**I. USE CASE NAME: LOGIN**

**1. Requirement/Specifications-Based System-Level Test Cases**

Admin Login:

Requirement: Admin should be able to log in to the system using valid credentials.

Test Case:

* Test Case ID: TC001
* Description: Verify that the admin can successfully log in to the system.
* Preconditions: The system is accessible, and the admin has valid login credentials.

Test Steps:

* Navigate to the login page of the system.
* Enter valid admin username and password.
* Click on the "Login" button.

Expected Results:

* The system should authenticate the admin's credentials.
* If the credentials are valid, the admin should be redirected to the admin dashboard.
* A success message confirming the successful login should be displayed.

User Login:

Requirement: Users should be able to log in to the system using valid credentials.

Test Case:

* Test Case ID: TC002
* Description: Verify that users can successfully log in to the system.
* Preconditions: The system is accessible, and the user has valid login credentials.

Test Steps:

* Navigate to the login page of the system.
* Enter valid user username and password.
* Click on the "Login" button.

Expected Results:

* The system should authenticate the user's credentials.
* If the credentials are valid, the user should be redirected to the user dashboard or home page.
* A success message confirming the successful login should be displayed.

**2. Traceability of Test Cases to Use Cases**

* Test Case TC001 is traced to the "Admin Login" use case, which specifies that admins should be able to log in to the system with valid credentials.
* Test Case TC002 is traced to the "User Login" use case, which specifies that users should be able to log in to the system with valid credentials.

**3. Techniques Used for Test Generation**

* Equivalence Partitioning: Testing with valid and invalid credentials (e.g., correct username and password, incorrect username, incorrect password) to cover different scenarios.
* Boundary Value Analysis: Testing with minimum and maximum length usernames and passwords to verify the system's behavior at the boundaries.
* Decision Tables: Creating decision tables to cover combinations of input conditions and expected outcomes for different login scenarios.

**4. Assessment of the Goodness of Your Test Suite**

* Test Coverage Analysis: Evaluate the coverage of different login scenarios (e.g., successful login, failed login due to incorrect credentials, failed login due to account lockout).
* Defect Detection Rate: Measure the number and severity of defects identified during testing to assess the effectiveness of the test suite in finding issues.
* Test Execution Time: Monitor the time taken to execute the test suite for login functionality to ensure efficiency and effectiveness in validating this critical system feature.

**II. USE CASE NAME: ADDING A NEW PRODUCT**

**1. Requirement/Specifications-Based System-Level Test Cases**

Requirement: Admin should be able to add a new product to the system.

Test Case:

* Test Case ID: TC003
* Description: Verify that the admin can successfully add a new product to the system.
* Preconditions: Admin is logged in to the system and has access to the product management functionality.

Test Steps:

* Navigate to the product management section.
* Select the option to add a new product.
* Enter the required product details such as name, description, price, and quantity.
* Click on the "Add Product" button to submit the new product.

Expected Results:

* The system should display a success message confirming that the product has been added.
* The newly added product should be visible in the product catalog and searchable by users.
* All entered product details should be accurately saved in the system.

**2. Traceability of Test Cases to Use Cases**

Test Case TC003 is traced to the "Add Product" use case, which is defined in the system requirements document. This use case specifies that admins should have the ability to add new products to the system.

**3. Techniques Used for Test Generation**

* Equivalence Partitioning: Testing with valid and invalid input values for product details to ensure the system handles different scenarios appropriately.
* Boundary Value Analysis: Testing with minimum and maximum values for numeric fields (e.g., price, quantity) to verify the system's behavior at the boundaries.
* Error Guessing: Intentionally providing incorrect or incomplete input data to assess the system's error handling and validation mechanisms.

**4. Assessment of the Goodness of Your Test Suite**

* Test Coverage Analysis: Evaluate the coverage of different product attributes and scenarios tested (e.g., adding a product with varying price ranges, descriptions).
* Defect Detection Rate: Measure the number and severity of defects identified during testing to assess the effectiveness of the test suite in finding issues.
* Test Execution Time: Monitor the time taken to execute the test suite for adding new products to ensure efficiency and effectiveness in validating this functionality.

**III. USE CASE NAME: ADDING A NEW STAFF**

**1. Requirement/Specifications-Based System-Level Test Cases**

Requirement: Admin should be able to add a new staff member to the system.

Test Case:

* Test Case ID: TC004
* Description: Verify that the admin can successfully add a new staff member to the system.
* Preconditions: The admin is logged in to the system and has access to the staff management functionality.

Test Steps:

* Navigate to the staff management section.
* Select the option to add a new staff member.
* Enter the required staff details such as name, email, role, and contact information.
* Click on the "Add Staff" button to submit the new staff member.

Expected Results:

* The system should display a success message confirming that the staff member has been added.
* The newly added staff member should be visible in the staff list with the provided details.
* All entered staff details should be accurately saved in the system.

**2. Traceability of Test Cases to Use Cases**

Test Case TC004 is traced to the "Add Staff" use case, which specifies that admins should have the ability to add new staff members to the system.

**3. Techniques Used for Test Generation**

* Equivalence Partitioning: Testing with valid and invalid input values for staff details (e.g., valid email format, unique email address, valid role).
* Boundary Value Analysis: Testing with minimum and maximum length values for text fields (e.g., name, email) to verify the system's behavior at the boundaries.
* Error Guessing: Intentionally providing incorrect or incomplete input data to assess the system's error handling and validation mechanisms.

**4. Assessment of the Goodness of Your Test Suite**

* Test Coverage Analysis: Evaluate the coverage of different staff attributes and scenarios tested (e.g., adding a staff member with varying roles, email formats).
* Defect Detection Rate: Measure the number and severity of defects identified during testing to assess the effectiveness of the test suite in finding issues.
* Test Execution Time: Monitor the time taken to execute the test suite for adding new staff members to ensure efficiency and effectiveness in validating this functionality.

