ST10085622

LIYEMA MANQOYI

PROGRAMMING 2B

PORTFOLIO OF EVIDENCE

PART 1

10 SEPTEMBER 2024

GIT-REPO LINK:

https://github.com/lemi-manqoyi/ContractMonthlyClaimSystem.git

DESIGN CHOICE

I decided to go with the C# MVC design pattern and ASP.NET (Entity Framework) as the framework of the web application Contract-Monthly-Claim-System (CMCS).

The MVC, according to (Kirinriki, 2024), stands for Model-View-Controller. It separates the application into three interconnected layers: the ***model,*** which represents the data and business logic of the application; the ***view,*** which handles the presentation layer, also known as the user interface; and the ***controller***, which serves as a middleman between the Model and View, regulating the exchange of data and business logic.

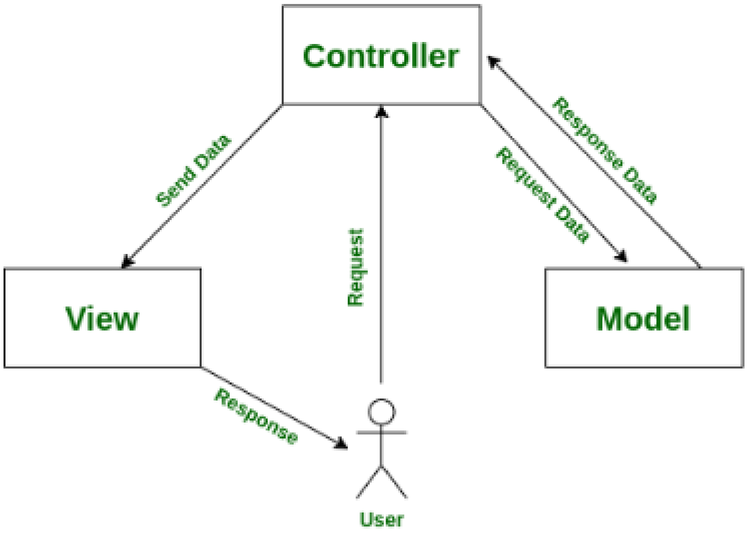


Figure 1: MVC Explained, adapted from (Anushka, 2021)

The user interacts with the view through the controller.

According to the research of (Anushka, 2021), the MVC approach enables for faster development by separating code into three layers. This lets one developer focus on a certain piece (e.g., view) while another can work on another section (e.g., controller). This simplifies the implementation of business logic while also accelerating the development process fourfold. It has been shown that, when compared to other development models, the MVC paradigm results in faster development times.

