

EMPLOYEE MANAGEMENT SYSTEM

BY

PADUA . B. GLORIA

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

HONOURS IN COMPUTER SCIENCE

UNIVERSITY OF THE WESTERN CAPE



UNIVERSITY *of the*
WESTERN CAPE

DATE FRIDAY, 14 SEPTEMBER 2012

Abstract

EMPLOYEE MANAGEMENT SYSTEM

By Padua Gloria

Supervisor: I.M Venter

Our project is based on problems faced by a manager of an automobile workshop in Rwanda. The manager faces problems such as:

- Knowing how many employees showed up to work on a specific day.
- Knowing who worked on a specific car.
- How long they took to complete their task.
- Which employee is free to take up a new task?

In order for these problems to be solved, all employees will be registered on database the time they clocked into work, which cars they are working on and how long they took to work on it will be recorded. The manager will be able to allocate jobs to the employees.

The system is web enabled and it will be possible to access it from anywhere.

LIST OF FIGURES	iv
LIST OF TABLES	v
ACKNOWLEDGMENTS	vi
GLOSSARY	vii
CHAPTER 1	1
The user requirements document.....	1
INTRODUCTION	1
DATA COLLECTION	1
PROBLEM DOMAIN	1
Chapter 2	3
requirements ANALYSIS DOCUMENT	3
REFERENCES	23
bibliography	24
QUESTIONNAIRE	2
Index	3

LIST OF FIGURES

Number

Page

LIST OF TABLES

Number

Page

ACKNOWLEDGMENTS

GLOSSARY

PHP(Hypertext pre-processor): General purpose scripting language that is used for web development and is embedded into HTML.

UML (Unified modelling language):standardized general-purpose modelling language in the field of object-oriented software engineering

WWW (World Wide Web): a system of interlinked hypertext documents accessed via the Internet.

HTTP (Hypertext transfer protocol):Set of rules for transferring files on World Wide Web.

MySQL (My structured Query language): relational database management system that runs as a server providing multi-user access to number of databases.

CHAPTER 1

THE USER REQUIREMENTS DOCUMENT

INTRODUCTION

Employees are the backbone of any company, management of employee performance plays a major role in deciding the success of the organization. The workshop is situated in Rwanda has a problem in management of employee performance. The current system running in the workshop is paper based. That is the workshop is still using cabinet files to store records of stock and employee information. Useful data is scattered all over the place. In this chapter we shall discuss the solutions to the problems being caused by the current system. We shall try to understand the manager's expectations of the new system we are to develop for him.

DATA COLLECTION

A questionnaire in the form of a survey was used to gather information. Why a survey??The database program proposed for this project is based on an automobile repair shop in Rwanda. So similar businesses' in South Africa and were surveyed to find out how they managed their employees. I asked them to respond to a few questions.Their responses to the questions are summarised in Appendix A. These were used to compare with the responses from the workshop in Rwanda.

PROBLEM DOMAIN

The problem domain is the computerisation of an employee management system that can be accessed online by the administrator and the stakeholders such as the manager.

THE CURRENT SYSTEM

People have different personalities and work ethics .So in order to manage their work efficiently and fairly, there has to be a system in place to allocate tasks to different workers. Currently a manual system is used in the Rwandan business that will provide most of the requirements for this project. The system used in Rwanda is based on "TRUST" the employer trusts the employees. Although noble the manager has little control over his business. In a manual system data is stored in a cabinet. Files are thus often misplaced or lost. And at times is difficult to find relevant files. Records for stocks are also not always filed correctly and thus information is not centralised and not easily accessible.

EXPECTATIONS FOR THE NEW SYSTEM:

The system should be:

- Secure.

- The system should be able to provide a list of the employees, the times they worked, the tasks they have been doing etc.
- The system must be able to list what is currently in stock.
- The system must provide easy access to employees' details (name, id number, employee number, address).