**IV. USE CASE NAME: ADDING A NEW MEMBERSHIP**

**1. Requirement/Specifications-Based System-Level Test Cases**

Requirement: Admin should be able to add a new membership for customers in the system.

Test Case:

* Test Case ID: TC005
* Description: Verify that the admin can successfully add a new membership for a customer in the system.
* Preconditions: The admin is logged in to the system and has access to the customer management functionality.

Test Steps:

* Navigate to the customer management section.
* Select the option to add a new membership.
* Enter the required customer details such as name, phone number, and membership type.
* Click on the "Add Membership" button to submit the new membership.

Expected Results:

* The system should display a success message confirming that the membership has been added for the customer.
* The newly added membership should be visible in the customer's profile with the provided details.
* All entered membership details should be accurately saved in the system.

**2. Traceability of Test Cases to Use Cases**

Test Case TC005 is traced to the "Add Membership" use case, which specifies that admins should have the ability to add new memberships for customers in the system.

**3. Techniques Used for Test Generation**

* Equivalence Partitioning: Testing with valid and invalid input values for customer details (e.g., valid phone number format, unique phone number).
* Boundary Value Analysis: Testing with minimum and maximum length values for text fields (e.g., name, phone number) to verify the system's behavior at the boundaries.
* Error Guessing: Intentionally providing incorrect or incomplete input data to assess the system's error handling and validation mechanisms.

**4. Assessment of the Goodness of Your Test Suite**

* Test Coverage Analysis: Evaluate the coverage of different membership attributes and scenarios tested (e.g., adding a membership with varying types, phone number formats).
* Defect Detection Rate: Measure the number and severity of defects identified during testing to assess the effectiveness of the test suite in finding issues.
* Test Execution Time: Monitor the time taken to execute the test suite for adding new memberships to ensure efficiency and effectiveness in validating this functionality.

**V. USE CASE NAME: ADDING A NEW COUPON**

**1. Requirement/Specifications-Based System-Level Test Cases**

Requirement: Admin should be able to add a new coupon to the system.

Test Case:

Test Case ID: TC006

* Description: Verify that the admin can successfully add a new coupon to the system.
* Preconditions: The admin is logged in to the system and has access to the coupon management functionality.

Test Steps:

* Navigate to the coupon management section.
* Select the option to add a new coupon.
* Enter the required coupon details such as code, discount amount, and validity period.
* Click on the "Add Coupon" button to submit the new coupon.

Expected Results:

* The system should display a success message confirming that the coupon has been added.
* The newly added coupon should be visible in the list of available coupons with the provided details.
* All entered coupon details should be accurately saved in the system.

**2. Traceability of Test Cases to Use Cases**

Test Case TC006 is traced to the "Add Coupon" use case, which specifies that admins should have the ability to add new coupons to the system.

**3. Techniques Used for Test Generation**

* Equivalence Partitioning: Testing with valid and invalid input values for coupon details (e.g., valid coupon code format, unique coupon code).
* Boundary Value Analysis: Testing with minimum and maximum values for numeric fields (e.g., discount amount) to verify the system's behavior at the boundaries.
* Error Guessing: Intentionally providing incorrect or incomplete input data to assess the system's error handling and validation mechanisms.

**4. Assessment of the Goodness of Your Test Suite**

* Test Coverage Analysis: Evaluate the coverage of different coupon attributes and scenarios tested (e.g., adding a coupon with varying discount amounts, validity periods).
* Defect Detection Rate: Measure the number and severity of defects identified during testing to assess the effectiveness of the test suite in finding issues.
* Test Execution Time: Monitor the time taken to execute the test suite for adding new coupons to ensure efficiency and effectiveness in validating this functionality.

**VI. USE CASE NAME: REGISTER**

**1. Requirement/Specifications-Based System-Level Test Cases**

Requirement: Users should be able to register for an account in the system.

Test Case:

* Test Case ID: TC007
* Description: Verify that users can successfully register for an account in the system.
* Preconditions: The system is accessible, and the user has not registered for an account before.

Test Steps:

* Navigate to the registration page of the system.
* Enter the required user details such as username, email, password, and contact information.
* Click on the "Register" or "Sign Up" button to submit the registration form.
* Check for a verification email or confirmation message.
* Optionally, verify the user's email address if required by the system.

Expected Results:

* The system should display a success message confirming that the user has been registered.
* The user should receive a verification email or confirmation message indicating successful registration.
* If email verification is required, the user should be prompted to verify their email address before accessing the system.

**2. Traceability of Test Cases to Use Cases**

Test Case TC007 is traced to the "User Registration" use case, which specifies that users should be able to register for an account in the system.

**3. Techniques Used for Test Generation**

* Equivalence Partitioning: Testing with valid and invalid input values for user details (e.g., valid email format, unique username).
* Boundary Value Analysis: Testing with minimum and maximum length values for text fields (e.g., username, password) to verify the system's behavior at the boundaries.
* Error Guessing: Intentionally providing incorrect or incomplete input data to assess the system's error handling and validation mechanisms.

**4. Assessment of the Goodness of Your Test Suite**

* Test Coverage Analysis: Evaluate the coverage of different registration attributes and scenarios tested (e.g., registration with varying email formats, password lengths).
* Defect Detection Rate: Measure the number and severity of defects identified during testing to assess the effectiveness of the test suite in finding issues.
* Test Execution Time: Monitor the time taken to execute the test suite for user registration to ensure efficiency and effectiveness in validating this critical system feature.