

Black Box Testing

1. User Registration (register_user method)

Equivalence Classes:

- EC1: Valid registration data (all fields valid)
- EC2: Invalid email format
- EC3: Invalid password format
- EC4: Invalid phone number format
- EC5: Invalid monthly income (negative/non-numeric)
- EC6: Missing required fields
- EC7: Duplicate email registration

Test Case	Input Data	Expected Output	Actual Output	Equivalence Class	Boundary Value	Choice Rationale
TC1	{"name": "John Doe", "email": "john@example.com", "password": "Password123", "contactNumber": "91234567", "monthlyIncome": "3000.50"}	{"authenticate": True}, 201	Pass	EC1	N/A	Valid data from all fields - represents typical successful registration
TC2	{"name": "Jane", "email": "invalid-email", "password": "Password123", "contactNumber": "91234567", "monthlyIncome": "2500"}	{"error": "Invalid email format"}, 400	Pass	EC2	N/A	Missing @ symbol - tests email validation from Login.jsx regex

TC3	{"name": "Bob", "email": "bob@test.com", "password": "weak", "contactNumber": "91234567", "monthlyIncome": "1000"}	{"error": "Password validation failed"}, 400	Pass	EC3	Boundary: 7 chars	Password < 8 characters - tests password regex from RegisterForm.jsx
TC4	{"name": "Alice", "email": "alice@test.com", "password": "Strong123", "contactNumber": "71234567", "monthlyIncome": "2000"}	{"error": "Invalid phone number"}, 400	Pass	EC4	Boundary: starts with 7	Phone not starting with 8/9 - tests Singapore phone validation
TC5	{"name": "Charlie", "email": "charlie@test.com", "password": "Valid123", "contactNumber": "81234567", "monthlyIncome": "-500"}	{"error": "Monthly income must be non-negative"}, 400	Pass	EC5	Boundary: -1	Negative income - tests income validation boundary
TC6	{"name": "", "email": "test@test.com", "password": "Password123", "contactNumber": "91234567", "monthlyIncome": "1000"}	{"error": "All fields required"}, 400	Pass	EC6	Boundary: empty string	Missing name field - tests required field validation

TC7	{"name": "David", "email": "existing@test.com", "password": "Password123", "contactNumber": "91234567", "monthlyIncome": "1500"}	{"error": "Email already registered"}, 409	Pass	EC7	N/A	Duplicate email - tests unique constraint validation
-----	--	--	------	-----	-----	--

2. User Login (login_with_password method)

Equivalence Classes:

- EC8: Valid login credentials
- EC9: Invalid email format
- EC10: Invalid password format
- EC11: Non-existent user
- EC12: Wrong password
- EC13: Missing credentials

Test Case	Input Data	Expected Output	Actual Output	Equivalence Class	Boundary Value	Choice Rationale
TC8	{"email": "user@example.com", "password": "ValidPass123"}	{"authenticate": true, "role": "C"}, 200	Pass	EC8	N/A	Valid credentials - represents successful login flow
TC9	{"email": "invalid.email", "password": "ValidPass123"}	{"error": "Email and password required"}, 400	Pass	EC9	N/A	Invalid email format - tests frontend validation bypass

TC10	{"email": "user@test.com", "password": "short"}	{"error": "Email and password required"}, 400	Pass	EC10	Boundary: 5 chars	Password too short - tests validation consistency
TC11	{"email": "nonexistent@test.com", "password": "ValidPass123"}	{"error": "Invalid credentials"}, 401	Pass	EC11	N/A	Non-existent user - tests user lookup failure
TC12	{"email": "user@example.com", "password": "WrongPass123"}	{"error": "Invalid credentials"}, 401	Pass	EC12	N/A	Wrong password - tests password verification failure
TC13	{"email": "", "password": ""}	{"error": "Email and password required"}, 400	Pass	EC13	Boundary: empty strings	Missing both fields - tests required field validation

3. Google OAuth Login (google_callback method)

Equivalence Classes:

- EC14: Successful OAuth with existing user
- EC15: Successful OAuth with new user creation
- EC16: OAuth failure (invalid token)
- EC17: OAuth failure (no email from Google)

Test Case	Input Data	Expected Output	Actual Output	Equivalence Class	Boundary Value	Choice Rationale
-----------	------------	-----------------	---------------	-------------------	----------------	------------------

