



**Employee Management System  
For  
Rising Logistics (Pvt) Ltd**

P.K. Yauwana Ravindra

BIT Registration Number: R081931

Index Number: 0819311

**Supervisor: Mrs. Fincyka Fernando**

December 2017

**BIT**




**This dissertation is submitted in partial fulfillment of the requirement of the  
Degree of Bachelor of Information Technology of the University of Colombo  
School of Computing**

**Declaration**

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

Signature of Candidate:  ..... Date: 09/11/2017  
Name of Candidate: .. Yauwana Ravindra ..

Countersigned by:  
Signature of Supervisor(s)/Advisor(s):  ..... Date: 11/11/2017  
Name(s) of Supervisor(s)/Advisor(s): .. K.S.F.K. Fernando ..

# Abstract

Rising Logistics is a growing Spare Parts provider in the Sri Lankan market. With the ongoing expansion Rising Logistics cannot carry out their Employee's leaves and Employee details in physical papers anymore and they want a process to manage those day to day issues.

In order to overcome those problems, they encounter daily, essential need of an Employee Management System developed by aiding the Information Technology. This Employee Management System will save their time and increase the efficiency of their work.

The Employee Management System solution will cover an area of recruiting employees, leave calculation, KPI (Key Performance Indicator) management, employee review management, add candidates, create users, viewing user logs and other similar functions. This will overcome the customer's current struggle to manage employee information and leave management.

The proposed solution was a Web-Based application and hosted locally in the Rising Logistics – Head office. The system developed within the Object-Oriented Paradigm and the requirements analysis and design aspects of the system took the advantage of the Unified Modelling Language Behavioral Diagrams. The system developed using Open Source technologies such as PHP 5.4, jQuery Framework, cascade styling sheets and MySQL 5.0 as the database development for the system. All the above-mentioned tools were operating system independent, therefore it was a big advantage since these tools are working in Windows well as Linux systems. The Employee management system created according to the user requirements of Rising Logistics management, likewise Rising Logistics achieved their targets in future with the help of this system.

# Acknowledgement

I would like to thank late Prof. V.K. Samaranayake for initiating BIT degree Programme and the contribution that he has provided to computing of Sri Lanka.

Next, I would like to express my gratitude to my project supervisor Mrs. Fincyka Fernando

Also, my sincere thank goes to Dr. Viraj Pinto Jayawardena and the staff at Matrix Institute of Information Technology for the support given during the past years and all the people that gave me their support to complete my project especially Mr. Hasitha Heshan Kapuduwa.

I appreciate the support given by the University of Colombo School of Computing (UCSC) for providing the resources and facilities to produce this project.

# Table of Contents

<b>Abstract.....</b>	<b>iv</b>
<b>Acknowledgement .....</b>	<b>v</b>
<b>List of Figures.....</b>	<b>ix</b>
<b>List of Tables .....</b>	<b>xi</b>
<b>List of Acronyms .....</b>	<b>xii</b>
<b>Chapter 1 Introduction.....</b>	<b>1</b>
1.1 Introduction.....	1
1.2 Motivation of the Project .....	1
1.3 Objectives of the Project .....	2
1.4 Scope of the Project .....	2
1.6 Structure of the Dissertation.....	3
<b>Chapter 2 Analysis .....</b>	<b>4</b>
2.1 Fact Gathering Techniques .....	4
2.2 Description of the current system .....	5
2.3 Drawbacks of the existing system.....	6
2.4 Existing similar systems.....	7
2.4.1 ADP Total Source [1].....	7
2.4.2 A Complete HRIS – OrangeHRM Product Features [2].....	7
2.5 Functional Requirements .....	8
2.6 Non-Functional Requirements .....	9
<b>Chapter 3 Design of Solution .....</b>	<b>11</b>
3.1 Introduction.....	11
3.2 Software Development Process Model .....	11
3.2.1 Waterfall process model.....	11
3.2.2 Rapid Application Development.....	13
3.2.3 Rational Unified Process.....	14
3.3 Justification of the selected approach .....	15
3.4 Alternate Solutions.....	15
3.4.1 Standalone System .....	15
3.4.2 Collection of Software .....	15
3.4.3 Web-Based system.....	16
3.4.4 Reasons to select Web-based system .....	16
3.5 High level architecture of the system.....	16
3.6 High level usecase of the system .....	17
3.6 Module Design.....	18
3.6.1 Candidate Registration Module.....	19

3.6.2 Review system Module .....	21
3.6.3 KPI system module .....	22
3.6.4 Leave Module .....	23
3.6.5 Employee Registration Module.....	25
3.6.6 User Registration module.....	27
3.6.7 Generate Reports.....	28
3.7 Sequence diagrams of the system .....	29
3.7.1 Sequence diagram for the login to the system.....	29
3.7.2 User login sequence diagram .....	29
3.7.3 Add review sequence diagram .....	30
3.8 Activity diagram for the system.....	30
3.8.1 Activity Diagram for the user login .....	31
3.9 Class Diagram .....	32
3.10 Database Design.....	33
3.11 User Interface Design.....	34
3.11.1 Login .....	35
3.11.2 Main Interface .....	35
3.11.3 Add Candidate form.....	36
<b>Chapter 4 Implementation .....</b>	<b>38</b>
4.1 Implementation Environment Development Tools .....	38
4.1.1 Hardware Requirement .....	38
4.1.2 Software Requirement.....	38
4.1.3 Development Tools .....	38
4.2 Architectural Overview.....	39
4.3 Network Diagram of the Implementation Environment.....	40
4.3 Major Code Structure.....	40
4.3.1 Model .....	40
4.3.2 Controller .....	42
4.3.3 View .....	44
4.3.4 Other code segments .....	46
4.4 Reused Modules.....	47
4.5 Security .....	47
<b>Chapter 5 Evolution.....</b>	<b>48</b>
5.1 Testing Procedure .....	48
5.2 Test Plan and Result.....	49
5.2 User Evaluation.....	52
5.2.1 User Feedback From .....	52

5.2.2 Result of user feedback form .....	53
<b>Chapter 6 Conclusion .....</b>	<b>55</b>
6.1 Overview .....	55
6.2 Problems Encountered .....	55
6.3 Lessons Learnt .....	56
6.3 Future Work .....	56
<b>References .....</b>	<b>57</b>
<b>Appendix A System Documentation.....</b>	<b>59</b>
System Documentation .....	59
How to Implement the System in Local Host .....	59
How to Implement the System on a Web Server .....	60
<b>Appendix B Design Documentation.....</b>	<b>61</b>
Sequence Diagram of Add an Employee .....	61
Usecase Diagrams with Detailed Information .....	62
<b>Appendix C User Documentation.....</b>	<b>69</b>
Introduction .....	69
Candidate Module .....	69
Performance Module.....	72
Leave Module .....	74
PIM (personal information manager).....	76
<b>Appendix D Management Reports .....</b>	<b>77</b>
Applied Vacancies according to the Job title .....	77
Leave Applied to Employees per month.....	78
Best 10 KPI marks per year .....	79
New Leaves to Approve.....	80
<b>Appendix E Test Results.....</b>	<b>81</b>
<b>Appendix F Code Listing.....</b>	<b>84</b>
Adding vacancy Details .....	84
Add Employee Function .....	85
Add KPI Records Function .....	86
Suggestion Box Function .....	86
Vacancy Manager Class.....	87
Ajax Delete Function .....	88
<b>Appendix G Client Certificate .....</b>	<b>90</b>
<b>Glossary .....</b>	<b>91</b>

# List of Figures

Figure 2.1 Usecase Diagram of the current System.....	6
Figure 3.1 Waterfall Process Model Diagram.....	12
Figure 3.2 Rapid Application Development Diagram .....	13
Figure 3.3 Rational Unified Process Model Diagram.....	14
Figure 3.4 High level Architecture of the Proposed System.....	17
Figure 3.5 High level Usecase Diagram of the Proposed System.....	18
Figure 3.6 Usecase Diagram for Candidate Registration.....	19
Figure 3.7 Usecase Diagram of Review System.....	21
Figure 3.8 Usecase Diagram of KPI System.....	22
Figure 3.9 Usecase Diagram of Leave Module.....	23
Figure 3.10 usecase Diagram of Employee Registration Module.....	25
Figure 3.11 Usecase Diagram of User Registration Module .....	27
Figure 3.12 Usecase Diagram of Generate Reports Module.....	28
Figure 3.13 Sequence Diagram for the Login to the System .....	29
Figure 3.14 User login Sequence Diagram .....	29
Figure 3.15 Add Review Sequence Diagram.....	30
Figure 3.16 Activity Diagram for the User Login.....	31
Figure 3.17 Overall Class Diagram of the Proposed System.....	32
Figure 3.18 Data Diagram.....	33
Figure 3.19 Login Interface of the System.....	35
Figure 3.20 Main Interface of the System.....	36
Figure 3.21 Add Candidate Form .....	37
Figure 4.1 MVC Architecture .....	39
Figure 4.2 Directory Structure .....	39
Figure 4.3 Network Diagram of the Implementation Environment .....	40
Figure 5.1 Check Title Combo Box .....	49
Figure 5.2 Checking Other Fields .....	50
Figure 5.3 Checking an existing Username .....	50
Figure 5.4 Email field with Main Validation.....	51
Figure 5.5 Delete confirmation box .....	51
Figure 5.6 User Feedback Form.....	52
Figure 5.7 Question Number 3 Result.....	53
Figure 5.8 Question Number 4 Result.....	54
Figure A.1 FileZilla FTP Client Interface.....	60
Figure B.1 Sequence Diagram for Add an Employee.....	61
Figure C.1 Vacancies Main Interface.....	70
Figure C.2 Job Description Panel .....	70
Figure C.3 Candidate Detail Panel.....	71
Figure C.4 Candidates Page .....	72
Figure C.5 Short Listed Candidates .....	72
Figure C.6 KPI Setup Interface.....	73



Figure C.7 Add Review Interface .....	73
Figure C.8 Add Leave Form .....	74
Figure C.9 Leave Entitlement Form.....	75
Figure C.10 Leave Summary Table .....	75
Figure C.11 Employee List Table .....	76
Figure C.12 Interface for Adding Employees .....	76
Figure D.1 Applied Candidates per Year .....	77
Figure D.2 Filtering Leaves for a Month .....	78
Figure D.3 List of Leave Filtered by the Month .....	78
Figure D.4 Best 10 KPI Marks per Year.....	79
Figure D.5 Leaves Pending Approval.....	80
Figure G.1 Client Certification .....	90

# List of Tables

Table 3.1 Candidate Registration Usecase.....	20
Table 3.2 Leave Module Usecase .....	24
Table 3.3 Employee Registration Usecase.....	26
Table 5.1 Testcase for checking the Job Combo box.....	49
Table 5.2 Testcase for Verifying other Fields.....	50
Table 5.3 Testcase for checking existing usernames .....	50
Table 5.4 Testcase to validate Email field .....	51
Table 5.5 Testcase to validate delete confirmation.....	51
Table 5.6 Answer to the question No 1.....	53
Table 5.7 Answer to the question No 2.....	53
Table 5.8 Answer to the question No 4.....	53
Table 5.9 Answer for the Question No 5 .....	54
Table B.1 Employee Update Usecase .....	62
Table B.2 Add Leave Usecase .....	63
Table B.3 Login Usecase .....	64
Table B.4 Get Approval Usecase.....	65
Table B.5 Add Designation Usecase.....	66
Table B.6 Set Leaves Usecase .....	67
Table B.7 Usecase for Deleting a System User .....	68
Table E.1 Test Results .....	83

# List of Acronyms

**AJAX**– Asynchronous JavaScript and XML

**CS** – Creative Suite

**DBMS**- Database Management System

**DVD** – Digital Video Disk

**ERD** – Entity Relationship Diagram

**FTP** – File Transfer Protocol

**GB** – Gigabyte

**GHz** – GigaHertz

**MB** – MegaByte

**MS** – Microsoft

**OOD**-Object Oriented Design

**PEO** - Professional Employer Organization

**PHP** – Hypertext Preprocessor

**SDLC** – System development life cycle

**UI** – User interface

**UML**-Unified Modeling Language

**URL**-Uniform Resource Locator

**WAMP** – Windows Apache MYSQL PHP

# Chapter 1 Introduction

This Chapter will give a brief introduction about the Rising Logistics (Pvt) Ltd, Motivation for the Project, and Scope of the project, Objectives of the system and finally the outlook of the rest of the dissertation chapters.

## 1.1 Introduction

Rising Logistics (Pvt) Ltd is a vehicle spare parts selling company and they do both retail and wholesale. They have employees who are responsible when dealing with the customers and wholesale partners. They have a good solid profit for past few months and they are recruiting more employees with the idea of expanding for new branches in Sri Lanka. They always try to maintain the trust between the customers by providing good quality spare parts at a reasonable price.

This kind of web-based information system will address the automation of the Employee Management processes and streamline them to meet certain standards and procedures. A systematic management of the Employees is a must as the organization's life depends on Employees. In addition to this, there are lots of other benefits the client can gain because of this system.

## 1.2 Motivation of the Project

Currently Rising Logistics carry their day to day work with a heavy load of paperwork. This manual system is a time wasting process of this company and most of the time the recruitment process need to redo again because of a single feedback paper of the interview. Also, leave process is currently doing via leave note in a paper. This need was taken as the root cause to fulfill the final year project requirement of Bachelor of Information Technology Degree program and also this is a valuable opportunity I got to practically apply gained knowledge and enhance skills in web application development.

### 1.3 Objectives of the Project

The main goal of this project is to develop and implement an automated web-based information system to accomplish employee management process and manage candidate details of the Rising Logistics to handle employee management in an efficient way.

- System provides an efficient way to manage new candidate details
- Acting as a coordinator to sort out the leave management issues
- Support for managing candidate details, employee leaves, Employee performances, Employee details as such
- Real-time information availability of the employees or candidates by giving the search facility
- Generate reports and let the decision maker to take effective decision within an appropriate time
- System provides 24 x 7 service with automated process

### 1.4 Scope of the Project

Rising Logistic Employee Management System is a customized Employee Management System solution tailored to the needs of Rising Logistic. The system is intended to provide a competitive advantage and excel the performance of the organization as a whole. Better Employee Management is the main focus of the system. Scope of the project described in a modularized manner.

- Candidate details management :
  - Add candidate details with or without uploading a CV, list the candidates
  - List candidates according to the status (shortlisted, considerable, rejected)
- Employee performance management
  - Add, give marks, search, delete, and edit of employee KPI records
  - Add, edit, delete, search reviews of an employee which will benefit from annual performance evaluation

- Leave management
  - Add leaves, Leave Entitlements, Leave summary of an employee
- Personal Information management
  - Add, view and print, delete, edit of employee records
  - View customize reports
- Admin process management
  - Add new designations, delete designations
  - There is a login interface for the users with their own username and password with two different (Admin, Employee) authentication.
  - Check user logins
- To cut the paperwork to make their work effectively.
- To store data in a safe proper way to secure and unauthorized access to data and in an optimal way for efficient searching.

## 1.6 Structure of the Dissertation

This Dissertation includes six main chapters will explain the effort carried out to successfully completion of the web-based information system to accomplish employee management need of the Rising Logistics (Pvt) Ltd.

Analysis chapter explains the requirement gathering techniques, details of the manual system, functional nonfunctional requirements and details of the existing systems. Design chapter explains the use case diagram of the proposed system, database design of the system and main interfaces of the system. Implementation chapter explains the hardware-software requirements, development tools which are used for the system, code features and reused existing codes of the system. Evaluation chapter explains the techniques of testing, details of software testing, high-level test plan of the system and client evaluation of the system. Conclusion chapter explains the future enhancements of the system and lesson learned of the overall project work.

# Chapter 2 Analysis

This chapter will deliberate the facts which are necessary to develop the system which was gained by distinct fact-finding techniques and also this chapter include about the description of the inspection of the documents, Current system, drawbacks of the existing system, existing similar system, functional requirements, non-functional requirements and methodology for the proposed system.

## 2.1 Fact Gathering Techniques

Facts and all relevant information related to the business environment should be collected prior to studying any system and those collected facts then transformed into design diagrams in the design phase by using object-oriented concepts. The success of any project depends on the accuracy of the requirement gathered in the analysis phase. Multiple fact-gathering techniques are available to identify system requirements clearly.

The following fact-finding techniques are used for this project among those techniques:

### **Interviews**

This is the ideal and effective method to clarify requirements without much of conflicts. At the beginning Owner or rather Chairman and the Human Resources manager of the Rising Logistics are interviewed and collected the details of the system, which they are currently doing a manual process

Even though the system requirements were not cleared initially, after conducting several interview sessions with employees, gradually realized what kind of a system they need.

Interviews conducted with management and Employees became a solution to overcome conflicts and finalize system functional and Non-functional requirements.

### **Document Review**

Since the Rising Logistics has been using a manual system, posing huge amount of physical paper documents, those existing physical documents help to get an idea about the process by referring their document.

## **Observation**

This technique is not randomly used, but it is used to clarify ambiguous user requirements and to eliminate conflicts of the user requirements. Some requirements were not cleared given to the Employee Management System. By observing their current manual system, ambiguous process and requirements were clear.

## **2.2 Description of the current system**

Rising Logistics has been doing their day to day employee management work such as filing new employee details, filing new candidate details, categorize the candidates filed, applying for a leave, check the leave balance and annual salary increment according to the performance.

An employee will fill a leave note which can be taken from the HR manager and employee will hand that form to the HR manager after completion. Then HR manager will write the leave information in the left log. Then end of the day HR manager will check if that leave can be approved or not. If the leave can be approved HR manager will mark the leave as approved and put a sign. Chairman will monitor the leave log once a month. Sometimes leave approved done by the operation manager or the Chairman.

All the candidate CVs they kept in a single location and they are all file together. Candidates were called for the interviews according to the sequence they get the CVs. Once the interview is done with the candidate they will write the status of the candidate on the CV itself. After that candidate will call for a second interview or will appoint as an employee.

They are giving every employee an annual salary increment. But there is no transparent method for that because there is no correct method to measure the increment amount each employee should get. Details of the employees are in separate files with the employee name. The following figure 2.1 represent the use case diagram of the current system.





Figure 2.1 Usecase Diagram of the current System

### 2.3 Drawbacks of the existing system

Data entry, validation, and processing are done manually. This can be erroneous. Historical data cannot be systematically viewed or structured at any time. Not more than one person can access to the data at the same time.

Difficulties are arisen due to lack of analyzed data, finding the history data and lack of summary reports.

Since the considerable large amount of data were gathered, the difficulties of finding urgent and accurate information in a timely manner to make administrative decisions.

## 2.4 Existing similar systems

### 2.4.1 ADP Total Source [1]



“ADP Total Source” delivers a sophisticated web-based HRMS, called “My Total Source”, which complements and supports the human element of our PEO solution. “My Total Source” features a suite of leading-edge online HR management applications and tools. It is simple, secure, and powerful, allows for employee self-service, and is designed to save you time, resources, and paperwork headaches.

### 2.4.2 A Complete HRIS – OrangeHRM Product Features [2]



OrangeHRM offers a flexible and easy to use HRIS solution for small and medium sized companies free of charge. By providing modules for personal information management, employee self-service, leave, time & attendance, benefits and recruitment companies are able to manage the crucial organization asset - people. The combination of these modules into one application assures the perfect platform for re-engineering and aligning your HR processes along with the organizational goals.

- Administration Module
- Personal Information Management
- Leave Module
- Time Module
- ESS Module
- Benefits Module
- Recruitment Module
- Performance Module
- Reports Module

After a thorough study of the existing similar systems able to get a clear picture of the structure of the Employee Management module. Functionalities which are the most required were chosen to be implemented in the Employee Management module.

## 2.5 Functional Requirements

Functional requirements capture the intended behavior of the system. The functional requirements of the web-based Employee management system for Rising Logistics are described as follows,

### **Registration**

- There are two types of user categories Employee and Admin.
- In order for an employee to get registered, he/she should request from the any of managers. Then managers will create an employee type user from the system and will give the username and the password to the employee.
- Only the Admin User type can create other manager user types

### **New Candidate Handling Process**

- Adding a new candidate can be done by any type of a user according to the given vacancies.
- Creating, Editing and deleting vacancies can be done by only the Admin User type.
- Also sorting the candidates, editing candidates details, delete candidates can be done by only the Admin User type.

### **KPI add, edit, delete, and give marks**

- Add, edit, delete and give marks of the KPI's can be done by only the Admin Users.
- Employees can see the KPIs and the KPI marks of his/her only.
- Employees also have the search function available as well.

### **Leave add, edit, approve, search and entitlement**

- Admin users have the privilege to edit, approve, see all the leaves and leave summary.
- Employee users can only add leaves, see his/her leave entitlement and see his/her leave balance.
- Admin user can search any user's leave, leave summary and leave entitlement.

## Reporting Module

- Generate the reports of various categories such as filtered by all data and date range.
- Only the Admin users able to see the generate reports menu.

## Admin Module

- All the administration facilities such as user creation, checking user logs, adding employees, print employee details are extracted under this module.
- This module only visible to the Admin users.

## 2.6 Non-Functional Requirements

Non-functional requirements capture the indented quality or property that the product must have in order to operate it smoothly without any problems. The non-functional requirements of the web-based information system to accomplish Employee management of the Rising Logistics as follows,

**User friendliness:** The proposed system should be easy to learn and use. It contains effective error handling, pleasant, easy to navigate through the GUI and many other UI guides for the user.

**Reliability:** The system should be able to consistently perform according to its specification with a minimum number of technical errors.

**Availability:** The system should be available 24 / 7 and be able to cope with concurrent users and operations.

**Security and Privacy:** Appropriate technical measures should take to prevent unauthorized interception of confidential personal data as well as to protect the system.

**Expandability:** The system should be designed to support future enhancement.

**Response time:** The response time of the system according to the user inputs should be fast as possible.

**Robustness:** The system should be able to cope with errors during the execution and needs to handle user input errors.

**Efficiency of the database:** Optimizing the database for efficient searching and processing.

# Chapter 3 Design of Solution

This chapter will deliberate the facts which are necessary to design the system. This chapter includes a methodology for the proposed system, alternative solutions, selected solutions, module design, database design and user interface design.

## 3.1 Introduction

After gathering requirements, analyzing it and identify the project scope, the next phase is the system design phase. Unified Modeling Language (UML) which is the standardized general-purpose modeling language in the field of object-oriented software engineering used to design the diagrams in the proposed computerized system.

Following UML modeling diagrams are used s designed diagrams in the new system.

- **Use case diagram** – shows how the user interact with the system
- **Class diagram** – Shows the static structure of the system
- **Entity Relationship Diagram (ERD)** – illustrate the relationship between entities in a database

## 3.2 Software Development Process Model

A software development process or life cycle is a structure imposed on the development of a software product. There are several models for such processes, such describing approaches to a variety of tasks or activities that take place during the process [3].

Three Popular process models are described below:

### 3.2.1 Waterfall process model

The best known and oldest process is the waterfall model, where developers follow the sequence of steps in order to develop a system. They state requirements, analyze them, design a solution approach, architect a software framework for that solution, develop code, test, deploy and maintain. After each step is finished, the process proceeds to the next step. The following figure 3.1 diagram illustrates the waterfall Process model.

