**Software Architecture Document**

**TafeSA Online Enrolment System**

**Revision History**

|  |  |  |
| --- | --- | --- |
| **Date** | **Version** | **Author** |
| **14/11/2023** | **1.0** | **Andre Alexandrov** |
|  |  |  |
|  |  |  |

**Section A – Stakeholder Identification/Communication Plan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Stakeholder** | **Description** | **Communication Strategies (Digital vs Non Digital)** | **Comments** |
| System Administrator | **Tafe SA, employee** | **Digital and non digital** | **Meetings (face to face and virtual), documentation, emails, consultation.** |
| **Students** | **Customer of TafeSA, end user** | **Digital** | **No direct communication would take place, email, PSA, FYI, Online portal etc** |
| **Lecturers** | **Employee of TafeSA, end user** | **Digital** | **No direct communication would take place, email, PSA, FYI, Online portal etc** |
| **Registrar** | **Employees of TafeSA who register students** | **Non Digital** | **Meeting, consultation etc** |

**Section B - Determine/Design the Business Model and Architecture**

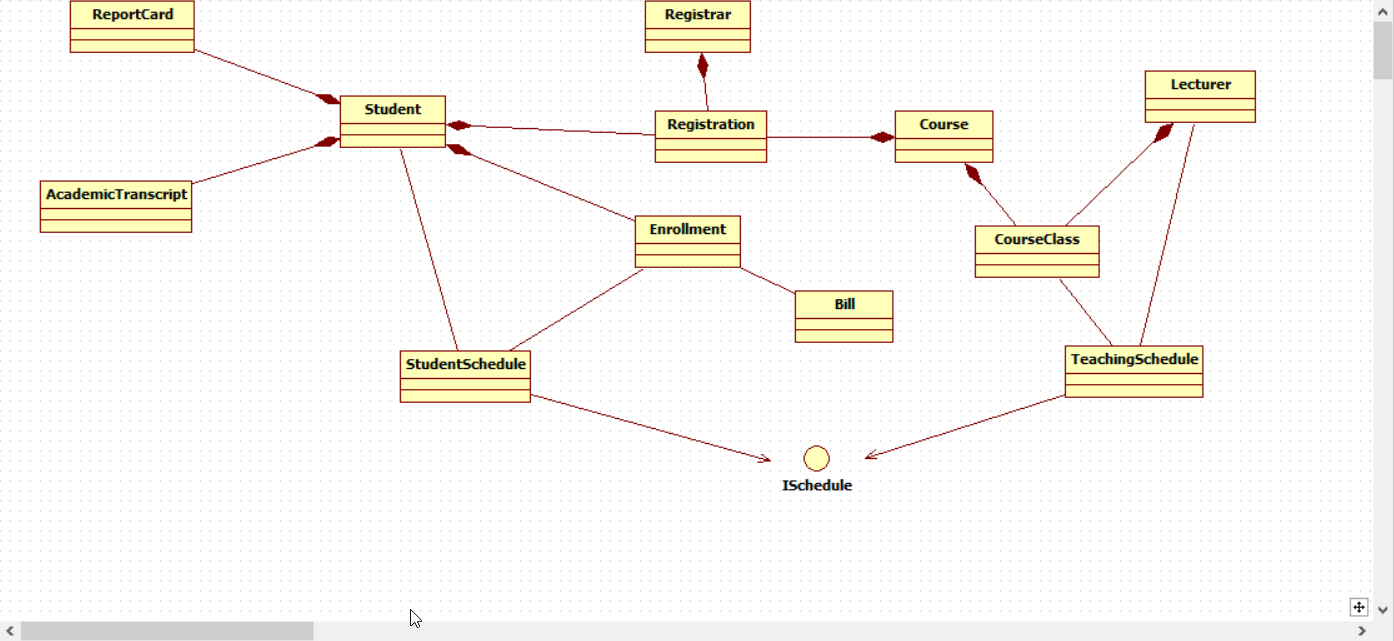
**The lead architect has called you into a meeting to discuss and determine the client’s business model from a list of potential alternatives before any concrete decision could be made on preliminary Architecture of the system.  The meeting will comprise of members of the team and the project as the major stakeholder.**

