**Student Name+Surname:** Sbongile Chauke

**Module Name:** Programming 2B

**Module Code:** PROG6212

**Assessment Type:** Portfolio of Evidence (POE)

**Due Date:** 22 November 2024

**POE: Automation of Web Application**

**Version Control:**

<https://github.com/SbongileC/LecturerClaimSystem_PROG6212_POE>

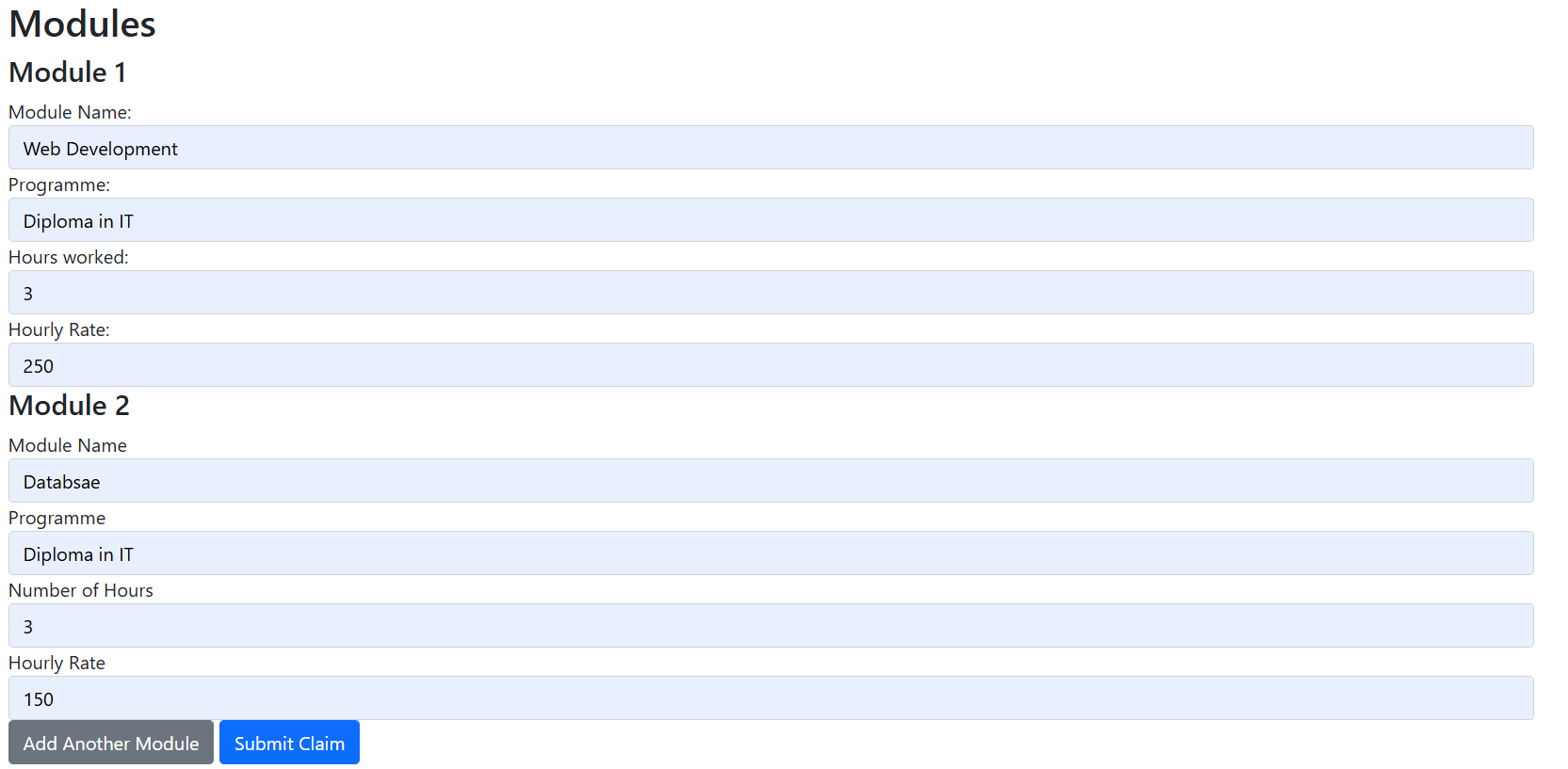
**Part 2 – Implement a Prototype Web Application**