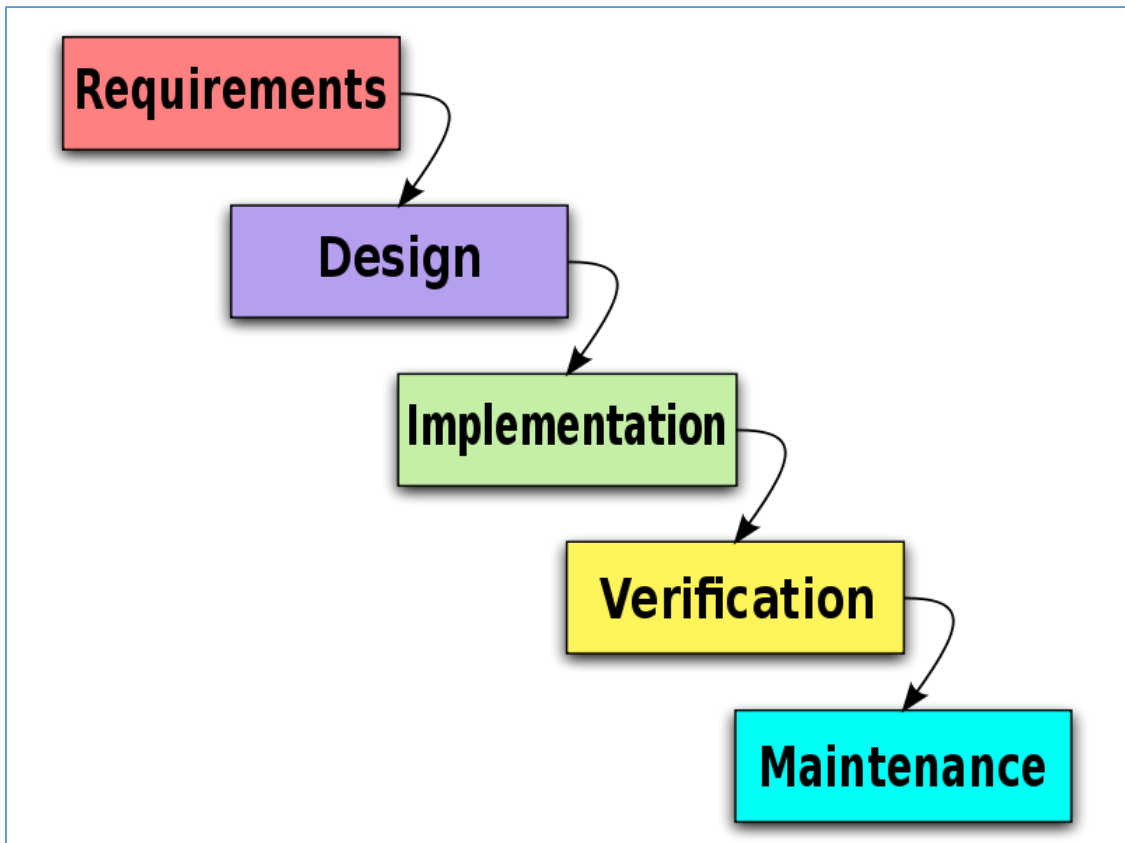


Figure 3.1 Waterfall Process Model Diagram

### **Advantages and Disadvantages of the waterfall process model**

#### Advantages

- Documentation is produced each phase

#### Disadvantages

- Inflexible partitioning of the project into distinct stages
- Commitments must be made at an early stage in the process, which makes it difficult to changing customer requirements

Waterfall process model requires complete system requirements at once, the user cannot handover the requirements of the employee management system straight away since the changes occur during the development stage. Therefore waterfall process model cannot be used for the development of this system.

### 3.2.2 Rapid Application Development

Rapid Application Development (RAD) is a software development methodology that uses minimal planning in favor of rapid prototyping. The “Planning” of software developed using RAD is interleaved with writing the software itself. [4]

The following figure 3.2 diagram illustrates the Rapid Application Development (RAD), model.

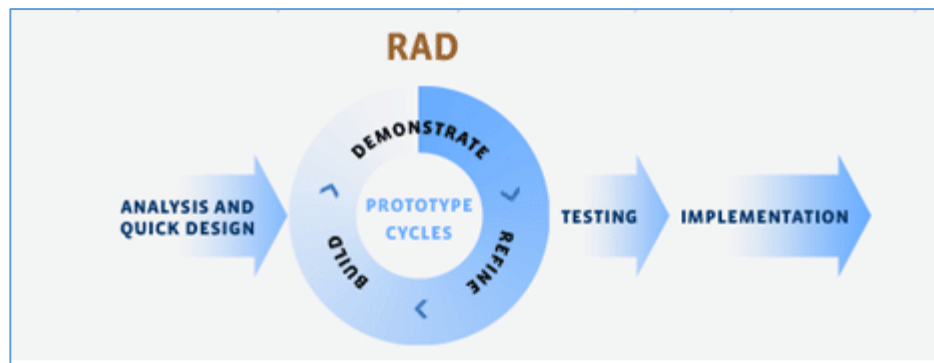


Figure 3.2 Rapid Application Development Diagram

#### **Advantages and Disadvantages of the Rapid Application Development (RAD) Model**

##### Advantages

- Adaptable to changes
- RAD uses specialized development tools to reduce the development time
- Feedback from customers at early stage of the development

##### Disadvantages

- Require special development tools to implement RAD
- All applications are not compatible with RAD
- Need participation of both customer and developer in the development process

Rapid Application Development (RAD) is mainly focused on teamwork and collaborate with each team member, since this an individual project, RAD model cannot be used for the development of this Employee Management system.



### 3.2.3 Rational Unified Process

Rational Unified Process (RUP) used as a process model to develop the proposed computerized system. Rational Unified Process is an interactive process model as well as a process model which can adapt the system development to the requirement changes.

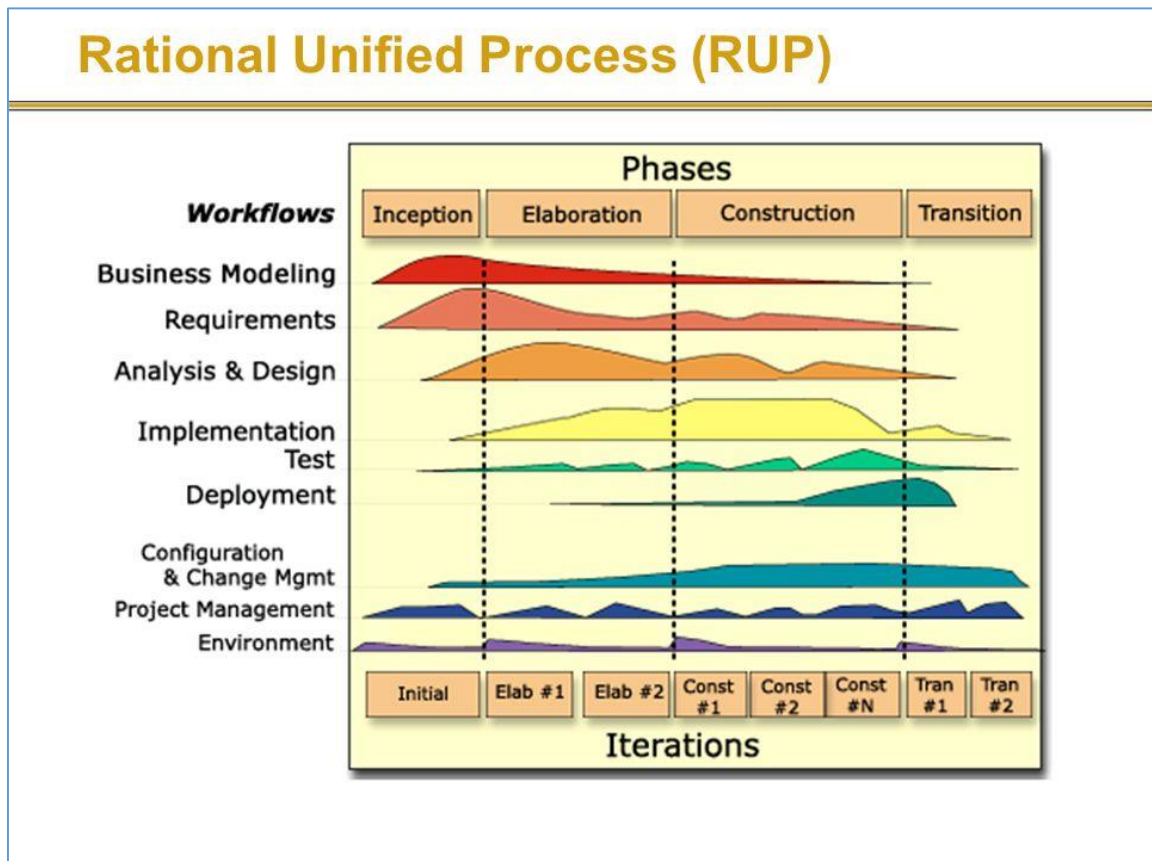


Figure 3.3 Rational Unified Process Model Diagram

There are four phases in Rational Unified Process (RUP) figure 3.3 depicts the phase of RUP. Inception, Elaboration, Construction, Transition. There is a summary detail given below under each phase,

#### **Inception**

The vision, scope and initial plan of the employee management system is established in this phase. Identify the business cases and Identify the actors interact with the system can be categorized under this inception phase. Moreover, the high-level functional and non-functional requirements were identified.

### **Elaboration**

The goals of the elaboration phase are to analyze the problem domain, establish a sound architectural foundation and eliminate the highest risks elements of the project

### **Construction**

In this phase, the primary objective is to build the software system, therefore, the main focus is given to the development of components and other features of the system

### **Transition**

The primary objective is to transit the system from development into operation. Making it available to use by the end user. This phase typically starts with a beta release of the system, which is then replaced with the enhanced version of the system.

## **3.3 Justification of the selected approach**

The client was unable to give all the accurate requirements at once about what sort of a system Rising Logistics needs. After considering all these facts and comparing with other development process models Rational Unified Process (RUP) model was selected as the best process model to develop this system.

## **3.4 Alternate Solutions**

### **3.4.1 Standalone System**

The standalone software may refer to a computer software that can work offline, and it operates independently, which is not connected to, an electric transmission and distribution network. Since this employee management system is to deal with a different people a standalone system is not fit into the problem domain.

### **3.4.2 Collection of Software**

By using a system with integrated software, extra features which are not useful also come with the final version of the system, therefore the Employee Management system becomes complex. Furthermore, when features unique to the Employee Management system integrate into to the software, lots of difficulties will arise. Therefore the

collection of the software is not the best way to implement Employee Management System.

### 3.4.3 Web-Based system

Currently, most of the systems are operated as web-based systems and help to do a job in an efficient and effective manner. When the web-based system is operating in a web-server which is a centralized location, any individual in the world can refer the particular system without installing it on their own computer. Web-based system access through the client browser, therefore web-based systems are operating systems and hardware independent. When a change has been made to the standalone system, it has to be done to the each and every computer, when the server is updated in the web-based system the update will be automatically applied to all client commuturs at once.

### 3.4.4 Reasons to select Web-based system

- Easy install and maintain: Once a new version or upgrade installed on the host server, all users can access it straight away. There is no need to upgrade each client PC.
- Cost effective development: Client will pay only for their requirements.
- Accessible anywhere and anytime: Unlike other software/application, can access anytime/anywhere via PC. The only requirement is connecting to the internet. This allows to the client to do their day to day work.
- MVC architecture can be use

## 3.5 High level architecture of the system

According to the proposed system, users will interact with the system and they will do the CRUD (create, read, update, delete) functions and those changes will save in the centralized DB (client-server). Diagram in figure 3.4 depicts the high-level architecture of the proposed system.

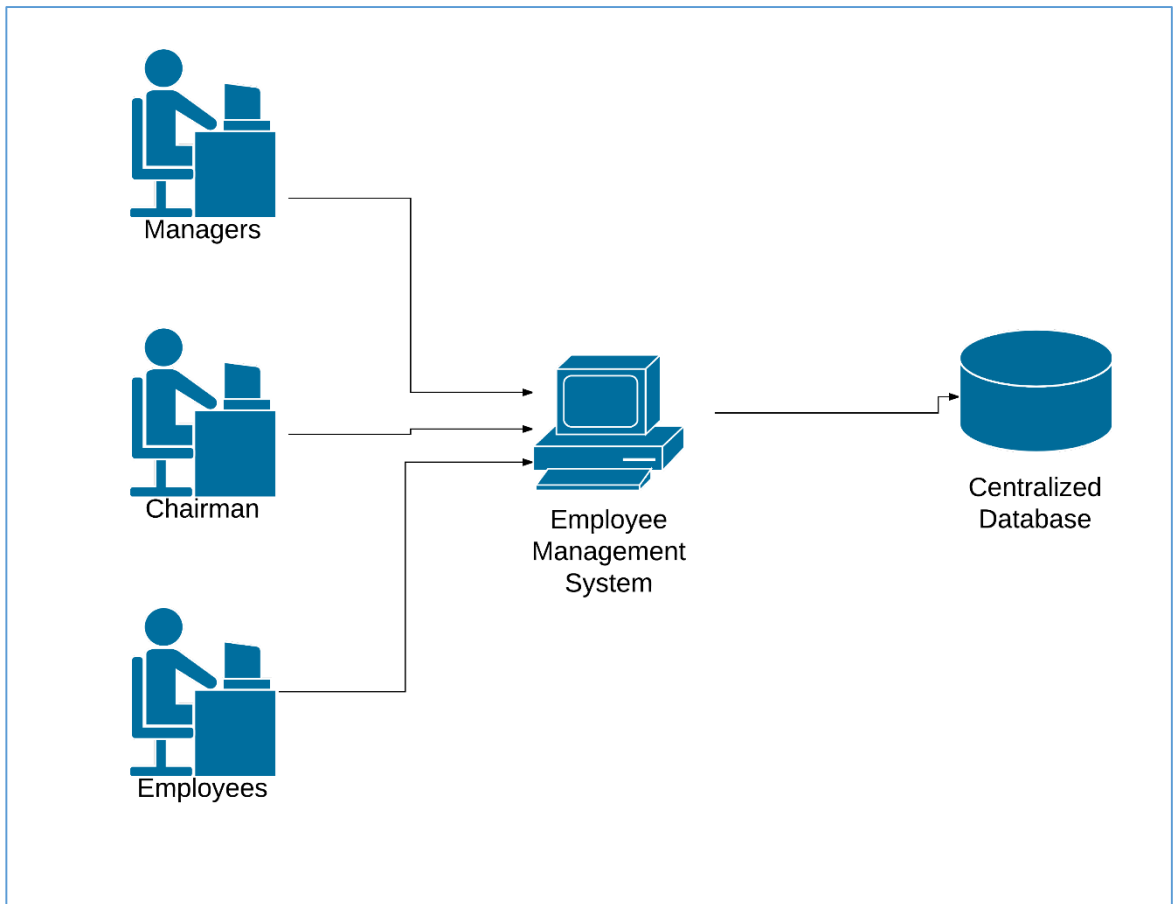


Figure 3.4 High level Architecture of the Proposed System

### 3.6 High level usecase of the system

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system [5]

Diagram in figure 3.5 depicts the high-level use case diagram of the proposed system

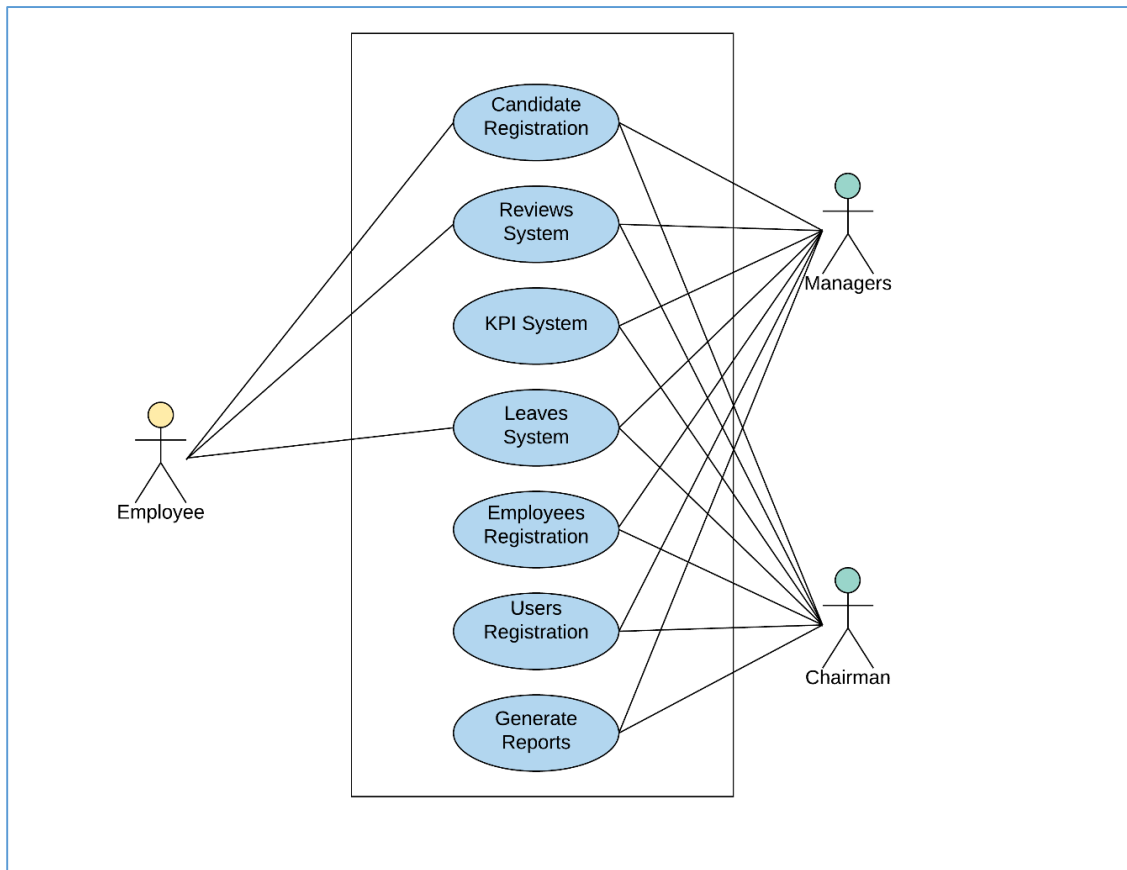


Figure 3.5 High level Usecase Diagram of the Proposed System

### 3.6 Module Design

The main modules of the system are described as follows,

### 3.6.1 Candidate Registration Module

The Candidate Registration module handles the registration of candidate according to the job title and can categorize the candidate according to three statuses which are Short Listed, Considerable and Rejected. Users from any user level can register a candidate. But only the users with Admin privileges can shortlist them according to the statuses mentioned above. Diagram in figure 3.6 depicts the use case diagram for Candidate registration.

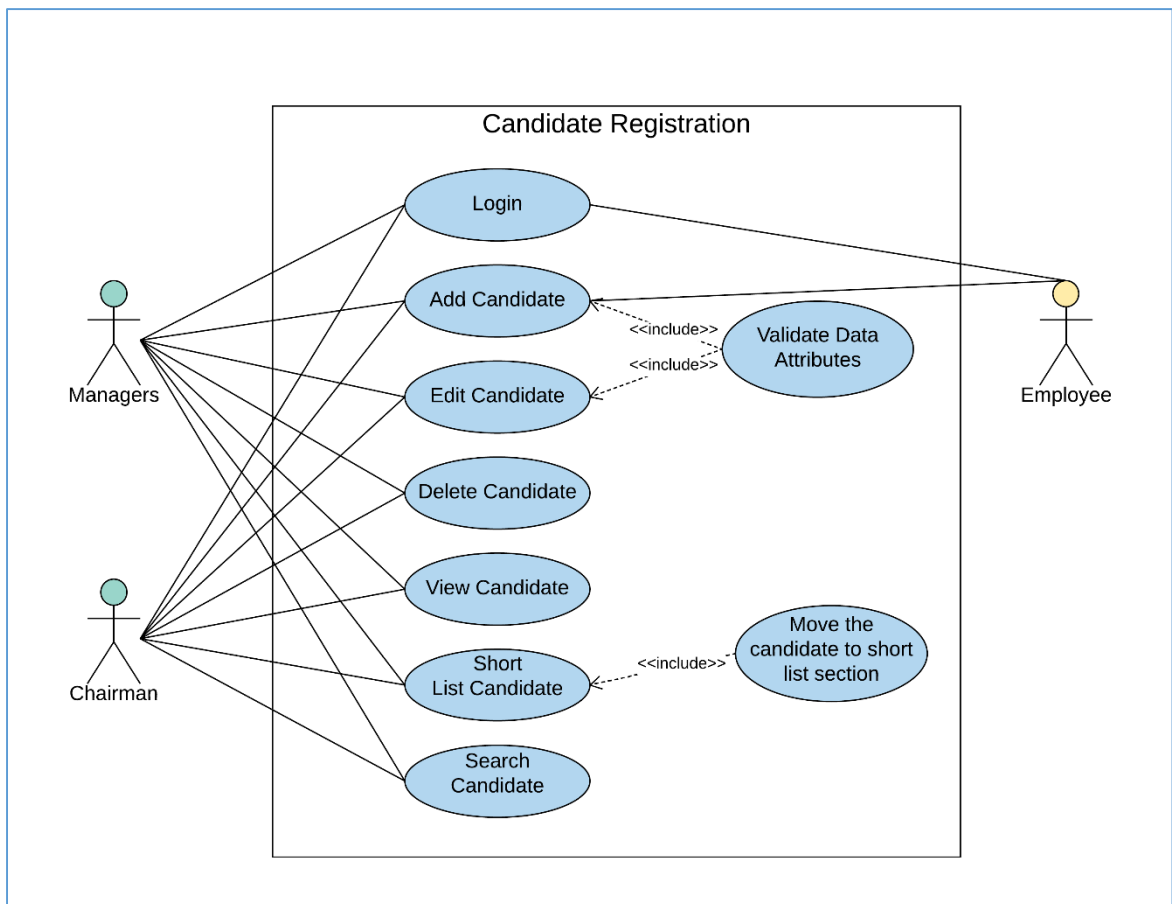


Figure 3.6 Usecase Diagram for Candidate Registration

## Candidate Registration Usecase

Following usecase (Table 3.1) illustrate registering a candidate to the system. Any user can register a new candidate for a selected vacancy. After adding a candidate admin users can shortlist the according to the given statuses.

Use Case	Candidate Registration
Actors	HR Executive, Employee, Manager, Chairman
Overview	
	Register new candidate to the system.
Preconditions	
	Actors must log in to the system.
Flow of Events	
	<ul style="list-style-type: none"><li>• Login</li><li>• Add Candidates according to the Job title</li><li>• Will list the added candidate in the list of vacancy section tables under the entered job title</li><li>• Shortlisted candidate will list in candidates section according to the shortlist status</li><li>• When uploading the CV if it violates given condition will redirect to the candidate add window with the specific error</li></ul>
Post Conditions	
	<ul style="list-style-type: none"><li>• New Candidate added to the system</li></ul>

Table 3.1 Candidate Registration Usecase

### 3.6.2 Review system Module

Review system module handles reviews of the employees which each and every employee can give reviews about other employees. Only the Admin user's type can edit or delete a review. Diagram in figure 3.7 depicts the use case diagram of Review system.

