

Project and Professionalism (6CS020)

Art Gallery Management System

Artefact Design and Test Plan

Student ID: 2042471

Student Name: Sujan Pyakurel

Supervisor: Sumanta Silwal

University Email:  [S.Pyakurel@wlv.ac.uk](mailto:S.Pyakurel@wlv.ac.uk)

Award/Course: BSc Hons Computer Science

Cohort/Batch: 4

Submission Date: December 19, 2020

Contents

[Artifact design 1](#_Toc59324348)

[1.FDD (Functional Decomposition Diagram) 1](#_Toc59324349)

[1](#_Toc59324350)

[2. System Requirement Specification (SRS) 1](#_Toc59324351)

[2.1 SRS Table 2](#_Toc59324352)

[2.1.1 User Management System 2](#_Toc59324353)

[2.1.2 Product Management System 4](#_Toc59324354)

[2.1.3 Checkout Management System 4](#_Toc59324355)

[2.1.4 Payment Management System 5](#_Toc59324356)

[2.1.5 Gallery Management System 5](#_Toc59324357)

[2.1.6 Shipping and Billing Management System 5](#_Toc59324358)

[3. Activity Diagram 6](#_Toc59324359)

[3.1 Activity Diagram of sign up and Login 7](#_Toc59324360)

[3.2 Activity diagram sowing adding and editing user by admin 8](#_Toc59324361)

[4. Use Case Diagram 9](#_Toc59324362)

[5. Wireframes 10](#_Toc59324363)

[5.1 Wireframe of Admin login 10](#_Toc59324364)

[5.2 Wireframe of Admin Password Reset 11](#_Toc59324365)

[5.3 Wireframe of Admin Dashboard 12](#_Toc59324366)

[5.4 Wireframe of New user adding section 12](#_Toc59324367)

[5.5 Wireframe of User Login 13](#_Toc59324368)

[6. ERD 14](#_Toc59324369)

[14](#_Toc59324370)

[7. Class Diagram 15](#_Toc59324371)

[8. Sequence Diagram 15](#_Toc59324372)

[8.1 Sequence Diagram of sign up 15](#_Toc59324373)

[8.2 Sequence diagram of Sign in 16](#_Toc59324374)

[9. State Transition Diagram 18](#_Toc59324375)

[10. Testing 18](#_Toc59324376)

Table of Figures

[Figure 1FDD 1](#_Toc59324334)

[Figure 2Activity Diagram of sign up and Login 7](#_Toc59324335)

[Figure 3Activity diagram sowing adding and editing user by admin 8](#_Toc59324336)

[Figure 4UseCase diagram of User Management System 9](#_Toc59324337)

[Figure 5Wireframe of Admin login 10](#_Toc59324338)

[Figure 6Wireframe of Admin Password Reset 11](#_Toc59324339)

[Figure 7Wireframe of Admin Dashboard 12](#_Toc59324340)

[Figure 8Wireframe of New user adding section 13](#_Toc59324341)

[Figure 9Wireframe of User Login 14](#_Toc59324342)

[Figure 10ERD 14](#_Toc59324343)

[Figure 11Class Diagram 15](#_Toc59324344)

[Figure 12Sequence Diagram of sign up 16](#_Toc59324345)

[Figure 13Sequence diagram of Sign in 17](#_Toc59324346)

[Figure 14State Transition Diagram 18](#_Toc59324347)

# Artifact design

# 1.FDD (Functional Decomposition Diagram)

## FDD.PNG

Figure 1FDD

# 2. System Requirement Specification (SRS)

SRS stands for System Requirement Specification which is the description of the general purpose surrounding for the software development. SRS defines how the application will function and manipulate with the system, their hardware and other components. It is based upon the business requirement, the system functionality are highlighted with table is given below:

**Legend**

UM-F-1.0

F: Functional Requirements

NF: Non-Functional Requirements

UR: Usability Requirements

Numbering

UM: User Management System

PM: Product Management System

CM: Checkout Management System

PM: Payment Management System

GM: Gallery Management System

SBM: Shipping and Billing Management System

## 2.1 SRS Table

### 2.1.1 User Management System

|  |  |  |
| --- | --- | --- |
| Requirement ID | Requirement Specification and justification | Moscow |
| UM-F-1 | Individual users should be provided a login and registering option. | Must have |
| UM-NF-1.1 | Weak password is not allowed. Passwords must contain alphanumeric, symbols and at least 8 characters. | Must have |
| UM -UR-1.1 | Show message in case a login attempt fails. | Must have |
| UM -UR-1.3 | Notification related to the credentials incorrect (i.e. either password or email) should be delivered to the user. | Should have |
| UM-F-2 | The users should be provided with options to manage their account/ profile. | Must have |
| UM-NF-2.1 | The management system should allow users and to sign-out if they are not interested. | Must have |
| UM-NF-2.2 | The users should be provided with options to change their password. | Must have |
| UM-NF-2.3 | The user’s password reset code should be sent to the user’s email account. | Must have |
| UM-UR-2.1 | User must enter registered email and password. | Should |
| UM-F-3 | Admin can manage, add or block any users from the system. | Must have |
| UM-NF-3.1 | Admin can login and logout of the system to perform administrative tasks. | Must have |
| UM-NF-3.2 | Admin should be able to view the list of users in the system. | Should have |
| UM-NF-3.3 | Admin should be able to create new user, edit user’s information, block any users and see the history of users from the system. | Should have |

### 2.1.2 Product Management System

|  |  |  |
| --- | --- | --- |
| Requirement ID | Requirement Specification and Justification | Moscow |
| PM-F-1 | Admin should be able to create a product or upload paintings and artworks in the web app. | Must have |
| PM-NF-1.1 | The system would only allow the creation of product with the minimum resolution of 1024 X 760 and the file size should not exceed 20 MB. | Must have |
| PM-NF-1.2 | The image could be upload using drag and drop feature so that process can be easy and time saving. | Could have |
| PM-UR-1.1 | The paintings should be given unique bar codes. | Should have |
| PM-F-2 | The system should allow admin to manage and categorize the products and add descriptions about the paintings. | Should have |
| PM-NF-2.1 | The descriptions should not exceed 150 words. | Should have |
| PM-NF-2.1 | The categorization can be done the basis of art types, styles, artist, mediums, subjects etc. | Should have |

### 2.1.3 Checkout Management System

|  |  |  |
| --- | --- | --- |
| Requirement ID | Requirement Specification and Justification | Moscow |
| CM-F-1 | The users must be able to checkout their cart and confirm their shop before making payments. | Must have |
| CM-NF-1.1 | Users must be logged in to the system. | Must have |
| CM-NF-1.2 | Customers must be able to remove the products(paintings) from the cart they would not like to buy. | Should have |
| CM-NF-1.3 | The checkout management system should display quantity, title and price of the product. | Should have |
| CM-UR-1.1 | The system should be able to sum (and give discount if any) the amounts of products and display to the system. | Should have |

### 2.1.4 Payment Management System

|  |  |  |
| --- | --- | --- |
| Requirement ID | Requirement Specification and Justification | Moscow |
| PM-F-1 | Customers should be able to make payments through various international systems. | Should have |
| PM-NF-1.1 | Customers could make payments through different means of payments such as Visa card, MasterCard, PayPal etc. | Should have |

### 2.1.5 Gallery Management System

|  |  |  |
| --- | --- | --- |
| Requirement ID | Requirement Specification and Justification | Moscow |
| GM-F-1 | Users should be provided with a feature to save paintings in their gallery. | Could have |
| GM-NF-1.1 | Users must login to the system to access this feature to save artworks. They can also view the saved paintings in their profile after logging in to the system. | Must have |

### 2.1.6 Shipping and Billing Management System

|  |  |  |
| --- | --- | --- |
| Requirement ID | Requirement Specification and Justification | Moscow |
| SBM-F-1 | Users should be provided with service for the delivery of the purchased artworks from the system. | Should have |
| SBM-NF-1.1 | Users must have added products to the cart, made checkout and completed payment before accessing the service for the delivery of the products. | Must have |
| SBM-NF-1.2 | People should fill up the form for the shipping and billing details including their email and contact number. | Must have |

