|  |
| --- |
| ICA Documentation |

|  |  |
| --- | --- |
| **Module Title: Web Apps and Services** | Module Leader: Soo Lian Kei |
| **Module Code: TU Module** |
| **Assignment Title: Event Management System** | **Deadline Date: 9th June 2020** |
| **Deadline Time: 5.00PM** |
| **Submission Method:**  Online |

|  |
| --- |
| **Message from Module Leader**  **Submission Files:**  You should save all the files (include the Design Document in PDF format, all databases and all source codes) in a Zip file.  **Submission Method:**  You should submit your project via one of the following methods:  **1. Submission via Email**  You can send an email to me, attached with your project Zip files to ***liankei.soo@firstcity.edu.my***. You should make sure that the sharing is ON and make sure that I can access/download the attached Zip file.  **2. Submission via Whatsapps**  You can send the project Zip file to me via Whatsapps @ ***011-65160287***. |

|  |  |  |
| --- | --- | --- |
| **Student Name** | **Student ID** | **Class Code** |
| Choo Han Yang | B0633 | BSE |

# Marking Guidelines (PART 1)

To achieve full marks you must achieve each aim as defined in the following tables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criterion** | **Excellent (85-100%)** | **Very Good (70-85%)** | **Good (55-70%)** | **Satisfactory (40-55%)** |
| **Product  (35%)** | Product executes without causing run-time exception or error.  Product meets or exceeds all of the functional requirements.  User interface and application workflow have been appropriately customised. | Product executes with appropriate bespoke pages upon error.  Product meets most functional requirements.  User interface and application workflow have some customisation. | Product executes with infrequent uncaught run-time exceptions.  Product meets all of the MUST and SHOULD functional requirements.  User interface has some customisation but application workflow is mostly scaffolded. | Product executes but has frequent uncaught run-time exceptions.  Product meets all of the MUST functional requirements.  User interface and application workflow is as scaffolded. |
| **Mark** |  | | |
| **Comments** |  | | |
| **Technique  (30%)** | ASP.NET and EntityFramework features are used extensively and successfully.  Services demonstrate correct use of HTTP with suitable end-points.  View Models and Data Transfer Objects are used appropriately.  Code design shows evidence of layering around presentation, business rules and data access.  AJAX calls are used to improve the user interface. | ASP.NET and EntityFramework features are used extensively and successfully.  Created services demonstrate successful, but not RESTful, use of HTTP with various end-points.  View Models and Data Transfer Objects are used appropriately.  Code design shows little evidence of layering.  AJAX calls are used to improve the user interface. | Most of the demonstrated features of ASP.NET and EntityFramework are used successfully.  Given HTTP services are used successfully.  Some additional View Models and Data Transfer Objects are used.  Code design shows little evidence of layering or design pattern usage.  AJAX calls are not used. | ASP.NET and EntityFramework are generally used successfully.  Given HTTP services are used successfully.  Only given View Models and Data Transfer Objects are used.  Code design shows little evidence of layering or design pattern usage.  AJAX calls are not used. |
| **Mark** |  | | |
| **Comments** |  | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Documentation  (15%)** | Code documentation identifies the key problems and challenges of the assignment and details alternative solutions to them.  Justification is given for choices made with reference to suitable knowledge sources.  Documentation is short because it is clear and concise. | Code documentation identifies the key problems and challenges of the assignment with some detail of alternative solutions.  Justification is given for choices made but without reference to suitable knowledge sources.  Documentation feels padded in places. | Code documentation identifies the key problems and challenges of the assignment.  There is no justification given for choices made.  Documentation frequently feels padded, too short or overly long. | Aspects of the code are documented, but there is no documentation that discusses the key problems or challenges of the assignment.  There is no justification given for choices made.  Documentation exists but is incomplete. |
| **Mark** |  | | |
| **Comments** |  | | |

# Marking Guidelines (PART 2)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criterion** | **Excellent (85-100%)** | **Very Good (70-85%)** | **Good (55-70%)** | **Satisfactory (40-55%)** |
| **Testing  (20%)** | Testing plan is extensive with multiple tests to verify each functional requirement.  Test results are complete, indicating the outcome of each test and whether the functional requirement is successfully verified.  The overall evaluation is justified by the test plan and results. | Testing plan is extensive with multiple tests to verify most functional requirement.  Test results are almost completed, indicating the outcome of most test and whether the functional requirement is successfully verified.  The overall evaluation is justified by the test plan and results. | Testing plan is extensive with multiple tests to verify some functional requirement.  Test results are partially indicating the outcome of partial test and whether the functional requirement is successfully verified.  The overall evaluation is roughly justified by the test plan and results. | Testing plan is extensive with few tests to verify each functional requirement.  Test results are incomplete and not indicating the outcome of each test and whether the functional requirement is successfully verified.  The overall evaluation is less justified by the test plan and results. |
| **Mark** |  | | |
| **Comments** |  | | |

# TOTAL MARK:

Contents

[Test Plan 5](#_Toc42610204)

[Overall Evaluation 12](#_Toc42610205)

