1. Documentation
•Design Choices

- Frontend Framework: Because of its adaptability, scalability, and interoperability with the .NET ecosystem, we decided to design the GUI using ASP.NET Core MVC. Robust Model-View-Controller (MVC) architecture is supported by this framework, guaranteeing a clear division of responsibilities between data access, business logic, and the presentation layer.

- Backend Framework: To manage server-side logic, the backend will also make use of ASP.NET Core MVC. To control data access to a SQL Server database, we chose to use Entity Framework Core as the Object-Relational Mapper (ORM). With this option, robust type-checking is enabled and the ASP.NET stack is easily integrated.

- Database: Due to its reliability, capacity to execute complicated queries, and interaction with the.NET ecosystem, an SQL Server database will be utilized to hold data for claims, lecturers, and approvals.

• Design Approach

- Based on the MVC pattern, the application is divided into three interdependent components in the design:

1. Model: Serves as a representation of the data along with the business rules controlling updates and access to it.
2. View: Offers the user interface (UI) for communication with users.
3. Controller: Oversees user input and updates the model or view as needed, managing communication between the model and the view.

• Database Structure
The Database will include the following tables:
- Lectures Table
- Claims Table

- Documents Table
- Approvals Table

• Layout of GUI

- Lectures Dashboard

A straightforward form for entering claim information (Notes, Hours Worked, Hourly Rate.

An easily accessible "Submit Claim" button that is clearly displayed.

An "Upload" button and a "Upload Documents" section allow you to attach supporting files.

A "Claim Status" area that shows the filed claims' current status in real time.

- Manager Dashboard

All filed claims are shown in a grid view that can be sorted by lecturer, date, status, and other factors.

There will be "Approve" and "Reject" buttons on each row.

Detailed claim view displaying claim information, documentation, and background

- HR Dashboard

A view to produce invoices or summary reports for claims that have been accepted.

Interface components for handling lecturer data, like contact information updates.

• **Assumptions & Constraints**

- To utilize the online application, all users should have access to the relevant hardware and software. Only authenticated users can access their individual dashboards, and only authorized individuals should have the authority to approve claims. For the program to be used on a variety of devices, the user interface (UI) must be responsive and accessible and correspond to data protection rules, guaranteeing the safe processing and storage of financial and personal information.

• Rationale Behind Design Choices

    - User Interface Design: To lower the learning curve for users, we gave simplicity and usability top priority in the GUI design. Users' roles are considered when designing the dashboards, guaranteeing access to the necessary features.

    - Entity Framework Core: Simplifies complicated data interactions to provide efficient data access and management.

    - Security: To ensure safe access and data integrity, we built authentication and authorization regulations.

("24 Things to Consider When Designing and Developing a Website," 2014)

