

25/09/2017

PROGRAMMING 2B (PROG6212)

POE – PART 1



ST10445050
KEONA MACKAN

1) Documentation:

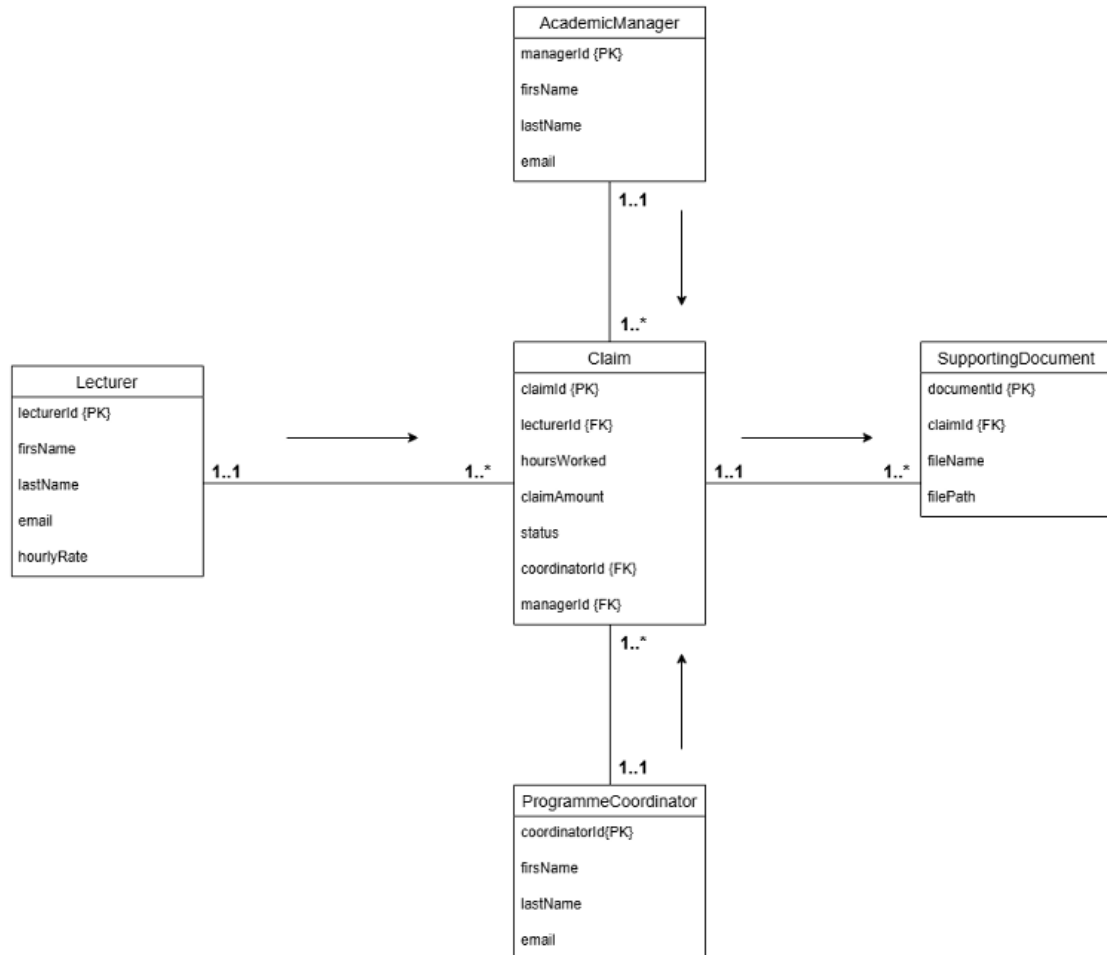
After careful planning and reflection, I decided that the best approach for my Contract Monthly Claim System (CMCS) is to build a clean, professional, and user-focused web-based application using C# and .NET Core. In designing this system, I prioritized usability, clarity, and efficiency: I wanted lecturers to easily submit claims, view their status, and optionally upload supporting documents, while coordinators and academic managers could efficiently review and either approve or reject them (Martin Martin, 2023).

To create a professional and trustworthy interface, I chose a palette of green tones: deep forest green for headers, muted medium green for accents, soft light green for buttons, and very light green for backgrounds (Martin Martin, 2023). I felt this consistent green theme conveys calmness, professionalism, and clarity, and I added hover effects and subtle shadows on buttons to improve interactivity and provide clear feedback to users.

