**CIS 634 Object-Oriented Software Engineering**

**Test Plan and Test Cases**

**Project Title: Power Management System**

Logo, company name

Description automatically generated

Instructor: Prof. Weidong Xiong

Group Number: 7

Group Members:

Tapan Desai CSU ID: 2777437

Shaishav Shah CSU ID: 2835285

Kinjal Patel CSU ID: 2794458

Mitee Patel CSU ID: 2817313

**1.0 Introduction**

The purpose of the Test Plan is to specify the scope, approach, resources, and schedule for each, and every testing activity related to the Web Application's operation, as well as its behavioral changes to specific approaches.

* 1. **Goals and objectives**

The items to be tested, the features to be tested, the kinds of testing to be done, the people in charge of testing, the resources and schedule needed to finish testing, and the risks associated with the plan are all part of the basic plan.

**2.0 Test Plan**

There are 3 stages of testing:

* Development testing: the testing is done during development to discover issues and bugs.
* Release testing: the testing is done by a separate testing team before the release of the system.
* User testing: the testing is done by the users of the system as a whole.

Here we will mainly focus of the development testing:

* API testing:
* This is done to verify each API developed for the software is working correctly or not.
* We are using Postman desktop application for this.
* Postman gives us a platform on which we can build and test our system’s API easily and accurately.
* For example: for user registration: we have tested save user details, get user details and get all users APIs on Postman.
* Unit testing:
* This is done by testing individual program units or object classes.
* Unit testing is one kind of functionality testing which focuses on testing the individual components in isolation.
* This is mostly done to find out the defects.
* Individual methods, object classes and composite components are considered the units which can be tested in unit testing.
* For example: for user registration, we have tested all the individual methods, attributes setting & interrogation and object state associated with the user class object for our web application.
* Unit testing is done through test cases. We choose test cases such that, when components are used as they are expected to, they produce the result that they are supposed to produce. So, any present defects will be identified through these test cases.
* Component testing:
* Component testing is done after unit testing.
* This is done by testing several individual units which are made up of various interacting objects to create composite components.
* This testing focuses on the component interfaces testing. Means, it should check whether the component interfaces are behaving as specified.
* Here, we can find any possible faults due to the interfaces and their interactions like misuse of interfaces, interface misunderstanding and timing errors.
* In this test, pointer parameters are testing with null pointers, stress testing is used in message system and generally testing is done to cause the components to fail.
* System testing:
* This is done by testing some or all the components in the system integrated together as a whole to test the emergent behavior of the system.
* Components are integrated during the development to create a version of the system.
* After the version of the system is integrated, that system will be tested to test all the different interactions between the components.
* This checks whether the components are compatible, interacting correctly, and the data transfer between interfaces is done correctly and in timely manner.
* The **use-case testing** is used for testing the system.
* Different use-cases are created for various system interactions that in turn will be used as the fundamentals for system testing.

**3.0 Test Cases**

Admin Test Cases:

1. Login:

|  |  |
| --- | --- |
| ID | 1.1 |
| Test Input | 1. Email 2. Password |
| Expected Output | Congratulations, you are successfully logged in. |
| Description | If the email and the password entered by the admin matches the database, then they are logged in. |

|  |  |
| --- | --- |
| ID | 1.2 |
| Test Input | 1. Email 2. Password |
| Expected Output | Please enter correct email and password. |
| Description | If the email and the password entered by the admin does not match the database, then they cannot log in. |

1. Forget Password:

|  |  |
| --- | --- |
| ID | 2.1 |
| Test Input | Click on forget password button. |
| Expected Output | A new password is shown on the screen of the admin. |
| Description | If the admin clicks on the forget password button, then a new random password is generated and shown on the screen of the admin. |

1. Admin Home Page:

|  |  |
| --- | --- |
| ID | 3.1 |
| Test Input | Click on view user details button in the users’ list on the admin home page. |
| Expected Output | The user details page will be opened. |
| Description | If the admin clicks on the view user details button in the users’ list on the admin home page, then a new page containing all the details of that particular user should be displayed. |

1. View User Details Page:

|  |  |
| --- | --- |
| ID | 4.1 |
| Test Input | Click on view bills button on the user details page. |
| Expected Output | The bills list page will be opened. |
| Description | If the admin clicks on the view bills button on the user details page, then a new page containing all the bills of that particular user should be displayed. |

1. View Bills Page:

|  |  |
| --- | --- |
| ID | 5.1 |
| Test Input | Click on view bill details button in the bill list on the view bills page. |
| Expected Output | The bill details page will be opened. |
| Description | If the admin clicks on the view details bills button in the bill list on the view bills page, then a new page containing all the details of that particular bill of the user should be displayed. |