TC14	Valid Google OAuth token for existing user	Redirect to /clienthome , 200	Pass	EC14	N/A	Existing user OAuth - tests user lookup and session creation
TC15	Valid Google OAuth token for new user	Redirect to /clienthome , 201	Pass	EC15	N/A	New user OAuth - tests auto-registration flow
TC16	Invalid/expired Google OAuth token	{"error": "Google authentication failed"}, 400	Pass	EC16	N/A	Invalid token - tests OAuth error handling

4. User Logout (logout_user method)

Equivalence Classes:

- EC18: Valid logout (session exists)
- EC19: Logout with no active session

Test Case	Input Data	Expected Output	Actual Output	Equivalence Class	Boundary Value	Choice Rationale
TC18	Active user session	{"ok": True}, 200	Pass	EC18	N/A	Normal logout - tests session clearing
TC19	No active session	{"ok": True}, 200	Pass	EC19	N/A	Logout without session - tests graceful handling

White Box Testing

1. create_request() Method

Cyclomatic Complexity Calculation

$$V(G) = \text{Decision Points} + 1$$

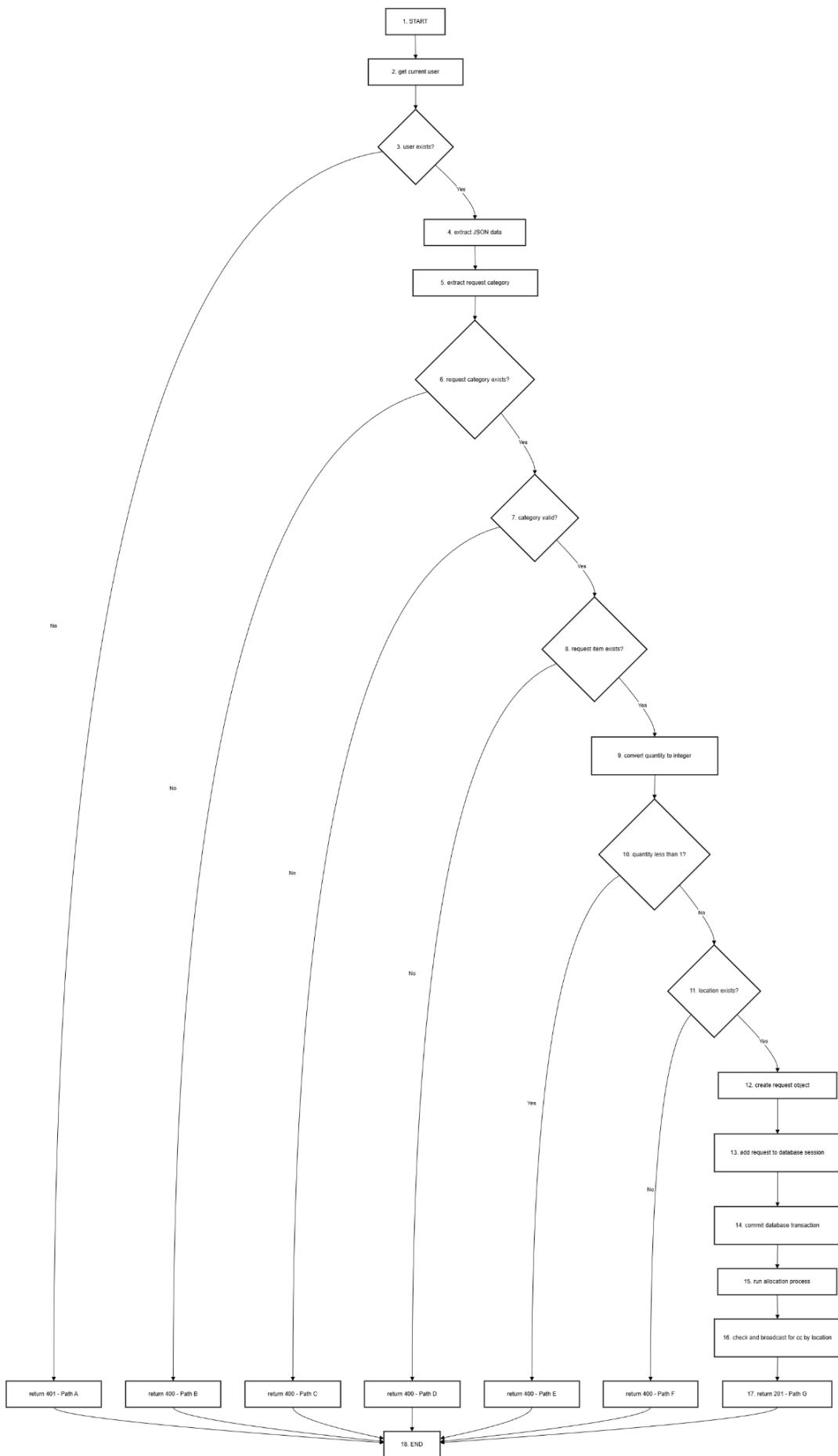
- Decision Points: 6 (if statements)
- $V(G) = 6 + 1 = 7$

