Form received by the users and Google Sheet that recorded the responses of the users.

Google Form	https://forms.gle/fwQf7rPMk9Her8BD7
Google Sheet	https://docs.google.com/spreadsheets/d/16YwAOFqbEoelYWZRsHbzhQ9bB_Bbez2yioPFiwIiNAk/edit?usp=sharing

Table 6.1. Google Form and Google Sheet

Due to the style on how the testing process is conducted, the team decided to merely provide a general guideline on how and what the users should test on. As a result, a table below lists the tasks that were covered throughout the usability testing, along with specific remarks on any unusual actions taken by the users during the process.

Test ID	**Question(s)**	**Student ID**						**Status**
		34444122	33192995	33295026	34074619	32622864	20391646	
Task 1: Logging Into the web application								
6.1.1	How would you rate the difficulty of the task listed above? 1 - Easy 2 - Not Difficult but Not Easy 3 - Difficult 4 - Not	1	1	1	1	1	1	Done

	Applicable / Didn't try this feature out							
6.1.2	If there is, what challenging issues did you experience for the listed task above?	No Challenges	Nope	NO	NA	Not Really	-	Done
6.1.3	Unusual / Special Action	Used Navigation Bar	-	Clicked on the banner	-	Clicked on the banner	-	Done
Task 2: Registering as a Patient								
6.2.1	How would you rate the difficulty of the task listed above? 1 - Easy 2 - Not Difficult but Not Easy 3 - Difficult 4 - Not Applicable / Didn't try this feature out	1	1	1	1	1	1	Done
6.2.2	If there is, what challenging issues did you experience for the listed task above?	No Challenges	Nope	NO	NA	Nope	-	Done
6.2.3	Unusual / Special Action	-	-	-	-	-	-	Done
Task 3: Insert Medication and Physician Record								
6.3.1	How would you rate the difficulty of the task listed above? 1 - Easy	1	1	1	2	1	1	Done

	2 - Not Difficult but Not Easy 3 - Difficult 4 - Not Applicable / Didn't try this feature out							
6.3.2	If there is, what challenging issues did you experience for the listed task above?	No Challenges	Nope	NO	NA	Nope	-	Done
6.3.3	Unusual / Special Action	Asked if there is a profile page. Member answered no	Clicked on the banner	-	Logged out by accident	-	-	Done
Task 4: Schedule Appointment and start encounter for the appointment								
6.4.1	How would you rate the difficulty of the task listed above? 1 - Easy 2 - Not Difficult but Not Easy 3 - Difficult 4 - Not Applicable / Didn't try this feature out	2	1	1	2	2	1	Done
6.4.2	If there is, what challenging issues did you experience for the listed task above?	No Challenges	Nope	NO	NA	Not Really	-	Done
6.4.3	Unusual / Special Action	Used navigation	Logged out by	Used navigation	Used navigation	Used the navigation	-	Done

		menu	accident by clicking on team member name	menu	bar	menu to navigate back previous page		
Task 5: How would you rate the overall user experience when using the web application?								
6.5.1	User Experience: User Interface (UI) 1 - Very Satisfied 2 - Satisfied 3 - Not bad but not good 4 - Bad 5 - Very Bad	1	1	1	1	1	1	Done
6.5.2	User Experience: Simplicity of navigating the web application 1 - Very Satisfied 2 - Satisfied 3 - Not bad but not good 4 - Bad 5 - Very Bad	1	1	1	1	1	1	Done
6.5.3	User Experience: Functionality of the web application 1 - Very Satisfied 2 - Satisfied 3 - Not bad but not good 4 - Bad 5 - Very Bad	1	1	1	1	2	1	Done
Task 6: Feedback from users								
6.6.1	Do you have any feedback for us? It can be	Impressed with the navigation	Like the dropdown menu	Maybe have a filter or	Put the recovery state of	Maybe have a dropdown	It’s very easy to use.	Done

	anything like how we can improve on or what you don't like or like about our web application. Short Answers	bar and menu. Like that it is clean and easy to fill in details. Can consider adding search or filter functions.	where it updates when new medicine is added.	search function for all the lists. Easier to find something instead of using CTRL+F.	patients.	menu from symptoms. Please add a back button though.		

Table 6.2. Usability Testing Form Response

Upon reviewing user input and responses, the team concluded that while the web application is user-friendly, there is room for improvement to make the entire program more user-friendly. This is mainly due to the fact that most responses asked for the availability of search functionality on the lists features of the web application as with more data inserted into the database, it is more difficult to navigate within the list to find a certain data like a specific patient. The team would probably not be able to implement these features due to time constraints and each member's obligations from other units, but if given the opportunity, they will be taken into consideration for future modifications. Nonetheless, the web application for this project has passed all important usability tests.

7. Software Limitations

One of the most notable software drawbacks is that it occasionally takes a long time for handwritten information to be extracted from forms before it appears on the patient registration form in the online application. The team has not yet determined the cause of this limitation, other than the fact that it stems from an issue with the ICROP or HTR models. The team intends to test the web application on a desktop computer on campus with a dedicated graphics card and higher RAM in order to determine whether the problem is due to the model's complexity or a device limitation. If it is found to be a model complexity issue, the team’s dedicated model sub-team will try to find time to slowly dissect the ICROP and HTR model source code to allow for a faster processing time.

8. Possible Software Improvements

One of the most frequently mentioned improvements throughout the usability testing process is the possibility of implementing a search engine for the different lists that are currently included in the web application. It would be more advantageous and effective for users if there was a search engine available because they could look for specific information without wasting time reading through listings one by one.

