

***Name:***

***Musasa Ncukana***

***Student Number:***

***St10448582***

***Course: PROG6212***

Table of Contents

[Introduction 3](#_Toc208306149)

[My Design Choice 3](#_Toc208306150)

[Assumptions 4](#_Toc208306151)

[Constraints and Limitations 4](#_Toc208306152)

[UML Diagram 5](#_Toc208306153)

## Introduction

The aim of this report is to show the design rationale, UML database class diagram, and project plan for prototyping the Contract Monthly Claim System (CMCS). The CMCS is to be utilized by departments to distribute, collect and approve monthly claims from their IC lecturers.

This report discusses the design decisions of the GUI and database, clarifies the assumptions and restrictions that were made and presents a project plan of the actual implementation of the non-functional prototype which uses WPF and MVVM structure.

## My Design Choice

The CMCS project uses a WPF MVVM-style GUI structure, organized as follows:

App.xaml, App.xaml.cs

LoginWindow.xaml,LoginWindow.xaml.cs

MainWindow.xaml, MainWindow.xaml.cs

Views Folder

DashboardView.xaml, DashboardView.xaml.cs

SubmitClaimView.xaml, SubmitClaimView.xaml.cs

UploadDocsView.xaml, UploadDocsView.xaml.cs

ApprovalsView.xaml, ApprovalsView.xaml.cs

StatusTrackerView.xaml, StatusTrackerView.xaml.cs

App.xaml saves application initialization code and global resources.

LoginWindow serves as the entry point to the operation, prompting users to log in before they can access the system.

The MainWindow works to produce a comfortable host for our UIL Shell, bringing things like navigation, views and even the command bar down into one place.

The views directory contains modular feature screens which are both independent and reusable. Here you'll find the Dashboard, Submit Claim, Upload Documents, Approvals and Status Tracker screens.

This modular development language solves main problems of maintainability and scalability. Each view can be linked to a ViewModel for responsible control logic or such. In this way project development naturally keeps room for growth.

The Database Structure

The database schema is tied together with a number of normalized tables. Additions, deletions or other changes to this design will require a lot of effort later. "Users," "Roles," "Claims," "Period Claims," "Approvals," "Applications" and " Rates" are just some elements in the Database Structure.

Splitting Claims into Claim header and Claim Lines allows multiple activities per claim.

Approval records reflect review decisions by Programme Coordinators and Academic Managers.

Documents link supporting evidence to claims.

Rates allow an update in hourly compensation rates without affecting work that came before.

This design is the result of trying to strike a balance between reality and the need for simplicity which a prototype imposes.

## Assumptions

teachers are already registered as system users, and Programme Coordinators (PCs) as well as Academic Managers (AMs) are two different roles essential for claim approvals. The Rate table holds hourly rates constant within a semester. Each lecturer submits a claim each month with multiple claim lines. Documents are limited for upload to a maximum size of 10MB and must be in PDF or image file formats only. Finally, the Part 1 prototype only illustrates navigation and interface design, with no real database functions yet.

## Constraints and Limitations

Its authentication is hard coded into non-functional concepts/placeholder data for now. The project deadline is six weeks away, focus in these six weeks should solely on claim and approval workflows. Payroll integration HR records detailed external reporting: these features will not appear in MRS's Part 1 and are out of scope. Furthermore, development is limited to WPF (.NET), or ASP.NET MVC, as per module requirements.

## UML Diagram

**User**

UserID

UserName

PassWord

1..1

1..1

1..1

1..1

**Lecturer**

LecturerID

Name

1..\*

**Approval**

ApprovalID

Decision

Comment

1..\*

1..1

1..\*

1..\*

**Claim**

ClaimID

Month

Status

1..\*

1..1

1..\*

1..\*

**ClaimLine**

ClaimLineID

WorkDate

Activity

**Documentation**

DocumentationID

FileName

FilePath

## Project Plan

