

POE Part 1

PROG6212



September 10, 2024

ST10393280

Daniel Luke James

Table of Contents

[Question 1 2](#_Toc176893474)

[***Documentation*** 2](#_Toc176893475)

[Question 2 5](#_Toc176893476)

[***UML Class Diagram*** 5](#_Toc176893477)

[Question 3 6](#_Toc176893478)

[***Project Plan*** 6](#_Toc176893479)

[Question 4 8](#_Toc176893480)

[***GUI UI*** 8](#_Toc176893481)

[Question 5 17](#_Toc176893482)

[***GitHub Version Control*** 17](#_Toc176893483)

[17](#_Toc176893484)

[Reference List 18](#_Toc176893485)

# Question 1

### ***Documentation***

**My Design Choices:**

The purpose of the Contract Monthly Claim System (CMCS) is to expedite the submission and approval of monthly claims for lecturers who are independent contractors (ICs). The main goal of the design is to give all users, lecturers, program coordinators, academic managers, and administrators with a seamless and intuitive experience.

* **Scalability & Maintainability:** The.NET Core MVC architecture used in the system's construction encourages the division of responsibilities and makes scaling and maintenance simple. By using razor pages, it enhances the user interface, which makes the coding tidy and front end manageable to use.
* **Security:** By using this design choice of ASP.NET framework, this can guarantee that only those who are authorized can access particular functions, identification and authorization of users are essential components of the system.
* **User Experience:** With a clear, contemporary UI that is responsive and easy to use, the system places a strong emphasis on usability. In order to ensure that the program is accessible on a variety of devices, responsive design makes use of bootstrap.
* **Database Structure:** All pertinent data pertaining to claims, users, duties, and documents is intended to be stored in the database. It has tables for users, roles, claims, claim statuses, and uploaded documents, all of which are organized to facilitate effective data retrieval and management.

**My Database Structure:**

* **Roles:** Describes the various roles users may play within the system for example a role for the coordinator will be accepting and rejecting the claim itself.
* **Users:** Maintains track of the user information for example as role of the lecturer’s or manager’s, email address, password, and username.
* **Claims:** Maintains a record of the information provided by lecturers regarding their claims, for example amount, hours worked, and progress.
* **ClaimStatuses:** Monitors each claim's current state for example submitted, under review, approved or rejected, etc.
* **LecturerProfiles:** Provides details on lecturers for example the department, hourly rate, name, surname etc.
* **Documents:** Holds metadata pertaining to files that instructors have uploaded to bolster their arguments for example simple contracts with information about the claim.

**My GUI Layout:**

* **Login Page:** Allowing the users to authenticate themselves and access their dashboards when logging into the system by entering their username and password.
* **Profile Page:** Users can manage their accounts and change their personal information on the profile page by allowing them to update their email address, address, phone number etc.
* **Admin Dashboard:** Provides administrators only to have all-inclusive interface for handling system settings, claims for their clients/users etc.
* **Lecturer Dashboard:** Enables instructors to upload supporting materials, submit claims, and examine claim histories.
* **Coordinator/Manager Dashboard:** Will offer resources for managing lecturers and reviewing and approving claims.

(payapps, 2024).

(freshworks, 2024).

(Monday, 2024).

**Assumptions and Constraints –**

**My Assumptions:**

* Users do not need much training to use the system because they are accustomed to web-based interfaces in a basic way.
* In order to handle payments automatically, the entire system is going to interact with an already-existing payroll system.
* Many individuals will be using the system at once, therefore responsive user interface design and effective database management are essential.

**My Constraints:**

* The program needs to work with several browsers and adjust to varied screen sizes.
* Future scalability should be considered in the system's architecture to enable the inclusion of novel capabilities with the least amount of disturbance.
* Data protection and confidentiality, particularly with regard to user data and claim details, must be guaranteed by the system.

(long international, 2024).

# Question 2

### ***UML Class Diagram***

**A diagram of a computer

Description automatically generated with medium confidenceFor Databases:**

(visual paradigm, 2024).

# Question 3

### ***Project Plan***

The main objectives, dependencies, and schedule for creating the CMCS prototype are described in the project plan. The project will be accomplished on schedule and within scope thanks to the realistic and attainable design of the plan.

**Stage 1: Strategy and Analysis (Week 1)**

* **Task 1:** Gather and record all requirements from interested parties.

**Time frame:** 2 days

**Dependencies**: Research and analyse system requirements.

* **Task 2:** Create the MVC structure and database schema as well as the system architecture.