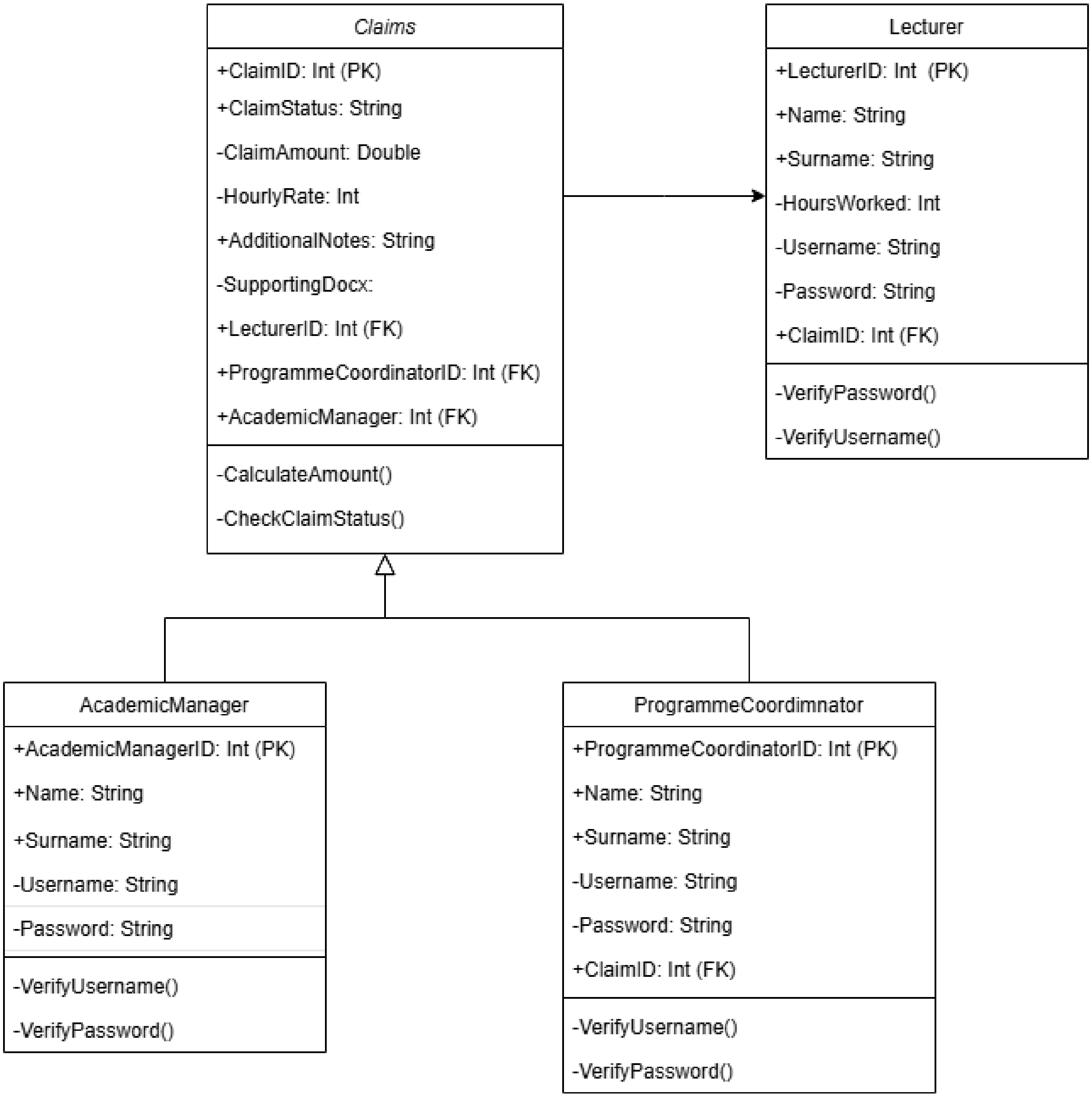
ASP.NET Core MVC model binding transforms client request data (form values, route data, query string arguments, and HTTP headers) into objects that the controller can process. As a result, your controller logic does not need to figure out the incoming request data; it just passes the data as parameters to its action methods.

The information above and below has been adapted from (Ardalis, 2023) and continues to add that ASP.NET Core MVC provides validation by covering your model object with data annotation validation characteristics. The validation characteristics are verified on the user's side prior to values being sent to the server, and on the server's end before the controller action is executed.

