

THE UNIVERSITY OF ZAMBIA
SCHOOL OF NATURAL SCIENCES
DEPARTMENT OF COMPUTER
STUDIES

EMPLOYEE
MANAGEMENT SYSTEM

BY

MWEEMBA SIMAANYA

**A thesis submitted in partial fulfillment of the requirements for the award of Bachelor's
Degree of Computer Science.**

Copyright 2014

DECLARATION

I, the undersigned hereby declare that the Employee Management System is my own work, that it has not been submitted for any degree or examination in any other university to my knowledge, and that all sources I have used or quoted have been indicated and acknowledged by complete references.

Name: Mweemba Simaanya

Signature:.....

Date:.....

Supervisor

Name: Mr. Martin Phiri

Signature:.....

Date:.....

Course Coordinator

Signature:.....

Date:.....

ACKNOWLEDGEMENTS

I am greatly indebted to the following for their contribution to my work:

The Almighty God, who gave me the insight and perseverance to accomplish this work. My parents for all the support. My siblings, Nanchengwa, Sylvia, Chimwemwe, Mwenzi, Karen and Boniface who I never had time for due to my busy schedule.

My supervisor, Mr. Martin Phiri who's sounding advice helped me steer this project in the right direction, a work I will forever cherish.

Members of staff at the Computer Studies department for the solid knowledge-base that enabled me carryout the research.

Musafili Chilembo, who's in-depth knowledge in advanced Web Technologies such as AJAX, JSON and JQUERY was a great asset to my project.

Finally, my classmates who's technical and moral support throughout my stay at the Computer Studies department was of great help.

DEDICATION

I dedicate this work to my parents who have always supported my dreams and aspirations, and given me all the financial support a child can ever need.

LIST OF TERMINOLOGIES

EMS – Employee Management System

MSS – Management Self-Service

HRMS – Human Resource Management System

HRIS – Human Resource Information System

HR – Human Resource

HOD – Head of Department

ESS – Employee Self-Service

WBS – Work Breakdown Structure

ERP – Enterprise Resource Planning

ABSTRACT

Employees are the backbone of any company therefore their management plays a major role in deciding the success of an organization [1]. Employees Management Software makes it easy for the employer to keep track of all records. This software allows the administrator to edit employees, add new employees, transfer/promote/terminate employees. Each employee in the database is associated with a position can be added and edited when need arises. Employees can be transferred between positions easily without having to retype back their information in the database. You can check to see if there are duplicate positions/employees in the database. Most of all, the employer can assign tasks to employees and assess their progress in order to keep track of employee performance.

A flexible and easy to use Employee Management software solution for small and medium sized companies provides modules for personnel information management thereby organization and companies are able to manage the crucial organization asset – people [2]. The combination of these modules into one application assures the perfect platform for re-engineering and aligning Human Resource processes along with the organizational goals. This system brings about an easy way of maintaining the details of employees working in any organization.

It is simple to understand and can be used by anyone who is not even familiar with simple employees system. It is user friendly and just asks the user to follow step by step operations by giving easy to follow options. It is fast and can perform many operations for a company.

The goal of this project is to design and develop an employee management system to fill existing gaps in the electronic management of employees.

Table of Contents

1	CHAPTER ONE: INTRODUCTION TO THE RESEARCH	11
1.1	PROBLEM STATEMENT	11
1.2	PROJECT BACKGROUND	11
1.3	OBJECTIVES	12
1.4	SCOPE	12
1.5	EXPECTED BENEFITS	13
1.6	REQUIREMENTS AND CONSTRAINTS	14
1.6.1	Functional Requirements:	14
1.6.2	Non-Functional Requirements:	16
	Since EMS application is a web-based application, internet connection must be established.....	16
	The EMS software personal database model will support MySQL environment as DBMS.	16
1.7	SUMMARY	16
2	CHAPTER TWO –RESEARCH.....	17
2.1	INTRODUCTION	17
2.2	LITERATURE REVIEWS ON TOPICS RELATED TO THE PROJECT	17
2.3	REVIEWS ON SEVERAL SOFTWARE METHODOLOGIES.....	18
2.4	REVIEW OF POSSIBLE DEVELOPMENT TOOLS AND SOFTWARE TO BE USED	20
2.4.1	Back-end Technology	20
2.4.2	Database Management System	22
2.5	REVIEW OF SIMILAR SYSTEMS	24
2.6	SUMMARY	28
3	CHAPTER THREE: SYSTEM ANALYSIS.....	29
3.1	INTRODUCTION	29
3.2	SOFTWARE DEVELOPMENT METHODOLOGY OF CHOICE	29
3.3	SYSTEM DESIGN	29
3.3.1	Use case analysis.....	29
3.3.2	Class diagram	36
3.4	DEVELOPMENT TOOLS	37
3.4.1	Front End Technologies.....	37
3.4.2	Back End Technologies.....	38
3.5	SUMMARY	39

4	CHAPTER FOUR: DESIGN	40
4.1	INTRODUCTION	40
4.2	EXPLANATION OF THE PROPOSED SYSTEM	40
4.3	SYSTEM AND ALGORITHM FLOWCHARTS.....	41
4.4	SYSTEM STRUCTURE CHART DFD AND ERD	43
4.4.1	Data Flow Diagram (DFD).....	43
4.4.2	ENTITY RELATIONSHIP DIAGRAM (ERD).....	44
4.5	STORYBOARD	45
4.5.1	Sequence Diagrams.....	45
4.6	INTERFACE DESIGN	47
4.7	Sketches of graphics.....	48
4.8	SUMMARY	50
5	CHAPTER 5: IMPLEMENTATION	51
5.1	INTRODUCTION	51
5.2	DESCRIPTION OF DEVELOPED SYSTEM	51
5.2.1	Accessing the system	51
5.3	TECHNICAL DETAILS OF IMPLEMENTED SYSTEM.....	52
5.3.1	Model View Controller architecture (MVC)	52
5.3.2	MySQL Triggers	53
5.3.3	Algorithms.....	54
5.3.4	Pseudocode.....	55
5.4	SCREENSHOTS OF DEVELOPED SYSTEM.....	57
5.5	SUMMARY	57
6	CHAPTER SIX: TESTING AND VERIFICATION	58
6.1	INTRODUCTION	58
6.2	SCOPE	58
6.3	TESTING GOALS.....	58
6.4	CONFIRMATION TESTING.....	58
6.5	REGRESSION TESTING	58
6.6	TEST PLANS AND RESULTS	59
6.7	SUMMARY	62
7	CHAPTER SEVEN: CONCLUSION	63

7.1	INTRODUCTION	63
7.2	RESULTS.....	63
7.3	PROBLEMS FACED	63
7.4	LIMITATIONS	63
7.4.1	Custom report generation	63
7.4.2	Head of Department heads all Projects.....	63
7.4.3	Leave Management	64
7.4.4	Browser support.....	64
7.5	FUTURE WORK	64
7.5.1	Leave Management	64
7.5.2	Integration with payroll system	64
7.5.3	Employee Performance.....	64
7.5.4	Information archiving.....	64
7.6	LEARNING EXPERIENCE	64
7.7	CONCLUSION.....	65
8	REFERENCES	66
9	APPENDIX.....	68
9.1	Appendix A – Sequence Diagrams	68
9.1.1	Report generation.....	68
9.1.2	Edit account details.....	68
9.2	Appendix B – Sample Code	69
9.2.1	Database Connectivity	69
9.2.2	User Authentication.....	69
9.2.3	PDF Library	71
9.2.4	Leave application	71
9.2.5	Create new project and add member.....	73
9.3	Appendix C – Screen shots of developed system	74
9.3.1	Employee interface	74
9.3.2	Admin interface.....	74
9.3.3	Leave management.....	75
9.3.4	Edit profile.....	75
9.4	Appendix I – Testing.....	76

9.4.1	Login.....	76
9.4.2	Add new user	76
9.4.3	Edit Account	77
9.4.4	Leave request denied due to pending leave	77
9.4.5	Leave request successfully submitted	78
9.4.6	Create Project WBS.....	78
9.4.7	Generate reports.....	79

1 CHAPTER ONE: INTRODUCTION TO THE RESEARCH

1.1 PROBLEM STATEMENT

Manual handling of employee information poses a number of challenges. This is evident in procedures such as leave management where an employee is required to fill in a form which may take several weeks or months to be approved. The use of paper work in handling some of these processes could lead to human error, papers may end up in the wrong hands and not forgetting the fact that this is time consuming. A number of current systems lack employee self-service meaning employees are not able to access and manage their personal information directly without having to go through their HR departments or their managers. Another challenge is that multi-national companies will have all the employee information stored at the headquarters of the company making it difficult to access the employee information from remote places when needed at short notice.

The aforementioned problems can be tackled by designing and implementing a web based HR management system. This system will maintain employee information in a database by fully privacy and authority access. The project is aimed at setting up employee information system about the status of the employee, the educational background and the work experience in order to help monitor the performance and achievements of the employee through a password protected system.

1.2 PROJECT BACKGROUND

Employees are the backbone of any company therefore their management plays a major role in deciding the success of an organization [1]. Human Resource Management Software makes it easy for the employer to keep track of all records. This software allows the administrator to edit employees, add new employees as well as evaluate an employee's performance. Employees can be managed efficiently without having to retype back their information in the database. You can check to see if there are duplicate positions/employees in the database.

A flexible and easy to use Employee Management software solution for small and medium sized companies provides modules for personnel information management thereby organization and companies are able to manage the crucial organization asset – people [2]. The combination of these modules into one application assures the perfect platform for re-engineering and aligning

Human Resource processes along with the organizational goals. This system brings about an easy way of maintaining the details of employees working in any organization.

It is simple to understand and can be used by anyone who is not even familiar with simple employees system. It is user friendly and just asks the user to follow step by step operations by giving easy to follow options. It is fast and can perform many operations for a company

1.3 OBJECTIVES

In this world of growing technologies everything has been computerized. With large number of work opportunities the Human workforce has increased. Thus there is a need of a system which can handle the data of such a large number of Employees. This project simplifies the task of maintaining records because of its user friendly nature.

The objective of this project is to provide a comprehensive approach towards the management of employee information. This will be done by designing and implementing an HR management system that will bring up a major paradigm shift in the way that employee information is handled.

The objectives of this system include:

- Design of a web based HR management system to fulfill requirements such as project management, leave management, report generation to assist in performance appraisal, ESS and employee trainings.
- Well-designed database to store employee information.
- A user friendly front-end for the user to interact with the system.

1.4 SCOPE

The scope of this project will be limited to the following:

- Employee profiles:
Employees will have access to their personal profiles and will be able to edit their details.
- Electronic leave application:

Complete elimination of paperwork in leave management by enabling an employee apply for leave as well as check their leave status through the system. This will also enable the HR manager to accept/reject leave application through the system

- **Project Management:**

Assign tasks and projects to employees, assign a project team and keep track of the progress.

- **Report generation:**

The HR manager will be able to generate timely reports in order to monitor employees and this can be used for performance appraisals. The reports will be have all the information of an employee from educational background, trainings attended, projects done as well as technical skills.

- **Recruitment Process:**

The admin will add an employee and a default password and employee id will be generated and sent to the new employees email. The HR manager will then have the ability to add an employee's information to the database.

1.5 EXPECTED BENEFITS

This system is expected to be user friendly and will offer easy access to data as well as services such as online leave management, e-recruitment, and timely report generation, monitoring employee trainings, task management, project management and employee tracking.

The employee is expected to have direct interaction with this system through a password protected user account therefore proposed system is web based to enable accessibility from any location as long as internet connectivity is available. This direct interaction with the system will enable employee self-service.

Without an employee management system, it's a tedious job for the human resource department to keep track of each and every employee and even harder for a project manager to assign tasks to the project team. The HR management system will be developed to provide information of employees and many other facilities at the click of a button.

1.6 REQUIREMENTS AND CONSTRAINTS

1.6.1 Functional Requirements:

Authentication

- Login- The user can login to the HRMS system with his/her username and password.
 - Logout- The user can log out from the HRMS system.
- Login failure- If the user does not exist in the database or the user has not yet being authorized by the HRMS admin.

Authorization

- User role check- After logging in, the user role will be checked from the database and the user interface will be displayed according to their role.

Process Data

- Display- User with defined roles can display the content of the database. Being more specific, employee can only view his/her personal information. HOD can not only see his/her personal information but also employee's information who are under his/her department or school. Admin and HR can display their personal information and all employees' information.
 - Edit- A user with employee role can edit his/her specific personal information. Dean or HOD can only edit employees' personal information that is under his/her coverage except user role type. Admin can edit all information related to all employees' including their user role type.
 - Search- User with Dean/HOD role can search the content of database for the employees' who are under his/her coverage. HR and admin roles can search all the employees' information in the database. Search feature works on specific keywords showing employee's characteristics, peculiarities, skills, features, and etc. For example, HR wants to find employees' who are well trained in "Java Programming Language". He/she will write the specific keyword in the search bar and press the available search button. Afterwards, he/she will find a list of all the employees' who know "Java Programming".
- Update authentication- This feature can be used only by admin role type. Admin can update the role type of a specific user. For example, an employee got promotion and his role type will be changed from employee role id to HOD or Dean role. Admin will be able to update this authentication mechanism.

Leave Application/Approval

- Leave application- The user can be able to fill in leave application form in the appropriate fields.
- Leave approval- The admin can be able to approve leave applications based on the reasons stated, length of leave as well as available HR on a department.
- Leave days accrued- The user shall be able to check the number of leave days accrued.

Recruitment

- Add new employee- HR role type is able to add a new employee to the database. The new employee will have all the required personal information related to him/her. The new created employee will have an id.
- Add a new user- After a new employee has being created by HR role, admin role is responsible for creating a new user by the specified id assigned in the “Add a new employee” feature. The unique id will be given by the system. Admin will assign a new role such as employee, Dean, HOD, HR, and admin to the new created user.

Report generation

- Report generation- HR shall be able to generate a report in pdf format for each employee based on the information in the database.

Project Management

- Create project team: The HOD of department or project manager shall be able to create a project and come up with a project team.
- Work Breakdown Structure (WBS): The HOD or project manager shall be able to assign tasks to the project team as well as monitor their progress.

Trainings and Task Management

- Trainings: The HOD shall create trainings and assign employees that are required to attend the trainings as well.
- Tasks: HOD shall assign tasks to employees in his/her department.

1.6.2 Non-Functional Requirements:

Performance requirements

There is no restriction on the number of the users to be added to the database.

Hardware requirements

EMS should be able to work on a computer with the following minimum hardware specifications:

OS: Windows XP/Vista/7/8 and Linux

CPU: Pentium III (700MHz) and above

Memory: 128 MB and above

Capacity: 4GB of hard drive

Others: Network interface card, mouse, keyboard, and monitor.

Software requirements

Since EMS application is a web-based application, internet connection must be established.

The EMS software personal database model will support MySQL environment as DBMS.

1.7 SUMMARY

This chapter began by giving a brief overview of the entire project, the background of the project and the scope. The problem definition and solution highlights the current problems faced with the use of the systems that are in place and outlines briefly the solution system to be developed. The next chapter will focus on the literature review. This is literature that relates to the project and similar systems.

2 CHAPTER TWO –RESEARCH

2.1 INTRODUCTION

This chapter summarizes the evaluation of the literature relevant to the Employee Management System. It examines theories, concepts, approaches, methods and techniques relevant to the project. Similar existing technologies relating to the development the EMS are discussed.

2.2 LITERATURE REVIEWS ON TOPICS RELATED TO THE PROJECT

A HRMS refers to the systems and processes at the intersection between human resource management (HRM) and information technology. It merges HRM as a discipline and in particular it's basic HR activities and processes with the information technology field whereas the programming of data processing systems evolved into standardized routines and packages of enterprise resource planning (ERP) software. [3]

An organization or company with a very large number of employees manages a greater volume of data. This activity can be daunting without a more sophisticated tool to store and retrieve data. The various levels of sophistication can be examined by looking at the evolutionary aspects of HR technology. These aspects can be characterized into four stages of development: Paper-based systems, early personal computer (PC) technology, electronic databases, and Web-based technology. [4]

The benefits of automation are becoming widely known to HR and other areas of the business. The focus has shifted to automating as many transactions as possible to achieve effectiveness and efficiencies.