CONCLUSION

This chapter dealt with the user's needs and expectations for the new system. The next chapter deals with requirements analysis that is requirements needed to solve the problem being faced by the workshop in Rwanda. An employee management system will be the best solution for the problem. It will provide easy online access to the employees that are currently at work and what they are working on. Also easy to allocate jobs to the employees that are done with what they were doing. The system will provide quick and reliable access to the running of the business saving the manager a whole lot of time and money.

Chapter 2

REQUIREMENTS ANALYSIS DOCUMENT

INTRODUCTION

In the previous chapter the user requirements were collected. In this chapter the requirements will be analysed. All the software development tools that will be used to implement the system will be identified.

DESIGNER'S INTERPRETATION OF THE PROBLEM

The manager and stakeholders' would like to have **remote access** to the employee system application. This means that it has to have a web-based interface but should also provide an interface for a mobile device. The data will be stored in a database.

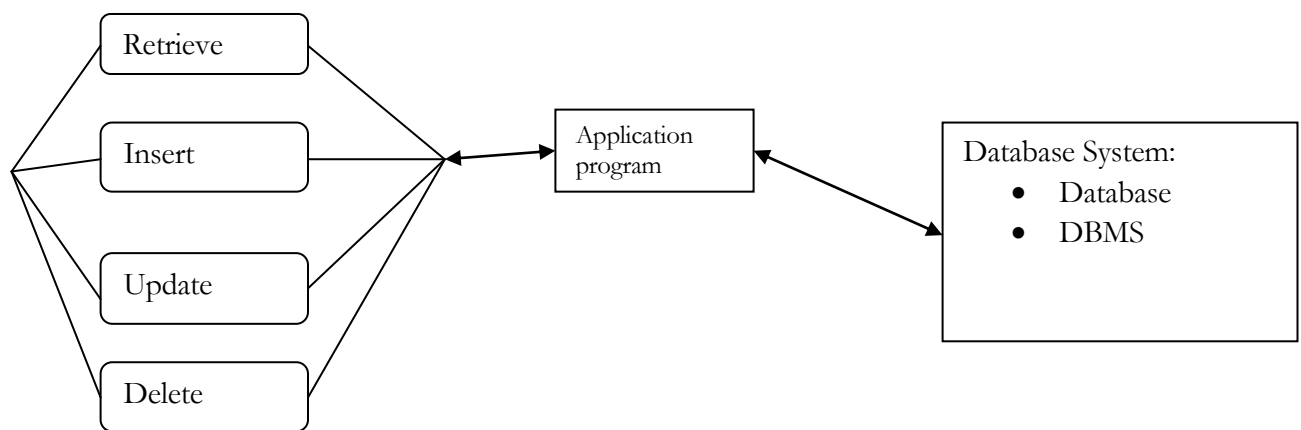
HIGH LEVEL CONSTITUENTS PARTS

The system will consist of two constituents. That is Software management and database management. Database management will be managed by the manager/administrator. Here are features or characteristics of these two constituents:

DATABASE AND SOFTWARE MANAGEMENT:

- Will be accessible on different devices or platforms.
- Will be to add or delete from database.
- Will enable editing of data on the database.
- Will enable retrieval of data from the database.
- Will enable searching through the data and make reports from his findings.

Below is depiction of how the users will interact with the system:



EXISTING SOLUTIONS

There are similar systems to employee management system. For example:

- Macros
- Pastel HR management.
- Halogen software (Strategic talent management).

ALTERNATIVE TECHNICAL SOLUTION

With the legacy system in mind, there is a need to computerize the whole system to be flexible.

BEST SOLUTION

The best solution will be an online database which will be flexible for the manager to access wherever he is. He can access from his mobile device (Laptop or cell phone). It can be achieved by making web-based software. For better planning we shall break down the work load in form of model and technology to be used.

MODEL

In order to put into consideration all the needs of the user, we will use Unified modelling language (UML) to model the system. UML will act as a blueprint to the whole system we are going to implement. UML will help me to break down the scope of the whole system so that it's flexible to implement the system.

TECHNOLOGY

Since it's web-based software, we decided to use:

- Apache (webserver)
- PHP
- MySQL

WHY??

This because they are:

- (Webslave/advantages of apache, 2011) Flexible.
- Its open source so it's cheap.
- It is efficient and easily managed.

We shall be using the following software.

APACHE:

Free available web server. This allows MySQL and PHP to run on it.

PHP:

Hypertext Pre-processor. It is a widely used general-purpose scripting language that is especially suited for web development and can be embedded into HTML. Above all it's free that is open source.

MySQL:

It's a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases.

PHPMyAdmin:

It is open source software written in PHP with the intention of handling the administration of MySQL over the World Wide Web. It also supports wide range of MySQL operations.

JAVASCRIPT:

It is a scripting language that used to make web pages interactive.

HTTP:

Hyper Transfer Protocol (HTTP) is a set of rules for transferring files (text, graphic, images, sound, video and other multimedia files) on the world wide web (WWW).

SYSTEM TESTING:

WHITE BOX TESTING

This is a verification technique for software developers use to examine if their code works as expected. We will use this method to detect errors during software development. White box which is also known as clear box helps in optimizing of my code. Using the aspect of unit testing we shall be go through each and every line of code. This makes it easy for us to remove extra lines of code that can bring about hidden errors.

USUABILITY TESTING

This is a technique used to evaluate the quality of the software. This is where random users are used to test the product using different data and their response to the system is put into consideration.

BLACK BOX TESTING:

It is also known as functional testing. This technique is used to test the complete system. Black box testing ignores all the whole system components and focuses only on the outputs generated for a particular input. Different data will be used for each function to check if the right output is gotten. This is to check the behaviour of the system as per different input data.

CONCLUSION

Employee management system (EMS) is going to be implemented using PHP, MySQL, Apache and PHPMyAdmin which are all open source applications. They are all a solution to web based system.

WHY??

After examining all the possible solutions, we decided on a web based system to address this problem.

CHAPTER 3

USER INTERFACE SPECIFICATION:

INTRODUCTION:

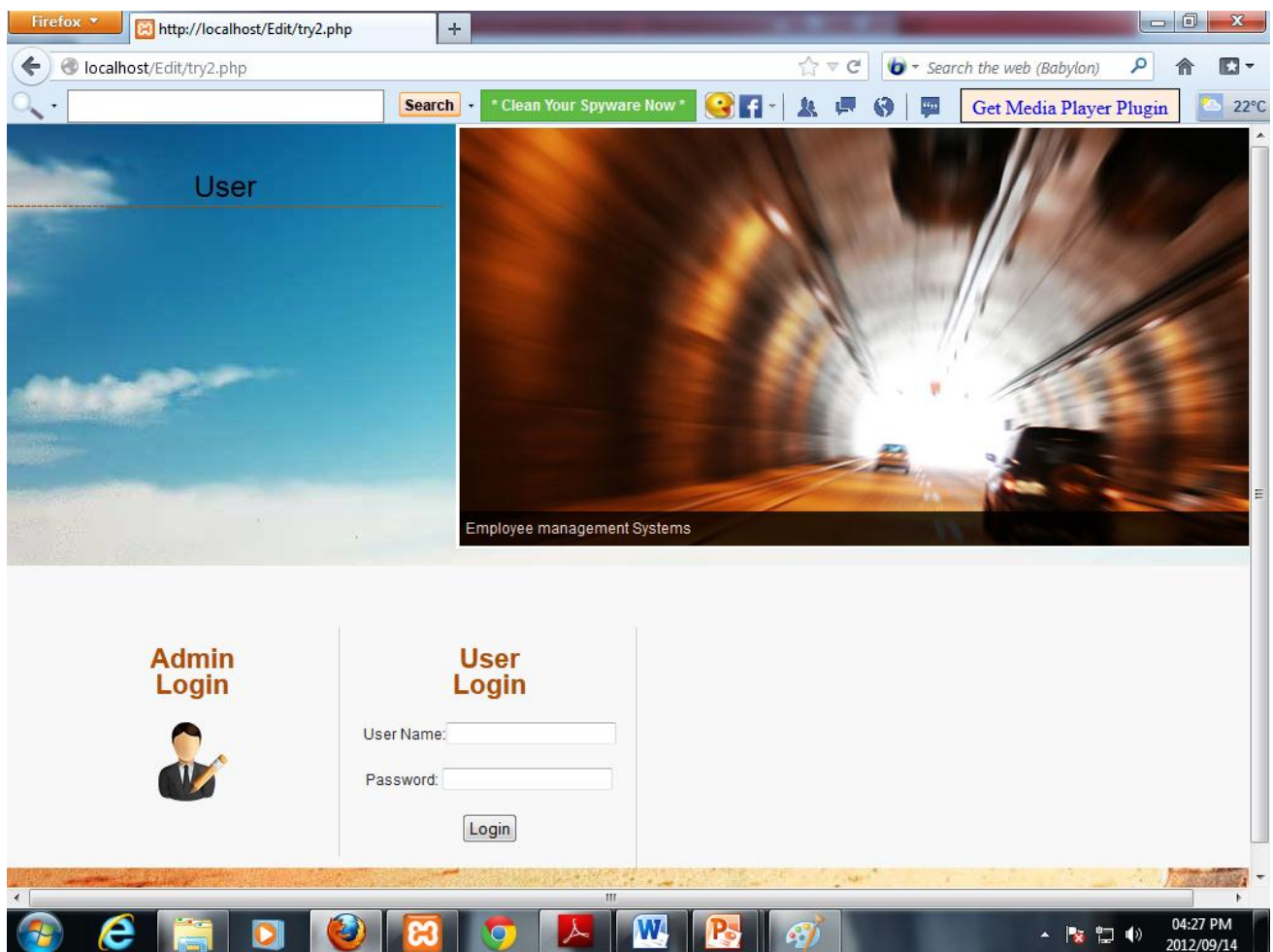
The previous chapters dealt with the user requirements specification. This chapter documents the current user interface and the various elements needed to fulfil the user requirements. The images of the user interface pages are included to demonstrate the application's look and feel.

DESCRIPTION OF COMPLETE USER INTERFACE SPECIFICATION:

The Login page:

The main screen will contain the login page (See Figure). The login page consists of Workshop link, contact link, Employee id, password and the login button. It consists of two text boxes that are employee id and password. There is also a submit button with the value login.

The contact link leads to a webpage that contains all the contact details of the administrators. The workshop link also leads to a webpage that gives a brief history of the workshop.