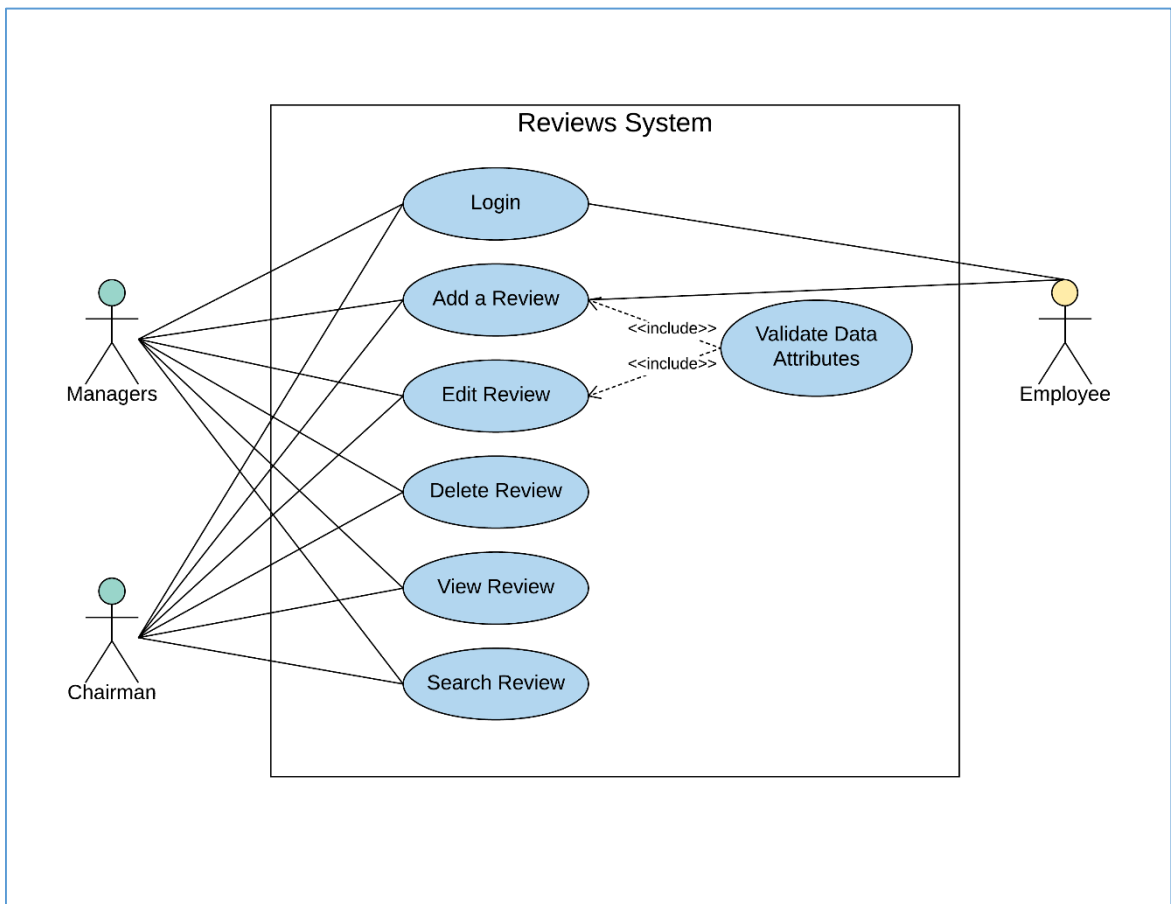


Figure 3.7 Usecase Diagram of Review System



### 3.6.3 KPI system module

KPI system module handles the KPI (Key performance indicator) functions which are added KPI, Edit KPI, Give marks for the each KPI as such. Admin users can give KPIs to the employees and they can give marks for each and every KPI after the completion of KPIs. This will benefit the Company management in the annual performance evaluation. Diagram in figure 3.8 depicts the use case diagram of KPI system

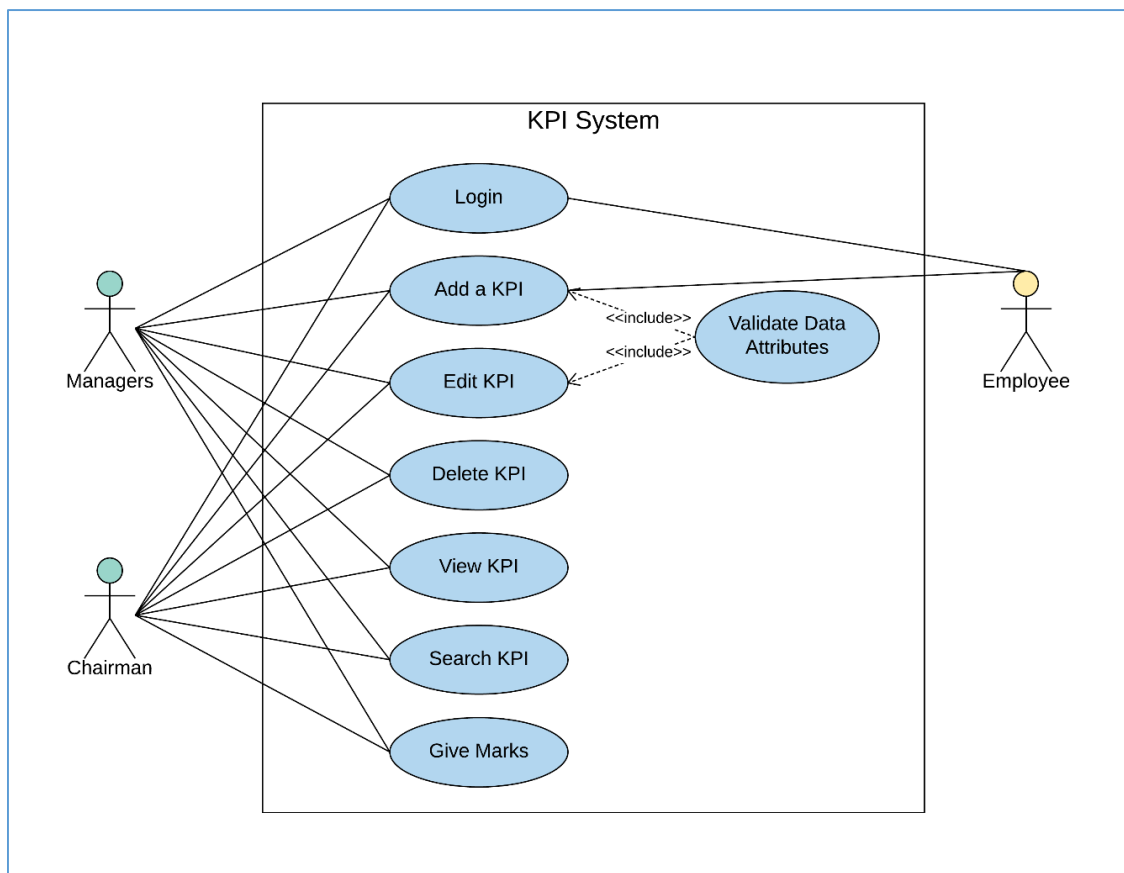


Figure 3.8 Usecase Diagram of KPI System

### 3.6.4 Leave Module

Leave module handle the leaves of the employees which are as annual leaves, medical leaves, and casual leaves. Every employee must get the approval before taking the leave. Only the Admin users can approve a leave and they will receive a relevant message after approving a leave. No employee can apply any type of leave after they are done the leave quota. Diagram in figure 3.9 depicts the use case diagram of Leave module.

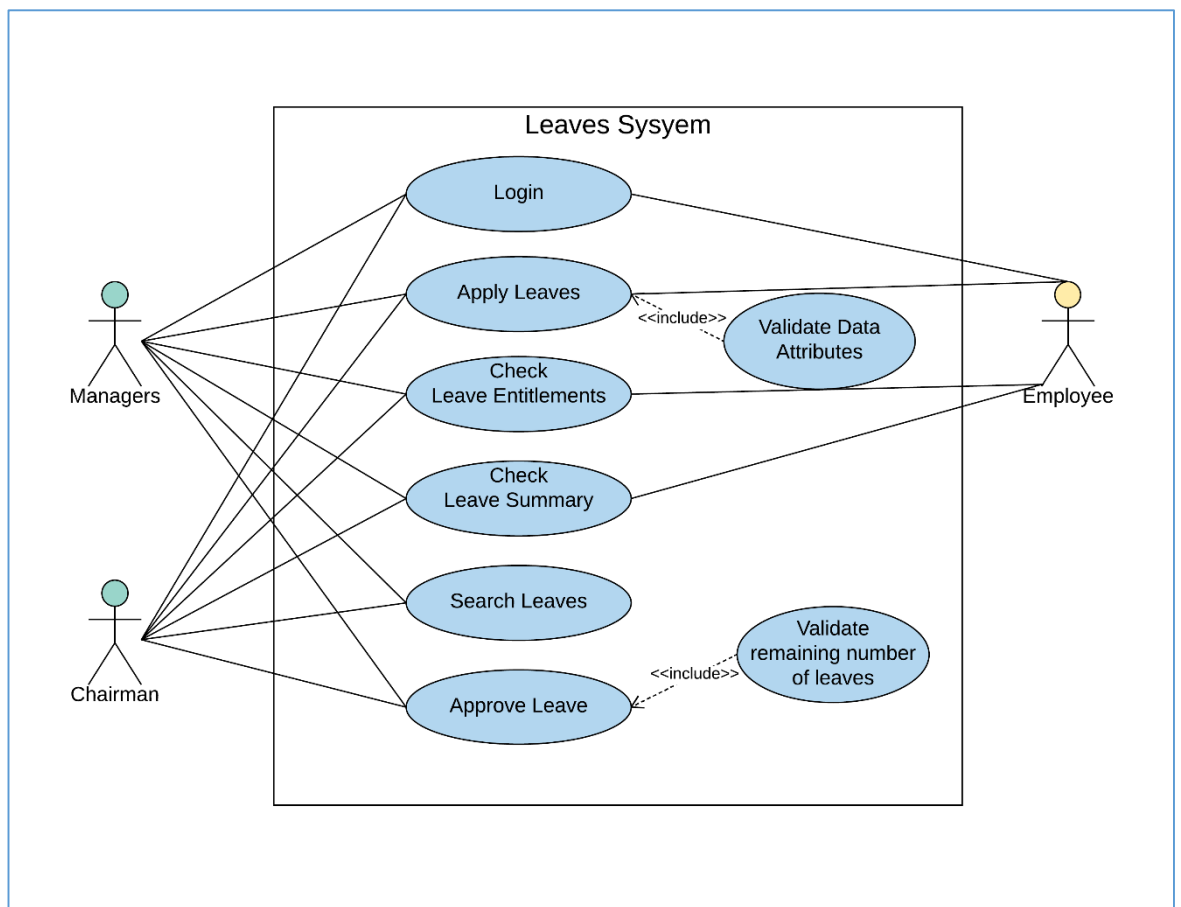


Figure 3.9 Usecase Diagram of Leave Module

## Leave Module Usecase

Following usecase (Table 3.2) illustrate all the leave module functions. Admin users can view and search other employee’s leaves, leave entitlements, leave summary and also can approve leaves on the other hand employee users can view and search their own leaves.

Use Case	Leave Module
Actors	Employee, Managers, Chairman
Overview	
Leave module functions.	
Preconditions	
Actors must log in to the system.	
Flow of Events	
<ul style="list-style-type: none"> <li>• Login</li> <li>• Apply a leave</li> <li>• Admin user approve the leave if it enough leaves remaining</li> <li>• Admin user will not approve the leave</li> <li>• Admin user get the successful message if the total leaves are less than applied leaves or error messages if the total leaves are exceeding the applied leaves</li> <li>• Admin user can see leave summary and leave entitlements of all the employees. On the other hand employee user can see only his/her leave summary and leave entitlement</li> </ul>	
Post Conditions	
<ul style="list-style-type: none"> <li>• Applied number of leaves will deduct from the system if the message is successful</li> <li>• Number of the remaining leaves in the data base will remain as it was</li> </ul>	

Table 3.2 Leave Module Usecase

### 3.6.5 Employee Registration Module

Employee Registration module handles the add, edit, delete, search and view of the employee details. Only admin users can handle the employee details. All the leave types are assigning to the employee after entering an employee record. Diagram in figure 3.10 depicts the use case diagram of Employee Registration module.

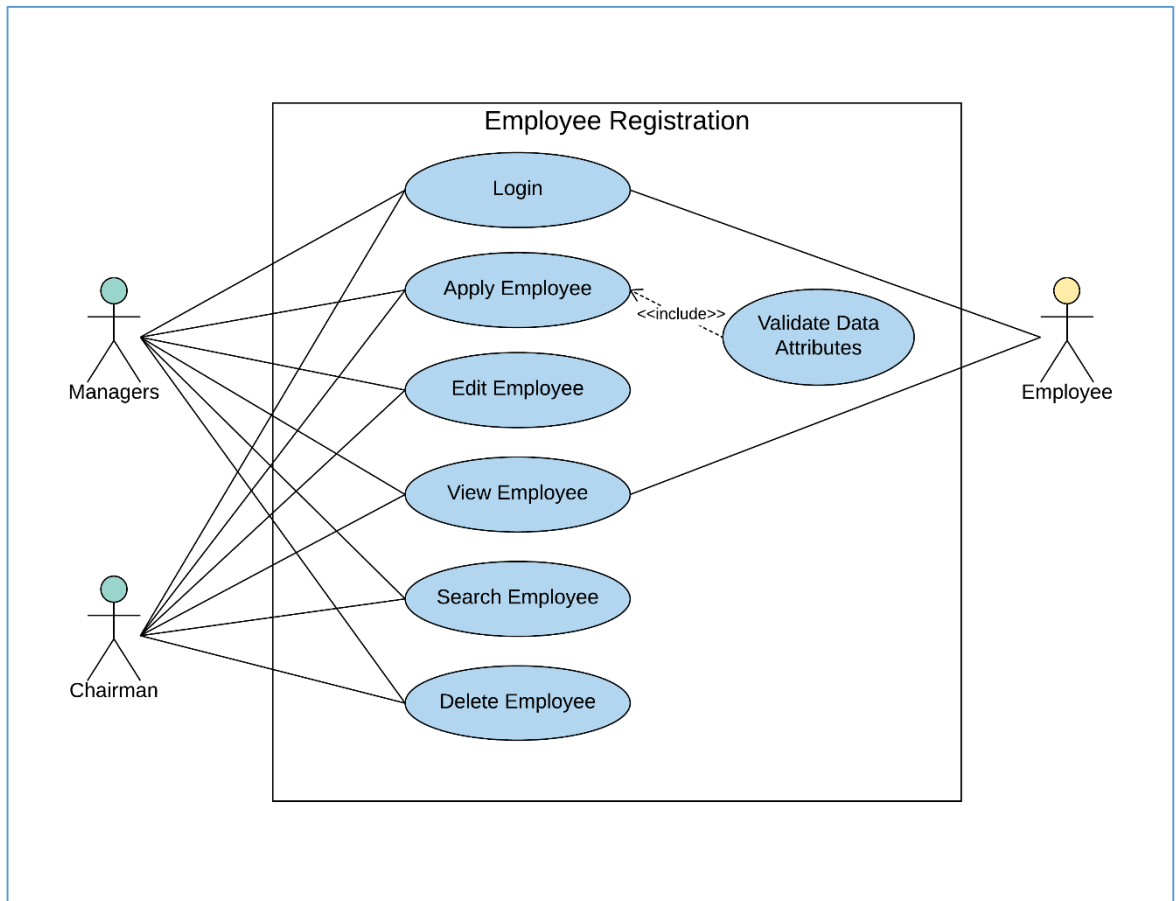


Figure 3.10 usecase Diagram of Employee Registration Module

## Employee Registration module Usecase

Following Usecase (Table 3.3) illustrate the Employee registration module functions. Only the admin user can register new employees to the system and the can edit, delete, search and view functions available.

Use Case	Employee Registration Module
Actors	Employee, Managers, Chairman
Overview	
Employee Registration module functions.	
Preconditions	
Actors must log in to the system.	
Flow of Events	
<ul style="list-style-type: none"><li>• Login</li><li>• Add an employee</li><li>• Edit employee details , search employee , view employee</li><li>• To print employee details click print button in view window</li></ul>	
Post Conditions	
<ul style="list-style-type: none"><li>• Employee record will add to the system</li><li>• Leaves of 14 annual,7 medical,7casual will automatically assign to the system</li></ul>	

Table 3.3 Employee Registration Usecase

### 3.6.6 User Registration module

User registration module handles the create, delete, update and search for the users of the system. When creating a user, the user will give a user type (admin or employee), username and password. The user can log in to the system using those give username and password. But those user levels have different accesses points of the system. Diagram in figure 3.11 depicts the use case diagram of user Registration module.

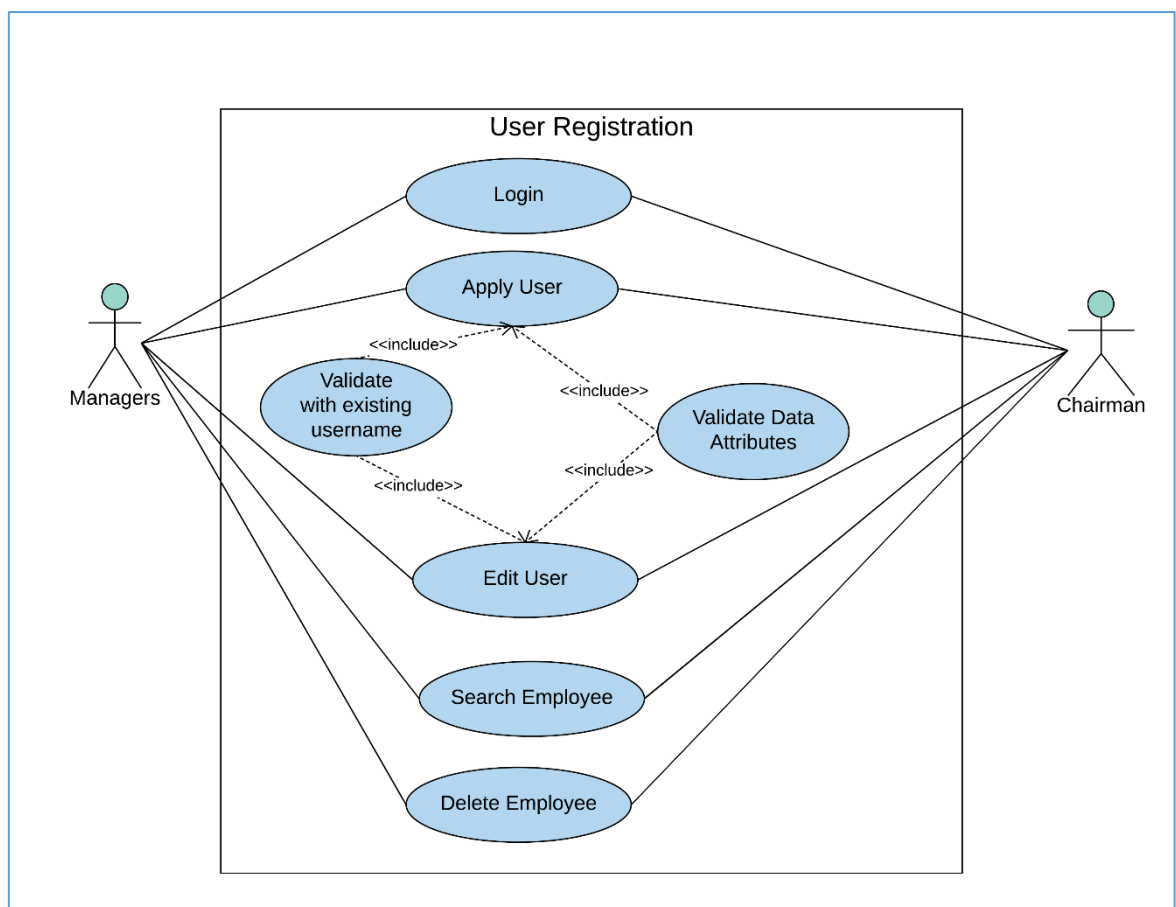


Figure 3.11 Usecase Diagram of User Registration Module

### 3.6.7 Generate Reports

The reporting module generate the summary reports of applied candidates per year, leave applied by employees per month, best 10 KPI marks per year, leave summary and leave entitlement. Only the admin users can see the repots generate menu item and the other functions. When making decisions, the managers or consultants can use generated reports in a timely manner. Diagram in figure 3.12 depicts the use case diagram of Generate reports module.

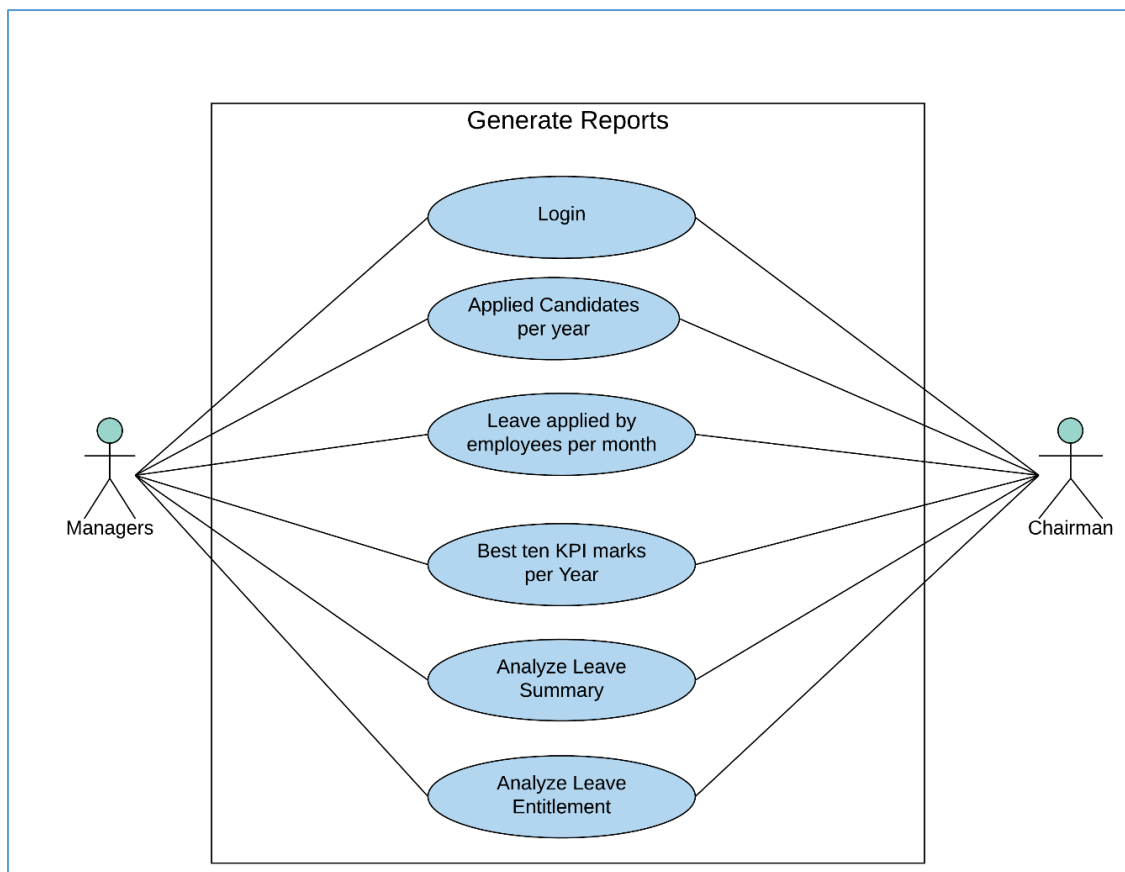


Figure 3.12 Usecase Diagram of Generate Reports Module

### 3.7 Sequence diagrams of the system

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. [6]