**During this meeting you will required to discuss the items below and document the outcomes as guided**

**Students, courses, enrollment, Lecturer, Registrars, LecturerCourse, *Ischedule* – teacher and student, rego**

**Given the requirements specification and other ancillary documents you are required to:**

1. **document and describe the system Business Model and its impact on the choice of Architecture.**

****

The proposed class diagram outlined above illustrates the incorporation of multiple models/entities that the system will encapsulate into objects. Given that the system is designed as a web application, incorporating numerous views and employing various controllers to segregate coding logic across different views, the adoption of the Model-View-Controller (MVC) pattern emerges as the most suitable choice for the system architecture.

Object-Oriented design/programming principles influenced the system architecture, as it integrates multiple models and entities encapsulated into objects. The use of a web application framework necessitates numerous views and controllers, aligning with OOD's encapsulation, inheritance, and polymorphism, MVC patterns follow these principles by using modular separation of data, the presentation of the system and the logic.

**Use Case: Register for Courses**

###### **Section C - Create the user experience model for above Use Cases**

*Copy your User Experience UML Models Here*

|  |
| --- |
|  |
|  |
| `A diagram of a project  Description automatically generated |
| Andre Alexandrov |

|  |
| --- |
|  |
|  |
| Simon Johnson |

|  |
| --- |
|  |
|  |
| Jesse Hamilton-Young |

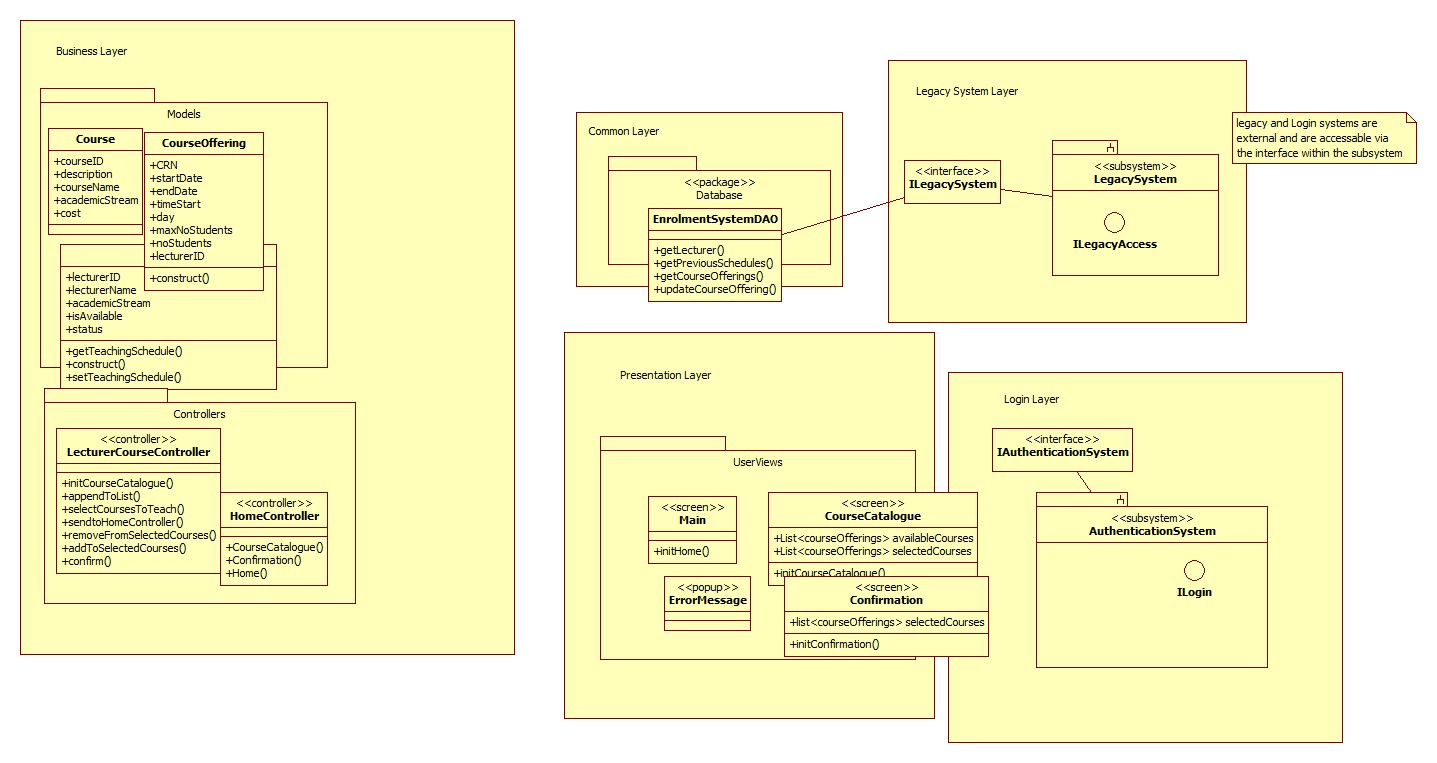
##### **Section D – Design and Implementation Mechanisms (Done by Individual)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Analysis Mechanisms** | **Design Mechanisms** | **Implementation Mechanism** | **Justification for choice and influence of Organizational procedures and standards** |
| Persistence | Object-Relational Mapping | **.net entity framework (EF)** | EF is a powerful ORM framework that enables developers to work with databases using object-oriented principles. It allows you to represent database entities as .NET objects, providing a natural and intuitive way to interact with the database. |