# 3. Activity Diagram

## 3.1 Activity Diagram of sign up and Login

Diagram

Description automatically generated

Figure 2Activity Diagram of sign up and Login

## 3.2 Activity diagram sowing adding and editing user by admin

Diagram

Description automatically generated

Figure 3Activity diagram sowing adding and editing user by admin

# 4. Use Case Diagram

Diagram

Description automatically generated

Figure 4UseCase diagram of User Management System

# 5. Wireframes

## 5.1 Wireframe of Admin login

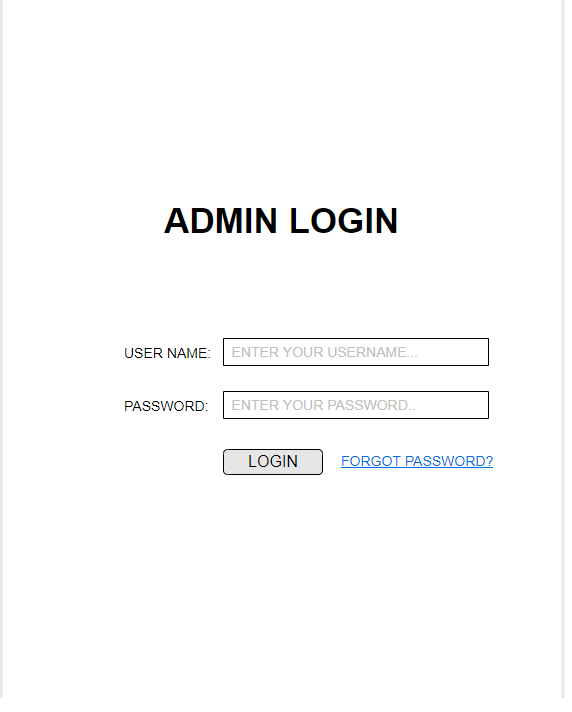


Figure 5Wireframe of Admin login

## 5.2 Wireframe of Admin Password Reset

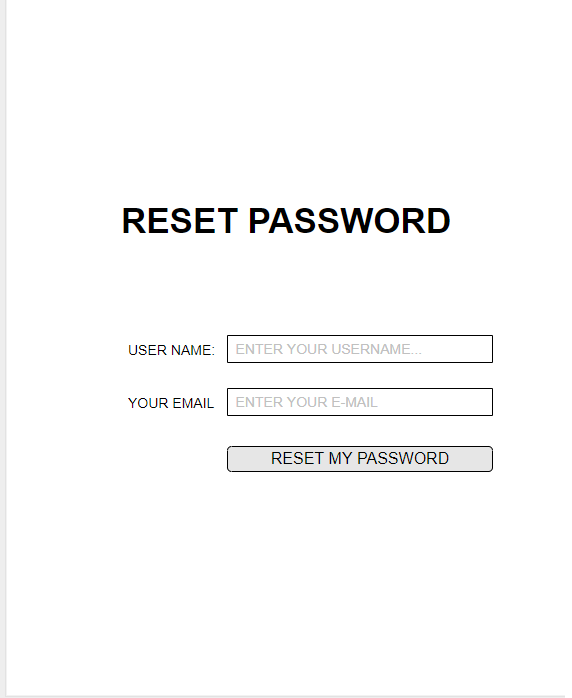


Figure 6Wireframe of Admin Password Reset

## 5.3 Wireframe of Admin Dashboard

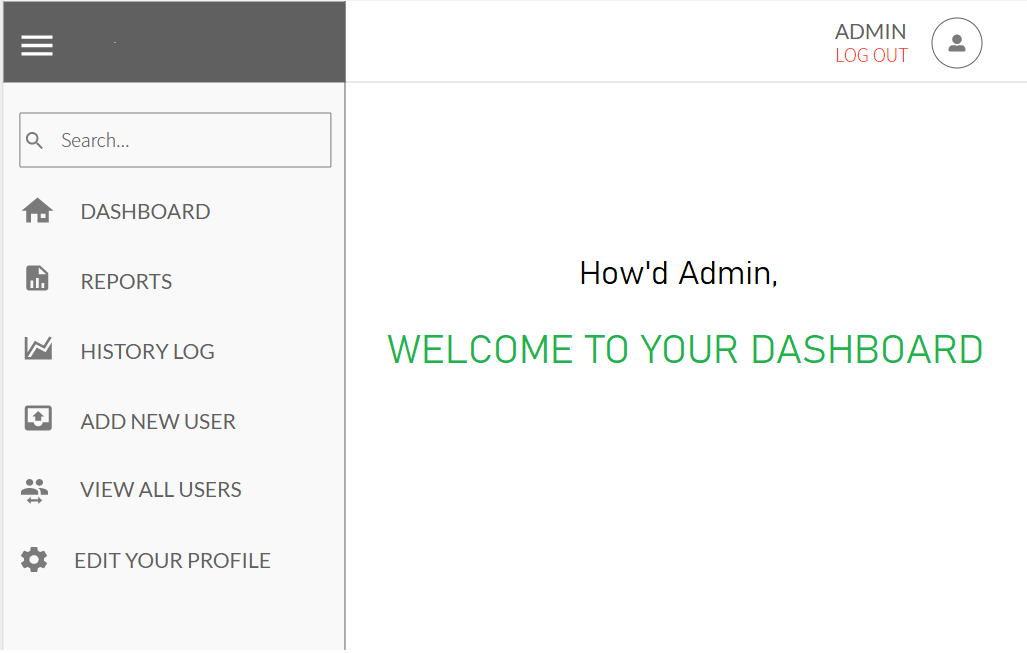


Figure 7Wireframe of Admin Dashboard

## 5.4 Wireframe of New user adding section

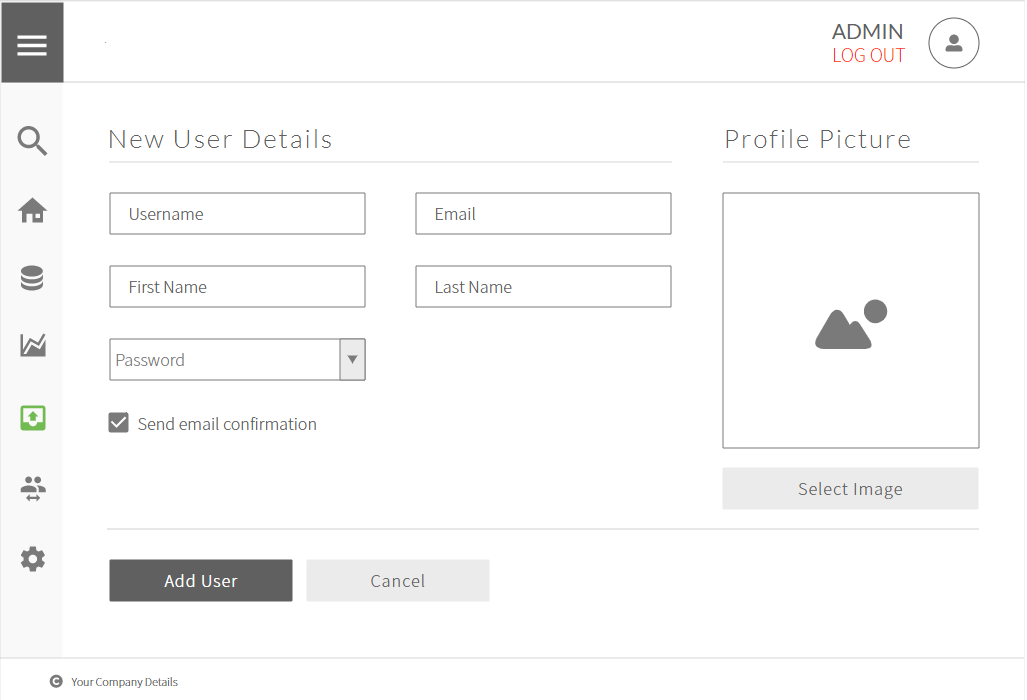


Figure 8Wireframe of New user adding section

## 5.5 Wireframe of User Login

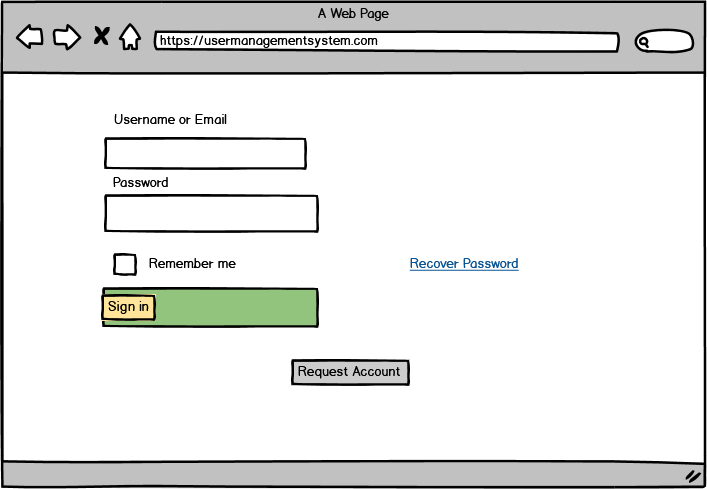


Figure 9Wireframe of User Login

# 6. ERD

# 

Figure 10ERD

# 7. Class Diagram

Diagram, schematic

Description automatically generated

Figure 11Class Diagram

# 8. Sequence Diagram

## 8.1 Sequence Diagram of sign up

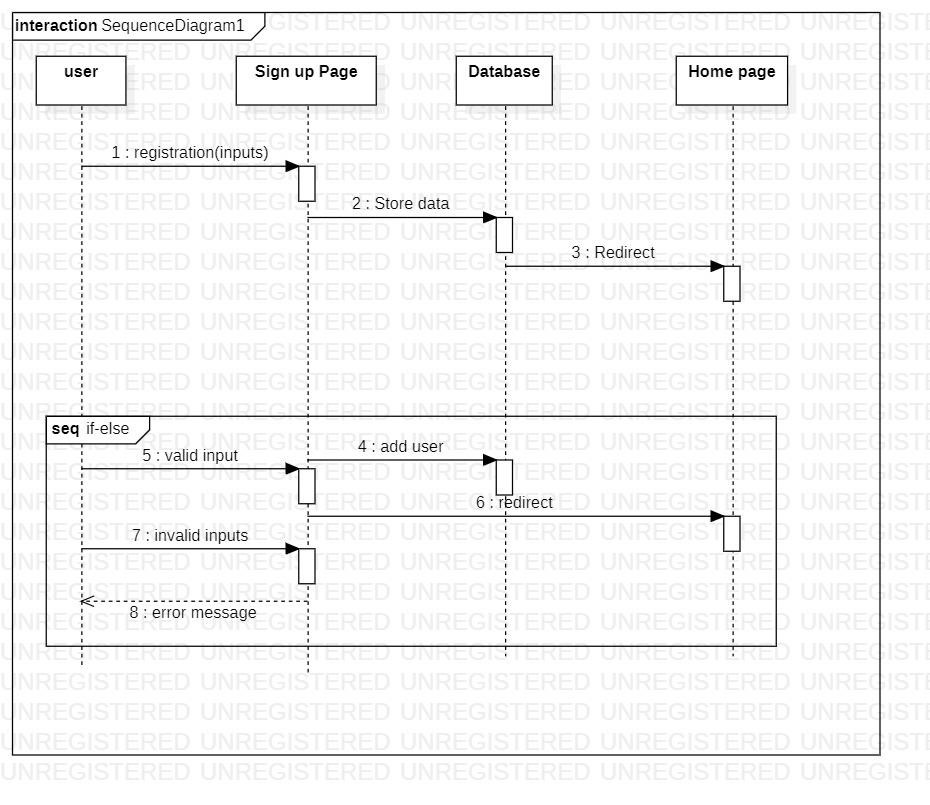


Figure 12Sequence Diagram of sign up

## 8.2 Sequence diagram of Sign in

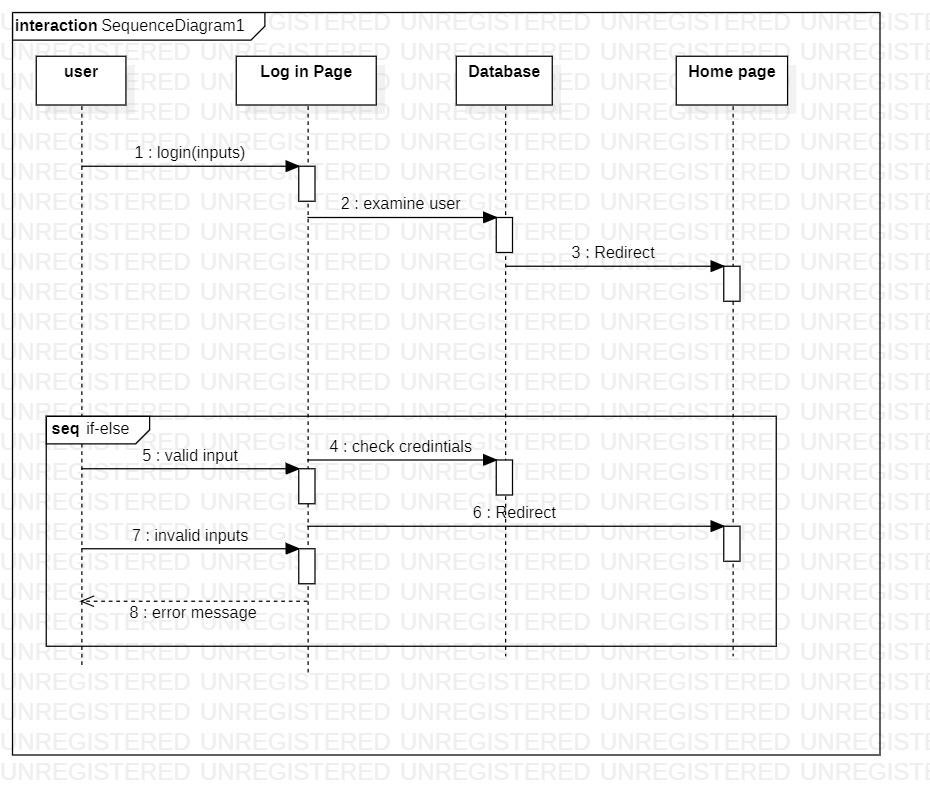


Figure 13Sequence diagram of Sign in