#### 3.7.1 Sequence diagram for the login to the system

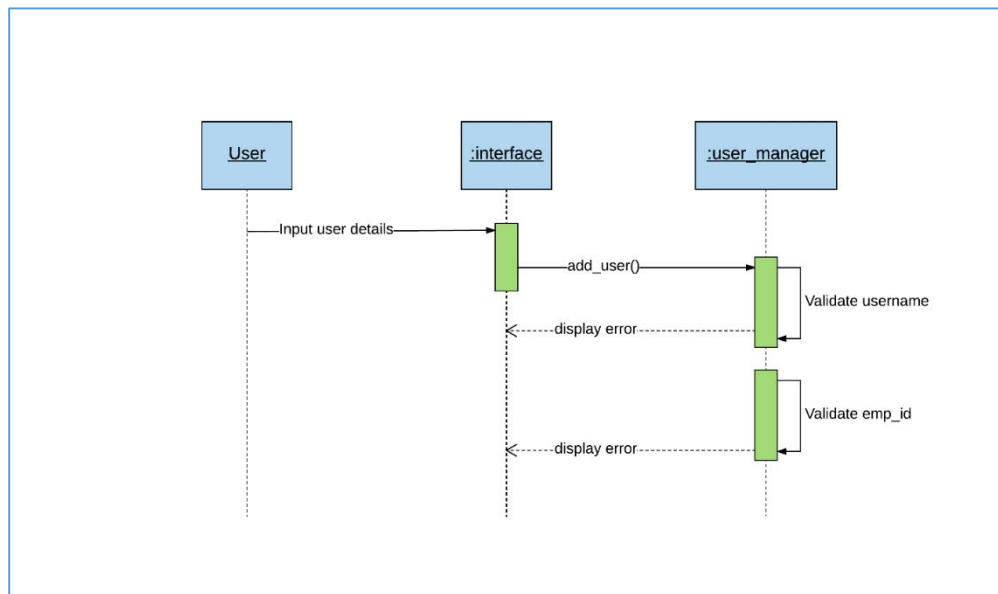


Figure 3.13 Sequence Diagram for the Login to the System

#### 3.7.2 User login sequence diagram

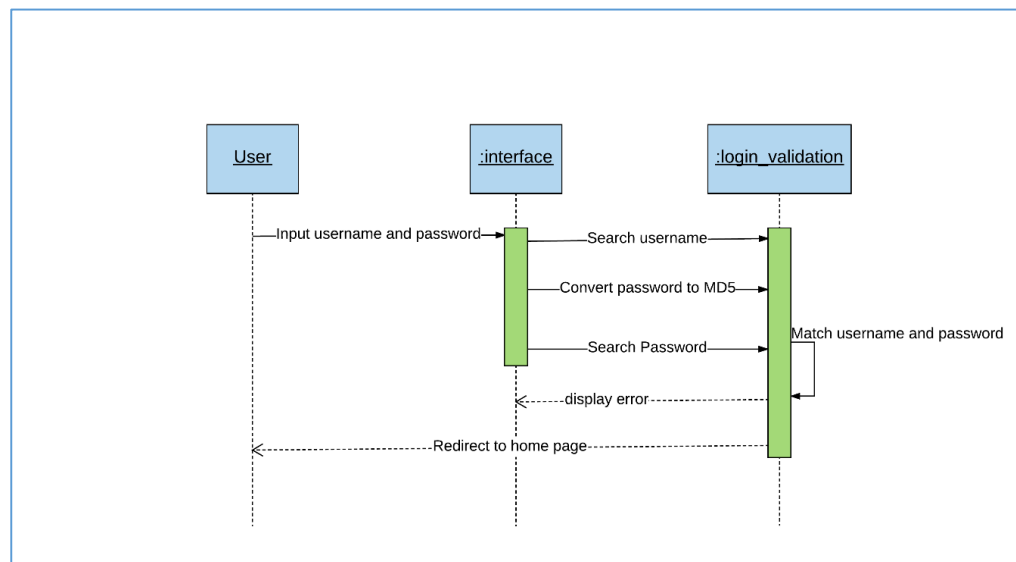


Figure 3.14 User login Sequence Diagram



### 3.7.3 Add review sequence diagram

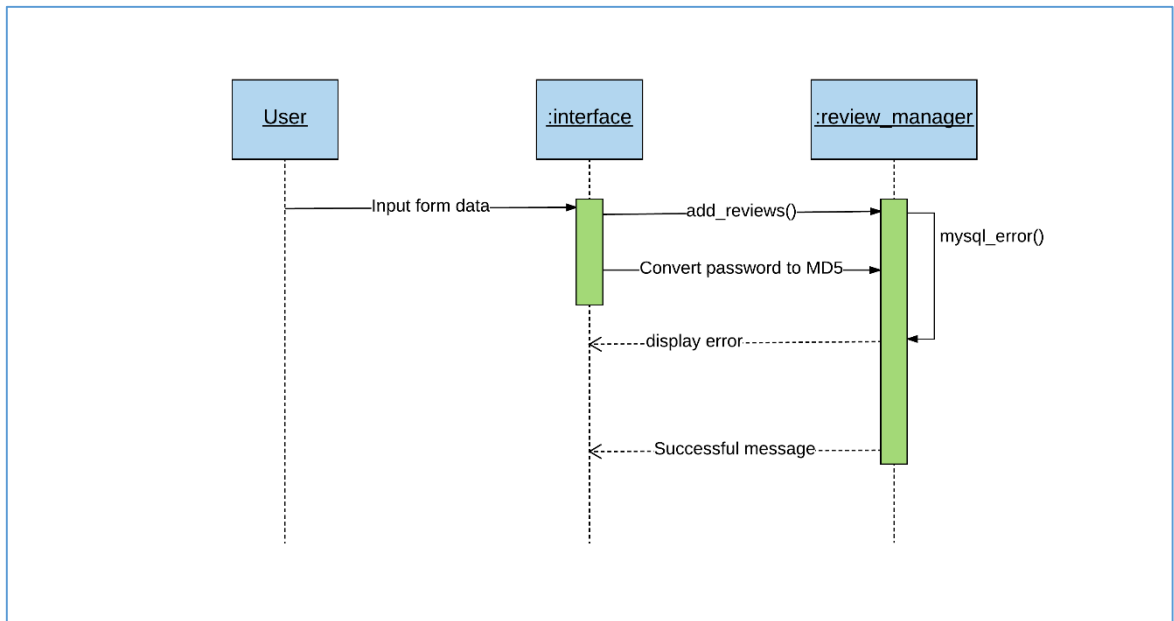


Figure 3.15 Add Review Sequence Diagram

### 3.8 Activity diagram for the system

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. Activity diagrams show the overall flow of control. [7]

### 3.8.1 Activity Diagram for the user login

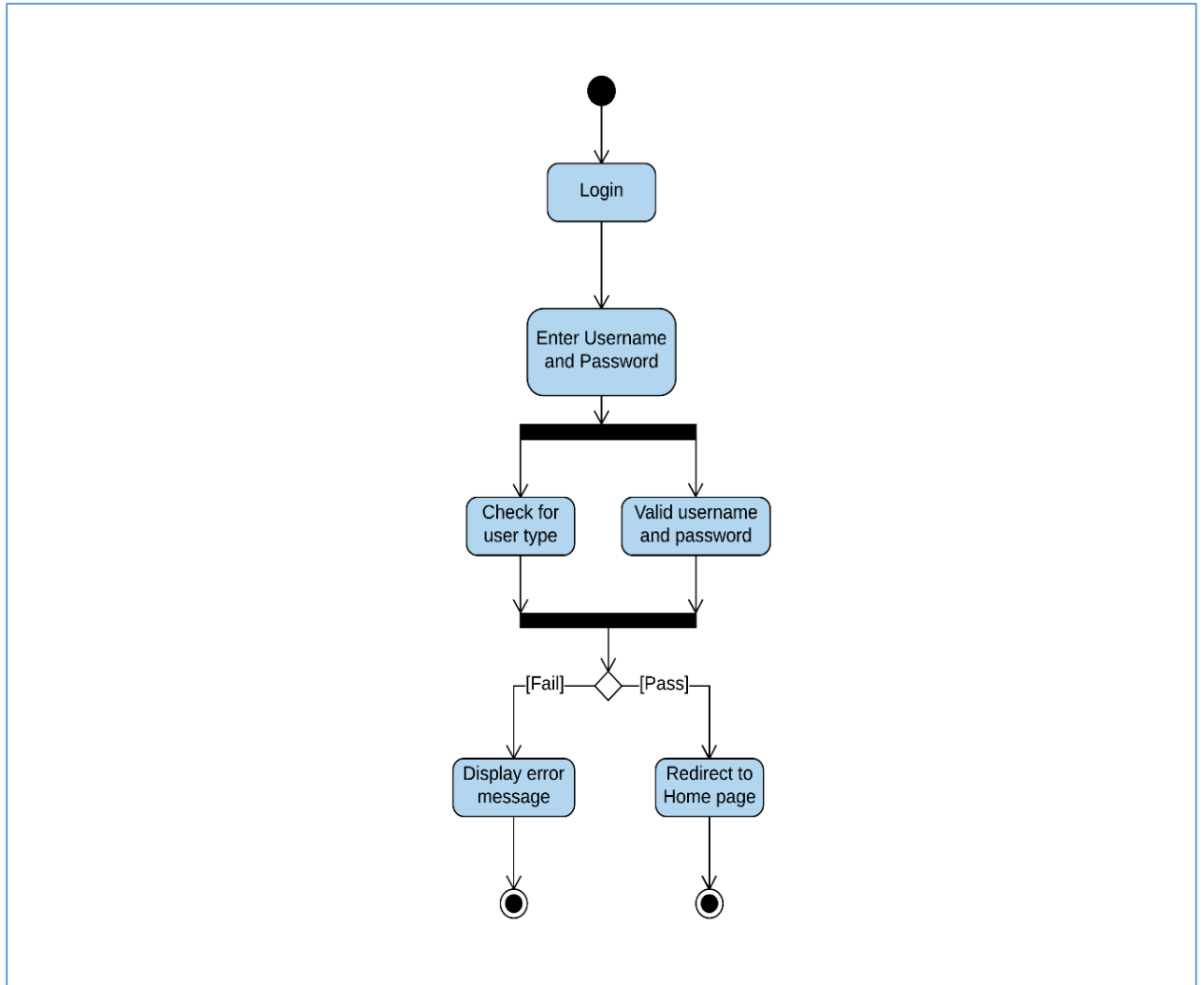


Figure 3.16 Activity Diagram for the User Login

### 3.9 Class Diagram

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes and the relationships among the classes[8].

The following figure 3.17 diagram illustrates the overall class diagram of the proposed system.

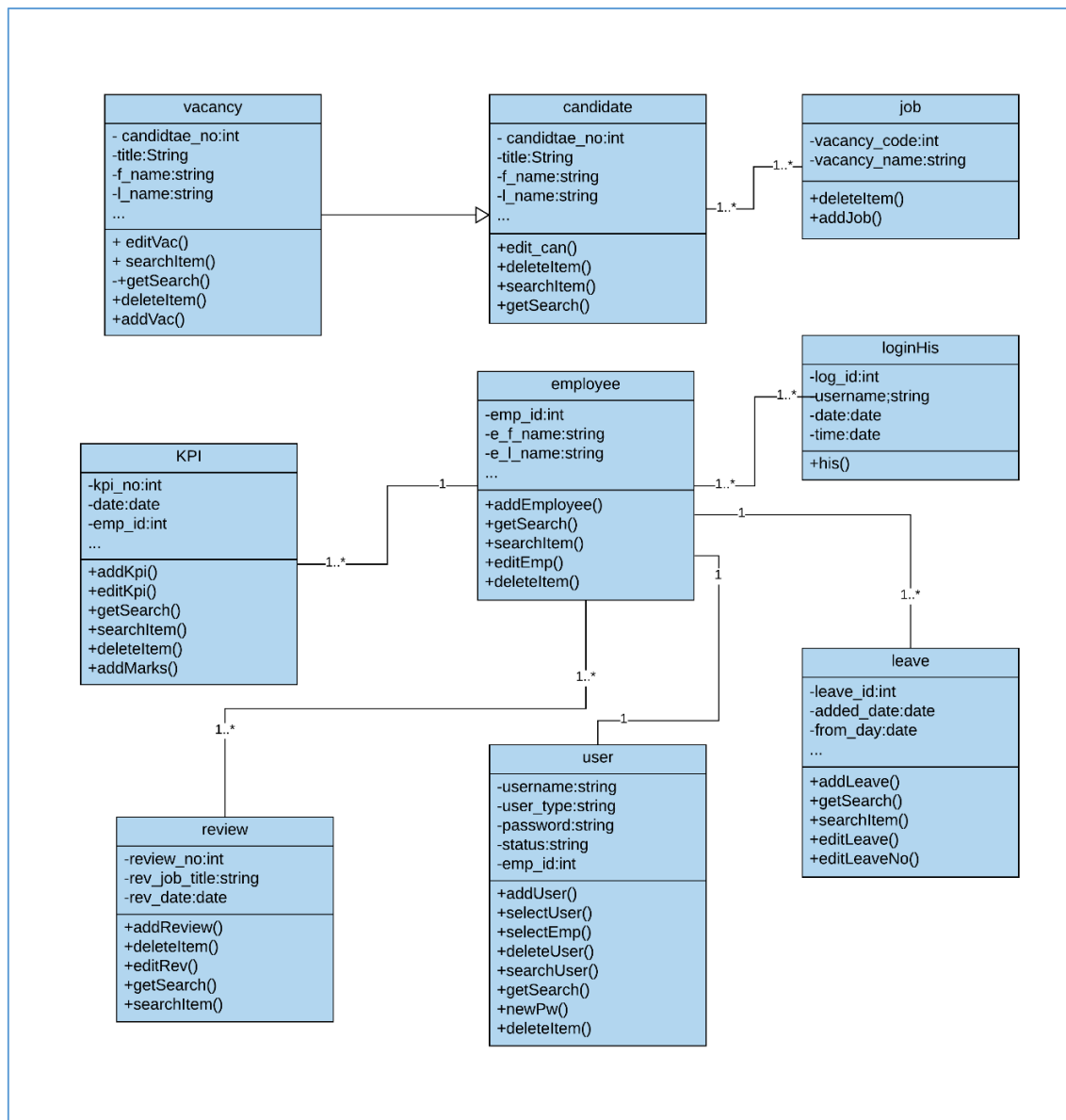


Figure 3.17 Overall Class Diagram of the Proposed System

### 3.10 Database Design

Database plays a major role in any application. Database design is the process of producing a detailed data model of a database.[9]

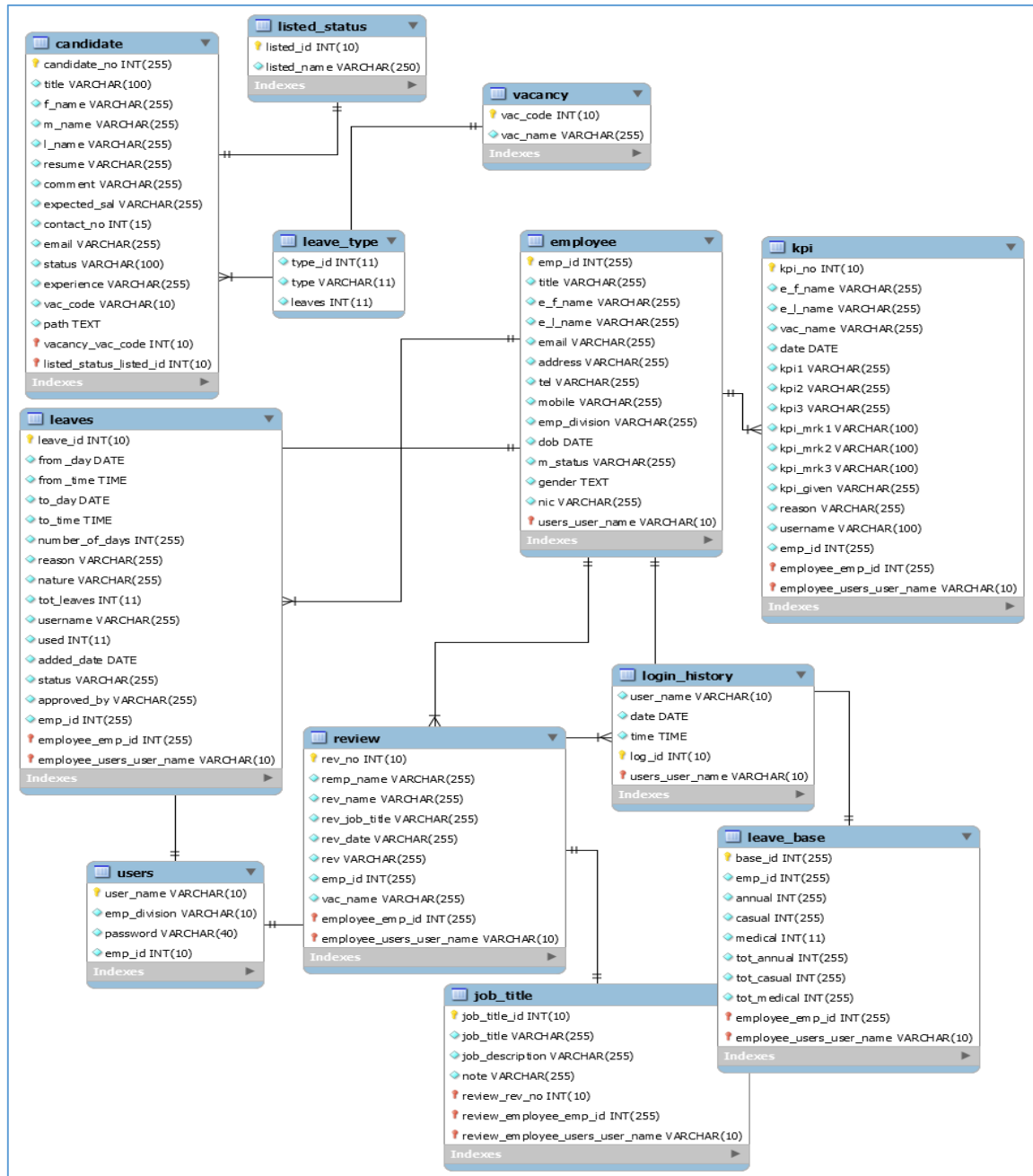


Figure 3.18 Data Diagram

### 3.11 User Interface Design

A GUI represents the information and actions available to a user through graphical icons and visual indicators such as secondary notation, as opposed to text-based interfaces, typed command labels or text navigation. The actions are usually performed through direct Manipulation of the graphical elements. [10]

There are many fundamental principals involved in the use of the visible Language.

- Organize: - provide a user with a clear and consistent conceptual structure.  
Speaking of consistency there are four views of consistency such as,
  - Internal – e.g.: dialogue boxes
  - External – e.g.: text tool icons
  - Real-world – e.g.: real-world signs
  - Not to be consistency
  
- Relationships:- linking related items and disassociating unrelated items can help  
Achieve visual organization
  
- Navigability:- There are main points of navigation
  - Provide an initial focus for the viewer’s attention
  - Direct attention to important, secondary or peripheral items
  - Assist in navigation throughout the material
  
- Simplicity: - includes only the elements that are most important for Communication. It should also be as unobtrusive as possible.
  
- Communicate: -GUI must keep in balance legibility , readability, typography, Symbolism, multiple views and color or texture in order to communicate Successfully

In this system, these tips have been used to create GUI for a better user-friendly system.

### 3.11.1 Login

This is the very first user interface which came across to the user when this system is used and the login screen is given in Figure 3.19. The user has to give a username and a password that is already given to them .Privileges are varied from user to user.

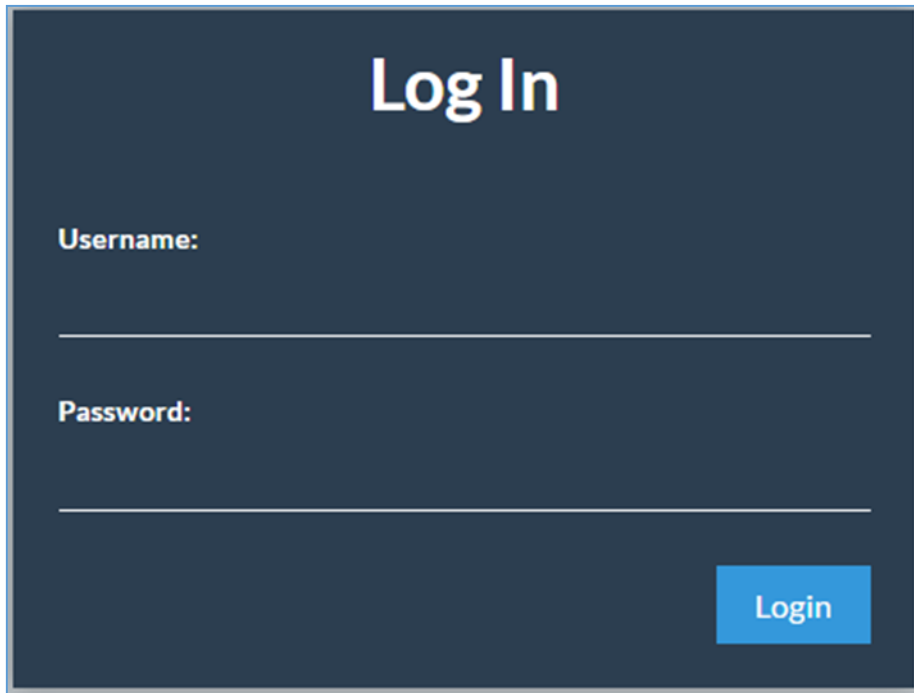
The image shows a dark blue login interface. At the top center, the text "Log In" is displayed in a large, white, sans-serif font. Below this, there are two input fields. The first is labeled "Username:" in white text, followed by a horizontal white line representing the input field. The second is labeled "Password:" in white text, followed by another horizontal white line. In the bottom right corner, there is a blue rectangular button with the word "Login" written in white text.

Figure 3.19 Login Interface of the System

### 3.11.2 Main Interface

The main interface which appears after bypassing the user login is included with all the functions of the system. Using this interface which is given in Figure 3.20, the user can search vacancy details and also he or she can modify or delete a particular record. The system has given the capability of searching a record in many ways such as first name, last name, employee Id, etc. Speaking of edit and delete functions, they happen within new windows which are loaded with all the details of the particular record.

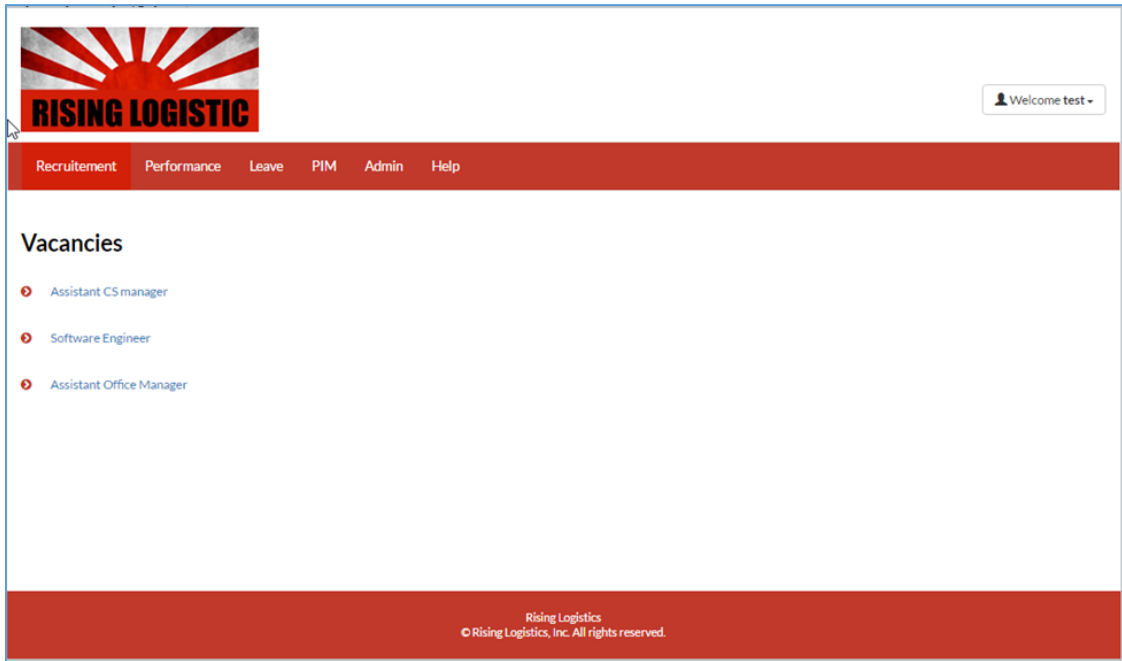



Figure 3.20 Main Interface of the System

### 3.11.3 Add Candidate form

Following figure 3.21 is the candidate adding form. All the required fields are marked in red color star and bottom of the page it is mentioned those marked fields are required fields. So the user will not misguide. The user can add a CV (curriculum vitae) of a candidate. But there are conditions for that and those conditions are mentioned in the bottom of the page under the “Submit Data” button.



Welcome test

Recruitment Performance Leave PIM Admin Help

### Add Candidate

Vacancy*	<input type="text" value="select--"/>	Title*	<input type="text" value="Choose an option"/>
First Name*	<input type="text"/>	Last Name*	<input type="text"/>
Email*	<input type="text"/>	Contact Number*	<input type="text"/>
Expected Salary*	<input type="text"/>	Experience	<input type="text"/>
Add CV	<input type="button" value="Choose File"/> No file chosen	Other Details	<input type="text"/>

\*Required Fields | Allowed cv file types are : .pdf, .doc, .docx and Max file size is 2Mb

Figure 3.21 Add Candidate Form



# Chapter 4 Implementation

This chapter will deliberate the facts which are necessary for implementation of the system. This chapter includes software environment, hardware environment, architectural overview, reused modules and network diagram of the implementation environment, code structure, and security.

## 4.1 Implementation Environment Development Tools

This system has implemented in a web server because this is web-based system. Also, in the company, they are having around five computers which have following hardware and software requirement.

### 4.1.1 Hardware Requirement

- Intel Core i3-370M processor 2.4 GHz
- 2GB RAM
- 320GB Hard disk

### 4.1.2 Software Requirement

- MS Windows 7 Home
- Microsoft Internet Explorer 6 Or latest version and Mozilla Firefox
- Do PDF
- Necessary sever software for web server

### 4.1.3 Development Tools

- Macromedia Dreamweaver cs5
- Photoshop CS5
- XAMP server 2.4
- Eclipse IDE
- Bootstrap 2

## 4.2 Architectural Overview

This system developed based on the Model-View-Controller (MVC) software architecture. MVC is probably one of the most quoted patterns in the web programming world in the recent years. The following figure represents the MVC design diagram.

