Kgosi Olifant

ST10384317

Programming 2B

POE PART 2 - UPDATED

Documentation – CMCS

The complete documentation for the Contract Monthly Claim System (CMCS) prototype development is provided in this paper. The CMCS is an online tool designed to make it easier for Independent Contractor (IC) lecturers to submit and approve their monthly claims by automating and streamlining the process. ICs can use the system to submit claims depending on the number of hours they have worked; Coordinators and Managers will then review and approve the claims. Transparent claim tracking and document submissions are also supported by the system.

Design Choices:

The Model-View-Controller (MVC) paradigm was selected to provide greater maintainability and scalability by separating the data, user interface, and control logic.

The database of the system is made to effectively handle various data entities, such as:

Lecturer: Keeps track of lecturer details including name, ID, and hourly rate.
Claim: Provides information on the hours worked, date, and status of the claim (pending, approved, or denied).
Supporting Documents: Holds references to files that were uploaded with the submission of the claim.
Manager: Oversees the process of approval and verification.

The modular and scalable structure of these entities makes it simple to add additional functionality as needed. To ensure that claims may be submitted, processed, and tracked independently while still having a strong relationship with lecturer data, for instance, the `Lecturer} and `Claim` entities have been separated.

GUI DESIGN:

A user-friendly interface built with .NET Core allows lecturers to submit claims and administrators to review and approve them. The design features a straightforward form layout and intuitive controls, ensuring easy navigation.

1 – Login Point

- This window presents two options for the user:
    1. Login as Lecturer
    2. Login as Manager

2 - Submit Claim

- The lecturer follows the steps to submit a claim.
- This also includes fields for supporting documents.
- After selecting "Submit," a confirmation message is shown and the claim information are verified and computed.
- The design prioritizes usability, with every input field prominently labeled and easily accessible.

Assumptions and constrains

Assumptions:

• Lecturers enter claims, accurately entering their data (e.g., valid names, proper hours worked).

• Coordinators have complete control over whether to accept or reject claims.

• The risk of duplicate claims will be decreased because the professor will only submit claims once per month.

• The hourly pay for lecturers is set.

Constrains:

• Just a few fields are checked to make sure they are not empty at the moment. It might be necessary to add more tests (such as data type validation and making sure the hours and rate are positive).

• At this point, claims are kept in RAM and are deleted when the program terminates. In the future, switching to persistent storage (such a database) would be essential.

• There is currently no real file processing implemented, hence the file upload feature is minimal.

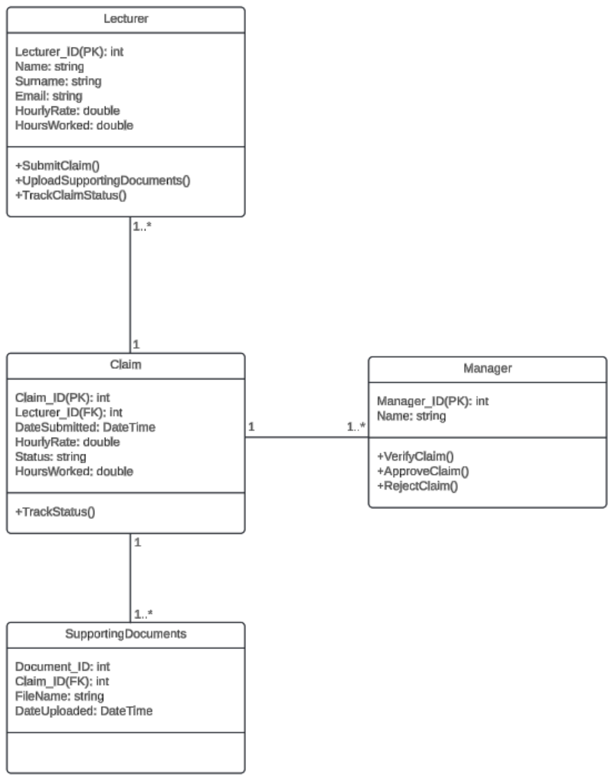
• The coordinator/Manager view and lecturer submission are distinct parts of the existing structure, and access is not controlled by authentication methods. To solve this, a login system would be implemented.

