**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** |
| Database Admin |  |  |  |
| **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**

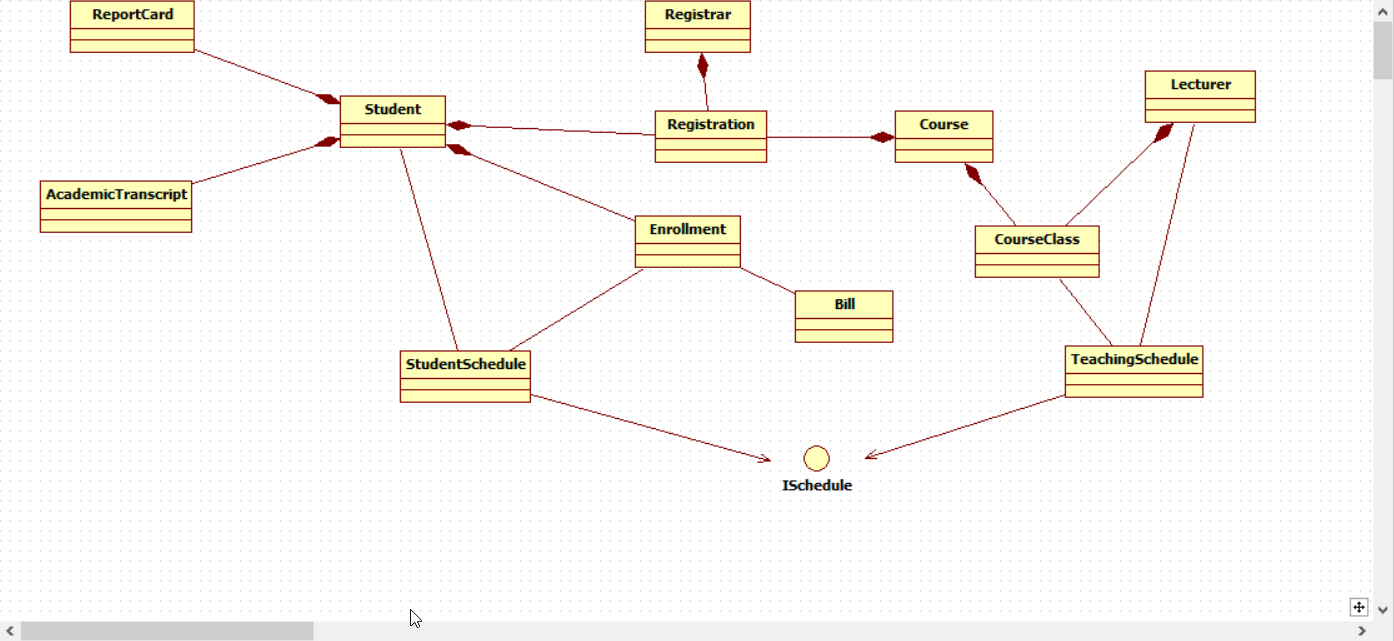
**The lead architect has called you into a meeting to discuss and determine the client’s business model from a list of potential alternatives before any concrete decision could be made on preliminary Architecture of the system.  The meeting will comprise of members of the team and the project as the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

****

* 1. **Bunch of entities so MVC 😊**

1. **Explain the Object-Oriented Design/Programming principles have influenced the systems architecture, with emphasis on the MVC pattern**
2. **Document above in the Software Architecture Document (approx. 200 words), under the proper heading.**

**Use Case: Register for Courses**

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

*Copy your User Experience UML Models Here*

##### **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*

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

*Copy your Use Case Realizations UML Models Here*

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

*Copy your Class UML Models Here*

##### **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** |
| Azure |  |  |
|  |  |  |
|  |  |  |

*Also upload your Deployment UML models here*

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

**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 |  |  |  |  |  |  |  |  |  |  |  |  |

**Section N – Post Project Analysis**