scs 203

Project Documentation

By

Mugo John Mwangi – Sc212/1211/2017

BSc. Software Engineering

# FINANCE MANAGEMENT SYSTEM

Abstract

Acknowledgements

My acknowledgements first go to God almighty for the gift of life and continuous providence of health in my life. It is by his hand I have made it this far in this course. I also want to acknowledge Murang’a University of Technology with regards to Computer Science Department for shaping me and making me realize my skills in computing. The enlightenment and skills that I acquired have assisted me in coming with this project.

Finally, yet importantly, I want to acknowledge my friends and relatives for their moral, financial support throughout my course to these far.

Contents

[FINANCE MANAGEMENT SYSTEM 1](#_Toc5259901)

[Table of Figures 3](#_Toc5259902)

[CHAPTER 1: INTRODUCTION 4](#_Toc5259903)

[1.1. Background 4](#_Toc5259904)

[1.2. Objectives 4](#_Toc5259905)

[1.3. Purpose, Scope and Applicability. 5](#_Toc5259906)

[1.3.1. Purpose 5](#_Toc5259907)

[1.3.2. Scope 5](#_Toc5259908)

[1.3.3. Applicability 5](#_Toc5259909)

[CHAPTER 2: SURVEY OF TECHNOLOGIES 6](#_Toc5259910)

[CHAPTER 3: REQUIREMENTS AND ANALYSIS 7](#_Toc5259911)

[3.1. Problem Definition 7](#_Toc5259912)

[3.2. Requirements Specification 7](#_Toc5259913)

[3.2.1. Hardware Requirements 8](#_Toc5259914)

[3.2.2. Software Requirements 8](#_Toc5259915)

[3.3. Conceptual Models 8](#_Toc5259916)

[3.3.1. Use Case Diagram 8](#_Toc5259917)

[3.3.2. Sequence Diagram 9](#_Toc5259918)

[3.3.3. ER Diagram 11](#_Toc5259919)

[CHAPTER 4: SYSTEM DESIGN 12](#_Toc5259920)

[4.1. Basic Modules 12](#_Toc5259921)

[4.2. Data Design 20](#_Toc5259922)

[4.2.1. Schema Design 20](#_Toc5259923)

[4.2.2. Data Integrity and Constraints 22](#_Toc5259924)

[4.3. User Interface Deisgn 22](#_Toc5259925)

[4.5. Security Issues 25](#_Toc5259926)

[4.6. Test Case Design 25](#_Toc5259927)

[CHAPTER 5: IMPLEMENTATION AND TESTING 26](#_Toc5259928)

[5.1. Implementation Approaches 26](#_Toc5259929)

[5.2. Coding details and Code Efficiency 27](#_Toc5259930)

[5.2.1. Code Efficiency 27](#_Toc5259931)

[5.2.2. Testing Approach 27](#_Toc5259932)

[5.2.3. Unit Testing 27](#_Toc5259933)

[5.2.4. Integrated Testing 28](#_Toc5259934)

[5.3. Modifications and Improvements 28](#_Toc5259935)

[CHAPTER 6: CONCLUSIONS 28](#_Toc5259936)

[6.1. Conclusions 28](#_Toc5259937)

[6.2. Limitations of the Sytem 28](#_Toc5259938)

### Table of Figures

1. Use case diagram ………………………………………… 9
2. Sequence diagram …………………………………..........10
3. ER Diagram ………………………………………………11

## CHAPTER 1: INTRODUCTION

### Background

With the introduction of the devolved government in Kenya, the responsibilities and obligations of the national government were split, with some being delegated to the county governments. These brought down the central management that was prior available due to the National government.

The resources that were previously managed by the central government were also shared out equally to the county governments. These resources include tax collections and various funds meant to improve and better the living conditions of the citizens. However, this proved not to be the case as this lack of central management of resources opened the gate way for rampant and run-away corruption. The leaders who were elected to head these county governments and who were supposed to safeguard and protect the interests of the citizens however took this as an opportunity to reap huge benefits from their positions. They have taken advantage of the situation to swindle the citizens’ their due development from the taxes they pay.

1.2. Objectives**.**

This system is meant to remedy the financial hiccups in the county governments and smooth line the payment of county and national government employees.

The proposed remedy to the problem is a centralized computerized financial management system that is meant to handle all the county governments’ employee’s payment. All the duties from payment of employees’ salaries, deducting the taxes and forwarding them to KRA. This system is supposed to minimize the amount of people who come into contact with the finances and thus create accountability reducing corruption.

### 1.3. Purpose, Scope and Applicability.

### 1.3.1. Purpose

This project’s sole purpose is to simplify the financial management in the government. It is meant to try to root out and do away completely with the widespread corruption that is the characteristic of government financial activities.

### 1.3.2. Scope

The main function of this system is to handle financial tasks such as paying of employees and remitting tax to the KRA. However, there some assumptions made such as there is an up to date national database of all employees in the nation, each and every working person is registered with KRA and thus has a pin and that all other employees in the counties who are lacking in the national database are ghost employees.

### 1.3.3. Applicability

This system is to be used by the central government purposely to ease its financial tasks like paying the employees in the national government and those in the county governments and in remitting tax directly to the KRA.

## CHAPTER 2: SURVEY OF TECHNOLOGIES

The system is developed using Visual Basic. This is an object-oriented programming language and it is utilized against procedural programming languages because:

1. It enables modularization where the system is broken down into smaller easily maintainable modules.
2. The use of objects reduces complexity and makes the code easy to understand.
3. It is easily extensible if one wants to add new features by simply adding new objects and or modifying the existing ones.
4. It is easy to maintain as the objects can be changed separately without affecting other parts of the system
5. It enhances reusability as the objects can be used in different parts of the code.

There are other object-oriented programming languages such as java. However, visual basic DOT NET is used because:

1. Its structure is very simple, particularly the executable code.
2. Its IDE – Visual studio, is highly optimized to support rapid application development. The development of graphical user interfaces and connecting them to handler functions is simplified.
3. The graphical user interface of visual studio provides appealing views for management of program structure in different types of entities such as forms, classes, modules and procedures.
4. It has a comprehensive, interactive and context-sensitive online help system.
5. VB is a component integration language attuned to Microsoft’s Component Object Model (“COM”). Components of COM can be easily called remotely via Distributed COM, which makes easy to construct distributed applications.

The database used is Oracle’s MySQL. This is a widely used database in developing many solution databases. It was selected for this system because of the following reasons:

1. It is a secure and very reliable database management system. It is a privilege and password security system.
2. It is globally renown and an open source software which is freely available.
3. MySQL is compatible with various operating system and various programming languages.
4. It has a high performance as compared to other database management systems.
5. MySQL has a friendly user interface in workbench which makes it easy to use.
6. It is replicable and thus can be mirrored across various nodes. This decreases the workload and scalability and accessibility of the application can be expanded.
7. It supports automatic transaction support.

## CHAPTER 3: REQUIREMENTS AND ANALYSIS

### 3.1. Problem Definition

This project is meant to remedy the financial hiccups in the county governments and smooth line the running and delivery of services to the citizens. The challenges it is meant to solve are:

1. Lack of a central body to govern and control the use of resources.
2. Lack of accountability by the counties and national government’s officials.
3. Lack of proper financial management systems which creates loop holes for corruption and theft of public resources and finances.