# 9. State Transition Diagram

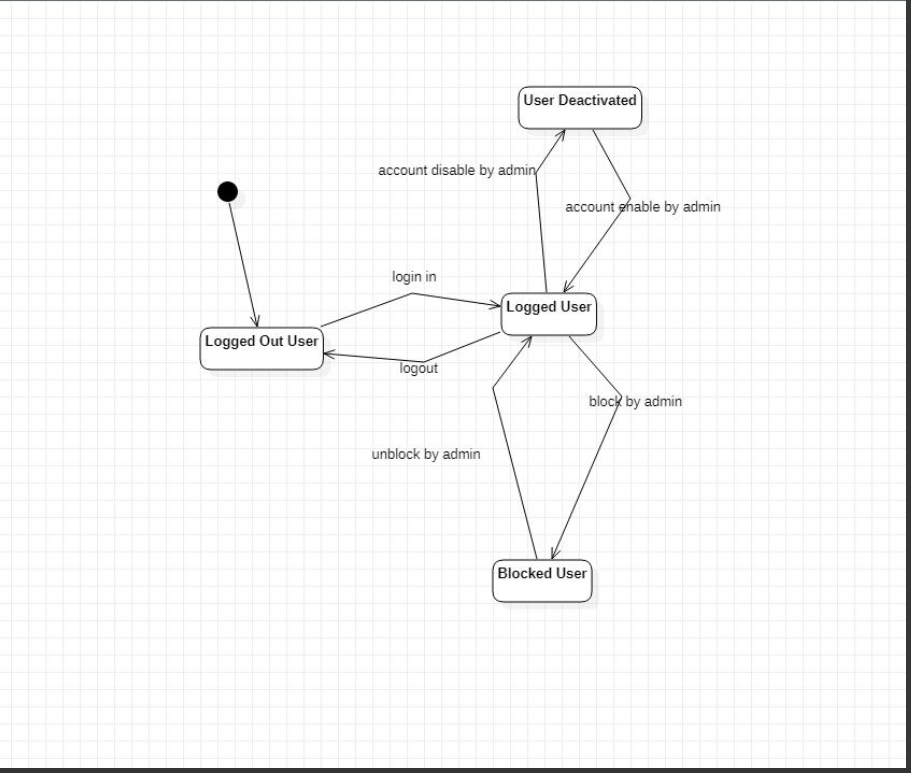


Figure 14State Transition Diagram

# 10. Testing

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ADMIN: |  |  |  |  |  |  |  |  |  |  |
| UserStoryID | TestCaseID | Title | Description | Pre Condition | TestStep | TestData |  | POSTCONDITION | Actual Result | Status |
| US001 | TC001 | Login | Test Admin Login | 1:User must be on Admin Login Page 2:No User is currenty logged in | 1:Go to login page 2:Enter username and password on the respective  input field 3:Press Login In Button | Username="admin" | verify sql query:select \* from user where username=admin and password=admin IF:username and password matches database values  then  redirect to dashboard | User Session is stored added action to admin history | pass | TRUE |
| password="admin" | else: return to login page with error msg | N/A | pass | TRUE |
| US002 | TC002 | Logout | Test Admin Logout | 1:Must Be Logged in as Admin | 1:Press Logout Button on the nav bar | N/A | Delete Session data and redirect to login page | Clear Session data add action to admin history | pass | TRUE |
| US003 | TC003 | Reset Password | Test Reset Password | 1:Must Be Logged in as admin 2:Must be on profile page | 1:Press on Change Password 2:Enter Old Password | password="admin" userId=101 username="admin" | sql:select password from user where userId=101 if password==db\_password: send reset email else: redirect to profile page with error msg | Password reset mail is send to the user email | pass | TRUE |