2.UML Class Diagram
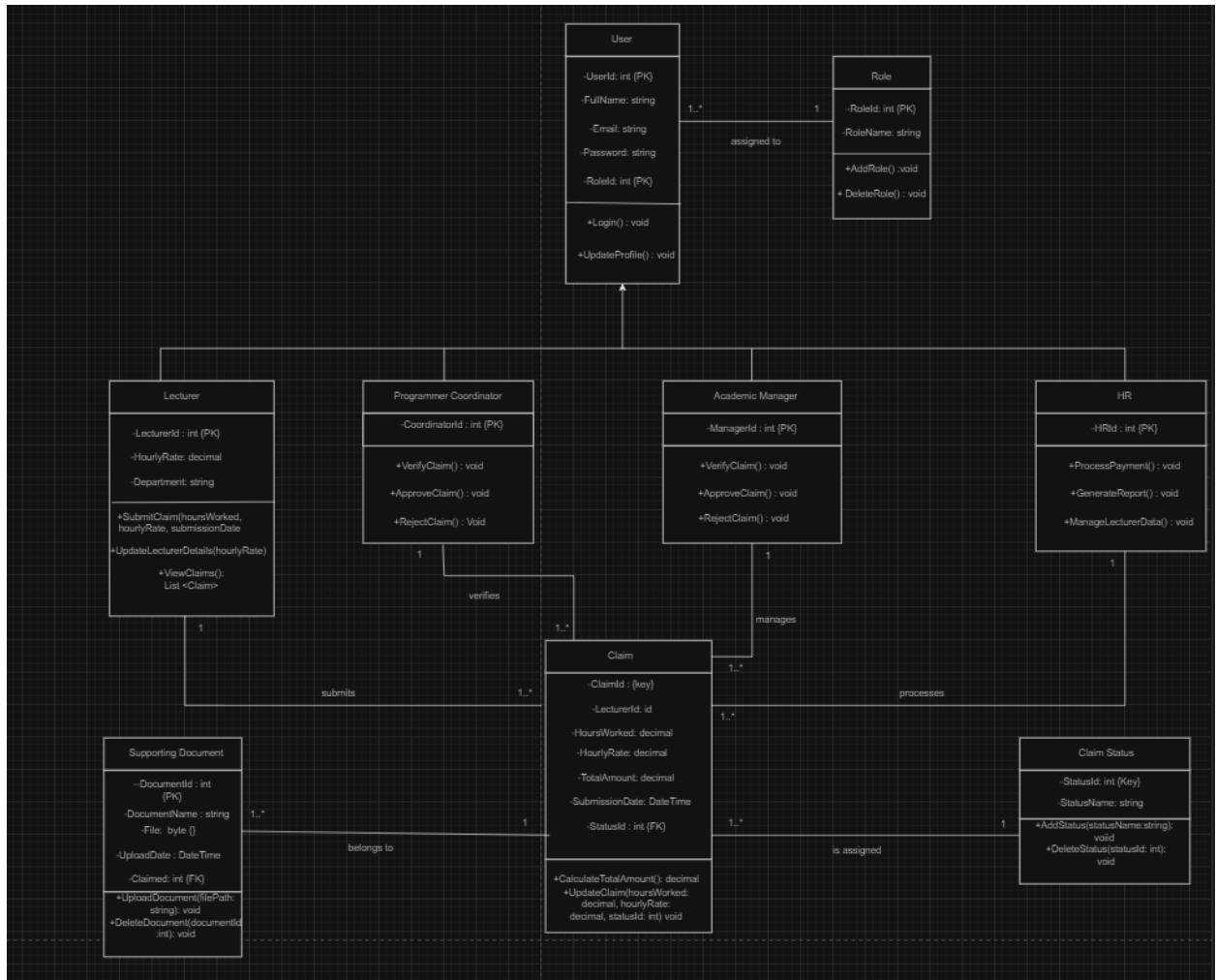
Satzinger, J., Jackson, R. and Burd, S. 2016. Systems Analysis and Design. 7th ed.Toronto: Cengage.

3.Project Plan

- Overview

This project entails creating a GUI prototype, putting the UML class diagram into practice, and creating documentation for the Contract Monthly Claim System (CMCS). The tasks, requirements, and schedule for finishing Part 1 of the assignment are described in this plan.

Week 1

Specify the goals and scope of the project. (One day)
 - Describe the prototype's goals in brief.
 - Specify the features and functionalities that will be present.

Research. (Two Days)
 - Identify and thoroughly research tools and possible events.

UML Class Diagram and Database Design.(Three Days)
- Based on the requirements, create a comprehensive UML class diagram. Then, examine and edit the diagram as necessary.
- Based on the UML diagram, define the database schema.

Week 2

Design GUI Layout. (Three Days)
 - Make sure the functionality and usability of the design satisfy the needs of the user.

Develop GUI Prototype. (Four Days)
 - Make sure the GUI has all the necessary components, including forms for submitting claims and tracking their progress.
 - Use.NET Core to implement the GUI design (MVC or Windows Forms).

Week 3

Project Documentation. (Three Days)

- Clearly explain the database structure, GUI layout, and design decisions you made.

  - Record any presumptions or limitations that were taken into account.

Finalize Documentation (One Day)

  - Make any necessary adjustments after checking the documentation for accuracy and completeness.