# Test Plan

The following user-based test plan are prepared based on the requirements stated on the project briefing. The test cases are to test all the inputs from user and check if the actual result match with expected result or not. Comments on how to fix the problem in the test cases that the actual result did not match with the expected result is recorded down.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User-based Test Plan (MUST)** | | | | |
| **Id** | **Description/**  **Purpose** | **Data/**  **Procedure** | **Expected Result** | **Actual Results/Comments** |
| 1 | Create new Customer | Surname: Choo  First Name: Han Yang  Email: [hanyang@gmail.com](mailto:hanyang@gmail.com) | New customer added to list. | New customer added to list. |
| 2 | Create new Customer with some blank fields | Surname: Choo  First Name: Han Yang  Email: *blank* | Customer creation rejected (need email) | Customer creation rejected (need email) |
| 3 | Create new Customer with some blank fields | Surname: Choo  First Name: *blank*  Email: [hanyang@gmail.com](mailto:hanyang@gmail.com) | Customer creation rejected (need first name) | Customer creation rejected (need first name) |
| 4 | Create new Customer with some blank fields | Surname: *blank*  First Name: Han Yang  Email: [hanyang@gmail.com](mailto:hanyang@gmail.com) | Customer creation rejected (need surname) | Customer creation rejected (need surname) |
| 5 | Display a list of customers | Click on Customer tab on the navigation bar | List of customers are displayed | List of customers are displayed |
| 6 | Edit customer | Surname: Law  First Name: Ben Ben  Email: [lbb@gmail.com](mailto:lbb@gmail.com) | Customer details are updated | Customer details are updated |
| 7 | Edit customer with some blank fields | Surname: Law  First Name: Ben Ben  Email: *blank* | Customer update rejected (need email) | Customer update rejected (need email) |
| 8 | Edit customer with some blank fields | Surname: Law  First Name: *blank*  Email: [lbb@gmail.com](mailto:lbb@gmail.com) | Customer update rejected (need first name) | Customer update rejected (need first name) |
| 9 | Edit customer with some blank fields | Surname: *blank*  First Name: Ben Ben  Email: [lbb@gmail.com](mailto:lbb@gmail.com) | Customer update rejected (need surname) | Customer update rejected (need surname) |
| 10 | Create new Event | Title: Padoru’s Wedding  Date: 1/12/2020 11:10 AM  Duration: 11:00:00  TypeId: WED | New event added to list | New event added to list |
| 11 | Create new Event with some blanks field | Title: *blank*  Date: 1/12/2020 11:10 AM  Duration: 11:00:00  TypeId: WED | Event Update rejected (need title) | Event Update rejected (need title) |
| 12 | Create new Event with some blanks field | Title: Padoru’s Wedding  Date: *blank*  Duration: 11:00:00  TypeId: WED | Event Update rejected (need date) | Event Update rejected (need date) |
| 13 | Create new Event with some blanks field | Title: Padoru’s Wedding  Date: 1/12/2020 11:10 AM  Duration: *blank*  TypeId: WED | Event Update rejected (need duration) | New event added to list.  Fixed this problem by making this field to be required field |
| 14 | Create new Event with some blanks field | Title: Padoru’s Wedding  Date: 1/12/2020 11:10 AM  Duration: 11:00:00  TypeId: *blank* | Event Update rejected (need typeid) | Event Update rejected (need typeid) |
| 15 | Edit an event | Title: Jojo’s Big 80  Duration: 05:00:00 | Event details are updated | Event details are updated |
| 16 | Edit an event with some blank fields | Title: Jojo’s Big 80  Duration: *blank* | Event update rejected (need duration) | Event details are updated.  Fixed this problem by making this field to be required field |
| 17 | Edit an event with some blank fields | Title: *blank*  Duration: 05:00:00 | Event update rejected (need title) | Event update rejected (need title) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 18 | Book a customer onto an event as a guest | Customer Name: Robert Robertson  Event: Jojo’s Big 80  Attended: *unchecked* | Guest Booking is created | Guest Booking is created |
| 19 |  | Customer Name: Jin Jellybeans  Event: Jojo’s Big 80  Attended: *checked* | Guest Booking is created | Guest Booking is created |
| 20 | List Guest with total count for an event | Click ‘Guest Booking’ besides any event | List of guests including total count of guest is displayed | List of guests including total count of guest is displayed |
| 21 | Register the attendance of guest | Click on ‘Update Attendance’ on the row that wanted to update, and check or uncheck the checkbox | The attendance of guest is updated | The attendance of guest is updated |
| 22 | Display details of a customer (including information about events which they are associated and their attendance | Click on Guest Booking tab on navigation bar | List of guests, information about the event that are associated, and their attendance are displayed | List of guests, information about the event that are associated, and their attendance are displayed |
| 23 | Cancel the booking of a customer from upcoming event | Click Delete on the guest booking row that wanted to be deleted | The booking is cancelled and removed from guest booking list | The booking is cancelled and removed from guest booking list |
| 24 | Reserve an appropriate available venue | Click on Reservations on the event row that wanted to reserve venue, pick the available venue and click Reserve | The venue is reserved for the event if there is appropriate and suitable venue is picked | The venue is reserved for the event if there is appropriate and suitable venue is picked |
