

**NANYANG
TECHNOLOGICAL
UNIVERSITY**
SINGAPORE

SC2006 - Software Engineering

Bin Buddy - An E-waste Recycling App

Lab 4 Deliverables

Lab Group	SCSI
Team	Group 3
Title	Bin Buddy
Member	HTOO MYAT NOE (U2422977K)
	ARMAN KHAN (U2421760A)
	CHUA YUE JUN (U2423015D)
	NIKHIL MADETI (U2421243D)
	SOH CEK CONG (U2423500C)
	YOONG HONG JUN, NICHOLAS (U2321582L)

Table of Contents

1. Black Box Testing	3
1.1 RegistrationController	3
1.1.1 Equivalence and Boundary Value Testing	4
1.1.2 Black Box Testing and Results	5
2. White Box Testing	7
2.1 Login	7
2.1.1 Control Flow Diagram	7
2.1.2 Basis Path Testing	8
2.1.3 Test Cases and Results	8
3.1 Change Password	10
3.1.1 Control Flow Diagram	11
3.1.2 Basis Path Testing	11
3.1.3 Test Cases and Results	12

1. Black Box Testing

For our black box testing, we have selected **RegistrationController**. RegistrationController is in charge of all the logic for new user registration.

1.1 RegistrationController

The main purpose of RegistrationController is to ensure proper user registration. When users register for the application, they are allowed to sign up by entering their full name, username, email address and an optional admin code if they were provided one by the administrator. The controller will firstly validate the inputs to ensure that all fulfill the necessary requirements. For instance, the password that the user inputs must have at least 6 characters with 1 number and 1 special character.

1.1.1 Equivalence and Boundary Value Testing

Equivalence Class Testing

Equivalence class testing involves finding valid and invalid equivalence classes that partition the input values for a specific use case. This aims to reduce the number of test cases needed while maintaining thorough test coverage. For this process therefore, at least 1 input value for each equivalence class of each input parameter must be chosen such that the testing coverage is sufficiently thorough. While testing valid inputs, it is important to test valid inputs of several parameters at the same time. Conversely, when testing invalid inputs of various parameters, only one invalid input from an equivalence class of one input parameter should be tested at the same time.

This approach therefore serves to comprehensively cover the scope of testing without requiring individual tests for every single possible permutation of inputs.

Boundary Value Testing

Boundary value testing builds upon equivalence class testing. This approach is only applicable to a range of values, where there is a focus on the boundaries of the equivalence classes of a range of values. In doing so, a minimal set of valid and invalid inputs including values on the boundary, just below, or just above can be used to test the system. Discrete values on the other hand have no boundary values.

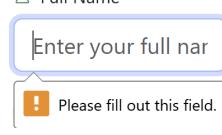
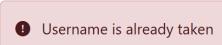
Valid and Invalid Equivalence Classes

Valid Equivalence Class: Full Name, Username, Email Address, Password, Confirm Password fields are all valid.

Invalid Equivalence Class: Full Name, Username, Email Address, Password, Confirm Password fields with invalid formats or incorrect length

1.1.2 Black Box Testing and Results

Input Parameters: Full Name, Username, Email Address, Password, Confirm Password

No.	Test Input	Expected Output	Actual Output	Expected Result
1.	(All Valid Inputs) Name: Tan Tan Username: Lim123 Email Address: binbuddy3@gmail.com Password: 12345! Confirm Password: 12345!	You are now registered and can log in	 You are now registered and can log in	Pass
2	(All Valid Inputs except Name) Name: ""	App will highlight the “Full Name” field and ask the user to fill in the field	 App will highlight the “Full Name” field and ask the user to fill in the field	Reject
3	(All Valid Inputs except Username) Username: Lim123 (already registered)	Username is already taken	 Username is already taken	Reject
4	(All Valid Inputs except Username) Username: Li (Length is 1-2 characters)	Please lengthen this text to 3 characters or more (you are currently using xx characters)	Please lengthen this text to 3 characters or more (you are currently using xx characters)	Reject
5	(All Valid Inputs except Username) Username: "" (Length is 0 characters)	Please fill out this field	Please fill out this field	Reject
6	(All Valid Inputs except Email) Email: binbuddy3 (no @ in input)	Please include an '@' in the email address. 'binbuddy3' is missing an '@'.	Please include an '@' in the email address. 'binbuddy3' is missing an '@'.	Reject
7	(All Valid Inputs except Password) Password: 12345 (less	Please match the requested format	 Please match the requested format	Reject