| TC004 |  | Test Forget Password | 1:Must not be logged in 2:Selected Forget Password Option | 1:Enter email or user\_name on input field 2:Reset Password | username="admin" | sql:select password from user where username='admin' send reset email |  | pass | TRUE |
| TC005 |  | Test Reset Password Page | 1:Must be check the email inbox | 1:Press the email 2:Enter new password on the input field 3:Press Reset Password | new\_password="admin123", user\_token="<token to identiy user>" | 1:check that user token if its valid and get user 2: redirect to password reset page | User session is stored added action to admin history | pass | TRUE |
| US004 | TC006 | check user list | Test List of User page | 1:Must be logged in as admin 2:Must be on users page | 1:Go to users page  2:SELECT a fliter option, default all users 3:SELECT number\_of\_list,default 10 | filter="ALL", number\_of\_list="10" | sql query:select \* from user; All user in the data base filtred by the criteria |  | pass | TRUE |
| US005 | TC007 | Create User | Test User creation page | 1:Must be logged in as admin 2:Must be on user creation page | 1:Enter all the user inforamtion ie username, password,email,phone,age,gender and so on 2:press create user button | username="bob" email="bob@bob.com", password="bob123", phone="987654321", age=20 gender="Male" | sql query:INSERT INTO USER VALUES(username,email,password, phone,age,gender);  return to user page with user creation message | added action admin history | pass | TRUE |