The technology of the future will be about speedy access to accurate current information, and reliability to access this information via multiple systems will give organizations a strategic edge. HR is expected to relinquish its role as sole owner of HR information, so that managers and employees can use this information to solve their own problems using Web-based systems. This new system will not necessarily mean reduction in HR staff. The new system will enable HR professionals to focus on transforming information into knowledge that can be used by the organization for decision making; it will be about HR and IT working together to leverage this technology. A recent study by the Hackett Group, a business process advisory firm found that

high-performing organizations spend 25 percent less than their peers on HR because they use technology effectively. [5]

The two most popular Web-based HR applications used today are self-service for employees and self-service for managers. These applications have enabled companies to shift responsibility for viewing and updating records onto individual employees and have fundamentally changed the manner in which employees acquire information and relate to their HR departments. [6]

2.3 REVIEWS ON SEVERAL SOFTWARE METHODOLOGIES

A software development methodology is a collection of procedures, techniques, tools, and documentation aids which will help the systems developers in their efforts to implement a new information system. [7]

There are a number of software development methodology each of which are adopted based on a number of factors relating to the project e.g. Time, cost, incorporation of requirement changes during the development process, system complexity, communication between customers and developers, software criticality, size of the development team. These generic models are not definitive descriptions of software processes. Rather, they are abstractions of the process that can be used to explain different approaches to software development. You can think of them as process frameworks that may be extended and adapted to create more specific software engineering processes. Below are a selected number of models:

The Waterfall Model

The waterfall model is a sequential design process, often used in software development processes. It takes the fundamental process activities of specification, development, validation, and evolution and represents them as separate process phases such as requirements specification, software design, implementation, testing, and so on. [8]

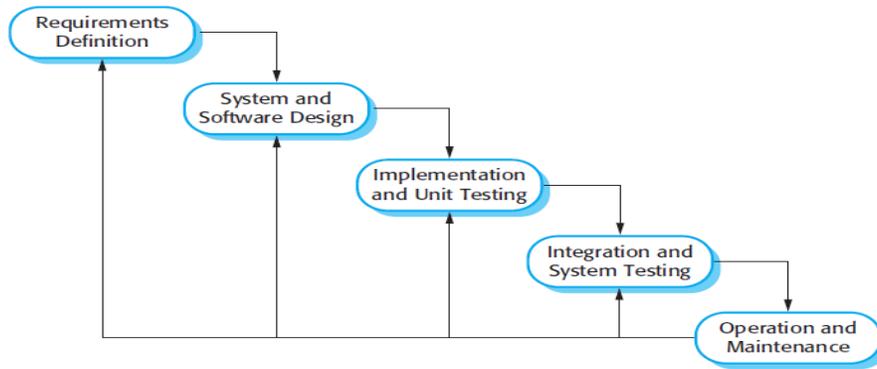


Figure 2.1 the Waterfall Model

Incremental Model:

This approach interleaves the activities of specification, development, and validation. The system is developed as a series of versions (increments), with each version adding functionality to the previous version.

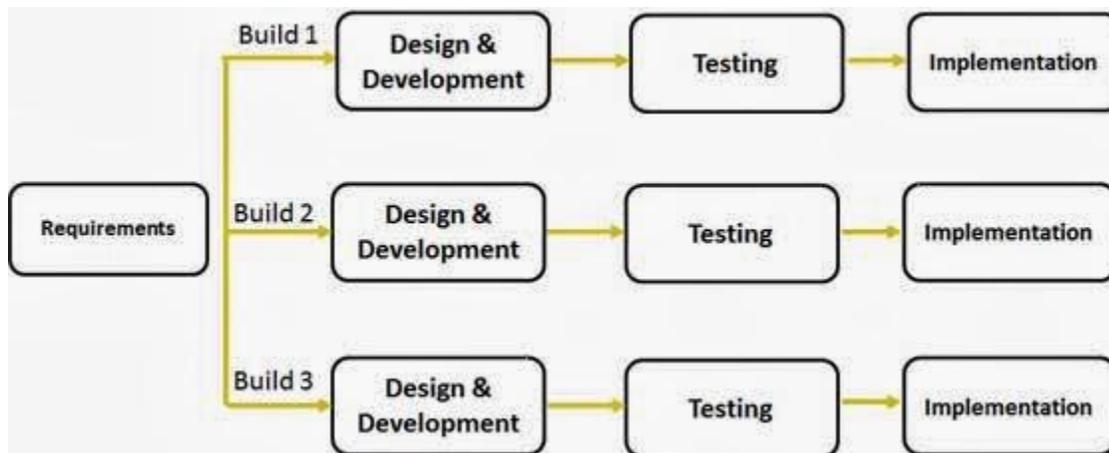


Figure 2.2 Incremental Model

Reuse-oriented methodology:

This approach is based on the existence of a significant number of reusable components. The system development process focuses on integrating these components into a system rather than developing them from scratch.

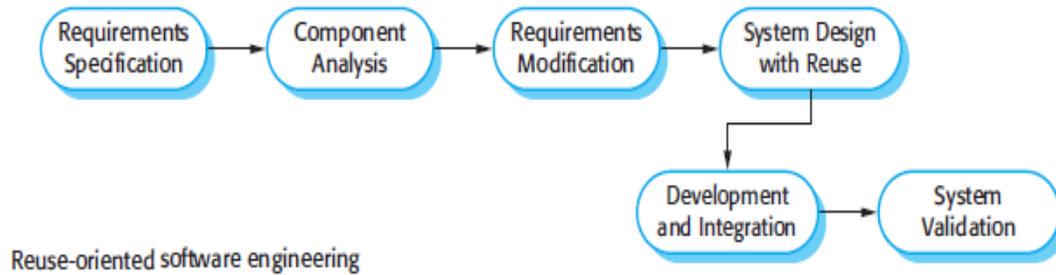


Figure 2.3 Reuse-oriented software engineering

2.4 REVIEW OF POSSIBLE DEVELOPMENT TOOLS AND SOFTWARE TO BE USED

The following are various development tools and software that could be used for the system.

2.4.1 Back-end Technology

JavaServer Pages

JavaServer Pages (JSP) is a technology that helps software developers create dynamically generated web pages based on HTML, XML, or other document types. Released in 1999 by Sun Microsystems, JSP is similar to PHP, but it uses the Java programming language.

To deploy and run JavaServer Pages, a compatible web server with a servlet container, such as Apache Tomcat or Jetty, is required. [9]

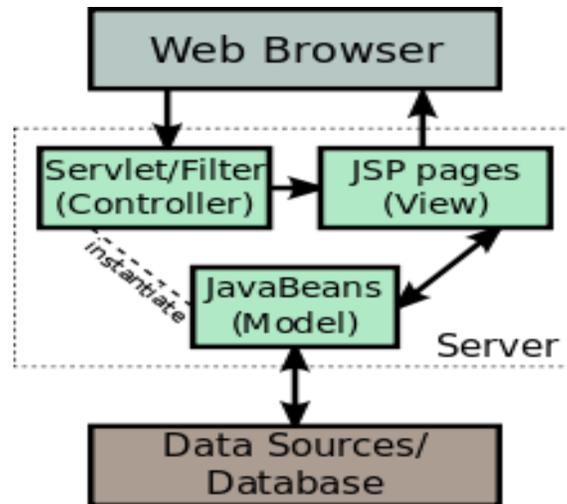


Figure 2.4 The JSP Model 2 architecture

ASP.NET

ASP.NET (Active Server Pages.NET) is a proprietary scripting language or application framework developed by Microsoft used to create enterprise wide web applications which can be accessible globally. ASP.NET:

- Drastically reduces the amount of code required to build large applications.
- The HTML produced by the ASP.NET page is sent back to the browser. The application source code you write is not sent and is not easily stolen
- ASP.NET makes for easy deployment. There is no need to register components because the configuration information is built-in
- ASP.NET validates information (validation controls) entered by the user without writing a single line of code.

ASP.NET makes development of any web based application or system easier and faster as it allows developers to drag and drop components and specify their functions while code is automatically generated, as a result of this ease complex applications can be developed in a short period of time even by programmers who are not too familiar with the language. Despite ASP.NET being robust it has also proven to be more expensive to implement

and the fact that it's not platform independent limits the number of places in which it can be used. [10]

PYTHON

Python is a widely used general-purpose, high-level programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C. The language provides constructs intended to enable clear programs on both a small and large scale.

Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library.

Like other dynamic languages, Python is often used as a scripting language, but is also used in a wide range of non-scripting contexts. Using third-party tools, such as Py2exe or Pyinstaller, Python code can be packaged into standalone executable programs. Python interpreters are available for many operating systems. [11]

PHP

PHP (Hypertext Preprocessor) is an open source server side scripting language, it is platform independent, meaning it can work on all major operating systems. PHP supports many types of databases including MySQL and is supported by a large community of users and developers. PHP is an excellent choice for developing web based systems because it's an open source technology and has a large community of users and developers, this makes PHP a language that is easy to learn and understand, furthermore coding solutions and bugs are resolved quickly. The fact that PHP is platform independent gives the developer the freedom to develop an application without worrying about the operating system on a user's machine. PHP has the ability to integrate with most web technologies thus it can be used as middleware. [12]

2.4.2 Database Management System

MySQL

MySQL is an open source database that is platform independent and can easily interface with a number of scripting languages, it works best with PHP though. The number of advantages of using MySQL which include, the ability to handle stored procedures, triggers, SQL and User-Defined functions. It also offers a high-speed data load utility and support for various drivers (ODBC, JDBC, .NET, PHP).

Deploying a MySQL database has proved to be cheap and easy as it doesn't require special hardware or software requirements, it can work well on any web server but most professionals recommend the apache web server. MySQL is an excellent database to use when developing web based applications because its platform independent and can easily interface with a number of scripting languages.

MS SQL (Microsoft SQL Server)

Microsoft SQL Server is Microsoft's relational web hosting database used to store website information like user information, it's mostly used on windows servers and it is not free. It has advanced features such as buffer management, logging and transaction, concurrency and locking, replication services, integration services, stored procedures and triggers. MS SQL databases work well with ASP.NET and also integrate well with other Microsoft products. MS SQL has been used to support large enterprise applications worldwide, its most common use is to store data for Customer Relationship Management(CRM) systems in large organization that need to keep track of their customers data for example mobile phone service providers, this database though is not platform independent and is also expensive to implement. A lot of web based help desk systems around the world created using ASP.NET or C# are all supported by MS SQL database.

Oracle Database

Oracle database is a powerful relational database management system that has a number of features. In today's market, oracle database management systems are one of the most popular and full featured databases. Oracle databases are widely used as backend database systems for most enterprise applications because they are robust and secure. Oracle is a power hungry database that requires a lot of system resources to function properly. One of its major

advantages is that it is platform independent. An Oracle database will work well with any web based system as long as there are enough resources required for it to run on.

2.5 REVIEW OF SIMILAR SYSTEMS

OrangeHRM

OrangeHRM is a powerhouse human resources tool that any small or midsize business can benefit from using. With OrangeHRM, you have options: You can download and install the system on your own hardware, or you can purchase a hosted solution. To get prices for the hosted solution, you have to contact them from their Request a Quote page.

OrangeHRM's features include: fully modular, addons (e.g., benefits, employee self-service, training, budget, job and salary history, etc.) for purchase, all standard HR functions (employees, leave, benefits, performance, etc.), and more.

The installation is fairly straight-forward. With a self-extracting Windows installer or full-source installations for Windows, Mac, and Linux, you can get OrangeHRM up and running on nearly every platform. If you don't have the hardware or the skills to set up Orange onsite, you can request a quote for a hosted instance of OrangeHRM. You can also purchase support plans and customizations.

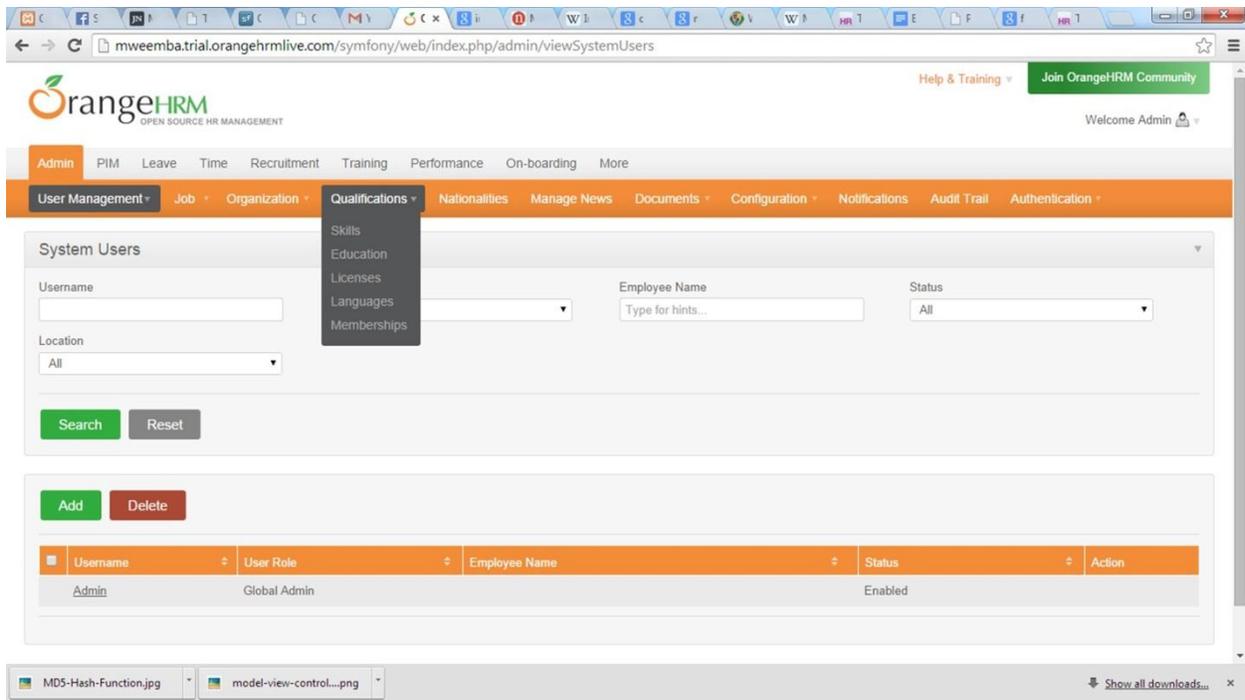


Figure 2.5 Snapshot of OrangeHRM interface

SimpleHRM

SimpleHRM offers an open source version of its professional platform. This version offers time management, and it can be installed on either a WAMP (Windows Apache MySQL PHP) or LAMP (Linux Apache MySQL PHP) server.

Once installed, SimpleHRM offers every feature you need to solidify your HRM department: employee information, leave management, travel management, expense management, benefit management, and task reporting. SimpleHRM allows you to assign a CV to an employee and define eligibility for rehire. Each major module offers plenty of granular control, and the user interface is well laid out.

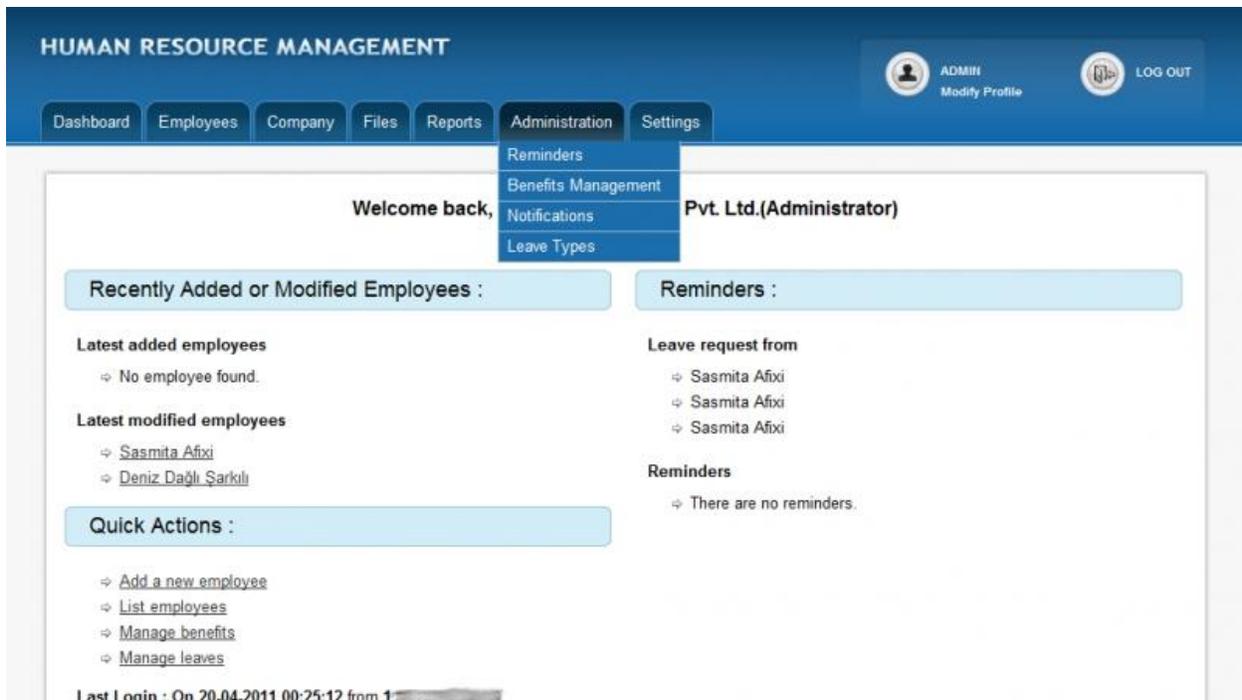


Figure 2.6 Snapshot of SimpleHRM interface

Waypoint HR

WaypointHR is the HR software for any small or midsize company looking for a platform that nearly any user, of any experience level, can use. WaypointHR can manage employee data, which include:

- Personal details
- Holiday/sickness/absence history
- Employment/contract/job/salary details
- Discipline and grievance records
- Performance appraisals
- Exit interviews and termination
- A five-step add employee wizard
- Export reports to PDF
- Multi-site facility layering

WaypointHR also offers an active online support forum, a dedicated support website (which includes developer support), as well as an on-demand solution (for those that do not want to bother with the installation of WaypointHR on a local machine). To get a quote for the on-demand solution, visit this page, fill out the questionnaire, and wait to hear from WaypointHR.

