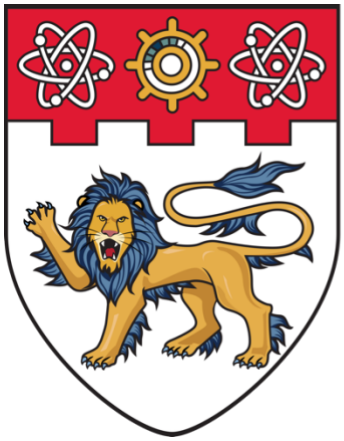


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

SC2006 – Software Engineering

Lab 4: Implementation, Testing, and Preparation for Demo

Lab Group	SCMB
Team	SPLIIT
Members	Goh Jun Keat
	Koh Ze Kai Leo
	Madhumita Thiruppathi
	Siah Yee Long
	Teo Liang Wei, Ryan
	Zhang Yichi

Table of Contents

Black Box Testing

Control class to test – **Login & Register Controller**

The **Login & Register Controller** is responsible for managing user authentication processes within the system. This includes two key functionalities: user registration and user login. These processes handle essential user data such as email, username, password, and display name, as well as ensure proper validation, data security, and storage in the database.

During **user registration**, the user must provide the following information: Email**,** Username**,** Password**,** Display and Preferred Colour

Once the user submits their registration form, the system performs several validation checks:

- **Email Validation**: The email entered must follow a valid email format (e.g., user@example.com). It should not be blank and must be unique within the system.
- **Username Validation**: The username should be unique, non-empty, and free of whitespace. Additionally, the system ensures that the username is not already registered.
- **Password Validation**: The password must meet the system's minimum-security criteria (e.g., length, character requirements) and should not contain whitespace.
- **Display Name Validation**: The display name should not be empty, ensuring the user has provided an identity for display purposes.
- **Preferred Colour**: The user must select a preferred colour from a predefined set of options.

If all validations pass, the system hashes the user's password before storing it in the database along with the other user information. The hashed password ensures that sensitive information is securely stored, protecting user privacy.

Once the user is successfully registered, they are either logged in automatically or redirected to the login screen, where they can enter their credentials to gain access.

During **user login**, the user is required to provide the following information: Username, Password

Once the user submits their login credentials, the system performs the following checks:

- **Username Validation**: The system checks whether the provided username exists in the database.
- **Password Validation**: If the username exists, the system compares the entered password (after hashing) with the hashed password stored in the database.

- **Successful Login**: If the username and password match, the user is authenticated and logged in.
- **Unsuccessful Login**: If either the username does not exist or the password does not match, the user is presented with an error message (e.g., "Invalid username or password.") and prompted to try again.

Equivalence Class and Boundary Value Testing

Equivalence Class Testing

Equivalence Class Testing is a technique used in black-box testing where the input data is divided into groups, known as equivalence classes. Each equivalence class represents a set of values that are treated in the same way by the system. The main idea behind this approach is that if the system handles one value from an equivalence class correctly, it is assumed to handle all other values from that class the same way. Therefore, instead of testing every individual value, a representative value from each class is selected for testing.

These classes are divided into two categories:

- **Valid equivalence classes**: These represent correct or acceptable inputs that the system is expected to process correctly.
- **Invalid equivalence classes**: These represent incorrect or unacceptable inputs that the system should reject or handle with an error.

This technique helps in reducing the number of test cases while ensuring comprehensive test coverage by targeting the different groups of inputs the system will encounter.

Boundary Value Testing

Boundary Value Testing is a specialized form of testing that targets the boundaries or limits of input ranges. It extends the idea of equivalence partitioning by focusing specifically on the values at the edges of the defined boundaries, as these often represent critical cases for the system. The login and signup process requires discrete values as inputs. As such, Boundary Value Testing will not be applicable.

Register Function

- **Valid Equivalence Class:** All required fields (Email, Username, Password, Display name and Favourite colour) are correctly filled and username does not match an existing user.
- **Invalid Equivalence Class:** Missing fields (Email, Username, Password, Display name and Favourite colour), invalid email format, the password length is less than 8. or user with the same username already exists.

Login Function

- **Valid Equivalence Class:** Username and password input values are in correct formats, length and match an existing user.
- **Invalid Equivalence Class:** Username and password input values are in incorrect formats, length or do not match an existing user.