Independent Paths Identified

1. Path A (Unauthorized): 1→2→3→END
2. Path B (Missing Category): 1→2→3→4→5→6→END
3. Path C (Invalid Category): 1→2→3→4→5→6→7→END
4. Path D (Missing Item): 1→2→3→4→5→6→7→8→END
5. Path E (Invalid Quantity): 1→2→3→4→5→6→7→8→9→10→END
6. Path F (Missing Location): 1→2→3→4→5→6→7→8→9→10→11→END
7. Path G (Success):
1→2→3→4→5→6→7→8→9→10→11→12→13→14→15→16→17→END

Test Case	Path	Input	Expected Output	Actual Output
TC1	A	user=None request_category="Food" request_item="Canned Goods" Request_quantity = 0 location="Henderson CC"	{"message": "Unauthorized"}, 401	{"message": "Unauthorized"}, 401
TC2	B	user=" user@example.com " request_category="" request_item="Canned Goods" Request_quantity = 2 location="Henderson CC"	{"message": "request_category is required"}, 400	{"message": "request_category is required"}, 400

TC3	C	user=" user@example.co m " request_category="xxx" request_item="Canned Goods" Request_quantity = 2 location="Henderson CC"	{"message": "request_category must be one of: Food, Drinks, Furnitures, Electronics, Essentials"}, 400	{"message": "request_category must be one of: Food, Drinks, Furnitures, Electronics, Essentials"}, 400
TC4	D	user=" user@example.co m " request_category="Food" request_item="" Request_quantity = 2 location="Henderson CC"	{"message": "request_item is required"}, 400	{"message": "request_item is required"}, 400
TC5	E	user=" user@example.co m " request_category="Food" request_item="Canned Goods" Request_quantity = -1 location="Henderson CC"	{"message": "request_quantity must be a positive integer"}, 400	{"message": "request_quantity must be a positive integer"}, 400
TC6	F	user=" user@example.co m " request_category="Food" request_item="Canned Goods" Request_quantity = 2 location=""	{"message": "location (CC) is required"}, 400	{"message": "location (CC) is required"}, 400
TC7	G	user=" user@example.co m " request_category="Food" request_item="Canned Goods" Request_quantity = 2 location="Henderson CC"	{"id": 1, "status": "Pending", "location": "Henderson CC", "request_item": "Canned Goods", "request_quantity": 2, "allocation": 0}, 201	{"id": 1, "status": "Pending", "location": "Henderson CC", "request_item": "Canned Goods", "request_quantity": 2, "allocation": 0}, 201



2. login_with_password() Method

Cyclomatic Complexity

$$V(G) = E - N + 2P$$

Where:

- E = Number of edges = 15
- N = Number of nodes = 14
- P = Number of connected components = 1

$$V(G) = 15 - 14 + 2(1) = 3$$

Independent Paths

1. Path A: 1→2→3→4→5→6→7 (Missing email or password)
2. Path B: 1→2→3→4→5→8→9→10 (User not found or no password hash)
3. Path C: 1→2→3→4→5→8→11→12→10 (Password verification fails)
4. Path D: 1→2→3→4→5→8→11→12→13→14 (Successful login)

Test Cases

Test ID	Path	Input	Expected Output	Actual Output
TC0 1	A	email="", password="ValidPass123"	{"error": "Email and password are required"}, 400	{"error": "Email and password are required"}, 400
TC0 2	A	email="user@example.com", password=""	{"error": "Email and password are required"}, 400	{"error": "Email and password are required"}, 400
TC0 3	B	email="nonexistent@example.com", password="ValidPass123"	{"error": "Invalid credentials"}, 401	{"error": "Invalid credentials"}, 401
TC0 4	C	email="user@example.com", password="wrongpassword"	{"error": "Invalid credentials"}, 401	{"error": "Invalid credentials"}, 401

TC0 5	D	email="user@example.com", password="ValidPass123"	{"authenticated": : True, "role": "C"}, 200	{"authenticated": True, "role": "C"}, 200
----------	---	--	---	--