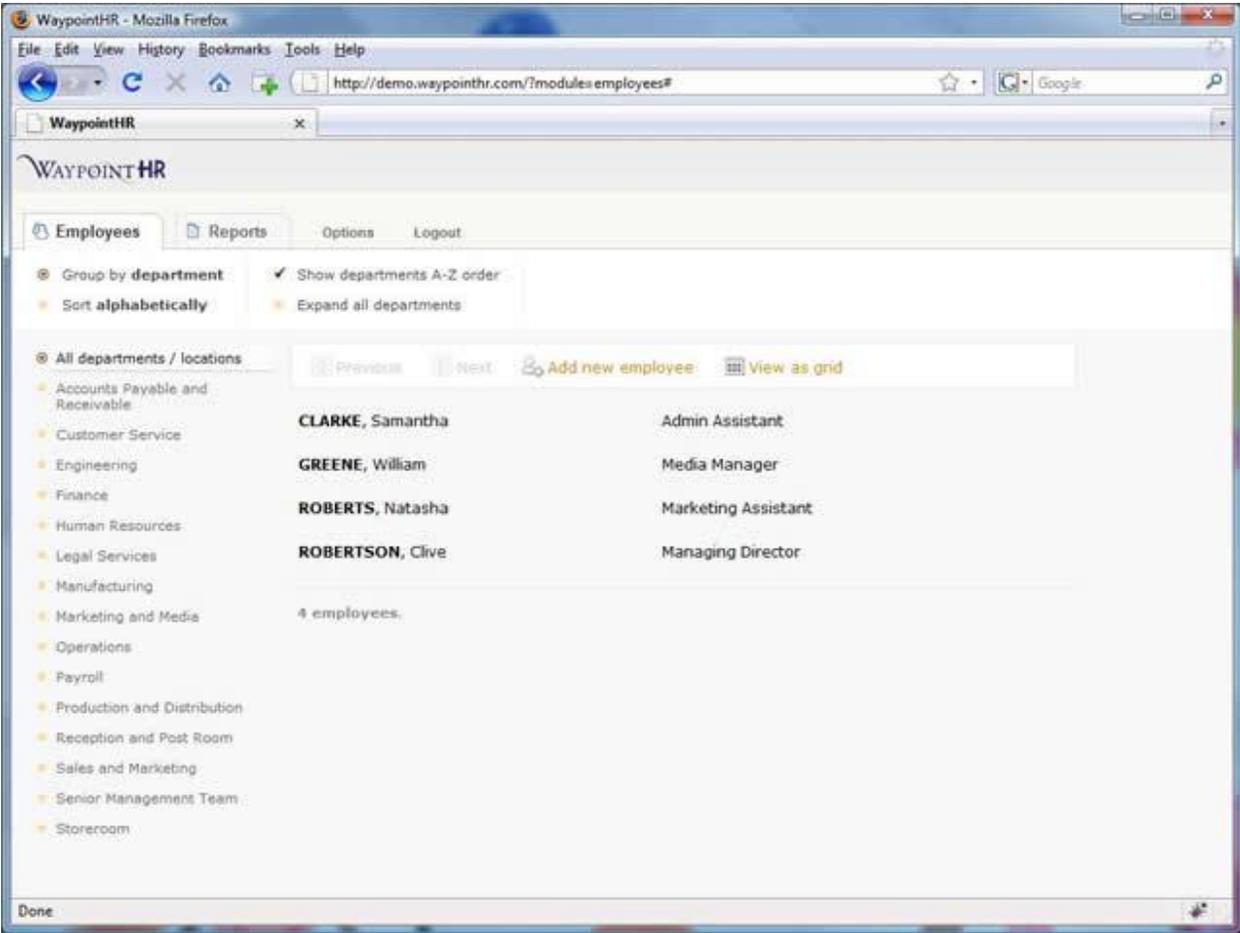


Figure 2.7 Snapshot of WaypointHR interface

Sage HR Africa

Sage HR Africa offers world-class HR and payroll software and services to the African continent. We have an African footprint of 35 countries that includes an extensive network of Strategic and Business Partners that can assist you with installation, training and any on-site

support that your business requires. We are committed to providing robust, innovative and easy-to-use human resource and software applications that will make your business life so much easier. We ensure statutory compliance with local authorities and with our software your business is always in line with country-specific payroll and HR rules and regulations. Our HR and Payroll software is ideal for any size and type of business. Whether you are just starting out, or if you have an existing business that is growing, Sage HR Africa's software solutions aim to support the growth of your business and to develop an ongoing partnership with you, our potential customer, for the long-term.

Sage HR Africa Leave application Workflow



Figure 1 Leave application Workflow

2.6 SUMMARY

The literature review in this chapter has looked at a brief overview of existing HR and employee management systems and what procedures have to be followed when executing these HR tasks. Various front and back end technologies were also reviewed highlighting the advantages and disadvantages of their use, lastly the chapter looked at the importance of security over the Internet and suggested ways in which a web application can be made secure. The next chapter will take a look at the system analysis of the developed system.

3 CHAPTER THREE: SYSTEM ANALYSIS

3.1 INTRODUCTION

This chapter gives a detailed outline of the software development methodology used in this project following up the various existing software development methodology discussed in chapter two. The strength and weaknesses of the chosen methodology have been outlined. Further, the functional and non-functional requirements of the system are explained in detail and the use cases which are a list of steps, typically defining interactions between a role and a system, to achieve a goal. Class diagrams have been given to show detailed data modeling of the system which will be translated into code.

3.2 SOFTWARE DEVELOPMENT METHODOLOGY OF CHOICE

Having briefly discussed a few software development methodologies in chapter two, the incremental method was favored for the following reasons:

- It allows for development of high-risk or major functions first
- Each release delivers an operational product
- Customer can respond to each build
- Uses “divide and conquer” breakdown of tasks
- Lowers initial delivery cost
- Initial product delivery is faster
- Customers get important functionality early
- Risk of changing requirements is reduced

3.3 SYSTEM DESIGN

3.3.1 Use case analysis

A use case defines a goal-oriented set of interactions between external users and the system under consideration or development. Thus a Use Case Scenario is a description that illustrates, step by step, how a user is intending to use a system, essentially capturing the system behavior from the user's point of view.

In order to create relevant use cases for the system, the following actors for the system have been identified:

- Employee (could be lecturers, accountants, technicians)
- Head Of Department (HOD)
- Human Resource (HR)
- Admin

Actors, Use Cases and their Description

Actor	Use case	Description
Employee	Edit Profile	Employee will be able to edit personal details such as emergency contacts as well as technical skills acquired.
Employee	Apply Leave	Employee will be able to submit leave request along with supporting documents.
Employee	View Tasks	The employee will be able to view tasks assigned by the HOD.
Employee	Check Leave days	Employee will be able to check leave days.
HOD	Assign tasks	HOD will assign tasks to employees in his department.
Admin	Add new employee	Add mean will be able to create new employees.
Admin	Edit user role	Admin will be able to edit user roles.
HR	Accept leave application	HR will accept leave

		applications from employees.
HR	Reject leave application	HR will reject leave applications from employees.
Admin	View user activity log	Admin will be able to view activity log of all users in the system
HOD	Create projects	The HOD is able to create a project, come up with a project teams as well as assign tasks to the project members breaking it down into a WBS.
HOD	Create trainings	HOD will create trainings and delegate employees that will attend the trainings.
HR	Generate reports	HR will be able to generate reports containing employee information.

Table 3.1 Actors, Use Cases and their Description

Use case diagrams:

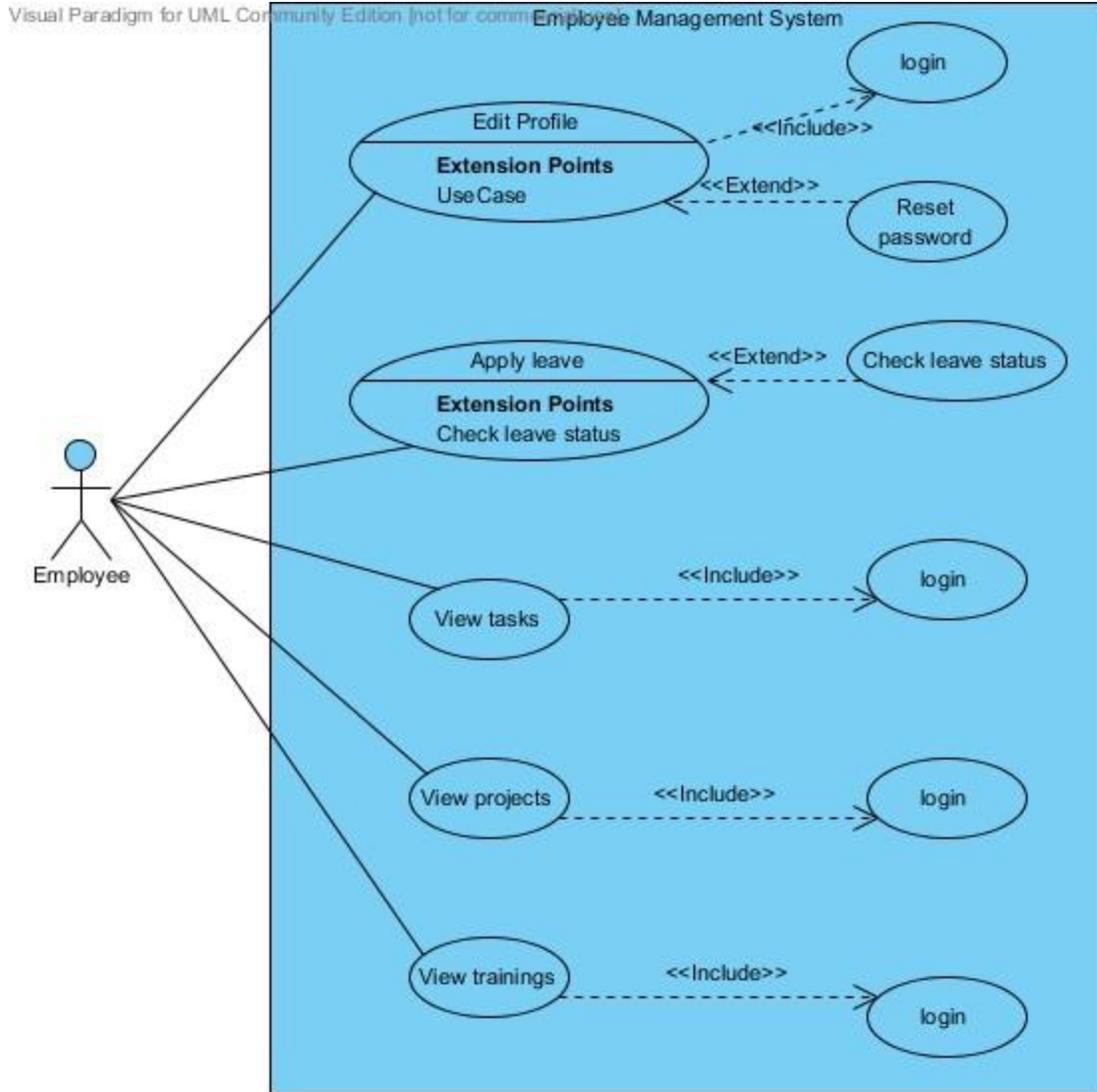


Figure 3.4 Employee use case

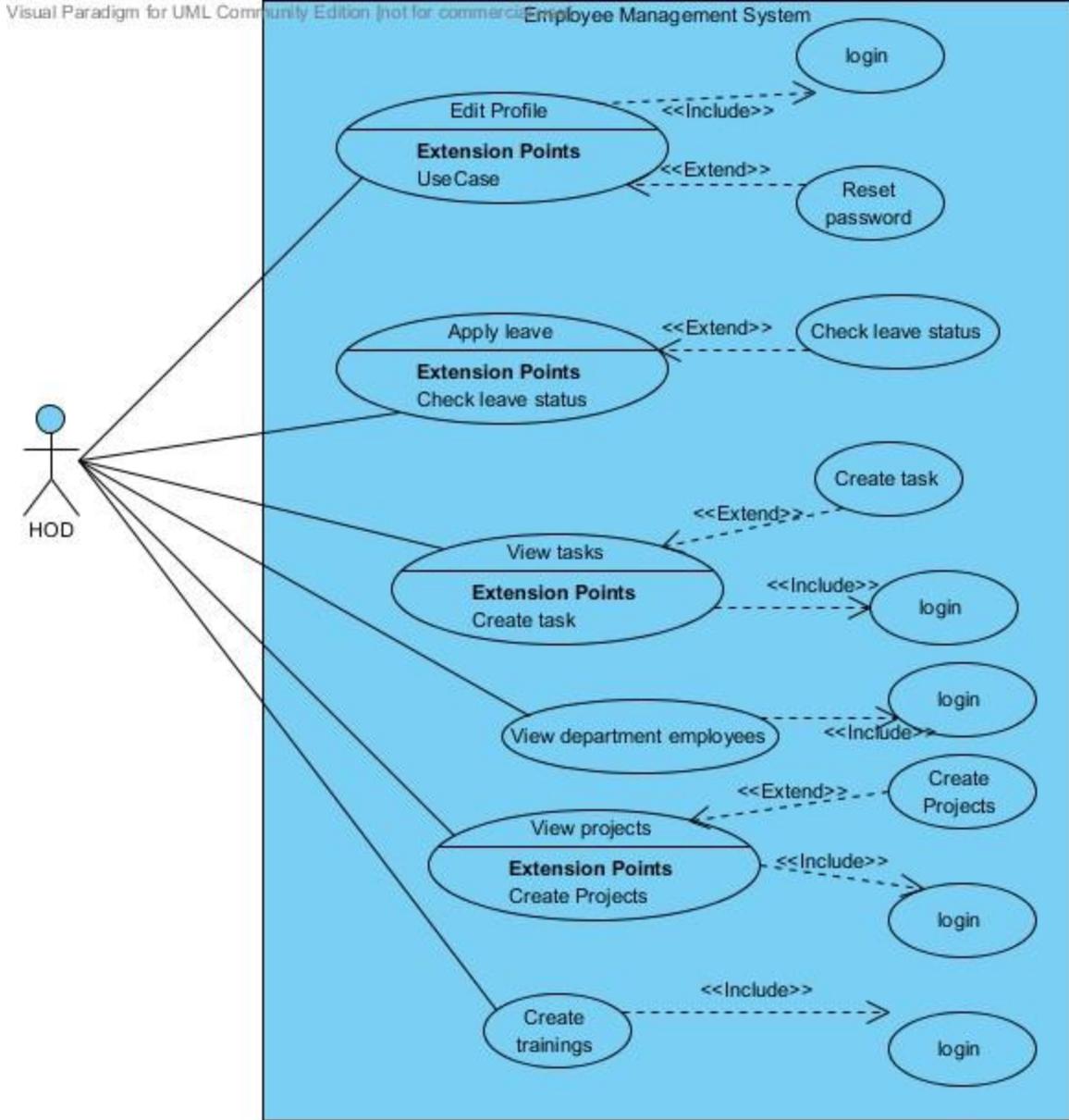


Figure 3.5 HOD use case

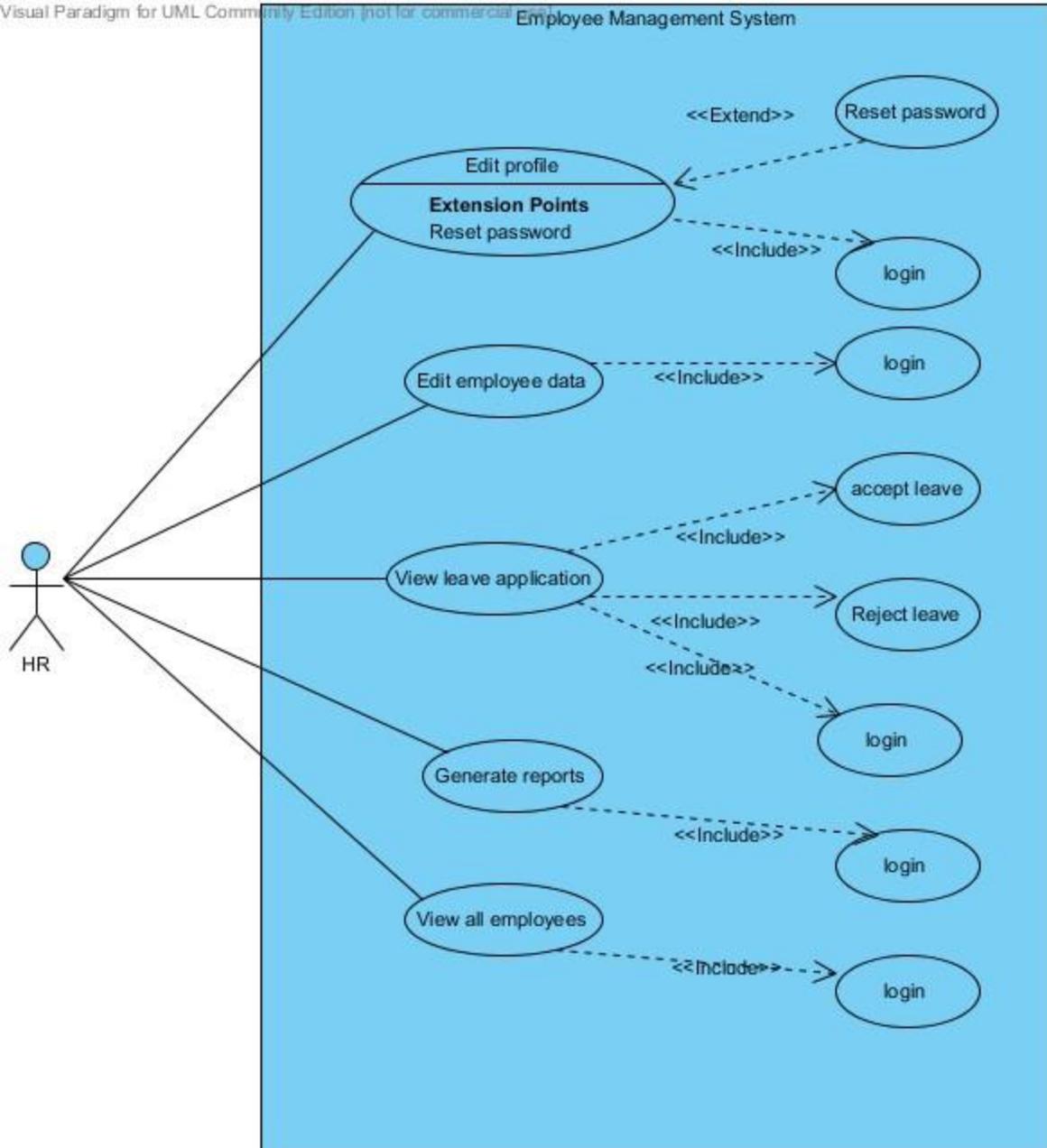


Figure 3.6 Human Resource use case

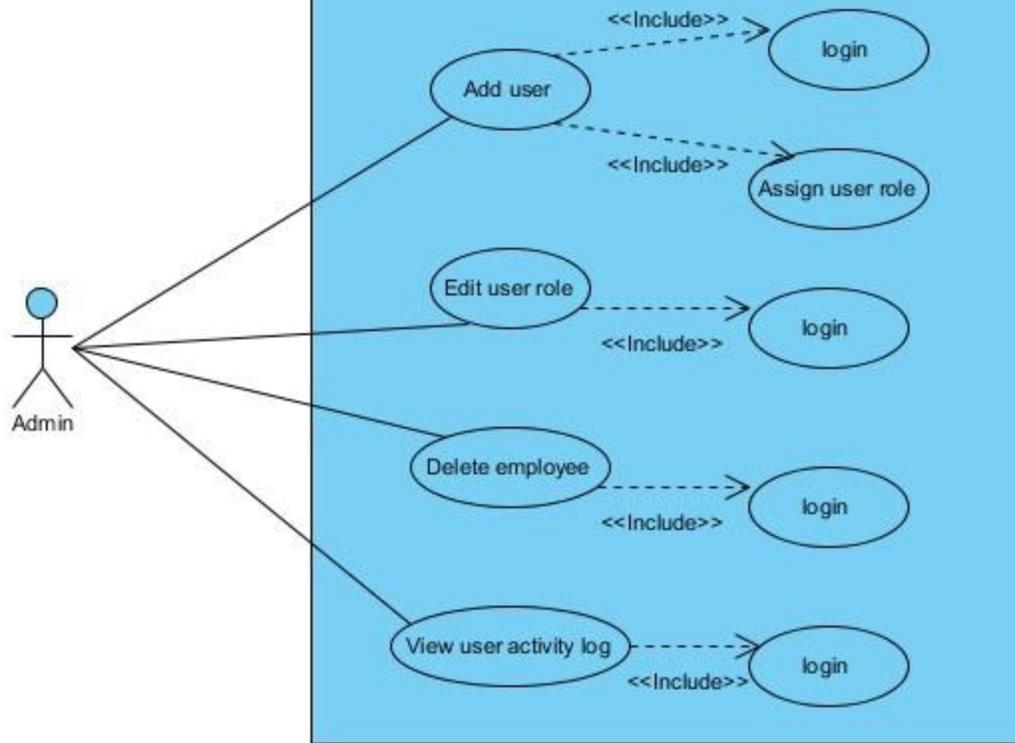


Figure 3.7 Admin use case

3.3.2 Class diagram

In the class diagram below, the Employee and Admin classes inherit from the User class. The employee class is also parent class to Human Resource class, Head of Department class and Ordinary employee class. An ordinary employee may include lecturers, accountants and all other employees that do not interact with the system with many privileges. These employees carry out the same operations.

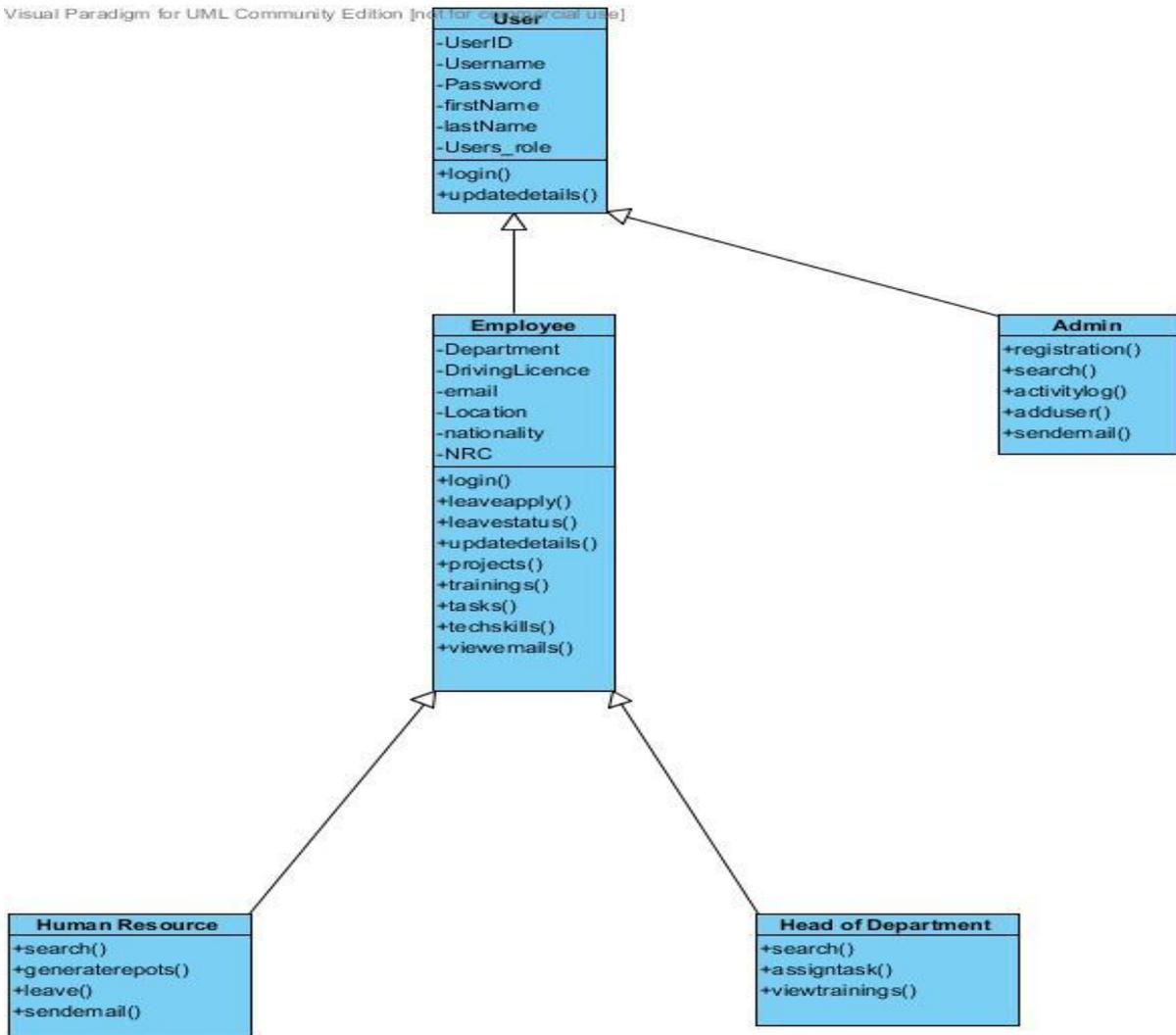


Figure 3.8 Employee Management System Class diagram

3.4 DEVELOPMENT TOOLS

This part of the chapter is an account of the technologies that could be used in the development of the system.

3.4.1 Front End Technologies

Front end-is a term used to characterize program interfaces and services relative to the initial user of these interface and services. It usually refers to the client side of an application. A front end application is one that users interact with directly. Turban et al (2008, p45) defines front end as the portion of an e-seller's portal, electronic catalogs, a shopping cart, a search engine and a payment gateway.

HTML

HyperText Markup Language (HTML) is a computer language devised to allow website creation. These websites can then be viewed by anyone else connected to the Internet. It is relatively easy to learn, with the basics being accessible to most people in one sitting; and quite powerful in what it allows you to create.

Having the basic knowledge of HTML will could help make or develop m-commerce websites and will also prove to be handy especially for editing and modifying web pages. Furthermore, it has some low cost benefits because of its many free online tutorials and advice support which is vital for m-commerce development.

JavaScript

JavaScript is a scripting language that is browser based and was developed by Netscape to enable web masters/authors to add interactivity and enhances behavior of web pages [11]. Some of the dynamic behavior that can be generated by JavaScript is validating form, performing specific actions e.g. after a mouse click, adding timestamps etc. JavaScript is an open language and anyone can use it. It also shares m any of the features and structures of the Java programming language, though it is not really related to Java. It was developed independently.

CSS

CSS is a style sheet language used to describe presentation and layout of HTML tags. CSS is used to enable separation of document content from document presentation. This refers to the

separation of document presentation aspects such as colors, layouts and fonts from the actual document content. CSS helps us achieve layout design and control much easier.

JSON

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.

JSON is built on two structures:

- A collection of name/value pairs. In various languages, this is realized as an *object*, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an *array*, vector, list, or sequence.

These are universal data structures. Virtually all modern programming languages support them in one form or another. It makes sense that a data format that is interchangeable with programming languages also be based on these structures.

jQuery

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

3.4.2 Back End Technologies

PHP

PHP, abbreviated to Hypertext Preprocessor is a server side web programming language that can be embedded into HTML. PHP is free software i.e. it is open source code. It is used for creating dynamic web pages that interact with the user and can include functionalities such as getting user input, manipulation of the input and storage of this data in a suitable DBMS. PHP is also easy to integrate with web pages. [12]

MySQL

MySQL stands for My Structured Query Language. It is the world's most popular open source relational DBMS. MySQL is available for free under the GNU General Public License for open source benefits/reasons related to development. Initially MySQL was free and some versions of it are still free though if you desire to use MySQL for commercial purposes you will need to purchase a license. It is non-proprietary, easily extensible and platform independent. Its downside is that it lacks a graphical user interface; therefore you need to know how the database works to make the most efficient use of it. [13]

3.5 SUMMARY

The core and emphasis of this chapter was the analysis of the current system. The various development tools used in the project were also discussed in this chapter. The next chapter will focus on the design characteristics and aspects of the system to be developed.

4 CHAPTER FOUR: DESIGN

4.1 INTRODUCTION

This chapter builds on the work done in the Analysis Chapter and gives documentation for the Design of the Employee Management System. The EMS is modeled in terms of objects and classes and their interactions with each other. Explanation of the proposed system is done as well structure of the Entity Relationship Diagram (ERD). Design of the User Interface is also discussed.

4.2 EXPLANATION OF THE PROPOSED SYSTEM

The proposed system is designed to eliminate all the drawbacks of the existing employee management software. The system shall be responsible for maintaining information about employees, thus their personal profile. The system shall incorporate leave management all the way from application to acceptance/rejection of leave requests as well as all employee projects with close monitoring of the projects from creation to completion and trainings to assist in monitoring active and inactive employees.

The main features to be added include:

- Employee profiles
- Leave management
- Task management
- Trainings
- Projects (Work Breakdown Structure)
- Notifications
- Employee Self-Service (ESS)
- Resume Tracking

4.3 SYSTEM AND ALGORITHM FLOWCHARTS

Activity Diagrams are used to model different aspects of a system. The following activity diagram is used to model the leave application function.

