	Digital Technologies Manukau Campus BDT - Bachelor of Digital Technologies 562.615 "Cloud Computing for Software Developers" PROJECT Q3 2018
Assignment:	"The students are expected to design and implement a SaaS solution"
Due Date:	**Midnight, Mon, 10 Sep 2018**
Assess. Weighting	40 marks contributing 40% towards course 562.615 total.

Statement of Original Authorship

I hereby confirm that this assignment is my own work, or in the case of team work it is the team work done for this assignment. In addition, the assignment has not previously been submitted for assessment, either in whole or in part, by either myself or any other student at either Manukau Institute of Technology or at any other tertiary institution. To the best of my knowledge and belief, the assignment contains no material which has been previously published or written by another person **except where due reference has been made**. All unpublished sources of information have been acknowledged. I make this statement in full knowledge and understanding that, should it be found false, I will, in most circumstances, receive zero marks for this assignment.

Note that agreement with this statement is a condition of handing in electronically and therefore that act of hand-in becomes your act of agreement.

For hand-ins on paper:

Signed by student: _____ Date: _____

Student Name: _____ ID: _____

This Project relates mainly to Learning Outcomes 2, 3, 4

1. Demonstrate an understanding of the fundamentals of cloud computing and different cloud delivery models (SaaS, PaaS, IaaS)
2. **Design software using Service-Oriented Architecture**
3. **Demonstrate an understanding of security best practices for SaaS**
4. **Implement and publish Software-as-a-Service**

The Task is to create:

A SaaS Solution that meets scenario needs and demonstrates an understanding of security best practices for SaaS.
Students/teams can select from these technical scenarios. In addition students need to create a business or game scenario that is different to others running in the class e.g. shoe shop, industrial robot provider.

- Prototype a proof of concept multiplayer online game with use of the Unity Platform

- Prototype an employee timesheet SaaS. Clients can be websites or phone applications or both. The system will have at least 2 services and their clients:

  TimesheetService that allows employees to enter weekly timesheets of hours worked breaking down to work sessions on different tasks. This has some similarities to the course work diary. Employees will need to login first. Admins should be able to manage users.

  AccountingService that will retrieve all new timesheets from TimesheetService. Accountants will be able to login to this service and generate payslips based on the timesheets. AccountingService needs to securely authenticate its service-to-service conversations with TimesheetService with OAuth2 preferred for this purpose.

- Prototype an online business to business ordering SaaS. Clients can be websites or phone applications or both. The system will have at least 3 services and their clients:
  - OrderService – Responsible for handling logins, and creating and processing user requests (Orders) for products and/or services.
  - InventoryService – Responsible for handling maintenance and supply of the products and/or services data and returning queries about the products and/or services including stock levels (how many available) for each product.
  - AccountingService – Downloads data from OrderService and InventoryService for the production and issuing of Invoices. Handles the billing for the case of a business ordering more of an item than we have in stock.

  The services need to securely authenticate their server-to-server conversations with each other. OAuth2 is preferred for this purpose.

Select from the scenarios listed above or students can suggest and negotiate other scenarios with the lecturer based on areas they are interested in.

You are able to work in teams of 2 or 3 members if first approved by your lecturer.
For teams there will be common requirements for a Team Mark. Teams will need to agree with the lecturer on individual requirements for member Individual Marks.

Teams of 2 or 3 need to select 1 member to control "merging of branches into the master" - the main task is checking that new code (a "branch") is working before it gets copied into the combined project (the "master branch").

Common Requirements:

1. Your programming editor and environment is
   Microsoft Visual Studio Team Services Option GIT with the lecturer added to your team.