Database Structure:

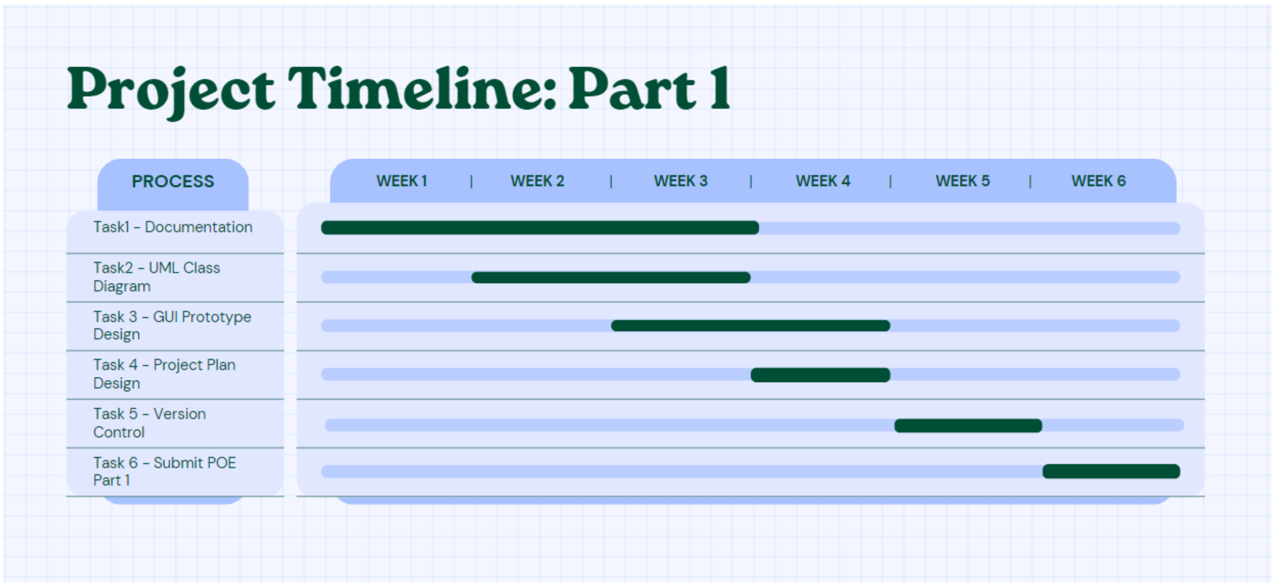
Relational database built on SQL and featuring normalized tables to effectively store the following:

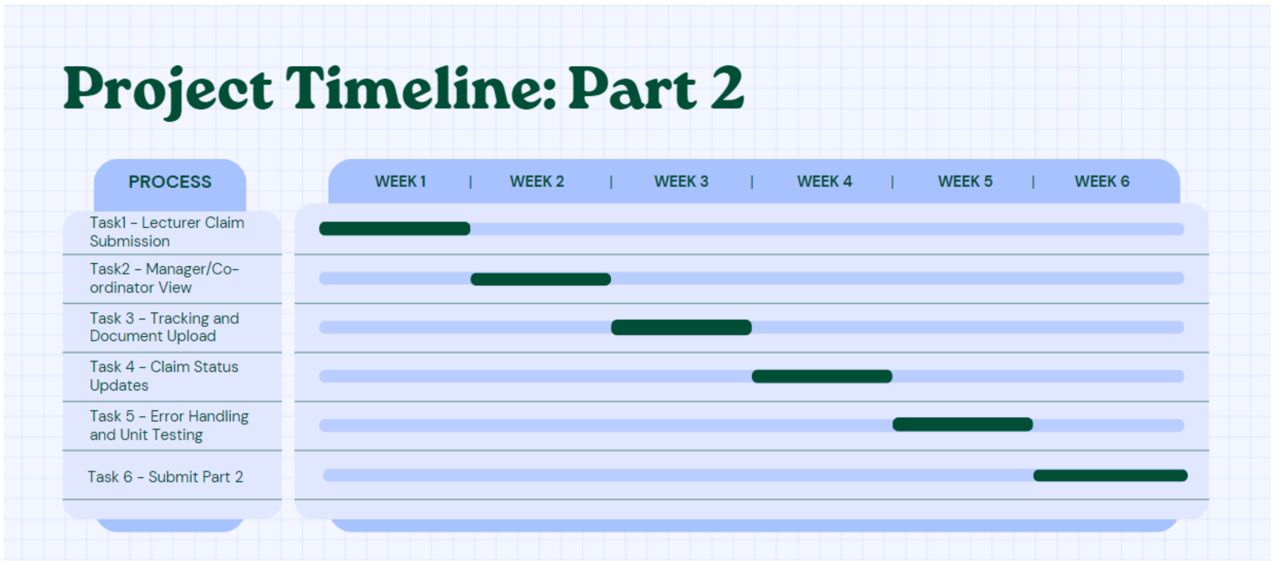
```
-- Lecturer Table Structure
CREATE TABLE Lecturer (
   Lecturer_ID INT PRIMARY KEY,
   Name VARCHAR(100),
   Surname VARCHAR(100),
   Email VARCHAR(100),
   HourlyRate DOUBLE,
   Date DATE
   HoursWorked DOUBLE
);

-- Claims Table Structure
CREATE TABLE Claims (
   Claim_ID INT PRIMARY KEY,
   Lecturer_ID INT,
   DateSubmitted DATE,
   HourlyRate DOUBLE,
   Status VARCHAR(20),
   HoursWorked DOUBLE,
   FOREIGN KEY (Lecturer_ID) REFERENCES Lecturers(Lecturer_ID)
);

-- Managers Table Structure
CREATE TABLE Manager (
   Manager_ID INT PRIMARY KEY,
   Name VARCHAR(100),
   Surname VARCHAR(100)
);

-- Supporting Documents Table Structure
CREATE TABLE SupportingDocuments (
   Document_ID INT PRIMARY KEY,
   Claim_ID INT,
   FileName VARCHAR(100),
   DateUploaded DATE,
   FOREIGN KEY (Claim_ID) REFERENCES Claims(Claim_ID)
);
```