For the database, I structured it to reflect real-world administrative workflows. I created five main classes: User, Lecturer, Claim, SupportingDocument, and Coordinator/Academic Manager, each with carefully defined primary keys (PK) and foreign keys (FK) to maintain data integrity. I linked claims to lecturers through a foreign key to ensure proper attribution. Both Programme Coordinators and Academic Managers are also linked to claims via foreign keys, allowing them to view, review, and manage all claims. Additionally, each claim can have multiple supporting documents, linked via a foreign key to ensure proper association with the corresponding claim (Martin Martin, 2023). I designed this relational structure with scalability in mind, so it could easily support multiple roles per user or additional reporting features in the future.

To ensure smooth operations, I implemented referential integrity so that every claim is always associated with a valid lecturer and that approvals or rejections are traceable to the responsible coordinator or manager. This design minimizes data redundancy, allows for accurate reporting, and ensures that approved claims persist even if the application restarts. Supporting documents are linked to claims to maintain a clear one-to-many relationship while keeping files encrypted for security. The database structure also enables easy retrieval of claims based on status, user, or date, supporting the dashboard functionality efficiently. Overall, the design balances usability, security, and maintainability, supporting both current functionality and future expansion (Martin Martin, 2023).

2) UML Class Diagram:



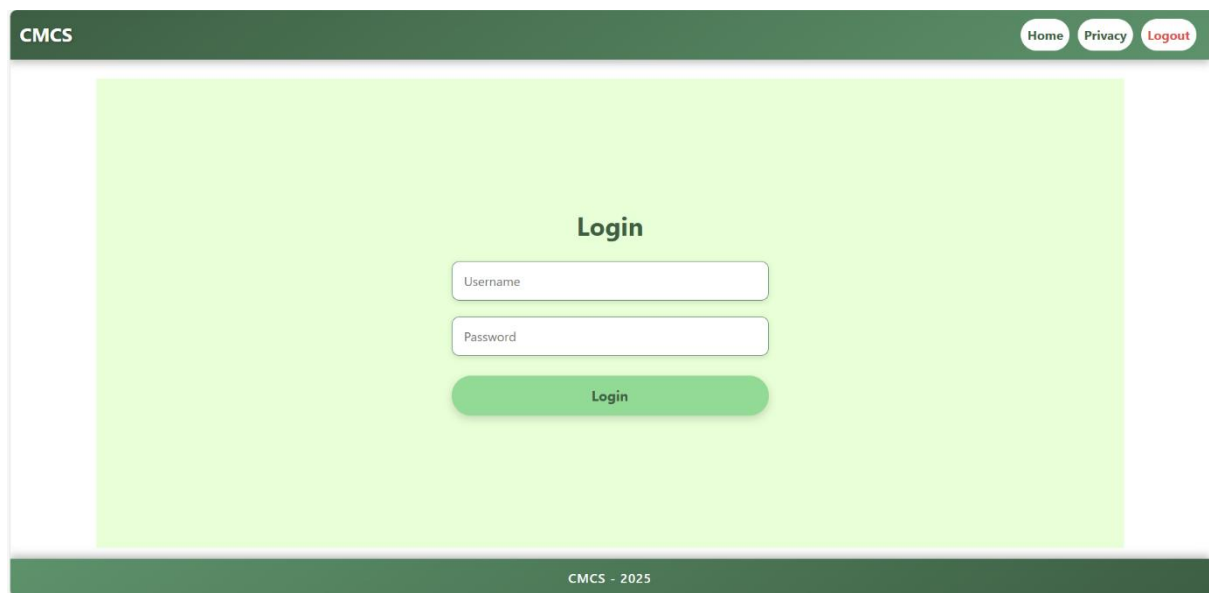
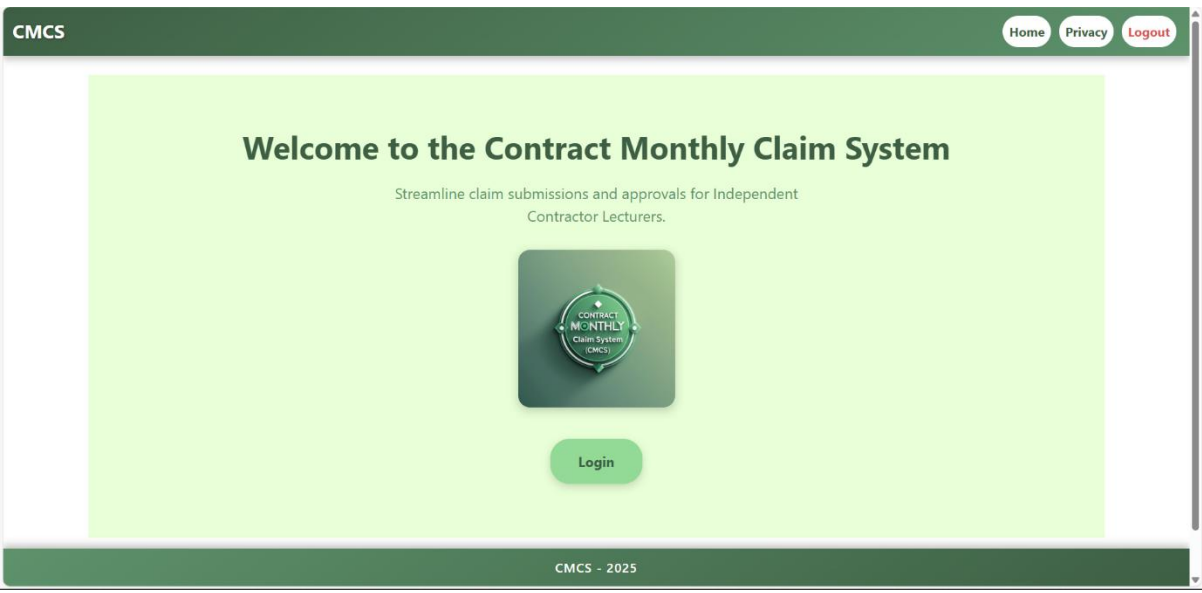
(draw.io, 2025).