1. View Bill Details Page:

|  |  |
| --- | --- |
| ID | 6.1 |
| Test Input | Click on edit bill button on the view bill details page. |
| Expected Output | The edit bill details page will be opened. |
| Description | If the admin clicks on the edit bill button on the view bill details page, then the admin should be able to edit all the details of that particular bill. |

1. Edit Bill Page:

|  |  |
| --- | --- |
| ID | 7.1 |
| Test Input | Click on save changes button on the edit bill details page. |
| Expected Output | The edited bill details should be saved, and a success message should be displayed to the admin. |
| Description | If the admin clicks on save changes button on the edit bill details page, then the edited bill details should be saved, and a success message should be displayed to the admin. |

|  |  |
| --- | --- |
| ID | 7.2 |
| Test Input | Click on save changes button on the edit bill details page. |
| Expected Output | An error message should be displayed to the admin if the changes were not done successfully. |
| Description | If the admin clicks on save changes button on the edit bill details page but the changes were not done successfully, then an error message should be displayed to the admin. |

User Test Cases:

1. Registration:

|  |  |
| --- | --- |
| ID | 1.1 |
| Test Input | 1. Name 2. DOB 3. Email 4. Home-Address 5. Password |
| Expected Output | User is registered successfully! |
| Description | If the user enters all the correct details and the formatting of email and password is correct, then the user is successfully registered. |

|  |  |
| --- | --- |
| ID | 1.2 |
| Test Input | 1. Name 2. DOB 3. Email 4. Home-Address 5. Password |
| Expected Output | Please enter correct details! |
| Description | If the user does not enter the correct details and the formatting of email and password is incorrect, then the user will get an error message saying please enter the correct details. |

1. Login:

|  |  |
| --- | --- |
| ID | 2.1 |
| Test Input | 1. Email 2. Password |
| Expected Output | Congratulations, you are successfully logged in. |
| Description | If the email and the password entered by the user matches the database, then they are logged in. |

|  |  |
| --- | --- |
| ID | 2.2 |
| Test Input | 1. Email 2. Password |
| Expected Output | Please enter correct email and password. |
| Description | If the email and the password entered by the user does not match the database, then they cannot log in. |

1. User home page:

|  |  |
| --- | --- |
| ID | 4.1 |
| Test Input | The page should have buttons for Bill Payment, View Bill details |
| Expected Output | Once the button is clicked, it should re-direct to the bill payment page or the view bill details page. |
| Description | The home page should display buttons and it should re-direct accordingly. |

|  |  |
| --- | --- |
| ID | 4.2 |
| Test Input | The page should have filters for the view page |
| Expected Output | Filters should be present on a drop-down menu which includes Sorting options and year selection. |
| Description | The filters should be applied to the bill’s list and display it accordingly. |

1. View User details page:

|  |  |
| --- | --- |
| ID | 5.1 |
| Test Input | Enter customer ID and password |
| Expected Output | Validate the credentials and display the customer details |
| Description | Customer details page is visible to the authorized customer only. Thus, the credentials that are entered must be validated. But the more important function is that it should only display but not edit. |

1. Edit User details page:

|  |  |
| --- | --- |
| ID | 6.1 |
| Test Input | The page allows the user to edit the details. |
| Expected Output | The changes made on this page should notify the database about the changes |
| Description | Through this page, the customer can make all the customer details like Address and  phone number. |

1. Edit Bill Page:

|  |  |
| --- | --- |
| ID | 7.1 |
| Test Input | Payment not issued / not paid / payment error |
| Expected Output | Pop-up message for Payment Due |
| Description | A message should be present on the customer portal if the payment is not issued, also for a prototype text message must be send to the customer. |

|  |  |
| --- | --- |
| ID | 7.2 |
| Test Input | To edit the customer details |
| Expected Output | The customer details can only be edited by the admin |
| Description | The Edited details must be changed in the database as well. |

1. View Complaints Page:

|  |  |
| --- | --- |
| ID | 9.1 |
| Test Input | Feedback form filled by the customer with 500 characters |
| Expected Output | Message must be received by the admin. It must contain less than 501 characters. The error message must appear if the submission done exceeding 500 characters. |
| Description | Message must have characters up to 500. The admin must be received the message. |

|  |  |
| --- | --- |
| ID | 9.2 |
| Test Input | Feedback form filled by the customer with type of complain scroll down menu |
| Expected Output | The scroll down must work properly. |
| Description | The complaint page must have a scroll down menu to select the category. |

1. Complaints Details Page:

|  |  |
| --- | --- |
| ID | 10.1 |
| Test Input | The complaint submitted by the customer must be available for him/her to review. |
| Expected Output | It should be assorted according to the date. It should mention the category, details, category, and the status of that complaint. |
| Description | The complaint windows should be accessible to the customer as well. It should have all the details. |