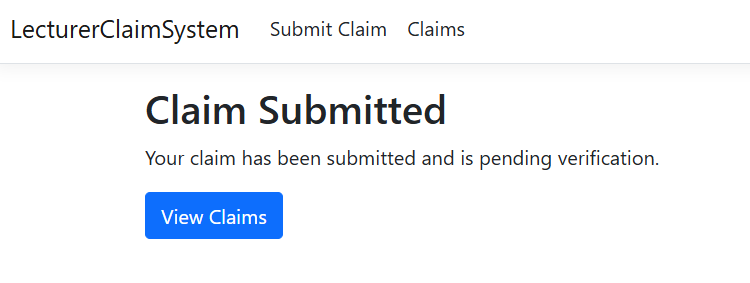
**Changes in Part 1**

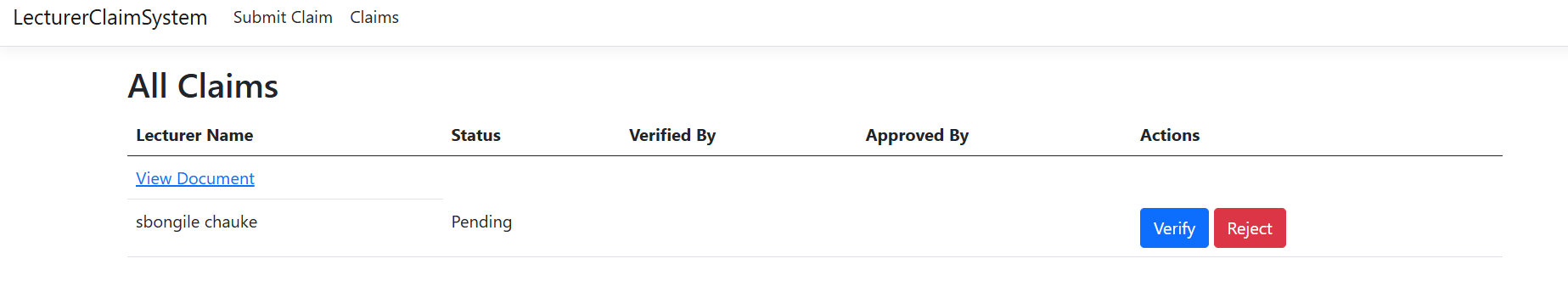
* Added a field for additional notes in the lecturer claim submission form.
* Added a reject button in the coordinator and managers view and a field to add a reason for the rejection.
* Ensured that the files uploaded are securely stored and linked to a corresponding claim submitted.
* Implemented restricting the file types to common formats such as pdf, docx and xlsx.

