**6CLP Assignment – Report Template**

|  |  |
| --- | --- |
| Student ID | Name |
| 000080409 | Jack Church |

**Gather and Document Requirements**

|  |  |
| --- | --- |
| **Considerations** | **Response** |
| **1.1** Cloud solution vs private ISP hosting – Which Option and Why | ISPs are in the business of providing access to the Internet first and foremost. A dedicated cloud host is in the business of providing a cloud host, although some ISPs provide cloud hosting from 3rd parties, you would be unable to change ISP for Internet if you went with their cloud. |
| **1.2** If it is a cloud service, which one – AWS, Google App Engine or Azure? (justify your choice) | IT Works has significant experience with Azure when working with .NET technologies over other cloud providers.  The integration between .NET and Azure is tight as both are made by the same company. Whereas AWS must look for new .NET technologies and implement them as soon as possible after the .NET release.  AWS is more expensive than Azure. TAFE SA, the client, is on a limited budget.  Google app engine is the next closest competitor to AWS and Azure in terms of size, but a significant way behind. The TAFE SA data needs to remain in Australia for legal purposes.  Google, for the present, only has 3 data regions in Australia, all located in the general Sydney area. AWS and Azure each have 3 data regions in Australia in 3 different cities. Azure is the only one that is expanding the location of data regions aggressively. |
| **1.3** Implementation of Core Services- choices are ASP.NET Web Service, MVC, WCF service or REST? (Justify your choice) | The decision has been made to continue to use .NET technology with WCF being the specific technology to use. The current app is written in C# with .NET and will have easier integrations with WCF. |
| **1.4** Database deployment options- current legacy system or deployed as a cloud-based database (Think scalability and Compatibility) | Having a cloud service connect to a database not in the cloud removes 99% of the point of having a cloud service and will increase cost with on-prem maintenance for the database, backups of the database and the secure connection between the database and the cloud hosted app.  Having the database local will require a physical server to host the database. If the database needs to expand, ICT will be required to purchase another server to load balance the database. Software will then need to retroactively update the database and related software to load balance the old database server with the new database server.  Moving the database to the cloud will allow for Software team to dynamically adjust the compute power as required. |

**Use Case Model (Add your model here)**



**Use Case Priority List**

|  |  |  |
| --- | --- | --- |
| **Use Case** | **Priority** | **Justification** |
| **Enrol student in course** | **1** | **The system’s main purpose is to enrol students. TAFE SA would like to re-enrol existing students to make money.** |
| **Add student** | **2** | **TAFE SA would like to add new students so they can make money.** |
| **View student timetable** | **3** | **Students need their timetable to know when and where to be.** |
| **Add course** | **4** | **After items relating to enrolling new students, TAFE SA would like to add new course to entice students.** |
| **View student info** | **5** | **Add this use case as TAFE SA business is training students. If TAFE SA’s customers (students) are happy, TAFE SA management is happy.** |
| **Display student bill** | **6** | **After new and existing students have enrolled, but before course information. Course information is nice to have, but not key to TAFE SA’s ability to make money.** |
| **View enrolments for course** | **7** | **Low on the priority as it is not crucial to make money.** |
| **View course info** | **8** | **Low on the priority as it doesn’t make money.** |

**Glossary (Define Application Terms here)**

|  |  |
| --- | --- |
| **Term/Phrase** | **Definition** |
| **Faculty** | **Someone who works for TAFE SA.** |
| **Student** | **Someone who goes to TAFE SA to study.** |
| **Lecturer** | **Faculty who teaches at TAFE SA.** |
| **Course** | **A time and place where students and a lecturer meet. The Lecturer teaches the students.** |
| **Enrolment** | **Adding a student into a course.** |

**Implementation Mechanisms**

|  |  |
| --- | --- |
| **Mechanism** | **Description** |
| Authentication | **Not yet implemented**. Preferred implementation via Azure Active Directory. **Included in this list to highlight the security issue.** |
| Persistency | WCF Entity Framework will control access to data in the database and maintain the session. |
| Security | Secure socket connections across the network to the WCF service. Provided by WCF. |
| Data transfer | WCF provides built in XML sterilization. It can be converted to JSON if required. |
| Legacy communication | XML and .NET provide robust support for data transfer to legacy systems. |
| Data storage | Legacy database managed by Entity Manager. |

**Development environment and tools required**

Processor: modern processor, within 5 years, Intel Core i5 or equivalent.