**Time frame:** 3 days

**Dependencies**: Finalization of system requirements.

**Stage 2: Front-end Development (Week 2-3):**

* **Task 3:** Construct the login page, by developing a safe login page so that users may authenticate themselves.

**Time frame:** 2 days

**Dependencies:** System design completion of front end for the login page.

* **Task 4:** Construct a profile page by where people can edit the details on their profiles.
* **Time frame:** 2 days

**Dependencies:** System design completion of front end for the profile page.

* **Task 5:** Construct an admin dashboard by developing a so they may control users and system configurations.

**Time frame:** 3 days

**Dependencies:** System design completion of front end for the admin dashboard.

* **Task 6:** Construct a lecturer dashboard by developing a dashboard so instructors can keep track of their claims.

**Time frame:** 3 days

**Dependencies:** System design completion of front end for the lecturer dashboard.

* **Task 7:** Construct a coordinator/manager dashboard by developing a dashboard so managers and coordinators can examine claims.

**Time frame:** 3 days

**Dependencies:** System design completion of front end for the coordinator/manager dashboard.

**Stage 3: Testing and Improvement (Beginning of Week 4):**

* **Task 8:** Evaluate the responsiveness and usability of the user interface.

**Time frame:** 1 day

**Dependencies:** Completion of front-end developing.

* **Task 9:** Enhancement and error correction by fixing any problems that arose during testing while implementing any necessary UI improvements.

**Time frame:** 2 days

**Dependencies:** Completion of the user interface.

**Stage 4: Final Evaluation and Submission (End of Week 4):**

* **Task 10:** Comprehensive evaluation by verifying that the prototype satisfies all requirements by doing a final review.

**Time frame:** 1 day

**Dependencies:** Completion of improvement and error correction.

* **Task 11:** Submission of the project by delivering the prototype and project documentation.