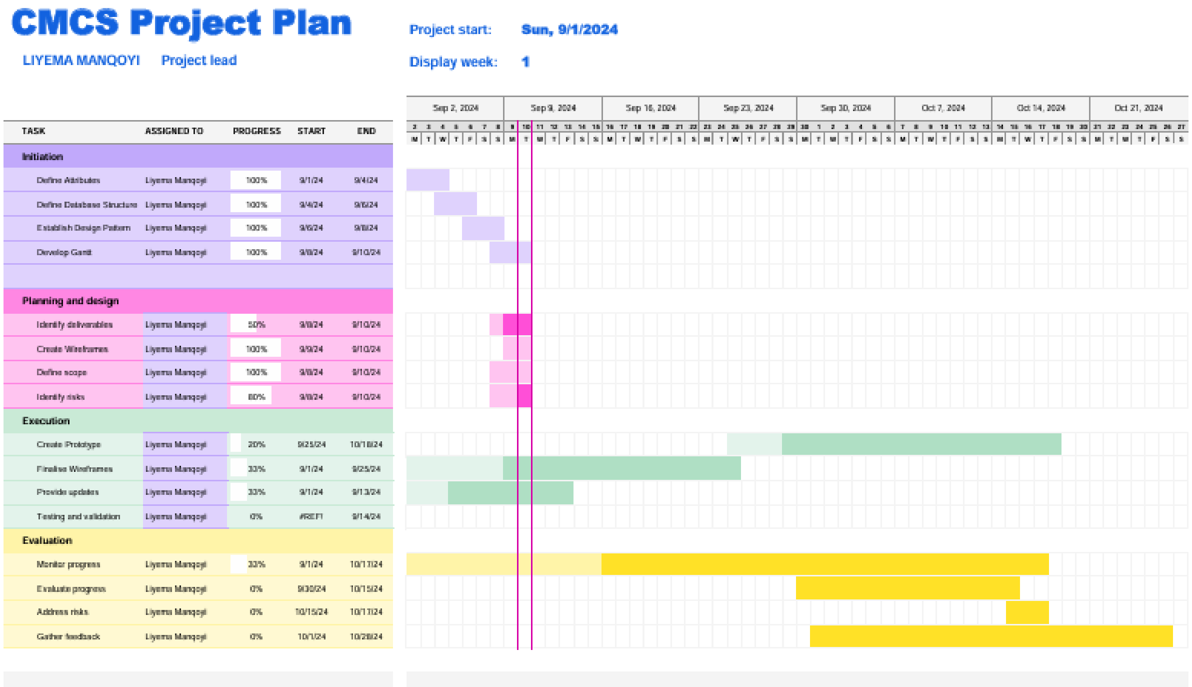
One of MVC's key downsides is that it complicates the application architecture when compared to simpler patterns. This can make it more difficult to understand and manage, particularly for smaller enterprises or those with fewer resources.

By leveraging all the benefits listed above, one can rule that developing large scale applications it would be best to use a MVC architectural pattern.

