

UNIVERSITY OF DAR ES SALAAM



COLLEGE OF INFORMATION AND COMMUNICATION TECHNOLOGIES

(CoICT)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

IS335: FINAL YEAR PROJECT MID OF SEMESTER 1 REPORT 2023/2024

PROJECT TITLE: CONSTRUCTION BUDGET MANAGEMENT SYSTEM

Student Name:	Ismail, Twalibu M	2021-04-02595	Computer Science
	Aidan Yusto F	2021-04-01820	BIT
	Emmanuel Emmanuel G	2021-04-01709	Computer Science
	Mrisho, Hussein	2021-04-07895	Computer Science

Supervisor Name: Dr. Aron Kandoro

Supervisor Signature:

Table of Contents

UNIVERSITY OF DAR ES SALAAM	1
Chapter One	4
INTRODUCTION.....	4
General Introduction.....	4
Statement of the problem	4
Main objective	5
Specific objective	5
Project scope.....	6
Organizational structure of the report.....	6
Chapter Two.....	7
LITERATURE REVIEW	7
Introduction	7
Related existing works	7
Project Gap	9
Chapter Three	10
METHODOLOGY	10
Chapter Four	12
SYSTEM ANALYSIS AND DESIGN.....	12
Introduction	12
Requirement analysis	13
Use-Case.....	18
Database Design	21
Chapter 5.....	23
SYSTEM IMPLEMENTATION	23
Technology Used	23
Technology Description and Contribution	24
Implemented Features	24
User Registration.....	24
User login	25

Project Management Module	26
Expense Management Module	27
Approval Management Module.....	29
User Management Module	30

LIST OF ABBREVIATION

COICT College Of Information and Communication Technology

BOQ Bill of Quantities

Chapter One

INTRODUCTION

General Introduction

The construction Budget Management System is a web-based tool helps keep a good eye on money matters in construction projects. It carefully watches all payments, like paying workers and buying materials, making sure they stick to the approved budget. It works step by step to ensure every money transaction matches the project's budget, stopping any spending without permission and giving a full view of how funds are used. The reports it creates also show a clear picture of the project's money situation, making it easier to make good decisions and manage the project well.

Statement of the problem

The problem we are tackling is that in construction projects, there are issues like spending money without permission, not keeping track of expenses in real-time, and lack of clear communication about the budget. These problems often lead to project costing more than planned, getting delayed, and causing disagreements among the people involved. We need a solution that brings all aspects of budgets management together including tracking payments to workers, managing

materials, and making sure all expenses are approved before. Without such system, it's hard to manage money well, and this can be a big challenge for the success of construction projects. Our goal is to create a simple and effective solution that improves how construction budgets are handled.

Main objective

The main objective is to build a website application to provide a simple and effective solution that improves how construction budgets are handled.

Specific objective

Here is a detailed breakdown of the specific objectives considering the main goal of creating a Construction Budget Management System: -

1. To gather the system requirement in order to obtain user needs and expectations related to construction budget management.
2. Define the technical and functional specification of the system based on the gathered requirements.
3. Design the system to insure friendly user interfaces
4. Design a database structure that can efficiently store and retrieve the data related to budgets, projects, documents, users.
5. Develop both front and back ends based on the established requirements features, and specification of the system.
6. Perform the system testing for performance, security, and compatibility issues

Significance of the project

The project is important because it helps make thing clear and responsible in the construction industry. It focuses on being open, accountable, and managing money better. This means it can bring good changes to how things are currently done. The project is valuable because it aims to make financial processes clearer and more organized, ensuring that every move is easy to

understand and responsible. Overall, it's a positive effort to improve how money is handled in the construction sector.

Project scope

- The project scope sets the rules and limits for the system. It includes things like signing up users, approving expenses, keeping track of workers payments and purchases, assigning user roles, budget tracking, labor management, material procurement, reporting, user authentication and using a notification system. These are the specific areas the project will focus on and the things it will do. The scope helps everyone understand what the system will cover and what it won't. It's like a map that shows where the system will go and what it will do to help with construction budget management.

Organizational structure of the report

This project is organizing into three chapters such as chapter one, two and three. Chapter one consists of general Introduction, Statement of the problem, main objective, specific objectives, significance of the project, scope of the project and the organizational structure of the report. Chapter two consists of literature review and chapter three is about methodology used in this project.

Chapter Two

LITERATURE REVIEW

Introduction

The literature review is like the starting point of our project. It helps us understand how things are currently in the world of construction budget management. We look at what others have already done, check out any new ideas they have, and see how things are generally going in this field. The main aim of this chapter is to connect our project to the bigger picture of managing money in construction projects. It's like putting our project into the larger context of what's happening in the world of construction finances.

Related existing works

Many people have worked on managing construction projects and money systems, and they've shared useful ideas. The Bill of Quantities (BOQ) is not software but a document used in construction projects. Also, systems like Procore, Viewpoint, and Archdesk are well-known for handling projects, but there's still a missing piece for fully managing budgets. These systems often don't have the specific features needed to make sure money processes are clear and under control.

Procore;

Procore started in 2003 when Tooey Courtemanche, a construction project manager, wanted a better way to manage his projects. Frustrated with the challenges of using paper and spreadsheets, he decided to create a digital solution.

In the beginning, Procore was a small software tool designed to make project management in construction easier. Over the years, it grew as more construction teams realized how helpful it was. Procore became a go-to platform for organizing plans, communicating with teams, and keeping everything in one place.

Now, Procore is a big player in the construction software world, helping teams around the globe build and manage their projects efficiently. It's come a long way from its early days, always evolving to meet the changing needs of the construction industry.

