Software Process

Analysis and Design

Vendor:

ZSFJ Ltd - Zac Holmes, Sheridan Boyd, Fili Kurene & Jethro Thomas

Created on: 18/07/2020

**Table of Contents**

[**Executive Summary** 3](https://docs.google.com/document/d/1sINTyXu88bBRAfRsm5cVGji4zmnzgwm0QxZIiCZwO2E/edit#heading=h.gjdgxs)

[**Requirements** 3](https://docs.google.com/document/d/1sINTyXu88bBRAfRsm5cVGji4zmnzgwm0QxZIiCZwO2E/edit#heading=h.30j0zll)

[**Functional Requirements** 3](https://docs.google.com/document/d/1sINTyXu88bBRAfRsm5cVGji4zmnzgwm0QxZIiCZwO2E/edit#heading=h.1fob9te)

[**Non-functional Requirements** 5](https://docs.google.com/document/d/1sINTyXu88bBRAfRsm5cVGji4zmnzgwm0QxZIiCZwO2E/edit#heading=h.3znysh7)

[**Process Mapping** 8](https://docs.google.com/document/d/1sINTyXu88bBRAfRsm5cVGji4zmnzgwm0QxZIiCZwO2E/edit#heading=h.2et92p0)

[**Process Modelling** 9](https://docs.google.com/document/d/1sINTyXu88bBRAfRsm5cVGji4zmnzgwm0QxZIiCZwO2E/edit#heading=h.tyjcwt)

[**Use Case Model** 9](https://docs.google.com/document/d/1sINTyXu88bBRAfRsm5cVGji4zmnzgwm0QxZIiCZwO2E/edit#heading=h.3dy6vkm)

[**Class Diagram** 22](https://docs.google.com/document/d/1sINTyXu88bBRAfRsm5cVGji4zmnzgwm0QxZIiCZwO2E/edit#heading=h.1t3h5sf)

[**Data Flow Diagram** 23](https://docs.google.com/document/d/1sINTyXu88bBRAfRsm5cVGji4zmnzgwm0QxZIiCZwO2E/edit#heading=h.4d34og8)

[***Level 0*** 23](https://docs.google.com/document/d/1sINTyXu88bBRAfRsm5cVGji4zmnzgwm0QxZIiCZwO2E/edit#heading=h.2s8eyo1)

[***Level 1*** 23](https://docs.google.com/document/d/1sINTyXu88bBRAfRsm5cVGji4zmnzgwm0QxZIiCZwO2E/edit#heading=h.3rdcrjn)

[**Appendices** 24](https://docs.google.com/document/d/1sINTyXu88bBRAfRsm5cVGji4zmnzgwm0QxZIiCZwO2E/edit#heading=h.lnxbz9)

[**Appendix 1 – Task list** 24](https://docs.google.com/document/d/1sINTyXu88bBRAfRsm5cVGji4zmnzgwm0QxZIiCZwO2E/edit#heading=h.35nkun2)

# Executive Summary

(project brief)

ZSFJ Ltd have been engaged to develop an ASP.Net Core web application to aid BICT Lecturers and students in booking Test dates in the BICT computer labs.

It is expected that the project team will produce a prototype application that will allow Lecturers to set a Test Schedule for students to book the time they will take the test.

# Requirements

The Requirements of the prototype are as follows: -

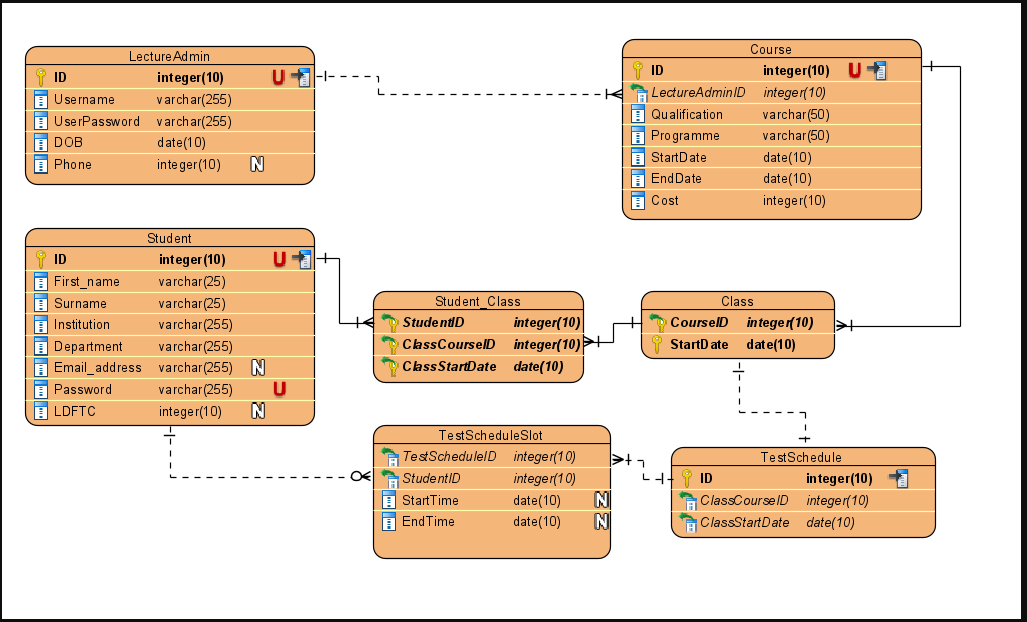
* The application should support multiple users and have two main user types (‘student’ and ‘lecturer admin’)
* The application should have a secure login.
* A class should be created for all courses
* A class should have a lecturer assigned to it
* A class should be populated with enrolled students
* A student should be able to be in many classes (currently on Active Directory you belong to a D202 group, and also other groups for your other courses)
* Lecturers should be able to create a Test Schedule for a class
* The Test Schedule should be made up of customisable time slots defined by the lecturer at time of creation
* Students in the class should be able to access the Test Schedule and book an available time slot
* Lecturers should be able to remove a student from a time slot if they have made a mistake choosing their time
* Lecturers should be able to use the application to reset any student’s lab password

A student should be able to reset their own password

# Solution Design

### Entity Relationship Diagram (ERD)

Below is the Entity Relationship Design (ERD) Diagram outlining the entity relationships for the solution design of the application.



### **Database Structure**

Run the SQL queries in the SQLQueries folder to initialise the database prior to first run of the application.

### Process Modelling

The following modelling resources have been created to provide details of the system structure and system behaviour: -

* Product Backlog User Stories
* UML diagrams & Use Case Models

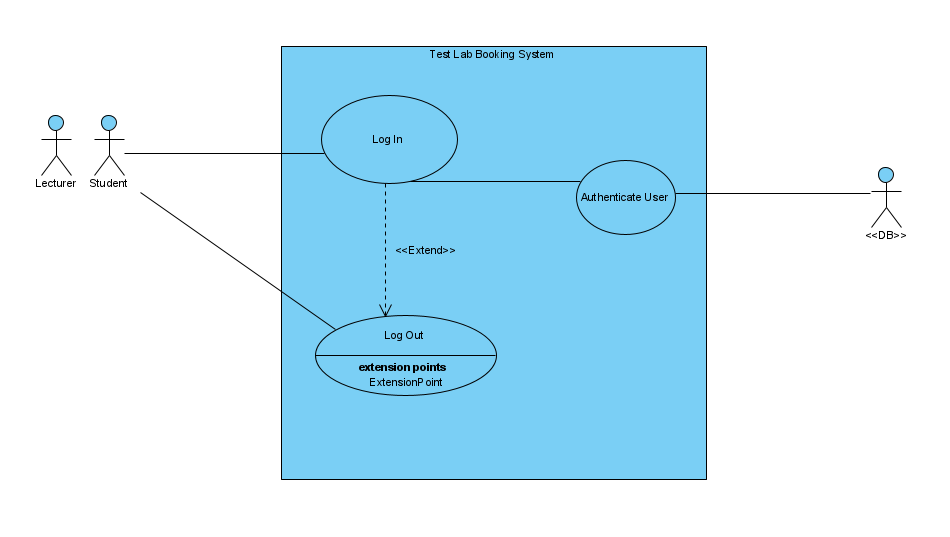
### User Stories

|  |  |  |
| --- | --- | --- |
| **User Story No.** | **User Story** | **Planned Sprint** |
| **1** |  |  |
| **2** |  |  |
| **3** |  |  |
| **4a** |  |  |
| **4b** |  |  |
| **5a** |  |  |
| **5b** |  |  |
| **6** |  |  |
| **7** |  |  |
| **8** |  |  |

### Use Cases

|  |  |
| --- | --- |
| **Use Case** | **Description of Use Case Scenario** |
| Log In / Log Out |  |
| Class Setup (create, add, delete) |  |
| Book Test Schedule (create, add, delete) |  |
| Reset Password |  |

