|  |  |
| --- | --- |
| **Use case ID:** 1 | **Use case name:** Record Payment |
| **Test number: 1** | |
| **Objective:** Test the primary path | |
| **Set up:** Customer name “Hello Magazine” must be set up and have outstanding balance of £187.18 of unpaid jobs. Date 2018/04/15 | |
| **Expected results:**   1. Payment is stored in the Database. 2. Payment amount deducts from the customer’s bill. | |
| **Test:**   1. Select the Payment functionality, select “Hello Magazine” 2. Enter the amount paid: £187.18. 3. Select date 2018/04/15. 4. Click on payment button. | |
| **Test record: Payment recorded successfully** | |
| **Date: 2018/04/16** | **Tester: Patrick G** |
| **Result: Passed** | |
| **Date:** | **Tester:** |
| **Result:** |  |

|  |  |
| --- | --- |
| **Use case ID:** 2 | **Use case name:** Record Payment: CardPayment |
| **Test number:** **1** | |
| **Objective:** Test the alternative path | |
| **Set up:** Customer name “Hello Magazine” must be set up and have outstanding balance of £187.18 of unpaid jobs. Customer card last 4 digits 9090. Expiry 01/20. (MasterCard). Date 2018/04/15 | |
| **Expected results:**   1. Payment is stored in the Database. 2. Payment card details are stored in the Database. 3. Payment amount deducts from the customer’s bill. | |
| **Test:**   1. Select the Payment functionality, select “Hello Magazine” 2. Enter the amount paid: £187.18. 3. Select the card payment option. 4. Select date (2018/04/19), 5. Enter card type (MasterCard), number (9090), expiry date (01/20). 6. Click on payment button. | |
| **Test record: Card payment recorded successfully** | |
| **Date: 2018/04/16** | **Tester: Patrick G** |
| **Result: Passed** | |
| **Date:** | **Tester:** |
| **Result:** |  |

|  |  |
| --- | --- |
| **Use case ID:** 3 | **Use case name:** Add User |
| **Test number: 1** | |
| **Objective:** Test the functionality of adding a user. | |
| **Set up:** User Manager with user type Office Manager is logged in. | |
| **Expected results:**   1. Database creates a new user. 2. Newly created user account has correct privileges. 3. The System informs that user account has been successfully created. | |
| **Test:**   1. Select User functionality. 2. Enter details for the user account (Username: John, Password: John1, User type: Office Manager, User Location: Finishing room). 3. The System informs that user has been successfully created. | |
| **Test record: User created successfully** | |
| **Date: 2018/04/16** | **Tester: Patrick G** |
| **Result: Passed** | |
| **Date:** | **Tester:** |
| **Result:** |  |

|  |  |
| --- | --- |
| **Use case ID:** **4** | **Use case name:** Add User: AlreadyExists |
| **Test number:** | |
| **Objective:** Test the System’s response to adding user of the same username | |
| **Set up:** User Manager with user type Office Manager is logged in. User with username Manager exists. | |
| **Expected results:**   1. The System informs the user that there already is user with the same username | |
| **Test:**   1. Select “User” functionality, 2. Enter details for the user account (Username: Manager, Password: Random123, User type: Shift Manager, User Location: Copy room). 3. The System informs that there already is user with the same username. | |
| **Test record: Alert comes up, user account has not been created** | |
| **Date: 2018/04/16** | **Tester: Patrick G** |
| **Result: Passed** | |
| **Date:** | **Tester:** |
| **Result:** |  |

|  |  |
| --- | --- |
| **Use case ID:** **5** | **Use case name:** Automatic backup |
| **Test number:** | |
| **Objective:** Test the automatic backup functionality. | |
| **Set up:** Automatic backup period (Day: 1, Hours:0, Minutes:0 (every day) is specified. | |
| **Expected results:**   1. The database server is backed up every set time period(every day) | |
| **Test:** 1. Specified time occurs (Each day), the System automatically creates new database backup. | |
| **Test record: Backup occurs each day.** | |
| **Date: 2018/04/16** | **Tester: Patrick G** |
| **Result: Passed** | |
| **Date:** | **Tester:** |
| **Result:** |  |

|  |  |
| --- | --- |
| **Use case ID:** **6** | **Use case name:** Update existing task |
| **Test number: 1** | |
| **Objective:** Test the “Update existing task” functionality. | |
| **Set up:** Office Manager (Username: Manager) is logged in. Task (ACT108) for “InfoPharma Ltd” is created. | |
| **Expected results:**   1. Task has been updated. (User: Manager) 2. The updated task user (Manager) corresponds to the one in the database records. | |
| **Test:**   1. Select Job Processing->Select “InfoPharma Ltd”-> Click on the task ACT108. 2. Change User from “Development” to “Manager”. 3. Check whether task User value (Manager) corresponds to the one in the database records. | |
| **Test record: Task has been successfully updated** | |
| **Date: 2018/04/16** | **Tester: Patrick** |
| **Result: Passed** | |
| **Date:** | **Tester:** |
| **Result:** |  |

|  |  |
| --- | --- |
| **Use case ID:** **7** | **Use case name:** Update existing task: NoUser |
| **Test number:** | |
| **Objective:** Test the System’s response to updating task with unexisting user. | |
| **Set up:** Office Manager is logged in (Manager). Unexisting user with username: Fake. | |
| **Expected results:**   1. The System doesn’t change the task value for User. | |
| **Test:**   1. Select Job Processing->Select “InfoPharma Ltd”-> Click on the task ACT108. 2. Change User from “Development” to “Fake”. 3. Check whether task User value (Development) corresponds to the one in the database records. | |
| **Test record: Task has been unchanged** | |
| **Date: 2018/04/16** | **Tester: Patrick** |
| **Result: Passed** | |
| **Date:** | **Tester:** |
| **Result:** |  |