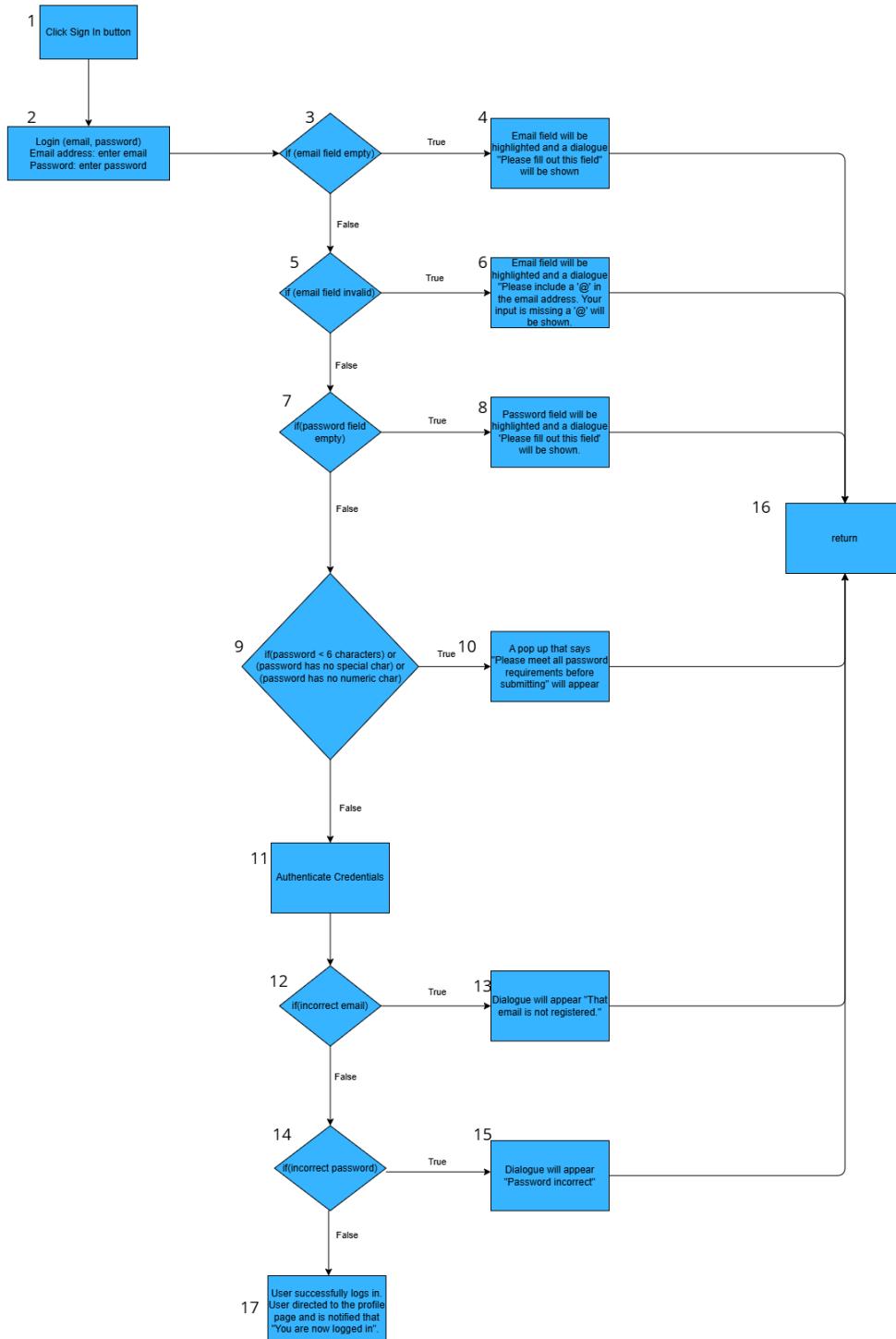
	than 6 characters)			
8	<p>(All Valid Inputs except Confirm Password)</p> <p>Password: 12345! Confirm Password: 12345</p>	<p>Passwords do not match.</p>	<p>! Passwords do not match Passwords do not match dialogue shows up</p>	Reject

2. White Box Testing

2.1 Login

We have chosen our log in method as one of the methods to test.

