**NAME(s): LETHABO MOKGETHWA**

**SURNAME: MAPHOTO**

**STUDENT NUMBER: ST10443029**

**MODULE CODE: PROG6212**

**CAMPUS: RC BRAAM**

***Design Choices***

* Technology:

ASP.NET Core MVC is selected for its clean separation of concerns through the Model–View–Controller pattern, which enhances maintainability, scalability, and ease of future extensions. This choice enables independent development and testing of business logic, user interface, and data access components, directly supporting long-term system growth.

* Front-end:  
  The combination of Razor views with Bootstrap ensures the user interface is both responsive and intuitive, catering to users on various devices with minimal learning curve. This supports accessibility and a professional look without excessive complexity.
* Database:  
  SQL Server forms the foundation for robust relational data storage, with EF Core in-memory used for rapid prototyping and testing. The core schema emphasizes simplicity and clarity, centering around Claims, Users, and Roles to streamline claim management while keeping future integration possibilities open.
* Security:  
  To maximize data integrity and enforce process accountability, only Programme Coordinators and Academic Managers are granted claim approval rights, while Lecturers are limited to submitting and viewing their own claims. This role-based access strictly minimizes errors and inappropriate approvals, reflecting a strong security rationale.
* Claim Tracking:  
  Each claim features a Status (Pending, Approved, Rejected) and a submission date for full transparency and auditability, supporting efficient workflow tracking.

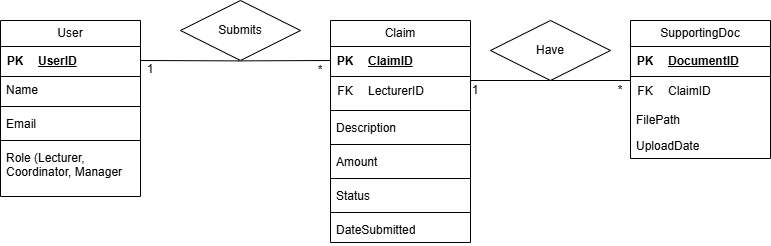
***GUI Layout***

* **Dashboard (Index Page):**  
  Presents a clear snapshot of all claims, their statuses, and actionable buttons (where permitted) to streamline daily workflow for all user types.
* **Claim Submission Form:**  
  Prioritizes usability with essential fields only—claim name, description, amount, and document uploads—reducing friction while ensuring comprehensive submissions.
* **Approval Buttons:**  
  Approve/Reject options dynamically appear only for authorized coordinators/managers, which maintains interface clarity and prevents user confusion or mistakes.
* **Consistency:**  
  A persistent navigation bar provides quick, predictable access to all major pages, ensuring efficient user journeys throughout the application.

***Assumptions & Constraints***

* **Authentication:**  
  Only registered and authenticated Lecturers, Coordinators, and Managers may access the system, guaranteeing privacy and compliance with institutional policy.
* **File Upload Limit:**  
  Supporting documents are limited to 10MB per upload to balance system performance and user needs.
* **Prototype Scope:**  
  No payroll integration exists in the prototype phase—final integration will be addressed in production, focusing development on the core claims process and avoiding unnecessary complexity at this stage.
* **Database Simplicity:**  
  The initial design purposely omits advanced features (such as auditing/logging) to ensure clarity but is extensible for future additions as organizational needs evolve.

***UML DIAGRAM***



|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Description** | **Dependency** | **Duration** |
| Requirements Gathering | Identify system features & roles | None | 2 days |
| Database Design | Define tables & relationships | Requirements | 2 days |
| Create MVC Project | Setup ASP.NET Core MVC structure | Database Design | 1 day |
| Implement Models | Create Claim, User, SupportingDoc models | Project Setup | 2 days |
| Implement Controllers | Add logic for submit, approve, reject | Models | 2 days |
| Develop Views (UI) | Razor pages for dashboard & forms | Controllers | 3 days |
| File Upload Feature | Allow lecturers to upload supporting docs | Views | 2 days |
| Testing | Validate claim submission & approval | All previous | 3 days |
| Documentation & UML | Write design explanation & diagrams | Parallel | 2 days |
| Final Review | Ensure prototype works as required | Testing | 1 day |