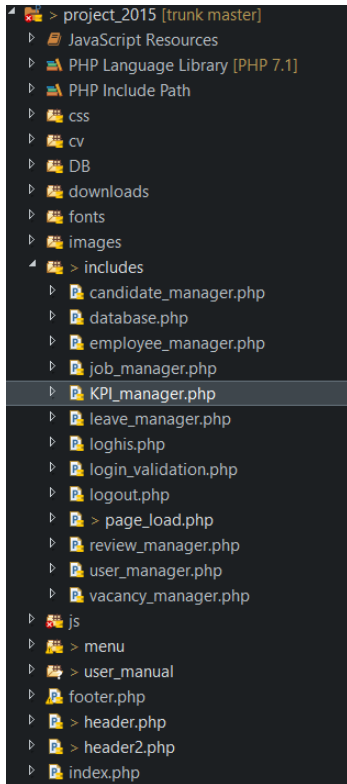


Figure 4.2 Directory Structure

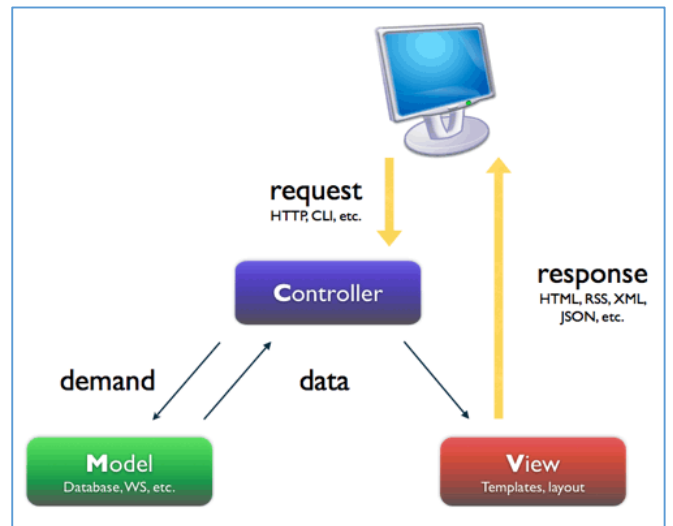


Figure 4.1 MVC Architecture

This application is divided into three main components, each one in charge of different tasks.

The **Model** is the data and the rules applying to that data, which represent concepts that the application manages. In any software system, everything is modeled as data that we handle in a certain way.

The **View** provides different ways to present the data received from the model. They may be templates where that data is filled. There may be several different views and the controller has to decide which one to use.

The **Controller** manages the user request (received as HTTP GET or POST requests when the user clicks on GUI elements to perform actions). Its main function is to call and coordinate the necessary resources/objects needed to perform the user action. Usually, the controller will call the appropriate model for the task and then select the proper view. [11]

### 4.3 Network Diagram of the Implementation Environment

The following figure 4.3 represents the network diagram of the implementation environment.

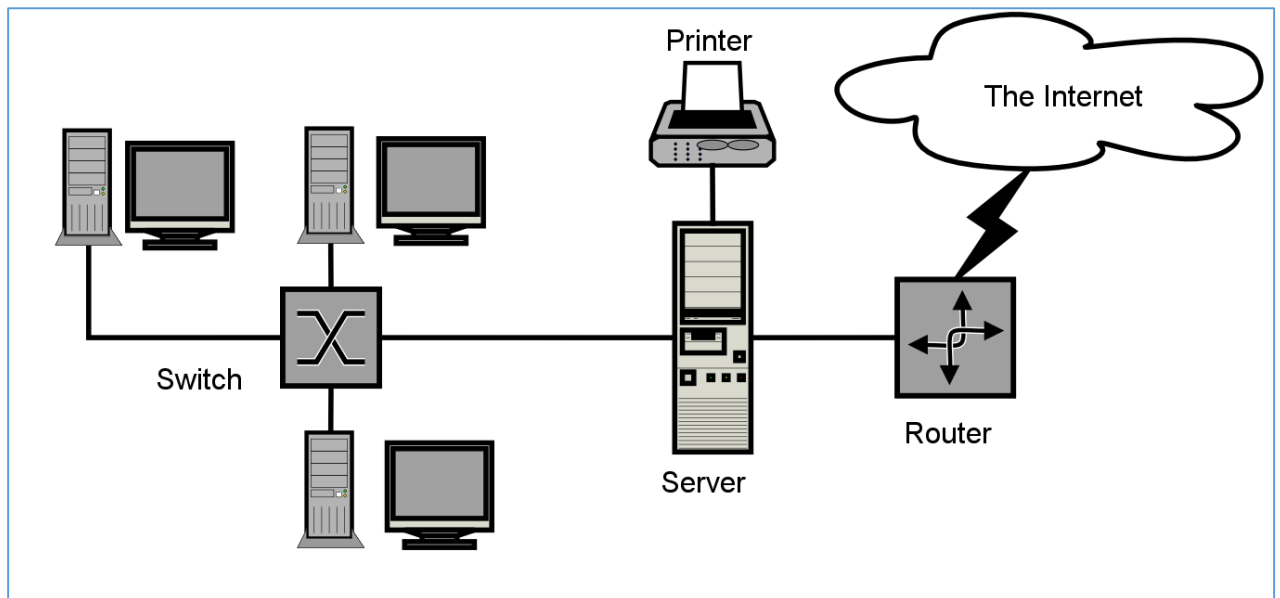


Figure 4.3 Network Diagram of the Implementation Environment

### 4.3 Major Code Structure

#### 4.3.1 Model

The model represents the data structures of the system. Typically Model classes contain functions that help to retrieve, insert and update information in the database. The following code fragment shows registration of a new candidate to the system.

```

<?php
public function addemployee($title, $fname, $lname, $email,
$address, $tel, $mobile, $user, $dob, $m_status, $nic, $gender) {
    $query = mysql_query ( "INSERT INTO employee
(title,e_f_name,e_l_name,email,address,tel,mobile,vacancy,dob,m_stat
us,gender,nic)VALUES(
        '$title','$fname','$lname','$email','$address
','$tel','$mobile','$user','$dob','$m_status','$gender','$nic') " )
or die ( mysql_error ( ) );
    $uid = 1;
    $eid = mysql_insert_id();
    return array($uid,$eid);

}
?>

```

The following code segment retrieve all the KPI details according to the search parameters

```

<?php
public function getSearch($id, $pa_id, $tbl_name) {
    //search with in the search page
    if ($pa_id == "f_name") {
        $sql = mysql_query ( "SELECT * FROM $tbl_name WHERE
e_f_name LIKE '%$id%' " ) or die ( mysql_error ( ) );
    } else if ($pa_id == "l_name") {
        $sql = mysql_query ( "SELECT * FROM $tbl_name WHERE
e_l_name LIKE '%$id%' " ) or die ( mysql_error ( ) );
    } else if ($pa_id == "date") {
        $sql = mysql_query ( "SELECT * FROM $tbl_name WHERE date
LIKE '%$id%' " ) or die ( mysql_error ( ) );
    } else {
        echo "<p>Please enter a search query</p>";
    }

    Return $sql;
}
?>

```

The following code segment edit the leave details in the database

```

<?php
public function editleave($from_date, $from_time, $days,
$review, $nature, $approved_by, $approved, $emp_id1) {
    $query = mysql_query ( "UPDATE leaves SET
from_day='$from_date', from_time='$from_time',
        number_of_days='$days', reason='$review',
nature='$nature', approved_by='$approved_by', status='$approved'
WHERE leave_id= '$emp_id1' " ) or die ( mysql_error
( ) );
}
?>

```

### 4.3.2 Controller

The controller serves as an intermediary between the model and view. It gets the request and responses from each model and updates the view files according to the results. The following code fragment shows registering new employee controller.

```
<?php
include ('../../includes/employee_manager.php');

include ('../../includes/database.php');

$database->getConnection();
$title = $_POST['com']; //employee title
$fname = $_POST['fname']; // first name
$lname = $_POST['lname']; // last name
$email = $_POST['email']; //email
$address = $_POST['address']; // address
$tel = $_POST['tel']; // land line number
$mobile = $_POST['mobile']; // mobile number
$user = $_POST['user']; // position
$dob = $_POST['dob']; //date of birth
$m_status = $_POST['m_status']; //marital status
$nic = $_POST['nic']; // NIC number
$gender = $_POST['gender']; //gender
$employee = new employee(); // cerate employee object
$val = $employee->addemployee($title, $fname, $lname, $email,
$address, $tel, $mobile, $user, $dob, $m_status, $nic, $gender);
//insert in to the employee table

if ($val[0] == 1)
{
    $employee->leave_base($val[1], 14, 7, 7, 14, 7, 7); //set leave
in leave base table while creating an employee

header("Location:../../template1.php?tab=add_employee&err=true2");
}
else
{
header("Location:../../template1.php?tab=add_employee&err=false");
}

?>
```

The following code fragment shows new leave request controller

```
<?php
session_start ();
include ('../../includes/leave_manager.php');
include ('../../includes/database.php');
$database->getConnection ();
$add_date = date ( "Y-m-d" );
$from_date = $_POST ['from_date'];
$from_time = $_POST ['from_time'];

$days = $_POST ['days'];
$review = $_POST ['reason'];
$nature = $_POST ['nat'];
```

```

$username = $_SESSION ['uname'];
$emp_id = $_SESSION ['emp_id'];

$result = mysql_query ( "SELECT * FROM leave_base WHERE
emp_id='$emp_id'" );
$res = mysql_fetch_array ( $result );
$annual = $res ['annual'];
$casual = $res ['casual'];
$medical = $res ['medical'];
if ($nature == "annual") {
    $annual = $annual - $days;
} elseif ($nature == "casual") {
    $casual = $casual - $days;
} elseif ($nature == "medical") {
    $medical = $medical - $days;
} else {
    echo "Something went wrong";
}
}
$addleave = new leave ();
if ($nature == "annual" && $annual > 0) {
    $addleave->countleave ( $annual, $casual, $medical, $emp_id );
    $addleave->addleave ( $from_date, $from_time, $days, $review,
    $nature, $username, $emp_id, $add_date );
    header ( "Location:../../templatel.php?tab=add_leave&err=true2"
);
} elseif ($nature == "casual" && $casual > 0) {

    $addleave->countleave ( $annual, $casual, $medical, $emp_id );
    $addleave->addleave ( $from_date, $from_time, $days, $review,
    $nature, $username, $emp_id, $add_date );
    header ( "Location:../../templatel.php?tab=add_leave&err=true2"
);
} elseif ($nature == "medical" && $medical > 0) {

    $addleave->countleave ( $annual, $casual, $medical, $emp_id );
    $addleave->addleave ( $from_date, $from_time, $to_date,
    $to_time, $days, $review, $nature, $username, $emp_id, $add_date );
    header ( "Location:../../templatel.php?tab=add_leave&err=true2"
);
} else {

    header ( "Location:../../templatel.php?tab=add_leave&err=true"
);
}
?>

```

The following code segment shows candidate details edit controller.

```

<?php
include ('../../includes/candidate_manager.php');
include ('../../includes/database.php');
$database->getConnection ();
if (isset ( $_POST ['submit'] )) {
    if (isset ( $_GET ['x'] )) {
        $emp_id = $_GET ['x'];
    }
    if (isset ( $_GET ['p'] )) {
        $page = $_GET ['p'];
    }if (isset ( $_GET ['last'] )) {
        $num of last page= $ GET ['last'];
    }
}

```

```

}if (isset ( $_GET ['tot_p'] )) {
    $total_pages= $_GET ['tot_p'];
}
if (isset ( $_GET ['val'] )) {
    $x = $_GET ['val'];
}
$email = $_POST ['Ename'];
$experience = $_POST ['exp'];
$name = $_POST ['name'];
$number = $_POST ['num'];
$salary = $_POST ['sal'];
$other = $_POST ['other_details'];
$resume = $_POST ['resume'];
$lname = $_POST ['lname'];
if ($x == 1) {
    $editv = new candidate ();
    $editv->edit_can ( $email, $experience, $name, $number,
    $salary, $other, $resume, $lname, $emp_id );
    if ($num_of_last_page == 1 && ($page == $total_pages &&
    $total_pages != 1)) {
        $page = $page - 1;
        header ( "Location:short_listed.php?page=$page&v=$resume" );
    } else {
        header (
"Location:short_listed.php?page=$page&v=$resume" );
    }
} else if ($x == 2) {
    $editv = new candidate ();
    $editv->edit_can ( $email, $experience, $name, $number,
    $salary, $other, $resume, $lname, $emp_id );
    if ($num_of_last_page == 1 && ($page == $total_pages &&
    $total_pages != 1)) {
        $page = $page - 1;
        header (
"Location:short_listed_search.php?page=$page&v=$resume" );
    } else {
        header (
"Location:short_listed_search.php?page=$page&v=$resume" );
    }
}
}
?>

```

### 4.3.3 View

The view is the information that is being presented to a user and normally a web page. All of the things that user can see and respond to the screen such as buttons, display boxes and so forth are listed in the view. View designing takes a considerable amount of time in the system development phase. The following code segment shows Candidate List page.

```

//some code at the beginning

<table class="table table-striped">
    <tr>
        <th>Candidate Id</th>
        <th>Title</th>

```

```

        <th>First Name</th>
        <th>Last Name</th>
        <th>Experience</th>
        <th>Contact No</th>
        <th>Edit</th>
        <th>Delete</th>
        <th>View</th>
    </tr>
    <tr>
    <?php
    while ( $row = mysql_fetch_array ( $result ) )
    {
        $p = $row ["candidate_no"];
        $value = "1";
        ?>
        <td><?php echo $row['candidate_no']; ?></td>
        <td><?php echo $row['title']; ?></td>
        <td><?php echo $row['f_name']; ?></td>
        <td><?php echo $row['l_name']; ?></td>
        <td><?php echo $row['experience']; ?></td>
        <td><?php echo $row['contact_no']; ?></td>
        <td class="text-center"><a
href="vacancy_details.php?x=<?php echo $p; ?>&id=<?php echo $page;
?>&vac=<?php echo $va; ?>&value=<?php echo $value; ?>&last=<?php
echo $num_of_last_page;?>&tot_p=<?php echo $total_pages;?>"><!---->
        <span class="glyphicon glyphicon-edit"></span></a></td>
        <td class="text-center"><div id="<?php echo
$p; ?>" style="width: 50px;"><a href="javascript:showUser(<?php echo
$p; ?>)"
onclick="return confirm('Are you sure you want to delete?')"></div>
        <span class="glyphicon glyphicon-trash"></span> <!----> </a></td>
        <td class="text-center"><a
href="view_vac.php?x=<?php echo $p; ?>"target="_blank"> <!----><span
class="glyphicon glyphicon-zoom-in"></span></a></td></tr>
    <?php
    } // End of while loop
    ?>
    <!--end of table --> //some more codes at the bottom

```



### 4.3.4 Other code segments

#### Database Access code

The system is connected to the database by this code. The 'die' condition used for any errors which might occur during database connectivity.

```
<?php
class DBconnection {
    public function getConnection() {
        $server = 'localhost'; // server name
        $username = 'root'; // username
        $password = ''; // empty password

        $connection = mysql_connect ( $server, $username, $password
); // mysql connection
        mysql_select_db ( 'pro_2012', $connection );
        if (! $connection) {
            die ( 'Could not connect: ' . mysql_error ( ) ); // mysql
error if could not connect
        }

        return $connection;
    }
}
$database = new DBconnection (); // new object of DBconnection class
```

#### Including Pages into Admin Home Page

Admin home page only contains header and footer with it. The body of the web page is loaded via a page\_load.php file. Several web pages use the admin home page as master Page. Page\_load.php file is needed to load the pages inside the home page .Following code is a Part of that page.

```
<?php
if (! isset ( $_GET ['tab'] )) {
    $tab = "home";
} else {
    $tab = $_GET ['tab'];
}
if ($tab == 'home') {
    if($_SESSION['ut'] != 'Employee'){
        $include = "menu/vacancies/va_list.php"; // load home page
    }else{
        $include = "menu/add vacancy/add_vacancy_details.php";
    }
} else if ($tab == 'vacancies') {
    $include = "menu/candidate/candidate.php"; // load vacancy page
} else if ($tab == 'kpi_list') {
    $include = "menu/KPI list/KPI_list.php"; // load kpi list page
} else if ($tab == 'add_kpi') {
    $include = "menu/add KPI/add KPI.php"; // load add kpi page
} else if ($tab == 'add_reviews') {
```

```
$include = "menu/add_reviews/add_reviews.php"; // load add
review page
} else if ($tab == 'reviews') {
    $include = "menu/reviews/reviews.php"; // load review list page
} else if ($tab == 'add_leave') {
    $include = "menu/leave/add_leave.php"; // load add leave page
} else if ($tab == 'view_leave') {
    $include = "menu/leave/view_leave.php"; // load leave list page
} else if ($tab == 'leave_entitlement') {
    $include = "menu/leave_entitlement/leave_entitlement.php"; //
load leave entitlement page
} // some codes below
```

## 4.4 Reused Modules

**Pagination** [12] – a module which creates a pagination to navigate from page to page.

**File Upload** [13] – to upload a CV in the candidate register form.

**Record Delete Ajax Function** [14] – Delete a record in the table using ajax.

**Form Validation** [16] – to validate the form attributes.

**Login page redirection** [17] – user redirect to login page if not properly logged in.

**Suggestion box** [18] – given query will list down the matching data while typing part of it in a text box.

**Graphical bar charts** [19] – graphical bar charts used to present reports.

## 4.5 Security

The system contains sensitive information related to the employees and candidates. It is important that the system implement a good security. The passwords used in the system are encrypted using the SHA1 algorithm. Each user type access is limited according to their user role. The administrator has full privileges to do modifications to the system.

# Chapter 5 Evolution

This chapter will deliberate the facts which are necessary for evaluation. This chapter includes testing, test approach, test cases, test data, test results and acceptance testing.

## 5.1 Testing Procedure

Testing individual system components and evaluate expected outputs while developing the system, will result in the reduction of the complexity of the system testing process

The RUP development methodology was used to develop the Employee Management System, therefore the system is implemented in iteratively and incrementally. As a result, the testing procedure is carried out individually to the system components from the beginning.

From the development point of view, the system testing is divided into Blackbox testing Whitebox testing.

- **Blackbox Testing:** a software testing technique that examines the functionality of a software application without having any knowledge of the coding.
- **Whitebox Testing:** a software testing technique that tests internal structure (codes) of an application.

### 5.1.1 Other testing technique

- **Integration Testing:** In integration testing, individual software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. Integration Testing focuses on checking data communication amongst these modules. Hence it is also termed as 'I & T' (Integration and Testing), 'String Testing' and sometimes 'Thread Testing' [20].
- **Regression Testing:** Regression testing is defined as a type of software testing to confirm that a recent program or code change has not adversely affected existing features. Regression Testing is nothing but a full or partial selection of already executed test cases which are re-executed to ensure existing functionalities work fine. This testing is done to make sure that new code changes should not have side effects on the existing functionalities. It ensures that old code still works once the new code changes are done [21].

- **Unit Testing:** Unit testing of software applications is done during the development (coding) of an application. The objective of Unit Testing is to isolate a section of code and verify its correctness. In procedural programming, a unit may be an individual function or procedure. The goal of Unit Testing is to isolate each part of the program and show that the individual parts are correct [22].

## 5.2 Test Plan and Result

Black box testing is performed on the system to diagnose errors. In addition, white box testing also performed after the black box testing to ensure the reliability and accuracy of the system.

### Test Cases

Test cases are prepared for almost every function and compared the actual output with the expected output. In this operation, some slight mistakes occurred and they were successfully re-corrected. Some test cases and screenshots are mentioned in Tables and others are referred in appendix E.

Objective	Input	Expected Output	Actual Output
Check combo box	No input	a combo box can not Be empty. Ex-: This is Required	a combo box can not Be empty. Ex-: This is Required

Table 5.1 Testcase for checking the Job Combo box

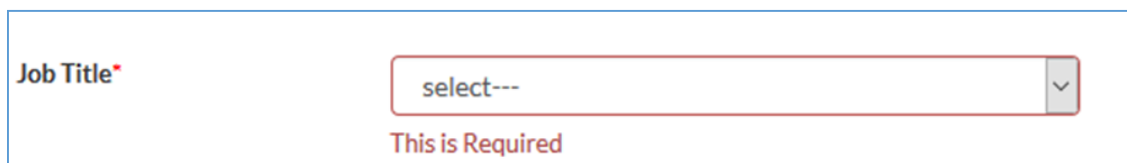


Figure 5.1 Check Title Combo Box

Objective	Input	Expected Output	Actual Output
Check all required fields	No input	Input box cannot be empty. Ex-: This is Required	Input box cannot be empty. Ex-: This is Required

Table 5.2 Testcase for Verifying other Fields

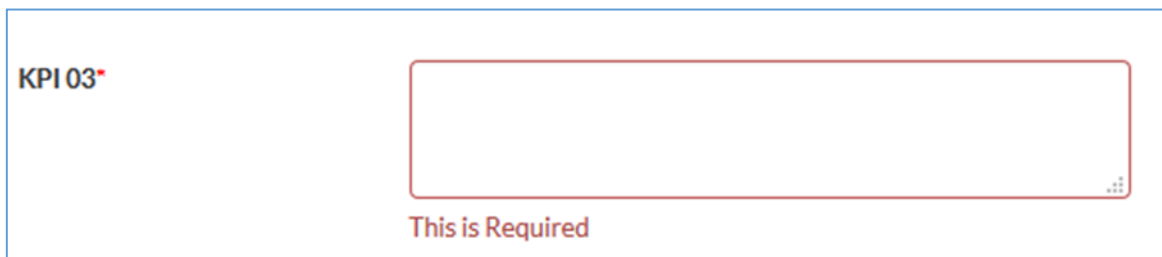


Figure 5.2 Checking Other Fields

Objective	Input	Expected Output	Actual Output
Check username already exists in database	Existing username	username cannot exist E.g. : This Employee Already has a Username	username cannot exist E.g. : This Employee Already has a Username

Table 5.3 Testcase for checking existing usernames

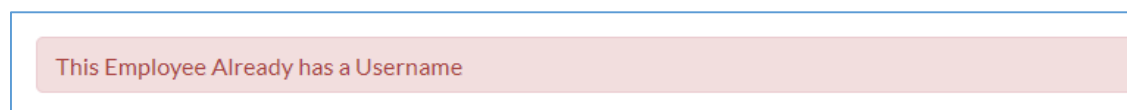


Figure 5.3 Checking an existing Username

Objective	Input	Expected Output	Actual Output
Check valid email address	Invalid email	Enter valid email E.g. : You have not given a correct e-mail address	Enter valid email E.g. : You have not given a correct e-mail address

Table 5.4 Testcase to validate Email field

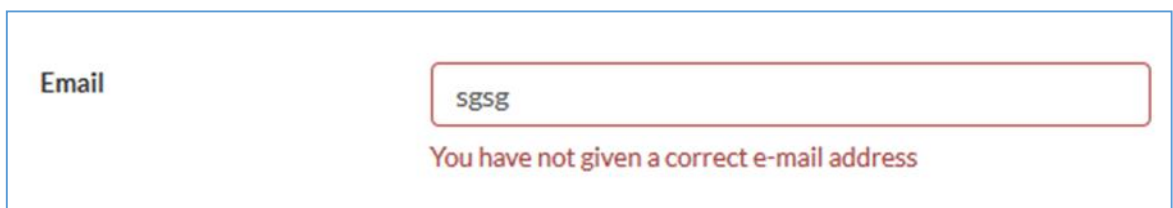


Figure 5.4 Email field with Main Validation

Objective	Input	Expected Output	Actual Output
Check the delete confirmation box	Click on delete icon	Confirmation box should display deletion	Confirmation box should display before deletion

Table 5.5 Testcase to validate delete confirmation

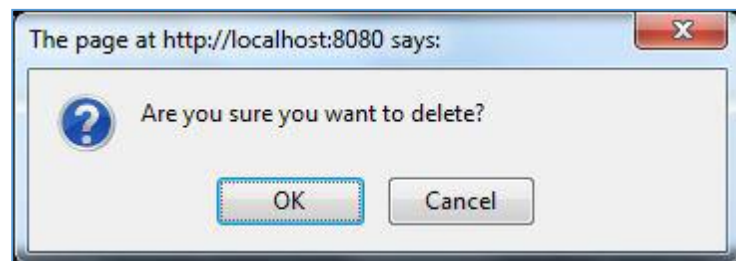
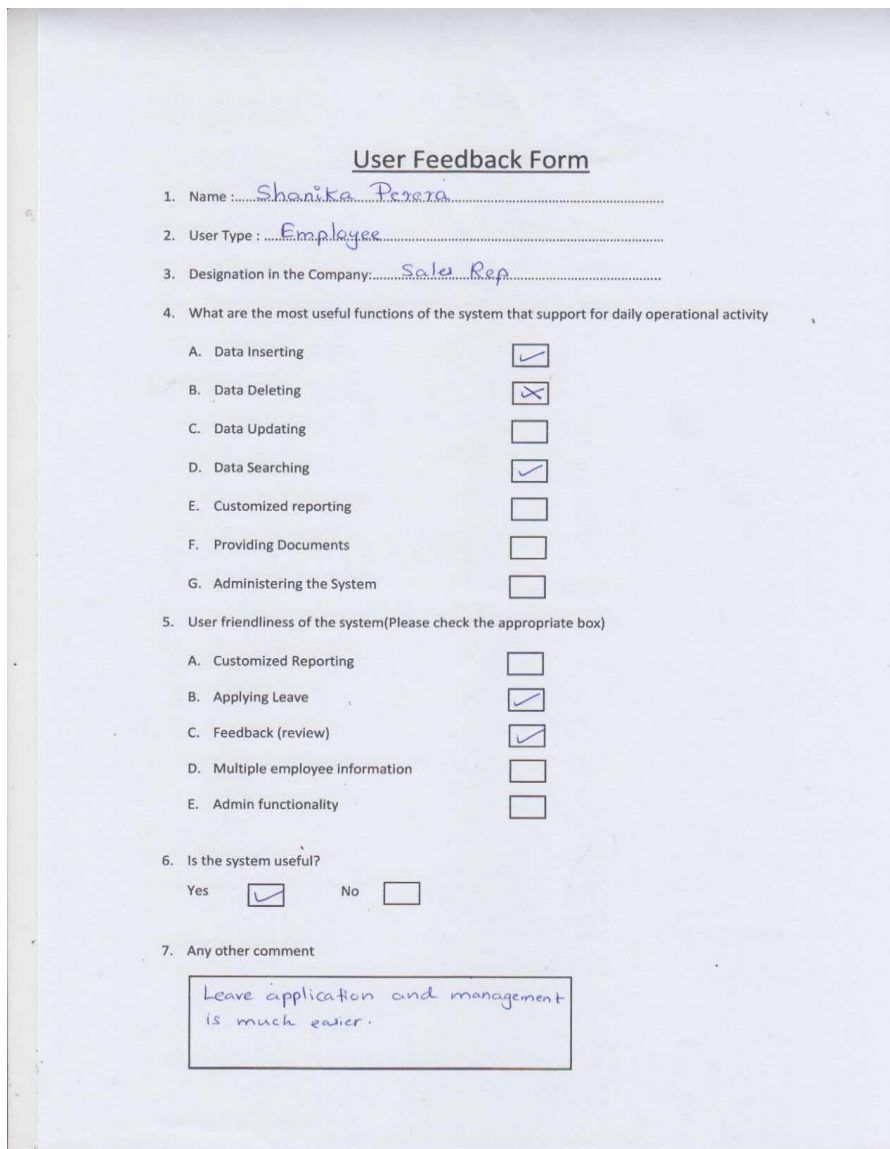


Figure 5.5 Delete confirmation box

## 5.2 User Evaluation

After System is implemented in Rising Logistics for beta testing, during beta testing; the system is uploaded to a local host and users to experience the system. It helped a lot for testing the system and to identify some errors in the system. Finally, beta testing was quite successful, because most of the bugs are fixed successfully and response of the users was acceptable.