2.1.1 Control Flow Diagram



2.1.2 Basis Path Testing

Cyclomatic Complexity: |decision points|+ 1 = 6+1=7

Basis Paths

Baseline Path: 1,2,3,5,7,9,11,12,14,17

Basis Path 2: 1,2,3,4,16

Basis Path 3: 1,2,3,5,6,16

Basis Path 4: 1,2,3,5,7,8,16

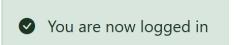
Basis Path 5: 1,2,3,5,7,9,10,16

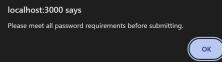
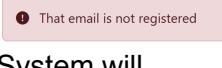
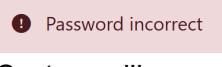
Basis Path 6: 1,2,3,5,7,9,11,12,13,16

Basis Path 7: 1,2,3,5,7,9,11,12,13,14,15,16

2.1.3 Test Cases and Results

Input Parameters: Email Address, Password

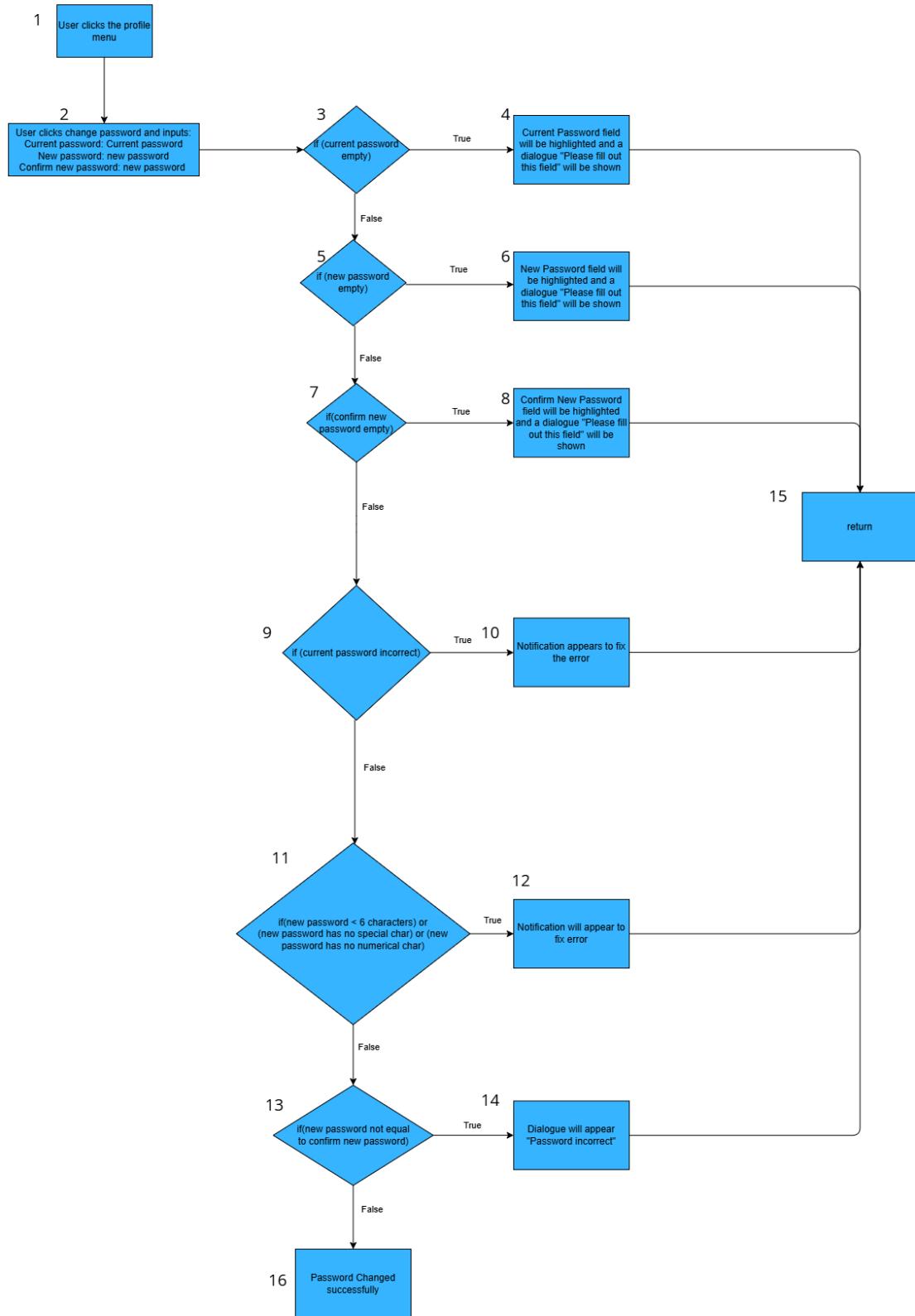
No.	Test Input	Expected Output	Actual Output	Expected Result
1.	(All Valid Inputs) Email Address: binbuddy3@gmail.com Password: 12345!	You are now logged in	 You are now registered and can log in	Pass
2	(All Valid Inputs except Email) Email Address: ""	App will highlight the "Email Address" field and ask the user to fill in the field	 App will highlight the "Email Address" field and ask the user to fill in the field	Reject
3	(All Valid Inputs except Email) Email Address: binbuddy3	Please include an '@' in the email address. 'binbuddy3' is missing an '@'.	Please include an '@' in the email address. 'binbuddy3' is missing an '@'.	Reject
4	(All Valid Inputs except Password) Password: ""	App will highlight the "Password" field and ask the user to fill in the field	 App will highlight the "Password" field and ask the user to fill in the field	Reject

