Contract Monthly Claim System (CMCS)

Documentation

1. Design choices: Goals and requirements

Streamlining and Submission: I want to make it easy for lecturers who work as independent workers to send in their monthly claims.

Approval Workflow: Make sure that claims are approved quickly after they are sent or submitted.

Monitoring and reporting: Provide services to track the status of claims and create reports.

1. Assumptions

There are two main types of users which are independent contractor lecturers and administrators' staff.

To protect sensitive data, access limits are important so there will be security.

The system needs to be bound by all applicable financial and data security laws, so there will be compliance.

Database structure

1. Tables:

Users:

* UserID(primary key)
* Username
* Password\_hash
* Role(lecture,admin)
* email

Claims:

* ClaimID(primary key)
* UserID(foreign key)
* Submission\_date
* Month
* Year
* Amount\_requested
* Status(pending, Approved, rejected)
* Description
* Created\_at
* Update\_at

Approvals:

* ApprovalID(primary key)
* ClaimID(foreign key)
* ApprovelID(foreign key)
* Approcal\_date
* Approval\_status(approved,rejected)
* Comments
* Created\_at

Audit\_Log:

* LogID(primary key)
* UserID(foreign key)
* Action
* Timestamp
* Details

1. Relationships:

A user can multiply claims.

Each claim have multiple approval records.

Each approval record is tied to a specific claim and approver.

GUI layout

My GUI layout will look like this:

1. Login Screen:

* Fields: Username, password
* Buttons: login, forgot password
* Links: register (for new users)

1. Dashboard:

* Navigation menu: links to home, submit claim, view claims, reports, profile, logout
* Overview section: Summary of pending claims, approved claims, and recent activity.

1. Submit claims:

Form fields:

* Month
* Year
* Amount requested
* Description
* Buttons: Submit, save as draft
* Validation: ensure all required fields are filled and data is in the correct format.

1. View claims:

* Filters: Date range, status (pending, approved, rejected)
* Table columns: claim ID, Data submitted, Amount requested, status, actions (view, edit, delete)
* Buttons: Approved, rejected (for approvers)

1. Reports:

* Types of reports: Claims summary, Approval status, monthly claims overview
* Filters: Date range, status
* Export options: PDF, Excel

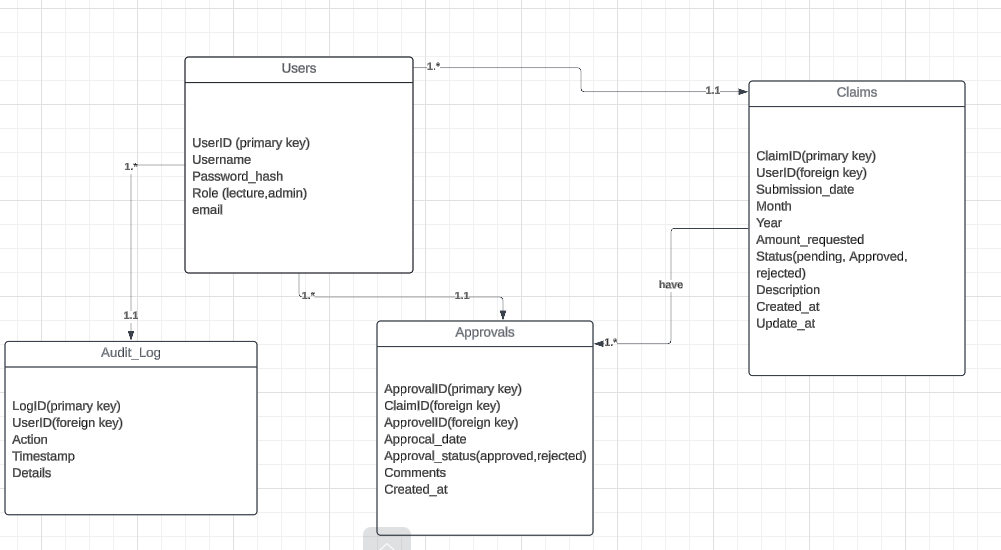
1. Profile:

* Fields: Username, email, role
* Buttons: edit profile, change password

1. Notifications:

* Alert: Upcoming deadlines, approval status updates
* Messages: System-generated notifications

2 UML diagram



3. Project plan

System development project plan

1.1. Project Name

Name: Contract Monthly Claim System

1.2. Project description

Description: The Contract Monthly Claim System is a web-based application designed to streamline the process of submitting, reviewing, and approving monthly claims for independent contractor lecturers. This system aims to enhance efficiency and accuracy in the claims management process, ensuring that independent contractors receive timely payments while providing administrative staff with robust tools for monitoring and managing claims.