### 5.2.1 User Feedback From



**User Feedback Form**

1. Name : Shanika Perera

2. User Type : Employee

3. Designation in the Company: Sales Rep

4. What are the most useful functions of the system that support for daily operational activity

A. Data Inserting	<input checked="" type="checkbox"/>
B. Data Deleting	<input checked="" type="checkbox"/>
C. Data Updating	<input type="checkbox"/>
D. Data Searching	<input checked="" type="checkbox"/>
E. Customized reporting	<input type="checkbox"/>
F. Providing Documents	<input type="checkbox"/>
G. Administering the System	<input type="checkbox"/>

5. User friendliness of the system (Please check the appropriate box)

A. Customized Reporting	<input type="checkbox"/>
B. Applying Leave	<input checked="" type="checkbox"/>
C. Feedback (review)	<input checked="" type="checkbox"/>
D. Multiple employee information	<input type="checkbox"/>
E. Admin functionality	<input type="checkbox"/>

6. Is the system useful?

Yes  No

7. Any other comment

Leave application and management is much easier.

Figure 5.6 User Feedback Form

### 5.2.2 Result of user feedback form

This feedback form was distributed to several employees, HR, and the Manager. All of them participated this questioner without hesitation. Eventually, results were positive considering all over results. Even though feedback session came up with too many positive feedbacks, there was a couple of negative feedback too. But, throughout further discussion, it revealed that those people came with negative feedbacks was very poor with computer literacy.

Question No	Answers to Results		
01	Owner - 1	Manager - 2	Employee - 10

Table 5.6 Answer to the question No 1

Question No	Answers to Results		
02	Admin - 1	Manager - 2	Employee - 10

Table 5.7 Answer to the question No 2

Question No	Answers to Results						
04	A-8	B - 03	C - 10	D - 10	E - 05	F - 06	G - 03

Table 5.8 Answer to the question No 4

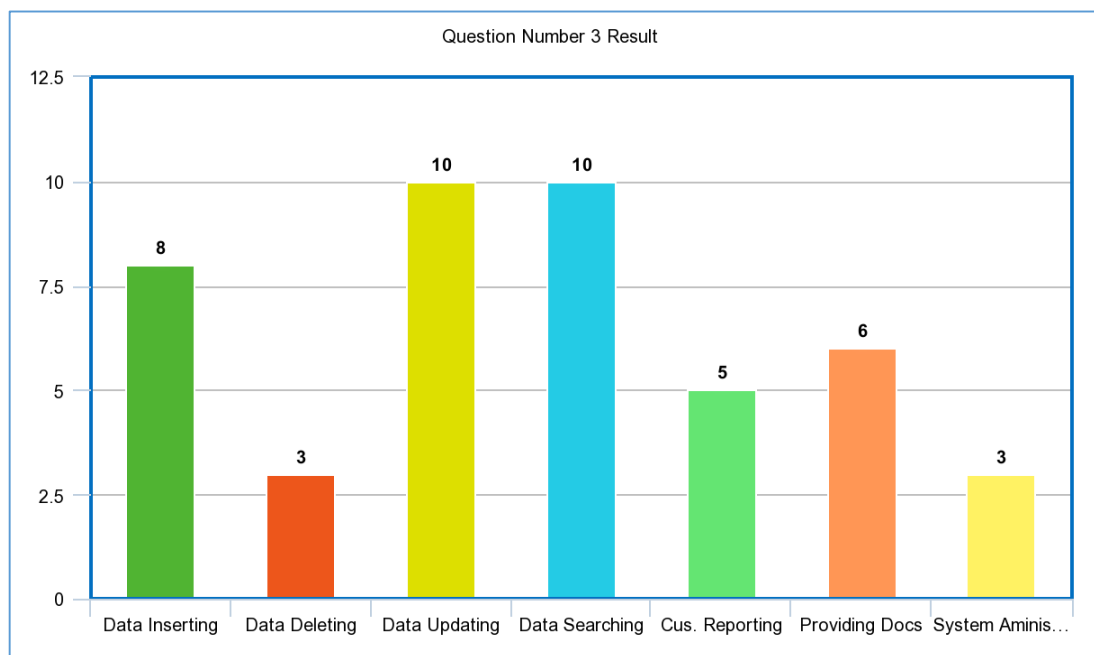


Figure 5.7 Question Number 3 Result



According to the above graph (Figure 5.7) generated from the Table 5.8 showed that the Data searching and Data updating functions were most useful to the users.

Question No	Answers to Results				
05	A-3	B-12	C-9	D-5	E-2

Table 5.9 Answer for the Question No 5

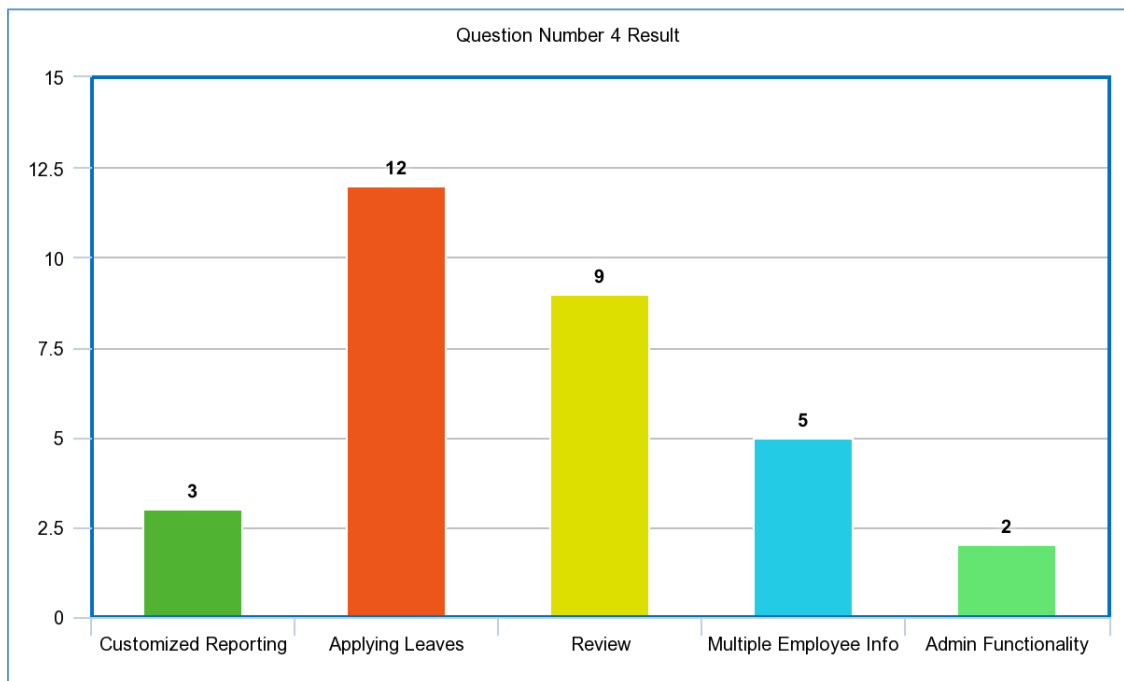


Figure 5.8 Question Number 4 Result

According to the above graph (Figure 5.8) generated from the Table 5.9 showed that the Applying Leaves function was very popular among the users. According to those figures in the graphs, Employee Management System has achieved the main objective of the project successfully.

# Chapter 6 Conclusion

This chapter will discuss the facts about conclusion. This chapter includes an overview, problems encountered, lessons learned, and future enhancement which could be made to the system.

## 6.1 Overview

Rising Logistics (Pvt) Ltd is a growing spare part selling the company in Sri Lanka. They do both retail and wholesale business with their customers. In the beginning, all the process are carried out manually. They needed to have new technological improvement of their business. To accomplish that requirement “Employee Management System” is presented. The productivity of the manual progression is improved by developing the system and also aided to improve the customer relationship with the business.

To develop the system functionalities, the manual process of the business supported in many ways. The system is saved all employee and candidate details. The attractive and simple user interface that was designed and developed, users can easily to absorb the system. The system allows efficient and effective leave apply and approve with no resource overlapping.

The system provides searching information. Most of the tables in the pages have search function according to the given parameters and also can search in a search page too. Also, the suggestion box which appears in most of the forms will guide the user to select exact data from the suggestion box while typing a part of a text of number.

The system was developed using the following development languages and technologies MVC, PHP, HTML, CSS, JavaScript, JQuery, and Ajax.

## 6.2 Problems Encountered

During the project, several difficult areas occurred and it was necessary to take some actions to overcome those difficulties. The main difficulties were reporting and the leave function of the system. Even though having a nice presentation of various kinds of reports in the Employee Management System there was a slight problem when it comes to print those reports. To overcome that had to use a JavaScript function that to put a print button that does the same thing via a browser. To overcome the issue with Session not expiring, found an open source code that added to the header.php page.

## 6.3 Lessons Learnt

- I was able to apply the knowledge learned throughout the BIT degree program for the system
- Learnt how to manage time effectively and efficiently by doing development process according to the time schedule
- Skill was gained with regards to effective interact and communicating with the client –improved communication skills
- I have enhanced the knowledge of PHP, HTML, Javascript, CSS, JQuery and many more development tools, techniques and frameworks.
- Learned about new model of MVC (model, view, controller) architecture
- Writing the dissertation gave me how to write a report in a professional manner and it helped me to improve my skills in writing and designing technical reports.

## 6.3 Future Work

Speaking of future work having an interactive website will help to increase the interaction between the user and the system as well as it will be a good marketing tactic to be followed. Besides, adding more information gathering part in the employee section such as skills, the image of the employee, education, emergency contact number as such and an attendance module will give this system a far more value and will create more customized reports which can be very easy to make decisions for the higher management.

Even though this system was built based on Bootstrap which is a UI framework, which the site can be converted to a nice responsive site did not have the enough time to do the UI because it is very time consuming and had a learning curve.

# References

- [1] Adp. “ADP Total Source” adp.com.[Online].Available:  
<http://www.adp.com/logins/adp-totalsource.aspx> [Accessed Feb 02,2017]
- [2] Orange Human Resource Management System. “Orange HR Management System” orangehrm.com. [Online].Available: <http://www.orangehrm.com/> [Accessed Feb 02, 2017]
- [3] Selectbs.”Software Development tools” selectbs.com.[Online].Available:  
<http://www.selectbs.com/analysis-and-design/what-is-a-software-developmentprocess>  
[Accessed Feb 10,2017]
- [4] Wikipedia. "Rapid application development" wikipedia.org. [Online].Available:  
[http://en.wikipedia.org/wiki/Rapid\\_application\\_development](http://en.wikipedia.org/wiki/Rapid_application_development) [Accessed Feb 11, 2017]
- [5] Wikipedia. "Usecase Diagram" wikipedia.org. [Online].Available:  
[http://en.wikipedia.org/wiki/Use\\_Case\\_Diagram](http://en.wikipedia.org/wiki/Use_Case_Diagram) [Accessed Feb 11, 2017]
- [6] Wikipedia. "Sequence Diagram" wikipedia.org. [Online].Available:  
[https://en.wikipedia.org/wiki/Sequence\\_diagram](https://en.wikipedia.org/wiki/Sequence_diagram) [Accessed Feb 11, 2017]
- [7] Wikipedia. "Activity Diagram" wikipedia.org. [Online].Available:  
[https://en.wikipedia.org/wiki/Activity\\_diagram](https://en.wikipedia.org/wiki/Activity_diagram) [Accessed Feb 12, 2017]
- [8] Wikipedia. "Class Diagram" wikipedia.org. [Online].Available:  
[http://en.wikipedia.org/wiki/Class\\_diagram](http://en.wikipedia.org/wiki/Class_diagram) [Accessed Feb 12, 2017]
- [9] Wikipedia. "Database Design" wikipedia.org. [Online].Available:  
[https://en.wikipedia.org/wiki/Database\\_design](https://en.wikipedia.org/wiki/Database_design) [Accessed Feb 15, 2017]
- [10] Wikipedia. "Graphical User Interface" wikipedia.org. [Online].Available:  
[http://en.wikipedia.org/wiki/Graphical\\_user\\_interface](http://en.wikipedia.org/wiki/Graphical_user_interface) [Accessed Feb 15, 2017]
- [11] Tuts+. “MVC” tutsplus.com. [Online]. Available:  
<http://code.tutsplus.com/tutorials/mvc-for-noobs--net-10488> [Accessed Feb 20, 2017]
- [12] Phpeasysteps. "Pagination" phpeasystep.com.[Online]. Available:  
<http://www.phpeasystep.com/phptu/29.html> [Accessed Mar 10, 2017]
- [13] Zymic. “File Upload From with PHP” zymic.com.[Online]. Available:  
<http://www.zymic.com/tutorials/php/creating-a-file-upload-form-with-php/> [Accessed Mar 12, 2017]
- [14] W3 Schools. "W3 Schools online Ajax tutorials" w3schools.com. [Online]. Available:  
[http://www.w3schools.com/ajax/tryit.asp?filename=tryajax\\_first](http://www.w3schools.com/ajax/tryit.asp?filename=tryajax_first) [Accessed Mar 18, 2017]

- [16] Form Validator. "*Form validator*" formvalidator.net. [Online]. Available: <http://www.formvalidator.net/> [Accessed Mar 22, 2017]
- [17] Adam Patterson. "*php user login with sessions*" adampatterson.ca. [Online]. Available: <https://www.adampatterson.ca/2010/php-user-login-with-sessions/> [Accessed Mar 25, 2017]
- [18] Codexworld. "*Autocomplete text box using jquery,php and mysql*" codexworld.com.[Online]. Available: <http://www.codexworld.com/autocomplete-textbox-using-jquery-php-mysql/> [Accessed May 12, 2017]
- [19] Jqplot. "*Bar Chart*".jqplot.com.[Online]. Available: <http://www.jqplot.com/examples/barTest.php#> [Accessed June 12, 2017]
- [20] guru99. "Integration Testing Tutorial".guru99.com. [Online].Available: <https://www.guru99.com/integration-testing.html> [Accessed Feb 12, 2017]
- [21] guru99. "Regression Testing Tutorial".guru99.com. [Online].Available: <https://www.guru99.com/regression-testing.html> [Accessed Feb 12, 2017]
- [22] guru99. "Unit Testing Tutorial".guru99.com. [Online].Available: <https://www.guru99.com/unit-testing.html> [Accessed Feb 12, 2017]

# Appendix A System Documentation

## System Documentation

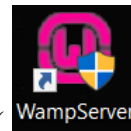
System and the database that needed to implement on computers are written to a DVD. It is important not to change the file structure in DVD. Ensure upload the all files and folders to the server as it is on DVD. DVD contains following files and folders.

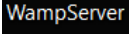
- System folder – This contains all the files that needed to run the system.
- Database file – system database is in this db.txt file.
- Software folder – This folder contains the required software to run the system such as FileZilla FTP, wamp server

## How to Implement the System in Local Host


To implement the system on a local host, following steps needed to be done.

- Install wamp server.




- Run wamp server by clicking on the shortcut (  ) on the desktop.
- Go to C:/wamp/www/ and copy paste the system folder without doing any changes.



- Click on the icon (  ) in the taskbar to go to phpmyadmin.
- Go to phpmyadmin and select the import tab.
- Browse and select the db.txt file.
- Click on GO button on the right side bottom.
- Wait till the pop-up message confirms the importing.



- Click on the icon (  ) in the taskbar to go to local host.
- Select the necessary folder to access and login page will appear.
- Log in to the system through the log in page.

## How to Implement the System on a Web Server

If the system will be implemented on a web server, FTP username and password will be provided by the web host service to the client. Then FileZilla FTP client should be installed in the computer to upload the system folder into the web server.

- Install FileZilla FTP client.
- Open the FileZilla --> click on manage site --> create new site --> type host provided by the web host service --> type port --> select the server type as FTP --> select login type as normal --> type ftp username and password provided by web host service.
- Press the connect button.
- the message will ensure the connection was successful
- Drag and drop the files from A to B that in the system folder without changing the structure of files as mentions below.
- Import the database file to the web server through the control panel or the direct link from the phpmyadmin.

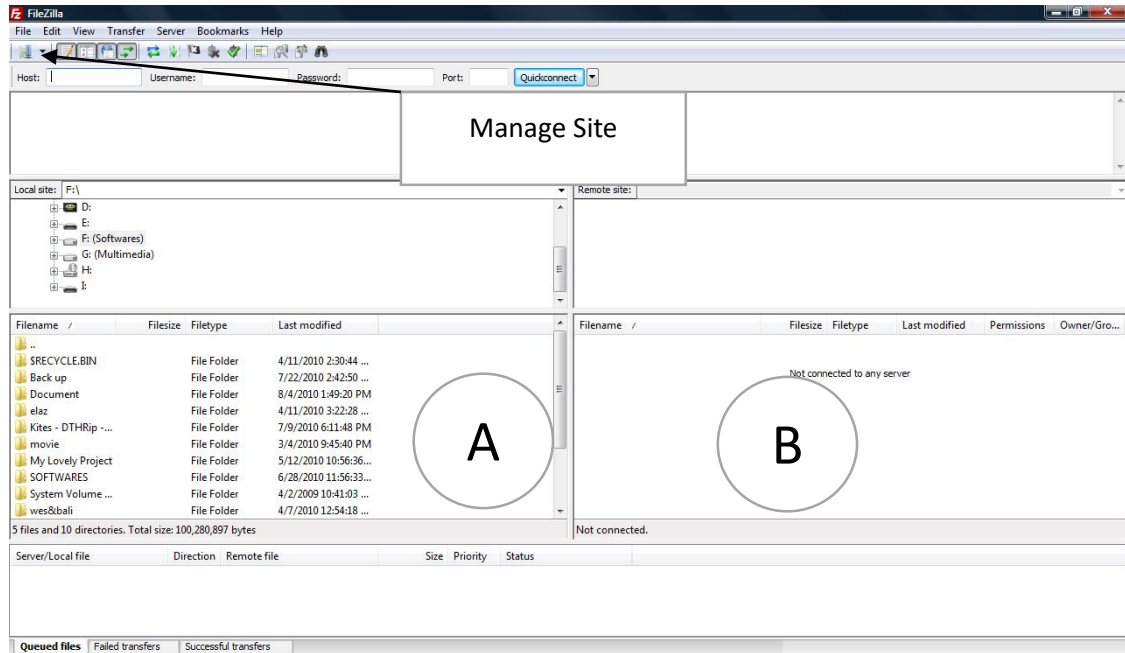


Figure A.1 FileZilla FTP Client Interface

# Appendix B Design Documentation

## Sequence Diagram of Add an Employee

This diagram illustrates the sequence of adding a new employee record to the system. As the first step user need to fill the add employee form with the correct information. If one of the required field not filled in the form validation message will show in the form itself. After clicking submit button controller class of add employee will get the data and pass it to the model class of the Employee. Finally success or fail JavaScript message will display in the Add Employee form.

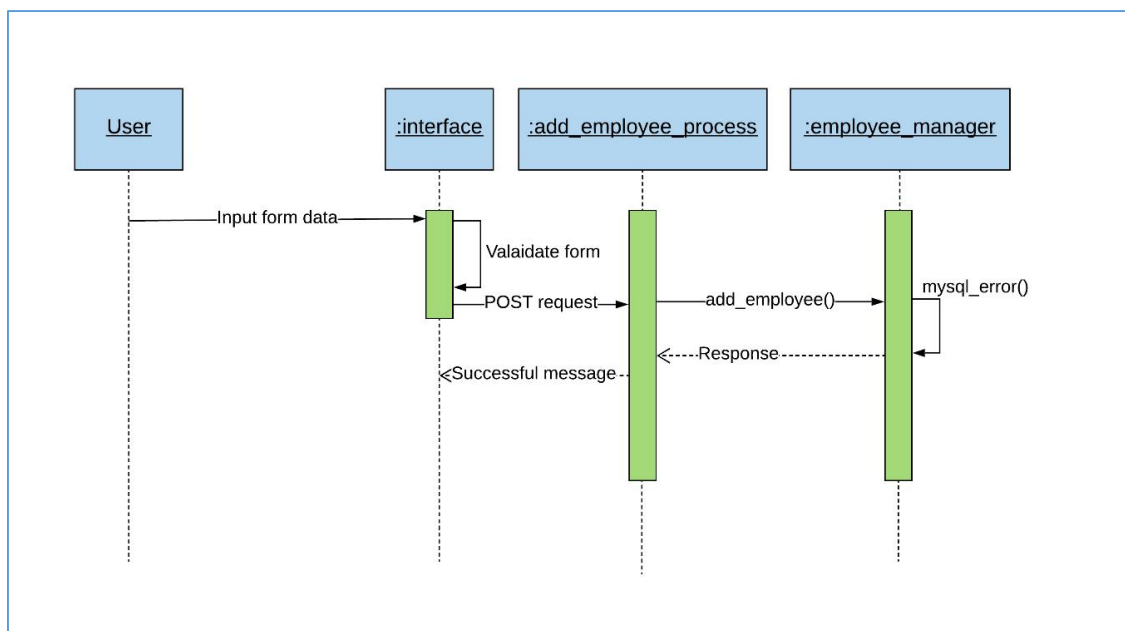


Figure B.1 Sequence Diagram for Add an Employee



## Usecase Diagrams with Detailed Information

### Employee Update Usecase

Following usecase illustrate editing an employee record of the system. This can only be done by a user who has admin privileges.

<b>Use Case</b>	Employee Update
<b>Actors</b>	HR Executive, Manager, Chairman
<b>Overview</b>	
	<ul style="list-style-type: none"><li>• Manage employee details</li></ul>
<b>Preconditions</b>	
	<ul style="list-style-type: none"><li>• User should log in to the system as an admin</li></ul>
<b>Flow of Events</b>	
	<ul style="list-style-type: none"><li>• Go to PIM menu</li><li>• Click Employee List</li><li>• Search for an employee (if the user type is admin only)</li><li>• Click Edit button</li><li>• Change the information</li><li>• Click Submit button</li></ul>
<b>Post Conditions</b>	
	Employee details will change in the DB as well

Table B.1 Employee Update Usecase

## Add a Leave Usecase

Following usecase illustrate adding a leave. This can be done by all user types. The user must fill the add leave form and then can see the leave approve status on the Leave Entitlement page.