RAM: minimum 8 GB

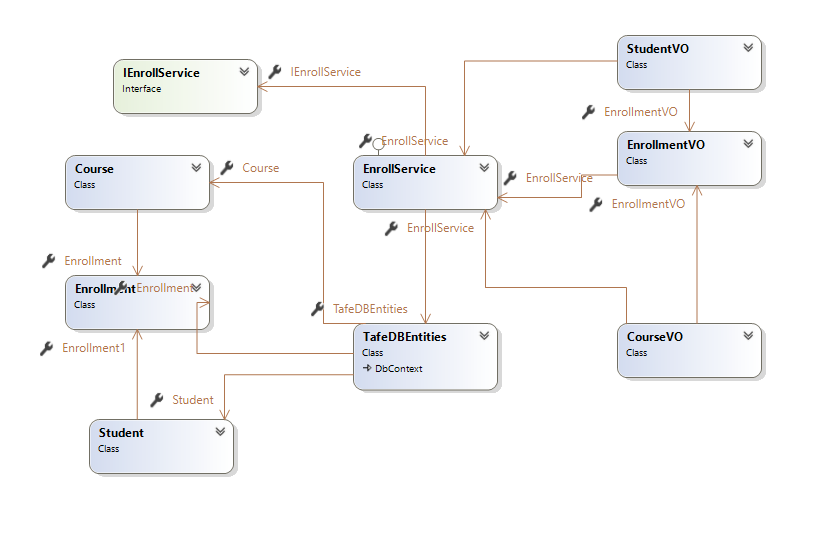
Storage: minimum 256 GB SSD

OS: Windows 10

Software: Visual Studio 2019, NUnit.

Services: Azure for hosting.

**UML Model - Design Classes and relationship**



**A UML component model**

******

**Test Plan**

Due to (i) the scope and budget of the project, the best test plan will be a regression test after each use case is finalized; and (ii) the nature of a service application where issues can be difficult to trace (not forgetting scope and budget), regression testing will be the best method of testing.

The goal of testing is to

1. Have a known database state before testing by running select commands on database.
2. Run the program to view data in each use case where data is viewed to verify connection to service and database.
3. Run the program and add data into each use case for adding data.
4. Run the program to view data in each use case where data is viewed to verify data was entered into the system.
5. Double check data was entered correctly by running select commands directly in the database.