| 25 | Free any associated venue | Click on Cancel on the event that wanted to free the venue | The venue is freed from the associated event | The venue is freed from the associated event |
| 26 | Display list of events with summary information about guest and venue | Click on Event tab on navigation bar | List of events with summary information are displayed | List of events with summary information are displayed |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User-based Test Plan (SHOULD)** | | | | |
| **Id** | **Description/**  **Purpose** | **Data/**  **Procedure** | **Expected Result** | **Actual Results/Comments** |
| 27 | Create new Staff | Surname: Choo  First Name: Han Yang  Email: [hy@gmail.com](mailto:hy@gmail.com)  First Aider: *checked* | New Staff added to list | New Staff added to list |
| 28 | Create new Staff with some blank fields | Surname: Choo  First Name: Han Yang  Email: *blanked*  First Aider: un*checked* | Staff creation rejected (need email) | Staff creation rejected (need email) |
| 29 | Create new Staff with some blank fields | Surname: Choo  First Name: b*lanked*  Email: [hy@gmail.com](mailto:hy@gmail.com)  First Aider: *checked* | Staff creation rejected (need first name) | Staff creation rejected (need first name) |
| 30 | Create new Staff with some blank fields | Surname: *blanked*  First Name: Han Yang  Email: [hy@gmail.com](mailto:hy@gmail.com)  First Aider: *checked* | Staff creation rejected (need surname) | Staff creation rejected (need surname) |
| 31 | Display a list of Staffs | Click on Staff tab on navigation bar | List of Staffs are displayed | List of Staffs are displayed |
| 32 | Edit Staff | Surname: Chow  First Name: Tzuyu  Email: [tzuyu@gmail.com](mailto:tzuyu@gmail.com)  First Aider: *checked* | Staff details are updated | Staff details are updated |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 33 | Edit Staff with some blank fields | Surname: Chow  First Name: Tzuyu  Email: *blanked*  First Aider: *checked* | Staff update rejected (need email) | Staff update rejected (need email) |
| 34 | Edit Staff with some blank fields | Surname: Chow  First Name: *blanked*  Email: [tzuyu@gmail.com](mailto:tzuyu@gmail.com)  First Aider: *checked* | Staff update rejected (need first name) | Staff update rejected (need first name) |
| 35 | Edit Staff with some blank fields | Surname: *blanked*  First Name: Tzuyu  Email: [tzuyu@gmail.com](mailto:tzuyu@gmail.com)  First Aider: *checked* | Staff update rejected (need surname) | Staff update rejected (need surname) |
| 36 | Assign available staff to event | Click Staff Booking tab in navigation bar, click on create new booking.  Pick the staff and event, then create | Staff that is picked is assigned to the event | Staff that is picked is assigned to the event |
| 39 | Remove assigned staff to event | Click Delete on the row of the staff that are going to be removed from event | The staff is unassigned from the event | The staff is unassigned from the event |
| 40 | Warn the user if there is no first-aider assigned to event | Remove any first-aider staff from event | Appropriate warning is displayed indicates that there is no first aider | Appropriate warning is displayed indicates that there is no first aider |
| 41 | Warn the user if there are fewer than one staff per 10 guests assigned to an event | Remove all the staffs assigned to the event | Appropriate warning is displayed indicates that the staff-guest ratio is imbalance | Appropriate warning is displayed indicates that the staff-guest ratio is imbalance |
| 42 | Display information about upcoming events that they are assigned to work | Click on the Event Booking on the staff that user wanted to check | Details and Information about upcoming events of the staff is displayed | Details and Information about upcoming events of the staff is displayed |
| 43 | Cancel an event | Click on cancel of the event that user wanted to cancel | Venue code of the event will be set back to default (empty)  All the assigned staff in the event will be unassigned | Venue code of the event will be set back to default (empty)  All the assigned staff in the event will be unassigned |
| 44 | Display details of specific event | Click on details of event that user wanted to check | Details of event is displayed along with invited guest and staff assigned | Details of event is displayed along with invited guest and staff assigned |
| 45 | Delete customer personal data permanently | Click delete on the customer that wanted to be deleted | Surname, first name and the email of the customer will be updated to “null” | Surname, first name and the email of the customer will be updated to “null” |

# Overall Evaluation

This project is a web-based event management intranet system that used for manage guests, events and venues.

The project is completed within the given timeframe, all the “MUST” and “SHOULD” functional requirements are achieved and fulfilled based on the test plans and results shown above.

Besides, there are some function requirements that are not done due to lack of time and lack of knowledge about ASP.NET Core framework. The requirements that are not done are fall under WOULD category of the functional requirements.

In addition, the user interface is customised to provide a better experience to the user. Besides, unused links are removed, and title of every page is modified to provide a better user experience.

There are a lot of problems faced during the development of the application, but most of the problems are managed to solve through online forums and seeking help from lecturer.