

PROG6212

PART 1



ITUMELENG KOLA

ST10075573

DISD2 GRP1

**Documentation:**

Design Decisions: Modularity, ease of use, and simplicity were all important factors in the creation of the Contract Monthly Claim System. There is a clear separation between the database, business logic, and front-end. Because of its effective division of Model, View, and Controller responsibilities which makes project maintenance and expansion easier ASP.NET Core MVC was chosen. To make sure that the application can be accessible on several devices, including PCs, tablets, and smartphones, responsive designs are made using Bootstrap.

User Experience:

The navigation is kept simple and intuitive. Users have access to different views based on their roles:

Lecturers: Can submit claims and upload documents.

Programme Coordinators/Academic Managers: Can verify, approve, and reject claims.

Claim Status: Everyone can track the status of the submitted claims transparently.

The home page features a clear call-to-action with buttons directing users to the main system functionalities.

Forms include the minimum required fields to ensure quick submission, making the system user-friendly for non-technical users.

Database Structure:

A relational database is ideal for this system, as it can store hierarchical data such as claims, users, and roles.

We use Entity Framework Core for object-relational mapping, allowing easy integration between the C# object model and the database schema.

Assumptions and Constraints:

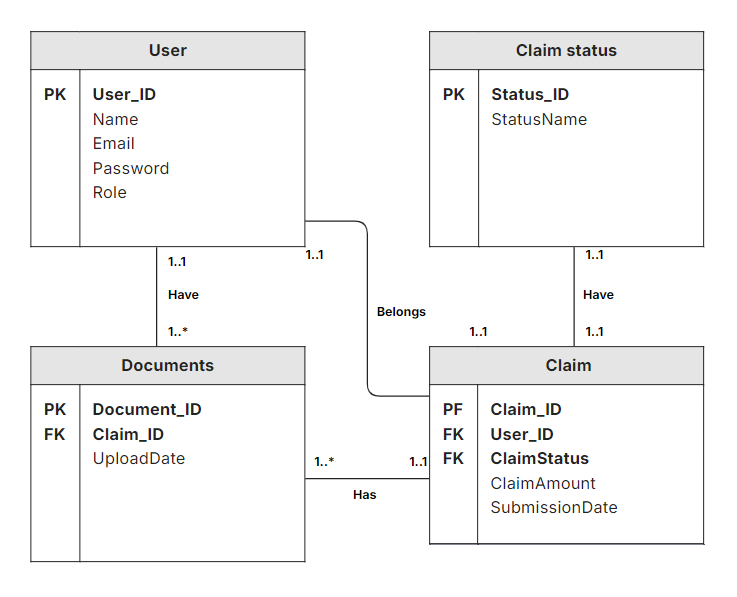
User Roles: We assume that each user has a pre-defined role that limits their access and privileges in the system. Lecturers can only submit and track claims, while managers and coordinators can verify and approve claims.

Claim Status Tracking: The system tracks the status of each claim from submission through verification to approval or rejection.

File Upload: Lecturers must be able to upload supporting documents for their claims.

Data Consistency: The system ensures consistent claim status updates by preventing multiple updates at the same time.

**UML**



**Project Plan**

|  |  |
| --- | --- |
| **Phase** | **Duration** |
| **1: Planning & Design**  Determine the necessary database schema, user roles, and essential functionalities. Designing Prototypes and Wireframing  Make prototypes for the home page and user interface. Create the UML diagram and the database structure.  **2. Development of Front-End**  Design and Homepage  Include the logo in the homepage and navigation bar design. Use Bootstrap to add some basic style. Send in the Claim Page  Create the claim submission form. Make input fields for uploading files and claiming details. Check and Accept the Claims Page  Create the table that will show the claims that have an approval or rejection option. Add buttons to accept or reject claims. View the Status of Your Claim.  Create a table view so that users may monitor the progress of their filed claims. | 3 Days |
| **3: Back-End**  Creating a Database: Use Entity Framework Core to implement the database schema. Establish connections between the tables for Users, Claims, Documents, and ClaimStatus. Logic for Controllers: Construct controller logic (Submit, Verify, Track) to handle requests. Put the repository pattern into practice for database queries. Testing and Troubleshooting : Check the usability of each form and view. Debugged problems with data submission, form validation, and file uploads. | 1 Week |
| **4: Review and Finalization**  Code Review & Documentation: Write thorough comments and code documentation. Review the code with the team and make any necessary changes. Final Testing: Verify that every user flow is operating as planned. Implementation and Transition: Install the prototype in a rehearsal setting. final distribution to interested parties. | 2 Days |
|  |  |

REFERENCES

*ASP.NET MVC Pattern: .NET* (no date) *Microsoft*. Available at: <https://dotnet.microsoft.com/en-us/apps/aspnet/mvc#:~:text=Model%20View%20Controller%20(MVC),to%20achieve%20separation%20of%20concerns>. (Accessed: 01 September 2024).

*Microsoft* *Microsoft Support*. Available at: <https://support.microsoft.com/en-us/office/database-design-basics-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5> (Accessed: 05 September 2024).