UML CLASS DIAGRAM



PROJECT PLAN



CMCS Project Plan

LIYEMA MANQOYI Project lead

Project start: **Sun, 9/1/2024**

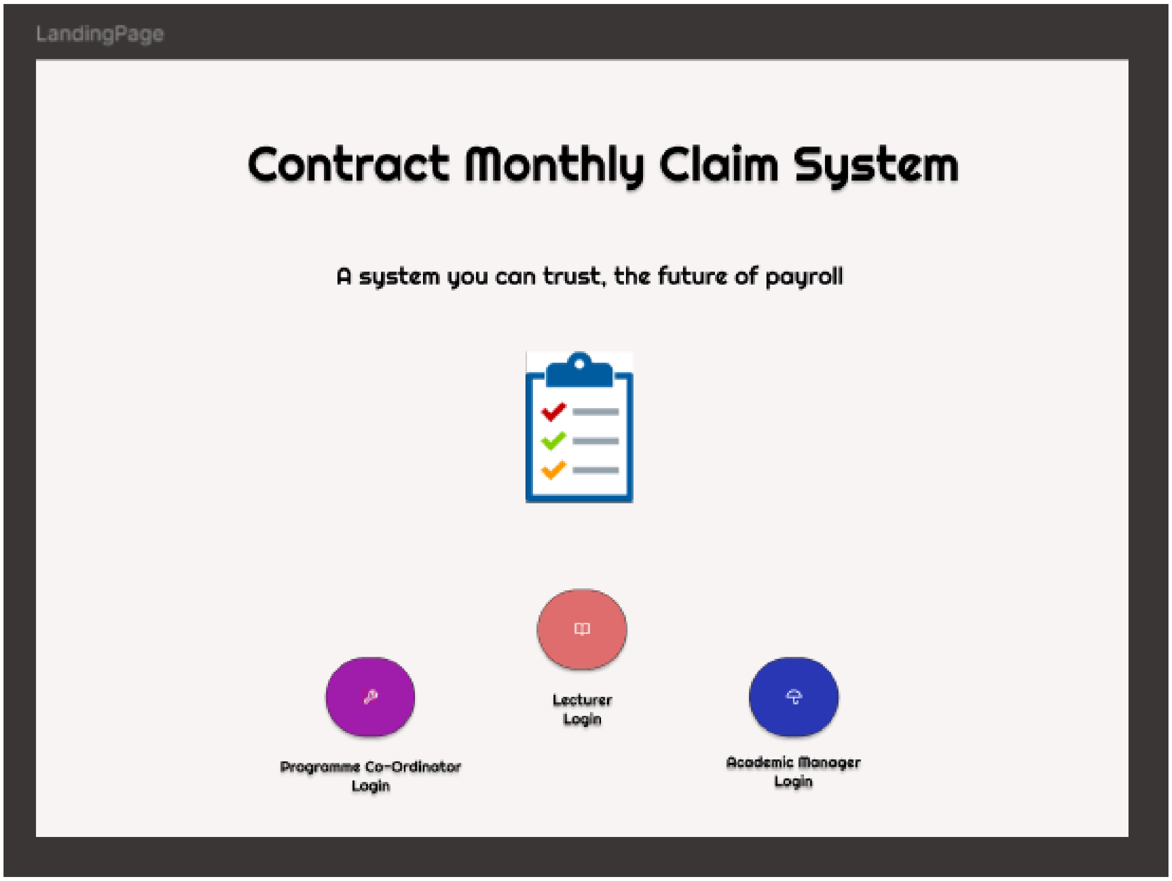
Display week: **1**

TASK	ASSIGNED TO	PROGRESS	START	END
Initiation				
Define Attributes	Liyema Manqoyi	100%	9/1/24	9/4/24
Define Database Structure	Liyema Manqoyi	100%	9/4/24	9/6/24
Establish Design Pattern	Liyema Manqoyi	100%	9/6/24	9/8/24
Develop Gantt	Liyema Manqoyi	100%	9/8/24	9/10/24
Planning and design				
Identify deliverables	Liyema Manqoyi	50%	9/8/24	9/10/24
Create Wireframes	Liyema Manqoyi	100%	9/9/24	9/10/24
Define scope	Liyema Manqoyi	100%	9/8/24	9/10/24
Identify risks	Liyema Manqoyi	80%	9/8/24	9/10/24
Execution				
Create Prototype	Liyema Manqoyi	20%	9/25/24	10/18/24
Finalise Wireframes	Liyema Manqoyi	33%	9/1/24	9/25/24
Provide updates	Liyema Manqoyi	33%	9/1/24	9/13/24
Testing and validation	Liyema Manqoyi	0%	#REF!	9/14/24
Evaluation				
Monitor progress	Liyema Manqoyi	33%	9/1/24	10/17/24
Evaluate progress	Liyema Manqoyi	0%	9/30/24	10/15/24
Address risks	Liyema Manqoyi	0%	10/15/24	10/17/24
Gather feedback	Liyema Manqoyi	0%	10/1/24	10/28/24

The Gantt chart is also available via the git-repo link, for better quality image.

GRAPHIC USER INTERFACE

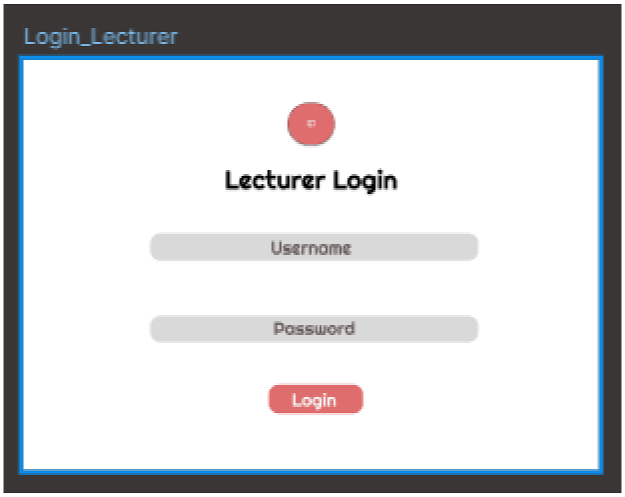
LANDING PAGE:

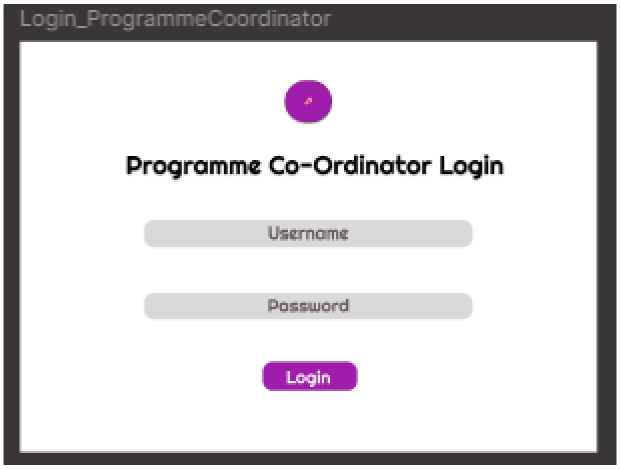


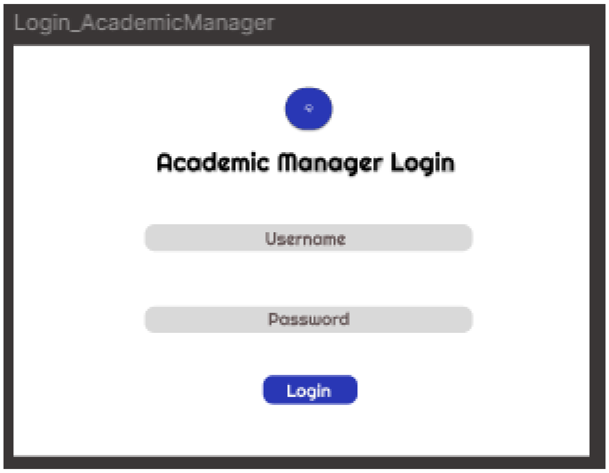
Contract Monthly Claim System

A system you can trust, the future of payroll

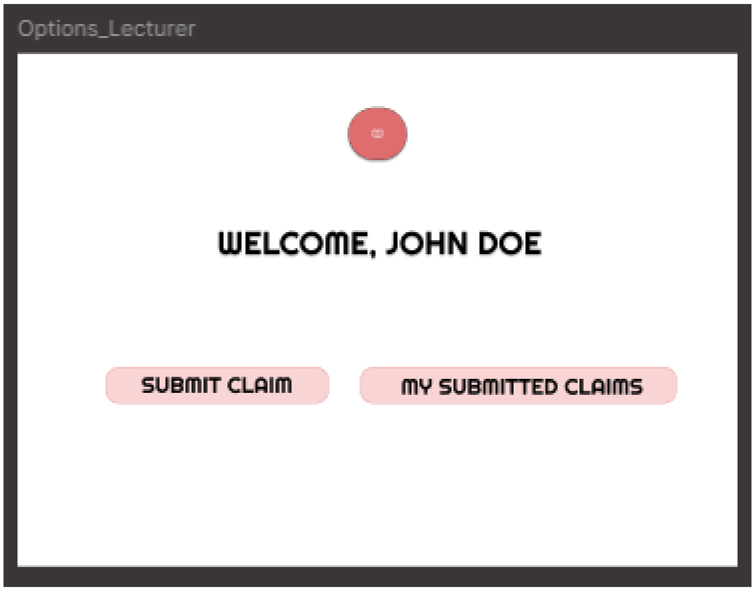
LOGIN(s)

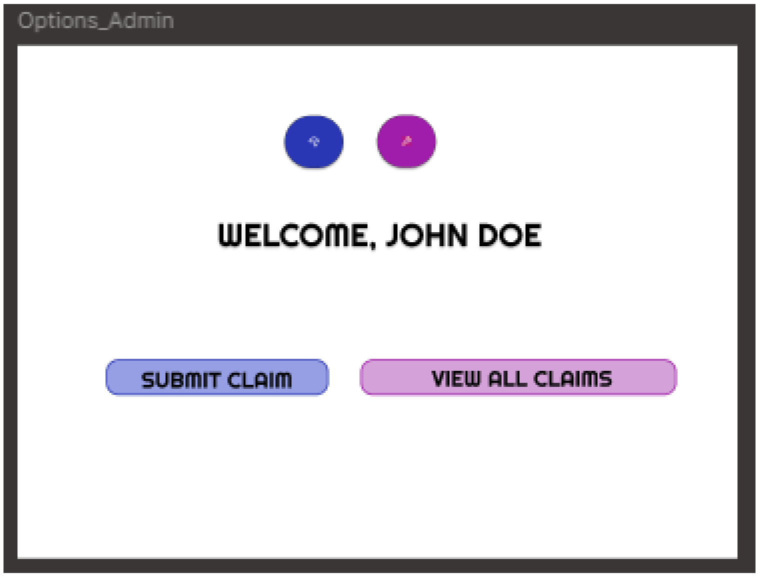




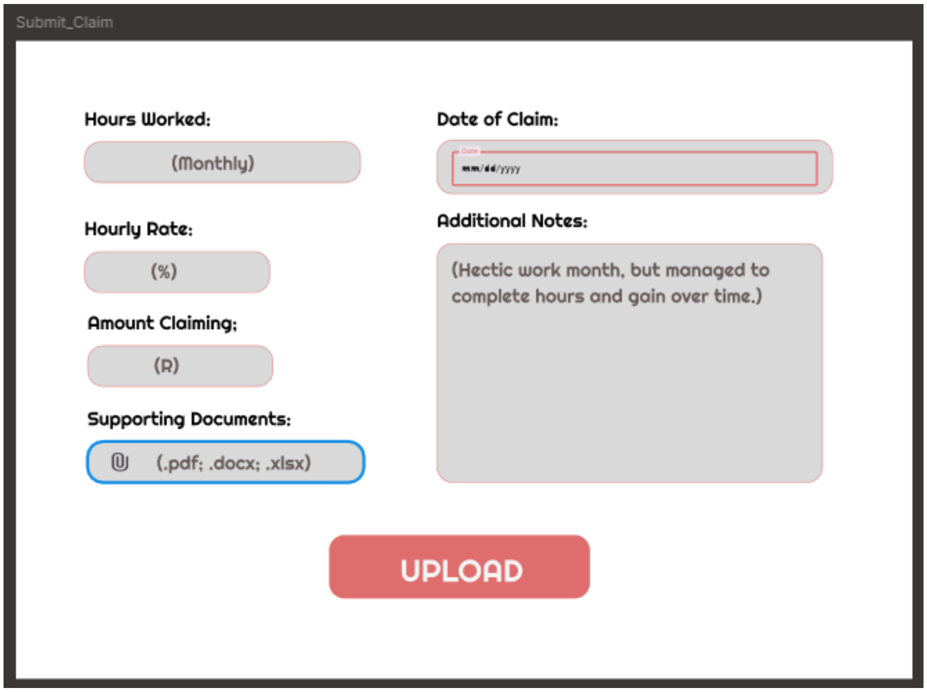


OPTIONS AFTER SUCCESSFUL LOGIN


Options_Lecturer
WELCOME, JOHN DOE
SUBMIT CLAIM
MY SUBMITTED CLAIMS


Options_Admin
WELCOME, JOHN DOE
SUBMIT CLAIM
VIEW ALL CLAIMS

LECTURER SUBMIT CLAIM



Submit_Claim

Hours Worked:

(Monthly)

Date of Claim:

Date

mm/dd/yyyy

Hourly Rate:

(%)

Additional Notes:

(Hectic work month, but managed to complete hours and gain over time.)