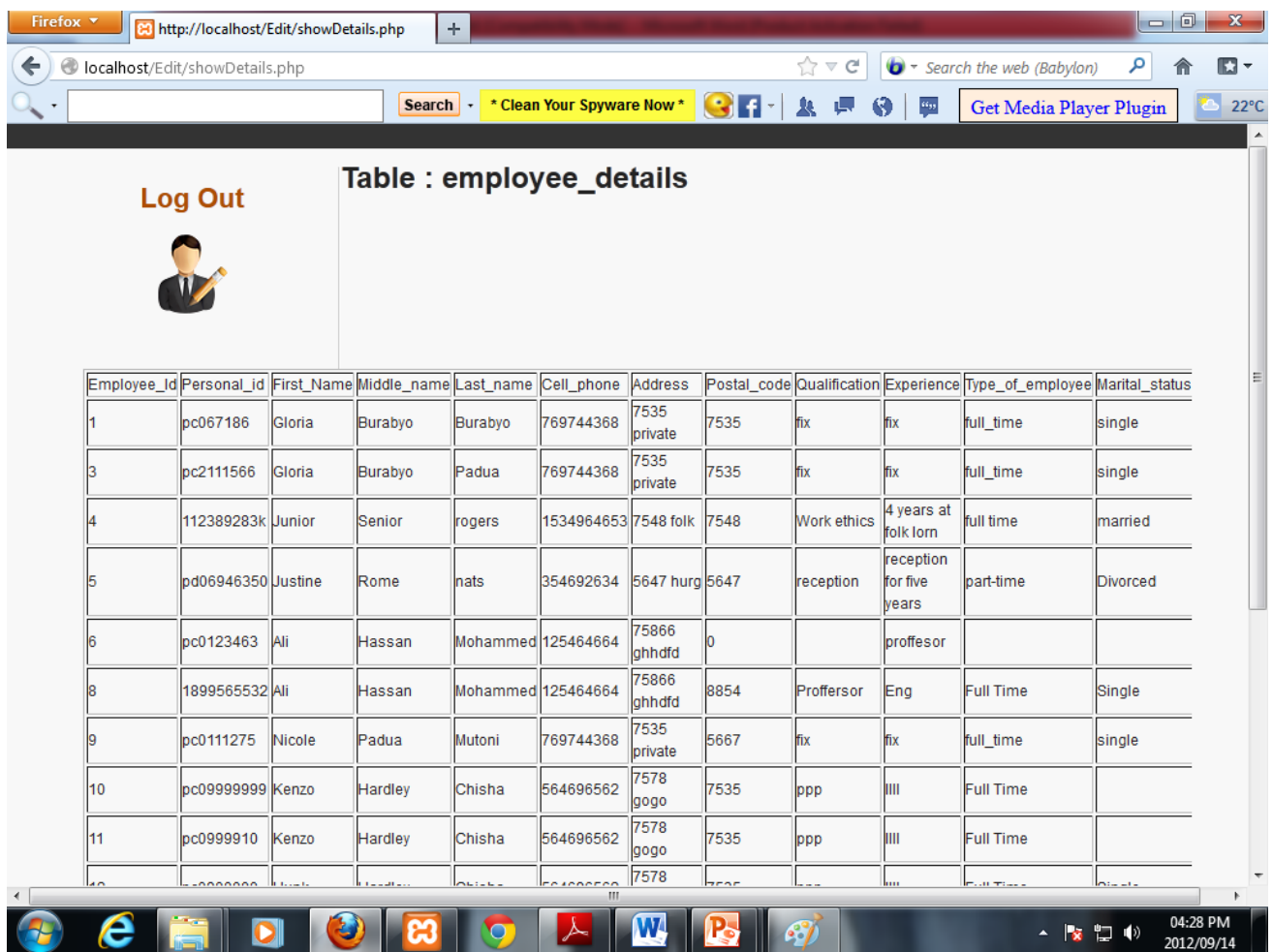
Application page:

There will be two kinds of users. They are employees and administrators. Once an employee is logged in, they are only able to view but not update anything in the database. They will be view links for:

- Work hours
- Contact details of fellow workmates and their own
- Holiday days they have left.

The feedback page:

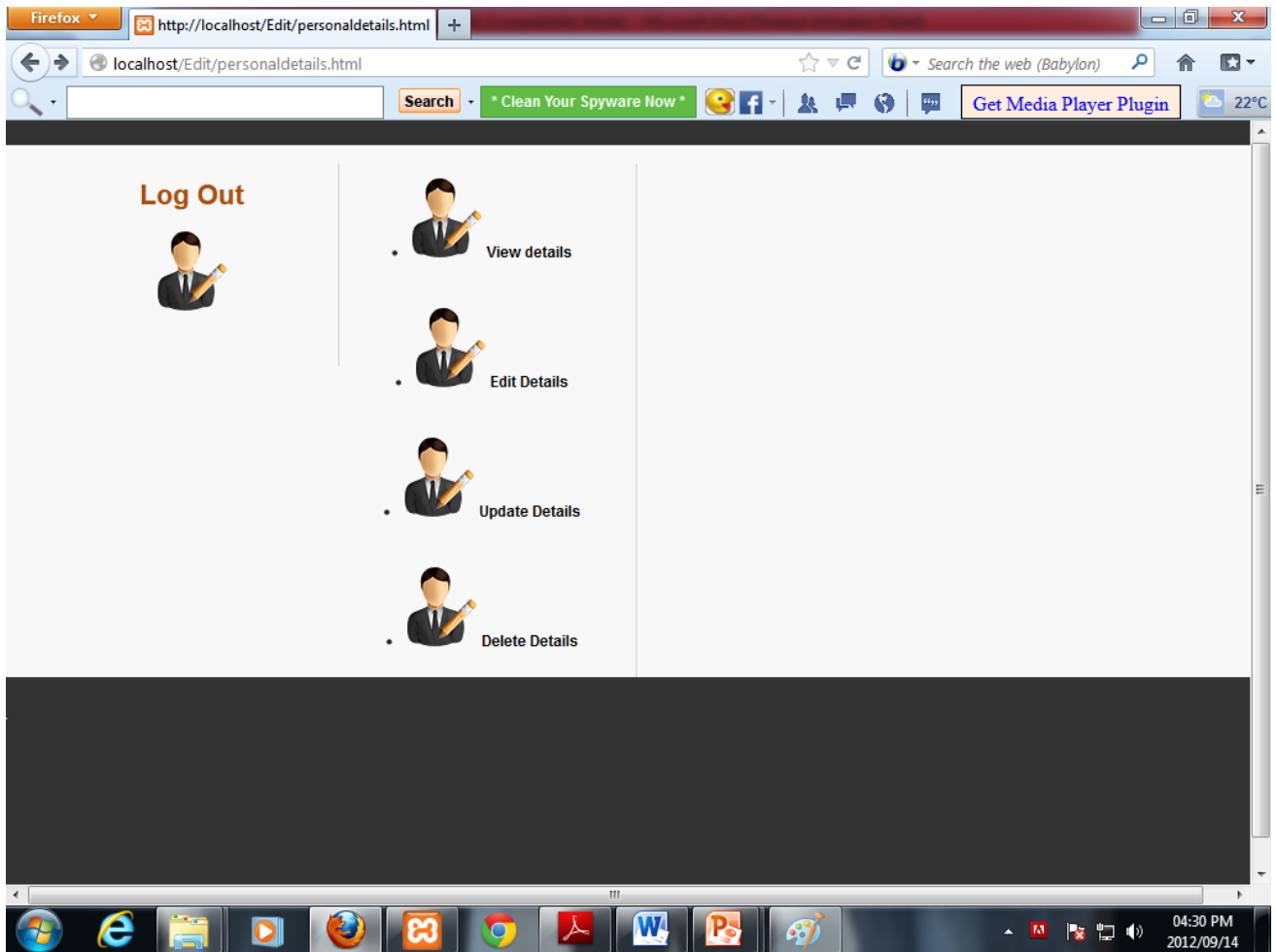
Figure 1.2 shows the feedback after clicking on the work hour's icon. This shows all the employee personal details. This however doesn't allow the employees' to change anything.



The administrator's application page on the other hand has very many different options mainly:

- **Salary link:** This deals with all the aspects concerned with salary of each individual employee. That is deducting taxes and insurances on their monthly allowances.
- **Employee link:** This deals with adding, deleting and updating all the employees' information.

Figure shows how the administrator application page looks like:

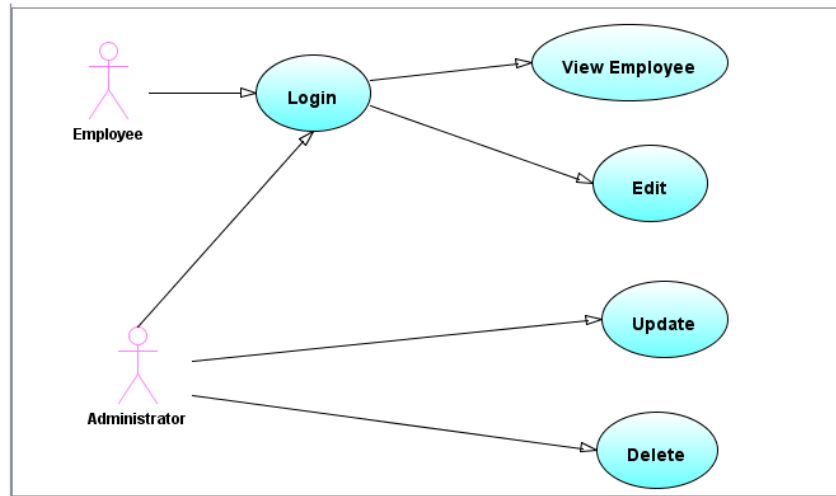


HOW THE USER INTERFACE BEHAVES:

When users input their username and password these must be authenticated before access is granted to the various applications. An error message is issued when the user enters the wrong username or password. The error message prompts the user that they have entered the wrong username or password. In the case of correct password and username then the user is granted access to the application that they are privileged to use. As shown in figure 1.3 the employee will only be able to view and edit his personal details and that of his /her emergency contact. On the other hand the administrator is privileged to use all the functionality of the application.

Figure 1.3 shows the use case diagram of the system's functionalities

USE CASE DIAGRAM



HOW THE USER INTERACTS WITH THE INTERFACE:

The activity diagrams (see figure 1.4) shows a summary of all the user activities. This all starts with logging into the system. The user enters his or her employee id and password. The input is authenticated by the system and when it's been successfully authenticated then the privileges are checked. The privileges are checked to ensure that the right main page is shown for each user. This is because the administrator and the employees have different privileges to the application system. For example an employee interacts with a system such as when he/ she logs in successful is selects a link to view the worksheet. The system displays the particular employee's work sheet. There is also an option for downloading the work sheet. The employee selects this option; the system returns a download of the spread sheet of the work sheet.