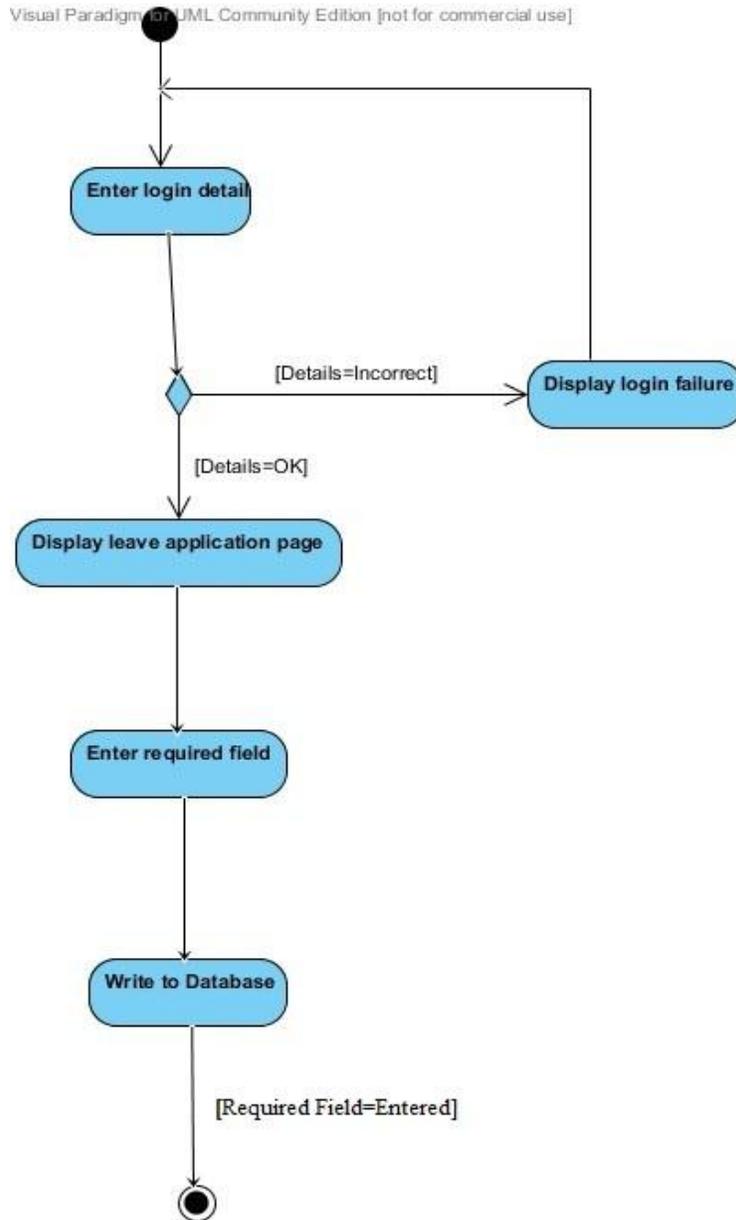


Figure 4.1 Activity diagram for leave application

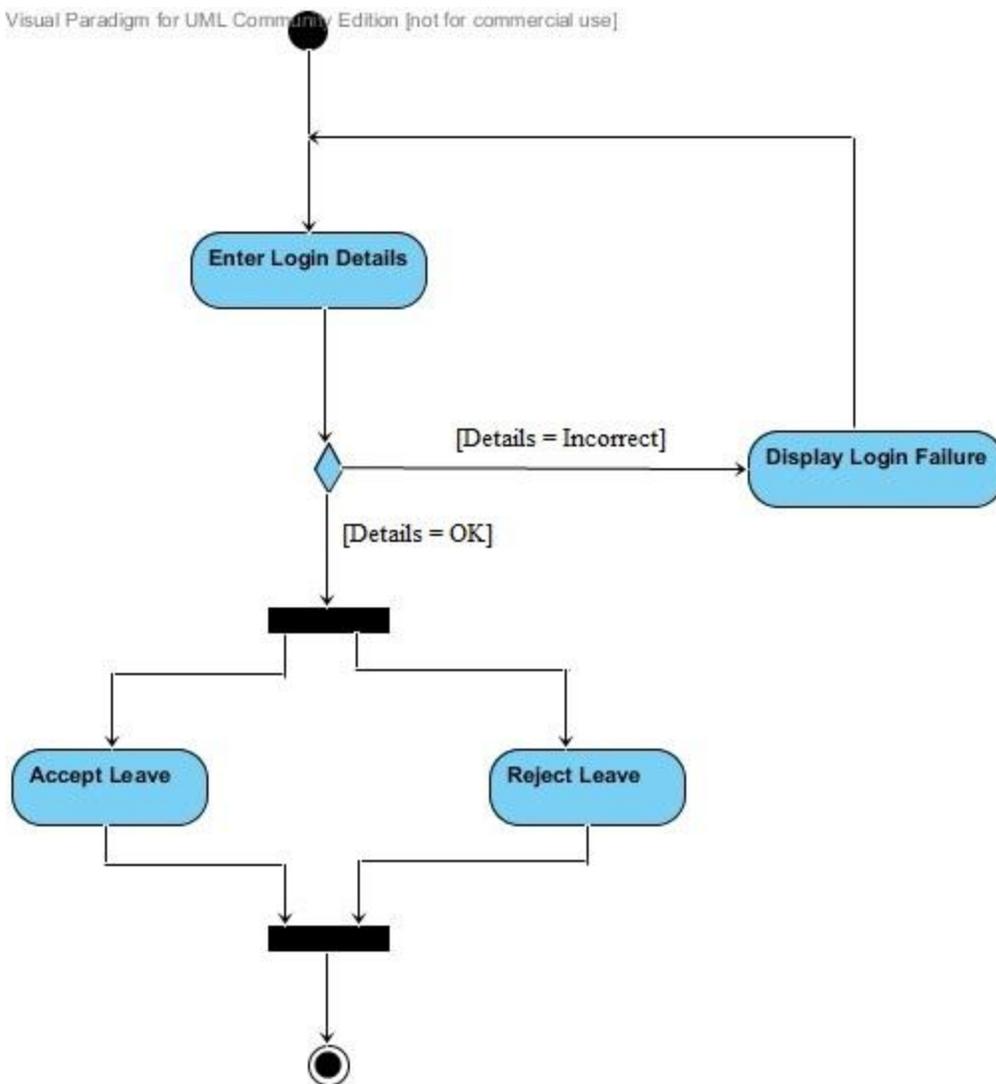


Figure 4.2 Leave Acceptance/Rejection

4.4 SYSTEM STRUCTURE CHART DFD AND ERD

4.4.1 Data Flow Diagram (DFD)

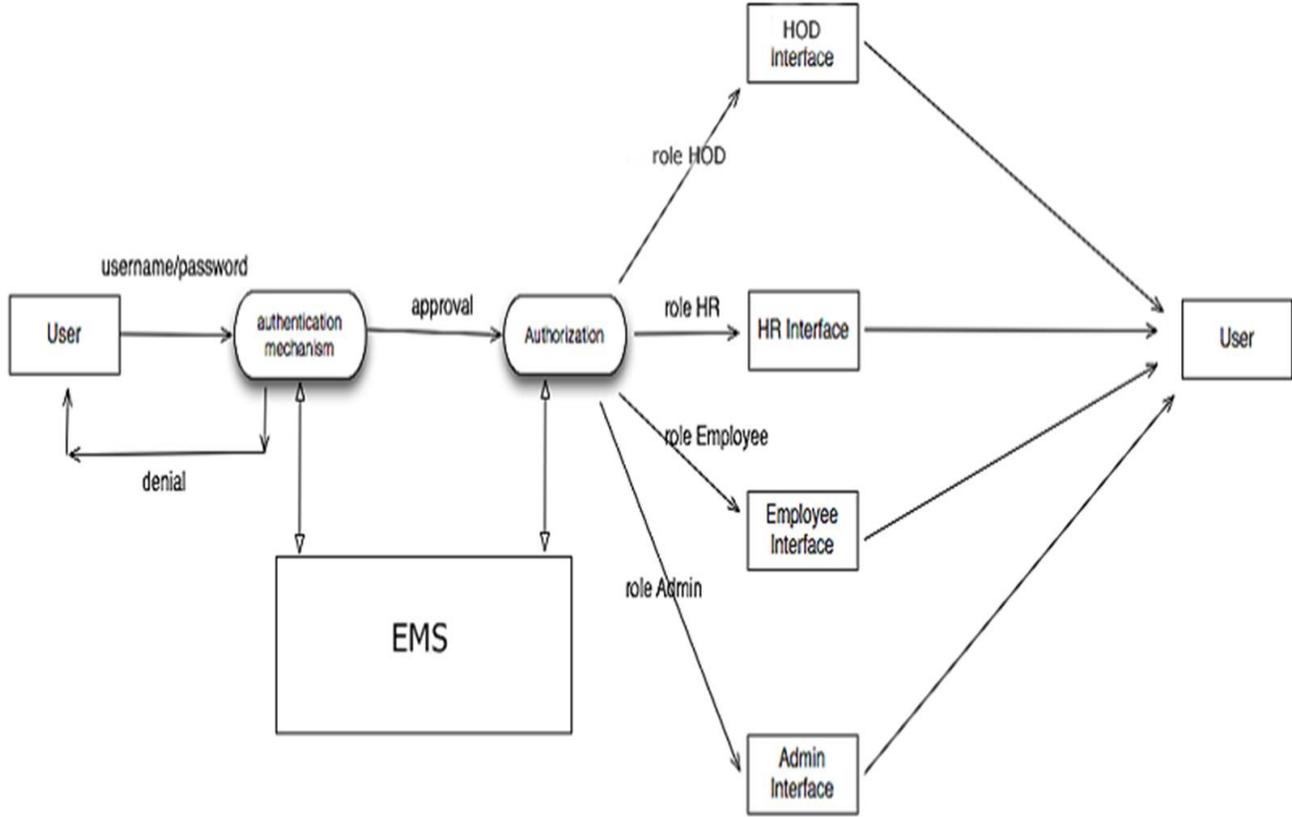


Figure 4.3 Authorization & Authentication DFD

4.4.2 ENTITY RELATIONSHIP DIAGRAM (ERD)

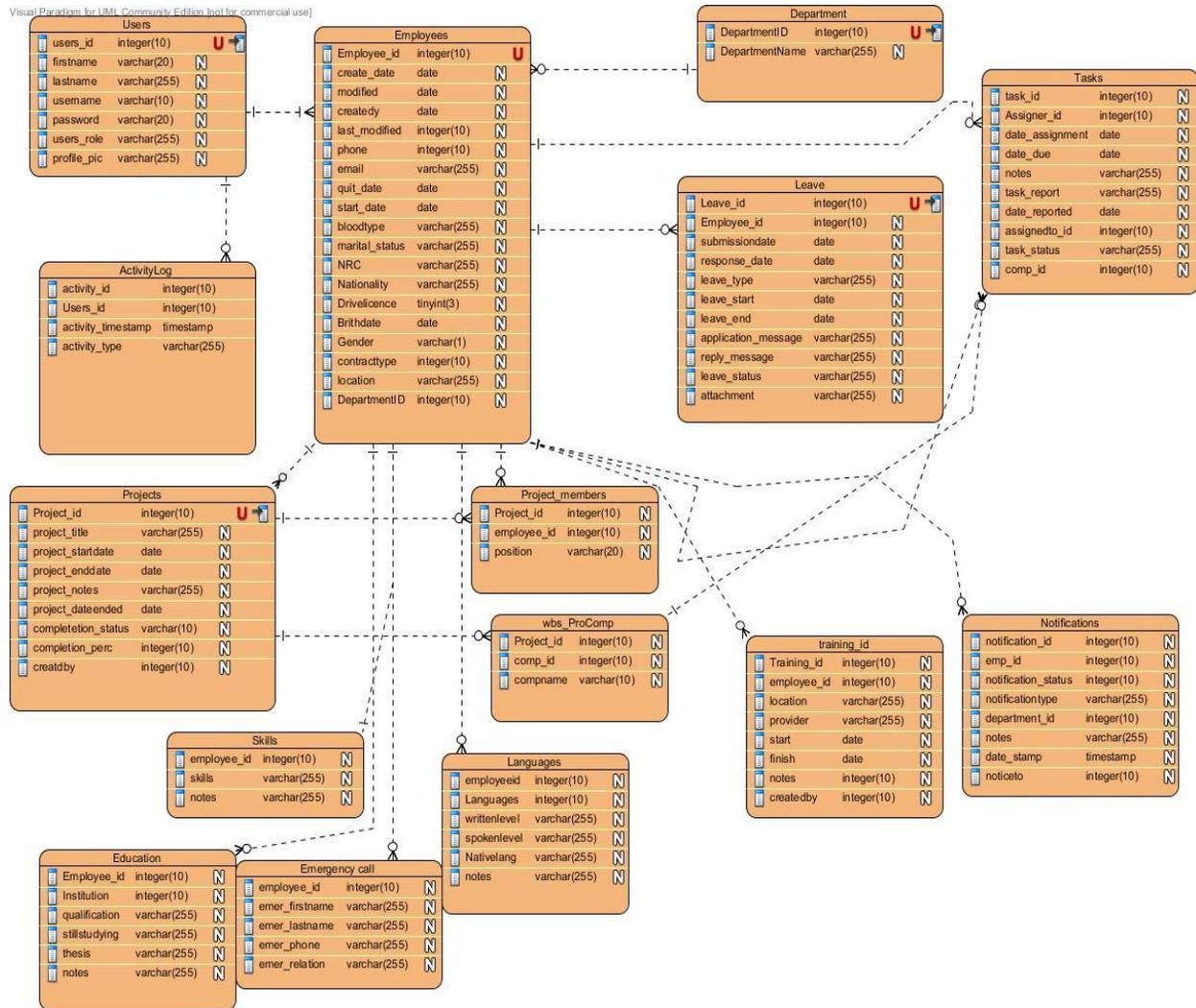


Figure 4.4 Entity Relationship Diagram

4.5 STORYBOARD

4.5.1 Sequence Diagrams

Sequence diagrams help in the identification of a detailed level of the operations required to implement the functionality depicted by a use case model.

Scenario 1: Admin add new employee

1. The user logs in by providing correct username and password.
2. If username and password are not found on the database access into the system is denied.
3. If the credentials are identical to the ones found on the database, access is granted.
4. User enters the details of the new employee.
5. The user input is written to the database.

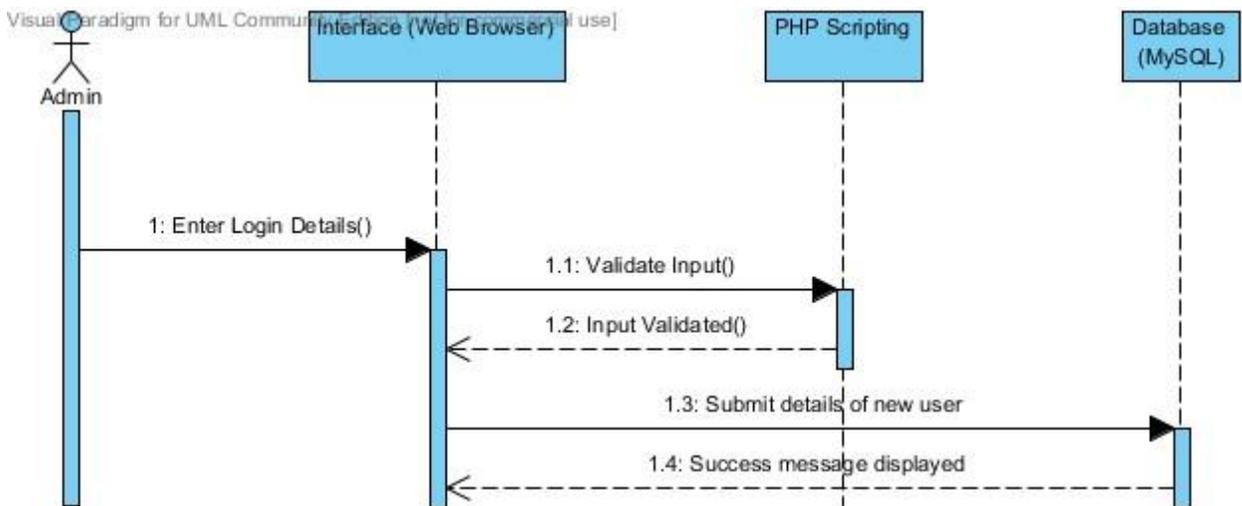


Figure 4.5 Add new user sequence diagram

Scenario 2: HOD Create Projects

1. The user logs in by providing correct username and password.
2. If username and password are not found on the database access into the system is denied
3. If the credentials are identical to the ones found on the database, access is granted.
4. The user creates a project and assigns members.
5. The user input is written to the database.

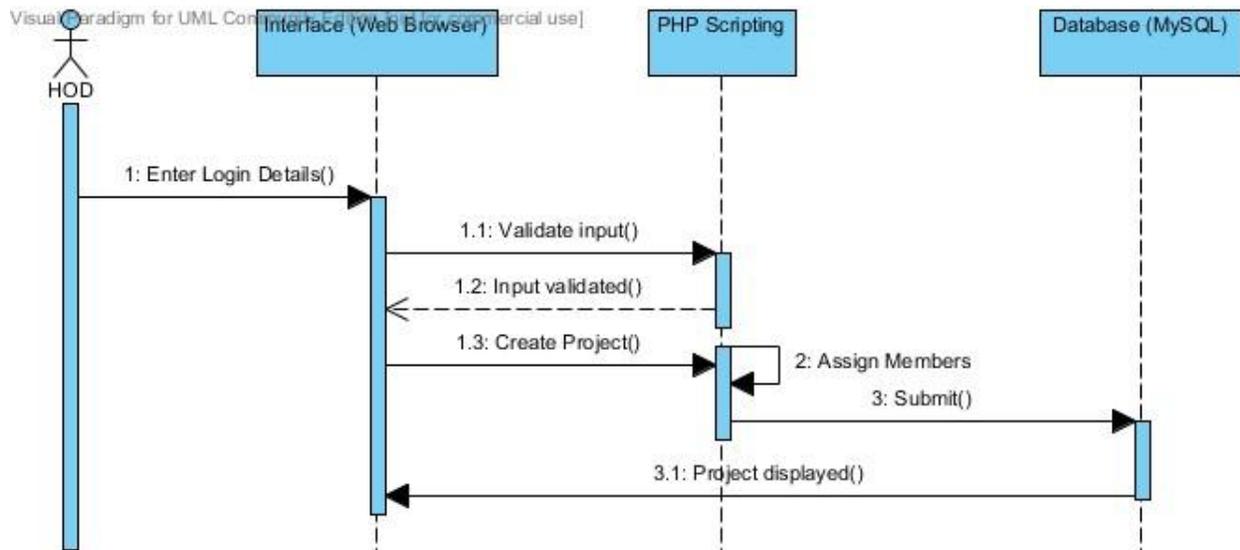


Figure 4.6 Create Projects Sequence Diagram

Scenario 3: Employee leaves application

1. The user logs in by providing correct username and password.
2. If the username and password are not found in the database access into the system is denied.
3. If the credentials are identical to the ones found on the database, access is granted.
4. User requests for leave form.
5. User enters leave details.
6. Details are written to the database.
7. A message confirming details have been submitted is displayed to the user.