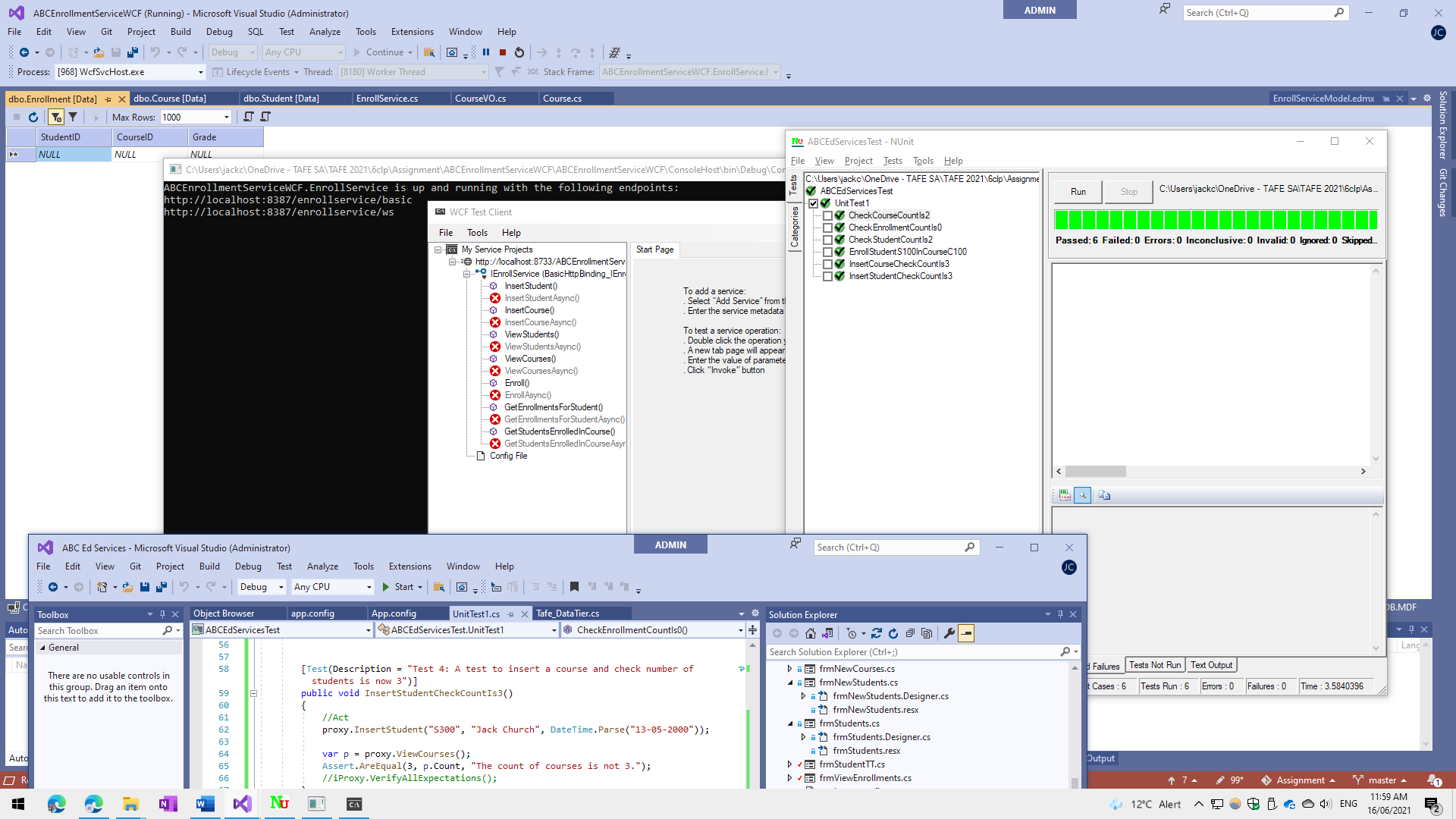
*Testing continued next page.*

**Testing Service Before Publishing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Service** | **Input** | **Expected Output** | **Actual Output** | **Comments** |
| View course table data | SELECT \* FROM dbo.COURSE | C100 | C100 | Pass. Only noting primary key in test. |
| View student table data | SELECT \* FROM dbo.STUDENT | S100  S200 | S100  S200 | Pass. Only noting primary key in test. |
| View enrollment table data | SELECT \* FROM dbo.ENROLLEMNT | No data | No Data | Pass |
| View Student | NA | S100 : Dale Van Heer  S200 : Odette Jansen | S100 : Dale Van Heer  S200 : Odette Jansen | Pass |
| View Courses | NA | 600.00 : Java | 600.00 : Java | Pass |
| View Enrollment | C100 | No enrollments | No enrollments | Pass |
| New Student | ID: S300  Name: Sandy  Date Enrolled: Tuesday, 15 June | Data saved | Student information saved | Pass. |
| New course | ID: C300  Name: Daffodil farmer  Cost: 50 | Data saved | New course information saved | Pass |
| Enroll Student | Student: S300  Course: C100 | Success and saved | New enrollment info saved | Pass |
| Enroll Student | Student: S300  Course: C300 | Success and saved | New enrollment info saved | Pass |
| Enroll Student | Student: S100  Course: C300 | Success and saved | New enrollment info saved | Pass |
| View Student | NA | S100 : Dale Van Heer  S200 : Odette Jansen  S300: Sandy | S100 : Dale Van Heer  S200 : Odette Jansen  S300: Sandy | Pass |
| View Courses | NA | 600.00 : Java  50.00 Daffodil farmer | 600.00 : Java  50.00 Daffodil farmer | Pass |
| View Enrollment | C100 | S300 Sandy | S300 : Sandy | Pass |
| View Enrollment | C300 | S300 Sandy  S100 : Dale Van Heer | S100 : Dale Van Heer  S300 : Sandy | Pass. Order of result was reversed but non-issue. |
| Student time table | S100 | Daffodil farmer | Daffodil farmer | Pass |
| Student time table | S200 | NA | No enrollments | Pass |
| Student time table | C100 | Sandy | S300 : Sandy | Pass |
| View enrollment for course | C300 | Dale Van Heer  Sandy | S100 : Dale Van Heer  S300 : Sandy | Pass |
| Display bill | S100 | $50  Daffodil farmer | $50  Daffodil farmer : 50.00 | Pass |
| Display bill | S200 | $0.0 | $0.00  No enrollments | Pass |
| Display bill | S300 | $650  Java  Daffodil farmer | $50  Java : 600.00  Daffodil farmer : 50.00 | Pass |
| View course table data | SELECT \* FROM dbo.COURSE | C100  C300 | C100  C300 | Pass. Only noting primary key in test. |
| View student table data | SELECT \* FROM dbo.STUDENT | S100  S200  S300 | S100  S200  S300 | Pass. Only noting primary key in test. |
| View enrollment table data | SELECT \* FROM dbo.ENROLLEMNT | S300 C100  S300 C300  S100 C300 | S100 C300  S300 C100  S300 C300 | Pass. Only noting primary key in test. |