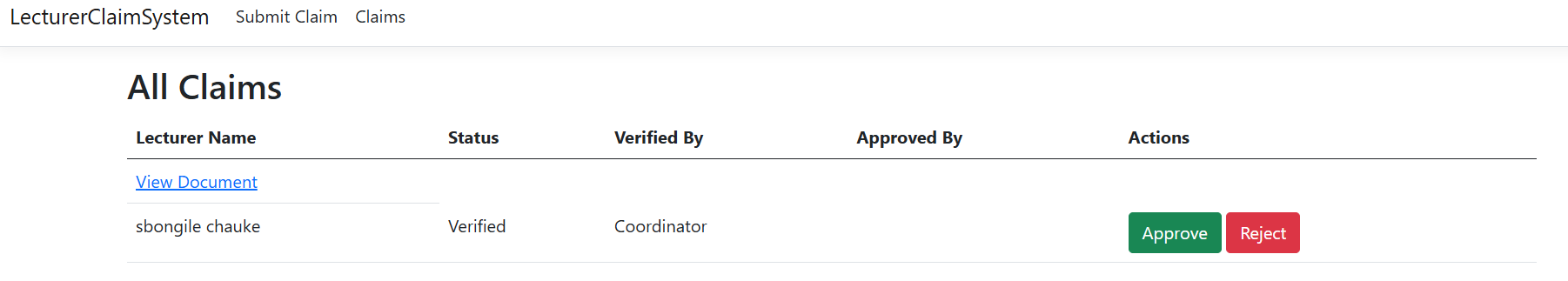
**Screenshots**

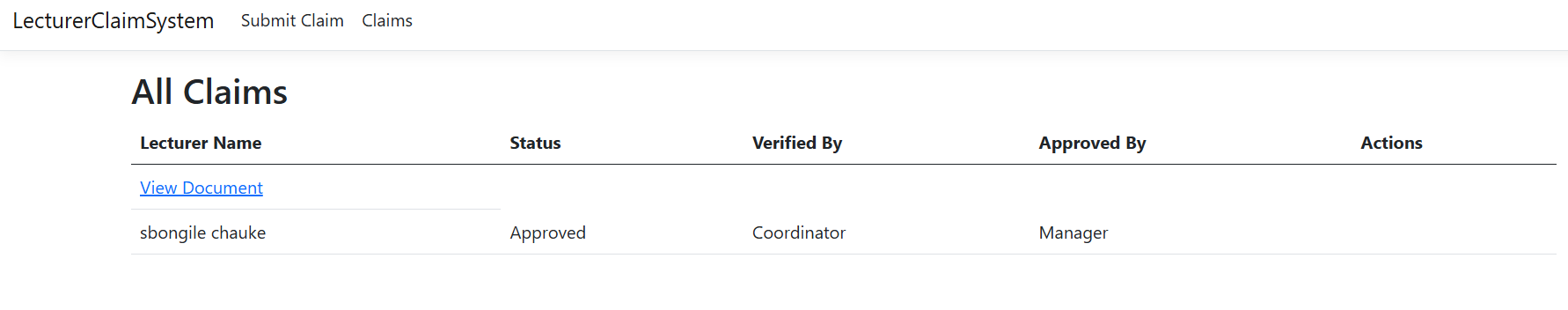


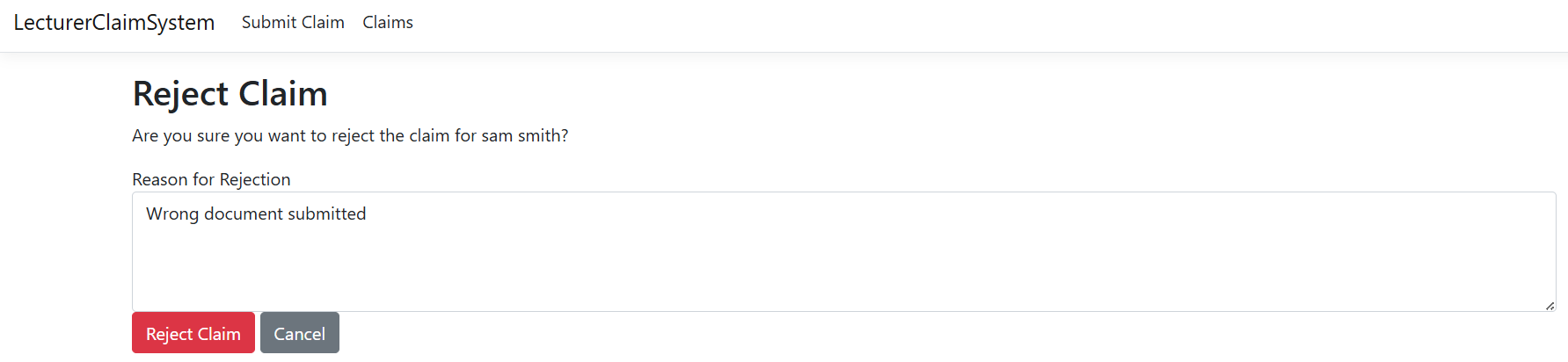


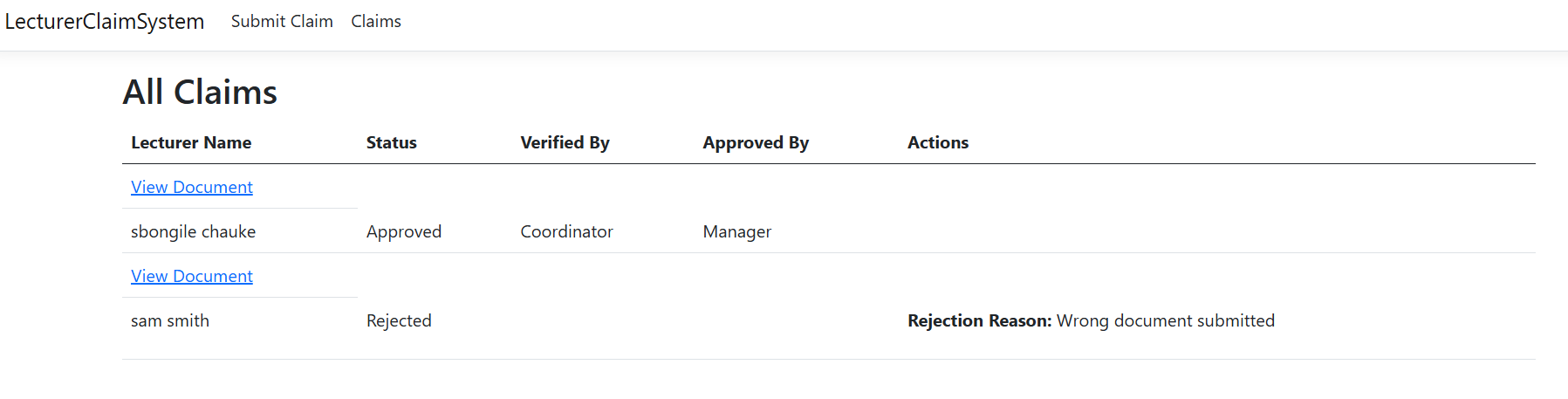












**Version Control**

<https://github.com/SbongileC/LecturerClaimSystem_PROG6212_Part2>

**Part 1 – Project Planning and Prototype Development**

1. **Documentation**

The contract monthly claim system serves as a useful tool for streamlining the process of submitting and approving monthly claims for independent Contractor lectures. The role of a lecturer extends beyond merely submitting claims, it involves complex calculations based on hours worked and corresponding hourly rates. These claims undergo observation by the Program Coordinator and the Academic Manager, highlighting the importance of accuracy and accountability in administrative processes. Furthermore, the system’s integration of features will go beyond basic claim submissions, providing a seamless platform for uploading essential supporting documents. By facilitating such functionalities, the system aims to not only increase efficiency but also enhance user satisfaction and reduce potential errors.