### 3.2. Requirements Specification

In the current system, each county manages its own payment system and the national government employees are paid by it.

The new system ought to:

1. Allow for centralized control. The system should be managed by one person who bears all the responsibilities thus promotes accountability.
2. Allow for different IT managers in different counties to add or delete employees in the database.
3. Allow the admin to view the in the database, national and different county employees’ tables.
4. Allow admin to query the county table with the national database to figure out the ghost employees
5. Allow for submission of tax to the KRA through automatic deduction from the salary once they are paid

### 3.2.1. Hardware Requirements

For proper working and performance of the system, it ought to be installed on a computer with the following features:

1. At least 2.0 Giga Hertz of processing power.
2. Hard disk with at least 250Gb of storage.
3. 4Gb Ram.

### 3.2.2. Software Requirements

The system can run on various operating systems such as windows and Linux. However, the latest versions of these operating systems such as windows 10 are recommended. The version of MySQL that one uses with the operating system should also be compatible with the operating system.

The .NET framework is also required to compile the code into the CPU - independent language – Microsoft intermediate language.

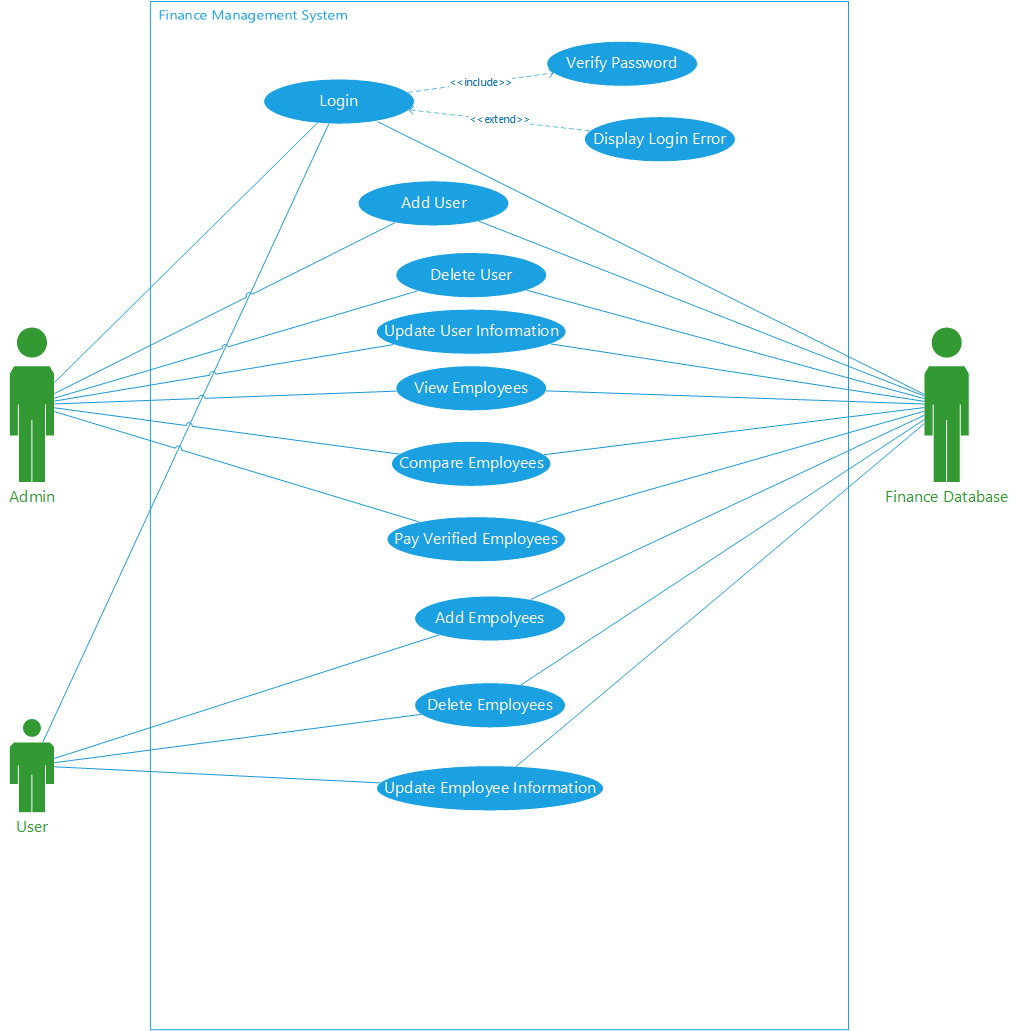
### 3.3. Conceptual Models

Some of the diagrams used in modelling, designing and developing the system are:

1. Use case diagram.
2. Sequence diagram.
3. ER diagram

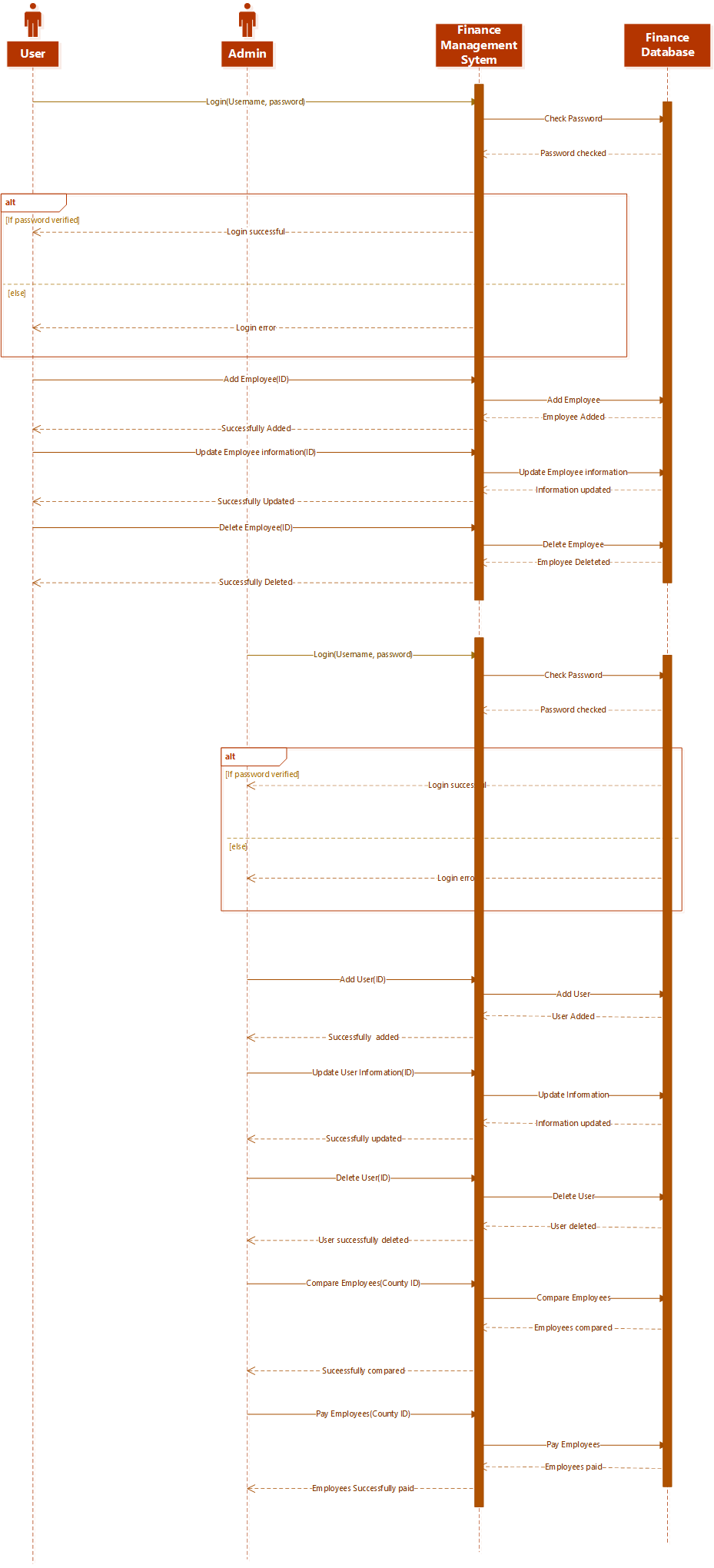
### 3.3.1. Use Case Diagram

It has modelled all the uses cases and scenarios that take place when the system is in use. It also shows the actors of the system, that is those who use the system.



### 3.3.2. Sequence Diagram

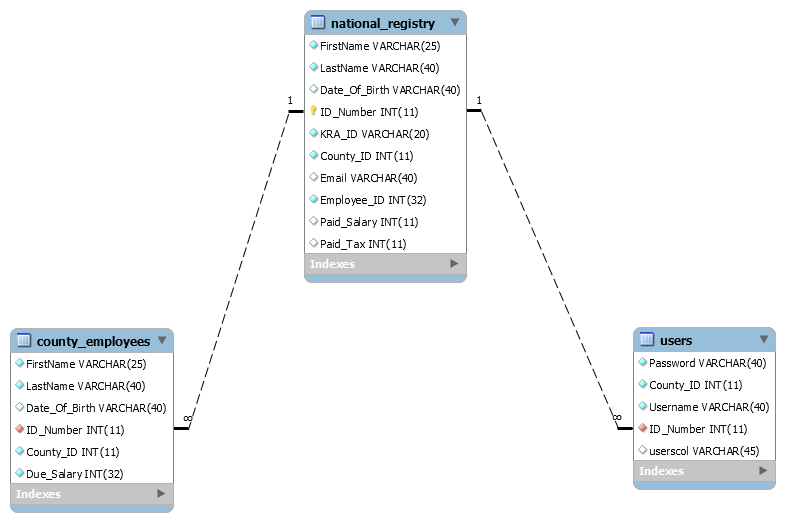
It is used to illustrate input and output events related to the system.



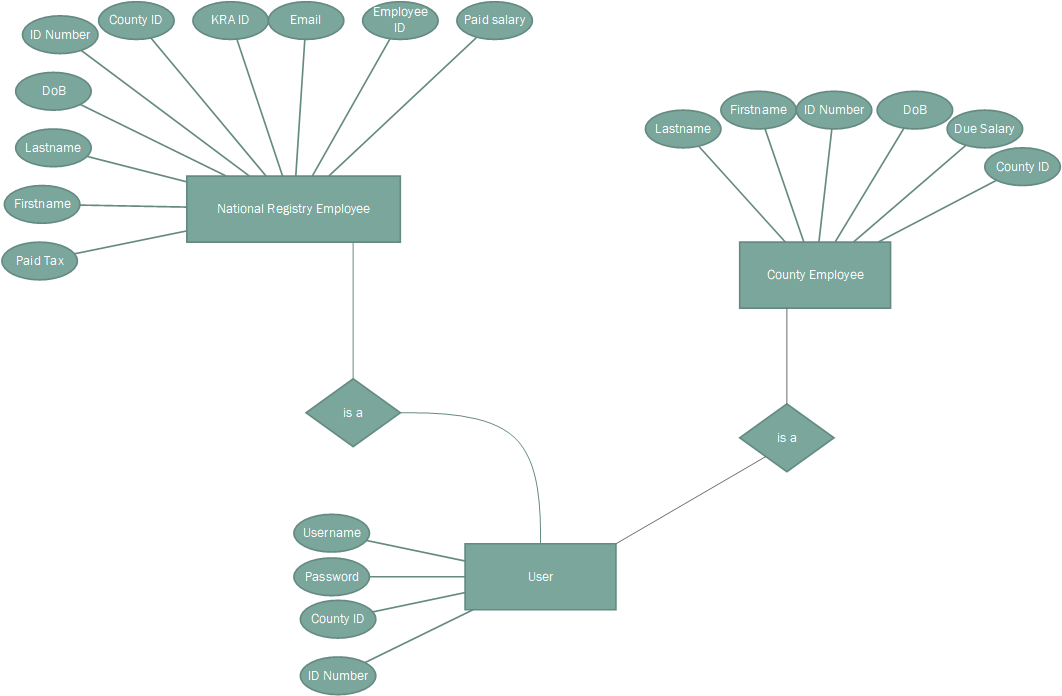
### 3.3.3. ER Diagram

This is a graphical representation of entities, attributes and their relationships in the database.

This is as generated by the database management system:



This is the drawn one:



## CHAPTER 4: SYSTEM DESIGN

The system is designed as follows:

1. It has a login page through which the different users and the admin are authenticated so as to be granted access to the system.
2. Once the different users are authenticated, each user is taken to a form responding to their county. Here, they can add and manipulated different employee information in their counties.
3. When the admin is authenticated, he is taken to an admin form where he can choose different options. These are:

a). Manage the different users. If he chooses this, he is taken to a different form where he can add, delete a user or change their information.

b). Manage employee information based on their county. On choosing this, he is taken to a form where he can compare the employees in the national registry against those in the selected county. He can view the employees in the national registry, those in the county and those who are in the county who are absent in the national registry who are then considered as ghost employees. The admin can also delete the ghost employee from database.

c). Pay the verified employees. Here, he can pay the county employees who are present in the national registry. He inputs the amount to be paid and the tax to be deducted as a percentage. The system then calculates each and then the banking system pays the salary into the employee’s account and the tax is remitted to the KRA.

Each of the forms has back button which leads back to the admin form.

1. Each form has an exit button which closes the program.

### 

### 4.1. Basic Modules

The system is divided into several modules for easy manageability. Some of the modules are as follows:

1. **The database connection module**

This module is used to create a connection to the database and to close the connection. Its code is:

Public Class Dbconnection

Dim Connect As New MySqlConnection("server= 'localhost';user id='root';database='Project';password='13115'")

Public Function Open() As MySqlConnection

Try

Connect.Open()

Catch ex As Exception

MsgBox(ex.Message)

End Try

Return Connect

End Function

Public Function Close() As MySqlConnection

Try

Connect.Close()

Catch ex As Exception

MsgBox(ex.Message)

End Try

Return Connect

End Function

End Class

1. **The login module**

This module is used to enforce security by allowing only authenticated users to have access to the system. It checks the username and password and compares them to those already stored in the database. If they match, the user is allowed access and vice versa. The code for the module is:

Private Sub BunifuLoginButton\_Click(sender As Object, e As EventArgs) Handles BunifuLoginButton.Click, MyBase.Enter

objdatapter = New MySqlDataAdapter("select Username, Password from users where Username= '" & BunifuUsernameTextbox.Text & "'AND Password='" & BunifuPasswordTextbox.Text & "'", myconnection.Open)

dtable.Clear()

objdatapter.Fill(dtable)

If dtable.Rows.Count = Nothing Then

MessageBox.Show("Please type a Valid Username or Password !", "Finance Management System", MessageBoxButtons.RetryCancel, MessageBoxIcon.Information)

BunifuUsernameTextbox.Text = ""

BunifuPasswordTextbox.Text = ""

BunifuPasswordTextbox.Focus()

ElseIf BunifuUsernameTextbox.Text = User Then

AdminForm.Show()

Me.Hide()

ElseIf BunifuUsernameTextbox.Text = User1 Then

NairobiForm.Show()

Me.Hide()

ElseIf BunifuUsernameTextbox.Text = User2 Then

NyeriForm.Show()

Me.Hide()

End If

myconnection.Close()

End Sub

1. **Add employee module**

It is used by the different county IT managers to add employees in their counties to the database. Its code is:

mysqlcon = New MySqlConnection

mysqlcon.ConnectionString =

"server= 'localhost';user id='root';database='Project';password='13115'"

Dim READER As MySqlDataReader

Dim message As String

message = MessageBox.Show("Are you sure you want to Add the Employee?", "Finance Management System", MessageBoxButtons.YesNo, MessageBoxIcon.Question)

If message = DialogResult.No Then Exit Sub

Try

mysqlcon.Open()

Dim Query As String

Query = "insert into project.county\_employees (FirstName, LastName, Date\_Of\_Birth, ID\_Number, County\_ID, Due\_Salary) values ('" & FirstNameTextbox.Text & "','" & LastNameTextbox.Text & "','" & DOBTextbox.Text & "','" & IDTextbox.Text & "','" & CountyTextbox.Text & "','" & SalaryTextbox.Text & "')"

Cmd = New MySqlCommand(Query, mysqlcon)

READER = Cmd.ExecuteReader

MessageBox.Show("New Employee Added!", "Finance Management System",

MessageBoxButtons.OK, MessageBoxIcon.Information)

mysqlcon.Close()

Catch ex As MySqlException

MessageBox.Show(ex.Message)

Finally

mysqlcon.Dispose()

End Try

1. **Update employee information module**

This module is used to update the information of those employees already in the database. Its code is:

mysqlcon = New MySqlConnection

mysqlcon.ConnectionString =

"server= 'localhost';user id='root';database='Project';password='13115'"

Dim READER As MySqlDataReader

Dim message As String

message = MessageBox.Show("Are you sure you want to Update the Employee Information?", "Finance Management System", MessageBoxButtons.YesNo, MessageBoxIcon.Question)

If message = DialogResult.No Then Exit Sub

Try

mysqlcon.Open()

Dim Query As String

Query = "update project.county\_employees set FirstName='" & FirstNameTextbox.Text & "',LastName='" & LastNameTextbox.Text & "',Date\_Of\_Birth='" & DOBTextbox.Text & "',County\_ID='" & CountyTextbox.Text & "',Due\_Salary='" & SalaryTextbox.Text & "' where ID\_Number= '" & IDTextbox.Text & "'"

Cmd = New MySqlCommand(Query, mysqlcon)

READER = Cmd.ExecuteReader

MessageBox.Show("Employee Information Updated!", "Finance Management System",

MessageBoxButtons.OK, MessageBoxIcon.Information)

mysqlcon.Close()

Catch ex As MySqlException

MessageBox.Show(ex.Message)

Finally

mysqlcon.Dispose()

End Try

1. **Delete employee module**

It is used to delete employees from the database.

mysqlcon = New MySqlConnection

mysqlcon.ConnectionString =

"server= 'localhost';user id='root';database='Project';password='13115'"

Dim READER As MySqlDataReader

Dim message As String

message = MessageBox.Show("Are you sure you want to delete the Employee?", "Finance Management System", MessageBoxButtons.YesNo, MessageBoxIcon.Warning)

If message = DialogResult.No Then Exit Sub

Try

mysqlcon.Open()

Dim Query As String

Query = "Delete from project.county\_employees where ID\_Number= '" & IDTextbox.Text & "'"

Cmd = New MySqlCommand(Query, mysqlcon)

READER = Cmd.ExecuteReader

MessageBox.Show(" Employee Deleted!", "Finance Management System",

MessageBoxButtons.OK, MessageBoxIcon.Information)

mysqlcon.Close()

Catch ex As MySqlException

MessageBox.Show(ex.Message)

Finally

mysqlcon.Dispose()

End Try

1. **Add user module**

This module is used by the admin to add different county IT managers as users to the database and thus grant them access to the system. Its code is:

mysqlcon = New MySqlConnection

mysqlcon.ConnectionString =

"server= 'localhost';user id='root';database='Project';password='13115'"

Dim READER As MySqlDataReader

Dim message As String

message = MessageBox.Show("Are you sure you want to Add the User?", "Finance Management System", MessageBoxButtons.YesNo, MessageBoxIcon.Question)

If message = DialogResult.No Then Exit Sub

Try

mysqlcon.Open()

Dim Query As String

Query = "insert into project.users (username,password,County\_ID,ID\_Number) values ('" & UsernameTextbox.Text & "','" & PasswordTextbox.Text & "','" & County\_IDTextbox.Text & "','" & IDNumberTextbox.Text & "')"

Cmd = New MySqlCommand(Query, mysqlcon)

READER = Cmd.ExecuteReader

MessageBox.Show("New User Added!", "Finance Management System",

MessageBoxButtons.OK, MessageBoxIcon.Information)

mysqlcon.Close()

Catch ex As MySqlException

MessageBox.Show(ex.Message)

Finally

mysqlcon.Dispose()

End Try

1. **Update user information module**

It used to update the different users’ information such as their passwords in the database. Its code is:

mysqlcon = New MySqlConnection

mysqlcon.ConnectionString =

"server= 'localhost';user id='root';database='Project';password='13115'"

Dim READER As MySqlDataReader

Dim message As String

message = MessageBox.Show("Are you sure you want to Update the User Information?", "Finance Management System", MessageBoxButtons.YesNo, MessageBoxIcon.Question)

If message = DialogResult.No Then Exit Sub

Try

mysqlcon.Open()

Dim Query As String

Query = "update project.users set username='" & UsernameTextbox.Text & "',password='" & PasswordTextbox.Text & "',County\_ID='" & County\_IDTextbox.Text & "',ID\_Number='" & IDNumberTextbox.Text & "' where username='" & UsernameTextbox.Text & "'"

Cmd = New MySqlCommand(Query, mysqlcon)

