

MAKERERE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATION SCIENCES

EMERGING WEB TECHNOLOGIES

EMPLOYEE RECORD MANAGEMENT SYSTEM

Software Design Document

Group E2

|  |  |  |
| --- | --- | --- |
| NAME | REGISTRATION NUMBER | STUDENT UMBER |
| MASEMBE SENDI JOSEPH | 21/U/17978/EVE | 2100717978 |
| OCHIENG DEVOTE BONIFACE | 21/U/06323/EVE | 2100706323 |
| APARO CECILIA | 21/U/05605/EVE | 2100705605 |
| MUKISA VANIAH CHRISTIAN | 21/U/13176/EVE | 2100713176 |
| KABENGE REAGAN | 21/U/11603/EVE | 2100711603 |

Contents

[INTRODUCTION 2](#_Toc134803363)

[Purpose: 2](#_Toc134803364)

[Scope: 2](#_Toc134803365)

[Goals: 2](#_Toc134803366)

[SYSTEM OVERVIEW 2](#_Toc134803367)

[Functionality 2](#_Toc134803368)

[SYSTEM ARCHITECTURE 3](#_Toc134803369)

[Architectural design: 3](#_Toc134803370)

[DECOMPOSITION DESCRIPTION 4](#_Toc134803371)

[Use case Design 4](#_Toc134803372)

[Design Rationale 5](#_Toc134803373)

[DATA DESIGN 5](#_Toc134803374)

[Data storage items description 5](#_Toc134803375)

[Data processing and organization 5](#_Toc134803376)

[HUMAN INTERFACE DESIGN 6](#_Toc134803377)

[Screen Images 6](#_Toc134803378)

[1Home Screen 6](#_Toc134803379)

[2Admin Dashboard 7](#_Toc134803380)

[3Employee Dashboard 7](#_Toc134803381)

# INTRODUCTION

## Purpose:

This document is intended to talk more about our employee management system that will be used by companies to keep track of their employees and employee processes.

## Scope:

This project scopes employee management functionalities such as adding employee details, management of employee leaves and others.

The functionality of the system revolves around interactions between company owners and their subjects for efficient business processes.

## Goals:

* + - To enable company employees to create accounts and add their details.
    - To allow employees request for leaves.
    - To enable admins to approve or deny leaves.
    - To enable admins add more admins to the system.
    - To enable admins delete employees.
    - To enable admins edit employee details.

The system will be vastly deployed by companies to keep track of their employees, manage employee processes such as handling leaves and many other functionalities.

# SYSTEM OVERVIEW

This system was given as a class coursework assignment in which we were required to develop an employee Record management system.

The system was designed in laravel framework whose base language is PHP that is well known for backend and server-side scripting.

## 

## Functionality

New users through a web interface register for new accounts where they choose a user name and a password. Upon doing so, they are authenticated and are then able to access their employee information such as their teams and salary.

In the system, a normal user is able to;

* Apply for leaves.
* Change password set upon registration.
* Access salary information.
* Access team information.
* View profile Details.
* Delete leaves.

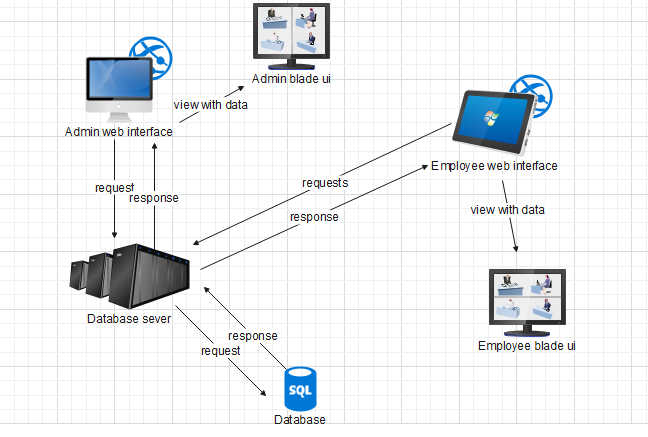
The other system users are the admins who handle normal user queries such as approving normal user, leaves and these users are more privileged than the normal users.

An admin user (employer) also registers for an account and selects a user name and a password or is made so by an already existing admin and while in the system he is able to;

* + - * + Make normal users (employees) admins.
        + Grant or deny leaves of normal users.
        + Add new employees to the system.
        + Edit employee information.
        + Change registration information such as the password.
        + View employees details such as salary and teams they belong to.