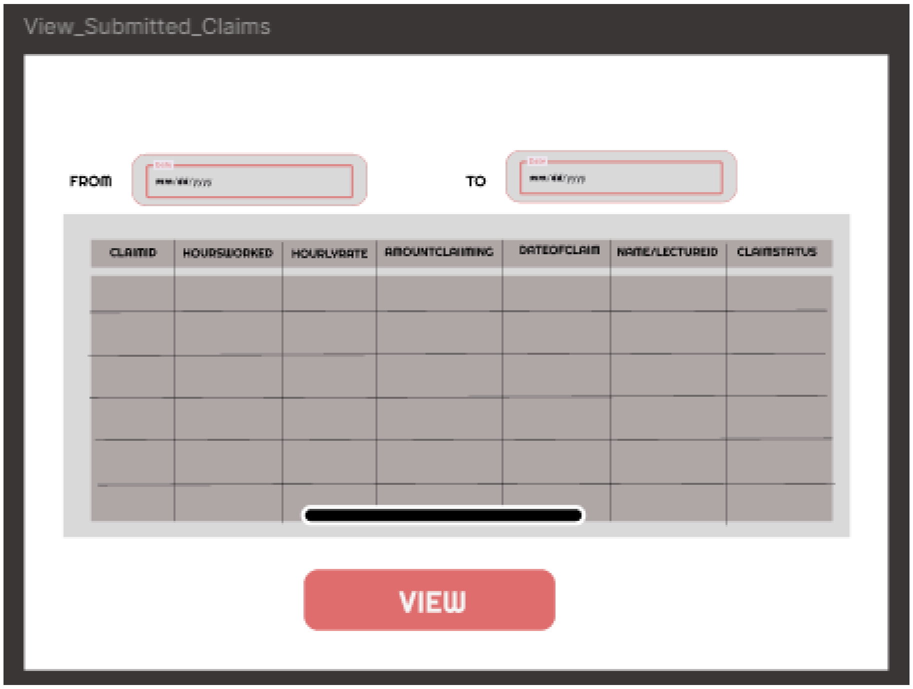
Amount Claiming:

(R)

Supporting Documents:

(.pdf; .docx; .xlsx)