<b>Use Case</b>	Add Leave
<b>Actors</b>	HR Executive, Employee, Manager
<b>Overview</b>	
<ul style="list-style-type: none"><li>• Add a leave</li></ul>	
<b>Preconditions</b>	
<ul style="list-style-type: none"><li>• Should login to the system</li></ul>	
<b>Flow of Events</b>	
<ul style="list-style-type: none"><li>• Go to Leave menu</li><li>• Select Add leave</li><li>• Fill the form and submit</li></ul>	
<b>Post Conditions</b>	
Added Leave will list in the Leave Entitlements list and can view if the leave approved or not	

Table B.2 Add Leave Usecase

## Login Usecase

Following usecase illustrate the login usecase. The first page of this system is the login page. Each and every user is given a unique username and password is user defined password. There are two user types in this system and each user type has different module visibility.

<b>Use Case</b>	Login
<b>Actors</b>	HR Executive, Employee, Manager, Chairman
<b>Overview</b>	
	Register user can log in to the system
<b>Preconditions</b>	
	User must have valid username and password
<b>Flow of Events</b>	
	<ul style="list-style-type: none"><li>• Type user name</li><li>• Type password</li><li>• Press login button</li></ul>
<b>Post Conditions</b>	
	User redirect to the home page

Table B.3 Login Usecase

## Get Approval Usecase

Following usecase illustrate the get approval usecase. Only the admin users can approve a leave. Admin users can see all the applied leaves in Leave Entitlement table and can decide to approve or not to approve. Each and every employee can see the leave approve status in the Leave Entitlement table.

<b>Use Case</b>	Get Approval
<b>Actors</b>	HR Executive, Manager, Chairman
<b>Overview</b>	
Approve / Reject requests	
<b>Preconditions</b>	
<ul style="list-style-type: none"><li>• Should log in as the admin user</li></ul>	
<b>Flow of Events</b>	
<ul style="list-style-type: none"><li>• Admin Logins</li><li>• Go to Leave Entitlement</li><li>• Click View button</li><li>• Give the approval</li></ul>	
<b>Post Conditions</b>	
-	

Table B.4 Get Approval Usecase

## Add Job Title Usecase

Following usecase illustrate adding a job title to the system. After adding a job title to the system, any user can vacancies under that job title. Added new job title will show on the candidate page and the job title combo box in add vacancy form.

<b>Use Case</b>	Add Job Title
<b>Actors</b>	HR Executive, Manager, Chairman
<b>Overview</b>	
<ul style="list-style-type: none"><li>• Manage Designations</li></ul>	
<b>Preconditions</b>	
<ul style="list-style-type: none"><li>• Logged in as Admin user</li></ul>	
<b>Flow of Events</b>	
<ul style="list-style-type: none"><li>• Go to Admin menu</li><li>• Go to Add job title menu</li><li>• Click add job title button</li><li>• Fill the form</li></ul>	
<b>Post Conditions</b>	
New job title added to the DB,	

Table B.5 Add Designation Usecase

## Set Leaves Usecase

Following usecase illustrate the set leaves function. Admin user can set Annual, Medical, Casual leaves of a particular employee manually. This will be very useful for an employee who is in the probation period.

<b>Use Case</b>	Set Leaves
<b>Actors</b>	HR Executive, Manager, Chairman
<b>Overview</b>	
	<ul style="list-style-type: none"><li>• Set leaves/manage leave of individual employees</li></ul>
<b>Preconditions</b>	
	<ul style="list-style-type: none"><li>• Should log in to the system as admin</li></ul>
<b>Flow of Events</b>	
	<ul style="list-style-type: none"><li>• Go to Leave menu</li><li>• Select set Leaves</li><li>• Search for an employee (auto-fill the name while typing)</li><li>• Change the leaves in the form</li><li>• Click submit</li></ul>
<b>Post Conditions</b>	
	Added designation in the add, edit candidate details form

Table B.6 Set Leaves Usecase

## Delete a System User Usecase

Following usecase illustrate deleting a system user of the system. Only the admin users have this privilege in the system to delete system user. After deleting a particular user, that user won't be able to login to the system using the previous user credentials. To log in to the system again new user profile should be created.

<b>Use Case</b>	Delete a system user
<b>Actors</b>	HR Executive, Manager, Chairman
<b>Overview</b>	
<ul style="list-style-type: none"><li>• Delete system user</li></ul>	
<b>Preconditions</b>	
<ul style="list-style-type: none"><li>• Should log in as admin user</li></ul>	
<b>Flow of Events</b>	
<ul style="list-style-type: none"><li>• Go to Admin menu</li><li>• Select User List submenu</li><li>• Search for a user</li><li>• Click Delete button</li><li>• Click Yes in the confirmation box</li></ul>	
<b>Post Conditions</b>	
System user will not show in the table and deleted from the database as well	

Table B.7 Usecase for Deleting a System User

# Appendix C User Documentation

## Introduction

Employee Management System for Rising Logistics (Pvt) Ltd is developed into three Main Categories

1. Candidate Module
2. Leave Module
3. Employee Module

According to the user-privileges of the user who is log in to the system is redirected to the Homepage. For instance when administrator login to the system he is redirected to the Vacancies page (Home Page) and no other user is capable of seeing the home page.

This user documentation is not willing to focus on each and every user as well as each relative module. This user documentation provides an overview impression of how to use the main functions in the system.

The main expectation of the user documentation is to guide the user independently without providing a comprehensive training.

## Candidate Module

After the login screen, this interface is the very first one that user will encounter. Administrator or user can search vacancy information with some option such as Job title.

- Go to login->type username and password -> click on login button

After Clicking or selecting a desired job title user will redirect into the candidate detail table (see figure C.1) of the selected job title.



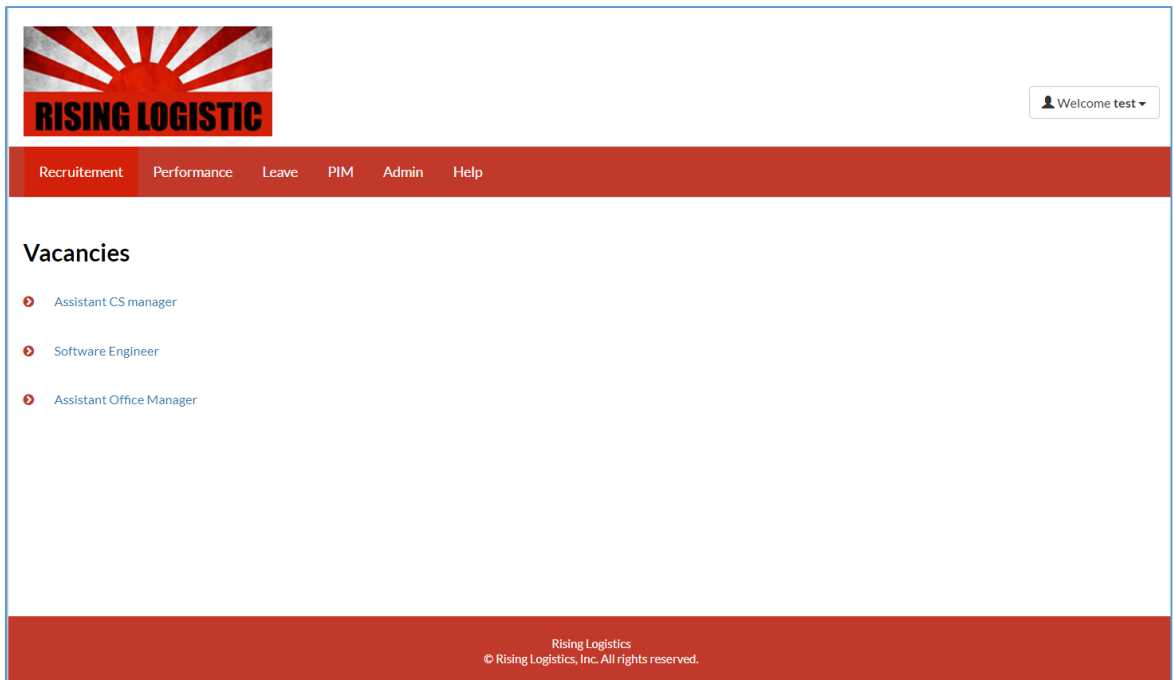


Figure C.1 Vacancies Main Interface

Using that table user can short list a candidate into three sections which are shortlisted, Considerable, Rejected by clicking the view button (see figure C.2).

- Click on a job title -> click on a name under a job title-> click on an option from selected, considerable, rejected

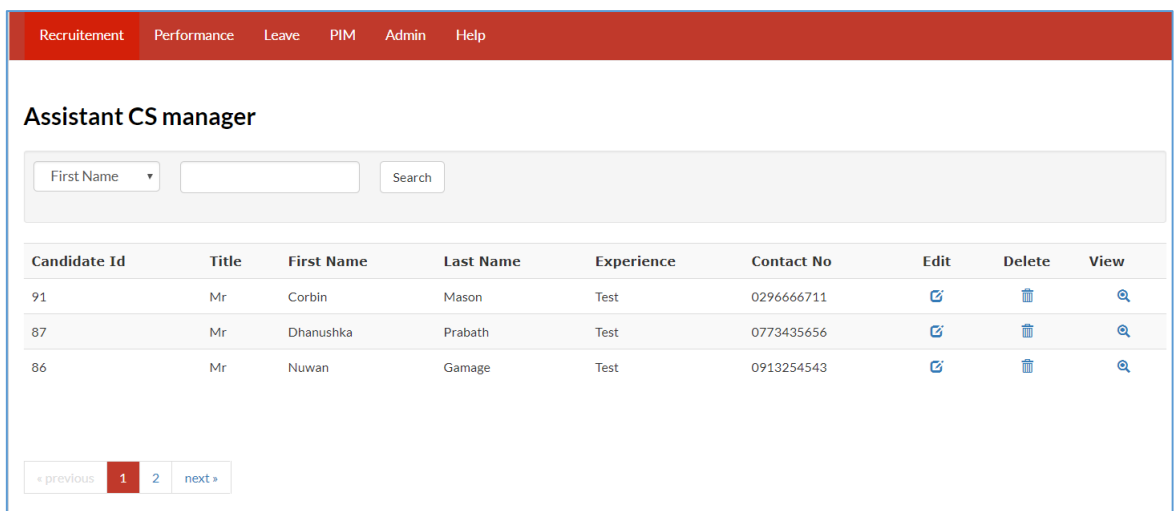


Figure C.2 Job Description Panel

**RISING LOGISTIC**

Welcome test ▾

Recruitment Performance Leave PIM Admin Help

### Candidate Details

Job Title	Assistant CS manager	Job Code	101
First Name*	yauwana	Last Name*	ravindra
Email*	dfgdfgd@dfd.com	Contact Number*	712419767
Expected Salary*	4663	Experience	
Download CV	Download	Other Details	
Resume	<input checked="" type="radio"/> Short Listed <input type="radio"/> Considerable <input type="radio"/> Rejected		

Figure C.3 Candidate Detail Panel

After clicking an option out of rejected, considerable, selected from the Candidate detail (see figure C.3) that record goes to the Candidate part in Recruitment (see figure C.4). From there the user can select the candidates under the selected option and furthermore those records are categorized under the same job title when they are in the vacancy part (see figure C.5).

- Click candidate from recruitment menu -> click a selected type -> click a job title -> click on a name as same as vacancy

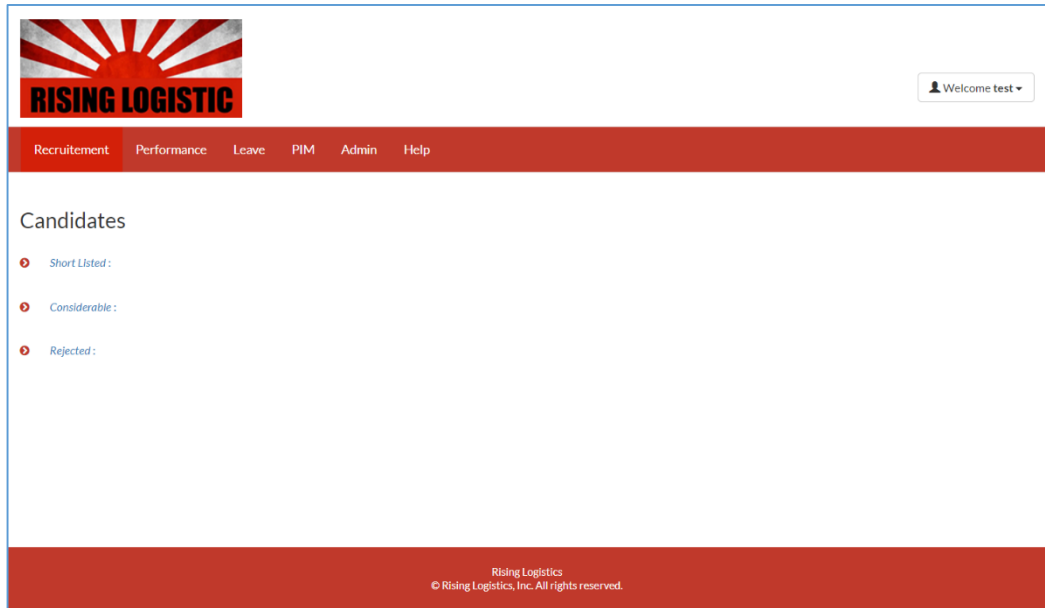


Figure C.4 Candidates Page

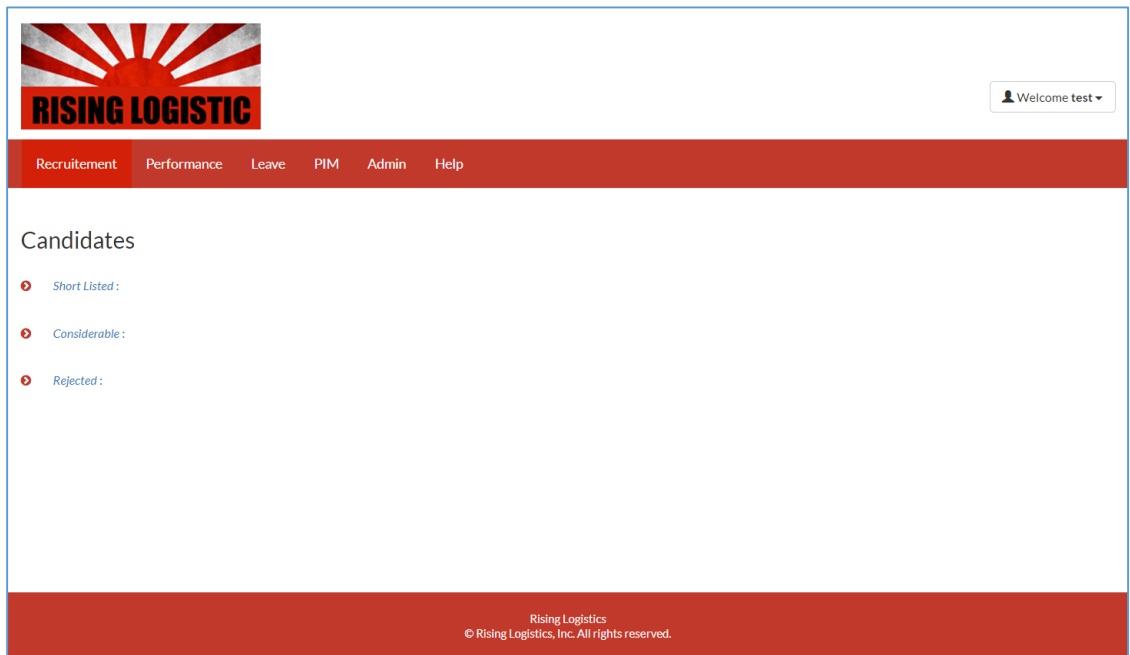


Figure C.5 Short Listed Candidates

## Performance Module

This will help HR Manager and other line managers to add KPIs and reviews about the employees. These KPI marks will be helpful in the annual performance evaluation of the employees. Added KPIs will list in a table as vacancy module and to give marks user must click the give marks icon in the KPI listing table.

- Go to performance menu -> click add KPI

**RISING LOGISTIC**

Welcome test

Recruitment Performance Leave PIM Admin Help

### Add KPI

Employee First Name\*  Employee Last Name\*

Job Title\*  date\* 2017-11-07

KPI 01\*  KPI 02\*

KPI 03\*  KPI given by Laxman Hettiarachchi

\*Required Fields

Rising Logistics  
© Rising Logistics, Inc. All rights reserved.

Figure C.6 KPI Setup Interface

Any user type can add reviews of other employees using the add review function. After adding a review it will create a record in the review table. Admin users can see all the review but the employee users can see a review which relates them only.

- Go to performance menu -> click add review

**RISING LOGISTIC**

Welcome test

Recruitment Performance Leave PIM Admin Help

### Add Review

Employee Full Name\*  Reviewer Full Name\* Laxman Hettiarachchi

Reviewer Job Title\* Software Engineer Date 2017-11-07

Review\*

Rising Logistics  
© Rising Logistics, Inc. All rights reserved.

Figure C.7 Add Review Interface

## Leave Module

This module consist of Add Leave, Leave Entitlement, Leave Summary, Set leaves submodules. Every user can see the add leave form but Leave Entitlement and Leave Summary is not the same. Admin Employees can see all the applied leaves on Leave Entitlement table and employee users can only see their own leave entitlements.

Add leave form is very simple and user-friendly for the user. If the user put a number of leaves more than user's quota user will get a warning message indicating the user has not enough leaves to apply.

- Go to Leave menu -> click add leave

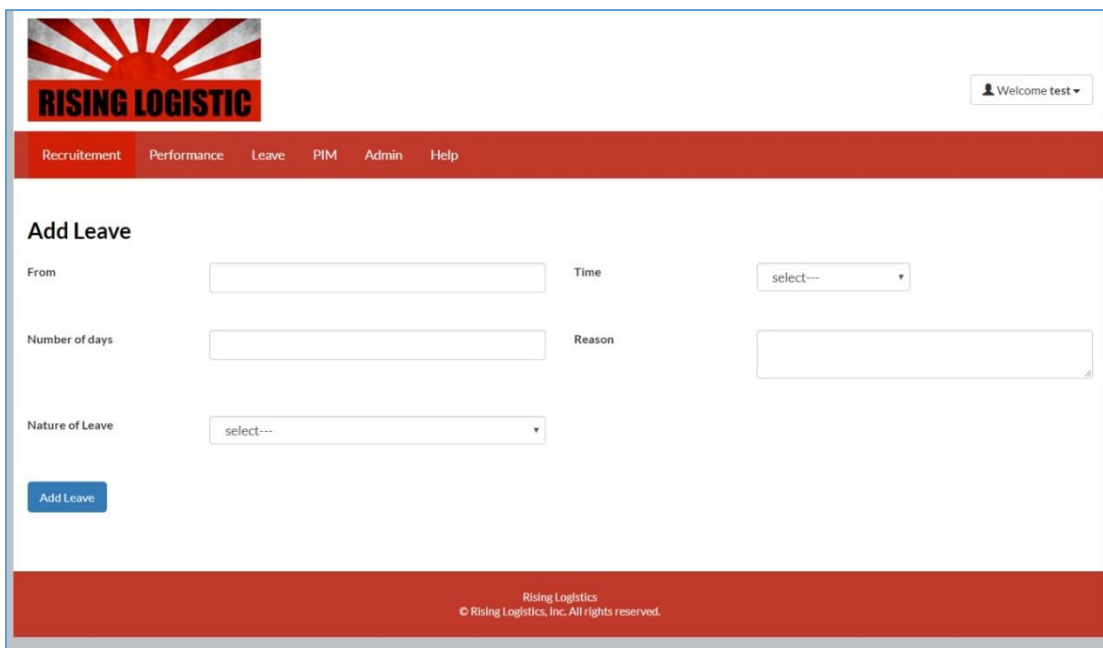


Figure C.8 Add Leave Form

In the Leave Entitlement table, admin users can see all the leaves applied in descending order. If user wants to apply a leave user should go to the Leave entitle page and select a particular leave which an employee has put by searching or just going through pagination. The user must click on the view icon to approve a leave. After approving a leave approve status and the person who approved the leave will show in the leave entitlement table (see figure C.9).

- Go to Leave menu -> click Leave Entitlement -> Click view icon -> set leave status

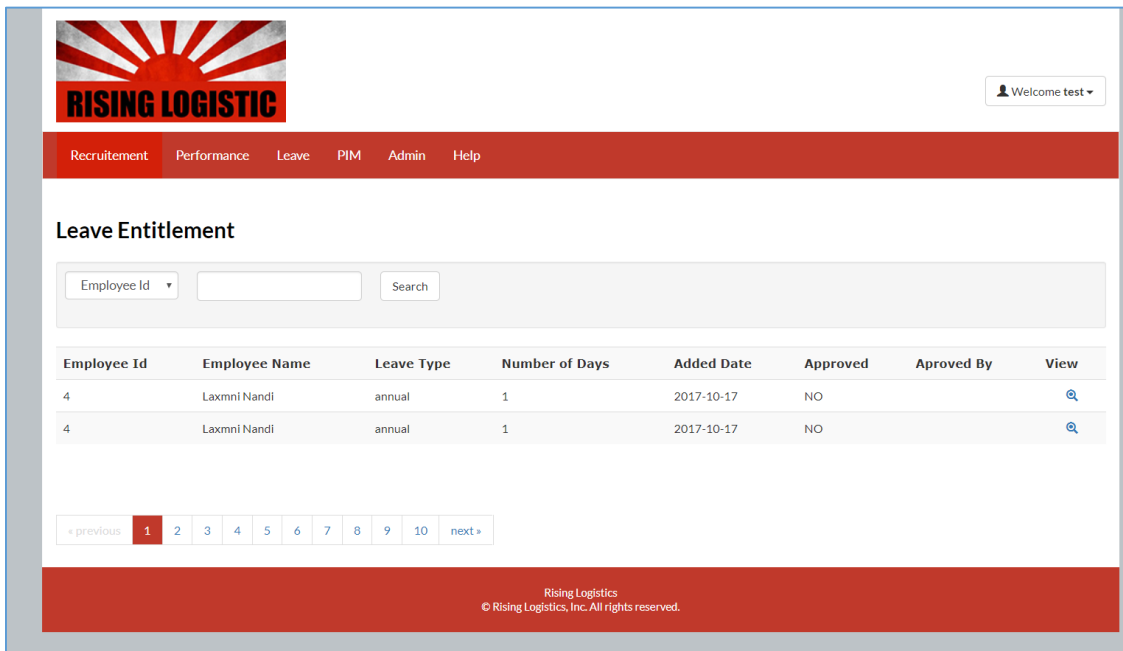


Figure C.9 Leave Entitlement Form

In the leave summary page, admin user can see all the employee's leave summary categorize into Annual, Casual and Medical. The user can see the remaining leaves as such as the total leaves user has in each leave category. (See figure C.10)

- Go to Leave menu -> click Leave Summary

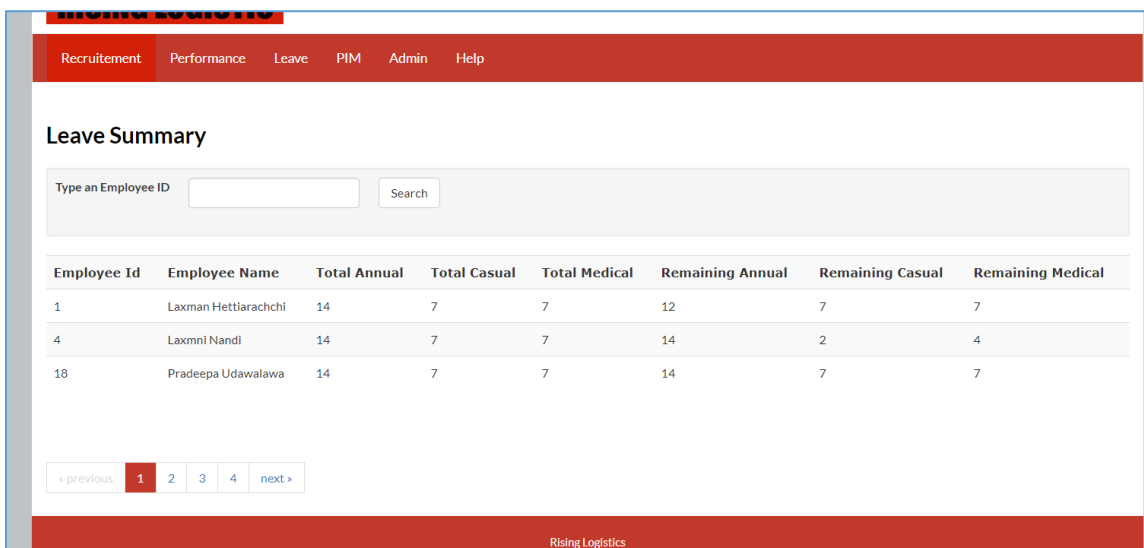
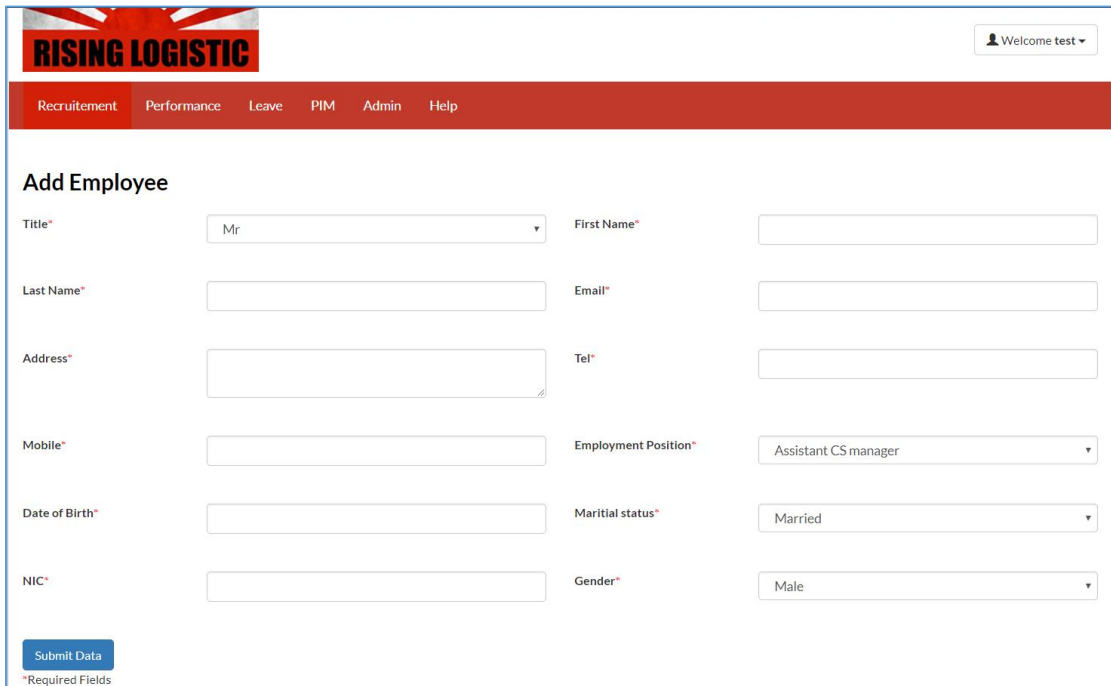


Figure C.10 Leave Summary Table

## PIM (personal information manager)

This will help to add employees (see figure C.11) to the system and will use when tracking an employee. All the details in the add employee form can see in the employee list table by clicking the view icon (see figure C.12). The user can also edit or delete an employee record as well.