1.3. Project objective

Objective 1: Boost the Effectiveness of Claims Submission

Objective 2: Automate Approval Workflow

Objective 3: Ensure System Security and Data Protection

1.4. Project scope

In scope:

System design and development

Claims submission

Reporting and analytic

User Management

Training and support

Compliance and Auditing

Intergration

Out of scope:

Non-web-based platforms

External system modifications

Custom integration

Advanced analytics

Internationalization

Hardware requirements

Post-implementation support

Regulatory compliance beyond data protection

1.5. Deliverables

Deliverables 1: System design documentation

Deliverables 2: Development deliverables

Deliverables 3: Testing and quality Assurance

1.6. Assumptions and Constraints

List of Assumptions:

* User Technology and Access
* System integration
* Data Accuracy and Integrity
* Regulatory compliance
* Support and training
* System Maintenance
* Project Stakeholders

List of Constraints:

* Budget and resources
* Timeline
* Technology stack
* User Experience
* Integration complexity
* Regulatory changes
* Internationalization
* Hardware and infrastructure
* Long-term support

2. Project Organization

2.1. Stakeholders

Stakeholders 1: Independent Contractor Lecturers

Stakeholders 2: IT and Development Team

Stakeholders 3: Project Managers

2.2. Project Team

Project manager: Carol Ravele

Team Members:

* Business analyst – Edwin Cadwin
* Software Developers - Rinae Magadagela
* Database Administrator – Azwindini

2.3. Roles and responsibilities

Roles and responsibilities:

* Business Analyst: They gather and analyze business requirements from stakeholders
* Software developers: They develop and implement the systems' front-end and backend components.
* Database Administrators: They design, implement and manage the database structure.

3. Project phases and milestones

3.1.

Project chatter

Project name: Contract Monthly Claim System

Project date: 30 October 2024

Project end date: 15 February 2025

Project Manager: Carol

Project Purpose and Justification:

The Contract Monthly Claim System aims to streamline the process of submitting, approving, and managing monthly claims for independent contractor lecturers.

Project Objectives:

* **Enhance Efficiency in Claims Submission**
* **Automate Approval Workflow**
* **Improve Claims Tracking and Management**
* **Facilitate Comprehensive Reporting and Analytics**
* **Ensure System Security and Data Protection**

Scope of work:

In- scope:

* Development of a web-based system for claims submission, approval, and management.
* Design and implementation of a relational database.
* Creation of user interfaces for different roles (Lecturers, Approvers, Admins).
* Development of reporting and analytics features.
* Implementation of security measures and user authentication.
* Provision of training and user documentation.

**Out-of-Scope:**

* Development of mobile applications or desktop software.
* Custom integrations with third-party systems not specified.
* Advanced analytics or machine learning features.
* Support for multiple languages or extensive internationalization.
* Hardware provisioning or extensive infrastructure changes.

Deliverables:

* **System Design Documentation:** Architecture, database schema, and UI/UX design.
* **Web Application:** Frontend and backend components, including claims submission, approval workflows, and reporting tools.
* **Testing and Quality Assurance:** Test plans, test cases, and results.
* **User Documentation and Training:** Manuals, help guides, training materials, and training sessions.

Milestones:

* **Project Kickoff**
* **Requirements Gathering Completed**
* **System Design Finalized**
* **Development Completed**
* **Testing and QA Completed**
* **Training and Documentation Completed**
* **System Deployment and Go-Live**
* **Post-Implementation Review**

Budget:

* **Estimated Total Budget:** [R180 000]
* **Breakdown:**
* Development Costs: [50 000]
* Design Costs: [25 0000]
* Testing and QA Costs: [30 0000]
* Training and Documentation Costs: [50 000]
* Deployment and Integration Costs: [25 000]
* Contingency: [10 000]

Project Team:

* **Project Manager:** [Rinae]
* **Business Analyst:** [Carol]
* **System Architect:** [Azwindini]
* **Software Developers:** [Edwin]
* **UI/UX Designer:** [Zwashu]
* **Quality Assurance Tester:** [Akonaho]
* **Database Administrator:** [Mulweli]
* **Security Specialist:** [Murendwa]
* **Technical Writer:** [Faith]
* **Training and Support Specialist:** [Anza]

Stakeholders:

* **Independent Contractor Lecturers**
* **Approvers**
* **Administrative Staff**
* **IT and Development Team**
* **Project Managers**
* **Financial and Compliance Officers**
* **End-Users**
* **Executive Stakeholders**

**Risks and Mitigation:**

1. **Integration Challenges:** Risk of difficulty integrating with existing systems. *Mitigation:* Conduct thorough integration testing and have contingency plans.
2. **Budget Overruns:** Risk of exceeding the budget. *Mitigation:* Monitor expenses regularly and manage scope carefully.
3. **Timeline Delays:** Risk of project delays. *Mitigation:* Maintain a detailed project plan and address issues promptly.
4. **Security Vulnerabilities:** Risk of data breaches. *Mitigation:* Implement strong security measures and conduct regular security assessments.