# SYSTEM ARCHITECTURE

## Architectural design:



# DECOMPOSITION DESCRIPTION

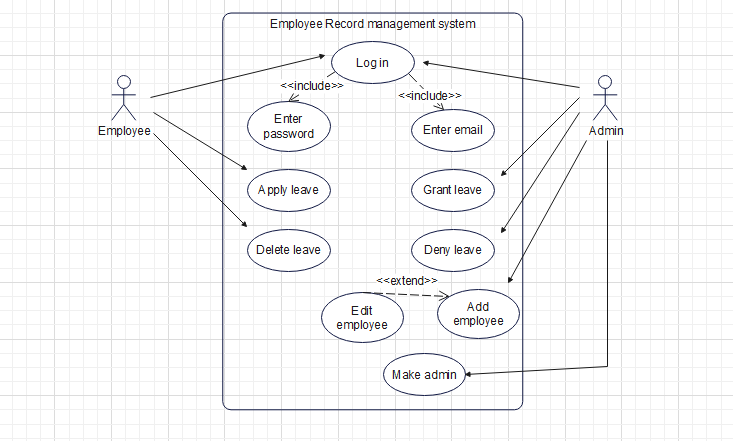
Both the admin and employees through the web interface register for new accounts and an admin user is separated from a normal user in the table base using the role column where the Admin has a role of “1” and normal users have a role of “0”.

Several requests keep on happening between the web server and the web interfaces which then go on and requests for queried data from the database which is passed on to the laravel blades through there route controllers. Let’s consider this scenario; where an employee is applying for a leave.

An employee can apply for a leave, this leave data is stored in the database initially with a pending status, when an admin logs into his account, when he loads the leaves page, a query is sent to the server through the page controller. The server also requests for the wanted data from the database and passes it on to the wanting view i.e. the view that has requested for the data.

The admin is now able to view the leaves both pending and approved, he can approve or deny leaves and also delete a leave.

## Use case Design



## Design Rationale

We selected such architecture because it makes it easy to pick data from the database, insert data and later on edit the data through simply making SQL queries.

The architecture is also laid in hierarchical order which makes it easy to track data flow and makes system traceability easy.

# DATA DESIGN

To create a functional employee management system, the information domain of employee management needed to be transformed into data structures through some number of steps.

Firstly, we identified the key information entities and relationships in the employee management domain, Such as; employees, Administrator, teams.

The attributes or properties that are associated with each entity. For example, attributes of an employee entity like name, salary and employment start date. This might involve using object-oriented programming techniques to create classes or entities, and defining their attributes and relationships. This helps to identify the key attributes like the employee’s name and the foreign keys in the different tables in the database that will be created.

The third step is to determine the relationships between the entities. For example, an employee is paid a salary and belongs to a team, a team has multiple employees. The operations that are performed on the data are included in the data structures. E.g. adding, modifying, and deleting employee records, assigning teams and approving leaves.

Implement the data to a database by creating tables for migrating the data to the corresponding entities in the data. Each table having columns that correspond to the attributes of the entity.

## Data storage items description

For this employee management system, the major data entities include Administrators, teams and employees. These entities are typically stored, processed, and organized using the following methods and technologies:

For the Data storage, we use a Relational database i.e., MySQL for inserting the data in the tables the database that was created to hold the records for the system. The users table in the database has an extra column ‘role’ that is used to check if a user is an admin or not (admin has a value of 1 and normal user ha value of 0) thus the information is stored by first checking the status of the current user for some methods like adding an employee and approving a leave are restricted to the admins only.

## Data processing and organization

- Laravel Framework and libraries were implemented for employee management system development using the PHP programming language used to create the links that interacted with the employee data in the database for the streamlined data processing and organization

-For Security and access control measures, some views and operations are restricted to the admin who is assigned a value of 1 in the role column in the table of the users in the Employee management database. More measures like password protection are implemented in the database by the PHP e.g. password encryption of the user passwords

