PROG6212

POE PART 1

ST10048826

James Weston Knox

Contract Monthly Claim System

Contents

[Documentation 2](#_Toc176869934)

[Assumptions 2](#_Toc176869935)

[Constraints 2](#_Toc176869936)

[Design choices 2](#_Toc176869937)

[Database structure 3](#_Toc176869938)

[GUI Layout 4](#_Toc176869939)

[Navigation 4](#_Toc176869940)

[Landing page 4](#_Toc176869941)

[Lecturer dashboard 4](#_Toc176869942)

[Programme Coordinator / Academic Manager dashboard 4](#_Toc176869943)

[Account page 4](#_Toc176869944)

[UML Class Diagram 5](#_Toc176869945)

[Project Plan 5](#_Toc176869946)

[GUI 7](#_Toc176869947)

[Homepage 7](#_Toc176869948)

[Account page 7](#_Toc176869949)

[Lecturer dashboard 8](#_Toc176869950)

[Admin dashboard 8](#_Toc176869951)

[References 9](#_Toc176869952)

# Documentation

## Assumptions

1. We assume lecturers will input valid and accurate data.
2. We assume documents uploaded will adhere to common file types.

## Constraints

1. System needs to be designed to accommodate different roles and different access and features depending on the user’s role.
2. System needs to support all popular document types to ensure users can upload documents without error.
3. System needs to display reliable and consistent information.
4. System needs to provide efficient tracking system to track the status of claims in real time.

## Design choices

Architecture:

For the design of the project, I have chosen to use a Model View Controller (MVC) architecture. This design allows us to separate the logic of the project and allows for better scalability as we expand. (Microsoft, n.d.)

MVC allows us to handle the front end in the Views. The logic in the Models and the Controllers act as the connecter between the two.

MVC allows for easy integration with Microsoft azure should that be used for hosting and databases.

The application consists of 4 pages:

Home page – Welcome message and navigation buttons to other dashboards

Account page – Create account or login to existing account.

Lecturer dashboard – Submit claims and view and track current claims.

Programme Coordinator dashboard – Review claims and view past approved claims

Colour Scheme:

I chose a grey, white and blue colour scheme to keep a simple yet professional look for the application.

Navigation Menu - #415256

Primary buttons - #1588fc

Title in navbar - #1588fc

Reject buttons – red

Other headings and text – black

Background – white

## Database structure

The database will be a relational database to handle the complex relationships between different entities.

The database will consist of 5 entities:

1. Lecturer table

Stores all information relating to lectures and their accounts

1. Programme Coordinator table

Stores all information relating to program coordinator and their accounts

1. Claim table

Stores all information relating to claims submitted by lectures.

1. Documents table

Stores all information regarding uploaded documents and claims they are linked to.

1. Claim review table

Stores all information regarding the status of claims and whether they have been approved or rejected by the program coordinators.

Relationships:

1. Lecturers can submit multiple claims, but individual claims can only be submitted by 1 lecturer.
2. Lecturers can upload multiple documents, but individual documents can only be uploaded by 1 lecturer.
3. Programme coordinators can review and approve multiple claims, and individual claims can be reviewed by multiple Programme coordinators (many to many relationship).

## GUI Layout

### Navigation

The navigation consists of a fixed top navigation bar with the title of the application on the left and buttons too the different pages on the right. As this is only the front-end prototype, I have left both dashboards on display but with the implementation of the login feature only the dashboard associated with the account will be shown.

### Landing page

The landing page consists of a welcome message and a short paragraph explaining the purpose of the system. Below that are 3 sections.

1. Message and button requesting user to login. Button takes them to account page

2. Button linking to lecturer dashboard.

3. Button linking to programme coordinator dashboard.

### Lecturer dashboard

Lecturer dashboard consists of two sections

1. Form to submit claims with all required fields including document uploading.
2. Table to view current claims (Filled with sample data for prototype)

### Programme Coordinator / Academic Manager dashboard

Programme coordinator dashboard consists of two sections

1. Table showing claims that still need to be reviewed.
2. Table showing past approved claims.

### Account page

Account page consists of two forms

1. Form to create a new account.
2. Form to login to existing account.

# UML Class Diagram

# Project Plan

Phase 1 – Planning

1. Identify stakeholders and assumptions and constraints – 2 days
2. Identify different user roles - 1 day
3. Decide on project architecture – 2 days
4. Create UML diagram to represent database structure – 2 days

Phase 2 – Design of front-end prototype

1. Setup MVC project with views, and controllers for front end - 3 days
2. Design html for pages and link pages through navbar and controllers – 2 days
3. Decide on colour scheme and other design choices – 1 day
4. Implement CSS styling to views – 3 days
5. Test front-end prototype

Phase 3 – Development

1. Setup relational database – 3 days
2. Implement login feature - 3 days
3. Implement claim uploading feature – 2 days
4. Implement document uploading feature – 3 days
5. Implement claim approval feature – 2 days
6. Implement clear and transparent claim tracking – 4 days

Phase 4 – Testing

1. Implement unit testing for all features. – 2 days
2. Implement error handling – 3 days

# GUI

## Homepage

## A screenshot of a login screen Description automatically generatedAccount page

## A screenshot of a computer Description automatically generatedLecturer dashboard

## A screenshot of a computer Description automatically generatedAdmin dashboard

# References

Bootstrap, n.d. *CSS.* [Online]   
Available at: https://getbootstrap.com/docs/3.4/css/#tables  
[Accessed 9 September 2024].

GreyCampus, n.d. *Constraints and Assumptions.* [Online]   
Available at: https://www.greycampus.com/opencampus/project-management-professional/constraints-and-assumptions#:~:text=Constraints%3A%20A%20factor%20that%20limits,needs%20only%20MBA%20pass%20outs).  
[Accessed 9 September 2024].

Harned, B., n.d. *How to Create a Realistic Project Plan with Templates & Examples.* [Online]   
Available at: https://www.teamgantt.com/project-management-guide/how-to-plan-a-project  
[Accessed 8 September 2024].

Microsoft, 2024. *Views.* [Online]   
Available at: https://learn.microsoft.com/en-us/sql/relational-databases/views/views?view=sql-server-ver16  
[Accessed 5 September 2024].

WordPress, 2022. *Best Website Color Schemes for Modern Web Design.* [Online]   
Available at: https://happyaddons.com/best-website-color-schemes-for-web-design/  
[Accessed 8 September 2024].