Design choices

A Model View Controller(MVC) ASP.Net core template was used to develop the web based application as it allows for efficient separation of focus under Models, Views and Controllers. Models focusses on the logic behind the web based application while Views focusses on User experience and Controllers will choose which pages to display and what to do after processing the users input. The framework also has built in features which allow for C# models to be used in databases instead of writing SQL code which can help in reducing amount of errors thus allowing faster productivity

Structure of Database

The database will have 6 tables:

**Lecturer table**: Stores information about the lectures who have submitted claims

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data type | kye | constraints | Description |
| LecturerID | int | Primary key | Not null, unique | Holds the specific Lecturers ID |
| FullName | VARCHAR(70) |  | NOT NULL | Lecturer name |
| PhoneNumber | VCHAR(15) |  |  | Lecturer phone number |
| Email | VARCHAR(70) |  | unique | Lecturer email address |

Program Coordinator: Stores information about the coordinator

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data type | Key | Constraints | Description |
| CoordinatorID | int | Primary | Nut null | Holds the specific Coordinators Id |
| FullName | Varchar(70) |  | Not null | Coordinators ful name |
| PhoneNumber | Varchar(15) |  |  | Coordinator phone number |
| Email | Varchar(70) |  | unique | Coordinator email address |

Academic Manager: Stores Information about the Academic Manager

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data type | Key | Constraints | Description |
| ManagerID | Int | Primary | Not null | Holds the specific Managers ID |
| FullName | Varchar(70) |  | Not null | Mangers full name |
| PhoneNumber | Varchar(15) |  |  | Managers phone number |
| Email | Varchar(75) |  | unique | Managers Email |

Claim: Holds information on claims made and records when the claim was made, the hours worked, lectures hourly rate, the amount of the claim,when the claim was made and the status of the claim

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data type | Key | Constraints | Description |
| ClaimID | Int | Primary | Not null | Holds the specific Claims ID |
| LecturerID | Int | Foreign | Not null | Comes from lecturer table |
| ClaimMonth | Date |  |  | When the claim was made |
| HourlyRate | Decimal |  |  | Amount charged per hour |
| HoursWorked | Decimal |  |  | Amount of hours worked |
| Amount | Decimal |  |  | HourlyRate multiplie by HoursWorked |
| Status | Varchaar(20) |  |  | Submitted,Approved or Rejected |

Document: For any document submitted with each claim to help with approvals

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Key | Constraint | Description |
| DocumentID | Int | Primary key | Not null | Holds specific documents ID |
| ClaimID | Int | Foreign | Not null | Holds specific claim ID |
| DocumentType | Varchaar(70) |  |  | Type of Document |

Claim Approval: Stores results of claim(Approved or not)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data type | Kye | Constraint | Description |
| ApprovalId | Int | Primary key | Not null | Holds specific approval id |
| ClaimID | Int | Foreign | Not null | Holds specific claim ID |
| ApproverId | Int |  | Not null | Holds specific approvers ID |
| ApproverRole | VARCHAR(75) |  |  | Manager or Coordinator |
| Decision | VARCHAR(20) |  |  | Approved or rejected |
| DecisionDate | DateTime |  |  | When decision was made |

Assumptions

* All claims require supporting documents to be submitted
* Claims are only done monthly

UML Class Diagram

A black background with white text

AI-generated content may be incorrect.

Project plan

Task1:

* Reviewing the requirements: Study the POE and spot what is needed to done
* Does not depend on anything
* Will take 2 hours

Task 2:

* Designing database & UML:creating the tables needed for the database and ERD diagram
* Depends on task 1 to be completed
* Will take 4

Task 3:

* GUI prototype: Designing screens that will be used in the we application
* Depends on task 2 to be completed
* Will take 1/2 day(s)

Task 4

* Documentation: documenting choices/decisions made about the structure of the database, its designs and any assumptions made
* Depends on task 2
* Will take 1 day

Task 5

* Test and submit: perform last tests to ensure efficiency and group everything and submit
* Depends on task 3 and 4
* Will take 3 hours