**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**

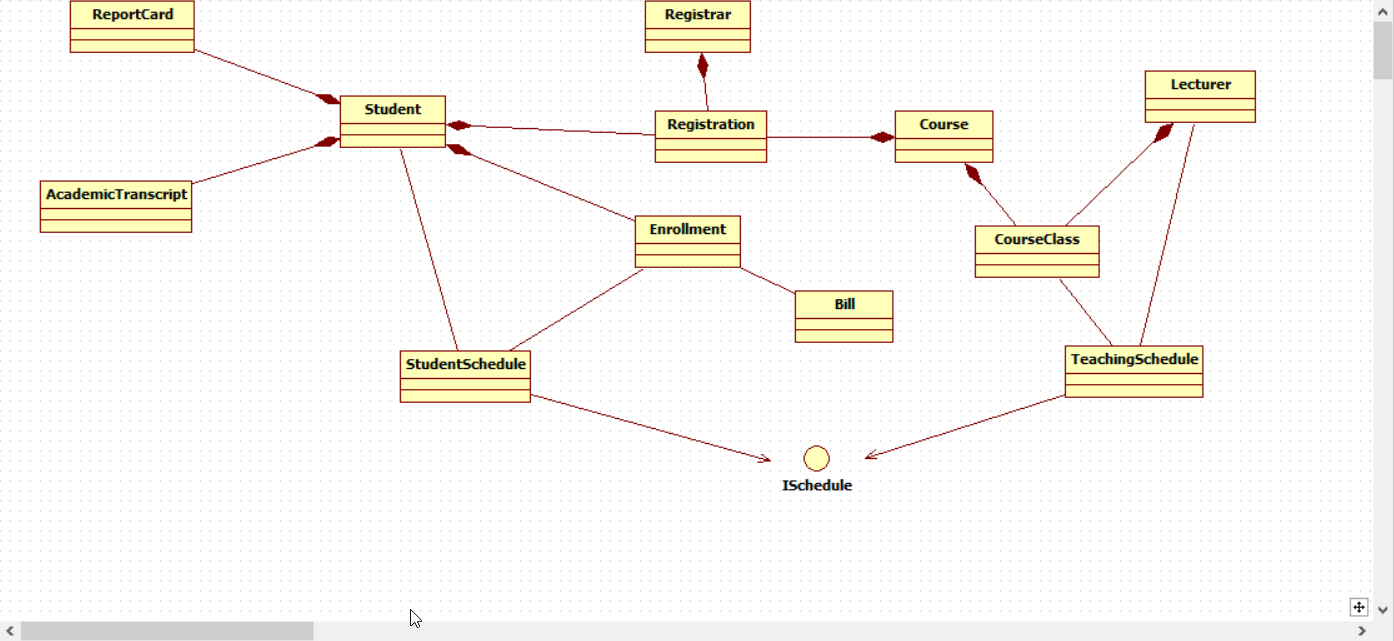
**The lead architect has called you into a meeting to discuss and determine the client’s business model from a list of potential alternatives before any concrete decision could be made on preliminary Architecture of the system.  The meeting will comprise of members of the team and the project as the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

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

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

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

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Analysis Mechanisms** | **Design Mechanisms** | **Implementation Mechanism** | **Justification for choice and influence of Organisational 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** | Selecting and adhering to established organisational procedures and standards ensures seamless integration between adapters and legacy systems, fostering interoperability, reducing errors, and enhancing system reliability and maintainability. |
| Cultural Specific and Worldwide considerations (Onshore and Offshore partner campus in Vietnam) and Distribution | Localisation support | .Net Localisation/Globalisation | .NET Framework supports localisation, accounting for linguistic and cultural differences |
| System Parameter Management | Parameter stored externally | Storing location reference in database | Storing parameters externally allows for centralised 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 | Selecting robust organisational procedures and standards ensures secure password hashing, username protection, and effective multi-factor authentication, enhancing overall cybersecurity resilience. |
| 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 | Selecting C# exception objects aligns with industry norms, enhancing code reliability. Adhering to organisational standards ensures consistent, resilient error handling. |
| 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 ASP.net Core 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 | Adopting RSA encryption for data at rest and TLS in transit ensures robust security, aligning with organisational standards and industry best practices, bolstering data protection. Utilise parameterised queries for enhanced security resilience. |

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

*Copy your Design/Component UML Models Here*

|  |
| --- |
| Several yellow papers with black text  Description automatically generated |
| **Andre Alexandrov** |