|  |  |  |  |  | username="bob" email="", password="bob123", phone="987654321", age=20 gender="Male" |  |  | pass | TRUE |
| US006 | TC008 | Edit User | Test User Info update page | 1:Must be logged in as admin  2:Must be on list of user | 1:Select a user from a list 2:Edit the user info  3:Press Save | username="bob"  email="bob@bob.com",  password="bob123",  phone="987654321",  age=20  gender="Male" | sql quer:UPDATE USER SET username="bob",email="bob@bob.com", phone="987456123",age=20,gender="male";  returnt to user edit page with user successfully edited | added action to admin history | pass | TRUE |
| US007 | TC009 | Block User | Test Block User | 1:Must be logged in as admin 2:Must be on user edit page | 1:Press the block user button | user\_status="block" | UPDATE USER SET user\_staus="blocked"  the user will not be able to user the application | added action to admin history | pass | TRUE |
| US008 | TC010 | View Profile | Test Admin Profile | 1:Must be logged in as admin | 1:Select the profile icon on the nav bar |  |  |  | pass | TRUE |
| US009 | TC012 | View History | Test History | 1:Must be logged in as admin 2:Go to history in nav bar | 1:Select a User | user\_id=101 | sql query:SELECT \* FROM HISTORY where user\_id=101; List of users history |  | pass | TRUE |