| External System interaction | **Facade Pattern** | **RESTful APIs** | RESTful APIs are widely used for their simplicity, scalability, and compatibility with various platforms. .NET provides a straightforward way to make HTTP requests, making it suitable for interacting with RESTful services. |
| Legacy Systems interaction | **Adapter Pattern** | **Integration Adapters** |  |
| Cultural Specific and Worldwide considerations (Onshore and Offshore partner campus in Vietnam) and Distribution | **Collaboration Design Pattern** | collaborative tools and technologies |  |
| System Parameter Management | Parameter stored externally | Storing location reference in database | Storing parameters externally allows for centralized management of configuration settings. This makes it easier to maintain and update parameters across multiple environments or instances of a system |
| Authentication | Password Authentication | Implement the hashing of passwords and user names and implementing multi factor authentication |  |
| Authorisation | Role Based Access Control (RBAC) | Active directory (AD) | RBAC and AD allow for segmentation of authorisation in a system providing greater security |
| Transaction Management | Transaction management service layers | .Net system.transactions | Ensure data remails consistent and reliable, allows for rollback features and isolation to prevent corruption, following the ACID (atomicity, consistency, isolation, durability) principle |
| Error Handling | Exception handling | Using C# exception objects |  |
| Concurrency | Asynchronous programming | C# Async await | Asynchronous programming enables non-blocking execution of code. This is particularly beneficial for operations that may take time to complete. By not blocking the main thread, the application remains responsive, providing a better user experience |
| User Interface/User Experience | User centred Design | Creating user centred Ui using C# MVC | User centred Design means developing a design that works best for the end user |
| Security | User input validation | Using RSA encryption for data at rest and TLS when in transit  Use Parameterised Queries |  |

##### **Section E– Identify Design Elements and interconnecting Components**

*Copy your Design/Component UML Models Here*

|  |  |
| --- | --- |
|  |  |
|  | **Andre Alexandrov** |

****

**Jesse Hamilton-Young**

|  |  |
| --- | --- |
|  |  |
|  | **Simon** |

##### **Section F –Model the use case realization (Done by Individual)**

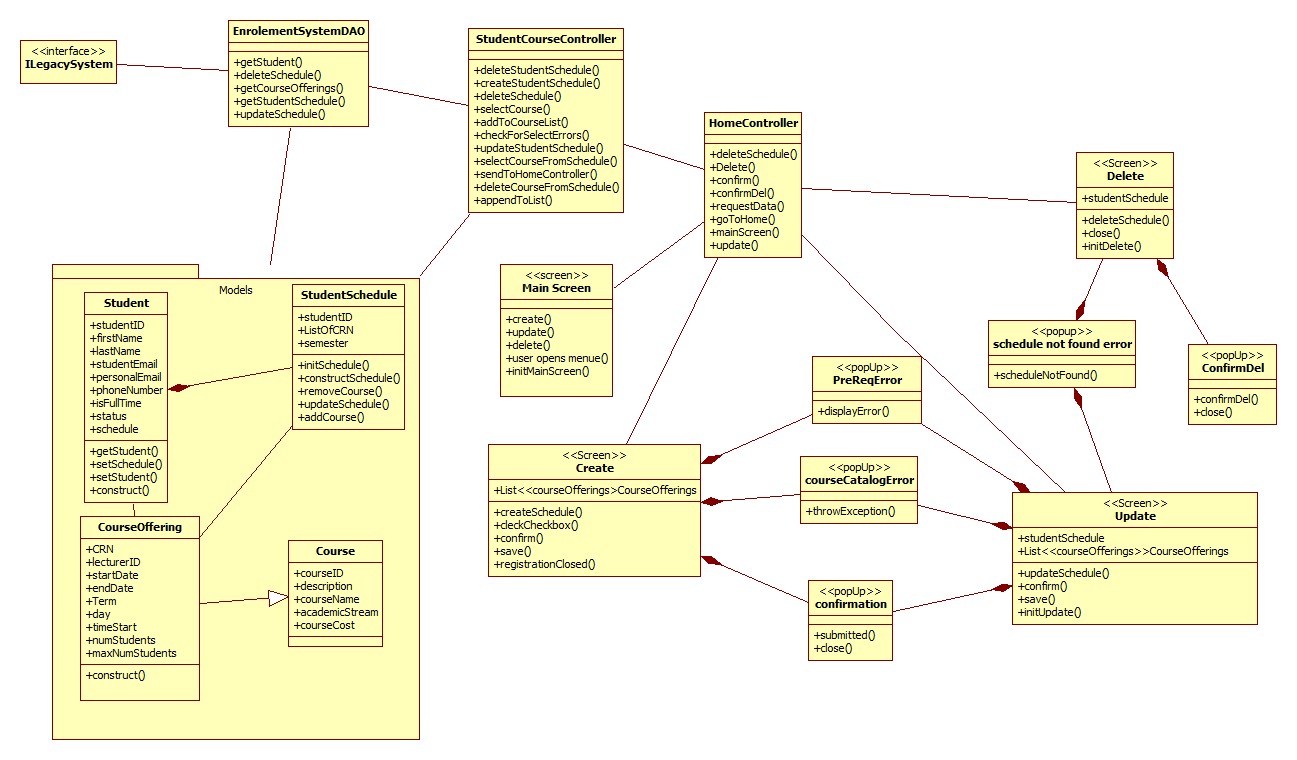
*Copy your Use Case Realizations UML Models Here*

|  |  |
| --- | --- |
|  |  |
|  | Andre Alexandrov |

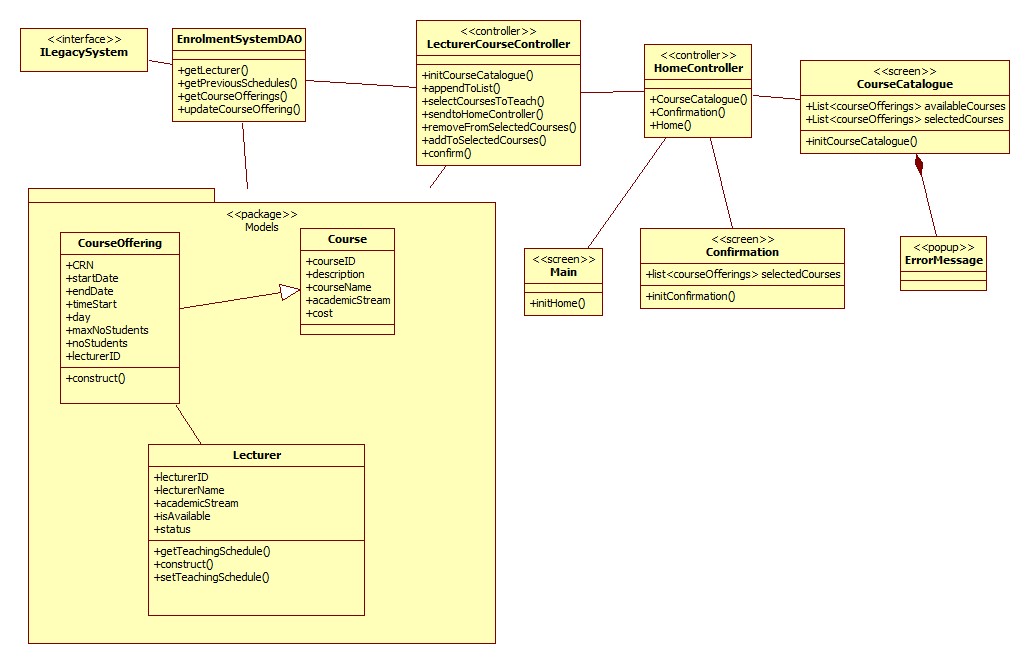
|  |  |
| --- | --- |
| **A diagram of a project  Description automatically generated with medium confidence** |  |
| **A screenshot of a computer program  Description automatically generated** | **Jesse Hamilton-Young** |

|  |
| --- |
|  |
|  |

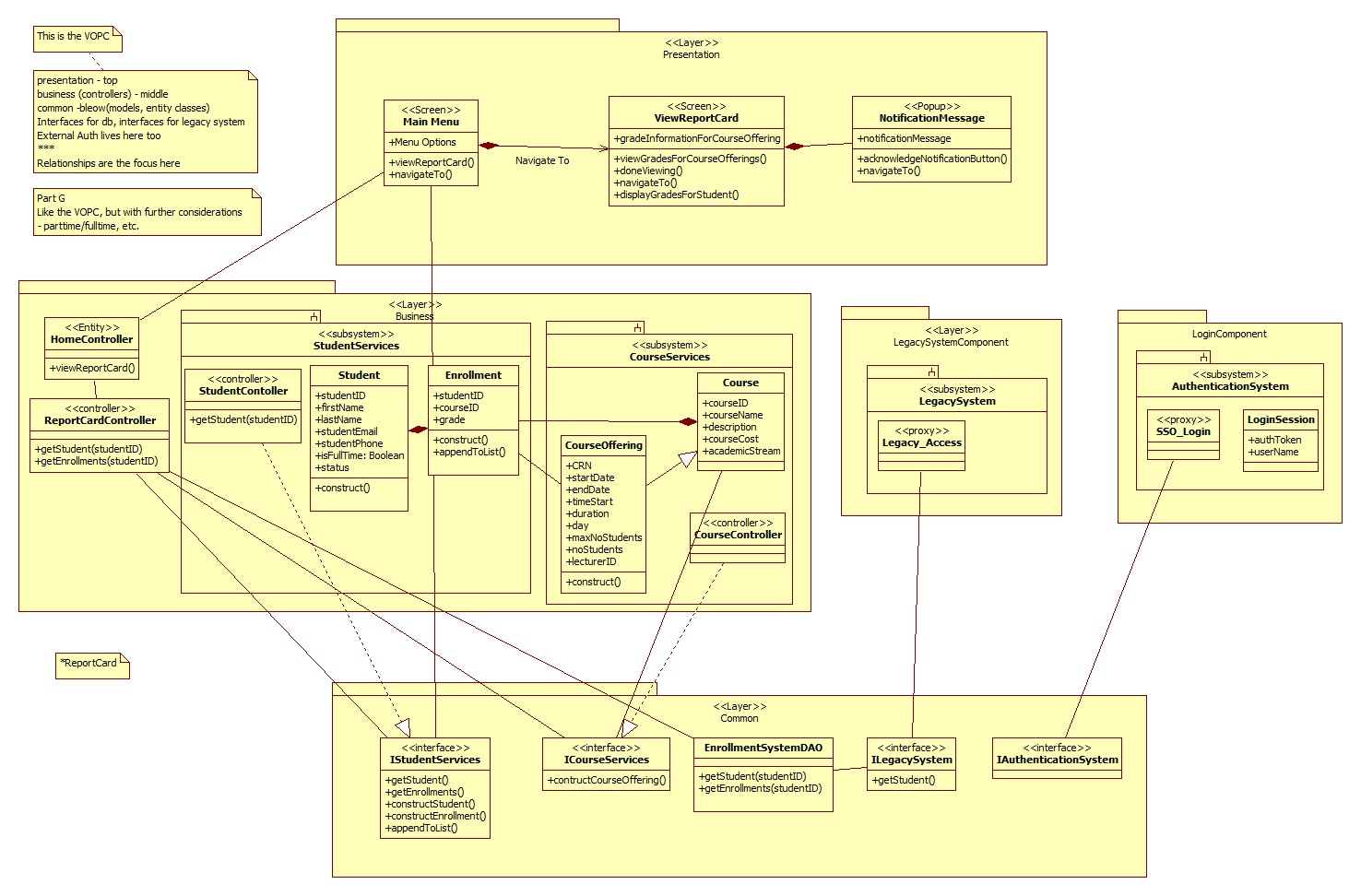
##### **Section G – Class Design (Done by Individual)**

*Copy your Class UML Models Here  
*

*Andre Alexandrov*

**

*Jesse Hamilton-Young*

**

*Simon*

##### **Section H – Database Design**

*Copy your Database Models Here*

|  |
| --- |
|  |

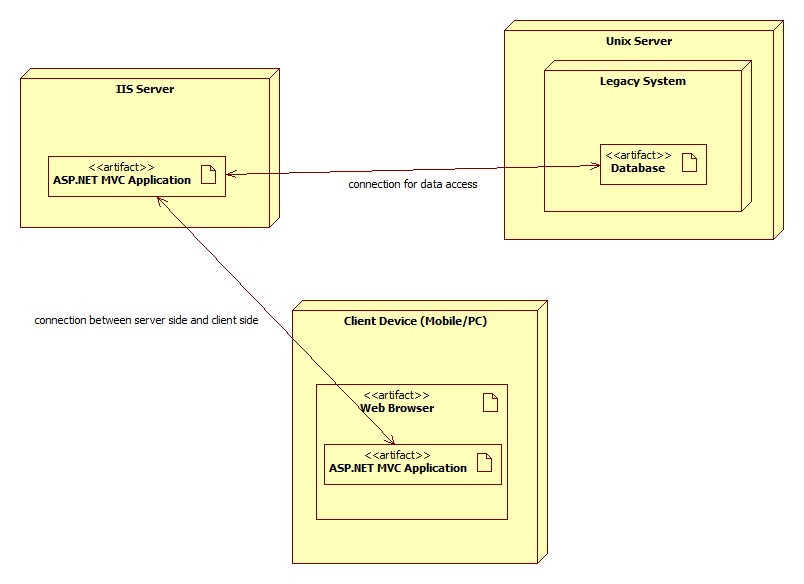
**Section I – Supplementary Specifications (Non-Functional Requirements)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Specification** | **Implementation Strategy** | **How is it measured** | **Benchmark Value** |
| Functionality |  |  |  |
| Usability |  |  |  |
| Reliability |  |  |  |
| Performance |  |  |  |
| Scalability |  |  |  |
| Re-usability |  |  |  |
| Testability |  |  |  |
| Security |  |  |  |
| Adaptability to different locations (Onshore and Offshore site in Vietnam) |  |  |  |
| Copyright and Intellectual property protection |  |  |  |
| Pre-release Testing (Refer to Section 2.1.4 of the Organizational Standards Document) |  |  |  |