READER = Cmd.ExecuteReader

MessageBox.Show("User Information Updated!", "Finance Management System",

MessageBoxButtons.OK, MessageBoxIcon.Information)

mysqlcon.Close()

Catch ex As MySqlException

MessageBox.Show(ex.Message)

Finally

mysqlcon.Dispose()

End Try

1. **Delete user**

It is used to delete the users from the database, its code is:

mysqlcon = New MySqlConnection

mysqlcon.ConnectionString =

"server= 'localhost';user id='root';database='Project';password='13115'"

Dim READER As MySqlDataReader

Dim message As String

message = MessageBox.Show("Are you sure you want to delete the User?", "Finance Management System", MessageBoxButtons.YesNo, MessageBoxIcon.Warning)

If message = DialogResult.No Then Exit Sub

Try

mysqlcon.Open()

Dim Query As String

Query = "Delete from project.users where username= '" & UsernameTextbox.Text & "'"

Cmd = New MySqlCommand(Query, mysqlcon)

READER = Cmd.ExecuteReader

MessageBox.Show("User Deleted!", "Finance Management System",

MessageBoxButtons.OK, MessageBoxIcon.Information)

mysqlcon.Close()

Catch ex As MySqlException

MessageBox.Show(ex.Message)

Finally

mysqlcon.Dispose()

End Try

1. **Compare employee module**

This module is used to compare the employees in the counties against those registered in the national registry and to produce a list of those who only appear in the county. This are then considered as ghost employees. Its code is:

mysqlcon = New MySqlConnection

mysqlcon.ConnectionString =

"server= 'localhost';user id='root';database='Project';password='13115'"

Dim SDA As New MySqlDataAdapter

Dim DbDataSet As New DataTable

Dim DSource As New BindingSource

Try

mysqlcon.Open()

Dim Query As String

Query = "select\* from project.county\_employees where county\_id = 101 and ID\_Number NOT IN( select ID\_Number from project.national\_registry)"

Cmd = New MySqlCommand(Query, mysqlcon)

SDA.SelectCommand = Cmd

SDA.Fill(DbDataSet)

DSource.DataSource = DbDataSet

NaiDataGridView2.DataSource = DSource

SDA.Update(DbDataSet)

mysqlcon.Close()

Catch ex As Exception

MessageBox.Show(ex.Message)

Finally

mysqlcon.Dispose()

End Try

1. **Pay verified employees’ module**

It is used to pay the employees who appear both in the national registry and in the county and are thus verified as real employees. Its code is:

Dim salary, x, y As Double

x = Double.Parse(NyeriTextBox.Text, Globalization.NumberStyles.Number)

y = Double.Parse(NyeriTaxTextbox.Text, Globalization.NumberStyles.Number)

salary = x \* ((100 - y) / 100)

mysqlcon = New MySqlConnection

mysqlcon.ConnectionString =

"server= 'localhost';user id='root';database='Project';password='13115'"

Dim READER As MySqlDataReader

Dim message As String

message = MessageBox.Show("Are you sure you want to Pay Nyeri Employees? ", "Finance Management System", MessageBoxButtons.YesNo, MessageBoxIcon.Question)

If message = DialogResult.No Then Exit Sub

Try

mysqlcon.Open()

Dim Query As String

Query = "update project.national\_registry set Paid\_Salary = '" & salary & "' where County\_ID= 102 "

Cmd = New MySqlCommand(Query, mysqlcon)

READER = Cmd.ExecuteReader

MessageBox.Show("Nyeri Employess Paid!", "Finance Management System",

MessageBoxButtons.OK, MessageBoxIcon.Information)

mysqlcon.Close()

Catch ex As MySqlException

MessageBox.Show(ex.Message)

Finally

mysqlcon.Dispose()

End Try

1. **Calculate tax module**

It is used to calculate tax according to a certain input percentage. Its code is:

Dim reader As MySqlDataReader

Dim tax, x, y As Double

x = Double.Parse(NyeriTextBox.Text, Globalization.NumberStyles.Number)

y = Double.Parse(NyeriTaxTextbox.Text, Globalization.NumberStyles.Number)

tax = x \* (y / 100)

Dim Query As String

mysqlcon = New MySqlConnection

mysqlcon.ConnectionString = "server= 'localhost';user id='root';database='Project';password='13115'"

Try

mysqlcon.Open()

Query = "update project.national\_registry set Paid\_Tax = '" & tax & "' where County\_ID= 102 "

Cmd = New MySqlCommand(Query, mysqlcon)

reader = Cmd.ExecuteReader

mysqlcon.Close()

Catch ex As MySqlException

MessageBox.Show(ex.Message)

Finally

mysqlcon.Dispose()

End Try

1. **Exit module**

This module is used to exit the application. Its code is:

Dim iExit As DialogResult

iExit = MessageBox.Show("Confrim if you want to exit", "Finance Management System",

MessageBoxButtons.YesNo, MessageBoxIcon.Question)

If iExit = DialogResult.Yes Then

Application.ExitThread()

End If

### Data Design

The system database as stated earlier is constructed Oracles’ MySQL database management system. It is the one utilized to manipulate the system’s data.

### 4.2.1. Schema Design

The database is organized into multiple tables which are used to store data. The tables are as follows:

1. **Employees national registry table.**

This table has the details of the employees in the nation as maintained by the registrar of persons. It has key information such as the KRA number and ID number. It should be synced regularly with the national registrar’s database to make sure it is up to date.

|  |  |
| --- | --- |
| **Column** | **Data Type** |
| First name | VARCHAR (25) |
| Last name | VARCHAR (40) |
| Date of Birth | VARCHAR (40) |
| ID Number | INT (11) (PRIMARY KEY) |
| KRA ID | VARCHAR (20) |
| County ID | INT (11) |
| Email | VARCHAR (40) |
| Employee ID | INT (32) |
| Paid Salary | INT (11) |
| Paid Tax | INT (11) |

1. **County employees table.**

This table stores information of the employees as recorded by the different IT managers or users.

|  |  |
| --- | --- |
| **Column** | **Data Type** |
| First name | VARCHAR (25) |
| Last name | VARCHAR (25) |
| Date of Birth | VARCHAR (40) |
| County ID | INT (11) |
| Due Salary | INT (11) |
| ID Number | VARCHAR (40) (PRIMARY KEY)  (foreign key references National registry table (ID Number)) |

1. **Users table.**

The table stores information of the different IT managers who are the users of the system as well as the details of the administrator.

|  |  |
| --- | --- |
| **Column** | **Data Type** |
| Username | VARCHAR (40) (PRIMARY KEY) |
| Password | VARCHAR (40) |
| County ID | INT (11) |
| ID Number | INT (11) (foreign key references National registry table (ID Number)) |

### 4.2.2. Data Integrity and Constraints

There are various steps taken ensure data integrity in the database. Data constraints have been implemented to enforce this and they are as follows:

1. **Primary keys.**

These are used to uniquely identify the data in each column of a table. Each table has its own primary key which as a rule must not be null. These are the primary keys in the database:

|  |  |
| --- | --- |
| **Table** | **Primary key** |
| National Registry | ID Number |
| County Employees | ID Number |
| Users | Username |

1. **Foreign keys.**

These are constraints used to link data in one table to its related data in another table. This table shows the foreign keys present in the database.

|  |  |  |  |
| --- | --- | --- | --- |
| **Table** | **Column** | **Referenced table** | **column** |
| County Employees | ID Number | National Registry | ID Number |
| Users | ID Number | National Registry | ID Number |

1. **Domain integrity.**

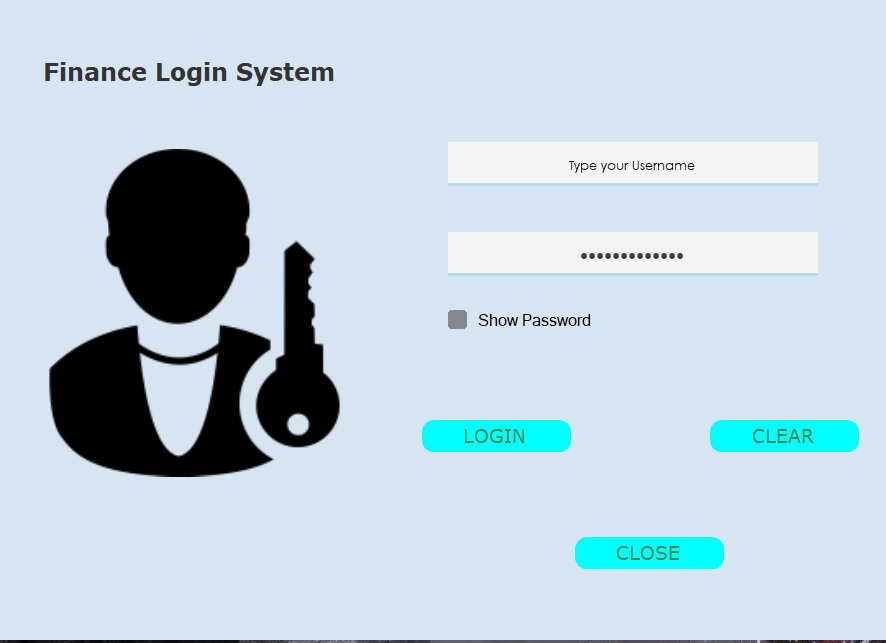
This is the assigning of a specific data type to different columns in the database. It restricts the entry of invalid data types according to the different attributes. Some of the data types employed are: integer and characters.

### User Interface Deisgn

The system is designed to enable the users to carry out their tasks as easily and effortlessly as possible. The interfaces which the users interact with are as follows:

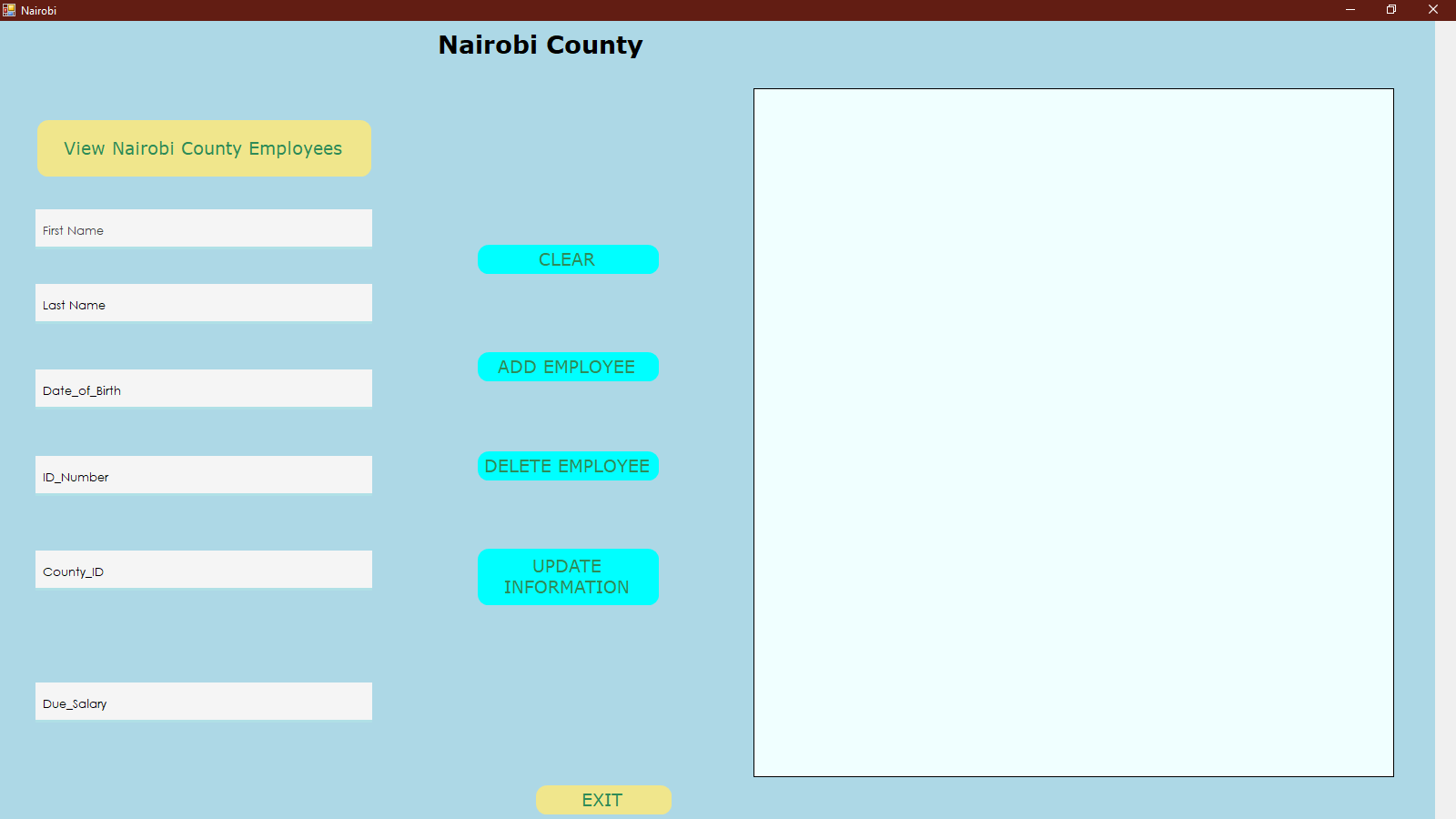
1. The login interface

This is where the users enter their credentials for verification and authentication. It looks like this:



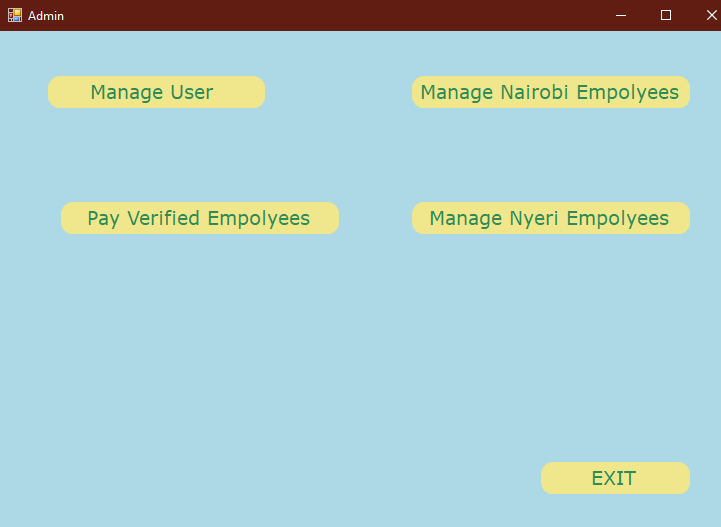
1. The County IT user interface

This interface allows the user to perform various tasks such as adding and deleting employees. It looks like this:



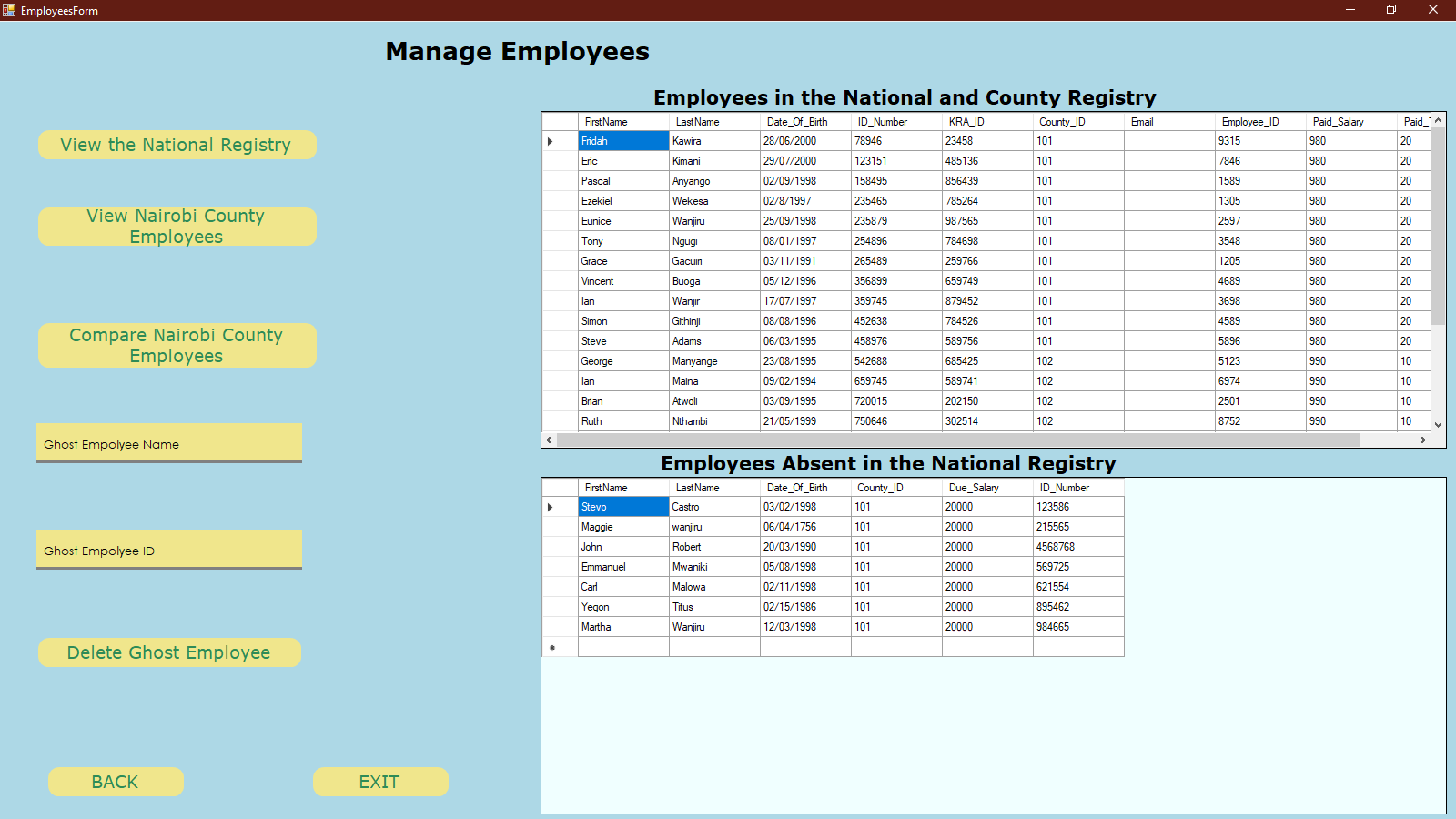
1. The admin interface

It gives the admin the option to choose to which form he wants to navigate to. It looks like this:



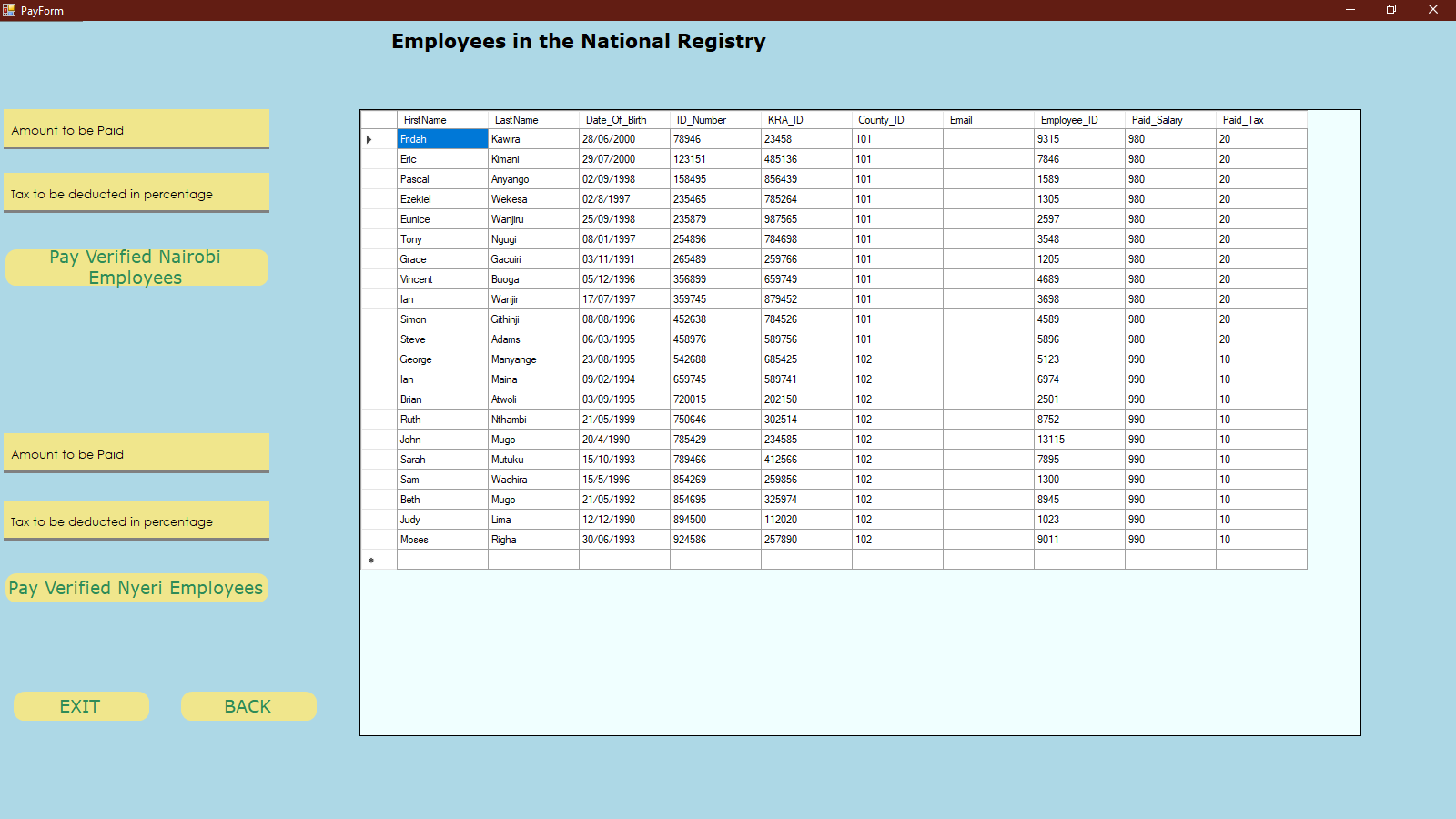
1. The compare employees’ interface

This is used to view the employees in the national registry and county and to compare the two. It looks like this:



1. The pay employees’ interface

It is used by the admin to input the salary to be paid and the tax to be deducted and to view this as it appears in the database. It looks like this:



### Security Issues

Security is a key concern in systems nowadays and especially this being a system handling payment, it is more susceptible to security issues such as hacking. In the development of this system one of the security features implemented is the development of a login form which is the first to run when the system is executed.

The user of the system must be authenticated before being allowed access to the system. The passwords and username are case sensitive such that if they are stored in the database as uppercase and the user types as lower case, they are not granted access to the system.

The database should be backed up to secure site to facilitate its recovery in the event of crashes, hacking or even calamities such as fire and flooding.

### Test Case Design

Several tests were conducted to easily identify errors in the system and thus facilitate their correction and the proper functioning of the system. These test case revolve around the major functioning of the system. They include:

|  |  |  |
| --- | --- | --- |
| **Test ID** | **Test Case** | **Definition** |
| 1. | Entering incorrect username and or password. | This test case was conducted to ensure security of the system by making sure that attempts to log on to the system with incorrect username and or password are rejected. Only with the correct username and password as sored in the database is one authenticated and allowed access to the system. |
| 2. | Trying to add a new employee/ new user | Conducted to ensure that new employee details and users are successfully added to the system database. If so, a message box notifies the user that it was successful. |
| 3. | Trying to update employee/ user information | Carried out to ensure that changes in the employee or user details are successfully added to the database. If so, a message box notifies the user that it was successful. |
| 4. | Trying to delete an employee/ user | Conducted to ensure that deletion of an employee or user is successfully done in the database. If so, a message box notifies the user that it was successful. |
| 5. | Trying to pay employee salary | Carried out to make sure that employees are successfully paid, tax calculated as required and there after remitted to the KRA. If so, a message box notifies the user that it was successful. |

## CHAPTER 5: IMPLEMENTATION AND TESTING

### Implementation Approaches

System implementation is the delivery of the system into actual use in the day to day running of the organization. In this regard to this system, it is meant to be the process of making it live in the operation of the national government in regards to management and payment of employees.

There are a couple of methods that can be utilized in the implementation such as parallel, phased, pilot and direct. However, for this system it is recommended to use parallel changeover where the old system and the new system run concurrently for some time before the new one completely replaces the old. This is because:

1. It enables the staff who will be using it time to learn how to use the system.
2. It gives a chance to compare the output of the old system with the new system. This will enable the government to find out how many ghost employees have been receiving payments.
3. It also allows a chance to fall back on the old system incase the new fails or incase it requires some adjustments before being implemented.

The method on the other hand has some draw backs such as it will cost more to run both systems.

### Coding details and Code Efficiency

### Code Efficiency

Code efficiency is used to describe the reliability, speed and programming methodology used to develop a system. It is key in ensuring high performance and high-speed execution of the system. Some of the methods used in coding to ensure code efficiency include:

1. Making use of reusable components where possible such as modules.
2. Making the use of best keywords, data types and variables.
3. Developing code that is compliant with the design logic and flow of the system.
4. Removing unnecessary code.

### Testing Approach

This defines how system testing is to be carried out. There are two main approaches to this, that is, proactive and reactive approach. In the testing of this system, proactive approach was employed in which the testing begun as early as possible in order to locate and fix bugs and defects.

### Unit Testing

This testing approach where individual components of a system are tested. It is meant to make sure that each unit of the system performs as designed and required.

In this system, each module ranging from the login module, add employees’ module and even the exit module was first tested as an individual component. These tests were conducted each time any code is changed or modified in a certain module. Some advantages accrued from unit testing of the system include:

1. Debugging is made easy. If attest fails, only the latest changes require to be debugged.
2. Codes are made reusable because to carry out unit testing, the codes must be modular.
3. It reduces the cost of fixing defects in times of time and effort.

### Integrated Testing

It is the testing of all the integrated components of a system to expose faults in the interaction of these units.

In this system, after all units were integrated, they were tested to make sure they work as required without defects. One of the approaches used here is the big bang. All the units were combined and tested at one go.

### Modifications and Improvements

After thorough testing, bugs and errors identified in the code were corrected. This modification of the code is done through corrective maintenance. In the life cycle of the system, other improvements may need to be implemented. They include:

1. Adaptive maintenance – to keep the system up date in the ever-changing world of technology.
2. Perfective maintenance – to keep the system in use for long by adding new features, improving its reliability and performance.
3. Preventive maintenance - modifications to prevent future problems.

## CHAPTER 6: CONCLUSIONS

### 6.1. Conclusions

The system would be of much help to the national government in curbing corruption and wastage of resources if adopted. It however does require more development and more input from stake holders to better and fine tune it.

### 6.2. Limitations of the Sytem

There were some limitations encountered development of the system. The main limitation is that this system needs to be integrated with other systems such as the banking system, the KRA system and it also requires to sync with the registrar of persons database. However, it was impossible to get access to these facilities and thus its capabilities could not be fully tested. For example, since it can not connect to the bank and the KRA, the system only inputs the calculated amounts into the Salary and KRA columns in the database.

### References

Kofler, M. (2006). *The definitive guide to MySQL 5*. Apress.<https://www.techopedia.com/definition/27151/code-efficiency>

Stucky, M. (2001). *MySQL: building user interfaces*. Sams Publishing.

<https://www.tutorialspoint.com/software_testing_dictionary/test_approach.htm>

<https://www.youtube.com/watch?v=XMLAXQR_YFI&list=PLS1QulWo1RIZxHuINmI3EIAjCq3zzsiFy>