Test Cases and Results

a) Register

Input Parameters:

1) Email
2) Username
3) Password
4) Display name
5) Favourite colour

No.	Test Input	Expected Output	Actual Output	Pass?
1	**(Valid)** Email: "testuser@gmail.com" **(Valid)** Username: "testuser123" **(Valid)** Password: "testpassword123" **(Valid)** Display Name: "Test User" **(Valid)** Favourite Colour: "Blue"	Register Success	Register Success	Pass
2	**(Invalid)** Email: "invalid-email" **(Valid)** Username: "testuser123" **(Valid)** Password: "testpassword123" **(Valid)** Display Name: "Test User" **(Valid)** Favourite Colour: "Blue"	Register Failed: "Please enter a valid email address."	Register Failed: "Please enter a valid email address."	Pass
3	**(Valid)** Email: "testuser@gmail.com" **(Invalid)** Username: "testuser123" (already exists) **(Valid)** Password: "testpassword123" **(Valid)** Display Name: "Test User" **(Valid)** Favourite Colour: "Blue"	Register Failed: "Please enter a unique username."	Register Failed: "Please enter a unique username."	Pass
4	**(Invalid)** Email: "testuser@gmail.com" (already exists) **(Valid)** Username: "testuser123" **(Valid)** Password: "testpassword123" **(Valid)** Display Name: "Test User" **(Valid)** Favourite Colour: "Blue"	Register Failed: "This email is already registered."	Register Failed: "This email is already registered."	Pass
5	**(Valid)** Email: "testuser@gmail.com" **(Valid)** Username: "testuser123" **(Invalid)** Password: "short" **(Valid)** Display Name: "Test User" **(Valid)** Favourite Colour: "Blue"	Register Failed: " Password must have at least 8 characters"	Register Failed: " Password must have at least 8 characters"	Pass

6	**(Valid)** Email: "testuser@gmail.com" **(Valid)** Username: "testuser123" **(Valid)** Password: "testpassword123" **(Invalid)** Display Name: "" **(Valid)** Favourite Colour: "Blue"	Register Failed: "Fill out this field"	Register Failed: "Fill out this field"	Pass
7	**(Valid)** Email: "testuser@gmail.com" **(Valid)** Username: "testuser123" **(Valid)** Password: "testpassword123" **(Valid)** Display Name: "Test User" **(Invalid)** Favourite Colour: ""	Register Failed: "Please select a favourite colour."	Register Failed: "Please select a favourite colour."	Pass

b) Login

Input Parameters:

1) Username
2) Password

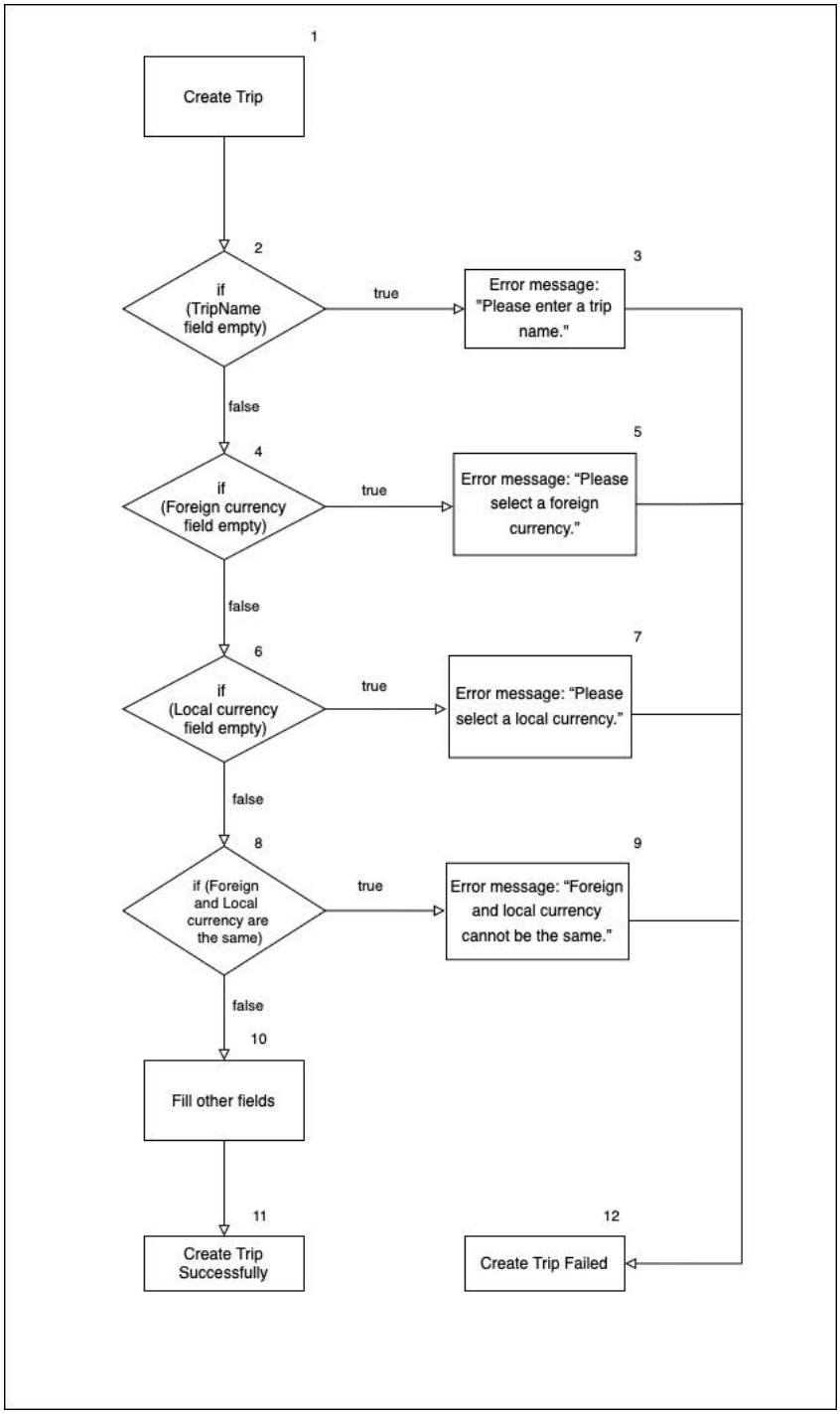
No.	Test Input	Expected Output	Actual Output	Pass?
1	**(Valid)** Username: "testuser123" **(Valid)** Password: "testpassword123"	Login Success	Login Success	Pass
2	**(Invalid)** Username: "invaliduser" **(Valid)** Password: "testpassword123"	Login Failed: “Invalid Username or Password.”	Login Failed: “Invalid Username or Password.”	Pass
3	**(Valid)** Username: "testuser123" **(Invalid)** Password: "wrongpassword"	Login Failed: “Invalid Username or Password.”	Login Failed: “Invalid Username or Password.”	Pass
4	**(Invalid)** Username: "" **(Valid)** Password: "testpassword123"	Login Failed: "Fill out this field"	Login Failed: "Fill out this field"	Pass
5	**(Valid)** Username: "testuser123" **(Invalid)** Password: ""	Login Failed: "Fill out this field"	Login Failed: "Fill out this field"	Pass
6	**(Invalid)** Username: "" **(Invalid)** Password: ""	Login Failed: "Fill out this field"	Login Failed: "Fill out this field"	Pass
7	**(Valid)** Username: "testuser123" **(Invalid)** Password: "short"	Login Failed: "Password	Login Failed: "Password	Pass

		should contain at least 8 characters."	should contain at least 8 characters."	

White Box Testing

Create Trip

Control Flow Graph



Basic Path Testing

Basis Path #1: 1, 2, 3, 12
Basis Path #2: 1, 2, 4, 5, 12
Basis Path #3: 1, 2, 4, 6, 7, 12

Basis Path #4: 1, 2, 4, 6, 8, 9, 12
Basis Path #5: 1, 2, 4, 6, 8, 10, 11

Test Cases and Results

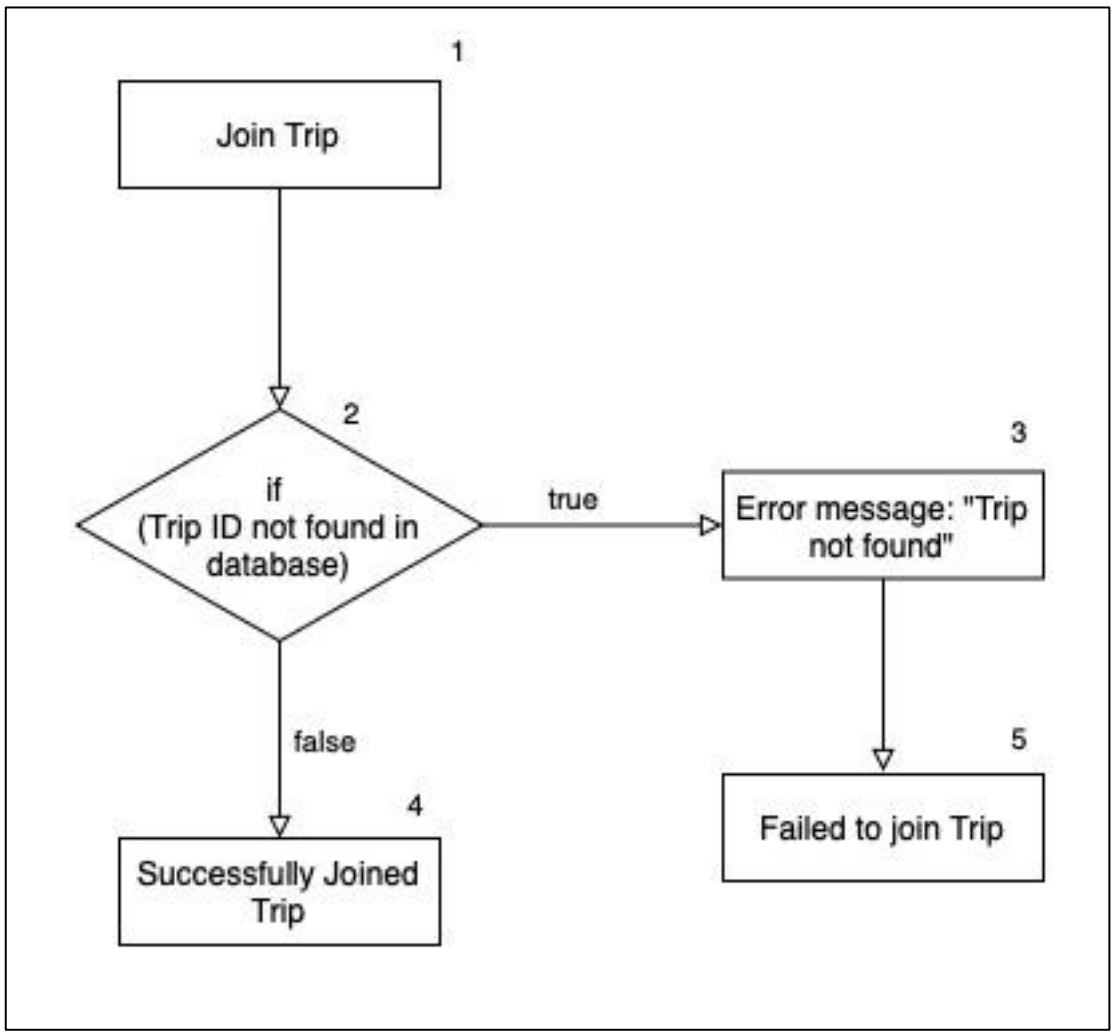
Input Parameters:

1) Trip Name
2) Foreign Currency
3) Local Currency
4) Other fields

No.	Test Input	Expected Output	Actual Output	Pass?
1	Trip Name = "" Foreign Currency = USD Local Currency = CNY Other fields = Filled	Error: "Please enter a trip name."	Error: "Please enter a trip name."	Pass
2	Trip Name = "Business Trip" Foreign Currency = Local Currency = CNY Other fields = Filled	Error: "Please select a foreign currency."	Error: "Please select a foreign currency."	Pass
3	Trip Name = "Business Trip" Foreign Currency = USD Local Currency = Trip ID = 12345 Other fields = Filled	Error: "Please select a local currency."	Error: "Please select a local currency."	Pass
4	Trip Name = "Business Trip" Foreign Currency = USD Local Currency = USD Other fields = Filled	Error: "Foreign and local currency cannot be the same."	Error: "Foreign and local currency cannot be the same."	Pass
5	Trip Name = "Business Trip" Foreign Currency = USD Local Currency = CNY Other fields = Filled	Create Trip Successfully	Create Trip Successfully	Pass

Join Trip

Control Flow Graph



Basic Path Testing

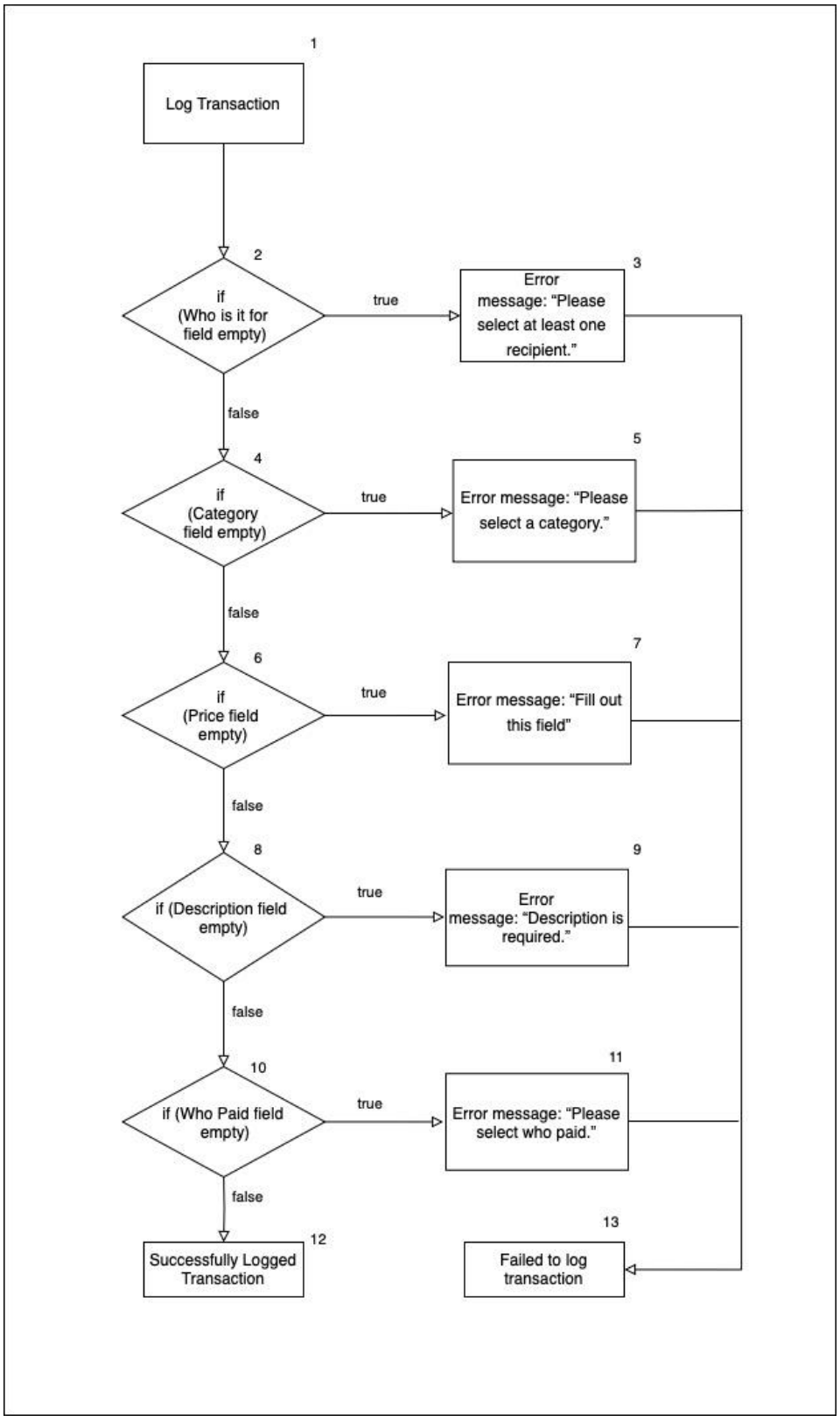
Basis Path #1: 1, 2, 3, 5
Basis Path #2: 1, 2, 4

Test Cases and Results

No.	Test Input	Expected Output	Actual Output	Pass?
1	Trip ID = RBNKS (not an existing Trip ID)	Error: “Trip not found.”	Error: “Trip not found.”	Pass
2	Trip ID = J22M3K	Successfully joined trip	Successfully joined trip	Pass

Log Transaction

Control Flow Graph



Basic Path Testing

Basis Path #1: 1, 2, 3, 13
Basis Path #2: 1, 2, 4, 5, 13
Basis Path #3: 1, 2, 4, 6, 7, 13
Basis Path #4: 1, 2, 4, 6, 8, 9, 13
Basis Path #5: 1, 2, 4, 6, 8, 10, 11, 13

Test Cases and Results

Input Parameters:

1) Who is it for
2) Category
3) Price
4) Description
5) Who Paid

No.	Test Input	Expected Output	Actual Output	Pass?
1	Who is it for = Category = food Price = “100” Description = “Lunch” Who Paid = user123	Error: “Please select at least one recipient.”	Error: “Please select at least one recipient.”	Pass
2	Who is it for = user123 Category = Price = “100” Description = “Lunch” Who Paid = user123	Error: “Please select a category.”	Error: “Please select a category.”	Pass
3	Who is it for = user123 Category = food Price = "" Description = “Lunch” Who Paid = user123	Error: “Fill out this field”	Error: “Fill out this field”	Pass
4	Who is it for = user123 Category = food Price = “100” Description = "" Who Paid = user123	Error: “Description is required.”	Error: “Description is required.”	Pass
5	Who is it for = user123 Category = food Price = 100 Description = Lunch Who Paid =	Error: “Please select who paid.”	Error: “Please select who paid.”	Pass

6	Who is it for = user123 Category = food Price = 100 Description = Lunch Who Paid = user123	Successfully logged Transaction	Successfully logged Transaction	Pass