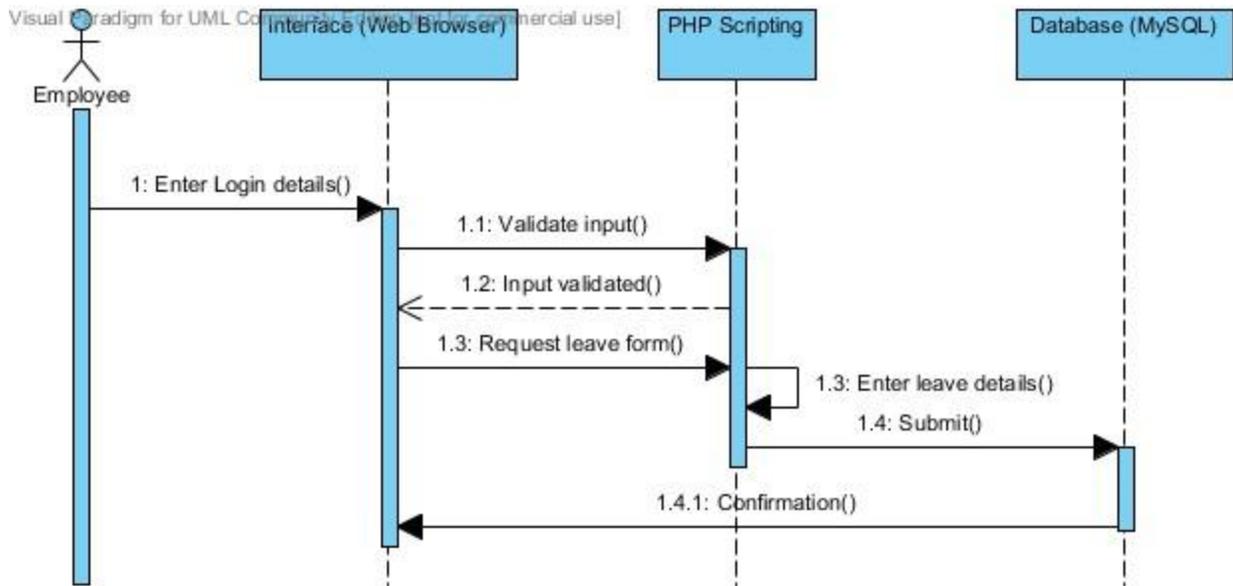


Figure 4.7 Employee Leave Application Sequence Diagram

Refer to appendix A for Report generation and edit account sequence diagrams.

4.6 INTERFACE DESIGN

The web application was created with the following design considerations in mind:

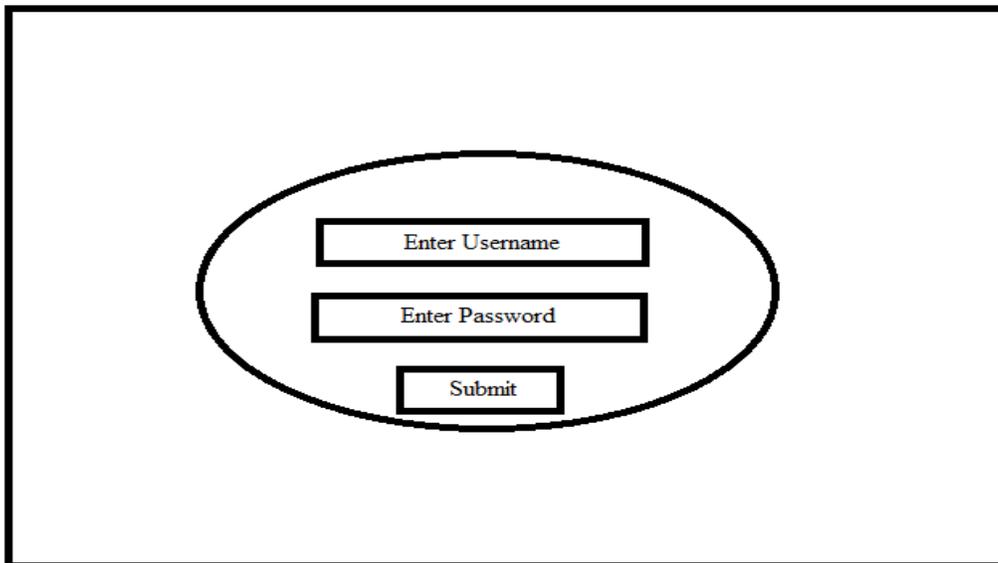
- **Consistent.** The website should have a similar look and feel on every page. Every page should have the same header/logo, heading style, fonts, navigations etc.
- **Efficient and easy to maintain.** This refers to the fact that there is need to separate content from layout, so that you can easily change your page design without editing every page on the site.
- **Layout.** The layout of each page should have a good contrast between the text and background area. This helps considerably with visibility as it will be difficult to read the

text if it is almost the same color as the background. Monitor size should also be taken into consideration.

- **Easy to navigate and use.** Users should not have a hard time trying to navigate the site. Navigation links should be consistent and clearly labeled. All navigation links should also be working properly and should point to the intended page/site.
- **Browser compatible.** When designing the site consider different browser environments. Extensive testing should be done on each page in all the major browsers and the design changed appropriately to cater for all.
- **Visually appealing.** The use of color, text, fonts and graphics should be carefully considered and used to ensure that the site is visually appealing to its visitors.
- **Speed.** The performance of a website is mostly rated by its up -time and downtime. These terms refers to the amount of time it takes the site to respond to requests. Graphics should be kept to a minimum to allow the site to load faster. The pages on the site should load within an acceptable time e.g. under 10seconds. [14]

4.7 Sketches of graphics

Below is the interface design for the website application:



The diagram shows a login interface design. It is enclosed in a large rectangular border. Inside this border is a horizontal oval shape. Within the oval, there are three rectangular boxes stacked vertically. The top box contains the text 'Enter Username', the middle box contains 'Enter Password', and the bottom box contains 'Submit'.

Figure 4.1 Login Interface

Company Logo	Navigation Menu
Profile picture & Username	Navigation Menu
Content	

Figure 4.2 Employee Interface after Login

Company logo	Profile picture & Username	Menu
Navigation Menu	Content	

Figure 4.3 Admin/HR/HOD Interface after Login

4.8 SUMMARY

This Chapter has specified the design of the EMS. The aspects of the design that have been discussed are system design, interface design, and database design by providing the DFD and ERD. The next chapter looks at the implementation of the EMS. This depends on the design specification given in this chapter.

5 CHAPTER 5: IMPLEMENTATION

5.1 INTRODUCTION

This chapter will explore the different aspects concerned with the implementation of the developed system. This project was concerned with the development and implementation the employee management system. We began with analysis of the current and proposed systems, the design of the system to be developed, and in this chapter we shall deal with implementation of the developed system.

5.2 DESCRIPTION OF DEVELOPED SYSTEM

The developed system encompasses various activities associated with managing employee information.

The main functionalities available in this system are:

- Maintaining employee profiles
- Leave management
- ESS
- Task management
- Employee Trainings
- Project Management

All these features include the ability to add user, update (edit), and retrieve through search results. It also contains a report generation system that can be saved in a pdf file format.

The system works in the following manner:

5.2.1 Accessing the system

Various companies and organizations may have different employee structures and hierarchy.

Being generic, the developed System has four main access levels which are:

- Employee
- Head of Department (HOD)
- Human Resource Manager (HR)
- Administrator

All users are presented with the same login interface. User must login the system by means of valid username/password combination. After access is granted to the system, the admin can add a new user to the system by entering the basic information which are the full names and email address. The admin also assigns the new user a role which will determine the access level. During the process of user registration, the all users are issued with a unique username and password combination. Seeing that the system holds private employee information, the admin has the ability to monitor all activity logs into the system by date and time. The newly added user logs into the system with a default password which can later be changed to a more secure password. All employees can edit basic information such as newly acquired technical skills and emergency contacts. Employees can apply for leave by filling in a form as well as submitting an attachment to support their leave request.

The HOD has the ability to view all employees under his/her department, assign a task and trainings. The HOD can also create a project, add members to the project and create a work breakdown structure. Being an employee, the HOD can apply for leave as well as check leave days accrued.

Upon logging in to the system, the HR manager gets notifications on the leave applications submitted and has the ability to approve or reject leave requests as they are submitted. The HR carries out all employee tasks which include the ability to view and edit basic details, view pending tasks, projects and trainings. The HR also has to the ability to generate employee reports in PDF format.

5.3 TECHNICAL DETAILS OF IMPLEMENTED SYSTEM

5.3.1 Model View Controller architecture (MVC)

In the implementation, as shown in figure 5.1, the whole application is broken down into a series of top-level components which may be referred to as tasks, actions, functions, operations or transactions (that's *user* transactions, not *database* transactions), each of which is may be related to a Use Case. Each transaction component references a single controller, one or more models, and usually a single view. Some components do not have a view as they are called from other components in order to perform a service, and once this service has been completed they return

control to the calling component. Each component is self-executing in that it deals with both the HTTP GET and POST requests. [15]

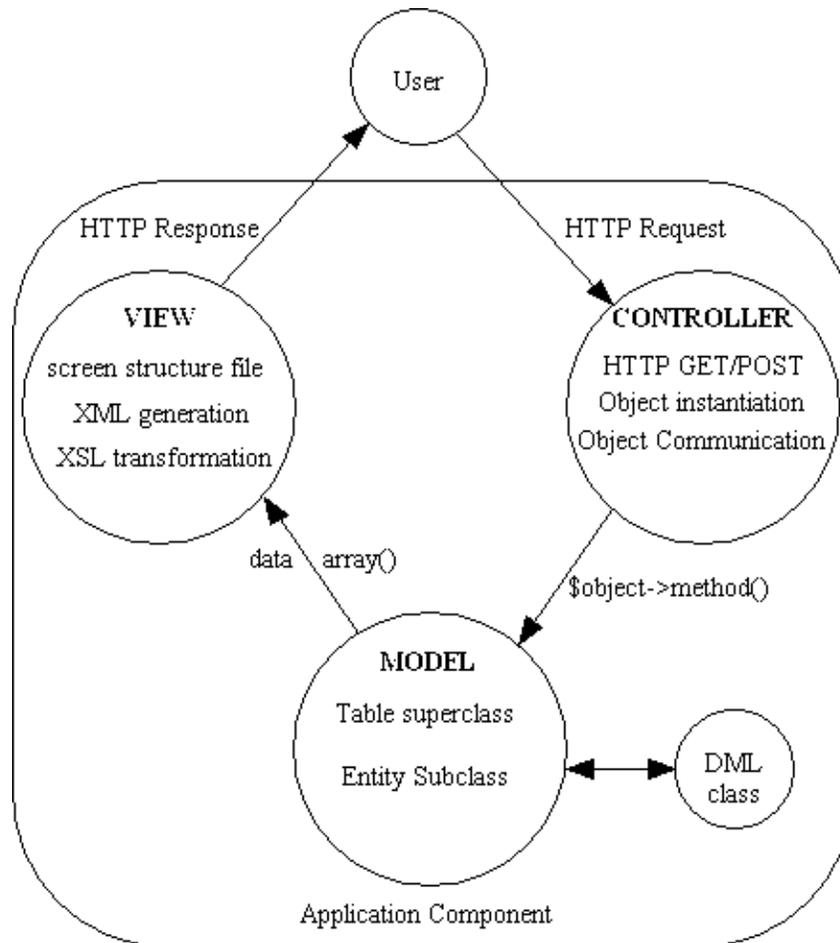


Figure 5.1 Model View Controller Architecture

5.3.2 MySQL Triggers

5.3.2.1 Implementation of MySQL Triggers

In MySQL, a trigger is a set of SQL statements that is invoked automatically when a change is made to the data on the associated table. A trigger can be defined to be invoked either before or after the data is changed by INSERT, UPDATE or DELETE statements. MySQL allows you to define maximum six triggers for each table.

- BEFORE INSERT – activated before data is inserted into the table.
- AFTER INSERT- activated after data is inserted into the table.
- BEFORE UPDATE – activated before data in the table is updated.
- AFTER UPDATE - activated after data in the table is updated.
- BEFORE DELETE – activated before data is removed from the table.
- AFTER DELETE – activated after data is removed from the table.

When you use a statement that makes change to the table but does not use INSERT, DELETE or UPDATE statement, the trigger is not invoked. For example, the TRUNCATE statement removes the whole data of a table but does not invoke the trigger associated with that table.

There are some statements that use the INSERT statement behind the scenes such as REPLACE statement and LOAD DATA statement. If you use these statements, the corresponding triggers associated with the tables if available will be invoked. [16]

5.3.3 System installation

The system was developed and tested on a laptop computer running Windows 7, and the XAMP Server. In order for the Web application to be accessible via the Internet it will have to be installed on a Web Server running Apache, PHP and MySQL. The suitable operating system for the web server will be Linux as it is more stable and less prone to virus but a windows based platform will equally do the job just as well. A suitable domain name will have to be chosen and registered in order for the web application to be accessed via a URL and hosting and administration fees paid to the web hosting company of choice either annually or monthly depending on the package and terms agreed upon. The web application will be accessible via most of the popular web browsers on the market. A suitable web browser e.g. Mozilla Firefox will have to be installed on the client machine wishing to access the web application. [16]

5.3.3 Algorithms

MD5 Encryption

MD5 algorithm was used for password encryption. MD5 stands for **M**essage **D**igest algorithm **5** is a widely used cryptographic hash function. The idea behind this algorithm is to take up a random data (text or binary) as an input and generate a fixed size “hash value” as the output. The input data can be of any size or length, but the output “hash value” size is always fixed. Here is an example (Figure 5.2) of MD5 Hash function at work:

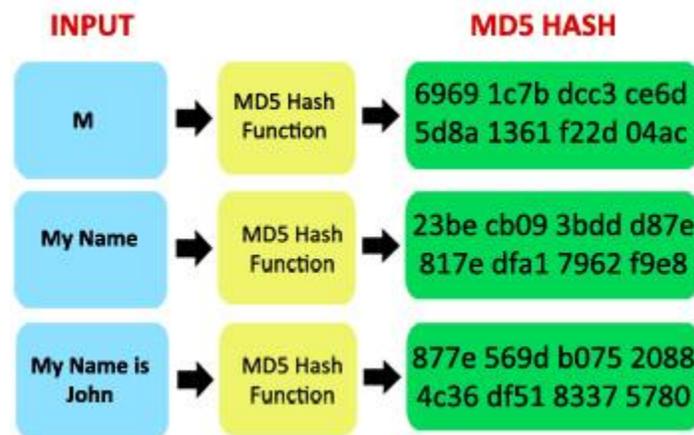


Figure 5.2 MD5 Encryption procedure

As can be seen from the above example, whatever input size is given, the algorithm generates a fixed size (32 digit hex) MD5 hash. [17]

5.3.4 Pseudocode

Login into the system

Startup system

Enter username and password

On clicking the login button

Connect to database

Query database to know whether user credentials are correct

If not

Deny access and return login page with an error message

If correct

Check if credentials are for administrator

If yes

Allow login

Set admin session

Redirect administrator to admin home page

If no

Allow login

Set user session

Redirect user to user home page

Add new user

Check if administrator is logged in

If correct

Check if all fields entered are correct

If not

System message: please enter all fields

If correct

Registration of new user successful

Apply for leave

Check if employee is logged in

If correct

Check if all fields are entered

If not

System message: please enter fields

Check if file has being attached

If not

System message: please attach file

If correct

Leave request has being made

5.4 SCREENSHOTS OF DEVELOPED SYSTEM

Refer to appendix C for screenshots of developed system.

5.5 SUMMARY

This chapter has outlined how the employee management system has been implemented using the Model View Controller (MVC) architecture. The method selected for systems change-over has been highlighted and justified, lastly the chapter concludes by showing how the system can be deployed. The next chapter is on Testing, it focuses on the tests carried out to ensure the system functions according to its specifications.

6 CHAPTER SIX: TESTING AND VERIFICATION

6.1 INTRODUCTION

Testing is very important and critical to the success of any project that aims at delivering working software. There are many types of testing that a system may be subjected to, however only the ones in the testing objectives will be carried out for this system.

6.2 SCOPE

The overall purpose of testing is to ensure the Employee Management System meets all of its functional and business requirements. The purpose of this chapter is to describe the overall test plan and strategy for testing the system.

6.3 TESTING GOALS

The goals in testing this system include validating the quality, usability, reliability and performance of the application. Testing will be performed from a black-box approach. Tests will be designed around requirements and functionality.

6.4 CONFIRMATION TESTING

Confirmation testing or re-testing: When a test fails because of the defect then that defect is reported and a new version of the software is expected that has had the defect fixed. In this case we need to execute the test again to confirm that whether the defect got actually fixed or not. This is known as confirmation testing and also known as re-testing. It is important to ensure that the test is executed in exactly the same way it was the first time using the same inputs, data and environments. [18]

Hence, when the change is made to the defect in order to fix it then confirmation testing or re-testing is helpful.

