# **Contract Monthly Claim System for Independent Lecturers**

# **Report**

# **Process document**

## Design choice:

The prototype design choice for this system will be as follows:

* Secure login for independent contractor lecturers, program coordinators, and academic managers.
* Online claim submission form with required fields (hours worked, hourly rates, date).
* User-friendly interface with clear navigation.
* Document upload functionality such as drag-and-drop.
* Claim status dashboard with real-time updates.
* Role-based access control to protect sensitive data.
* Notification system for claim submissions, approvals, and rejections.

## Reason why I am using this design choice for the prototype of the system:

I choose this design choice because the core functionalities of this prototype are:

* **Claim approval and verification** where the project coordinator and academic managers can review claims.
* **Claim submissions** where the independent contractor lecturer can submit their claims.
* **Claim status tracking** where the independent contractor lecturer, project coordinators, and academic managers can all view the claim status.
* **Document upload** where the claiming independent lecturers can attach supporting documents.

# **Database Structure:**

In the system that I will be developing, I will structure my database as follows to ensure data integrity and minimize redundancy in my systems. I will have the following tables in the database:

* Academic institution Table
* Lecturer Table
* Claim Table
* Payment Table
* Claiming Status Table
* Supporting Documents Table

## Explaining the information of each table:

### *Academic institution Table:*

Will store information about the academic institutions, such as their identification and contractual details.

### *Lecturer Table:*

Will allow the user to store information specific to the independent contractor lecturers, such as their contact details, qualifications, experience, hours worked, and hourly rates.

### *Claim Table:*

 Will capture the details of each monthly claim submitted by the lecturers, including the claim amount, date, approving project coordinators and academic managers, and supporting documents.

### *Payment Table:*

Will store detailed payment information, such as payment date, amount, payment method, reference number, and bank details. Additionally, the payment table will maintain a historical record of all payments made, allowing for easy reference and analysis.

### *Claim Status Table:*

Will track the approval status of each claim, indicating whether it is approved, declined, or pending.

### *Supporting Documents Table:*

Claims often require supporting evidence. Having a supporting document table will allow proper organization, storage, and retrieval of these documents. All the uploaded files will be received and stored here.

# **GUI Layout:**

I will be using the following layout for my GUI:

1. To ensure security and result reliability, my system will have a welcome home page with 2 buttons asking the user to login according to the type of user they are. The buttons will be ‘IC Lecturer’, Contractor (‘Project Coordinator’ OR ‘Academic Manger’).
2. After selecting the user type, a login page that prompts the user their login information according to the user type they selected, will be displayed.
3. If the user selected to login as IC lecturer and has entered the correct login details and pressed the login button on the login page, immediately redirect them to the ‘Claim Submission’ page where they can submit their claim.
4. On the ‘Claim Submission’ page the lecturer will have to fill in a form with the details of the contractor that they are submitting a claim to and then upload their supporting documents by clicking on the ‘File Upload’ button. Additionally, a button named ‘Claim status’ will be added to allow the lecturer the option of navigating to their claim status page.
5. On the ‘Claim Status’ page the result of the claim will be displayed to the lecturer and the details of the contractor who approved or declined the claim will display. If a claim has not been responded to by the contractor, the word ‘Pending’ will be displayed.
6. On the ‘Claim Status’ page the lecturer will be able to view their claim and see its status. If the claim is approved the total payment that the lecturer must receive will appear and a conditional form will display that will ask the lecturer to fill in their banking details so that they can receive their money(claim).
7. If the user selected to login as a Contractor, entered the correct login details, and pressed the login button on the login page, immediately redirect them to the ‘Claim approval and verification’ page (Claim Respond page). One this page display, all the claims from different lecturers and display 2 buttons to approve or decline the claims. While the project coordinator or academic managers have not viewed the claim, display to the lecturer in the claim status page the word ‘Pending’.

# **Potential assumptions or constraints:**

I considered having a login page **without asking** the user to select as which user type they want to login into the system. There are 2 user types that can login: Independent Contractor Lecturer (IC Lecturer) AND Contractor (Project Coordinator OR Academic Manager).

The problem with this approach is that it does not ensure security, accuracy, integrity, and result reliability. For example, a lecturer could login and go straight to the ‘Claim and Verification’ page and approve their own claim. They would have access to all the pages as long as they can login into the system.

# **UML Class Diagram:**

Contractor

ContractorID: int (PK)

ContractorName: string

Address: string

ContactNumber: string

ContractorEmail: string

ClaimID: int (FK)

getContractor()

setContractor()

Claim

ClaimID: int (PK)

ClaimDate: date

ClaimAmount: decimal

ContractorID: int (FK)

LecturerNumber: int (FK)

ClaimStatusID: int (FK)

ModuleCode: string

Program: string

SuppDocID: int (FK)

calculateClaimAmount()

getClaim()

setClaim()

ClaimStatus

ClaimStatusID: int (PK)

Status: string

LecturerNumber: int (FK)

ClaimID: int (FK)

getClaimStatus()

setClaimStatus()

1 \*

\* 1

1

1

\* \*

ContractorLecturer

ContractorLecID: int (PK)

ContractorID: int (FK)

LecturerNumber: int (FK)

Payment

PaymentID: int (PK)

PaymentDate: date

PaymentMethod: string

ClaimAmount: decimal

LecturerNumber: int (FK)

ClaimID: int (FK)

getBankDetails()

setBankDetails()

\* 1

1 \*

\*

\* 1 1

Lecturer

LecturerNumber: int (PK)

LlecFisrtName: string

LecLastName: string

LecEmail: string

ModuleCode: string

Program: string

HourlyRate: int

HoursWorked: time

ContractorID: int (FK)

ClaimID: int (FK)

SupportingDoc

SuppDocID: int (PK)

FileName: string

FilePath: string

UploadDate: date

ClaimID: int (FK)

# **Project Plan**

## **1. Project Overview**

### **1.1 Project Name**

Name: Contractor Monthly Claim System

### **1.2 Project Description**

Description: A system that allows Independent Contractor(IC) Lecturers to submit monthly claims and receive a respond from the contractor that state whether the claims are approved or rejected.

### **1.3 Project Objectives**

**Objectives:**

- **Objective 1:** The system must allow Independent Contractor lecturers to submit monthly claims.

- **Objective 2:** The system must allow Contractors to respond to the Independent Contractor lecturers claims by approving or declining the claims.

- **Objective 3:** The system must allow Independent Contractor lecturers to view the status of their claims.

### **1.4 Project Scope**

-**In Scope:** I will include the project objectives, deliverables, timelines, resources, stakeholders, tasks and activities.

-**Out of Scope:** I will exclude assumptions and constraints, future enhancements, tools, frameworks, and client responsibility.

### **1.5 Deliverables**

**Deliverables:**

- **Teaching sessions:** The lecturer will have to submit the number of hours they have worked whether the sessions were completed as online classes or contact classes.

- **Module and Program:** Lecturers would have to submit the modules and program they have taught during the period of a month.

- **Course Material:** Lecturers will have to submit the presentations, lecturer notes, and reading lists that they did with students and would have to be submit this course material in a specific time to support the learning objectives.

### **1.6 Assumptions and Constraints**

**Assumptions:**

* **Understanding of the scope:** The project scope us clearly understood and agreed upon by all team members and stakeholders.
* **Stakeholder engagement:** Stakeholders provide feedback within the specified time so that my team and I can start building the contractor monthly claim system.
* **Resource Availability:** The necessary resources must be available when needed.

**Constraints:**

* **Cost:** Budget limitations that restrict how much money can be spent on resources and activities needed to build a successful system.
* **Time:** Parts of the project that must be submitted at a specific date, but stakeholders do not give us feedback on time, so our entire project is delayed.
* **Quality of project:** Certain parts of the project being delayed resulting in my team members having to work in a rush to complete other tasks, which results in low quality project and contractor monthly claim system.

## **2. Project Organization**

### **2.1 Stakeholders**

**Stakeholders:**

- Contractor

- Programme Coordinator

- Academic Manager

### **2.2 Project Team**

- Project Manager: Mathabo Motaung

Team Members:

- Project Manager – Mathabo Motaung

- Web Developer – Bianca Stuurman

- Software Developer – Tyson Sybatelli

- System Designer – Marcus Edwards

- System Analyst – Luthando Josephs

### **2.3 Roles and Responsibilities**

**Roles and Responsibilities:**

- **Project Manager:** Responsible for the planning, obtaining and execution of a project with a defined scope, start and finish, which was agreed upon with stakeholders.

- **System Analyst:** Responsible for analysing the system, risk mitigation planning, maintaining, and improving the monthly claim system.

- **Software Developer:** Responsible for coding, testing, and deploying the contractor monthly claim system.

- **System Designer:** Responsible for gathering all software and hardware required to build the system, produce prototypes, designing the system, and oversee quality assurance process and implement changes to the contractor monthly claim system.

- **Web Developer:** Responsible for developing functional and user friendly web apps and websites, validating tests routines to ensure the quality of the external and internal interface, testing web apps and websites before deployment, and connecting the database and maintaining integration.