The Contract Monthly Claim System stands as a testament to innovation in administrative processes, offering a glimpse into the future of streamlined claim management. With its user-centric design and seamless integration of features, the system aims to revolutionize the way claims are processed and approved. Streamlining repetitive tasks and providing interfaces to empower both lecturers and administrators to focus on more strategic initiatives enhancing organizational efficiency and productivity.

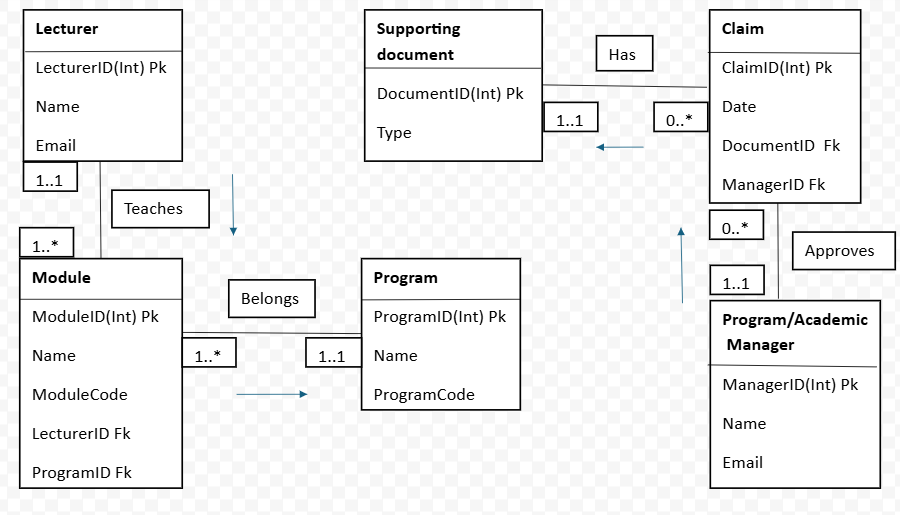
The database's structure will be designed as a relational database (SQL Server) selected for its scalability and ability to handle complex queries, essential for managing financial data and generating reports. It will be normalized up to the third normal form to reduce data redundancy and ensure data integrity. The UML class diagram will represent the data requirements of the Contract Monthly Claim System. It will include all necessary entities (Claim, Program, Lecturer, Module, supporting document, Program Manager, Academic Manager, Contract, Payment), attributes, and relationships and show how they are represented in the database.