**Time frame:** 1 day

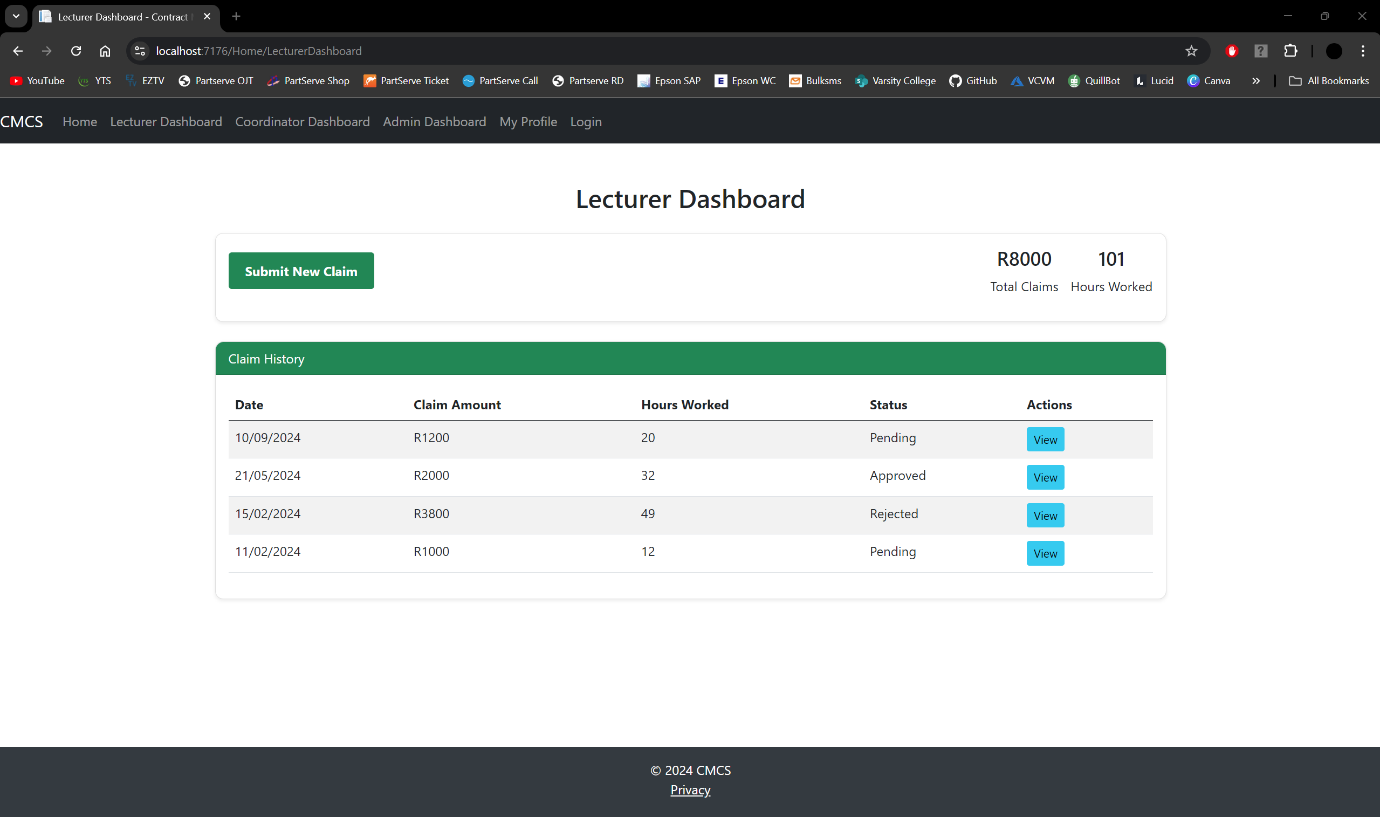
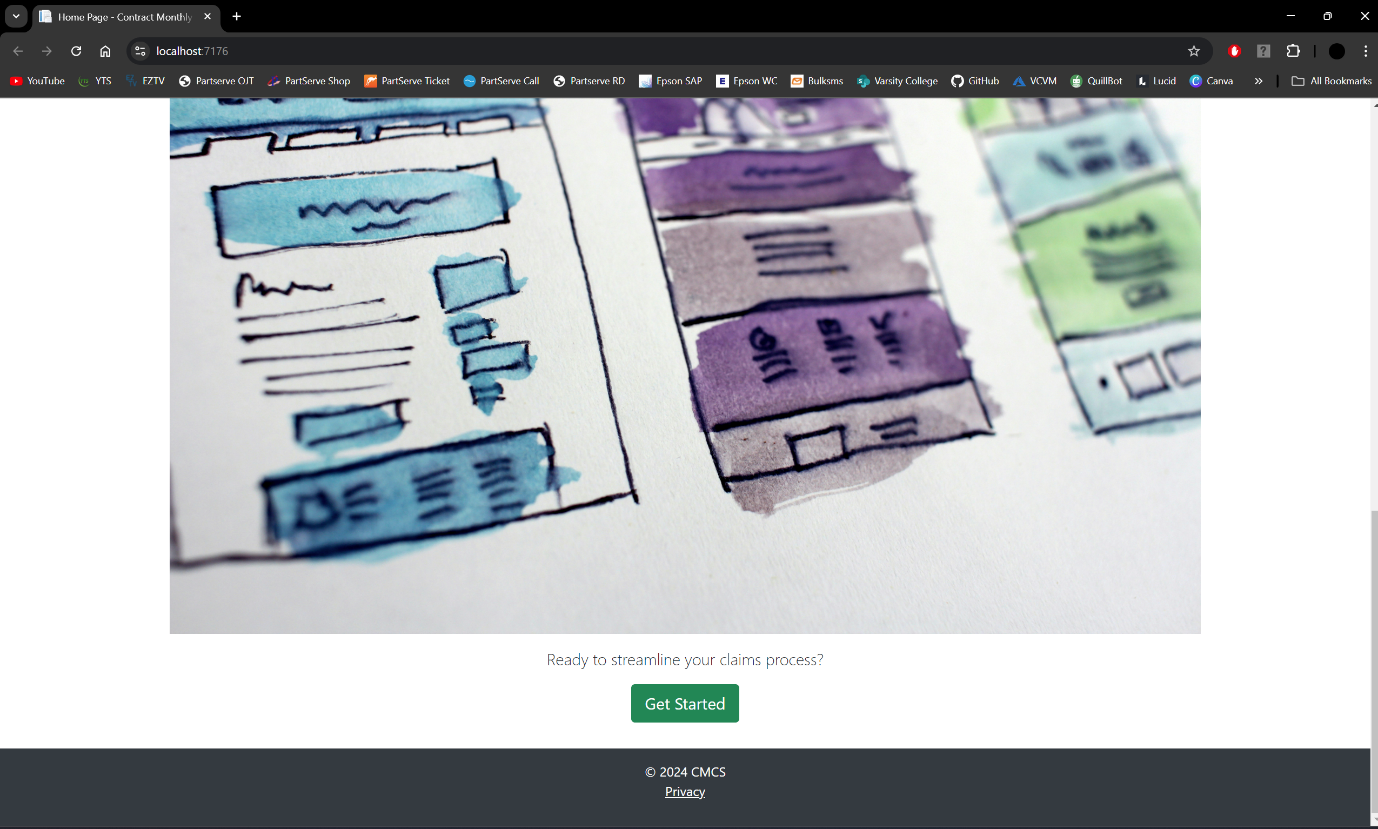
**Dependencies:** Final review completion and submission.