| US010 | TC013 | Generate Report | Test Report generation | 1:Must be logged in as admin 2:Be a History page | 1:Select generate report  2:Select date , and user status 3:Press generate | date=2020/12/10 user\_status="blocked" | sql query:SELECT \* FROM HISTORY where user\_status="blocked" and date="2020/01/05";  the information in printed on a paper |  | pass | TRUE |
| GUEST: |  |  |  |  |  |  |  |  |  |  |
| GS001 | TC0014 |  | Test Client sign up Register | 1:Guest are not allowed to view any page 2:No Guest is currenty Register | 1:Go to Sign up page 2:Enter username and password on the respective  input field 3:Click Sign up Button | Username="Guest" , password=" ", Email=" ", Location=" ", Phone=" " | verify sql query:select \* from user where username=guest and password=12345  IF:username and password matches database values  then  redirect to dashboard | Guest Session is stored and  added to User history | pass | TRUE |
| GS002 | TC0015 |  | Test Client sign up Register | 1:Must Be Register as a guest | 1:Click Account Deactivate Button on the nav bar | Username="Guest" , password="65984 ", Email="madhav@gmail.com ", Location="kailali ", Phone="112346455 " | Delete Session data and redirect to register page | Clear Session data add to admin history | pass | TRUE |
| USER: |  |  |  |  |  |  |  |  |  |  |
| US011 | TC016 | Register | Test user register | 1. Client must not be registered. 