**Enrolment System - Report Template**

|  |  |
| --- | --- |
| Student ID | Name |
| 00116411 | Andre Alexandrov |

**Part I - Gathering and Documenting Requirements**

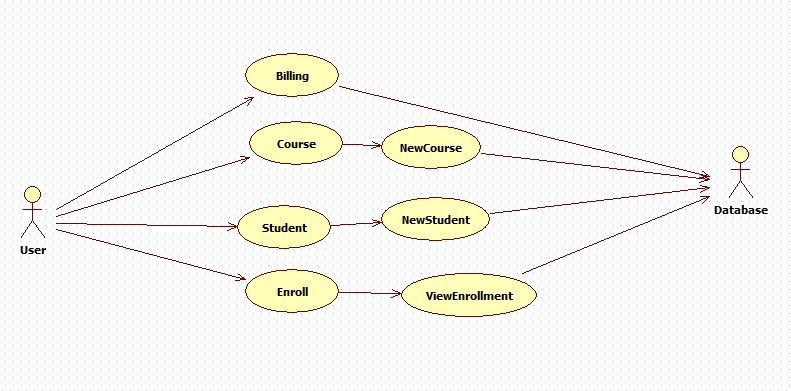
**Technical requirements:**

|  |  |
| --- | --- |
| **Requirement ID** | **Description** |
| ***RE01*** | ***Legacy UI must be maintained*** |
| RE02 | Multi user application |
| RE03 | Must implement Cloud storage |
| RE04 |  |
| RE05 |  |

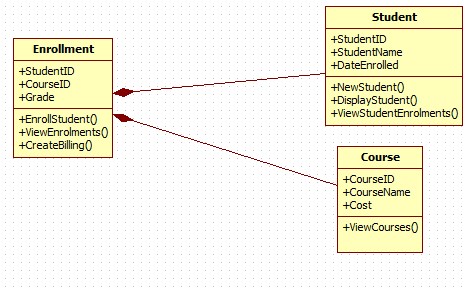
**Technical Considerations:**

|  |  |
| --- | --- |
| **Considerations** | **Response** |
| 1. Hosting | The recommendation is to host the service on the cloud, this is due to the cost of setting up a server locally it can be a massive initial cost with ongoing monthly cost, space is also a consideration, the last consideration is that of scalability, cloud allows for near instant scaling where as on prem can be expensive and time consuming |
| 1. Cloud Model | The delivery Model chosen is PaaS as it allows for enough flexibility with data management and storage and also application creation and usage without so much freedom of needing to create/choose the OS and middleware. |
| 1. Service Provider (Max 100 words) | Azure is the chosen service provider as they have great integration with Visual studio 2022 with creating multi user services and also provides scalable storage suited for this application |
| 1. Database (Max 100 words) | Azure Cosmos DB is a suitable alternative to local SQL Server on Azure. It offers globally distributed, multi-model databases with low-latency, high-throughput performance. Its multi-model support and automatic scaling make it well-suited for this application |
| 1. IDE | Microsoft Visual Studio 2022 |
| 1. Architecture and Framework | WCF is the chosen framework as it is used extensively for the creation of service applications |
| 1. Uptime strategy (Max 100 words) | Implement a robust uptime strategy by leveraging Azure's Availability Zones, distributing resources across multiple data centers for high availability.  Utilize services with SLAs that include uptimes, such as Azure App Service (99.95% uptime).  Design auto-scaling and load balancing.  Regularly monitor and analyze service health.  Employ backup and disaster recovery mechanisms to ensure minimal downtime.  Regularly review and update the strategy |

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



**Classes and relationship (Add class diagrams here)**

****

**List of services to be implemented**

|  |
| --- |
| EnrollStudent |
| DisplayStudent |
| ViewEnrolments |
| CreateBilling |
| NewStudent |
| ViewStudentEnrolments |
| ViewCourses |