|  |
| --- |
|  |
| Acceptance Test Plan |
| Indus Recreational Centre System |
|  |
| **John Stuby, Jorge Pinochet, Corey Cantley, Kerri Lynch-O’Loughlin** |
| **3/30/2010** |

|  |
| --- |
|  |

Indus Recreational Centre System

Acceptance Test Plan

**Team Members:**

Corey Cantley

John Stuby

Jorge Pinochet

Kerri Lynch – O’ Loughlin

**Client:**

Indus Recreational Centre

**Instructors:**

Randy Kaltenbach

**Due Date:**

March 31, 2010

Indus Recreational Centre System

# Acceptance Test Plan

Acceptance Test Objective:

This test will verify that the system meets the Client’s requirements and performs to the satisfaction of the Client.

Acceptance Test Entry Criteria:

The system will have successfully completed Module, Unit and System testing to the satisfaction of the system designers.

Acceptance Test Exit Criteria:

The system will successfully meet the criteria as identified by the client.

Acceptance Test Responsibility:

The criteria identified by the client will be tested by Corey Cantley, Jorge Pinochet, Kerri Lynch O’Loughlin, and John Stuby and we will initiate and complete the tests, verify the test results, and declare that the Test Exit Criteria have been met.

Test Procedures:

Specific test procedures will be identified on the Testing Form for each functional requirement and each non-functional requirement.

## Acceptance Test Criteria

The following criteria have been identified as necessary for Client approval of Indus Recreational Center System.

Generating a schedule:

Add a booking

Remove a booking

Modify an existing booking

Available time for facility

Choosing a facility

Adding a booking

Create a new user

Surname

Event title

Event type

Start time

End time

Setup time

Tear down time

Rate

Additional Charges

Invoice number

Database Maintenance:

Client

Add a new client

Delete a client

Modify information for an existing client

Facility

Add a new facility

Delete a facility

Modify information for an existing facility

Rate

Add a new rate

Delete a rate

Modify information for an existing rate

Employee

Add a new employee

Delete a employee

Modify information for an existing employee

Organization

Add a new organization

Delete a organization

Modify information for an existing organization

Additional Charge

Add a new additional charge

Delete a additional charge

Modify information for an existing additional charge

Booking Type

Add a new booking type

Delete a booking type

Modify information for an existing booking type

Invoice

Add a new invoice

Delete a invoice

Modify information for an existing invoice

Booking Creation:

Create a booking that allows for:

Automatic generation of a booking\_id

Data entry of:

Client name

Surname

Event title

Event type

Start time

End time

Setup time

Tear down time

Rate

Additional Charges

Invoice number

Save and keep booking on active status until complete

Cancel the generation of the current booking

Update all information on an existing booking, with specific attention to:

Adjusting Client name and surname

Adjusting Event title

Adjusting Event type

Adding additional charges

Adjusting start time

Adjusting end time

Delete an existing booking

Invoice Creation:

Correct calculation of price including additional charges, rates, subtotal, GST, and total

Report Generation:

Reports are to be generated automatically on the first or last day of the month, and also when requested by the Client. Report modification by the Client is not required.

The following reports have been identified as required by the Client:

Email Customer report:

Client/Organization Name, Billing Address (street address, city, province, postal code), Invoice number, bookings.

Print Customer Report:

Client/Organization Name, Billing Address (street address, city, province, postal code), Invoice number, bookings.

Export invoices to QuickBooks

Client/Organization name, invoice numbers, bookings, etc

Non-Functional Requirements Tests (may be incorporated into Functional Tests)

Database size – At least 1000 customers and reasonable associated data can be accommodated.

Speed – Each report will be generated within 5 seconds

Multiple database connections support. All the IRC customers, estimated maximum 100 at any one time, and all the IRC employees, estimated maximum 15, should all be able to access a facilities booking schedule at the same time.

# Acceptance Testing Form

| **Client**: Indus Recreational center |  |
| --- | --- |
| **System**: Indus Recreational Center Scheduling System |  |
| **Objective**:  Verify that when a booking is created, all information is updated in the booking table located in database | |
| **Tester Name:** Kerri Lynch-O’Loughlin | |
| **Developer Name:** John Stuby | |