**Testing Service After Publishing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Service** | **Input** | **Expected Output** | **Actual Output** | **Comments** |
| View course table data | SELECT \* FROM dbo.COURSE | C100 | C100 | Pass. Only noting primary key in test. |
| View student table data | SELECT \* FROM dbo.STUDENT | S100  S200 | S100  S200 | Pass. Only noting primary key in test. |
| View enrollment table data | SELECT \* FROM dbo.ENROLLEMNT | No data | No Data | Pass |
| View Student | NA | S100 : Dale Van Heer  S200 : Odette Jansen | S100 : Dale Van Heer  S200 : Odette Jansen | Pass |
| View Courses | NA | 600.00 : Java | 600.00 : Java | Pass |
| View Enrollment | C100 | No enrollments | No enrollments | Pass |
| New Student | ID: S300  Name: Sandy  Date Enrolled: Tuesday, 15 June | Data saved | Student information saved | Pass. |
| New course | ID: C300  Name: Daffodil farmer  Cost: 50 | Data saved | New course information saved | Pass |
| Enroll Student | Student: S300  Course: C100 | Success and saved | New enrollment info saved | Pass |
| Enroll Student | Student: S300  Course: C300 | Success and saved | New enrollment info saved | Pass |
| Enroll Student | Student: S100  Course: C300 | Success and saved | New enrollment info saved | Pass |
| View Student | NA | S100 : Dale Van Heer  S200 : Odette Jansen  S300: Sandy | S100 : Dale Van Heer  S200 : Odette Jansen  S300: Sandy | Pass |
| View Courses | NA | 600.00 : Java  50.00 Daffodil farmer | 600.00 : Java  50.00 Daffodil farmer | Pass |
| View Enrollment | C100 | S300 Sandy | S300 : Sandy | Pass |
| View Enrollment | C300 | S300 Sandy  S100 : Dale Van Heer | S100 : Dale Van Heer  S300 : Sandy | Pass. Order of result was reversed but non-issue. |
| Student time table | S100 | Daffodil farmer | Daffodil farmer | Pass |
| Student time table | S200 | NA | No enrollments | Pass |
| Student time table | C100 | Sandy | S300 : Sandy | Pass |
| View enrollment for course | C300 | Dale Van Heer  Sandy | S100 : Dale Van Heer  S300 : Sandy | Pass |
| Display bill | S100 | $50  Daffodil farmer | $50  Daffodil farmer : 50.00 | Pass |
| Display bill | S200 | $0.0 | $0.00  No enrollments | Pass |
| Display bill | S300 | $650  Java  Daffodil farmer | $50  Java : 600.00  Daffodil farmer : 50.00 | Pass |
| View course table data | SELECT \* FROM dbo.COURSE | C100  C300 | C100  C300 | Pass. Only noting primary key in test. |
| View student table data | SELECT \* FROM dbo.STUDENT | S100  S200  S300 | S100  S200  S300 | Pass. Only noting primary key in test. |
| View enrollment table data | SELECT \* FROM dbo.ENROLLEMNT | S300 C100  S300 C300  S100 C300 | S100 C300  S300 C100  S300 C300 | Pass. Only noting primary key in test. |

NUnit Tests:

**Steps - Deploying /Publishing your services**

*JC note: Due to issues with learning costs running out in Azure, I have created a Console host to host the service. However, this document will list the steps to deploy to Azure.*