The GUI layout will be designed using Model View Control (MVC) with .NET Core because of its flexibility in creating modern, responsive desktop applications. It will display interfaces of tasks that will be performed such as lecturers can submit their claims with a click of a button, Programme Coordinators and Academic Managers can easily verify and approve the claims as well as lecturers can upload supporting documentation for claims.

Considering the assumptions, all contractors have access to the internet and can use a desktop application, the system will handle up to 5 000 claims per month as well as the database residing on a secure server with regular backups.

On the other hand, the system must comply with financial regulations and data privacy rules, budget constraints limit the use of certain technologies or third-party services, and the system should be implemented within a brief time.

**2. UML Class Diagram**



**3. Project Plan**

**System Development Project Plan**

**1. Project Overview**

**1.1 Project Name**

**Name:** Contract Monthly Claim System

**1.2 Project Description**

**Description:** The contract monthly claim system serves as a crucial tool for streamlining the often-complex process of submitting and approving monthly claims for independent Contractor lectures.

**1.3 Project Objectives**

Objectives:

* Develop a system that automates the submission, tracking and processing of monthly claims by lectures.
* Implement robust validation and verification mechanisms to ensure all claims and supporting documentation are accurate and comply with contractual terms.
* Design a user-friendly and intuitive interface.

**1.4 Project Scope**

In Scope:

* Development of a module that allows lecturers to submit monthly claims electronically. Design and implementation of a relational database to store claims, supporting documentation, user information and approval history.
* Development of user-friendly GUI. implementation of security measures to protect data integrity and privacy.
* Comprehensive testing to ensure the system meets all functional and non-functional requirements.
* Planning and execution of the deployment of the system into a live environment.

Out of Scope:

* Development of dedicated mobile applications.
* Ongoing support and maintenance of the system beyond the initial data.
* Features not related to the monthly claim support.

**1.5 Deliverables**

Deliverables:

1. Requirements Documentation.
2. System Design Documentation.
3. Database Schema and Implementation.
4. Developed Application (Frontend and Backend).
5. User Interface Mockups and Prototypes.
6. Test Plan and Test Cases.
7. Test Results and Bug Reports.
8. Deployment Plan.
9. Final Project Report.

**1.6 Assumptions and Constraints**

**Assumptions:** All contractors have access to the internet and can use a desktop application, the system will handle up to 5 000 claims per month as well as the database residing on a secure server with regular backups.

**Constraints:** The system must comply with financial regulations and data privacy rules, budget constraints limit the use of certain technologies or third-party services, and the system should be implemented within a brief time.

**2. Project Organization**

**2.1 Stakeholders**

Stakeholders:

* Lecturers
* Program/Academic Manager
* Finance Department
* IT Department
* Project Manager
* System Developers
* Human Resource

**2.2 Project Team**

Project Manager: Sbongile Chauke

Team Members:

* Developing the whole project: Sbongile Chauke

**2.3 Roles and Responsibilities**

Software Developer: Develop the whole project

**3. Project Phases and Milestones**

**3.1 Initiation Phase**

Tasks:

The contract monthly claim system aims to automate the claims process, ensuring compliance and improving efficiency.

Stakeholders include:

* Lecturers
* Program/Academic Manager
* Finance Department
* IT Department
* Project Manager
* System Developers
* Human Resource

**3.2 Planning Phase**

Tasks:

* Develop project plan
* Define project scope
* Create work breakdown structure (WBS)
* Develop schedule and budget
* Identify risks and create a risk management plan

Milestones:

* Project plan approval
* Schedule and budget approval

**3.3 Execution Phase**

Tasks:

* Develop system architecture and design of the contract monthly claim system, including database schema.
* Coding and unit testing for all system components and perform unit testing to ensure the project works as intended.
* System integration for all components of the system to ensure they work together smoothly.
* Prepare User training materials and conduct training sessions for end-users to familiarize them with the new system.

Milestones:

System architecture approval is reviewed by technical stakeholders and project leads.

**3.4. Monitoring and Controlling Phase**

Tasks:

* Track project progress against the project plan and adjust, as necessary.
* Perform quality assurance such as code reviews, testing and audits to ensure deliverables meet the required standards.
* Manage changes
* Mitigate risks.