LECTURER VIEW OWN SUBMITTED CLAIM



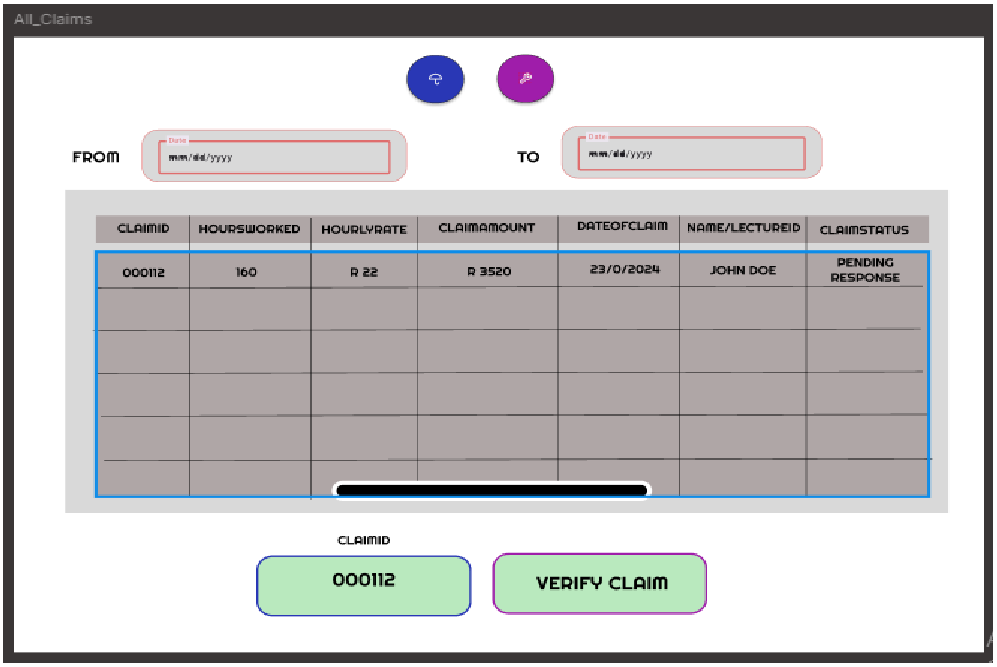
View_Submitted_Claims

FROM Date mm/dd/yyyy

TO Date mm/dd/yyyy

CLAIMID	HOURSWORKED	HOURLYRATE	AMOUNTCLAIMING	DATEOFCLAIM	NAME/LECTUREID	CLAIMSTATUS

PROGRAMME COORDINATOR & ACADEMIC MANAGER VIEW ALL SUBMISSIONS



FROM Date mm/dd/yyyy

TO Date mm/dd/yyyy

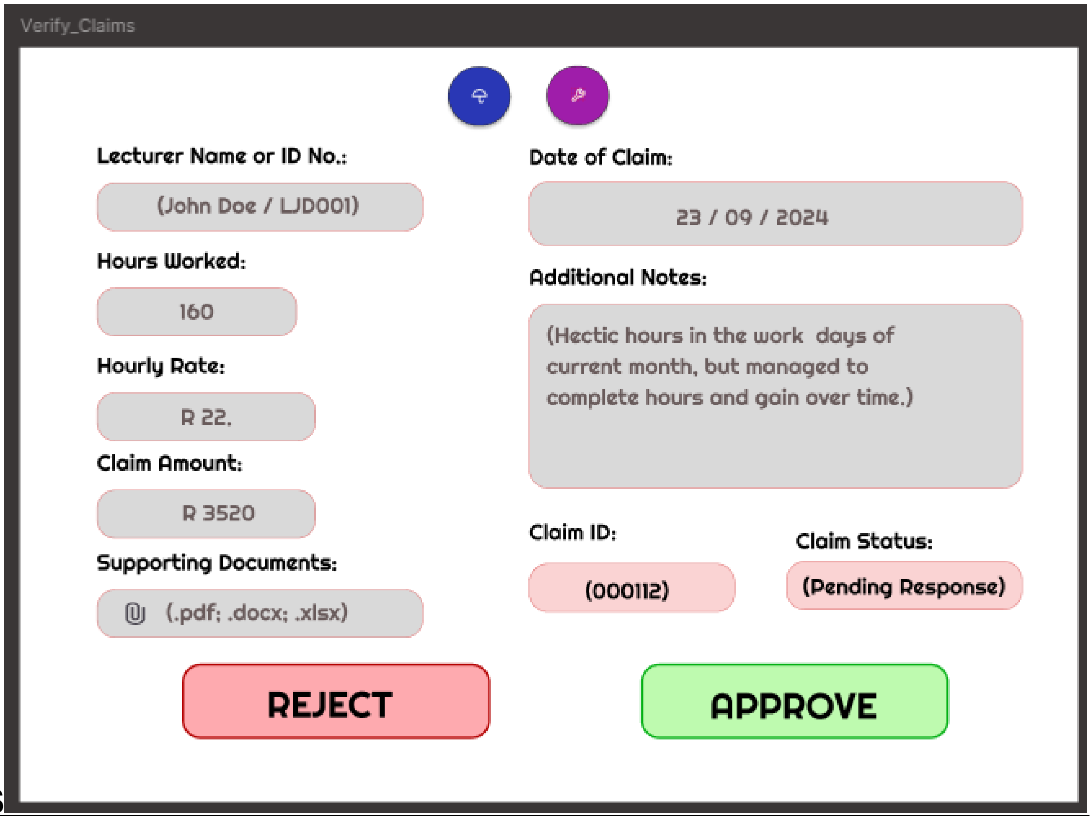
CLAIMID	HOURSWORKED	HOURLYRATE	CLAIMAMOUNT	DATEOFCLAIM	NAME/LECTUREID	CLAIMSTATUS
000112	160	R 22	R 3520	23/0/2024	JOHN DOE	PENDING RESPONSE

CLAIMID

000112

<u>VERIFY AND REVIEW</u>

CLAIMS



Lecturer Name or ID No.: (John Doe / LJD001)

Hours Worked: 160

Hourly Rate: R 22.

Claim Amount: R 3520

Supporting Documents: (.pdf; .docx; .xlsx)

Date of Claim: 23 / 09 / 2024

Additional Notes: (Hectic hours in the work days of current month, but managed to complete hours and gain over time.)

Claim ID: (000112)

Claim Status: (Pending Response)

REFERENCES

Anushka, V. (2021). *Benefit of using MVC*. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/benefit-of-using-mvc/ [Accessed 8 Sep. 2024].

Ardalis (2023). *Overview of ASP.NET Core MVC*. [online] Microsoft.com. Available at: https://learn.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-8.0#mvc-pattern [Accessed 8 Sep. 2024].

Kirinriki (2024). *In the Model-View-Controller principle, what is the Frontend and what is the Backend?* [online] Stack Overflow. Available at: https://stackoverflow.com/questions/11066958/in-the-model-view-controller-principle-what-is-the-frontend-andA-what-is-the-bac [Accessed 8 Sep. 2024].