* Create resource in Azure: jc-6clp-resource.
* Create server in previously created resource: jc-6clp-server,
  + Username jchurch
  + Password Password01
* Create SQL resource in Azure within server just created.
* Set SQL firewall to allow incoming connections from green lit IPv4 Addresses.
* Get the server connection string: jc-6clp-server.database.windows.net
* Setup the SQL database tables as per the requirements.
* Modify the WCF app.config to use the SQL database :jc-6clp-server.database.windows.net
* Clean, Build and then run the Publish wizard to publish the WCF service to Azure: with a unique name: ABCEnrollmentServiceWCF210615
* Click Publish to finalise and make live.

|  |  |
| --- | --- |
| **URL** | https://abcenrollmentservicewcf210615.azurewebsites.net/ |

**Further Research**

**-Big Data Considerations (500 words)**

Big data is often associated with scalable databases, which is why I will concentrate on databases in this section. I will also keep in mind that the client is on a budget, thus Oracle database products are not an option.

The connection between the service and database is the weakest part of any application or service. The current service and database will be sufficient for less than 30 simultaneous connections. Modifying the application to support database connection pools, recreating the database with a non-relational database product, and a more robust service architecture will allow for expansion to big data needs.

A big data database is one that is not used for traditional relational database uses. Its goal is to have unstructured tables that look up data stored in a data pool, collect all the data types being passed to it from the data pool and store it as close to raw data as possible without any metadata associated with each entry. The enrollment system currently uses a SQL database. A different kind of database that stores data in columns, documents or key-values will be required, such as NoSQL, CosmosDB, Hardoop, MongoDB to name a few.

Creating a big data database and storage pool for TAFE SA can be something that IT Works will take on for a purpose other than the enrollment system. The enrollment system just needs connection pooling and extra servers to load balance to serve the needs of the foreseeable future.

The big data that TAFE SA could use is with:

1. Students submitting works to the system.
2. Gathering information from other learning services to build an independent plagiarism service.
3. Gain insights and see the trends to somewhat predicate the future of staff and course by looking at the performance of how a specific student is performing in association with a group of students they are with while being taught by certain lectures teaching specified subjects in a specific location.
4. Students and lecturers routinely connect USB devices to TAFE SA computers, this has posed a security threat for several years and will only pose as a greater threat in the future.
   1. Big data stored could be used for machine learning to detect threats and automatically isolate threats in a few seconds.
   2. This will be quicker than having anti-malware software find and alert ICT Security and then needing ICT Security to find the affected devices and intervene, by which time the spread could be exponential.

Azure and AWS are both well suited to handling a big data database with storage pool of data. The cost associated may not be stomached by TAFE SA management as it requires having large data storage, a large database to live process the data storage and fast connections to both the storage and database from clients.

The product that may put it over the top for TAFE SA is the security angle making computing safer at TAFE SA while allowing staff and students to use BYOD USB devices.

**-Xml Serialization vs. JSON Serialization**

XML data formats have no structure and requires the receiving program to parse the file before it can get any meaning full data. JSON files are created with datatypes defined and require minimal processing power to de-serialize into the final data form.

WCF services use XML by default and have better integration (and hopefully efficiency) than the JSON format.

As JSON is a native JavaScript format, if TAFE SA wish to create the Enrollment service as a web app, it would be best to use JSON for performance reasons alone. The XML parser in web clients may not be the most current and thus not the most compatible for XML serialization.

**-ASP.NET MVC vs. WCF Services**

ASP.NET MVC was used to created web applications where access can only be though a website due to the inherent nature of an integrated MVC application.

WCF Services allow for the splitting of the backend service from the front-end client. It will allow for the front-end client to be a website, web app, Windows app, Mac app, iPhone app, etc. It is worth noting that the ASP.NET MVC project could connect to the WCF Service in Azure.

TAFE SA have requested that we reengineer the legacy Windows app as a new cloud based remote service and ensure that the legacy Windows app can connect to the new cloud based remote service.

For the reason of ensuring the legacy Windows app can connect to the new cloud based remote service, it is only feasible to use the WCF Service.

**-Windows Forms vs. ASP.NET Html Views**

Windows Forms provide a legacy framework to quickly create WYSIWYG screens for application that only run-on Windows. Any MVC pattern will need to be manually introduced.

ASP.NET Html Views are Microsoft’s way to easily create a single-screen-web-app following the standard MVC pattern.

ASP apps are easier to create and update with business changes, easier to deploy as there is no software to install, more secure as security is built into the framework. It is recommended that TAFE SA move to a web app even if it is not an ASP app.