|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objective | Pre-requisites | Test Conditions | Data Functional Verification | Expected Result(s) |
| Verify that the user can log in. | User: Admin/Manager/Warden is registered in the system. | 1. User enters Username and password in the login form. 2. User is registered in the system. | 1. User can enter data in the fields. 2. Check if the mandatory fields are empty and exists in the system. 3. Upon clicking the login button user can login. | 1. System should allow the user to enter data in the form. 2. System should display error message. 3. System should allow the user to log in with correct credentials. |
| Verify that the user can add lease(s). | User: Admin/Manager is logged in. | 1. User invoked the lease form upon clicking the create lease button. 2. Room cleaning status is set to “clean” or “dirty”. | 1. User can enter data in the fields of the form. 2. Check if the mandatory fields are empty. 3. Form can be cancelled any time by clicking cancel button. 4. Upon clicking the create lease button user can save lease. | 1. System should allow user to enter data in the form. 2. System should display error message. 3. System clear all fields of the form. 4. System should cancel the lease. 5. System should save the lease. |
| Verify that the system should calculate and display the lease duration. | User: Admin/Manager is logged in. | 1. User chooses the “view lease” option. | 1. Date is in correct format. 2. Calculated lease duration is displayed in “view lease”. | 1. System should calculate lease duration and display in the correct field. |
| Verify that the system makes sure the lease Id is unique. | User: Admin is logged in. | 1. User has saved a lease. | 1. Unique id is displayed in the lease list with lease details. | 1. System should not allow the lease id to be the same for multiple leases. 2. System should display unique id with the lease details. |
| Verify that every field in the lease form only accepts respective input. | User: Admin is logged in. | 1. Lease form is invoked by the user and is displayed to the user. | 1. All fields must only accept required input in the fields i.e., Student Name field should only accept letters supported by the alphabet. | 1. System should only allow correct input in the fields. |
| UPDATING THE LEASE | | | | |
| Verify that user can select a lease from existing lease list. | User: Admin/Manager/Warden is logged in. | 1. Lease list is displayed to the user. 2. User selected a lease and clicked the button “Update Lease” to invoke the update lease interface. 3. Lease is already entered in the system. | 1. Upon invoking the update lease interface, lease information is displayed to the user. 2. User can make changes to the fields. 3. User can clear the update lease interface. 4. Upon clicking update button, information is updated. 5. Mandatory fields must be filled. | 1. System should display. 2. System should allow users to make changes in the field where they have privilege. 3. System should clear. 4. System should update the form. 5. System should check and if empty generate error message. |
| Verify that the system saves the changes against the same unique id. | User: Admin/Manager/Warden is logged in. | 1. Unique id is already inputted by the user against a lease. 2. Invoked lease exists in the system with a unique id. | Upon updating lease, the system must save the lease against the existing lease allotted if by the user. | 1. System should update lease against same id. |
| Verify that every field in the lease form accepts respective input for updating. | User: Admin is logged in. | 1. Update lease interface is invoked. | 1. All fields must only accept required input in the fields i.e., Student Name field should only accept letters supported by the alphabet. | 1. System should only allow correct input in the fields. |
| **ADD STUDENTS/USERS/HALLS ETC.** | | | | |
| Test the input values for adding a user. | User: Admin is logged in. | 1. Mandatory fields are entered. | 1. User can enter data in the desired fields. 2. Check if any mandatory fields are missing. | 1. Systems should only accept valid date. 2. System shall respond if mandatory fields are missing. |
| Verify that the User ID is unique for adding a student. | User: Admin is logged in. | 1. Mandatory fields are entered. | 1. System generates student Id. | 1. System should only accept a Unique ID. 2. System shall respond if student ID is mandatory field and/or missing. |
| Test the input values for adding a room. | User: Admin is logged in. | 1. Mandatory fields are entered. | 1. User can enter data in the desired fields. 2. Check if any mandatory fields are missing. | 1. System should only accept valid data. 2. System shall respond if mandatory fields are missing. |
| Test the input values for adding a hall. | User: Admin is logged in. | 1. Mandatory fields are entered. | 1. User can enter data in the desired fields. | 1. System should accept valid data. |
| **UPDATE USERS/ROOMS/HALLS.** | | | | |
| Test the input values for editing a room. | User: Admin is logged in. | 1. Add room with mandatory fields. | 1. User click on edit room and re-enters desired fields. | 1. System accepts valid data only. 2. System shall respond if mandatory fields are missing. |
| Test the input values for editing halls. | User: Admin is logged in. | 1. Add hall with mandatory fields. | 1. User click on the edit hall and re-renters desired fields. | 1. System accepts valid data only. 2. System shall respond if mandatory fields are missing. |
| Test the input values for editing users. | User: Admin is logged in. | 1. Add user with mandatory fields. | 1. User click on the edit user and re-enters desired fields. | 1. Systems accepts valid data only. 2. System shall respond if mandatory fields are missing. |