5	<p>(All Valid Inputs except Password)</p> <p>Password: 12345 (less than 6 characters) Password: 123456 (no special character) Password: abcde! (no numeric character)</p>	Please meet all password requirements before submitting	 Please meet all password requirements before submitting	Reject
6	<p>(All Valid Inputs except Email)</p> <p>Email Address: 123456@gmail.com (unregistered email)</p>	System will output "That email is not registered"	 System will output "That email is not registered"	Reject
7	<p>(All Valid Inputs except Password)</p> <p>Email Address: binbuddy3@gmail.com</p> <p>Password: 12345& (wrong password)</p>	System will output "Password incorrect"	 System will output "Password incorrect"	Reject

3.1 Change Password

The next functionality that will be tested is the user's ability to change their own password.

3.1.1 Control Flow Diagram



3.1.2 Basis Path Testing

Cyclomatic Complexity: |decision points|+ 1 = 6+1=7

Basis Paths

Baseline Path: 1,2,3,5,7,9,11,13,16

Basis Path 2: 1,2,3,4,15

Basis Path 3: 1,2,3,5,6,15

Basis Path 4: 1,2,3,5,7,8,15

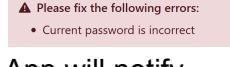
Basis Path 5: 1,2,3,5,7,9,10,15

Basis Path 6: 1,2,3,5,7,9,11,12,15

Basis Path 7: 1,2,3,5,7,9,11,13,14,15

3.1.3 Test Cases and Results

Input Parameters: Current Password, New Password, Confirm New Password

No.	Test Input	Expected Output	Actual Output	Expected Result
1.	(All Valid Inputs) Current Password: 12345! New Password: 12345^ Confirm New Password: 12345^	Password changed successfully		Pass
2	(All Valid Inputs except Current Password) Current Password: ""	App will highlight the "Current Password" field and ask the user to fill in the field	 App will highlight the "Current Password" field and ask the user to fill in the field	Reject
3	(All Valid Inputs except Current Password) Current Password: 12345 (wrong match)	App will notify "Please fix the following errors: Current password is incorrect"	 App will notify "Please fix the following errors: Current password is incorrect"	Reject

4	(All Valid Inputs except New Password) New Password:	App will highlight the “New Password” field and ask the user to fill in the field	<p>App will highlight the “New Password” field and ask the user to fill in the field</p>	Reject
5	(All Valid Inputs except New Password) Password: 12345 (less than 6 characters) Password: 123456 (no special character) Password: abcde! (no numeric character)	App will notify “Please fix the following errors: New password must be at least 6 characters and contain at least 1 number and 1 special character”	<p>App will notify “Please fix the following errors: New password must be at least 6 characters and contain at least 1 number and 1 special character”</p>	Reject
6	(All Valid Inputs except Confirm New Password) Confirm New Password: “”	App will highlight the “Confirm New Password” field and ask the user to fill in the field	<p>App will highlight the “Confirm New Password” field and ask the user to fill in the field</p>	Reject
7	(All Valid Inputs except Confirm New Password) New Password: 12345\$ Confirm New Password: 12345 (does not match with new password)	System will output “Error Changing Password”	<p>System will output “Error Changing Password”</p>	Reject