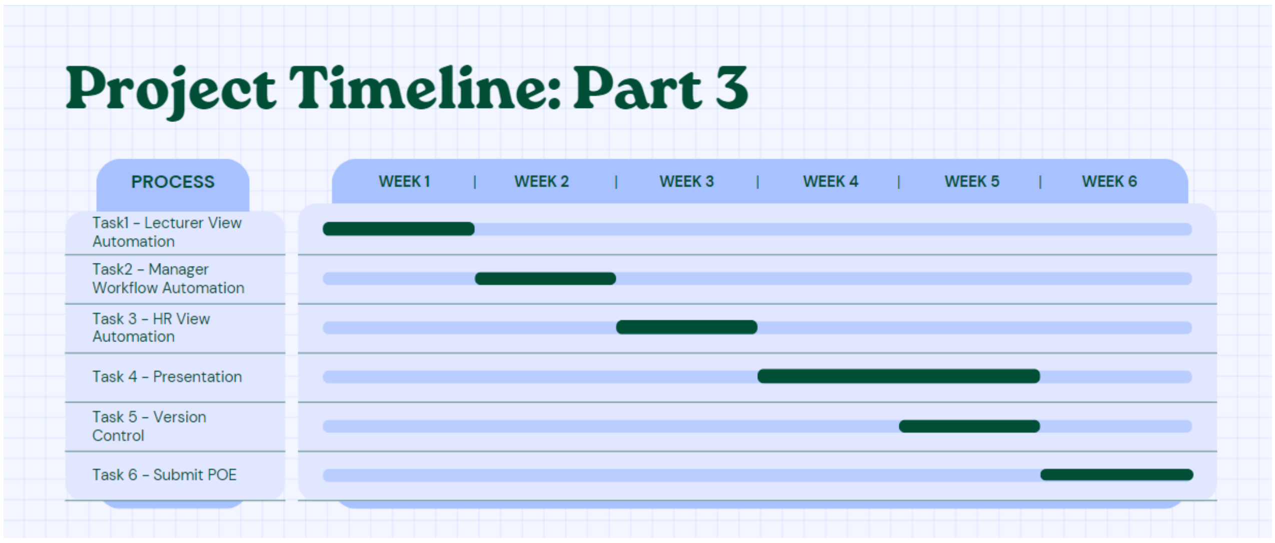
UML Class Diagram



Project Plan







GUI Design

WPF/MVC are used in.NET Core to develop the graphical user interface (GUI). The prototype at this point is mostly concerned with the visual design without any functional implementation. The main components of the user interface consist of:

Login Interface

We'll be using Windows Presentation Foundation (WPF) to build the graphical user interface (GUI), ensuring it is straightforward and user-friendly. Below are the design details along with explanations:

Lecturer:

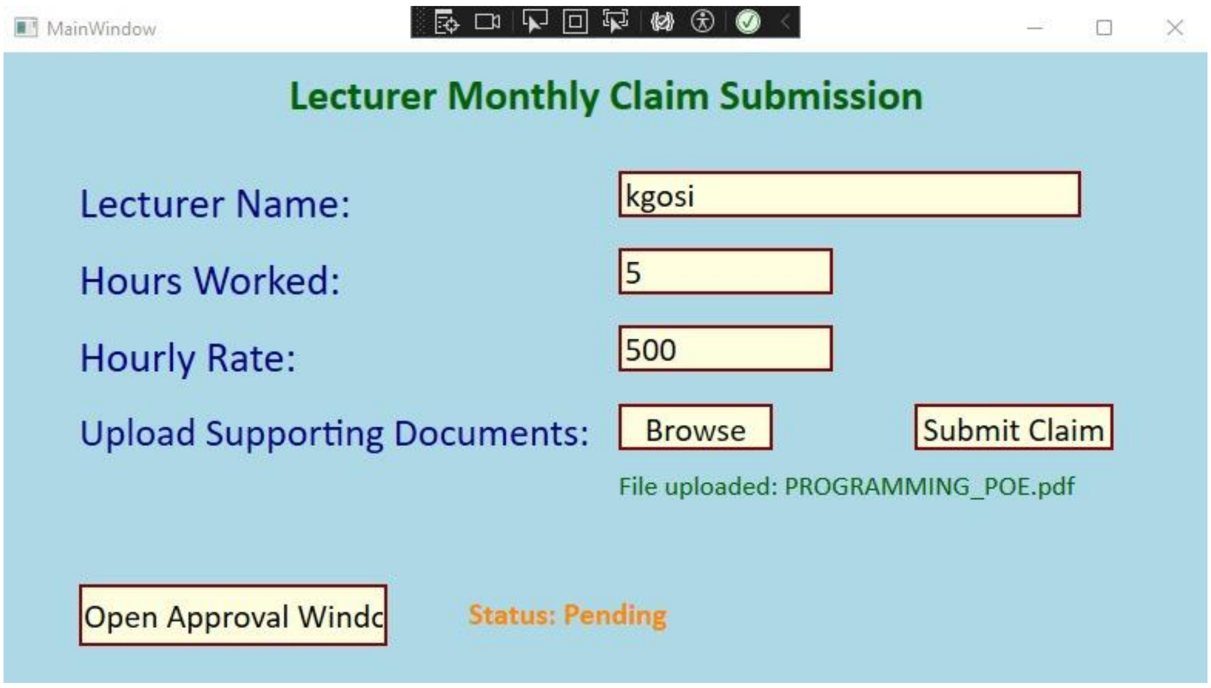
- **Claim Submission Form**: A user-friendly form for lecturers to log their work hours and upload necessary supporting documents.
- **Upload Functionality**: A straightforward button for easily uploading files or documents.
- **Submit Functionality**: A button that allows lecturers to submit their claims, which triggers a validation process before saving the details to the database.
- **Claim Status Display**: A section where lecturers can view the progress and current status of their submitted claims.
- **Approval Window Access**: A button that allows lecturers to open the claim approval form.

Programme Manager Interface:

- **Claim Review and Approval Panel**: A screen displaying a list of claims that require verification or approval. Each claim can be expanded to review details such as work hours, total amount claimed, and attached documentation.
- **Approval/Denial Options**: Buttons for approving or rejecting claims, with the ability to provide reasons for denial.

UI Considerations:

- The system will offer real-time updates on the status of claims.
- The claims will move through different stages: Submitted → Verified → Approved → Settled.
- The interface will focus on accessibility, using clear labels and instructions for ease of use.