- Go to PIM menu -> click add employee



**RISING LOGISTIC** Welcome test

Recruitment Performance Leave PIM Admin Help

### Add Employee

Title\*  First Name\*

Last Name\*  Email\*

Address\*  Tel\*

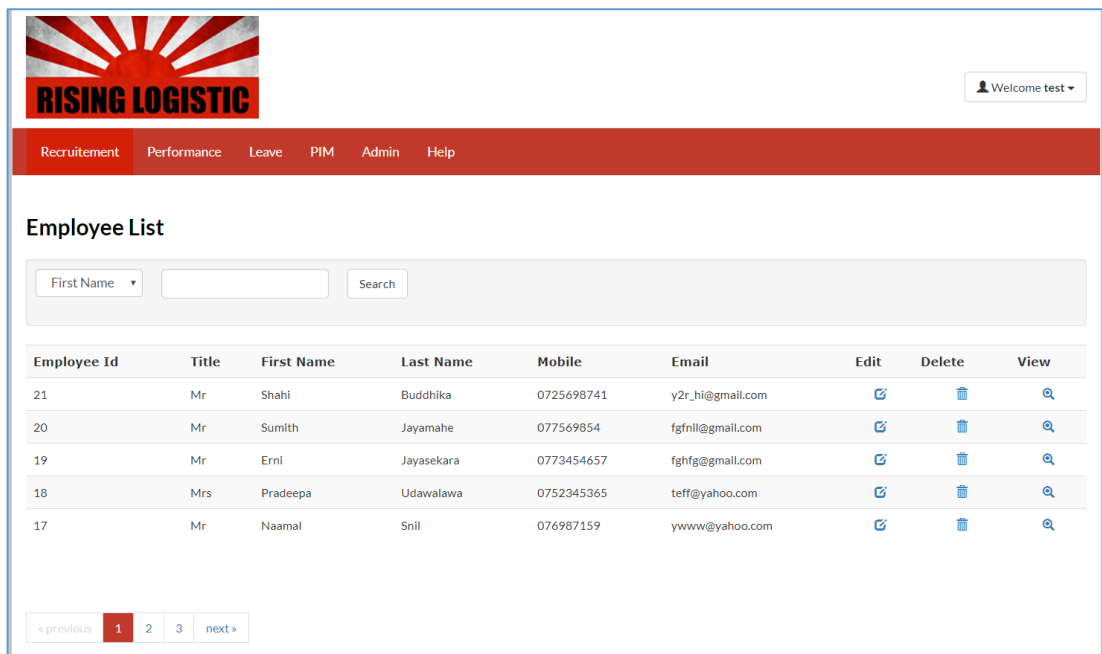
Mobile\*  Employment Position\*

Date of Birth\*  Marital status\*

NIC\*  Gender\*

\*Required Fields

Figure C.11 Employee List Table



**RISING LOGISTIC** Welcome test

Recruitment Performance Leave PIM Admin Help

### Employee List

First Name  Search

Employee Id	Title	First Name	Last Name	Mobile	Email	Edit	Delete	View
21	Mr	Shahi	Buddhika	0725698741	y2r_hi@gmail.com			
20	Mr	Sumith	Jayamahe	077569854	fgfnll@gmail.com			
19	Mr	Erni	Jayasekara	0773454657	fgfhfg@gmail.com			
18	Mrs	Pradeepa	Udawalawa	0752345365	teff@yahoo.com			
17	Mr	Naamal	Snil	076987159	ywww@yahoo.com			

< previous 1 2 3 next >

Figure C.12 Interface for Adding Employees

# Appendix D Management Reports

## Applied Vacancies according to the Job title

The following report shows the number of candidates apply against the months of the year. A number of candidates will show on the Y-axis and month's names will show on the X-axis. Before coming to this bar chart there is another interface to select a year.



Figure D.1 Applied Candidates per Year



## Leave Applied to Employees per month

This report will show the records of the employees who have applied a particular leave type for a selected month in a selected year. The first user has to select the Year, Month and Leave type and click “Submit Data” button (see figure D.2). This will help the management to identify the employees who have taken the most leaves for a certain month in a selected year (see figure D.3).


annual Leave Applied by Employees per Month July

Employee First Name	Employee Last Name	Leave added date	From date	Number of days
Laxmi	Nandi	2017-07-12	2017-07-18	3
Hasitha	Kapuduwa	2017-07-21	2017-07-27	3
Mali	Karanaa	2017-07-21	2017-07-25	3
Laxman	Hettiarachchi	2017-07-22	2017-07-26	3
Nimal	Kariyawasam	2017-07-22	2017-07-25	2
Laxman	Hettiarachchi	2017-07-22	2017-07-27	3
Yauwana	Ravindra	2017-07-27	2017-07-27	3

[Go Back](#)

Rising Logistics  
© Rising Logistics, Inc. All rights reserved.

Figure D.3 List of Leave Filtered by the Month



Welcome test

Recruitment Performance Leave PIM Admin Help

### Leave Applied by Employees per Month

Select a year: 2016

Select a Month: 8

Select a Leave Type: Casual

[Submit Data](#)

Rising Logistics  
© Rising Logistics, Inc. All rights reserved.

Figure D.2 Filtering Leaves for a Month

## Best 10 KPI marks per year

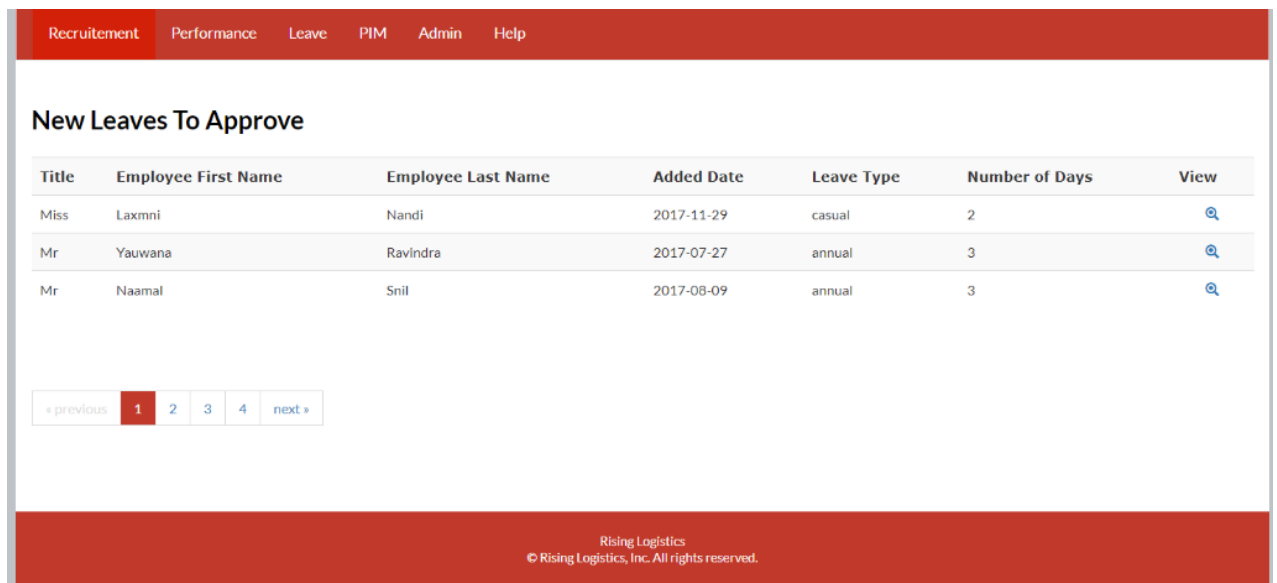
This report shows the Best ten employees who have taken the marks out of 30 per year. Marks shows on the Y-axis and employee first name and last name shows on the X-axis. Bars of this bar chart is in ascending order.



Figure D.4 Best 10 KPI Marks per Year

## New Leaves to Approve

This report extracts the leaves which not approved by any admin user and will organize in a table in descending order. So the user can see the recently applied leaves at once and can approve or not approve any leaves. This will ease the day to day work of the management since these reports can see only the admin users.



Title	Employee First Name	Employee Last Name	Added Date	Leave Type	Number of Days	View
Miss	Laxmi	Nandi	2017-11-29	casual	2	<a href="#">View</a>
Mr	Yauwana	Ravindra	2017-07-27	annual	3	<a href="#">View</a>
Mr	Naamal	Snit	2017-08-09	annual	3	<a href="#">View</a>

« previous 1 2 3 4 next »

Rising Logistics  
© Rising Logistics, Inc. All rights reserved.

Figure D.5 Leaves Pending Approval

# Appendix E Test Results

Test cases which are critical and high priority test cases are mention below. These test cases did not mention in the Evaluation chapter. A complete test result table will be available in the CD-ROM for all the other modules.

<b>objective</b>	<b>input</b>	<b>Expected output</b>	<b>Actual output</b>
Check the username already in the system	Give an existing username	Username already exist	Username already Exist
Check the system user created successfully	Fill the user add form and submit	THANK YOU. YOUR DATA HAS BEEN SUBMITTED	THANK YOU. YOUR DATA HAS BEEN SUBMITTED
Check employee id already have a username	Give an existing employee id that has a username	This employee already have a username	This employee already have a username
Check the search function based on employee id	Yauwana	No result to display	No result to display
Check the search function based on employee id	21	One record with employee id equal to 21	One record with employee id equal to 21
Check the search function based on a first name	s	Five results with the the first name has 's'	Five results with the the first name has 'd'
Check the search function based on last name	ravi	One results with last a name with 'Ravi'	One results with last a name with 'ravi'

Check the search function based on employee id in employee record	21	One results with employee id equal to 21	One results with employee id equal to 21
Check the search function based on First Name	Hari	A KPI display no equal to 'Hari'	A KPI display no equal to 'Hari'
Check the search function based on KPI Date	2017-11	Many items display which KPI given date equal to 2017-11	Many items display which KPI given date equal to 2017-11
Check the search function based on username	nimal	No record will display which username equal to nimal	No record will display which username equal to nimal
Check the search function based on employee division	admin	Record will display which employee division equal to admin	Record will display which employee division equal to admin
Check the search field the text box	Nothing and press search button	This field is required	This field is required
Check the password retype	Type invalid retype password	Input values could not be confirmed	Input values could not be confirmed
Check the password minimum length	Type a password less than five characters	This must be at least five	This must be at least Five
Check the edit icon works	Click on edit icon	Edit page should display with current information	Edit page should display with current information

Check the view icon works	Click on view icon	View page should display in a window with a view template	View page should display in a window with a view template
---------------------------	--------------------	---	---

Table E.1 Test Results

# Appendix F Code Listing

Following code represents the functions and particular classes.

## Adding vacancy Details

```
<?php
$baseurl = "http://localhost/yauwana-fyp-hrm/trunk";
// Configuration - Your Options
$allowed_filetypes = array (".pdf",".doc",".docx"); // These will be
the types of file that will pass the validation.
$max_filesize = 2097152; // Maximum filesize in BYTES (currently
2MB).
$upload_path = '../..cv/'; // The place the files will be uploaded
to (currently a 'files' directory).
$filename = mysql_real_escape_string ( $_FILES ['userfile'] ['name']
); // Get the name of the file (including file extension).
$ext = substr ( $filename, strpos ( $filename, '.' ), strlen (
$filename ) - 1 ); // Get the extension from the filename.
var_dump($allowed_filetypes) ;
    $size=(filesize ( $_FILES ['userfile'] ['tmp_name'] ));
    var_dump($size);
    if ($size != false) {
        // Check if the filetype is allowed, if not DIE and inform the
user.
        if (! in_array ( $ext, $allowed_filetypes )) {
            header ( "Location:../..templatel.php?tab=add_vac&err=true"
);
            $pass = 0;
            echo $pass;
            // Now check the filesize, if it is too large then DIE and
inform the user.
        } elseif (filesize ( $_FILES ['userfile'] ['tmp_name'] ) >
$max_filesize) {
            header (
"Location:../..templatel.php?tab=add_vac&err=true1" );
            $pass = 1;
            // Check if we can upload to the specified path, if not DIE
and inform the user.
        } elseif (! is_writable ( $upload_path )) {
            header (
"Location:../..templatel.php?tab=add_vac&err=true3" );
            $pass = 3
            // Upload the file to your specified path.
        } elseif (move_uploaded_file ( $_FILES ['userfile']
['tmp_name'], $upload_path . $filename )) {
            $pass = 2;
        } else
            echo 'There was an error during the file upload. Please try
again.'; // It failed :(.)
    }else{
        $pass = 4;
    }
    if($pass == 4){
        $path = NULL;
    }
    else {
```

```

$spath = $baseurl . "/cv/" . $filename;
}
echo $spath;
include ('../../includes/vacancy_manager.php');
include ('../../includes/database.php');
$database->getConnection ();
$email = $_POST ['email']; //Email
$experience = $_POST ['exp']; //experience
$name = $_POST ['fname']; //First name
$number = $_POST ['num']; // Number
$salary = $_POST ['sal']; //salary
$other = $_POST ['other_details']; //comment
$lname = $_POST ['lname'];
$job = $_POST ['com'];
$title = $_POST ['title'];
$date = date ( "Y" . "-" . "m" . "-" . "d" );
$x = "NO";
if ($pass == 2 || $pass == 4) {
    $add = new vacancy ();
    $add->addVac ( $title, $name, $lname, $other, $salary, $number,
    $email, $experience, $job, $x, $spath, $date );
    header ( "Location:../../templatel.php?tab=add_vac&err=true2" );
}
?>

```

## Add Employee Function

```

<?php
include ('../../includes/employee_manager.php');
include ('../../includes/database.php');
$database->getConnection();
$title = $_POST ['com']; //employee title
$fname = $_POST ['fname']; // first name
$lname = $_POST ['lname']; // last name
$email = $_POST ['email']; //email
$address = $_POST ['address']; // address
$tel = $_POST ['tel']; // land line number
$mobile = $_POST ['mobile']; // mobile number
$user = $_POST ['user']; // position
$dob = $_POST ['dob']; //date of birth
$m_status = $_POST ['m_status']; //marital status
$nic = $_POST ['nic']; // NIC number
$gender = $_POST ['gender']; //gender
$employee = new employee(); // cerate employee object
$val = $employee->addemployee($title, $fname, $lname, $email,
$address, $tel, $mobile, $user, $dob, $m_status, $nic, $gender);
//insert in to the employee table
if ($val[0] == 1)
{
    $employee->leave_base($val[1], 14, 7, 7, 14, 7, 7); //set leave
in leave base table while creating an employee

header ("Location:../../templatel.php?tab=add_employee&err=true2");
}
else
{
header ("Location:../../templatel.php?tab=add_employee&err=false");
}

```



```
}  
?>
```

## Add KPI Records Function

```
<?php  
include ('../../includes/KPI_manager.php');  
include ('../../includes/database.php');  
$database->getConnection ();  
// get all the values coming from the URL  
if (isset ( $_GET ['x'] )) {  
    $emp_id = $_GET ['x'];  
}  
if (isset ( $_GET ['p'] )) {  
    $page = $_GET ['p'];  
}  
if (isset ( $_GET ['val'] )) {  
    $x = $_GET ['val'];  
}  
$f_name = $_POST ['fname']; // first name  
$l_name = $_POST ['lname']; //last name  
$j_title = $_POST ['com']; // job title  
$date = $_POST ['date']; // KPI given date  
$kpi1 = $_POST ['kpi1']; //KPI 1  
$kpi2 = $_POST ['kpi2']; //KPI2  
$kpi3 = $_POST ['kpi3']; //KPI 3  
$given = $_POST ['given']; //KPI given by  
$emp_id = $_POST ['emp_id'];  
$addkpi = new KPI (); //create new KPI object  
$addkpi->addKPI ( $f_name, $l_name, $j_title, $date, $kpi1, $kpi2,  
$kpi3, $given,$emp_id ); //call to add KPI mehod  
header ( "Location:../../templatel.php?tab=add_kpi&err=true2" );  
?>
```

## Suggestion Box Function

```
<?php  
require_once '../../includes/database.php';  
$term = $_GET ["term"]; //search term user type in the textbox  
$database->getConnection (); //data base conection  
  
//search for the relevant user from the DB  
$query = mysql_query ("SELECT * FROM employee WHERE e_f_name LIKE  
'%".$term."%'ORDER BY e_f_name ASC" ) or die ( mysql_error ( ) );  
  
$json = array (); // crete an array  
  
while ( $student = mysql_fetch_array ( $query ) ) { //put data base  
values to the array  
    $json [] = array (  
        'value' => $student ["e_f_name"],  
        'label' => $student ["e_f_name"] . " " . $student  
["e_l_name"] . " - " . $student ["emp_id"],  
        'number' => $student ["e_l_name"],  
        'id' => $student ["emp_id"],  
        'job' => $student ["vacancy"]  
    );  
}
```

```

    );
}
echo json_encode ( $json );//encode that array to a json object
?>

```

## Vacancy Manager Class

```

<?php
class vacancy {
    public function addEmp($title, $fname, $lname, $divi, $address,
    $stel, $email) {
        $query = mysql_query ( "INSERT INTO employee
    (emp_f_name,emp_l_name,title,emp_add,emp_tel,email,emp_division)VALU
    ES(
        '$fname','$lname','$title','$address ','$stel','$email','$divi') "
    ) or die ( mysql_error ( ) );
    }
    public function editVac($email, $experience, $name, $number,
    $salary, $other, $resume, $lname, $emp_id) {
        $query = mysql_query ( "UPDATE candidate SET f_name='$name',
    l_name='$lname',comment='$other',expected_sal='$salary',contact_no='
    $number'
    ,email='$email',status='$resume',experience='$experience'
        WHERE candidate_no='$emp_id'" ) or die ( mysql_error
    ( ) );
    }
    public function searchItem($id, $pa_id, $tbl_name, $start,
    $limit) {
        //search with in the search page
        if ($pa_id == "f_name") {
            $sql = mysql_query ( "SELECT * FROM $tbl_name WHERE
    f_name LIKE '%$id%' AND status='NO' LIMIT $start, $limit" ) or die (
    mysql_error ( ) );
        } else if ($pa_id == "l_name") {
            $sql = mysql_query ( "SELECT * FROM $tbl_name WHERE
    l_name LIKE '%$id%' AND status='NO' LIMIT $start, $limit" ) or die (
    mysql_error ( ) );
        } else if ($pa_id == "emp_id") {
            $sql = mysql_query ( "SELECT * FROM $tbl_name WHERE
    candidate_no LIKE '%$id%' AND status='NO' LIMIT $start, $limit" ) or
    die ( mysql_error ( ) );
        } else {
            echo "<p>Please enter a search query</p>";
        }
        return $sql;
    }
    public function getSearch($id, $pa_id, $tbl_name) {
        //search query according to the search term
        if ($pa_id == "f_name") {
            $sql = mysql_query ( "SELECT * FROM $tbl_name WHERE
    f_name LIKE '%$id%' AND status='NO'" ) or die ( mysql_error ( ) );
        } else if ($pa_id == "l_name") {
            $sql = mysql_query ( "SELECT * FROM $tbl_name WHERE
    l_name LIKE '%$id%' AND status='NO'" ) or die ( mysql_error ( ) );
        } else if ($pa_id == "emp_id") {

```

```

        $sql = mysql_query ( "SELECT * FROM $tbl_name WHERE
candidate_no LIKE '%$id%' AND status='NO'" ) or die ( mysql_error ( )
);
    } else {
        echo "<p>Please enter a search query</p>";
    }
    return $sql;
}
public function deleteItem($s) {
    $quer = mysql_query ( "DELETE FROM candidate WHERE
candidate_no= $s" ) or die ( mysql_error ( ) );
}
public function addVac($title, $name, $lname, $other, $salary,
$number, $email, $experience, $job, $x, $path, $date) {
    $query = mysql_query ( "INSERT INTO
candidate(title,f_name,l_name,date,comment,expected_sal,contact_no,e
mail,experience,vac_code,status,path)VALUES (
'$title', '$name', '$lname', '$date', '$other', '$salary', '$number', '$ema
il', '$experience', '$job', '$x', '$path')" ) or die ( mysql_error ( ) );
}
public function lastPageRec($total_pages,$limit){
    $floor_values = floor($total_pages/ $limit);
    $complete_records = $limit*$floor_values;
    $last_page_records= $total_pages - $complete_records;
    return $last_page_records;
}
} // end of class
?>

```

## Ajax Delete Function

```

<!-- delete ajax function begin -->
<script type="text/javascript">
function showUser(p)
{
if (window.XMLHttpRequest)
    {// code for IE7+, Firefox, Chrome, Opera, Safari
    xmlhttp=new XMLHttpRequest();
    }
else
    {// code for IE6, IE5
    xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }

xmlhttp.onreadystatechange=function()
{
if (xmlhttp.readyState==4 && xmlhttp.status==200)
    {
        //delete table row
        var tr = document.getElementById(p).parentNode.parentNode;
        tr.parentNode.removeChild(tr);
        location.reload();
    }
}
var full = "?s="+p;

```

```
xmlhttp.open("GET","menu/employee  
list/employee_delete.php"+full,true);
```

# Appendix G Client Certificate



Figure G.1 Client Certification

# Glossary