



**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

**SC2006 Software Engineering
Lab 4 Deliverables**

Dream Neighbourhood

Group Member	Matric Number
How En Hsin	U2421325J
Veda Ho Yong Qian	U2321709B
Jarrel Ng Tze Xuan	U2421691L
Isaac Teo Yi Jay	U2421749A
Sean Winston Susanto	U2423834G
Joel Ng Wei Heng	U2422193L

Table of Contents

Testing	3
Black Box Tests	3
White Box Tests	4
API Testing	5

Testing

Black Box Tests

Test Cases

a) Login

Input parameters: Email address and password

ID	Test Input	Expected Output	Actual Output	Result
1	<i>Valid</i> email: "user1@gmail.com" <i>Valid</i> password: "Password123"	Successful login, redirects to home page	Redirects to home page	Pass
2	<i>Invalid</i> email: "useergmail.com" <i>Valid</i> password: "Password123"	Error message: Invalid Credentials	Error message: Invalid Credentials	Pass
3	<i>Valid</i> email: "user1@gmail.com" <i>Invalid</i> password: ""	Error message: Please fill in password field	Error message: Please fill in password field	Pass
4	<i>Valid</i> email: "user1@gmail.com" <i>Invalid</i> password: "pAssword123"	Error message: Invalid Credentials	Error message: Invalid Credentials	Pass
5	<i>Invalid</i> email: "useergmail.com" <i>Invalid</i> password: "pAssword123"	Error message: Invalid Credentials	Error message: Invalid Credentials	Pass

b) Sign-up

ID	Test Input	Expected Output	Actual Output	Result
6	Valid username: "test" Valid email: "testuser@gmail.com" Valid password: "Password" Confirm password: "Password"	Successful Sign up. Success message: Check email to verify account	Success message: Check email to verify email	Pass

7	Valid username: "Test" Valid email: "testuser@gmail.com" Valid password: "Password" Confirm password: "Passssssword"	Error message: Passwords do not match	Error message: Passwords do not match	Pass
8	Invalid email. Email already registered.	Error message: Email already exists	Error message: Email already exists	Pass
9	Name already in database, Valid email: "newuser@mail.com" Valid Password: "Password "	Successful sign up. Success message: Check email to verify account	Success message: Check email to verify account	Pass
10	Password already in database, new username and email. Valid email: "newuser@mail.com"	Successful sign up. Success message: Check email to verify account	Success message: Check email to verify account	Pass

White Box Tests

Test Cases

a) Authentication Middleware

ID	Test Input	Expected Output	Actual Output	Result
1	Valid JWT Token: "jwt-token"	next() function is called	next() function is called	Pass
2	Invalid JWT Token: "mock-jwt-token"	Res status code 403: "Invalid Token"	Res status code 403: "Invalid Token"	Pass
3	Missing JWT Token: ""	Res status code 401: "Invalid Token"	Res status code 401: "Invalid Token"	Pass

b) Database Queries

Mocking the implementation of our backend database configuration file to test operations on database queries. The table that is used to test is an empty table (rows = 0)

ID	Test Input	Expected Output	Actual Output	Result
4	Insert mockUser: { Id: 1, Username: 'prophunt', Email: prop@hunt.com , Password: 'hashedpassword123', Is_verified: true }	Number of rows == 1	Number of rows == 1	Pass
5	Select { Email: 'prop@hunt.com' }	Number of rows = 1 (number of rows remains unchanged), Selected username == mockUser.username	Number of rows == 1, Username == mockUser.username resolves to true	Pass

API Testing

ID	Method	Endpoint	Date Tested		
6	POST	/api/v1/search/rank-properties	2/11/2025		
Description	Testing valid input in request body and parameters. Valid input includes a dictionary of amenity types and their rankings in the request body, and the name of a town in the request query.				
Request			Response		
Headers	Body	Query	Expected	Actual	Pass/Fail
nil	{ amenityTheme: 'school', rank: 1, maxDistance: 1000 }, { amenityTheme:	Town: 'TAMPINES'	Response code 200	Response code 200 Returns ranked properties in JSON format	Pass

	{ 'park', rank: 2, maxDistance: 1000 }				
--	--	--	--	--	--

Search and Filtering Tests

ID	Method	Endpoint	Date Tested		
7	POST	/api/v1/search/rank-properties	2/11/2025		
Description	Testing invalid input if town is not specified in request query				
Request			Response		
Headers	Body	Query	Expected	Actual	Pass/Fail
nil	{ amenityTheme: ‘school’, rank: 1, maxDistance: 1000 }	nil	Response code 400	Response code 400 Error message: “town is required”	Pass

ID	Method	Endpoint	Date Tested		
8	POST	/api/v1/search/rank-properties	2/11/2025		
Description	Testing invalid input if the ranking array is not included in the request body.				
Request			Response		
Headers	Body	Query	Expected	Actual	Pass/Fail
nil	nil	Town: "TAMPINES"	Response code 400	Response code 400 Error message: "ranking array is required in body"	Pass

ID	Method	Endpoint	Date Tested		
9	POST	/api/v1/search/rank-properties	2/11/2025		
Description	Testing result when no output amenities are found				
Request			Response		
Headers	Body	Query	Expected	Actual	Pass/Fail
nil	{ amenityTheme: ‘school’, rank: 1, maxDistance: 1000 }	Town: “NONEXISTENT”	Response code 404	Response code 404 Error message: “No properties found”	Pass

ID	Method	Endpoint	Date Tested		
10	POST	/api/v1/search/rank-properties	2/11/2025		
Description	Testing error handling, using mockResolvedValue() to simulate an error result				
Request			Response		
Headers	Body	Query	Expected	Actual	Pass/Fail
nil	{ amenityTheme: ‘school’, rank: 1, maxDistance: 1000 }	Town: “TAMPINES”	Response code 500 Internal Server Error	Response code 500 Internal Server Error	Pass