How it works

Procore is like a digital toolbox for construction projects. Imagine it as a place where everyone working on a project can share information, like plans and schedules. It helps teams stay organized and communicate better.

Challenges users may face.

Learning Curve: It might feel a bit tricky at first because there are many things you can do with it. It requires a lot of time for the guest user to become familiar with it.

Cost: For smaller projects, the cost might be a bit high to afford.

Bill Of Quantities (BOQ);

It is a detailed document that lists and describes the quantities of materials, labor, and other items needed to complete a construction project. The BOQ is an essential part of the tendering and

bidding process, providing a basis for contractors to estimate project costs and submit competitive bids.

The Bill of Quantities (BOQ) is not a software but a document used in construction projects. It is a comprehensive list of items and works that need to be executed in a construction project, along with the corresponding quantities and unit rates.

Weakness of Bill of Quantities

While the Bill of Quantities (BOQ) is a valuable tool in construction projects, it does have some weaknesses and limitations. Here are a few:

Subject to Changes: The BOQ is based on the information available at the time of its preparation. Changes in project scope, design alterations, or unforeseen circumstances can occur during the construction phase, leading to variations that may not be adequately covered in the original BOQ.

Limited to Quantifiable goods: The BOQ is most effective for items that can be precisely quantified, such as materials and specific types of labor. It may not capture all project costs, especially those related to unforeseen conditions or complex work that cannot be easily quantified.

May Lack Details: Depending on the level of detail provided during the project planning phase, a BOQ may lack the granularity needed for certain construction activities. This can lead to uncertainties and disputes if not addressed.

Doesn't Address Quality: The BOQ typically focuses on quantities and may not explicitly address the quality standards required for construction materials and workmanship. Quality issues might arise if the specifications are not detailed enough.

Project Gap

While existing literature addresses various aspects of Construction Budget systems accountability, a notable gap exists in the cost structure of the existing systems in which some

systems are very expensive to adopt in the small construction projects, Lack of real-time tracking, and accountability within some existing solution like what is done in the Bill of Quantities.

The Construction Budget Management System addresses critical deficiencies in existing solutions, offering an affordable, real-time tracking, and accountable alternative. By doing so, it aims to revolutionize budget management in construction projects, regardless of size, contributing to more efficient and transparent construction processes.

Chapter Three

METHODOLOGY

Introduction

This outlines the methodological approach used for the development of the Construction Budget Management System. The main objective of the system is to provide a simplified and efficient solution for managing construction budgets in the industry. The development process followed an object-oriented programming (OOP) approach and incorporated various methods and tools to gather requirements, design the system, and develop the frontend and backend components.

Methodological Approach

The development of the Construction Budget Management System followed a structured approach that encompassed several key phases: requirements gathering, specifications, system design, database design, system development, testing, and documentation. Each phase was carefully planned and executed to ensure the successful implementation of the system.

Data Collection Methods

To gather requirements for the Construction Budget Management System, two primary methods were employed:

1. **Open Discussions:** Open discussions were conducted with stakeholders, including construction project managers, budget analysts, and other relevant personnel. These discussions helped identify pain points, requirements, and expectations for the system.
2. **Interviews:** Structured interviews were conducted with key stakeholders to obtain specific information about their workflows, processes, and desired system features. These

interviews provided valuable insights into the needs of the users and guided the development process.

System Design

The system design phase involved the use of the Unified Modeling Language (UML) to create visual representations of the system's structure and behavior. UML diagrams, including use case diagrams and entity relationship diagrams were used to illustrate the relationships between different components, the flow of actions, and the overall system architecture.

Database Design

The database design phase focused on designing a strong and efficient database schema for storing and retrieving data related to construction budgets. UML Entity-Relationship Diagrams (ERD) were employed to visualize the entities, attributes, relationships, and constraints of the database. This helped ensure data integrity and facilitate future scalability.

System Development

The system development phase followed an Agile development approach, specifically utilizing the Scrum methodology. Agile practices were implemented to promote collaboration, flexibility, and continuous improvement throughout the development process. The development team worked in short iterations called sprints, with each sprint lasting 1-2 weeks. User stories were used to define system requirements, and daily stand-up meetings were conducted to ensure effective communication and coordination.

The frontend of the Construction Budget Management System was developed using ASP.NET Core MVC, a web application framework. Visual Studio, an integrated development environment (IDE), was used for coding and debugging. The backend utilized C# for implementing the business logic and interacting with the database, which was managed using MS SQL Server.

Testing

Throughout the development process, thorough testing was conducted to ensure the functionality, performance, security, and compatibility of the system. This included unit testing, integration testing, and user acceptance testing. The testing phase aimed to identify and resolve any issues or bugs, ensuring a high-quality and reliable product.

Methodology Summary

Objective	Tools/Technologies	Methods
Data Collection Methods	Open Discussions, Interviews	Gather insights from stakeholders
System Design	UML	Visualize system structure and behavior

Database Design	UML ERD	Design database schema
System Development	Agile (Scrum)	User stories, sprints, daily stand-ups
Testing		Unit testing, integration testing, user acceptance testing

The development of the Construction Budget Management System followed a systematic and structured approach, incorporating OOP principles and utilizing various tools and technologies. The methodological approach involved gathering requirements through open discussions and interviews, designing the system using UML, developing the frontend and backend components using ASP.NET Core MVC and C#, and conducting thorough testing

Chapter Four

SYSTEM ANALYSIS AND DESIGN

Introduction

System analysis involves the collection and interpretation of requirements, the definition of concerns, and the breakdown of a system into its components. The goal is to ensure the effective operation of all system components in achieving their intended purpose by understanding user needs. Conversely, system design is a process that involves creating a new system or replacing an existing one by detailing its components or modules to meet specific specifications.