2. Client should be in login or SignUp page 3.No any user is logged in currently | 1. Client enters the username 2.Client enters the email address 3.Client enters the Password and confirms the password | 1. First Name: "Sujan" 2. Last Name: "Pyakurel" 3.Username: "SuzanCule" 4. Contact number: "9860637387" 5.Email: "Suzan@gmail.com" 6.Password: "12345678" 7.Confirm Password: "12345678" |  |  | pass | TRUE |
| US012 | TC017 | Login | Test user Login | 1. User must be in user login page 2. No user is currently logged in | 1:Go to login page  2:Enter username and password on the respective  input field  3:Press Login In Button | 1. Username: "SuzanCule" 2. Password: "12345678" | verify sql query:select \* from user where username="SuzanCule" and password="12345678" IF:username and password matches database values then redirect to dashboard else: return to login page with error msg |  | pass | TRUE |
| US013 | TC018 | Logout | Test user Logout | 1.User must be logged in to the system | 1. Press logout button on the navbar | N/A | 1.Delete Session data and redirect to login page |  | pass | TRUE |
| US014 | TC019 | History | Test User History | 1. Client must be logged in | 1. Client clicks on dashboard 2.Clients clicks on history | user id =1 | data is displayed from history table as per mock098 select \* from history where user id= 1 | Client is directed to history page | pass | TRUE |
| US015 | TC020 | Dashboard | Test user dashboard | 1. Client must be logged in | 1. Client checks the dashboard 2. Client checks the functionality of dashboard 3. Client surfs the system easily | N/A | 1. User is shown in the dashboard |  | pass | TRUE |