Other than that, one feedback that particularly stood out to the team was the recommendation to add a new section or feature to the patient database that would clearly show each patient's state of recovery. This would give medical staff more thorough and up-to-date information regarding the patient to help the medical staff to monitor patients' recovery progress.

9. Testing Limitations

Everything has its limitations, and our testing procedure is no exception. One of the most noticeable limitations would be the targeted audience intended for our project. The team had previously acknowledged this issue, given that the primary end users of the web service are healthcare professionals such as nurses and physicians. Despite the fact that one of the team members has relatives who work in the healthcare sector, they were not able to ask the relative to test the web applications because they were located in a different state and had a very busy schedule. As a result, the team is unable to determine with precision if the feedback received accurately captures the needs of healthcare professionals as well as how user-friendly and successful the online application is in the healthcare sector.

Another limitation identified would be the limited scope in testing data. The forms that were used to test the handwritten text recognition aspect of the project were all filled in by the members of the team. This restricts the testing datasets to come from a small group of people. In addition to that, one of the team members created the form used in the web application, and the ICROP model fixed the coordinates on how the form should be cropped before feeding into the HTR model. Therefore, if a different form was used in the process, the entire handwritten text extraction process from the uploaded images would not function.

Hardware limitations were also one of the more noticeable limitations for our project. The team was unable to ascertain and validate the speed of text extraction from the uploaded forms due to insufficient RAM and the lack of a dedicated graphics card for each member's device. As a result, the processing speed and power for the ICROP and HTR model were significantly slowed when it was ran on the members' devices.

It should be noted that all testing, including usability testing, was carried out on each member's personal device; as a result, certain scenarios or tests might have escaped the team's notice and not been tested on. Because there are many things end users could do on web applications and it's unclear which actions end users could take to lead to the discovery of errors and bugs. The team does not have enough personnel to test every possible situation, including but not limited to what data was inputted into a particular input field.

10. Possible Testing Improvements

From the testing limitations listed above, the team has identified 3 possible improvements on how testing could be done for this project. First modification would be the target testing users. To obtain a more accurate feedback that will help improve the usability of the web application in the healthcare industry, the team will proactively change the target testing users to personnels that works in the healthcare industry which includes but not limited to physicians, nurses, and pharmacists. By involving potential end-users for our web application testing process, the team will be able to gain insights on the requirements and expectations for the web application. Currently, only one of the testing users is currently associated with the healthcare industry; this person is a student pharmacist. However, the feedback this user provides should be interpreted cautiously because, despite being a student pharmacist, the user lacks experience working in the healthcare industry, which makes it difficult for them to accurately understand the needs and requirements of the users who work in the industry.

Deploying the web application to the cloud before starting the testing process is another potential way to improve the testing process. In this manner, despite geographical limitations, the test can be carried out virtually via a video conference call, in which a single team member converses with the healthcare professional to carry out the testing procedure. The web application can also be run locally on the testers' devices in this manner to detect any potential compatibility problems early on before officially launching the web application into the healthcare industry.

Last potential improvement that could be done on the testing procedure is to test the web application on a computer or device with a higher computational power that has a dedicated graphics card and higher RAM. In this way, the team can conduct internal testing on the performance of the ICROP and HTR model on the web application before releasing the web application for usability testing.

11. Conclusion

Testing must be carried out correctly and thoroughly because it is an essential phase in the software development process. These tests ensure that users will experience trouble-free and error-free software. The tests involved for this project are black-box testing, integration testing, and usability testing approaches. Thorough testing techniques on the web application's functionality will ensure long-term user satisfaction and help the web application perform effectively in real-life situations.

12. References

Awati, R. (2022, January 13). integration testing or integration and testing (I&T). Software Quality. https://www.techtarget.com/searchsoftwarequality/definition/integration-testing#:~:text=Integration%20testing%20%2D%2D%20also%20known,tested%20as%20a%20combined%20entity.

Testlio. (2024, September 14). Integration Testing: Types, Processes, Examples & More. Testlio. https://testlio.com/blog/what-is-integration-testing/

Tasnim, R. (2024, August 12). Classification of defects in software testing. Software Testing Stuff. https://www.softwaretestingstuff.com/2008/05/classification-of-defects-bugs.html#types-of-defects-in-software-testing

Prescott, J., & Crichton, M. (1999). Usability testing. Proceedings of the 27th Annual ACM SIGUCCS Conference on User Services Mile High Expectations - SIGUCCS '99. https://doi.org/10.1145/337043.337137

Sfenrianto, Pitauli, M., & Wang, G. (2018). User satisfaction analysis for service-now application. IOP Conference Series: Materials Science and Engineering, 420, 012100. https://doi.org/10.1088/1757-899x/420/1/012100

What is Black Box Testing? (n.d.). Check Point Software. https://www.checkpoint.com/cyber-hub/cyber-security/what-is-penetration-testing/what-is-black-box-testing/

13. Acknowledgement

I acknowledge the use of Microsoft Copilot (https://copilot.microsoft.com/) and/or ChatGPT (https://chatgpt.com/) to generate materials for background research and self-study in the process of completing this assessment. I entered the following prompts on 5th October 2024:

- ➢ Explain what is testing methodology and how it affects testing approach for a web application.
- ➢ Explain and elaborate on what is the purpose of usability testing and why it is conducted.
- ➢ Explain the difference between black box testing and white box testing.

The generated output from the artificial intelligence was adapted, modified, and used for some of the final responses.