Requirement gathering

Is the process of using requirement elicitation techniques like questionnaire, interview and others to obtain requirement from different people who can be part of using a system. This will help developers to obtain many data that will contribute in system design and implementation easily and by knowing customer and user preferences. In our project Construction Budget management system we used to interview Civil Engineering students to obtain data.

We asked them about the method for estimation of the budget and tracking of all expenses incurred during the construction process and the response was the Bill of Quantities(BOQ).

According the responses we received we discovered that there was no existing software that would simplify the two main activities, and then we asked if the existence of Construction Budget Management system would be helpful and the response from them was absolutely Yes. Then we asked for any recommendations to our project from the Civil Engineering because they the main target users of our system.

Requirement analysis

Requirement analysis stands as a critical phase in the development lifecycle, ensuring that the final product aligns with the needs and expectations of its users. This process involves systematically collecting and documenting information about the system's objectives, functionalities, and user interactions. Below we have functional and non-functional requirement of our project, Construction Budget Management System.

Functional requirement

These functional requirements aim to address the identified problems and objectives', ensuring the Construction Budget Management System is comprehensive, secure, and user-friendly for effective financial management in construction projects.

Ref No.		Functional Description	Category
F1		Management of users	
	F1.1	System should allow users to register and login	Evident
	F1.2	System should allow users to edit their profile or details	Evident

F2		Budget Tracking	
	F2.1	A user should be able to create Project Budget, Access budget creation module, Input project details and allocate funds, Save and finalize the project budget.	Evident
	F2.2	A user should be able to Update a Budget, navigate to budget management section, identify budget items to be updated, adjust budget allocations based on changes, Save and update the budget.	Evident
F3		Expense Approval Workflow	
	F3.1	Initiate Expense Approval Request, log in to the system, navigate to the expense approval module, fill in details: purpose, amount, and supporting documents, Submit the expense approval request.	Hidden
	F3.2	Review Expense Details, Access the expense approval module, View details: purpose, amount, supporting documents.	Evident
	F3.3	Approve or Reject Expense, based on the review, a user chooses to approve or reject. If approving, specify conditions or notes, If rejecting, provide reasons for rejection, Submit the final decision.	Evident
F4		Payment Processing	
	F4.1	Initiate Payment Request, log in to the system, Access the payment processing module, provide details for the payment request, including payee, amount, and purpose.	Evident

	F4.2	Review Payment Details, Access the payment processing dashboard, Review details of pending payment requests, including amounts and supporting documentation	
	F4.3	<p>Authorize or Reject Payment, based on the review, authorize or reject the payment request.</p> <p>If authorizing, specify payment details and confirm.</p> <p>If rejecting, provide reasons for rejection</p>	
	F4.4	<p>Generate Payment Order, for authorized payments, generate a payment order for processing.</p> <p>Include relevant information such as payee details, amount, and payment method</p>	
	F4.5	<p>Execute Payment, Initiate the payment transaction through the chosen payment method (e.g., bank transfer, check issuance).</p> <p>Ensure compliance with approval and authorization processes.</p>	
	F4.6	<p>Record Payment Transaction, Log the completed payment transaction in the system.</p> <p>Update relevant project and budget records with the payment details.</p>	
	F4.7	<p>Receive Payment Confirmation, Receive confirmation of successful payment processing.</p> <p>Update payment status in the system.</p>	
	F4.8	Handle Payment Rejections, in case of payment rejection (e.g., insufficient funds), follow the defined procedure for	

		resolution. Update the system with details of the rejected payment.	
F5		Reporting and Analytics	
	F5.1	Select Report Type, Choose the type of report needed, such as budget breakdown, expense summary, or variance analysis.	Hidden
	F5.2	Define Report Parameters, specify parameters for the report, including project selection, date range, and specific budget categories.	Evident
	F5.3	Data Retrieval, retrieve relevant data from the database based on the defined parameters for the selected report	
	F5.4	Data Processing, Process raw data to aggregate, filter, or transform it as required for the specific report	
	F5.5	Generate Report, utilize reporting tools to generate the selected report format, ensuring it meets predefined criteria.	
	F5.6	Review and Validate Report, Review the generated report for accuracy, completeness, and adherence to predefined standards, Validate the report against expected outcomes.	

Table 1: Functional Requirements

Non-functional requirement

Non-functional requirements define the characteristics and qualities of the system, specifying how the system should perform rather than what it should do. Here are some non-functional requirements for the Construction Budget Management System:

These ensure that the Construction Budget Management System not only meets functional expectations but also delivers a reliable, secure, and user-friendly experience.

SN	Non-functional Requirement	Description
1	Performance	-Response Time: The system should provide quick response times for user interactions, ensuring a seamless and responsive user experience
2	Security	<ul style="list-style-type: none"> - Data Encryption: All sensitive data, including user credentials and financial information, should be encrypted during transmission and storage. - Access Control: Implement strict access controls to ensure that users can only access data and functionalities relevant to their roles and responsibilities.
3	Scalability	<ul style="list-style-type: none"> - Capacity Planning: Regularly assess and plan for increased system capacity to handle growing data volumes and user loads. - Resource Utilization: Optimize resource utilization to ensure efficient use of server resources and minimize bottlenecks
4	Reliability	<ul style="list-style-type: none"> - Availability: The system should be available 24/7, with minimal downtime for maintenance or updates. Availability should be at least 99.9%. - Fault Tolerance: The system should be resilient to failures, and it should have mechanisms in place to recover gracefully from unexpected errors.
5	Usability	<ul style="list-style-type: none"> - User Interface Consistency: Maintain a consistent and intuitive user interface design across all modules and screens. - Accessibility: Ensure the system is accessible to users with disabilities, complying with relevant accessibility standards.