Milestones:

* Regular status reports
* Quality audits

**3.5 Closure Phase**

Tasks:

* Finalize documentation including user manuals, technical guides, and final reports.
* Release and deployment
* Conduct project review

Milestones:

Project closure report, summarizing project outcomes, challenges, and lessons learnt for future reference.

**4. Project Schedule**

**4.2 Key Dates**

* **Start Date:** 09 September 2024
* **End Date:** 22 November 2024

**5. Risk Management**

**5.1 Risk Identification**

Risks:

* Insufficient testing leads to undetected bugs.

**5.2 Risk Mitigation Plan**

Mitigation Strategies:

* Develop a detailed and realistic budget. Monitor expenditures closely and adjust the budget as needed.

**6. Communication Plan**

**6.1 Communication Methods**

Methods:

* Meetings
* Email updates
* Status reports

**7. Quality Management Plan**

**7.1 Quality Objectives**

Objectives:

* Ensure system meets requirements
* Maintain high code quality

**7.2 Quality Assurance**

QA Activities:

* Code reviews
* Testing (unit, integration, system)

**8.3 Quality Control**

* Defect tracking
* Performance testing

**9. Change Management Plan**

**9.1 Change Request Process**

Process:

* Submit change request
* Review and approve change request
* Implement change
* Update project plan

**10. Project Closure**

**10.1 Final Deliverables**

Deliverables:

* Final system release
* Documentation

**10.2 Lessons Learned**

Lessons Learned:

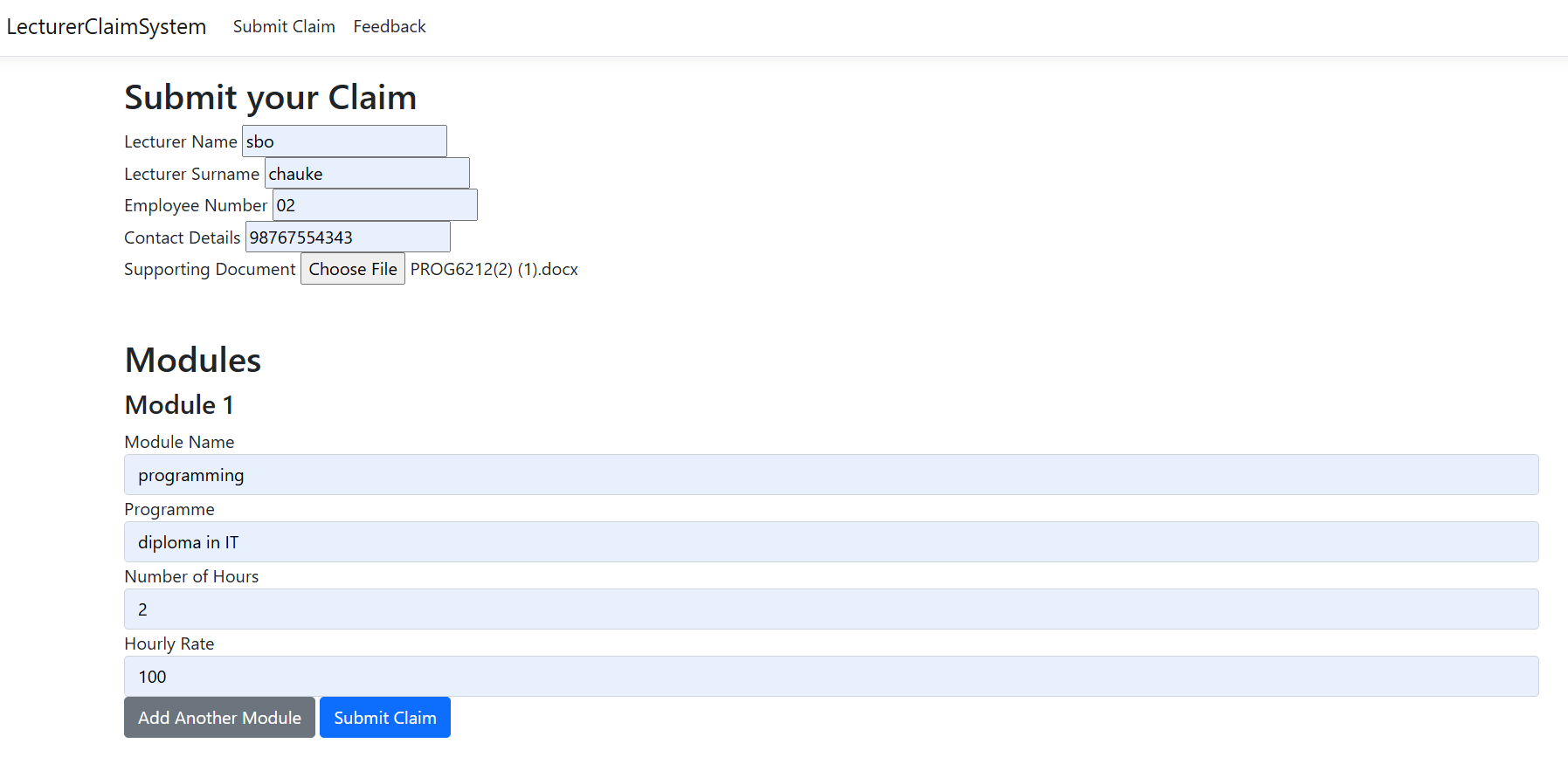
* Clear and detailed requirements from stakeholders are crucial for project success.
* Continuous and active engagement with stakeholders helps in aligning the project’s goals with business needs.
* Integration with existing systems needs careful planning and early testing.
* Proactive risk management helps in minimizing the impact of potential issues.