6.5 REGRESSION TESTING

During confirmation testing the defect got fixed and that part of the application started working as intended. But there might be a possibility that the fix may have introduced or uncovered a different defect elsewhere in the software. The way to detect these ‘unexpected side-effects’ of fixes is to do regression testing. The purpose of a regression testing is to verify that modifications in the software or the environment have not caused any unintended adverse side

effects and that the system still meets its requirements. Regression testing are mostly automated because in order to fix the defect the same test is carried out again and again and it will be very tedious to do it manually. Regression tests are executed whenever the software changes, either as a result of fixes or new or changed functionality. [19]

6.6 TEST PLANS AND RESULTS

The Test Plan is derived from the Requirements, Functional Specifications, and detailed Design Specifications. The Test Plan identifies the details of the tests, identifying the associated test case areas within the product. [20]

Test Case	Test Purpose	Test Condition	Expected Outcome	Actual Result
Login	Check username and Password	If user details are not correct, display error message	Grant Access to the applicable main system	User successfully logs into the system upon submission of correct login credentials.
Add new user	To ensure that a new user is added to the system successfully.	If user already exists in the system, error message should display.	New user should be successfully added to the system.	If email address entered already exists in the system, error message is displayed. If the email address of the new employee does not exist in the system, new employee is successfully added.

Edit personal details	To ensure that once different details are provided on the edit personal details form and submitted, these details are altered in the database to reflect the recent changes	On the edit personal details form provide different information from what is currently being displayed	When the form is altered the details should be altered in the database and a confirmation message of the change should be displayed.	Once the data in the form is altered and the submit button clicked the details in the database are altered and a confirmation message of the change is displayed.
Apply leave	To test if all employees can successfully apply for leave.	Whenever an employee applies for leave, information as well as attachments should be submitted to the HR manager.	Leave request should be sent when all required fields are submitted and necessary documents have being attached to the request.	Leave request is submitted as required and a message of success is displayed.
Create project	Test if a Head of Department can create a project and later assign a project team.	Whenever a project is created, HOD should be able to assign project team and view members.	A project should be created as well as coming up with a project team and WBS.	Project is created and HOD can view the project team.

View notifications	Test if employee is notified when leave has being accepted or rejected and when new tasks, trainings or projects have being assigned.	If employee has being added to a project, he/she should get a notification.	Notifications should be displayed on the employee interface whenever a task has being assigned; employee has being added to a project team project, or trainings. As well as when a leave request has being accepted or rejected.	Notifications appear on the employee interface.
Upload picture	Test if users can upload a profile picture associated to their account	Employee should be able to upload a profile picture if they so wish.	Employee is able to upload profile picture.	Message of success is displayed when employee uploads picture and they are asked to log out and back in for changes to take place.
Generate reports	Test if HR can generate employee reports.	To ensure that the selected report is displayed	Once a choice of report is made by clicking the link of choice the	When the choice of report is made and link clicked a report is

			report should be displayed.	displayed.
--	--	--	-----------------------------	------------

Table 6.1 Shows system test plan and results

View appendix D for screenshots of test results.

6.7 SUMMARY

The chapter discussed how the proposed system was subjected to various types of testing. This brought to light why it is very cardinal to test a new system before it is introduce on the main stream of an organization’s business.

7 CHAPTER SEVEN: CONCLUSION

7.1 INTRODUCTION

The aim of this chapter is to draw conclusions of the work done or achieved and to give an assessment of the completed system, discuss the Problems faced, limitations of the system and give future recommendations on how the system can be improved.

7.2 RESULTS

The software product produced was fairly good, it achieved most of the user requirements, the user interface is good and is very easy to navigate, and even novice users can find their way around the web application easily. The client side validation is excellent. The lack of integration with a payroll system is the major drawback and the system was also unable to generate structured reports i.e. reports based on specific information the Human Resource is interested in.

7.3 PROBLEMS FACED

The biggest challenge faced was getting hold of employee information for Human Resource. The project was initially aimed at implementing an HR system for UNZA but that proved to be a challenge because employee information is very critical. This led to the implementation of a generic system thereby drawing a few assumptions were possible meaning that requirements kept on changing as different views were put into consideration as compared to software which has a specific customer

7.4 LIMITATIONS

7.4.1 Custom report generation

In the designed system, the HR manager can generate a report containing all the employee information from personal profile, skills and educational background. Sometimes, the HR manager may be looking for specific employee information e.g. Educational background, technical skills or language and location in cases where a transfer is to be made, this poses a challenge because the HR will always have to generate a report of all the information about an employee, some of which may be not be needed.

7.4.2 Head of Department heads all Projects

The head of department creates a project and assigns tasks to a project team, therefore he becomes the project manager. This brings about a challenge in cases where a project must be managed by an employee that does not have the role of head of department.

7.4.3 Leave Management

Successful leave submission can only be made when an attachment is added to the leave request. This is a challenge for leave requests that may not have supporting documents.

7.4.4 Browser support

The highly sleek and intuitive interface was made in order to improve Human Computer Interaction (HCI). However, this comes with challenges because lower versions of Internet Explorer (i.e. IE9 and lower) do not support certain features such as column-fill, column-span, align-self, backface-visibility etc.

7.5 FUTURE WORK

7.5.1 Leave Management

The leave management module can be improved by having all leave requests approved by the head of department before submission rather than going straight to the HR manager. This feature is important because the HOD /Supervisor should know which of his/her employees which to go on leave.

7.5.2 Integration with payroll system

In order for the system to be more comprehensive, I'd recommend an integration of the system to a payroll system that will enable employees view and download their pay slips on demand.

7.5.3 Employee Performance

The designed system provides the HOD with the ability to assign tasks to project members. If further worked on, this functionality can assist in determining the performance of employees based on their ability to finish tasks on time.

7.5.4 Information archiving

A system holding all the employee information should have some form of archiving system so that retired, suspended or fired employees are archived rather than been completely deleted from the system. This is so because cases may occur where details of an ex-employee may be required especially in cases where an employee did several projects and there details are required for future reference.

7.6 LEARNING EXPERIENCE

This project assisted me to gain a practical experience and apply the knowledge assimilated from the previous courses undertook. Putting the knowledge gained earlier and applying different techniques from past courses was interesting and certain concepts, tools and techniques only made sense after seeing their application in a real world scenario. It was extremely challenging at times but it has been a great and worthwhile learning experience.

There is not at all any doubt that the employee management system would be an asset to any company, small or large.

7.7 CONCLUSION

In this chapter, the results were discussed, problems faced and limitations were elaborated. Future recommendations for the extension and improvement of the system have also been discussed as well as well as an assessment of achieved functionality.

8 REFERENCES

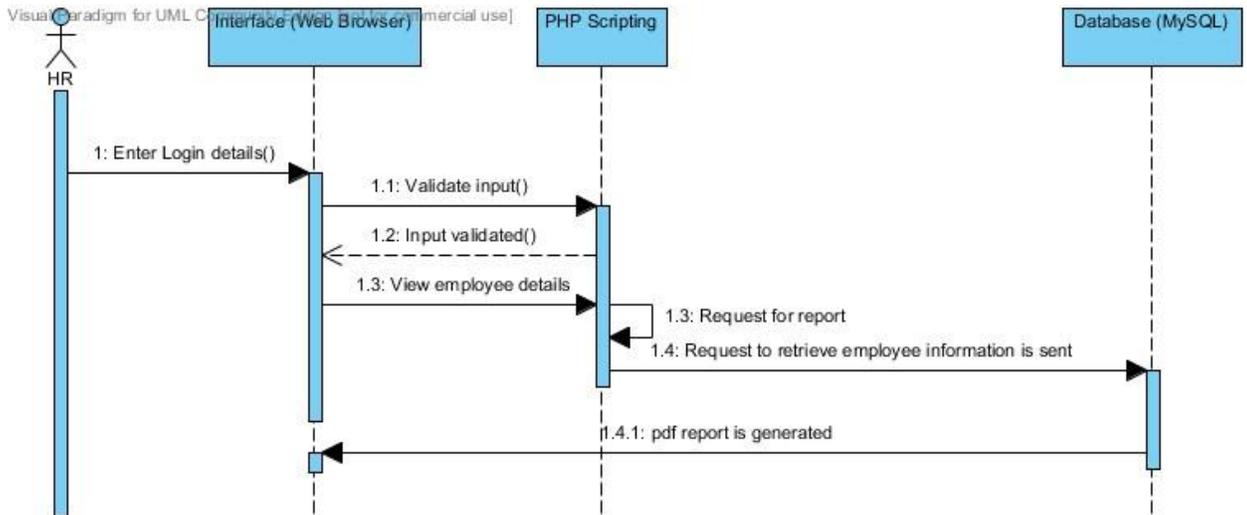
- [1] OrangeHRM Open Source, Retrieved: November 4, 2013. From:
<http://www.orangehrm.com/open-source-product-features-pim.shtml>
- [2] A.S.SyedNavaz, A.S.SyedFiaz, C.Prabhadevi, V.Sangeetha, S.Gopalakrishnan, “Human Resource Management System”, IOSR Journal of Computer Engineering (IOSR-JCE), Volume 8, Issue 4 (Jan. – Feb. 2013) Page 62-71.
- [3] Julie Bulmash, “Human Resource Management and Technology”, Chapter 3
- [4] TECH HRM (Human Resource Management System), Retrieved: November 4, 2013. From: <http://www.techjetsolutions.com/brochure/TECHHRM.pdf>
- [5] Renae Broderick, John W. Boudreau, “Human resource management, information technology, and the competitive edge”, Academy of Management Executive, 1992 Vol. 6 No. 2
- [6] Centralized Employee Information, Retrieved: July 8th, 2014. From: <https://www.zoho.com/people/employee-management-system.html>
- [7] HR and Employee Mngement Software, Retrieved: July 8th, 2014. From: <http://www.getapp.com/hr-employee-management-software>
- [8] Ian Sommerville, “Software Engineering”, 9th Edition, Addison-Wesley, 2011.
- [9] What is Enterprise Resource Planning (ERP)? Webopedia. Retrieved: January 17, 2014, from: <http://www.webopedia.com>
- [10] Avison, D. and Fitzgerald, G. (2003).Information systems Development Methodologies, Techniques and Tools.3rd Edition. McGraw-Hill Education Limited Bershire
- [11] Zhiming, L, July 2002 ,Object-Oriented Software Development with UML Retrieved: July 16th 2010. From: <http://www.iist.unu.edu/www/docs/techreports/reports/report259.pdf>

- [12] Juan Manuel Munoz Palacio, Information systems development methodologies for Data-driven Decision Support Systems, 2010,
- [13] Deitel, PJ & Deitel, HM, 2008, Internet & World Wide Web How To Program, Dorling Kindersley, India
- [14] Web design best practices checklist 2009. Retrieved: October 17, 2009, from: <http://terrymorris.net/bestpractices/>
- [16] Connolly, T, Begg, C, 2005, Database Systems A Practical Approach to Design Implementation and Management, 4th Edition, Dorling Kindersley, India
- [17] Avison, D. and Fitzgerald, G, 2003. Information systems Development Methodologies, Techniques and Tools, 3rd Edition, McGraw-Hill Education Limited, Berkshire
- [18] ASP.NET, 2010. Retrieved: July 9th 2010, from: <http://www.startvbdotnet.com/aspsite/asp/>
- [19] Introducing JSON, Retrieved: June 21, 2014, from: <http://www.json.org>
- [20] Employee Self Service, Retrieved: January 15, 2014, from: http://en.wikipedia.org/wiki/Employee_self-service_%28web-based_application%29

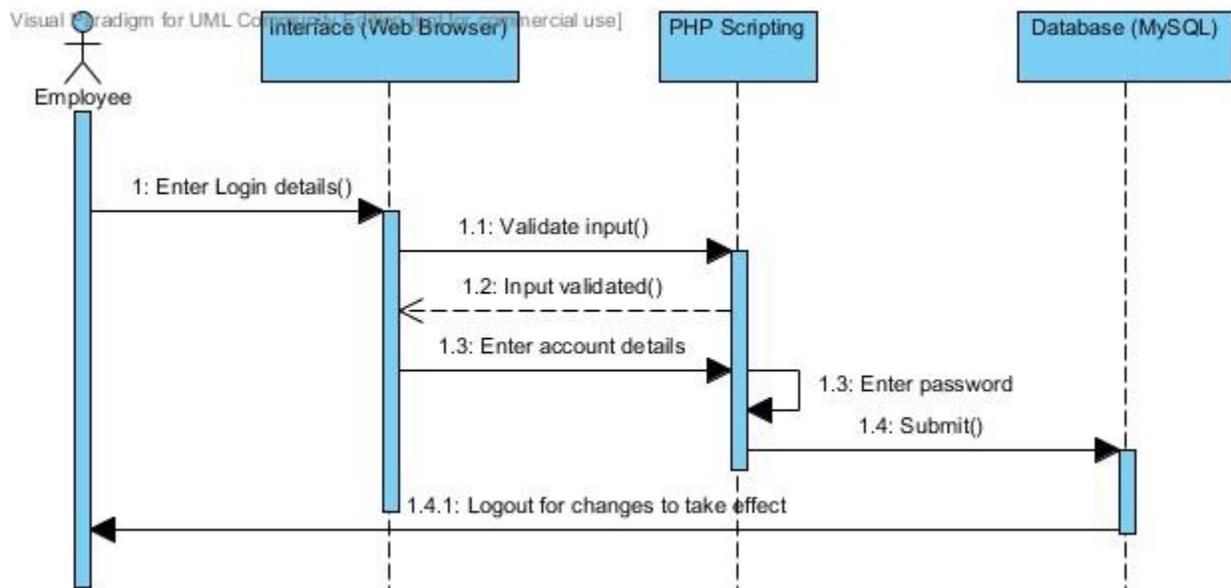
9 APPENDIX

9.1 Appendix A – Sequence Diagrams

9.1.1 Report generation



9.1.2 Edit account details



9.2 Appendix B – Sample Code

9.2.1 Database Connectivity

```
$active_group = 'default';
$active_record = TRUE;
$db['default']['hostname'] = 'localhost';
$db['default']['username'] = 'root';
$db['default']['password'] = 'root';
$db['default']['database'] = 'emp_db';
$db['default']['dbdriver'] = 'mysql';
$db['default']['dbprefix'] = "";
$db['default']['pconnect'] = TRUE;
$db['default']['db_debug'] = TRUE;
$db['default']['cache_on'] = FALSE;
$db['default']['cachedir'] = "";
$db['default']['char_set'] = 'utf8';
$db['default']['dbcollat'] = 'utf8_general_ci';
$db['default']['swap_pre'] = "";
$db['default']['autoinit'] = TRUE;
$db['default']['stricton'] = FALSE;
```

9.2.2 User Authentication

```
<?php
class auth extends CI_Model {
    function __construct()
```

```

{
    // Call the Model constructor
    parent::__construct();
}
// login function
function authenticate($username,$password)
{
    $query = $this->db->query("select * from users where username=
".$username." and password =".md5($password).""");
    return $query->row_array();
}
function activity($userid,$activity)
{
    $sql="insert into activity_log(emp_id,activity)
values(".$userid.", ".$activity.)";
    $this->db->query($sql);
}
function
updateuserdetails($userid,$firstname,$lastname,$username,$password,$profil
epic){
    $sql="update users set
fname=".$firstname.",lname=".$lastname.",username=".$username.",passw
ord=".md5($password).",profile_pic=".$profilepic." where
users_id=".$userid.""";
    $this->db->query($sql);
}
}

```

9.2.3 PDF Library

```
<?php if (!defined('BASEPATH')) exit('No direct script access allowed');
class pdf {
    function pdf()
    {
        $CI = & get_instance();
        log_message('Debug', 'mPDF class is loaded.');
```

```
    }
```

```
    function load($param=NULL)
```

```
    {
```

```
        include_once APPPATH.'third_party/mpdf/mpdf.php';
```

```
        if ($params == NULL)
```

```
        {
```

```
            $param = "'en-GB-x','A4','','',10,10,10,10,6,3';
```

```
        }
```

```
        return new mPDF($param);
```

```
    }
```

```
}
```

9.2.4 Leave application

```
function leaveapplication($employeeid,$type,$message,$filename){
```

```
$this->db->query("INSERT INTO emp_db.leaves(
```

```
`leave_id` ,
```

```
`employee_id` ,
```

```

`submission_date` ,
`leave_type` ,
`leave_start` ,
`leave_end` ,
`application_message` ,
`reply_message` ,
`leave_status` ,
`attachment`
)
VALUES (
NULL , "".$employeeid."" , "".$date('Y/m/d')."" , '$type' , NULL , NULL ,
"".$message."" , NULL , 'pending' , "".$filename.""
)" );

```

```

    $sql3="select DepartmentID from employee where
EmployeeID="".$employeeid.""";
    $result=$this->db->query($sql3);
    $res=$result->row_array();
    if($this->session->userdata('users_role')!='hod'){
    $this->db->query("insert into notifications
(department_id,emp_id,notes,noticeto)values("".$res['DepartmentID']."" , "".$empl
oyeeid."" , 'Leave application by employee of employee id:', 'hod') ");

    }
}
function leaveinfo($empid){
    $query = $this->db->query("SELECT
leave_status,submission_date,response_date,leave_start,leave_end,reply_mess
age,attachment,(leave_end-leave_start) as accepted_days,(date(Now())-
leave_start) as days_accrued FROM `leaves` WHERE
employee_id="".$empid.""");

```

```
return $query->result_array();  
}
```

9.2.5 Create new project and add member

```
function  
newproject($projecttitle,$startdate,$enddate,$projectnotes,$createdby){  
  
$this->db->query("insert into  
projects(project_title,project_startdate,project_enddate,project_notes,createdb  
y)  
values('".$projecttitle."','".$startdate."','".$enddate."','".$projectnotes."','".$cre  
atedby."')");  
}  
function newprojectmember($projectid,$empid,$position){  
  
$this->db->query("insert into  
project_members(project_id,employee_id,position)values('".$projectid."','".$sem  
pid."','".$position."')");  
$sql3="select * from employee where EmployeeID='".$empid.'";  
$result=$this->db->query($sql3);  
$res=$result->row_array();  
$this->db->query("insert into notifications  
(department_id,emp_id,notes,noticeto)values('".$res['DepartmentID']."','".$res['  
EmployeeID']."','Your hae been added to a new Project:', 'emp') ");
```

9.3 Appendix C – Screen shots of developed system

9.3.1 Employee interface

The screenshot displays the 'Employee Management System' interface. At the top, there is a navigation bar with 'MY WORK', 'MY ACCOUNT', and 'LEAVE' options. Below this, a user profile section shows the name 'Employee mweembasimaanya@yahoo.com' and notification counts for 'Tasks(0)', 'Notifications(3)', and 'Settings'. The main content area is titled 'Calendar - showing this month events and Tasks' and features a calendar for August 2014. The calendar grid shows days from Sunday to Saturday, with the 5th of August highlighted in yellow. A sidebar on the left is labeled 'Tasks and Events'.