3)Project Plan:

	①	WBS	Task Mode	Task Name	Duration	Start	Finish	Deliverables	Dependencies
1		0		CMCS Project	48 days	Wed 25/09/17	Fri 25/11/21	Project Initiation	-
2		1		Project Initiation & Planning	3 days	Wed 25/09/17	Fri 25/09/19	Approved project scope & sched	-
3		1.1		Define objectives & scope	1 day	Wed 25/09/17	Wed 25/09/17	CMCS scope document	-
4		1.2		Develop project schedule	2 days	Thu 25/09/18	Fri 25/09/19	MS Project plan	1.1
5		2		Requirements Gathering & Analysis	5 days	Mon 25/09/22	Fri 25/09/26	Requirements specification	1.2
6		2.1		Identify user roles	2 days	Mon 25/09/22	Tue 25/09/23	Role matrix (Lecturer,Coordinator	1.2
7		2.2		Define claim workflow	2 days	Wed 25/09/24	Thu 25/09/25	Process map (submit → review →	2.1
8		2.3		Confirm document upload requirements	1 day	Fri 25/09/26	Fri 25/09/26	Document tracking & upload spec	2.2
9		3		System & Database Design	7 days	Mon 25/09/29	Tue 25/10/07	ERD, UI wireframes, workflow dia	2.3
10		3.1		Design database schema	3 days	Mon 25/09/29	Wed 25/10/01	Tables: Lecturers,Claims,Documen	2.3
11		3.2		Create UI mockups	3 days	Thu 25/10/02	Mon 25/10/06	Home,Claim Submission,Approval	3.1
12		3.3		Document system workflows	1 day	Tue 25/10/07	Tue 25/10/07	UML activity diagrams	3.2
13		4		Front-End Prototype (GUI Only	10 days	Wed 25/10/08	Tue 25/10/21	Non-functional MV GUI	3.2
14		4.1		Lecturer submission interface	4 days	Wed 25/10/08	Mon 25/10/13	Submit claims form (hours,rate,do	3.2
15		4.2		Claim tracking page	3 days	Tue 25/10/14	Thu 25/10/16	Status table with badges	4.1
16		4.3		Coordinator/Manager approv	3 days	Fri 25/10/17	Tue 25/10/21	Approvals	4.2

	①	WBS	Task Mode	Task Name	Duration	Start	Finish	Deliverables	Dependencies
16		4.3		Coordinator/Manager approv	3 days	Fri 25/10/17	Tue 25/10/21	Approvals	4.2
17		5		Backend Setup & Integration (Part 2 – later)	7 days	Wed 25/10/22	Thu 25/10/30	Entity models & relations	3.1, 4.1
18		5.1		Implement Lecturer model	2 days	Wed 25/10/22	Thu 25/10/23	Lecturer entity with details	3.1
19		5.2		Implement Claim model	3 days	Fri 25/10/24	Tue 25/10/28	Claim auto-creation logic	5.1
20		5.3		Implement Document & Role models	2 days	Wed 25/10/29	Thu 25/10/30	Document upload + role entity	5.2
21		6		Documentation & Reporting	5 days	Fri 25/10/31	Thu 25/11/06	Draft PoE submission	3.3, 4.3, 5.3
22		6.1		System design documentation	2 days	Fri 25/10/31	Mon 25/11/03	ERD,UML,mockups	3.3, 5.3
23		6.2		Prototype documentation	2 days	Tue 25/11/04	Wed 25/11/05	Screens & workflow notes	4.3
24		6.3		Compile PoE report	1 day	Thu 25/11/06	Thu 25/11/06	Draft report ready	6.1, 6.2
25		7		Testing & Review	10 days	Fri 25/11/07	Thu 25/11/20	Usability & GUI validation, final a	6.3
26		7.1		Usability testing	3 days	Fri 25/11/07	Tue 25/11/11	Feedback on GUI design	4.3
27		7.2		Functional validation	4 days	Wed 25/11/12	Mon 25/11/17	Verify claims,Approvals,document	7.1
28		7.3		Final review & sign-off	3 days	Tue 25/11/18	Thu 25/11/20	Testing completion report	7.2
29		8		Project Closure	1 day	Fri 25/11/21	Fri 25/11/21	Final submission	7.3

