|  |  |
| --- | --- |
| **Use case ID:** 1 | **Use case name:** Record Payment |
| **Test number:** | |
| **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 2018/04/19 | |
| **Expected results:**   1. The System created a payment object. 2. The System found the outstanding (unpaid) job/s of the customer and added them to the payment object. 3. The payment object is stored in the Database. | |
| **Test:**   1. Select the Payment functionality, select “Hello Magazine”, enter the amount paid £187.18 make sure that the System created and stored the payment object with correct recorded payment amount £187.18 . | |
| **Test record: Passed** | |
| **Date: 2018/04/20** | **Tester: Patrick** |
| **Result:** | |
| **Date:** | **Tester:** |
| **Result:** |  |

|  |  |
| --- | --- |
| **Use case ID:** | **Use case name:** Record Payment: CardPayment |
| **Test number:** | |
| **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 2018/04/19 | |
| **Expected results:**   1. The System created a payment object. 2. The System created card object. 3. The System found the outstanding (unpaid) job/s of the customer and added them to the payment object. 4. Payment object is referenced to the card object and has correct last digits (9090) and expiry (01/20) | |
| **Test:**   1. Select the Payment functionality, select “Hello Magazine” enter the amount paid £187.18, Select the card payment option. Select date (2018/04/19), enter card type(MasterCard), number(9090), expiry date(01/20). | |
| **Test record: Passed** | |
| **Date: 2018/04/20** | **Tester: Patrick** |
| **Result:** | |
| **Date:** | **Tester:** |
| **Result:** |  |

|  |  |
| --- | --- |
| **Use case ID:** | **Use case name:** Add User |
| **Test number:** | |
| **Objective:** Test the functionality of adding a user. | |
| **Set up:** User Manager with user type Office Manager is logged in. | |
| **Expected results:**   1. User inputs 2. Newly created user account has correct privileges. 3. Database creates a new user entry. 4. The System informs that user account has been successfully created. | |
| **Test:** 1. Select User functionality, enter details for the user account (Username: John, Password: John1, User type: Office Manager, User Location: Finishing room). The System informs that user has been successfully created. | |
| **Test record: Passed** | |
| **Date: 2018/04/20** | **Tester: Patrick** |
| **Result:** | |
| **Date:** | **Tester:** |
| **Result:** |  |

|  |  |
| --- | --- |
| **Use case ID:** | **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 John exists. | |
| **Expected results:**   1. The System informs the user that there already is user with the same username | |
| **Test:** 1. Select “User” functionality, enter details for the user account (Username: John, Password: John2, User type: Shift Manager, User Location: Copy room). The System informs that there already is user with the same username. | |
| **Test record: Passed** | |
| **Date: 2018/04/20** | **Tester: Patrick** |
| **Result:** | |
| **Date:** | **Tester:** |
| **Result:** |  |

|  |  |
| --- | --- |
| **Use case ID:** 9 | **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: Passed** | |
| **Date: 2018/04/20** | **Tester: Patrick** |
| **Result:** | |
| **Date:** | **Tester:** |
| **Result:** |  |

|  |  |
| --- | --- |
| **Use case ID:** 11 | **Use case name:** Update existing task |
| **Test number:** | |
| **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. Change User from “Development” to “Manager”. Check whether task User value (Manager) corresponds to the one in the database records. | |
| **Test record: Passed** | |
| **Date: 2018/04/20** | **Tester: Patrick** |
| **Result:** | |
| **Date:** | **Tester:** |
| **Result:** |  |

|  |  |
| --- | --- |
| **Use case ID:** | **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. Change User from “Development” to “Fake”. Check whether task User value (Development) corresponds to the one in the database records. | |
| **Test record: Passed** | |
| **Date: 2018/04/20** | **Tester: Patrick** |
| **Result:** | |
| **Date:** | **Tester:** |
| **Result:** |  |