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

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

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

*Copy your Use Case Realisations UML Models Here*

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

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

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

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

|  |
| --- |
| *Copy your Class UML Models Here* |
| *Andre Alexandrov* |

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

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

##### **Section H – Database Design**

*Copy your Database Models Here*

|  |
| --- |
|  |

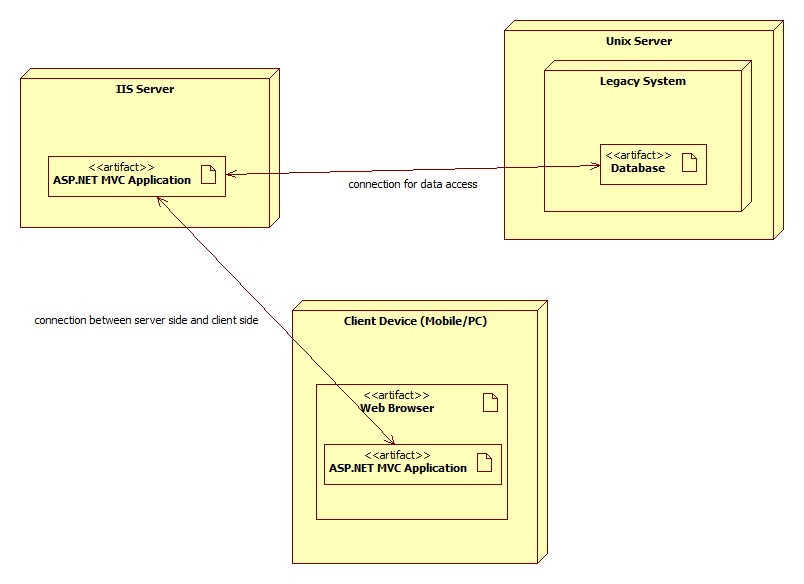
**Section I – Supplementary Specifications (Non-Functional Requirements)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Specification** | **Implementation Strategy** | **How is it measured** | **Benchmark Value** |
| Functionality | To ensure the correct functionality and alignment with user use cases, the implementation strategy should focus on user-driven development and iterative testing | Users driven functionality surveys i.e. quantitative and qualitative feedback from users regarding use case functionality. | To be in an acceptable state for sign-off we will require a user acceptance rate of above 85% |
| Usability | Follow HTML and CSS standards, use screen reactive design choices. C# generated HTML is in line with the standards and is customisable. | Web application works on multiple browsers during testing. | Application is displayed correctly in Chrome, Edge Safari and Firefox as well as a variety of screen classes. |
| Reliability | Implementation of load balancing to avoid system crashes due to high traffic, use of redundant servers and databases to ensure if one fails another can take over. | Measured by total downtime percentage over the year. | Downtime to not exceed 10% |
| Performance | Parallel processing and Asynchronous programming to optimise data retrieval from legacy database.  A horisontally scalable system in order to implement proper load balancing capable of handling 500 concurrent users. | Use load testing to simulate load testing to simulate 500 hundred users across database and servers.  Identify and log transaction times for performance analysis. | Legacy Database must be accessible with no more than a 10 second latency.  The system must be able to complete 80% of all transactions within 2 minutes. |
| Scalability | Implement a horisontally scalable system, to accommodate low - high volume traffic as well as availability requirements. | Incremental load testing allowing us to measure system and scaling performance at incrementally increasing loads. | System can handle 500 concurrent users on the servers without bottlenecking. |
| Re-usability | Adhering to the IT Works organisational policies and legislative requirements section 2.1.5 on reusing existing assets | Code duplication metrics.  Tracking the number of reusable components. | A code duplication metric of below 10%.  At least 70% of the system should consist of reusable components. |
| Testability | Unit Testing,  Integration Testing  Coding best practises. | Test Pass Rate | A test pass rate of 80% meaning the testers can easily understand what the code should be doing and can anticipate the likely output. Less than 20% unexpected result regardless of pass or fail. |
| Security | Authentication and Authorisation  RBAC - Role Based Access Control, | Penetration testing to identify vulnerabilities in authentication and authorisation mechanisms. | Regulatory and legislative compliance with ISO/IEC 27001. |
| Adaptability to different locations (Onshore and Offshore site in Vietnam) | Implementation of .net framework localisation features. | User based testing – user acceptance testing for  translation accuracy,  Cultural appropriateness and usability. | For every five users three users feedback should be positive for user satisfaction metrics in regards to Vietnamese users as well as onshore users. |
| Copyright and Intellectual property protection | Legal compliance.  Documented policies and procedures.  IP Asset management | Regular audits on requirement compliance.  Regular reviews of documentation and employees understanding of said documentation.  Periodic valuation of IP Assets and their protection status. | 100% legal compliance.  Up to date policies and 100% of staff acknowledge understating of their contents.  Protection strategies in place for identified intellectual property assets. |
| Pre-release Testing (Refer to Section 2.1.4 of the Organisational Standards Document) | A pre-release version of the application is distributed too beta testers for validation | MTTF – Mean Time to Failure.  MTBF – Mean Time Between Failures.  EPKLOC – Errors Per Kilo of Code.  Functional Point Errors | MTTF – to be determined  MTBF – to be determined  EPKLOC – to be determined  Functional point errors – to be determined.  The above metrics require further data in order to benchmark. |