| **Component**: Create booking | | **Test Date:** |
| --- | --- | --- |
| **Test** | **Description** | **Actual Result** |
| 1. | *Select free time in schedule for facility*  **Procedure**:  Open the program  -The application should display the home page  Log in with a valid username and password  -Login with username as kerri and password as password  Click on the schedule tab  -Located on the top left in the menu bar  Click on a free time on the schedule  -Click on white space of the timetable displayed  **Expected Result**:  Opens up the booking screen  -A popup menu should appear with empty fields  -The time fields however will auto fill | The Booking screen is opened |
| 2. | *Fill in all information in booking screen*  **Procedure**:  Fill in all available fields in booking screen  -Fill in fields as follows:  -given name : test  -surname: test  -client: testClient  -Event name: testEvent  -BookingType: Hockey Game  -Rate: PrimeTime Ice 1  -Guest Count: 100  -Start time: N/A  -Length: 60 minutes  -Setup time: N/A  -Teardown Time: N/A  -Additional Charges: Ctrl click each of the following:  -Pop Fountain  -Ice Machine  **Expected Result**:  All fields are filled in  -Fields should contain information the user has input | All fields are filled in |
| 3. | *Click on save to update schedule to show new booking*  **Procedure**:  Click on save to update schedule and return to main schedule page for the selected facility  -The booking created should appear with appropriate time on the time schedule  **Expected Result**:  Save: updates schedule with new booking and returns to the schedule page for selected facility  -The time schedule from the booking will be filled into the schedule and colored red | On main schedule page, new booking is created for specified time frame with requested event name |
| 4. | *Click on delete to delete current booking and return to previous screen*  **Procedure:**  Click on delete to delete current booking and return to main schedule page for the selected facility  -Select the booking you just created. Click on the delete button  **Expected Result:**  Select the previously created booking  Click on the “delete” button: deletes current booking and return to main schedule page for the selected facility  -The time schedule will no longer display the booking that was scheduled | Returned to previous screen |

|  |  |
| --- | --- |
| **Comments**: | |
| **Client Signature of Approval**: | |
| **IT Use Only:**  Redesign Complete: 🞏 Yes 🞏 No  Retest Complete: 🞏 Yes 🞏 No  Details if Retest not complete: | |
| **Date:** | |
| **Developer Name and Signature:** | |
| **Client**: Indus Recreational Center |  |
| **System**: Indus Recreational Center Scheduling System |  |
| **Objective**:  Verify that invoice information is returned to the invoice screen and proper exporting options are taken | |
| **Tester Name:** Kerri Lynch-O’Loughlin | |
| **Developer Name:** John Stuby | |

| **Component**: Create Invoice | | **Test Date:** March 27, 2010 |
| --- | --- | --- |
| **Test** | **Description** | **Actual Result** |
| 1. | *Select bookings to add to invoice*  **Procedure**:  Open the program  -The application should display the home page  Log in with a valid username and password  -Login with kerri and password  Click on the accounting tab  -Located on the menu bar, top-left  Search for a client  -Use the available search box located by client and type in “testClient”  Click on correct client  **Expected Result**:  Opens up the invoice screen  -Two tables appear with available bookings and booking included on invoice | Invoice screen is opened |
| 2. | *Click on the selected bookings*  **Procedure**:  Select appropriate bookings available in booking table  Once all bookings selected  - Ctrl click the first 3 bookings on the list  -Click the *add to invoice* button  **Expected Result**:  All bookings selected will move from bookings table to invoice table | Selected bookings are moved from booking table to invoice table. |
| 3. | *Remove unnecessary bookings*  **Procedure:**  Select unnecessary bookings from invoice table  -Click the remove booking button  **Expected Result:**  All bookings selected will be removed from invoice table and put back in bookings table | Removes the unnecessary bookings selected from the invoice table, and returns them to the booking table. |
| 4. | *Save the invoice*  **Procedure**:  Click on the save button  -Located at the bottom of the screen  **Expected Result**:  Invoice list will be updated in accounting list | Invoice list is updated to include the new invoice |
| 5. | *Select option to pay for invoice*  **Procedure**:  Click on the due date located in the paid column  **Expected Result**:  Payment window pops up  -First field shows payment amount for invoice  -Second field is open allowing the user to choose amount to pay. | Payment window pops up, the first field shows required payment, second field is for the user to input the amount they want to pay for invoice at current time |
| 6. | *Click on the save*  **Procedure**:  Click on save to update paid column in invoice table and return to previous page  **Expected Result**:  Return to accounting page with updated information | Returns to the accounting page with updated information for the invoice. The information in the paid column will be updated to paid if the entire amount of invoice was paid. |
| 7. | *Delete the invoice*  ***Procedure:***  Click on the invoice created earlier.  Click on the delete button if invoice is not to be completed  -Located at the bottom of the screen  **Expected Result:**  Invoice will be deleted and the user will be returned to accounting screen | Invoice is deleted and user is returned to accounting screen. |
| 8. | *Print an invoice*  **Procedure**:  Click on print icon  Select printer MD201  **Expected Result**:  Invoice is sent to printer. Invoice printed | Printer menu pops up, after selecting which printer to print to, the invoice is sent to the printer. |
| 9. | *Email an invoice*  **Procedure**:  Click on email icon  **Expected Result**:  Invoice is emailed to selected customer | Invoice is emailed to selected customer |
| 10. | *Export to Excel*  **Procedure**:  Click on invoice icon  **Expected Result**:  Invoice is exported to excel, it is exported in CSV format | Invoice is exported to excel, it is exported in a CSV format. |
| **Comments**: | | |
| **Client Signature of Approval**: | | |
| **IT Use Only:**  Redesign Complete: 🞏 Yes 🞏 No  Retest Complete: 🞏 Yes 🞏 No  Details if Retest not complete: | | |
| **Date:** | | |
| **Developer Name and Signature:** | | |