**Section J – Deployment**

|  |  |  |
| --- | --- | --- |
| **Hardware Requirements** | **Software Requirements** | **Notes** |
| Web server – linux OS | IIS | In C# is the preferred web server software for handling HTTPS requests |
| Client workstation – multiple supported OS Variations | Web browser (chrome, safari, Firefox, ect.) | As the enrolment system is a web app, a browser will be necessary to access and interact with the system |
| Legacy system hosted on a Unix Server | Open SQL interface | Legacy System hosted locally on Unix Server and contains the DB. It is accessed via an API call and translated by the entity framework via the DAO |

*Also upload your Deployment UML models here*



**Section K – Analyse and Document the Impact of the new system**

1. 3 reasons the new system will add value to the existing business
   1. A Lecturer selecting a course to teach
   2. Student checking their report card
   3. Student registering for a course offering in the current semester
   4. A lecturer submitting grades
   5. A registrar maintaining lecturer information
2. Identify and explain 3 reasons the new system will add value to the exiting business.
   1. One system in Australia and Vietnam whilst remaining fast locally
   2. Increased efficiency of data access through the
   3. Increased availability and access to the new system, anywhere were internet is available.
3. Training gaps/needs of relevant stakeholders
   1. System Administrator
      1. They will need full technical documentation for the system, this includes, upgrades, maintenance and monitoring.
   2. Students
      1. Videos and webpages on how to enrol in a course offering, how to view, update, and delete a schedule
   3. Lecturers
      1. The lecturers will need documentation on how to use the system i.e. selecting interest in courses to teach, view schedule, record student grades
   4. Registrar
      1. Documentation on how the new enrolment system and their ‘native’ registrar’s system interact/how to interact with the new system in general

**Section L – Work Breakdown Structure (WBS) , Gantt Charts and Project metrics/costs calculations**

|  |
| --- |
|  |
|  |
|  |
|  |

**Section M – Verification/Validation and Signoff**

*Please tick-off Sections A-L completed*

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case** | **A** | **C** | **D** | **E** | **F** | **G** | **H** | **I** | **J** | **K** | **L** | **Project Lead Comments/Signoff** |
| View Report Card |  |  |  |  |  |  |  |  |  |  |  |  |
| Select Courses to Teach |  |  |  |  |  |  |  |  |  |  |  |  |
| Register for Courses |  |  |  |  |  |  |  |  |  |  |  |  |

**Section N – Post Project Analysis**

1. A range of at least 3 software development methodologies being used for similar Projects in industry - A table format with the name of the methodology, a brief description, the advantages, and disadvantages would suffice.

|  |  |  |  |
| --- | --- | --- | --- |
| Methodology | Description | Advantage | Disadvantage |
| Agile |  |  |  |
| Waterfall |  |  |  |
| Rapid Application Development (RAD) |  |  |  |

1. What would have been the most suitable methodology for this type of application. Explain your choice by describing the software development life cycle in the context of the selected methodology.

The most suitable methodology for this type of application the Agile methodology, as the Enrolment System Application does not have the rigid time structure that is associated with the waterfall method, leading to a more flexible approach more common with the Agile and RAD methodologies, allowing for iterative development.   
The reason Agile was chosen over RAD is because the Enrolment System Application does not require the prototyping used in the RAD approach. Instead of prototyping, implementing documentation that closely follows Agile’s iterative style proves to be the more effective strategy. Agile also adapts better to changes throughout the entire development life cycle whereas RAD only allows for changes during each iteration. Under ITWorks organisational policies and legislative requirements 2.1.2, Agile is listed as the preferred methodology, further increasing the candidacy of Agile as the methodology of choice.

1. A list of suggestions of how the Organizations current policies and procedures could be improved. Your suggestion focus on the technical (2 suggestions) and non-technical (2 suggestions) processes, procedures and standards.
2. Analyze the impact of any post project changes on the supply of hardware, software and skilled software engineering personnel and document how the organizations current business supply chain procedures and strategies could be improved to meet these challenges. This will usually take the form of a table as shown with the following columns:

|  |  |
| --- | --- |
| Risk | Analysis/Improvements/Mitigation Strategies |
|  |  |
|  |  |
|  |  |
|  |  |