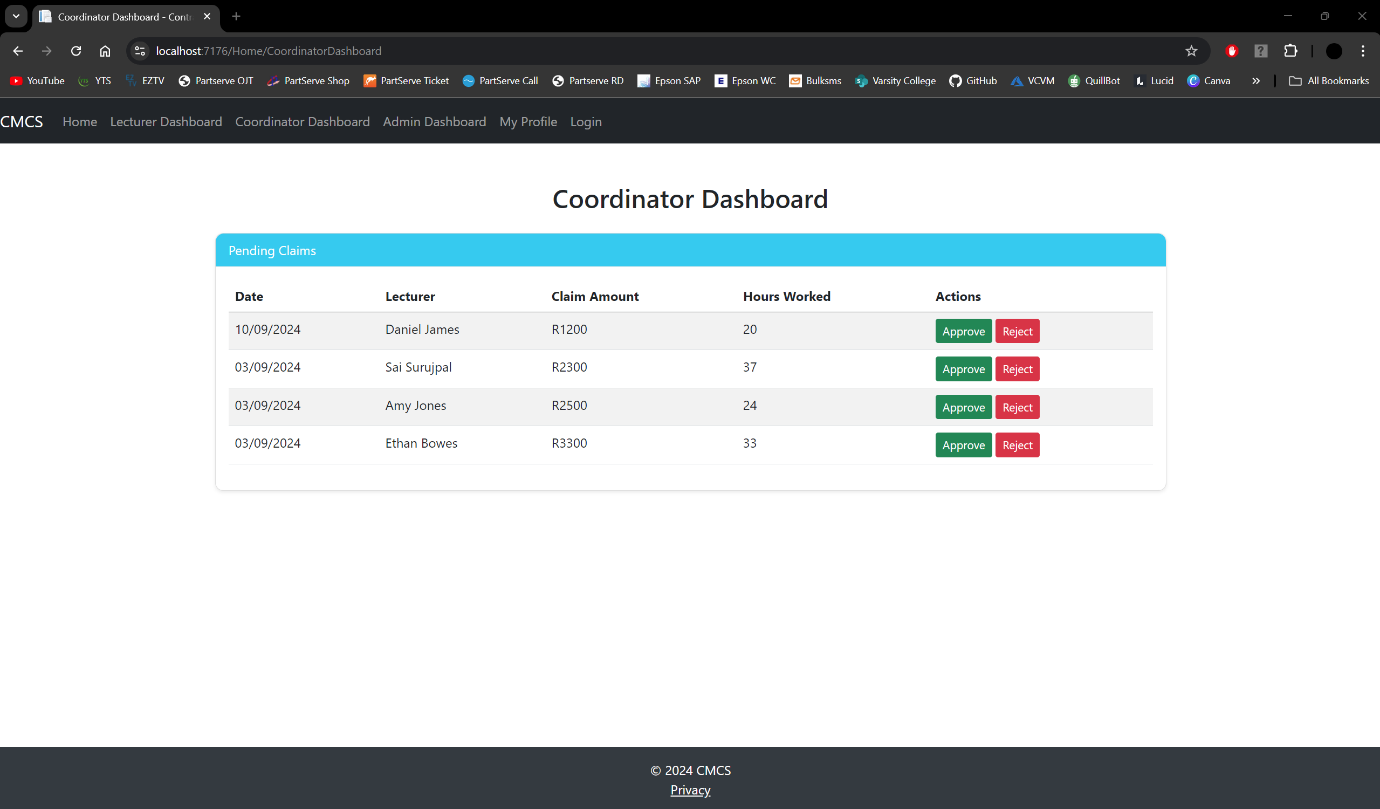
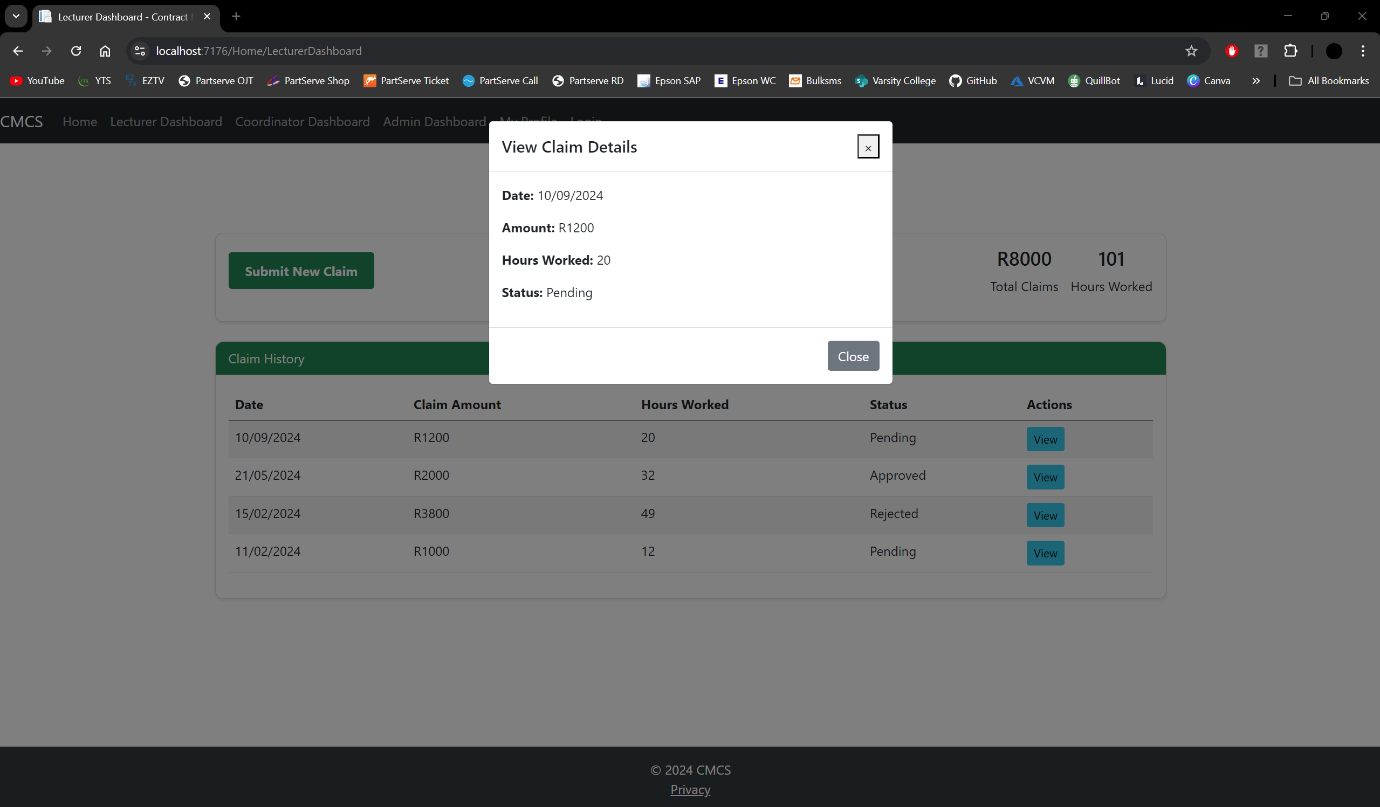
(asana, 2024).

(Monday, 2024).

# Question 4

### ***GUI UI***

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

# Question 5

### ***GitHub Version Control***

# A screenshot of a computer Description automatically generated

# Reference List

* payapps. 2024. Helping subcontractors, 25 June 2024.

[Online] Available at:

<https://www.payapps.com/subcontractors/>

[Accessed on 4 September 2024].

* monday, 2024. Your go-to work platform, 2024.

[Online] Available at:

<https://monday.com/>

[Accessed on 4 September 2024].

* freshworks. 2024. Simplified ITSM, powered by AI, 9 April 2024.

[Online] Available at:

<https://www.freshworks.com/freshservice/>

[Accessed on 5 September 2024].

* long international. 2024. Construction claims analysis checklist, 10 February 2024.

[Online] Available at:

<https://www.long-intl.com/articles/construction-claims-analysis/>

[Accessed on 6 September 2024].

* visual paradigm, 2024. UML class diagram tutorial, 2024.

[Online] Available at:

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/>

[Accessed on 8 September 2024].

* asana, 2024. What is project planning? (Plus, 7 steps to write a successful project plan), 2 February 2024.

[Online] Available at:

<https://asana.com/resources/project-management-plan>

[Accessed on 8 September 2024].