## **3. Project Phases and Milestones**

### **3.1 Initiation Phase**

- Tasks:

- Define objectives, assumptions, and constraints

- Identify stakeholders

- Define database structurer

- Develop a UML Class diagram

- Milestones:

- Project objectives, assumptions, and constraints approval

- Database structurer approval

### **3.2 Planning Phase**

- Tasks:

- Develop project plan

- Define project scope

- Develop schedule and budget

- Identifying team members

- Milestones:

- Project plan approval

- Team members approval

### **3.3 Execution Phase**

- Tasks:

- Develop system architecture and design

- Coding and unit testing

- System integration

- Developing prototype

- Milestones:

- Prototype approval

- Code complete

- System integration complete

### **3.4 Monitoring and Controlling Phase**

- Tasks:

- Track project progress

- Perform quality assurance

- Manage changes

- Mitigate risks

- Milestones:

- Tracking project progress complete

- Perform quality assurance complete

### **3.5 Closure Phase**

- Tasks:

- Conduct user acceptance testing (UAT)

- Finalize documentation

- Release and deployment

- Conduct project review

- Milestones:

- UAT sign-off

- Project closure report

## **4. Project Schedule**

### **4.1 Gantt Chart**

- **Gantt Chart:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Task Name** | **24-Jul-24** | **30-Jul-24** | **5-Aug-24** | **12-Aug-24** | **19-Aug-24** |  |
| **Planning** |  |  |  |  |  |  |
| **Research** |  |  |  |  |  |  |
| **Design** |  |  |  |  |  |  |
| **Implementation** |  |  |  |  |  |  |
| **Follow Up** |  |  |  |  |  |  |

### **4.2 Key Dates**

- Start Date: 24 July 2024

- End Date: 19 August 2024

## **5. Budget and Resources**

### **5.1 Budget**

- Estimated Budget: Total Budget: R 200,000

- Team Members: R 80,000

- System Security Specialists: R 60,000

- Programming tools and frameworks: R 20,000

- Data Scientist to develop data structures and algorithms: R 40,000

### **5.2 Resource Allocation**

**Resources Required:**

- System security specialists: 3

- Team members: 5

- Data Scientists: 2

- Programming tools and framework licenses: 10

## **6. Risk Management**

### **6.1 Risk Identification**

**Risks:**

- **Risk 1:** Building one system for 2 operations -> The system must assist IC lecturers, and the system must enable Contractors, Project Coordinators, and Academic Managers to approve or decline claims using one system

- **Risk 2:** Financial Management -> If contractors do not track the income and expenses of IC lecturers it can lead to unexpected tax liabilities and financial strain, as IC lecturer are responsible for their own taxes.

- **Risk 3:** Payment disputes -> There might arise disputes between stakeholders and project manager if the expectations regarding deliverables are not clearly defined

### **6.2 Risk Mitigation Plan**

**Mitigation Strategies:**

- **Risk 1:** Building two separate systems for lecturers and for contractors

- **Risk 2:** When a contractor is hiring a IC lecturer inform them that a certain percentage will be taxed from their monthly income to avoid unexpected tax liabilities and financial strain

- **Risk 3:** Clearly define the deliverables before starting to build a system

## **7. Communication Plan**

### **7.1 Communication Methods**

- Methods:

- Team member Meetings

- Boardroom meetings with stakeholders

- Email updates

- Online Meetings on apps like Microsoft Teams

### **7.2 Frequency**

- Frequency:

- Weekly team member meetings

- Monthly meetings with stakeholders

- Monthly progress reports

## **8. Quality Management Plan**

### **8.1 Quality Objectives**

**Objectives:**

- Ensure system meets requirements

- Plan contractor monthly claim system with team members and stakeholders

- Maintain high code quality

### **8.2 Quality Assurance**

**Quality Assurance Activities:**

- Code reviews

- Testing (unit, integration, system)

### **8.3 Quality Control**

**Quality Control Activities:**

- 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

### **9.2 Change Log**

**Change Log:**



## **10. Project Closure**

### **10.1 Final Deliverables**

**Deliverables:**

- Final system release –> 24 October 2024

### **10.2 Lessons Learned**

- **Lessons Learned:**

- **Lesson 1:** I have learned that the project plan and process document is just as important as the prototype of a system.

- **Lesson 2:** I have learned time management while I was working on this part of the project.

- **Lesson 3:** I have learned expend my knowledge on different aspects of project management and leadership.

### **10.3 Project Review**

**Review Activities:**

- Project debrief -> In conclusion, the system that I will build a contractor monthly claim system that will enable independent contractor lecturer (IC lecturer) to submit their monthly claims, view their claim status and allow contractors to respond to the lecturers’ claims. Both contractors and independent contractor lecturers will operate on this single system even though they will each be executing different activities.

# **Reference List:**

A. Neumeyer.2024. Here’s my Change Log Template for Project Managers,©2024,tactical project manager.[Blog].Available at: <https://www.tacticalprojectmanager.com/change-log-template/> [Accessed on 5 September 2024].

Gantt.2024. What is a Gantt Chart?,©2024.[Online].Available at: <https://www.gantt.com/> [Accessed on 5 September 2024].