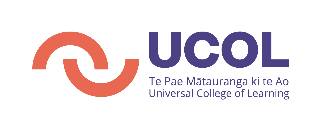
**Assessment**



**D202 Team Development Project**

|  |  |  |
| --- | --- | --- |
| **Due Date**  Tuesday, 2nd November 2021 | **Faculty**  Humanities and Business | |
| **Due Time**  In Class | **School**  Business and ICT | |
| **Weighting**  45% | **Programme**  Bachelor of Information and Communications Technology | |
| **Submission Method**  In class observation and  delivery, Upload Sprint Retrospectives | **Learning Outcomes Assessed**   1. Undertake a team based iterative development project   3. Implement processes to ensure quality | |
| **Conditions**  This assessment is to be completed in teams | **Team:**  Oscar Grey, Tom Lyne, Talitha Mao-Adams, Michael Salisbury, Keaton Winkworth | |
| **Lecturer**  Sandra Cleland | |  |

## Assignment Overview

In a team, you will follow an Agile Development Method to manage a given project.

1. Apply Agile Development Methods (SCRUM) including:

* Functions will be developed in bi-weekly ‘Sprints’.
* Teams will conduct a SCRUM at the beginning of each development session
* Coding will be carried out by ‘Pair Programming’ practices
* Product Owner will be consulted for feedback (at the very least as part of each sprint review)
* Product Owner prioritisation and change requests will be happily accommodated into the following ‘Sprint’
* Team members will take on a variety of roles within the group

1. Develop a prototype application as given in the requirements.

* All major functions have been implemented as prioritised by the product owner.
* Software quality processes have been applied:
* Source control with Git in Visual Studio
* code format & comments,
* analysis / design documentation
* testing.

All development is restricted to class time, research can be conducted outside of class but no coding.

## Project Requirements

Your team must design and develop an **ASP.Net Core** web application to aid the BICT lecturers and students in the BICT labs.

* 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 could 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 / Presentation **Schedule** for a class
* The schedule should be made up of **customisable** time slots for a date (defined by the lecturer at time of creation – ex. time slot E.g. hour, 15 students then?)
* Students in the class should be able to **access** the Test / Presentation 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**

## Deliverables

The deliverables for the assignment are:

* Documentation from the Sprint
  + Sprint documentation will be in the form of the **SCRUM wall** and **burndown charts**
* Sprint **retrospectives** will be conducted at the end of each sprint (**online tool preferred**)
* **Documentation** of quality assurance procedures that the team has been using during the project development
  + **Unit tests** are present within the VS solution
* Design diagrams and documentation, at a minimum this will include:
  + **ERD**
  + **Use Case Model**
  + **Product Backlog User stories**
* **Prototype** application.

Be warned that the Product Owner may throw in additional deliverables at any time – the nature of the Agile Methods is the ability to adapt rapidly to change.

## marking Guide

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Mark** | **Marks Awarded** | **Comments** |
| **Sprint Planning and progression**  Documentation of work allocation and progress for each bi-weekly ‘sprint’. Progression and resource allocation on each task is clearly shown. (ie. Sprint Backlog + Burn-down Chart (updated on the SCRUM wall) | 8 |  |  |
| **Daily Stand Up**  Scrum meetings are held weekly and documented | 4 |  |  |
| **Product Backlog (Scrum Wall)**  Documentation of product backlog and evidence of updates. | 4 |  |  |
| **Product Owner Consultation**  Product owner has been consulted during the sprint planning to negotiate feature priority. Owner requests for changes are implemented during the following ‘sprints’. Complete Features are demonstrated | 6 |  |  |
| **Quality Assurance Processes**  All required quality assurance processes are implemented and documented (source control, self-documenting code, analysis documents, inspection records, unit testing)  Sprint retrospectives are documented the end of each sprint | 10 |  |  |
| **Software Design Documentation**  Documentation of User Requirements (User Stories), Use case model, ERD, Acceptance tests | 8 |  |  |
| **Software Application**  Prioritised user requirements have been completed, where a feature is incomplete that code should be sufficient to demonstrate progress on the task. Marks awarded here based on complexity and amount of functionality completed | 30 |  |  |
| **Student participates in the group project**  Student has participated in all sprints and has made an equal contribution to the group development as reflected in lecturer observation log (10) | 30 |  |  |
| **Total** | | 100 |  |

**Your attendance will be used as a multiplier for the assignment marks.**

**100% attendance makes you eligible for 100% of the earned group marks  
50% attendance makes you eligible for 50% of the earned group marks**

**Programme** – BICT **Course** – D202 Paper **Class** – Lab / Lecture or combined

User Story**:**

Assumption that there will be a core database/table holding all student personal info/details.

1. As a Lecturer Admin, I want the system to support multiple users and two main user types, so that multiple Students can log in with separate Student and Lecturer accounts.
   1. *create home page. (1)*
   2. *create log in page and function to log in based on info entered. (2)*
   3. *make database table to record log in details 🡪 Student 🡪 Lecturer. (5)*
   4. *connect to database – test connection. (½)*
   5. *don’t break when lots of people log in. (3)*
2. As a Lecturer Admin, I want to be able to create and populate classes with enrolled Students, so that only Students from that class can add themselves to the class hours schedule. (pick a smaller time block from the bigger class time block ^)  
   1. *create course page. (1)*
   2. *create database tables of courses. (2)*
   3. *\_inject students enrolled in courses into classes. (3) – calendar type?*
   4. *connect to database. (1)*
   5. *create class schedule page. (5) ½/hour sections from 2-3 hours class* blocks (page with options for days then in days - shows available rooms and hours then choose hours for class then divide into ½/hour sections with max student limit)
   6. *populate class schedule with enrolled students. (8?)*
   7. *test only students in that class can see schedule ? (2)*
3. As a Lecturer Admin, I want to be able to create a Test / Presentation Schedule, so that my Students can view a schedule for classes.  
   1. *create a schedule page. (a. 🡪 b. a. 🡪 c.) (1)*
   2. *create a test page. (1)*
   3. *create a presentation page. (1)*
   4. *only Lecturers able to create schedules. (1)*
   5. *test only Students from that class can view schedule. (2)*
4. As a Lecturer Admin, I want to be able to edit the Test / Presentation Schedule so that I can remove bookings that clash or are incorrect, when needed.  
   1. *open presentation schedule – with Lecturer log in? (2)*
   2. *only Lecturer can edit presentation schedule. (2)*
   3. *only Lecturer can remove bookings from presentation schedule. (2)*
   4. *only Lecturer can save presentation schedule. (2)*
5. As a Student, I want to be able to book a specific available time for my class, so that I can view my time slots for class.  
   1. *log in as Student. (1)*
   2. *open presentation schedule. (2)*
   3. *view available time slots. (2)*
   4. *select from available time slots. (2)*
   5. *save selection. (2)*
6. As a Lecturer Admin, I want to be able to reset Student Lab passwords so that Students can access their accounts in case of forgotten passwords.  
   1. *create profile page for people. (1)*
   2. *access core database to populate page. (3)*
   3. *log in as Lecturer. (1)*
   4. *view Student details. (3) – database find from id then profile*
   5. *edit/reset Student password only. (2)*