6	Compatibility	<p>- Browser Compatibility: The system should be compatible with popular web browsers (e.g., Chrome, Firefox, Safari, Edge) to accommodate diverse user preferences.</p> <p>- Device Compatibility: The system should be accessible and usable across various devices, including desktops, tablets, and mobile phones.</p>

Table 2: Non - functional Requirements

Use-Case

Admin's use-cases

1. User login
2. Manage users
3. User profile management

Technician's use-case

1. User login
2. Manage tools
3. User profile management
4. Review expensive details

Director's use-cases

1. User login
2. User profile management
3. Reporting analysis
4. Expenses approval workflow
5. Reviews expensive details

Project manager's use-case

1. User login
2. User profile management
3. Review expensive details
4. Manage project
5. Generate report

Cashier use-case

1. User login
2. User profile management
3. Payment processing

Finance manager's use-case

1. User login
2. User profile management
3. Manage budget

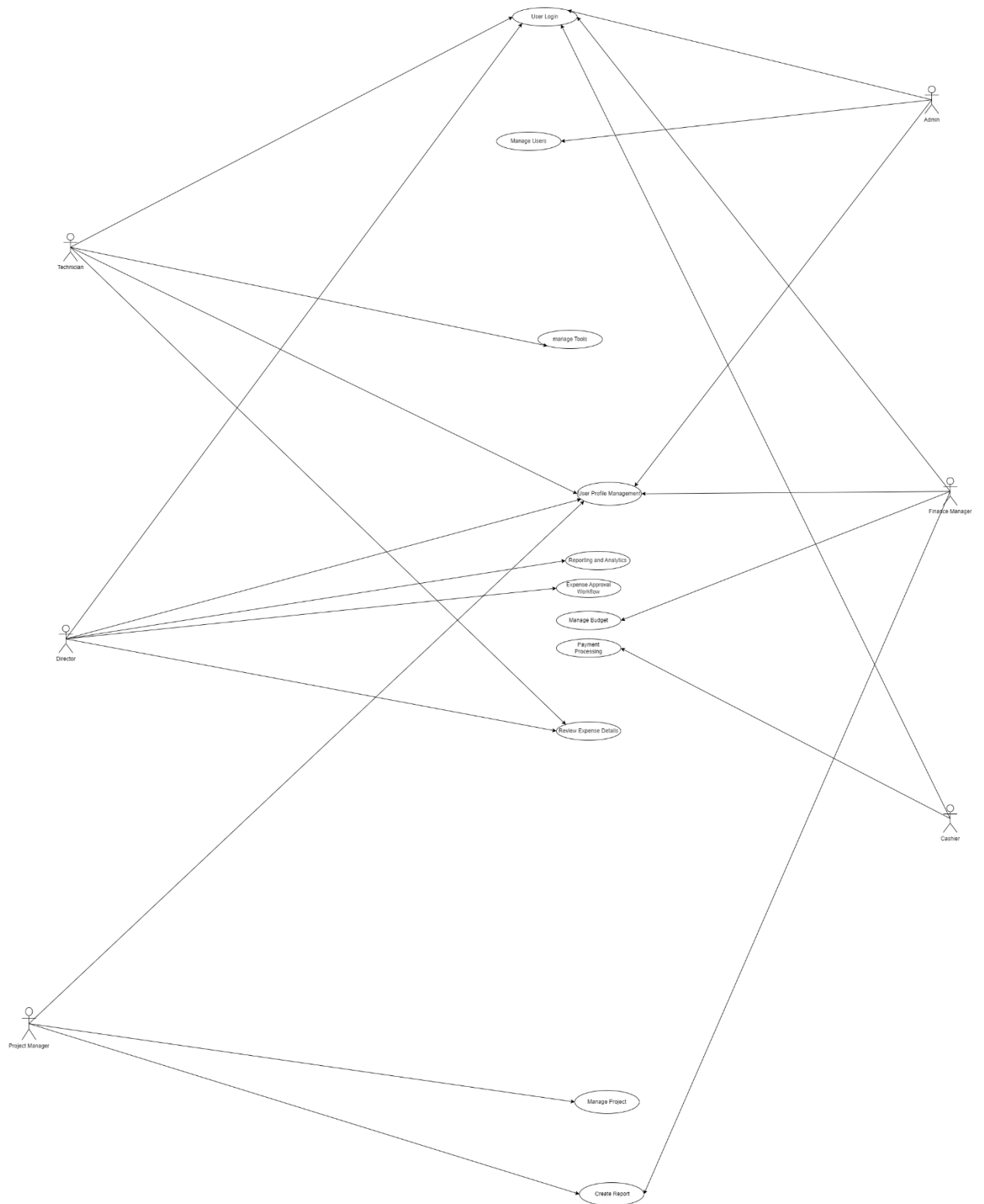


Figure 1: Use - Case Diagram

Database Design

In this project, Construction Budget Management System we are going to use Microsoft SQL Server with database name “ConstructionDB”. The below is the Entity Relational Diagram (ERD) which visually represents the database schema of our project.