Submission

  - Create a report by combining the documentation, GUI design, project plan, and UML class diagram, and then submit it in compliance with the requirements.

Week 4

Final Adjustments

  - Commit and Push changed to GitHub (5 times)

  - Make any final adjustments if needed.

Dependencies

  - Before beginning to create the GUI, the UML Class Diagram must be finished.
- Before preparing documentation, GUI design must be finished.
-Prior to project submission, documentation must be completed.

Notes

  - All tools that will be used have to be available to me.

  - Ask for feedback to see what needs to be changed or improved.

  - Time allocations are based on effort and may need to be adjusted on actual progress or interruptions.

(deBara, 2024)

4.GUI IU

Login and Register Page

Welcome

Contract Monthly Claim System

Email

Value

Password

Value

Sign In

Forgot password?

Lecturer Dashboard Page

Submit New Claim

Claim ID	Rate / hour	Hours Worked	Amount	Submission	Supporting Document	Status
1	R100	10	R1000.00	2024-05-20	View Submitted Document	Approved
2	R200	20	R4000.00	2024-06-20	View Submitted Document	Rejected
3	R150	15	R2250.00	2024-07-20	View Submitted Document	Approved
4	R100	17	R1700.00	2024-08-20	View Submitted Document	Approved
5	R200	27	R5400.00	2024-09-20	View Submitted Document	Pending

Submit Your Claim Page

Hours Worked   Submit

Hourly Rate   Submit

Additional Notes   Submit

Upload Supporting Documents

HR Dashboard Page

Claim ID	Rate / hour	Hours Worked	Amount	Submission	Supporting Document	Status	Action
1	R100	10	R1000.00	2024-05-20	View Submitted Document	Approved	Reviewed
2	R200	20	R4000.00	2024-06-20	View Submitted Document	Rejected	Reviewed
3	R150	15	R2250.00	2024-07-20	View Submitted Document	Approved	Reviewed
4	R100	17	R1700.00	2024-08-20	View Submitted Document	Approved	Reviewed
5	R200	27	R5400.00	2024-09-20	View Submitted Document	Pending	Approve Reject

HR Generate Report Page

Select date

Enter date 📅

Date
2024/05/20

Cancel   OK

To

Select date

Enter date 📅

Date
2024/06/20

Cancel   OK

Generated Reports

Claim ID	Rate / hour	Hours Worked	Amount	Submission	Supporting Document	Status
1	R100	10	R1000.00	2024-05-20	View Submitted Document	Approved
2	R200	20	R4000.00	2024-06-20	View Submitted Document	Rejected

Dowload Report

HR Process Payments Page

Claim ID	Rate / hour	Hours Worked	Amount	Submission	Supporting Document	Status
1	R100	10	R1000.00	2024-05-20	View Submitted Document	Approved
3	R150	15	R2250.00	2024-07-20	View Submitted Document	Approved
4	R100	17	R1700.00	2024-08-20	View Submitted Document	Approved

Select All ☐

Process Payments

Manage Users Page

Name	Role	Status	Last Login	Action
Alice Johnson	Lecturer	Active	2024-09-08 10:15	Edit Delete
Mark Thompson	HR Manager	Active	2024-09-09 14:30	Edit Delete
Sarah Williams	Program Coordinator	Active	2024-09-07 09:50	Edit Delete
David Brown	Academic Manager	Inactive	2024-08-30 16:00	Edit Delete

Add New User

Update User Role Page

Track Your Claims Page

References

24 Things to Consider When Designing and Developing a Website. (2014, March 6). *Social Media Today*. https://www.socialmediatoday.com/content/24-things-consider-when-designing-and-developing-website

Accessed: 5 September 2024

deBara, D. (2024, February 16). *How to write an effective project plan in 6 simple steps - Work Life by Atlassian*. Work Life by Atlassian. https://www.atlassian.com/blog/project-management/write-an-effective-project-plan

Accessed: 6 September 2024

The IIE. 2024. Databases Advanced Databases Module Manual 2024, 1st ed. The IIE.

Accessed: 8 September 2024