| **Client**: Indus Recreational Center |  |
| --- | --- |
| **System**: Indus Recreational Center Scheduling System R1.0 |  |
| **Objective**:  Verify that schedule is generated correctly and shows all created bookings | |
| **Tester Name:** Kerri Lynch-O’Loughlin | |
| **Developer Name:** John Stuby | |

| **Component**: Generate Schedule | | **Test Date:** March 27, 2010 |
| --- | --- | --- |
| **Test** | **Description** | **Actual Result** |
| 1. | *Select free time in schedule for facility*  **Procedure**:  Open the program  -The application should display the home page  Log in with a valid username and password  -Login with username as test and password as password  Click on the *Schedule tab* button.  -Click of the Schedule tab on the top menu bar  -Select a different facility: Banquette Hall  -Select a monthly time view for next month  **Expected Result**:  Opens up the schedule for the selected facility and timeframe | Opens up schedule for the selected facility and time frame |
| 2. | *Select an existing booking*  **Procedure**:  Click on booking in the schedule for the requested facility  -Click on a red booking called “changeThis”  **Expected Result**:  Brings up information on selected booking  -The booking details , will now be displayed | Brings up information on selected booking  -The booking details, will now be displayed |
| 3. | *Change existing booking*  **Procedure**:  Inside of booking screen change information if necessary, change first name, surname, address  -Change event title to testing  **Expected Result**:  The updated details of the booking will be changed | Booking being updated will have details that were changed updated |
| 4. | *Save changed booking*  **Procedure**:  Select save button  -Single click upon the button labeled “Save”  **Expected Result**:  Updates booking information and schedule for the selected facility  -The booking that was previously selected will now be updated | Updates booking information and schedule for the selected facility  -The booking that was previously selected will now be updated |
| 5. | *Delete existing booking*  **Procedure**:  Inside of booking screen delete booking  -While in the detailed booking notes select the delete button  **Expected Result**:  Deletes booking from schedule for facility and returns to schedule with updated bookings  -The schedule will be updated and the free time, white space, will be displayed where the booking use to take place | Deletes booking from schedule for facility and returns to schedule with updated bookings  -The schedule will be updated and the free time, white space, will be displayed where the booking use to take place |
| 6. | *Create new booking for facility for schedule*  **Procedure**:  Select empty timeframe for selected facility  -Select white space that is not colored in  **Expected Result**:  Opens up new booking screen  -A popup screen will prompt you for event details | Opens up new Booking screen.  - A popup screen will prompt for event details |
| 7. | *Create new booking*  **Procedure**:  Fill in all requested fields for booking  -Fill in fields as follows:  -given name : test  -surname: test  -client: testClient  -Booking Type: Wedding  -Rate: Night Banquette Standard  -Guest Count: 300  -Start time: N/A  -Length: 60 minutes  -Setup time: N/A  -Teardown Time: N/A  -Additional Charges: Ctrl-click each of the following:  -Bar Service  -Bar BQ  **Expected Result**:  All selected fields are filled  -There should no longer be anymore blank spaces unless previously indicated. | All selected fields are filled  -There should no longer be anymore blank spaces unless previously indicated. |
| 8. | *Save new booking*  **Procedure**:  Click on save button  -In the details popup menu, select the button named “save”  **Expected Result**:  New booking is created  Schedule for facility is updated to include new booking | New booking is created  Schedule for facility is updated to include new booking |
| 9. | *Delete Booking*  **Procedure**:  Click on the the previously created booking  Click on *delete*  -In the booking details popup menu, select the button named “delete”  **Result**:  All fields are set to original state and user is returned to previous screen  -The schedule will update with whitespace where the booking use to be in. | All fields are set to original state and user is returned to previous screen  -The schedule will update with whitespace where the booking use to be in. |
| **Comments**: | | |
| **Client Signature of Approval**: | | |
| **IT Use Only:**  Redesign Complete: 🞏 Yes 🞏 No  Retest Complete: 🞏 Yes 🞏 No  Details if Retest not complete: | | |
| **Date:** | | |
| **Developer Name and Signature:** | | |