9.3.2 Admin interface

The screenshot shows the 'ADMIN PANEL' interface. On the left, a sidebar contains 'STATISTICS', 'USERS', and 'ACTIVITY LOG' options. The main dashboard area features several key performance indicators (KPIs) in purple boxes: 'Total Users 33', 'Admins 6', 'Hr 1', 'Head Of Departments 1', and 'Employees 25'. Below these, there are two main sections: 'Daily Visits' and 'Top Users'. The 'Daily Visits' section displays a donut chart with the text '8 employees'. The 'Top Users' section lists six users with their profile pictures and names: 'admin Musafi Chilembo', 'user mweemba simaanya', 'user mweemba simaanya', 'user mweemba simaanya', 'user mweemba simaanya', and 'admin Musafi Chilembo'.

9.3.3 Leave management

LEAVES

Show 10 entries Search all columns:

Sub_date	Status	Message	BeginDate	EndDate	Respdate	type	attachment	Emp	RESPOND
2014-08-03	pending	Popover on right				sick	1407092786_employee_25.rar	View	Respond
2014-08-04	accepted	Popover on right	2014-08-01	2014-10-10	2014-08-04	Vacation	1407176145_employee_33.rar	View	Respond
2014-08-04	pending	Popover on right				maternity	1407176304_employee_37.rar	View	Respond

Showing 1 to 3 of 3 entries [Previous](#) [Next](#)

9.3.4 Edit profile

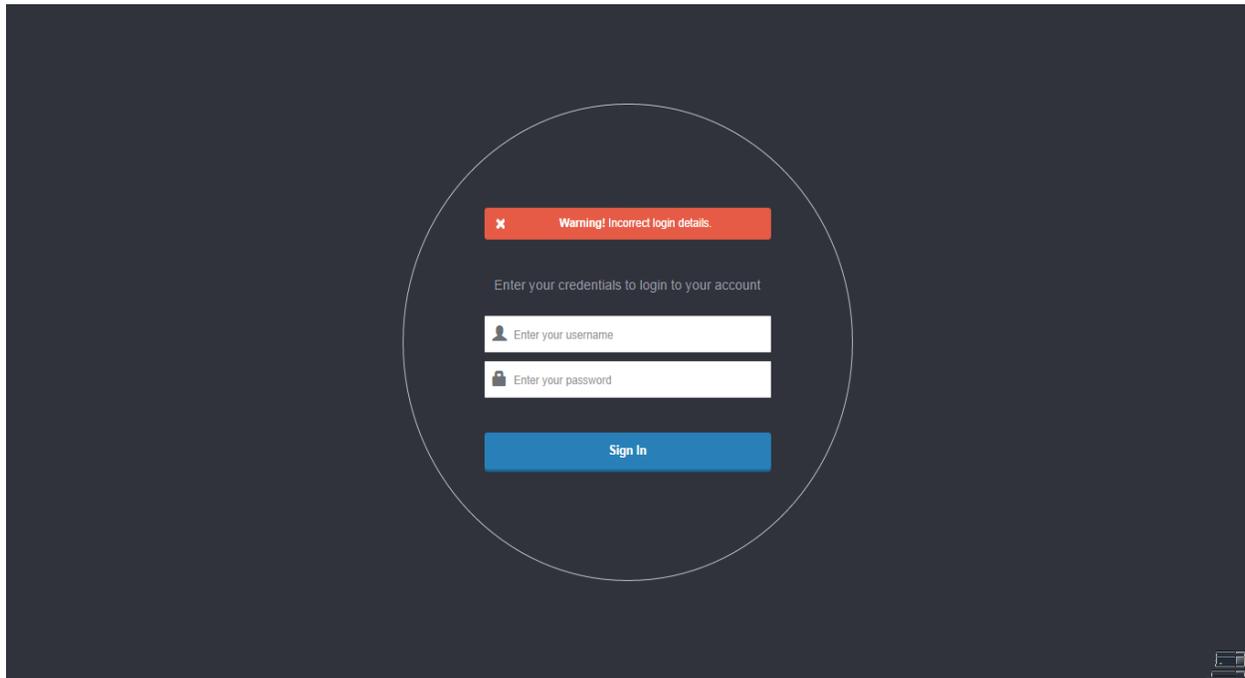
Edit My Profile details

Description	Input Field
Languages	
Language	Bemba, Tonga, Luvale, Lozi
Speaking	fluent
Written	good
Native	Tonga
Tech Skill	
Skills	Java, C++, C#, Python, ASP.net, MyS
Notes	I did Programming language paradigms so it is easy for me to learn a new language within a short period of time.

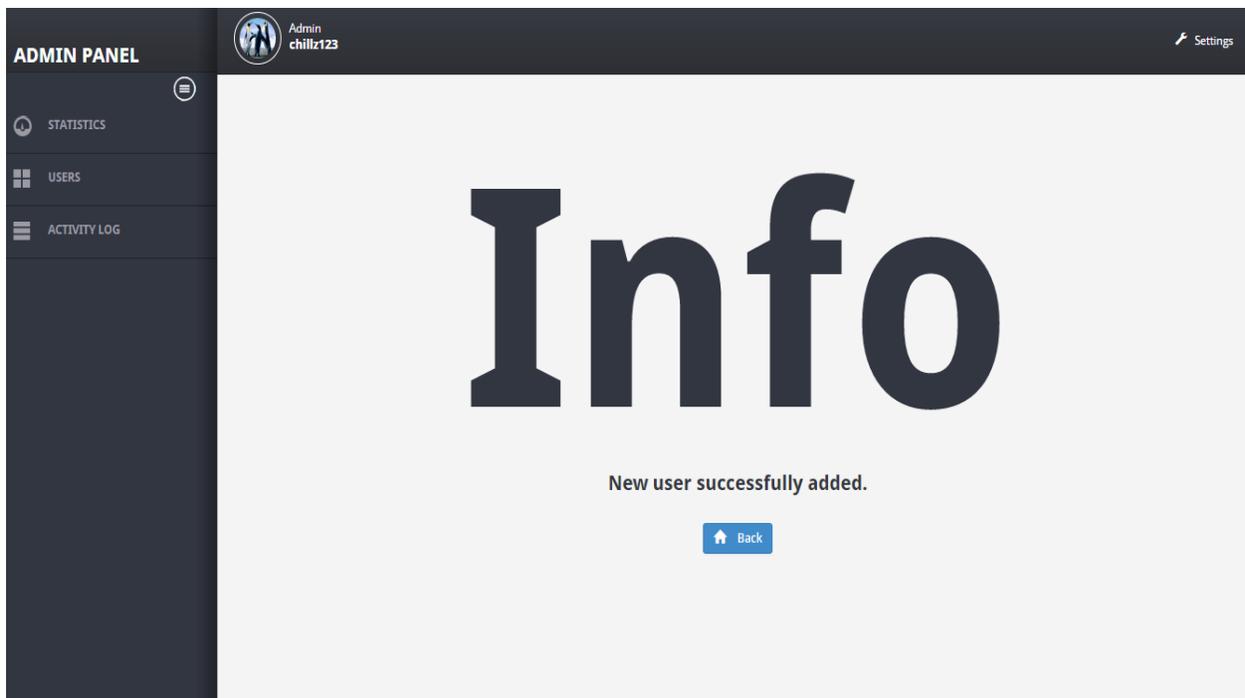
[Close](#) [Save changes](#)

9.4 Appendix I – Testing

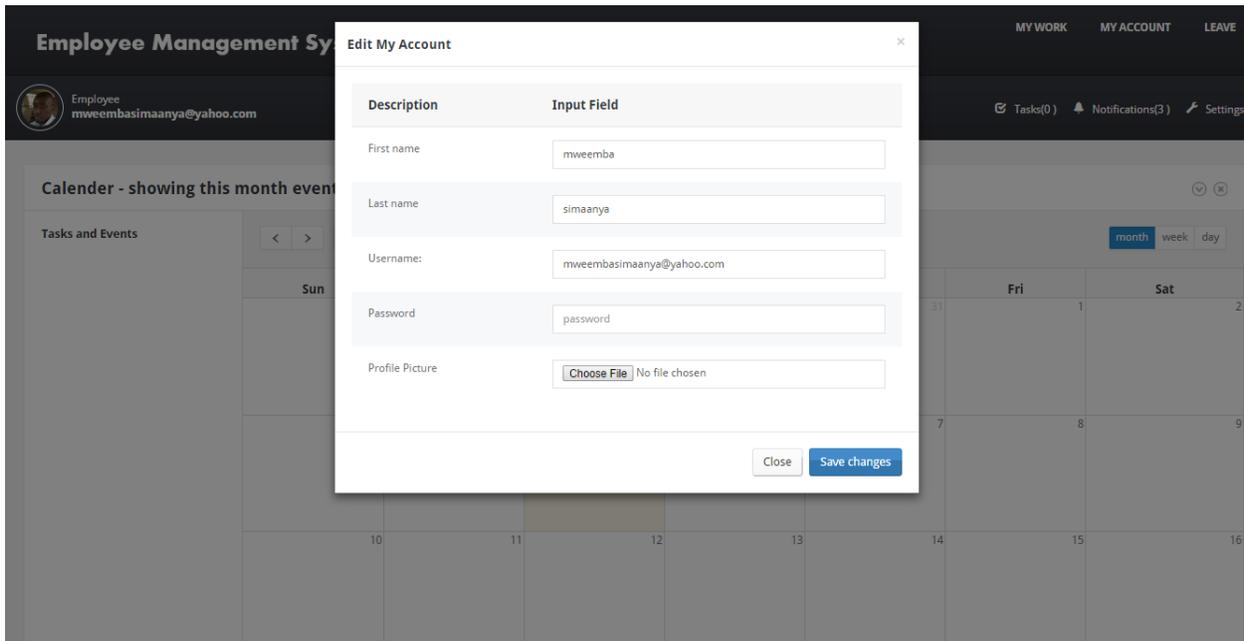
9.4.1 Login



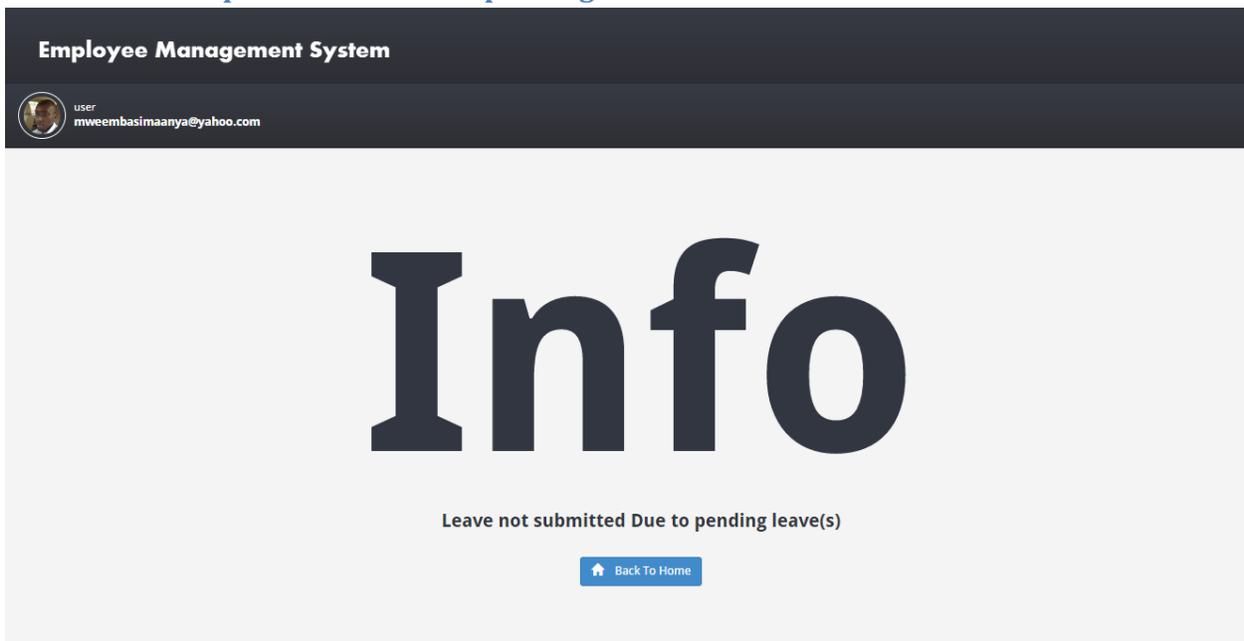
9.4.2 Add new user



9.4.3 Edit Account



9.4.4 Leave request denied due to pending leave



9.4.5 Leave request successfully submitted

Employee Management System

Computer Studies
mweembasimaanya@yahoo.com

Info

Leave has been Successfully Submitted.

[Back To Home](#)

9.4.6 Create Project WBS

Project Work BreakDown Structure

Component	Task																
programmer	<table><thead><tr><th>Description</th><th>Date Assigned</th><th>Date Due</th><th>Date Submitted</th><th>AssignedTo</th><th>Task report</th><th>Completed</th><th>action</th></tr></thead><tbody><tr><td>program the system</td><td>2014-08-04</td><td>2014-07-10</td><td>2014-08-04</td><td>mweemba_simaanya-33</td><td>I have finished my task.</td><td>completed</td><td>delete</td></tr></tbody></table>	Description	Date Assigned	Date Due	Date Submitted	AssignedTo	Task report	Completed	action	program the system	2014-08-04	2014-07-10	2014-08-04	mweemba_simaanya-33	I have finished my task.	completed	delete
Description	Date Assigned	Date Due	Date Submitted	AssignedTo	Task report	Completed	action										
program the system	2014-08-04	2014-07-10	2014-08-04	mweemba_simaanya-33	I have finished my task.	completed	delete										
	<table><thead><tr><th>Description</th><th>Date Assigned</th><th>Date Due</th><th>Date Submitted</th><th>AssignedTo</th><th>Task report</th><th>Completed</th><th>action</th></tr></thead><tbody><tr><td>designer</td><td>2014-08-04</td><td>2014-08-05</td><td>2014-08-04</td><td>gilbert_chintu-34</td><td>I have designed the system.</td><td>completed</td><td>delete</td></tr></tbody></table>	Description	Date Assigned	Date Due	Date Submitted	AssignedTo	Task report	Completed	action	designer	2014-08-04	2014-08-05	2014-08-04	gilbert_chintu-34	I have designed the system.	completed	delete
Description	Date Assigned	Date Due	Date Submitted	AssignedTo	Task report	Completed	action										
designer	2014-08-04	2014-08-05	2014-08-04	gilbert_chintu-34	I have designed the system.	completed	delete										

[Close](#) [New Component](#)

9.4.7 Generate reports



My Profile

Basic Info

First name: mweemba
Last name: simaanya
Email Address: mweembasimaanya@yahoo.com
phone: 973709777
Department: Computer Studies
NRC: 188725/10/1
Drivers licence: 1234567
Blood Type:
Gender: m
Date of Birth: 1992-10-21
Contract Type: full-time
Nationality: Zambian
Marital Status:

Emergency Contact

Firstname: mwenzi
Lastname: mwenzi
Relation: sister
phone #: 966

Education

Qualification:: MSc of Computer Science
Institution:: UNZA
StillStudying: Completed
Notes: Designed and Implemented a Web based Employee Management System to streamline the management of employees.
Thesis: Employee Management System

Languages