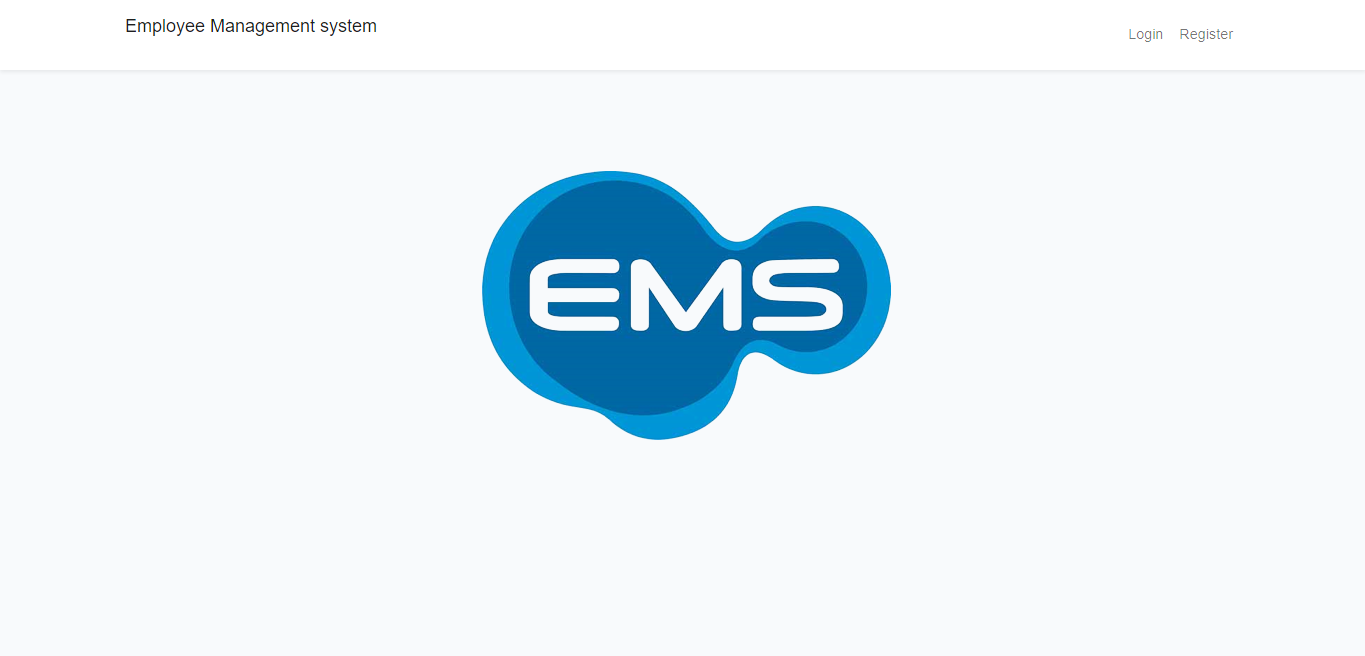
In summary, an employee management system typically uses a relational database for storing employee data. Data is processed using PHP language.

# HUMAN INTERFACE DESIGN

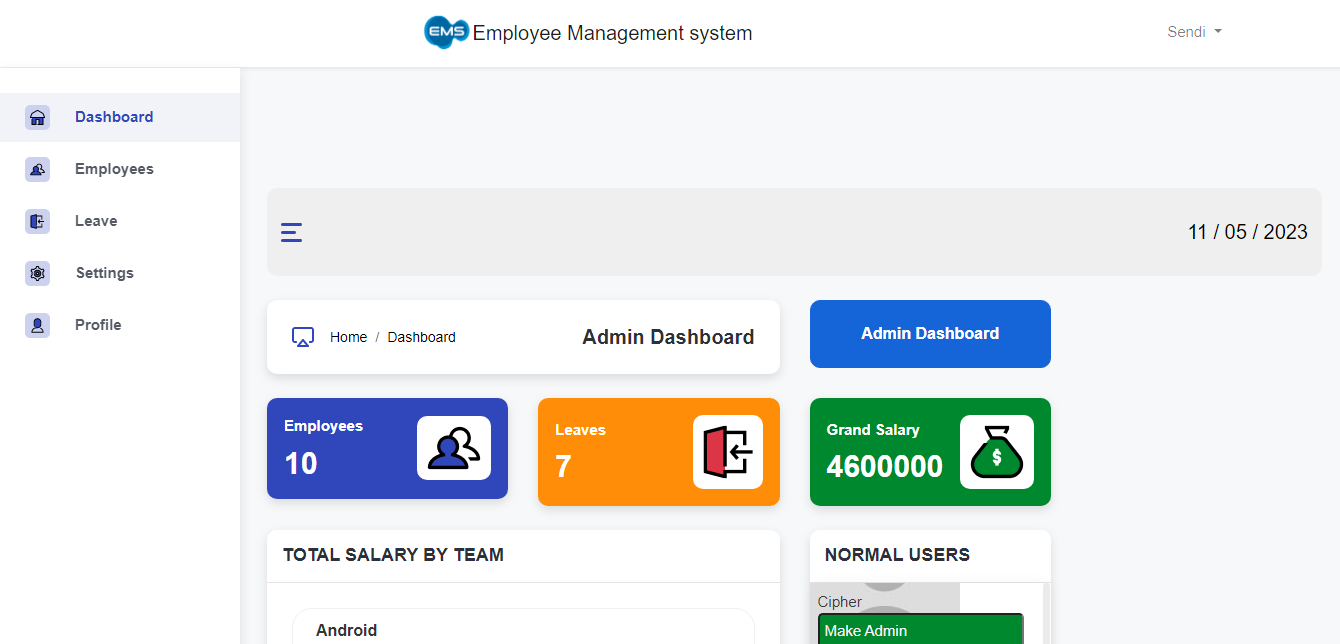
* Users are categorized as either admin or normal users and both access the system through a web interface.
* Web interfaces were designed using laravel blade files to create user friendly UI such as login panels, Forms and others.
* Users mainly interact with the system through these interfaces and make queries that are handled by the back-end that technical and accessed by programmers.

## Screen Images

### Home Screen



### Admin Dashboard



### Employee Dashboard