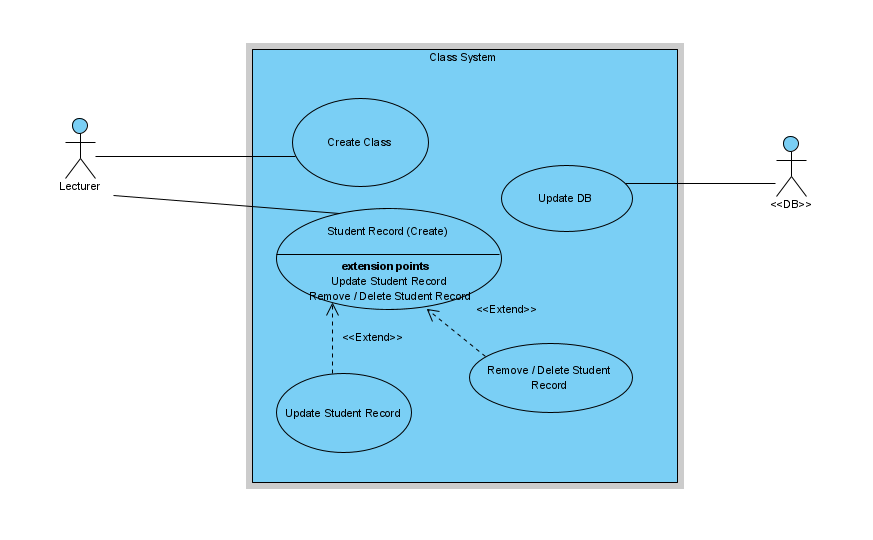
**Log In / Log Out Use Case**



|  |  |
| --- | --- |
| **Name** | **Log In** |
| **Identifier** | **UC 01** |
| **Description** | This use case describes how a user Logs Into the System |
| **Goal** | Log User into the system |
| **Preconditions** | 1. User must exist in the ARION database (as Lecturer or current Student) prior to the DB Migration. 2. DB Migration has been completed. |
| **Frequency** | More than 20 times. |
| **Basic Flow** | 1. The use case begins when the user wishes to log into the system. 2. The system requests that the user enter identification details. 3. The User enters their User ID and Password. 4. The system validates the details and presents default page of the system to the User. 5. The use case ends when the User is presented with the welcome page or system default page. |
| **Alternative Flow** | **Invalid ID**   1. If in step 3 of the Basic Flow the user enters an invalid ID or invalid Password. 2. The system displays an error message. 3. The use case ends when the user chooses to navigate to another part of the system or log out (either by selecting Log out or Closing the window). |
| **Post Conditions** | 1. User is currently enrolled at UCOL or is a Lecturer currently employed by UCOL. |
| **Actors** | UCOL Lecturer, UCOL currently enrolled Student |
| **Included Use Cases** | N/A |
| **Excluded Use Cases** | N/A |
| **Notes** | User can log in more than once. |

|  |  |
| --- | --- |
| **Name** | **Log Out** |
| **Identifier** | **UC 02** |
| **Description** | This use case describes how a user Logs Out of the System |
| **Goal** | Log User out of the system |
| **Preconditions** | 1. User must exist in the ARION database (as Lecturer or current Student) prior to the DB Migration. 2. DB Migration has been completed. 3. User is Logged in the system. |
| **Frequency** | If the User is logged into the system they are only able to Log Out once. |
| **Basic Flow** | 1. The use case begins when the user wishes to log out of the system. 2. User selects the X icon of the Window or selects “Log Out”. 3. The use case ends when the User logged out of the system. |
| **Alternative Flow** | N/A |
| **Post Conditions** | 1. User is currently enrolled at UCOL or is a Lecturer currently employed by UCOL.. |
| **Actors** | UCOL Lecturer, UCOL currently enrolled Student |
| **Included Use Cases** | N/A |
| **Excluded Use Cases** | N/A |
| **Notes** | N/A |

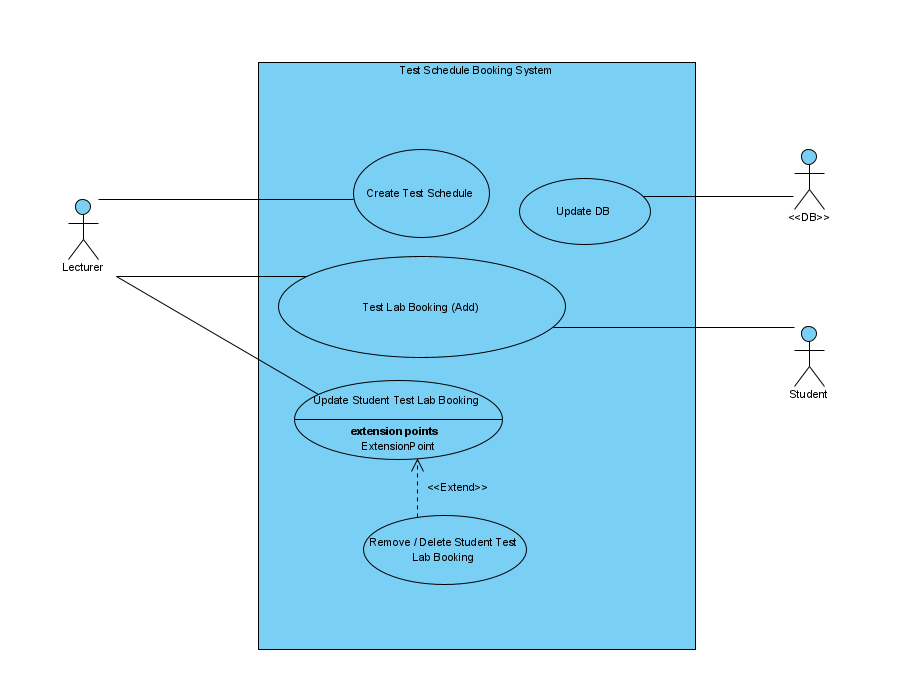
**Class Setup & Student Record (CRUD) Use Cases**



|  |  |
| --- | --- |
| **Name** | **Class Setup** |
| **Identifier** | **UC 03** |
| **Description** | This use case describes how a Lecturer will setup up the Class with Students. |
| **Goal** | Lecturer is able to setup up the Class with students. |
| **Preconditions** | 1. User must be logged into the system. |
| **Frequency** | Lecturer is able to create as many classes with a number of students. |
| **Basic Flow** | 1. The use case begins when the User (Lecturer) wishes to setup the Class. 2. The User selects the class to be setup and creates the class name. 3. The User can add new students to the class. 4. The use case ends when the new student record has updated successfully to the DB. |
| **Alternative Flow** | N/A |
| **Post Conditions** | 1. User is logged in. |
| **Actors** | UCOL Lecturer |
| **Included Use Cases** |  |
| **Excluded Use Cases** | N/A |
| **Notes** | User can log out more than once. |

|  |  |
| --- | --- |
| **Name** | **Create, Remove, Update, Delete Students** |
| **Identifier** | **UC 04** |
| **Description** | This use case describes how a Lecturer will perform the following functions against Students in the Class: -   * Create Student(s) * Remove Student(s) * Update Student(s) * Delete Student(s) |
| **Goal** | More than 20 times. |
| **Preconditions** | 1. User must be logged into the system. |
| **Frequency** | More than 20 times. |
| **Basic Flow** | 1. The use case begins when the User (Lecturer) wishes to make changes to Student Data record in the class. 2. The User is able to select <Create> to add a new Student Record. 3. On selecting <Save> the new Student Record is saved successfully to the database. 4. The User then logs out of the Class area of the system. 5. The use case ends when the User has completed the User |
| **Alternative Flow** | N/A |
| **Post Conditions** | 1. User (Lecturer) is logged in. |
| **Actors** | UCOL Lecturer |
| **Included Use Cases** | UC 03 |
| **Excluded Use Cases** | N/A |
| **Notes** | User (Lecturer) is able to create, remove, update and delete a number of student records from the Class. |

**Test Schedule Create & Booking Test Lab**



|  |  |
| --- | --- |
| **Name** | **Create Test Schedule** |
| **Identifier** | **UC 05** |
| **Description** | This use case describes how a Lecturer will setup up the Class with Students. |
| **Goal** | Lecturer is able to setup up the Class with students. |
| **Preconditions** | 1. User must be logged into the system. |
| **Frequency** | Lecturer is able to create as many classes with a number of students. |
| **Basic Flow** | 1. The use case begins when the User (Lecturer) wishes to setup the Class. 2. The User selects the class to be setup and creates the class name. 3. The User can add new students to the class. 4. The use case ends when the new student record has updated successfully to the DB. |
| **Alternative Flow** | N/A |
| **Post Conditions** | 1. User is logged in. |
| **Actors** | UCOL Lecturer |
| **Included Use Cases** |  |
| **Excluded Use Cases** | N/A |
| **Notes** | User can log out more than once. |

|  |  |
| --- | --- |
| **Name** | **Remove, Update, Delete Student Test Lab Booking** |
| **Identifier** | **UC 06** |
| **Description** | This use case describes how a Lecturer will perform the following functions against Students in the Class: -   * Create Student(s) * Remove Student(s) * Update Student(s) * Delete Student(s) |
| **Goal** | More than 20 times. |
| **Preconditions** | 1. User must be logged into the system. |
| **Frequency** | More than 20 times. |
| **Basic Flow** | 1. The use case begins when the User (Lecturer) wishes to make changes to Student Data record in the class. 2. The User is able to select <Create> to add a new Student Record. 3. On selecting <Save> the new Student Record is saved successfully to the database. 4. The User then logs out of the Class area of the system. 5. The use case ends when the User has completed the User |
| **Alternative Flow** | N/A |
| **Post Conditions** | 1. User (Lecturer) is logged in. |
| **Actors** | UCOL Lecturer |
| **Included Use Cases** | UC 03 |
| **Excluded Use Cases** | N/A |
| **Notes** | User (Lecturer) is able to create, remove, update and delete a number of student records from the Class. |