4)GUI/UI Design:



CMCS

Home

Privacy

Logout

Lecturer Dashboard

Submit New Claim

ID	Lecturer Name	Hours Worked	Hourly Rate	Claim Amount	Status	Supporting Document
----	---------------	--------------	-------------	--------------	--------	---------------------

CMCS - 2025

Submit a New Claim

Lecturer ID

Enter your ID

Lecturer Name

Enter your name

Hours Worked

Hours Worked

Hourly Rate

Hourly Rate

Claim Amount

☐ Upload Supporting Document?

Submit Claim

Submit a New Claim

Lecturer ID

Enter your ID

Lecturer Name

Enter your name

Hours Worked

Hours Worked

Hourly Rate

Hourly Rate

Claim Amount

☒ Upload Supporting Document?

Supporting Document

Choose File

No file chosen

Submit Claim

CMCS

HomePrivacyLogout

Coordinator Dashboard

All Submitted Claims

ID	Lecturer	Hours Worked	Hourly Rate	Claim Amount	Status	Supporting Doc	Action
----	----------	--------------	-------------	--------------	--------	----------------	--------

CMCS - 2025

GitHub Repo Link: <https://github.com/VCWVL/prog6212-part-1-ST10445050/tree/master>

Proof of Submission:

VCWL / prog6212-part-1-ST10445050

Type to search

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

prog6212-part-1-ST10445050 Private

Watch 0 Fork 0 Star 0

master had recent pushes 6 minutes ago

Compare & pull request

master 2 Branches 0 Tags

Go to file

Add file

<> Code

This branch is 7 commits ahead of, 1 commit behind main.

Contribute

ST10445050 Commit 5: Added in Developer Details and Reference List b3c1829 · 6 minutes ago 7 Commits

PROG6212_POE_PART1_ST10445050 Commit 5: Added in Developer Details and Reference List 6 minutes ago

.gitattributes Add .gitattributes and .gitignore. 36 minutes ago

.gitignore Add .gitattributes and .gitignore. 36 minutes ago

PROG6212_POE_PART1_ST10445050.sln Add project files. 36 minutes ago

README

About

prog6212-part-1-ST10445050 created by GitHub Classroom

Activity

Custom properties

0 stars

0 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Commits

master

All users All time

Commits on Sep 17, 2025

Commit 5: Added in Developer Details and Reference List

ST10445050 committed 7 minutes ago b3c1829

Commit 4: Added in views and css styling

ST10445050 committed 11 minutes ago 4929ede

Commit 3: Added in controller logic which is aligned with models

ST10445050 committed 24 minutes ago b88be25

Commit 2: Created all necessary models and added properties for each

ST10445050 committed 29 minutes ago 4b46c30

Initial Commit: Added program logic and structure

ST10445050 committed 34 minutes ago 7fa62fa

Add project files.

ST10445050 committed 37 minutes ago c4eaeab

Add .gitattributes and .gitignore.

ST10445050 committed 37 minutes ago 633f785

REFERENCE LIST:

draw.io. 2025. *Flowchart Maker & Online Diagram Software*. [online] app.diagrams.net. Available at: <https://app.diagrams.net/>.

[Accessed 15 September 2025].

Martin Martin. 2023. *5 Min Tutorial (ASP.NET Core 7) CSS style sheet for Razor page*. [online] YouTube. Available at: <https://www.youtube.com/watch?v=v6mf5H6KWak>

[Accessed 16 September 2025].

Studocu. 2024. *[Solved] Design a UML class diagram that accurately represents the data - Programming 2B - Studocu*. [online] Available at: <https://www.studocu.com/en-za/messages/question/8319394/design-a-uml-class-diagram-that-accurately-represents-the-data-requirements-of-the-contractmonthly>

[Accessed 14 September 2025].