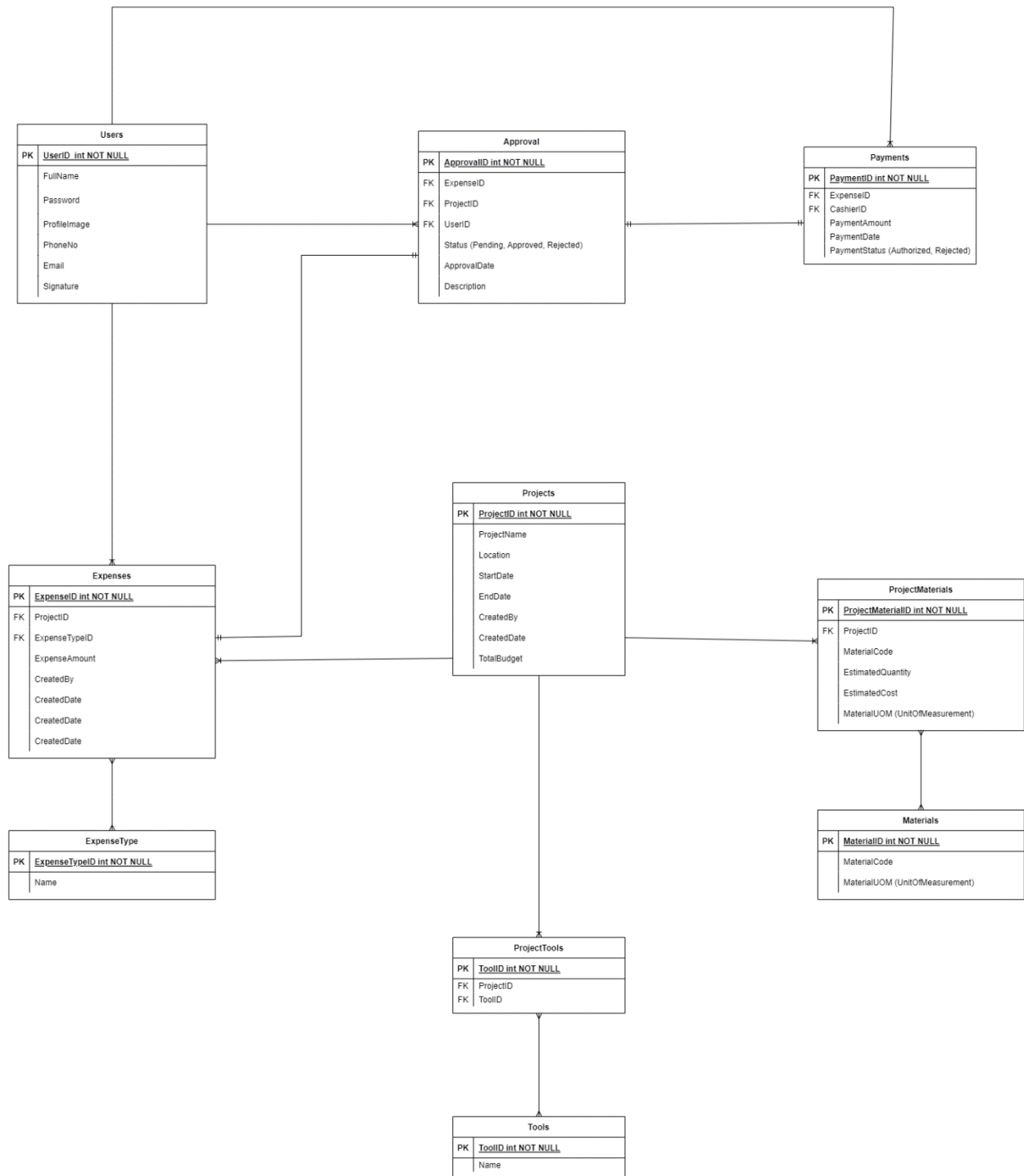
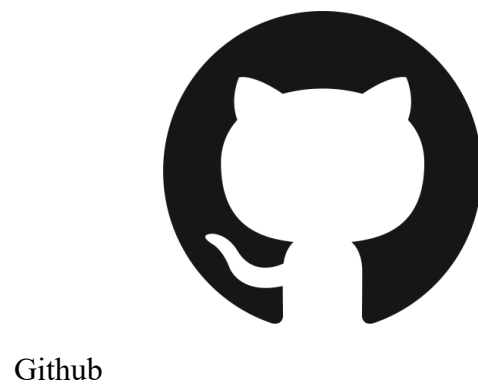
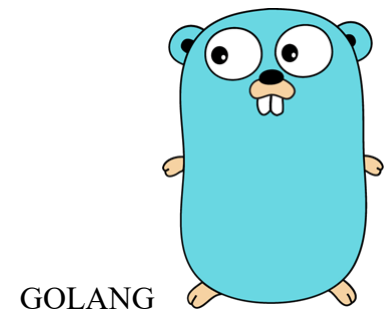
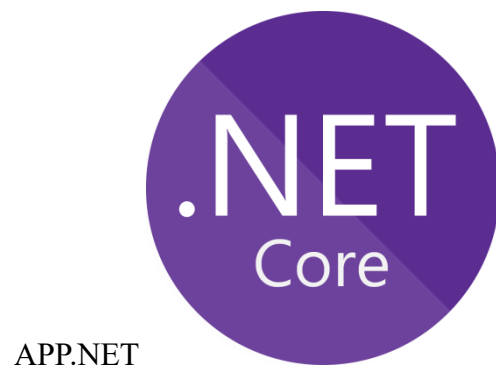


Figure 2: Entity Relationship Diagram

Chapter Five

SYSTEM IMPLEMENTATION

Technology Used



Technology Description and Contribution

MSSQL, Microsoft SQL Server (Structured Query Language) is a proprietary relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications—which may run either on the same computer or on another computer across a network (including the Internet).

This facilitate sql query around the different kinds of data populated with the our system for the easy retrieve and storage even it complex query.

Github, is a developer platform that allows developers to create, store, manage and share their code. It uses Git software, providing the distributed version control of Git plus access control, bug tracking, software feature requests, task management, continuous integration, and wikis for every project.

That facilitates remote working across the local area and consistence data storage over its cloud infrastructure.

Golang, is a statically typed, compiled high-level programming language designed at Google.

As platform independent language so that can works across the all kind of devices, Linux, window and IOS.

That offer powerful concurrence features of working with live API between the system and external ends with high scalability and stability due to its go-routines features and concurrence as its backbone.

Implemented Features

User Registration

User registration can be initiated by the admin, who sends the login password to the user via email. The system ensures secure access by requiring users to log in using their email as the username and the provided password.

Registration Form

FirstName

LastName

Email

PhoneNumber

Signature

Choose File

No file chosen

Role

Select Role

Can Approve

☐

Request For Someone

☐

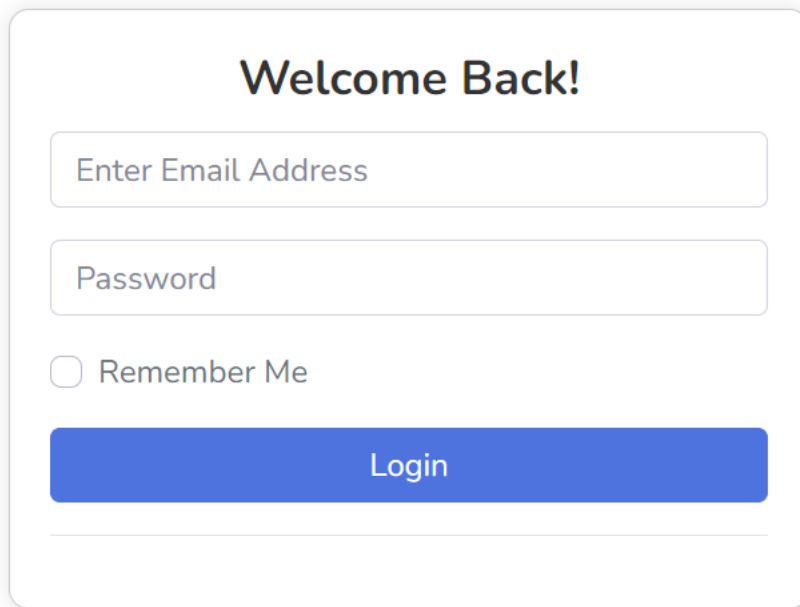
Register

[← Back To List](#)

Figure 3: Registration Form

User login

A user will be required to enter a username (email registered) and a valid password. It is not valid if a user enters a valid username/password with a wrong username/password.



A login form titled "Welcome Back!". It features two input fields: "Enter Email Address" and "Password". Below these is a checkbox labeled "Remember Me". At the bottom is a blue "Login" button. The form is enclosed in a light gray rounded rectangle with a subtle drop shadow.

Welcome Back!

Enter Email Address

Password

☐ Remember Me

Login

Figure 4: Login Form

Project Management Module

The system allows users to create, edit, view, and delete projects. Users can also access project details, including associated materials and tools. Additionally, users have the ability to manage materials and tools specific to each project, including creating, editing, deleting, and viewing them. This can be done by the user who has the access on project module where each role has its permission to access the resources.

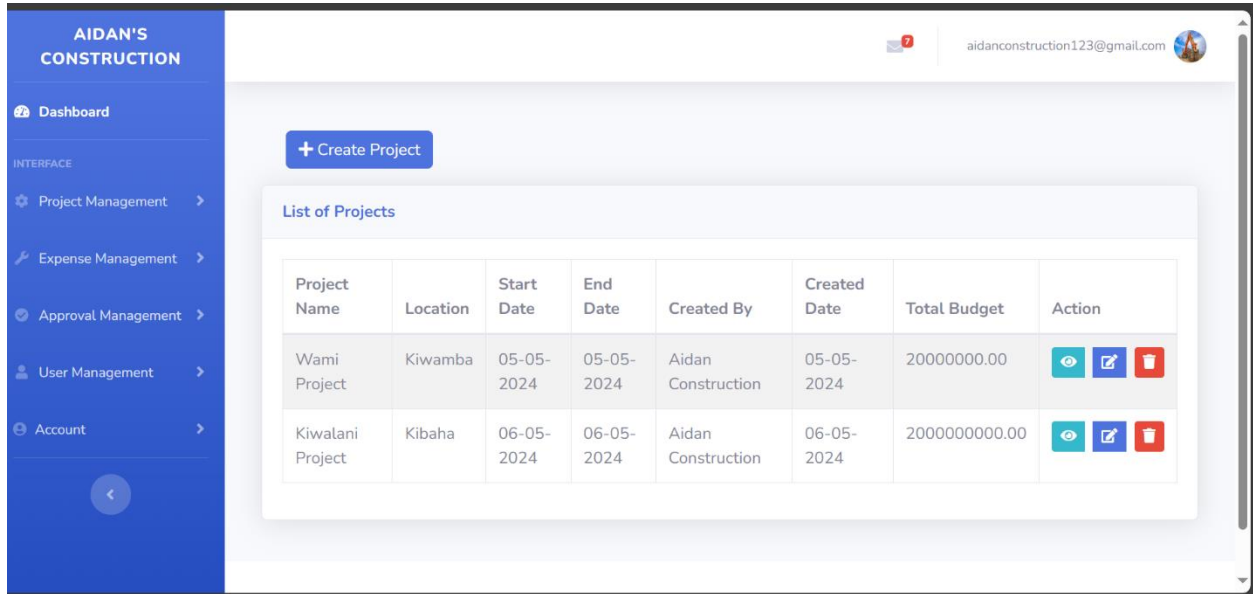


Figure 5: Projects Lists

Expense Management Module

Users can create, edit, view, and delete expenses within the system. A form is available for creating new expenses, ensuring consistent data entry. Users can also send requests for approval of specific project expenses and expense types. The system includes the ability to manage expense types, allowing users to create, edit, view, and delete them. Users can view lists of approved, pending, and rejected expenses, providing transparency and tracking capabilities.

The 'Expense Form' interface contains the following fields and controls:

- Project:** A dropdown menu with the placeholder text 'Select Project'.
- ExpenseTypeId:** A dropdown menu with the placeholder text 'Select Expense Type'.
- Created By:** A text input field containing 'Aidan Construction'.
- Created Date:** A date input field containing '05/08/2024' with a calendar icon.
- Buttons:** A blue '+ Add Expense' button and a grey '< Back To List' button.

Figure 6: Request Form for Expense

Approval Management Module

The system facilitates the approval process for expenses. Users receive notifications for pending expense requests and can approve or reject them. In the case of rejection, users are required to provide a description explaining the reason. Users can also view lists of all rejected and approved requests, including descriptions, enabling them to track and review their own actions.

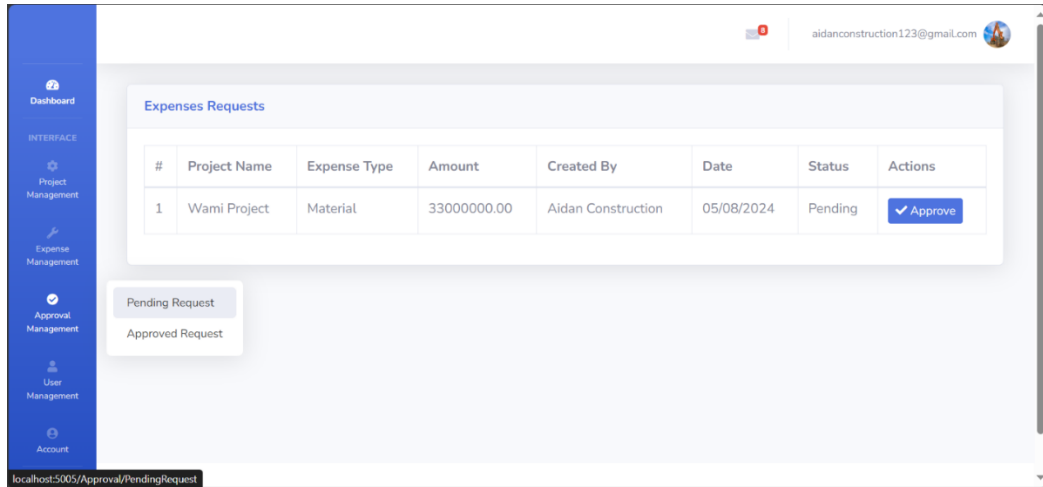


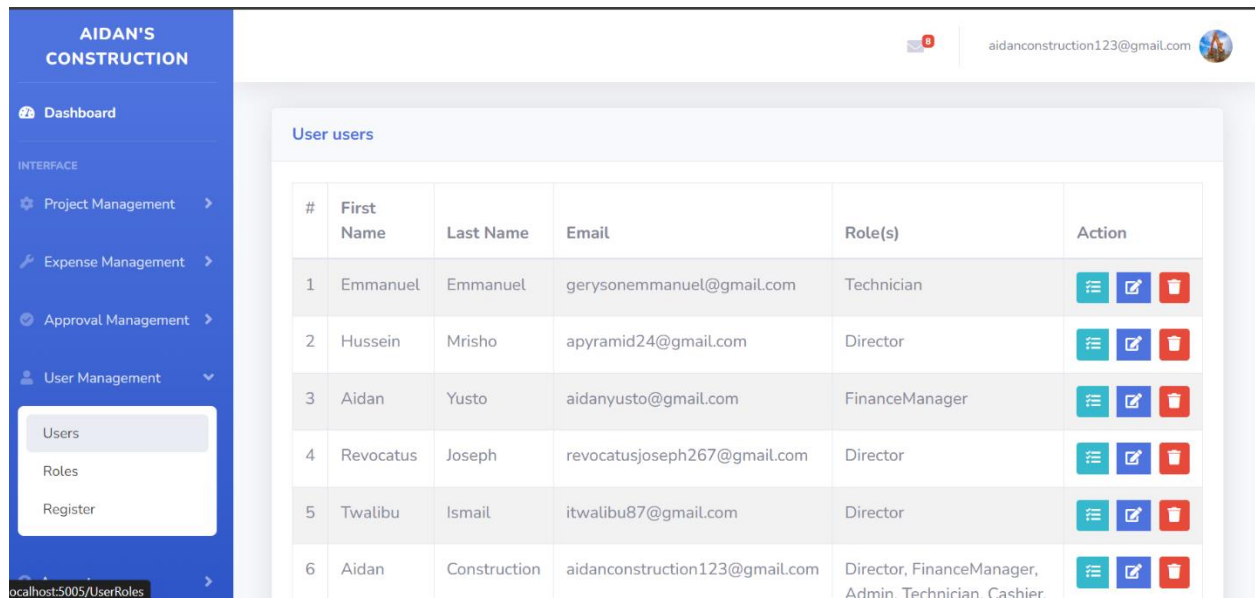
Figure 9: Expense Request Waiting for Approval

The screenshot shows a modal window titled 'Approve Expense' with a close button (X) in the top right corner. The form contains a 'Description' label followed by a large text input field. Below the input field are two radio buttons: 'Approved' and 'Rejected'. At the bottom right of the modal are two buttons: 'Close' and 'Approve'.

Figure 10: Form to Approve the Request

User Management Module

The system incorporates a role-based user management system. Each user is assigned one or more roles, with the admin having all roles and permissions. The admin can add, edit, delete, or view roles, as well as assign or remove roles for individual users. The system provides a comprehensive list of all users and their corresponding roles, allowing the admin to manage user access effectively.





















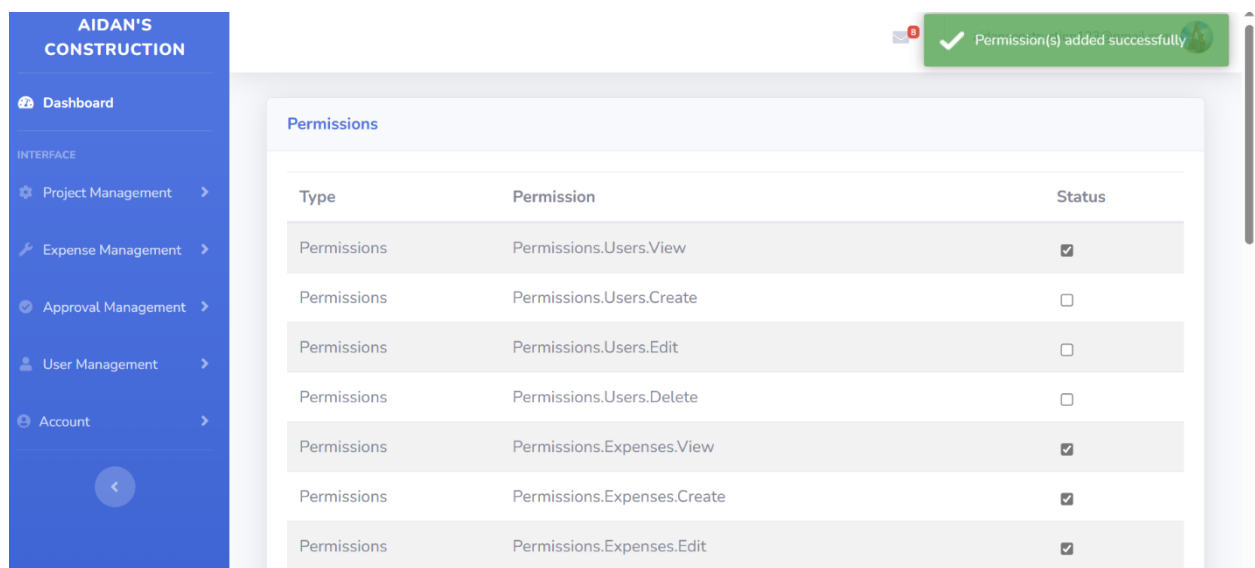
#	First Name	Last Name	Email	Role(s)	Action
1	Emmanuel	Emmanuel	gerysonemmanuel@gmail.com	Technician	  
2	Hussein	Mrisho	apyramid24@gmail.com	Director	  
3	Aidan	Yusto	aidanyusto@gmail.com	FinanceManager	  
4	Revocatus	Joseph	revocatusjoseph267@gmail.com	Director	  
5	Twalibu	Ismail	itwalibu87@gmail.com	Director	  
6	Aidan	Construction	aidanconstruction123@gmail.com	Director, FinanceManager, Admin, Technician, Cashier	  

Figure 11: All Users



Type	Permission	Status
Permissions	Permissions.Users.View	<input checked="" type="checkbox"/>
Permissions	Permissions.Users.Create	<input type="checkbox"/>
Permissions	Permissions.Users.Edit	<input type="checkbox"/>
Permissions	Permissions.Users.Delete	<input type="checkbox"/>
Permissions	Permissions.Expenses.View	<input checked="" type="checkbox"/>
Permissions	Permissions.Expenses.Create	<input checked="" type="checkbox"/>
Permissions	Permissions.Expenses.Edit	<input checked="" type="checkbox"/>

Figure 12: Permissions List

Dashboard

INTERFACE

Project Management

Expense Management

Approval Management

User Management

Account

<

Role Manager

Enter Role Name

Add New Role

#	Role Name	Action
1	Director	<div><div></div><div></div><div></div></div>
2	FinanceManager	<div><div></div><div></div><div></div></div>
3	Admin	<div><div></div><div></div><div></div></div>
4	Technician	<div><div></div><div></div><div></div></div>
5	Cashier	<div><div></div><div></div><div></div></div>
6	ProjectManager	<div><div></div><div></div><div></div></div>

Figure 13: Roles

References

archdesk. (2022, august 11). Retrieved from Construction Budget: a quick and comprehensive guide: <https://archdesk.com/blog/construction-budget-guide/>

Construction Software. (2023, january 03). Retrieved from Archdesk: <https://archdesk.com/blog/thebest-construction-budget-software-tools/>