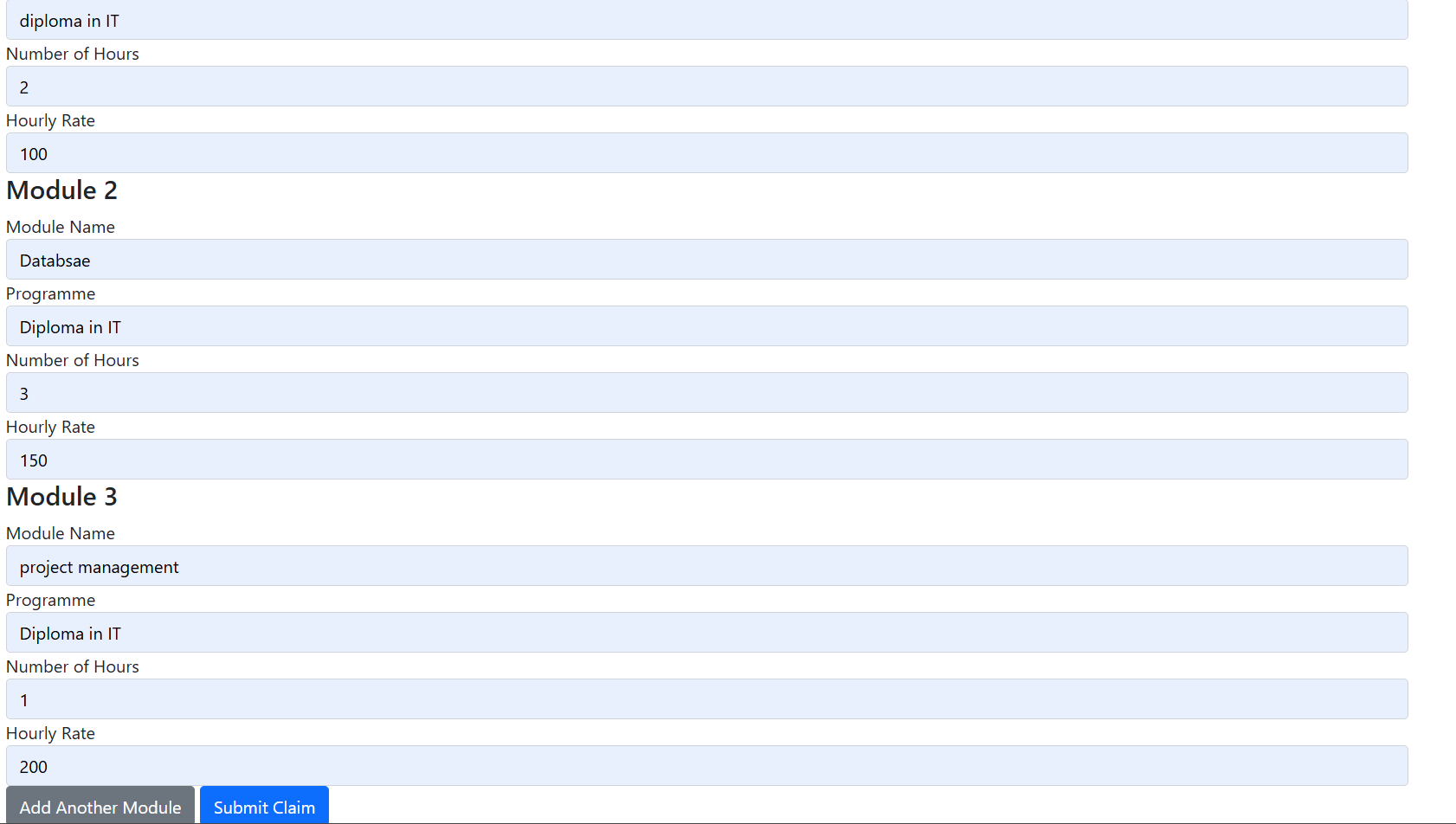
**10.3 Project Review**

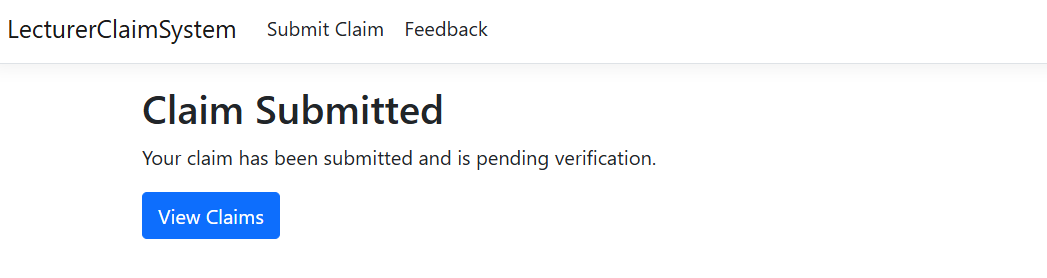
Review Activities:

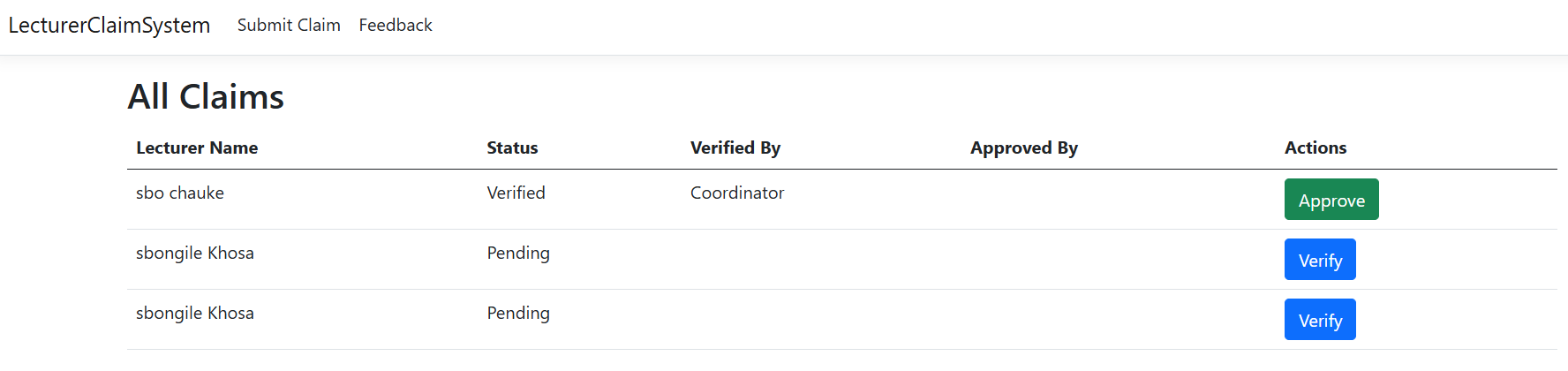
* Conduct projects debrief
* Archive project documents

**4. GUI UI**









**5. Version Control**

<https://github.com/SbongileC/LecturerClaimSystem_PROG6212_Part1>