2. Hand-ins are by file uploads to the Canvas team area.
   This includes individual projects which will operate as a Canvas team of one.
   You also need to publish the website into our "hybrid cloud" environment.
   as a Microsoft Azure Web App connecting to a database hosted on manukautech.info

3. Use the APA referencing style for all referencing in your bibliographies.

4. Use technologies specified by the lecturer or negotiated with the lecturer.
   For Q3 2018 the specified standard is: Microsoft ASP.NET Core 2.1.
   The database standard is: Microsoft SQL Server 2016
   as provided at manukautech.info or xmsql04.australiaeast.cloudapp.azure.com

5. Clearly state in your user interfaces that this is a student project.

6. Record help given to or received from others in your Work diary.

Submit the following items. More detailed descriptions are in the marksheet below:

1. [Team] Planning documents.
2. [Team] Requirements Checklist document - template supplied
3. [Team] Bibliography with Bibliography Added Notes
4. [Team] SaaS Application
5. [Individual] Work Diary Timesheet using supplied Web App. No separate hand-in needed.
6. [Individual] Metrics Report - "MS Word" document.

Professional Practice regarding media copyright.
We expect this is not a big issue for most SaaS Business applications but it could be significant for scenarios like online multiple player games.
Any music, sounds, images and video needs to be material that you would be allowed to use in a business project which **could be accessible to the public**.
Where you copy or adapt you must include your sources in your Bibliography

Actors need to sign permission forms

Work needs to be acceptable for all viewing audiences.

Original Authorship
The cover sheet which you sign includes this wording:
"To the best of my knowledge and belief, the assignment contains no material which has been previously published or written by another person except where due reference has been made."

This means, you are not allowed to copy from anyone or anything unless you make "due reference" which means write it into your Bibliography.
This becomes important when you ask for help from other students or give help to other students.

When you are helping your friends, do this by talking to them to guide them as they (not you) do the work.
Never, NEVER, give files to your friends. And if you are having problems then do NOT say to your friends "please give me your files so I can learn from them" because that puts unfair pressure on your friends and you can get them into big trouble along with yourself.

Complexity/High Level work
Many students misunderstand the idea of getting marks for “complexity”, and think they should try advanced elements eg forcing unnecessary programming into their project to try to impress the lecturer. This often ends badly. Your first priority is to meet scenario needs. Being agile means knowing when to drop complex efforts if effective use of an existing resource can do the job better. Eg I have given advice in one case to stop trying to program a calendar and instead do a SaaS connection to the API of the client’s existing Gmail calendar.

Marksheet 562.615 ("Cloud4Devs")

Item	Mark	Out-of
[TEAM] Bibliography The Bibliography includes at least 3 items per student in APA format. To get a good grade of 1.5 or 2 will require more than 3 items per student from a range of sources demonstrating a wide-ranging investigation interest.		(2)
[TEAM] Bibliography added notes Write a commentary on which of your bibliography sources of information you recommend as the most useful, and why.		(2)
[TEAM] Complete Requirements Checklist Documentation of which requirements you have met, which requirements you have not met, and how and where the marker can find these elements in your work. This document needs to be effective in supporting marking.		(4)
[TEAM] Planning documents. Use Case and/or Flow Diagrams.		(4)
[TEAM] Effective communication of the scenario e.g. by Products, Forms, Controls, Visible dynamic functionality, Titles, Advertising copy text, use of images and other media.		(2)
[TEAM] Demonstrate Publication on the Windows Azure platform		(3)
Team Sub-Total		[17]
[INDIVIDUAL] Work Diary		
Work diary demonstrates good quantity and quality of time recording with appropriate use of categories.		(2)
Metrics - Create this as a separate "MS Word" document hand-in using data from your Work Diary - Timesheet. Breakdown of hours per category from your Timesheet. The Timesheet app has a built in SQL query which can provide this for you. Add comments on trends: especially which categories needed the most time and why.		(2)
Work diary notes effectively record your development process including: • notes on your challenges and successes as they happen • notes on help you have given to others and help you have received from others		(3)
[INDIVIDUAL] Web App Production		
Complexity, scope, completeness and effectiveness of programming elements, including code comments, security best practice and testing, in meeting the Team and Individual Requirements above.		(16)
Individual Sub-Total		[23]
PROJECT TOTAL		[40]

END of 562.615 ("Cloud4Devs")