MainWindow

Lecturer Monthly Claim Submission

Lecturer Name: kgosi

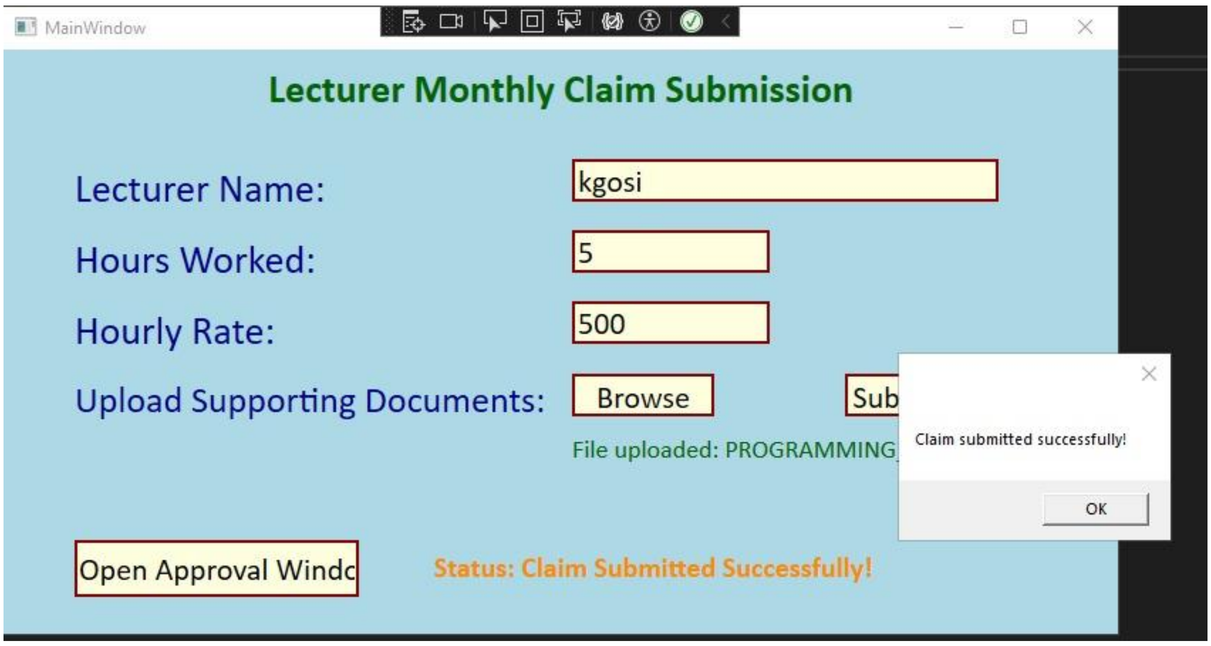
Hours Worked: 5

Hourly Rate: 500

Upload Supporting Documents:

File uploaded: PROGRAMMING_POE.pdf

Status: Pending


MainWindow

Lecturer Monthly Claim Submission

Lecturer Name: kgosi

Hours Worked: 5

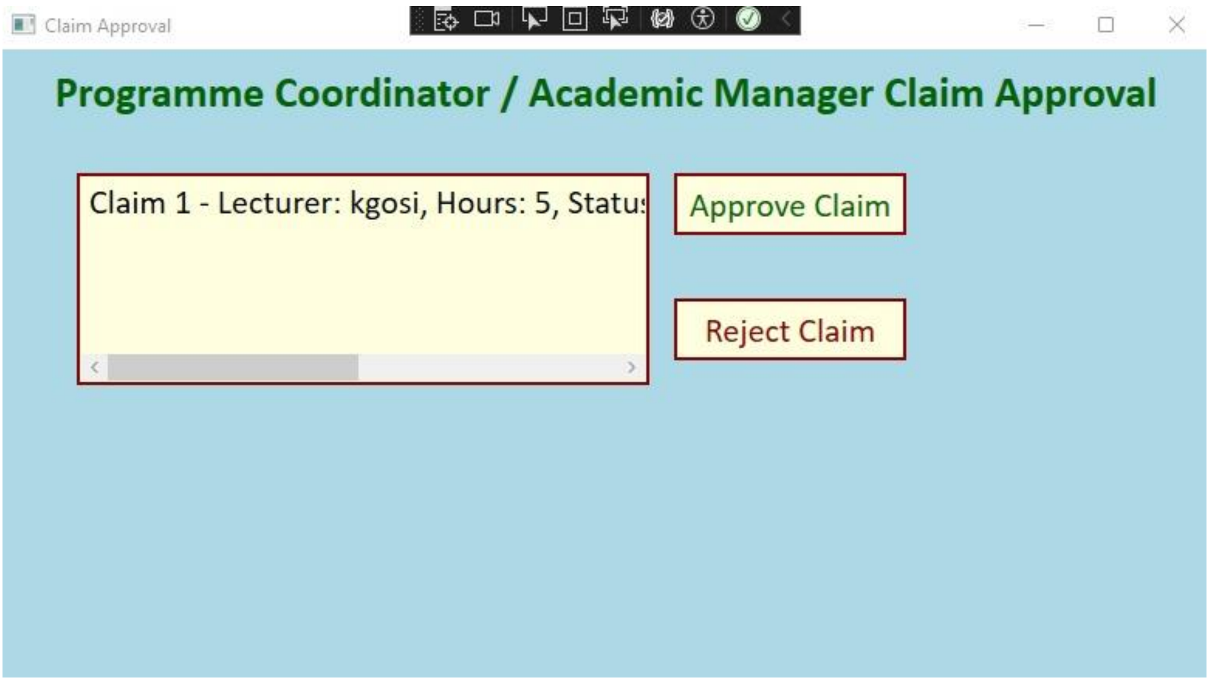
Hourly Rate: 500

Upload Supporting Documents:

File uploaded: PROGRAMMING_

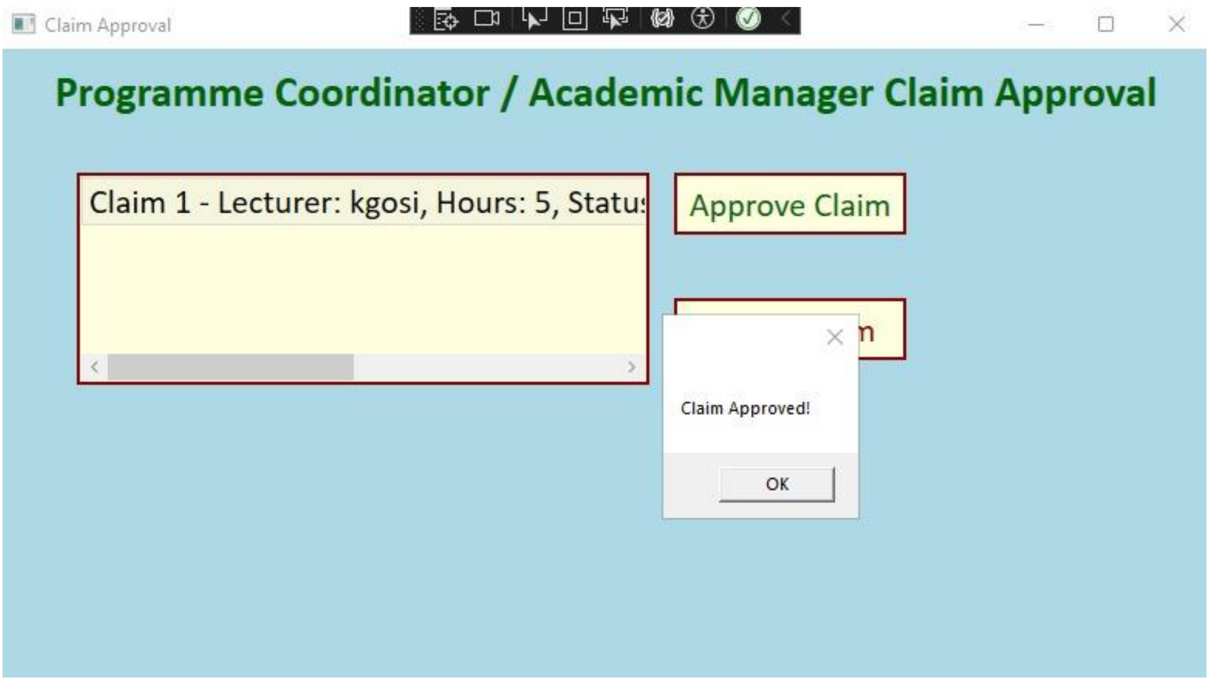
Claim submitted successfully!

Status: Claim Submitted Successfully!


Claim Approval

Programme Coordinator / Academic Manager Claim Approval

Claim 1 - Lecturer: kgosi, Hours: 5, Statu


Claim Approval

Programme Coordinator / Academic Manager Claim Approval

Claim 1 - Lecturer: kgosi, Hours: 5, Statu

Claim Approved!