**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 and mobility through the use of middleware
   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**

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

**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 | Iterative and incremental approach with a focus on flexibility and adaptation | Fast feedback and delivery,  Adapts to changing requirements,  Focus on continuous improvement | Requires strong communication and teamwork, Can lack upfront planning and documentation,  Can be difficult to manage large projects |
| Waterfall | A sequential, linear approach with defined phases | Clear roadmap and scope control,  Well documented processes, Predictable delivery timelines | Less flexible to changing requirements,  Can lead to late-stage errors,  Can be slow for smaller projects |
| Rapid Application Development (RAD) | Prototyping-based approach with an emphasis on quick feedback and delivery | Fast initial development and time-to-market,  Enables early testing and user involvement,  Can reduce development risks | Can lack detailed documentation and design,  Less suited for complex or large projects,  Can be challenging to maintain and scale |

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.

Agile is the preferred development methodology for online business applications at ITWorks, and is the most suitable in the case of the development of the Enrolment System Application (a web app) as it is not held to the same rigid time structure that is normally associated with the Waterfall methodology. This leads to a more flexible approach more in line with the Agile and Rapid Application Development (RAD) methodologies, allowing for iterative development.

Again, Agile was selected over RAD, as development of the Enrolment System Application does not require the prototyping required within the RAD methodology. Planning and documentation instead is implemented to support Agile’s iterative development process.

1. A list of suggestions of how the Organisations current policies and procedures could be improved. Your suggestion focus on the technical (2 suggestions) and non-technical (2 suggestions) processes, procedures and standards.

|  |  |  |  |
| --- | --- | --- | --- |
| **Current Policy and Procedure** | **Category** | **Proposed Improvement** | **Justification** |
| 2.1.3.1 Modelling tool is Star UML version 3 or 5 | Technical | Modelling tool should be standardised to Star UML version 5 | Star UML version 5 is cleaner, easier to use, standardising to one version creates a more unified, seamless environment to document systems |
| 2.1.4 Pre-Release Testing | Technical | Delivery In Full, On Time (DIFOT) used as a measure for testing, replacing MTTF and MTBF | MTTF and MTBF are not the best metrics to measure the testing of a software application, DIFOT is more consistent within the Agile approach |
| 3.2 Standards | NonTechnical | OWASP to be used in conjunction with ISO27001 | ISO27001 is more comprehensive when considering the Australian ICT landscape, is a global industry best practice standard |
| 3.3 Breaches | NonTechnical | The process for assessing the risks in the event of a breach of personal data should be clearer, as well as an escalation procedure including exactly who we are reporting the breach to | Clearer guidelines provide for a quicker, more streamlined response, which in turn could mitigate some damage from the event |

1. Analyse the impact of any post project changes on the supply of hardware, software and skilled software engineering personnel and document how the organisations 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** |
| Hardware Supply | Increased demand may strain hardware availability or lead to delays. To mitigate this, we should establish strong relationships with reliable hardware suppliers, as well as implementing concise maintenance schedules to predict issues and prolong hardware lifespan. |
| Software Supply | Changes in software requirements may necessitate updates or acquisitions of new software. To mitigate this, we can continue to invest in agile development methodologies, providing flexibility. We should also be regularly assessing and updating software licenses as required. |
| Skilled Personnel | Increased demand for specialised skills may create challenges for recruitment. To mitigate this, we can build ‘talent pipelines’ with educational institutions as well as offering attractive developmental opportunities. |
| Supply Chain | Developments and changes in global events may disrupt the supply of goods and services. To mitigate this, we should closely monitor key supply chains, and diversify suppliers wherever possible. |