**Approval and Sign-Off:**

* **Project Sponsor**
* **Project Manager**
* **Business Analyst**
* **System Architect**

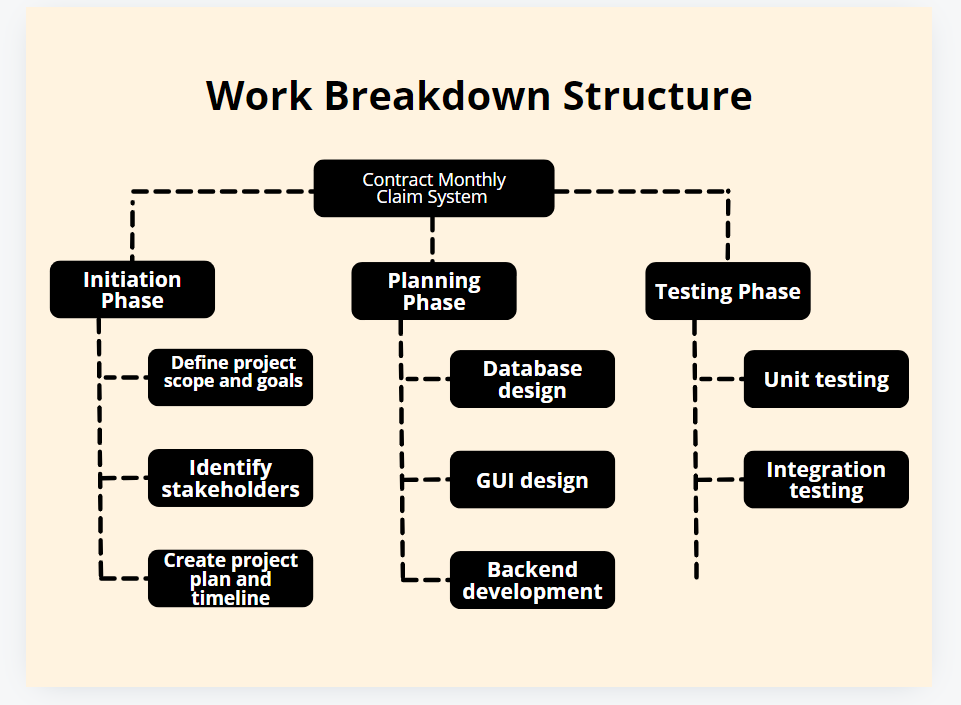
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Work breakdown structure



4. Project Schedule

4.1.

A screenshot of a calendar

Description automatically generated

<https://rinae-carol.monday.com/boards/1623054076/views/18956263>

4.2.key dates

Start date: 30 October 2024

End date: 15 February 2025

5. Budget and resources

5.1 Budget

Estimated budget: 180 000

Developing: R85 000

Licensing fees: 25 000

Hardware services: 50 000

5.2. Resource Allocation

Resources Required:

Human resources: project manager

Technological resources: Development Tool

Infrastructure: cloud/servers services

6. Risk Management

6.1. Risk identification

Risks:

Technical risks

Project Management risks

Resource risk

6.2. Risk Mitigation Plan

Mitigation strategies:

Technical risk: integration testing and backup solutions

Project Management risks: Scope Management and schedule management

Resource risk: Staffing plan and resource allocation

7. Communication Plan

7.1. Communication Methods

Methods:

Meetings

Email updates

Status report

7.2. Frequency

Frequency:

Weekly status meetings

Monthly progress reports

8. Quality management plan

8.1. Quality objectives

Objectives:

Ensure system meets requirements

Maintain high code quality

8.2. Quality Assurance

Activities:

Code reviews

Testing

8.3. 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

10. Project closure

10.1. Final deliverables

Deliverables:

Final system release

Documentation

10.2. Lessons learned

Lessons learned:

Lesson 1: I learned that poor communication could lead to misunderstanding and misalignment.

Lesson 2: I learned that failing to have stakeholders early can result in missed requirements

Lesson 3: learned that unidentified or poorly managed risks can derail a project.

10.3. Project Review

Review Activities:

Conduct projects debrief

Archive project documents.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated