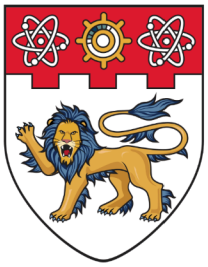


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

# SC2006 - Software Engineering

# Lab 4 Deliverables

| | |
|---|---|
| **Lab Group** | SCMA |
| **Team** | FetchMeHome |
| **Members** | BENJAMIN YEOH JUN JIE (U2321438C) |
| | BASKAR KAYALVIZHI ATHITHIYA (U2322885L) |
| | TIN JING LUN JAVIER (U2310906E) |
| | TRISTAN AMADEUS SURYA (U2320674L) |
| | TOMAR YASHWARDHAN SINGH (U2323694E) |
| | ZOU XUEHAO (U2322789L) |

# Table of Contents

# 1. Black Box Testing

## 1.1. Selected Control Class

The control class selected for testing will be the OllamaChat class (PersonalityController).

The OllamaChat class manages interaction with the Ollama API for generating AI responses, including sending prompts to the AI model and determining personality traits based on quiz answers.

## 1.2. Equivalence Class Testing

The **ask()** and **determinePersonality()** methods require discrete values as inputs. As such, Boundary Value Testing will not be applicable.

### 1.2.1. ask() Function

- Valid Equivalence Class: Prompt input is a non-empty string in proper format.
- Invalid Equivalence Class: Prompt input is empty, non-string value, or extremely long.

### 1.2.2. determinePersonality() Function

- Valid Equivalence Class: Quiz answers are provided as a well-formed object with all required question-answer pairs.
- Invalid Equivalence Class: Quiz answers object is incomplete (missing some required questions), empty object, or non-object value.

# 1.3. Test Cases and Testing Results

## 1.3.1. Test Cases for ask() Method

### Input Parameters:

1. Prompt

| Test Case Name | Test Input | Expected Output | Actual Output | Test Result |
|---|---|---|---|---|
| Ask-01 (Valid) | prompt: "What is the capital of France?" | A non-empty string response about Paris | A non-empty string response about Paris | Pass |
| Ask-02 (Invalid) | prompt: "" (Empty string) | "Error fetching response from AI." | "Error fetching response from AI." | Pass |
| Ask-04 (Invalid) | prompt: "a".repeat(10000) (Extremely long string) | error message | error message | Pass |
| Ask-05 (Invalid) | prompt: null | "Error fetching response from AI." | "Error fetching response from AI." | Pass |
| Ask-06 (Invalid) | prompt: undefined | "Error fetching response from AI." | "Error fetching response from AI." | Pass |

### 1.3.2. Test Cases for determinePersonality() Method

**Input Parameters:**

1. Prompt (Quiz answers by users)

| Test Case Name | Test Input | Expected Output | Actual Output | Test Result |
|---|---|---|---|---|
| Personality-01 (Valid) | quizAnswers: {"Question 1": "Option A", "Question 2": "Option B", "Question 3": "Option C"} | Three comma-separated personality traits (e.g., "Adventurous, Active, Relaxed") | Three comma-separated personality traits | Pass |
| Personality-02 (Invalid) | quizAnswers: {} (Empty object - user didn't fill any options) | "Unknown, Please, Retry" | "Unknown, Please, Retry" | Pass |
| Personality-03 (Invalid) | quizAnswers: {"Question 1": "Option A"} (Incomplete - user only filled 1 of 3 questions) | "Unknown, Please, Retry" | "Unknown, Please, Retry" | Pass |

| | | | | |
|---|---|---|---|---|
| Personality-04 (Invalid) | quizAnswers: {"Question 1": "Option A", "Question 2": "Option B"} (Incomplete - user only filled 2 of 3 questions) | Three comma-separated personality traits (possibly less accurate) | Three comma-separated personality traits | Pass |
| Personality-05 (Invalid) | quizAnswers: null (System error) | "Unknown, Please, Retry" | "Unknown, Please, Retry" | Pass |

# 2. White Box Testing

The 2 methods selected for white box testing will be:

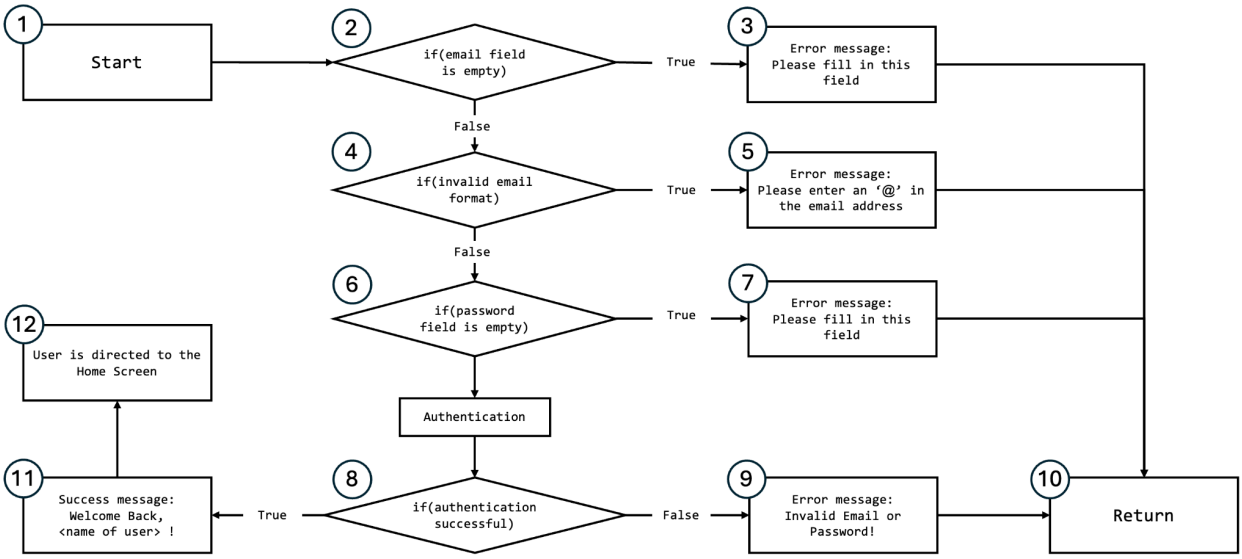
1. Login
2. PostLostPetListing

The **Login** function is responsible for authenticating users by validating their email and password inputs. It checks for missing fields, ensures the email format is correct, verifies the user's existence in the database, and compares the input password with the stored password hash. Upon successful validation, it generates and returns an authentication token for the user to log into the registered account.

The **PostLostLetListing** function enables users to submit a report for a lost pet. It performs a series of validations to ensure that all required fields are filled, the uploaded image is valid, the description does not exceed a set word limit, and both phone number and email formats are correct. Upon verification that all fields are filled up correctly, it publishes the listing for all users to view.

# 2.1. Login

## 2.1.1. Control Flow Graph




## 2.1.2. Cyclomatic Complexity

Login Method

Cyclomatic Complexity (CC) = | binary decision points | + 1 = | 4 | + 1 = 5

## 2.1.3. Basis Paths

### Login Method Basis Paths

- Basis Path #1 (Baseline): 1 → 2 → 4 → 6 → 8 → 11 → 12 (Successful Login)
- Basis Path #2: 1 → 2 → 3 → 10 (Email field empty)
- Basis Path #3: 1 → 2 → 4 → 5 → 10 (Invalid email format)
- Basis Path #4: 1 → 2 → 4 → 6 → 7 → 10 (Password field empty)
- Basis Path #5: 1 → 2 → 4 → 6 → 8 → 9 → 10 (Failed authentication)

### 2.1.4. Test Cases and Testing Results

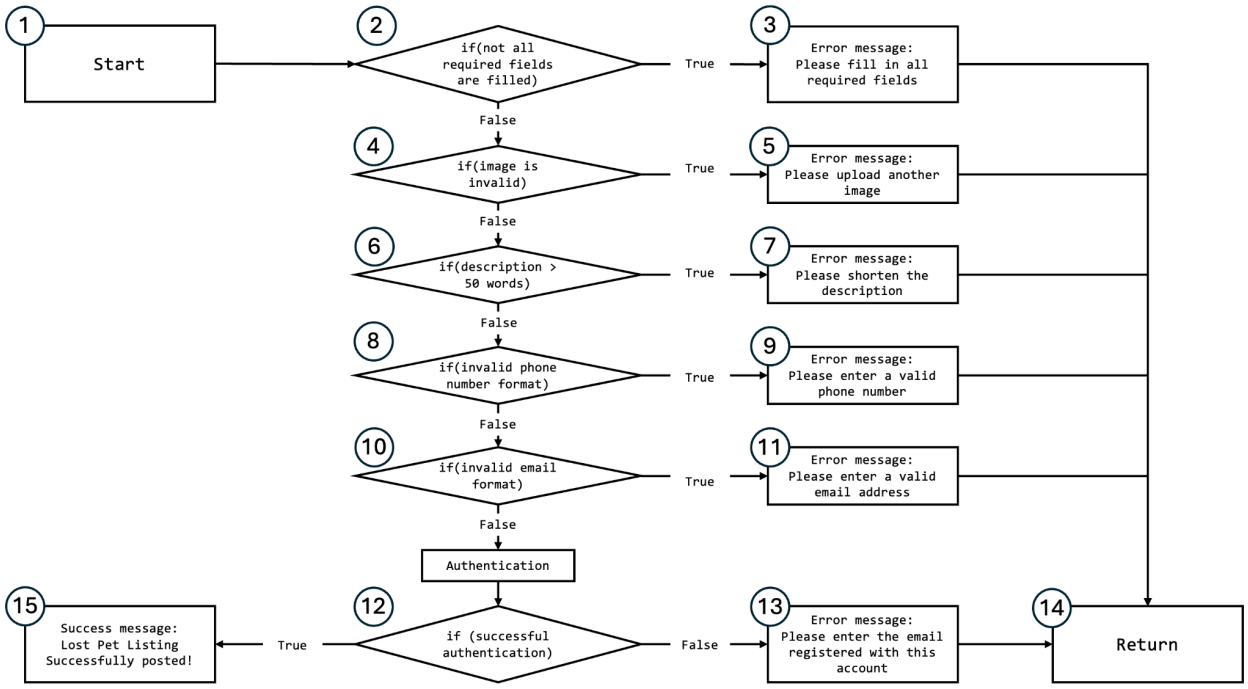
**Input Parameters:**

1. Email
2. Password

| No. | Test Input | Expected Output | Actual Output | Test Result |
|---|---|---|---|---|
| **1** | Email: admin@gmail.com<br>Password: admin | “Welcome back, Admin!” | “Welcome back, Admin!” | Pass |
| **2** | Email:<br>Password: admin | “Please fill in this field!” | “Please fill in this field!” | Pass |
| **3** | Email: admin<br>Password: admin | “Please include an ‘@’ in the email address” | “Please include an ‘@’ in the email address” | Pass |
| **4** | Email: admin@gmail.com<br>Password: | “Please fill in this field!” | “Please fill in this field!” | Pass |
| **5** | Email: xyz@gmail.com<br>Password: admin | “Invalid email or password!” | “Invalid email or password!” | Pass |
| **6** | Email: admin@gmail.com<br>Password: xyz | “Invalid email or password!” | “Invalid email or password!” | Pass |

## 2.2. PostLostPetListing

### 2.2.1. Control Flow Graph




### 2.2.2. Cyclomatic Complexity

Cyclomatic Complexity (CC) = | binary decision point | + 1 = | 6 | + 1 = 7

### 2.2.3. Basis Paths

#### handleAdoptionRequest Method Basis Paths

Basis Path #1 (Baseline): 1 → 2 → 4 → 6 → 8 → 12 → 15 (Successful posting)

Basis Path #2: 1 → 2 → 3 → 14 (Required fields not filled)

Basis Path #3: 1 → 2 → 4 → 5 → 14 (Invalid image)

Basis Path #4: 1 → 2 → 4 → 6 → 7 → 14 (Description too long)

Basis Path #5: 1 → 2 → 4 → 6 → 8 → 9 → 14 (Invalid phone)

Basis Path #6: 1 → 2 → 4 → 6 → 8 → 10 → 11 → 14 (Invalid email)

Basis Path #7: 1 → 2 → 4 → 6 → 8 → 10 → 12 → 13 → 14 (Invalid authentication)

## 2.2.4. Test Cases and Testing Results

**Input Parameters:**

1. PetName
2. PetAge
3. PetType
4. PetPhoto
5. LastSeenLocation
6. Description
7. Email
8. PhoneNumber

| No. | Test Input | Expected Output | Actual Output | Test Result |
|---|---|---|---|---|
| **1** | Pet Name: TestPet<br>Pet Age: 1<br>Pet Type: Dog<br>Pet Photo: <Valid Image><br>Last Seen Location: <Location><br>Description: Small<br>Email: User@gmail.com<br>Phone Number: 88888888 | “Lost Pet Listing Successfully Posted” | “Lost Pet Listing Successfully Posted” | Pass |
| **2** | Pet Name:<br>Pet Age:<br>Pet Type:<br>Pet Photo:<br>Last Seen Location:<br>Description:<br>Email:<br>Phone Number: | “Please fill in all required fields” | “Please fill in all required fields” | Pass |

| | | | | |
|---|---|---|---|---|
| **3** | Pet Name: TestPet<br>Pet Age: 1<br>Pet Type: Dog<br>Pet Photo: <INVALID Image><br>Last Seen Location: <Location><br>Description: Small<br>Email: User@gmail.com<br>Phone Number: 88888888 | “Please upload another image” | “Please upload another image” | Pass |
| **4** | Pet Name: TestPet<br>Pet Age: 1<br>Pet Type: Dog<br>Pet Photo: <Valid Image><br>Last Seen Location: <Location><br>Description: <A description longer than 50 words><br>Email: User@gmail.com<br>Phone Number: 88888888 | “Please shorten the description” | “Please shorten the description” | Pass |
| **5** | Pet Name: TestPet<br>Pet Age: 1<br>Pet Type: Dog<br>Pet Photo: <Valid Image><br>Last Seen Location: <Location><br>Description: <A description longer than 50 words><br>Email: admin@gmail.com<br>Phone Number: zzz | “Please enter a valid phone number” | “Please enter a valid phone number” | Pass |
| **6** | Pet Name: TestPet<br>Pet Age: 1 | “Please enter a valid email address” | “Please enter a valid email address” | Pass |

| | | | | |
|---|---|---|---|---|
| | Pet Type: Dog<br>Pet Photo: <Valid Image><br>Last Seen Location: <Location><br>Description: <A description longer than 50 words><br>Email: User<br>Phone Number: 88888888 | | | |
| **7** | Pet Name: TestPet<br>Pet Age: 1<br>Pet Type: Dog<br>Pet Photo: <Valid Image><br>Last Seen Location: <Location><br>Description: <A description longer than 50 words><br>Email: notUser@gmail.com<br>Phone Number: 88888888 | “Please enter the email address registered with this account” | “Please enter the email address registered with this account” | Pass |