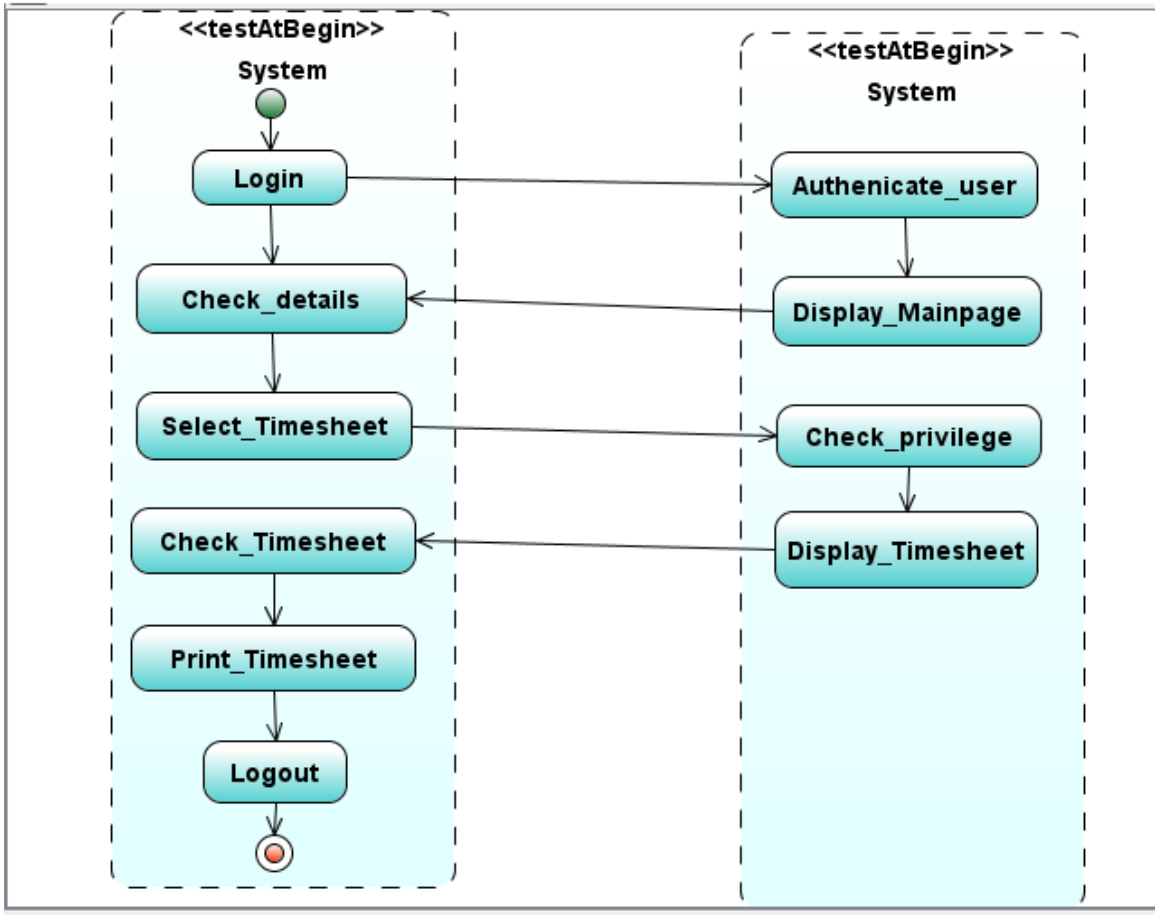


Figure 1.4

In the case of the administrator, he/she interacts with the system such as when the administrator is successfully logged in. The activity diagram below shows how an administrator can add data into the database:

HIGH LEVEL DESIGN

Introduction

This chapter concentrates on the object oriented analysis (OOA) or high level design of the problem. In this chapter every object is described and documented and the data dictionary provides the detail of all the objects. The class diagrams will show how the subsystems interact with each other. Furthermore, the objects, attributes and methods will be identified.

Data dictionary:

A data dictionary is a collection of descriptions of the data objects or items in a data model for the benefit of programmers and others who need to refer to them. The dictionary gives a brief description of the objects that will be needed to form the online employee management system.

Detailed breakdown of the technical solution

Employee details Object:

This contains employees' personal details and qualification. This object contains record for only one employee.

NAME	MEANING	VALUE
Employee_id	Employee number given by the workshop	6 numbers
Personal_id	Passport or id number	10 characters
Title	Employee's title	20 characters
First name	Their given name	20 characters
Last name	Family name	20 characters
Date of birth	Their birthday/ age	date
Cellular Number	Cell phone number for the employee	10 numbers
Home Number	Home phone number/landline of employee	10 numbers
City	Current city the employee lives	20 characters
Address	Home address of employee	30 characters
Email	Email address of the employee	20 characters
Postal code	Postal code number of employee	10 numbers
Qualification	Level of qualification the employee has reached	50 characters
Current experience	Current work experience of the	50 characters
Start date	The date employee started to work at the workshop	date
End date	The date employee left the workshop	date

Type of employee	Position the employee holds in the workshop	50 characters
Gender	Employee's gender coded M (Male), F (Female)	20 characters

Administrator's details:

This contains administrator's login details:

NAME	MEANING	VALUE
Admin_id	The administrator's given id	6 numbers
password	Chosen password	5 characters

Work history Object:

This contains the former employer's details or their former work place of an employee:

NAME	MEANING	VALUE
Company name	Employee's former work place	20 characters
Employer name	Employee's former Employer's name	20 characters
Address	Former work place address	20 characters
Cell number	Employee's former employer's cell phone number	10 numbers
Office number	Employee's former employer's office number	10 numbers

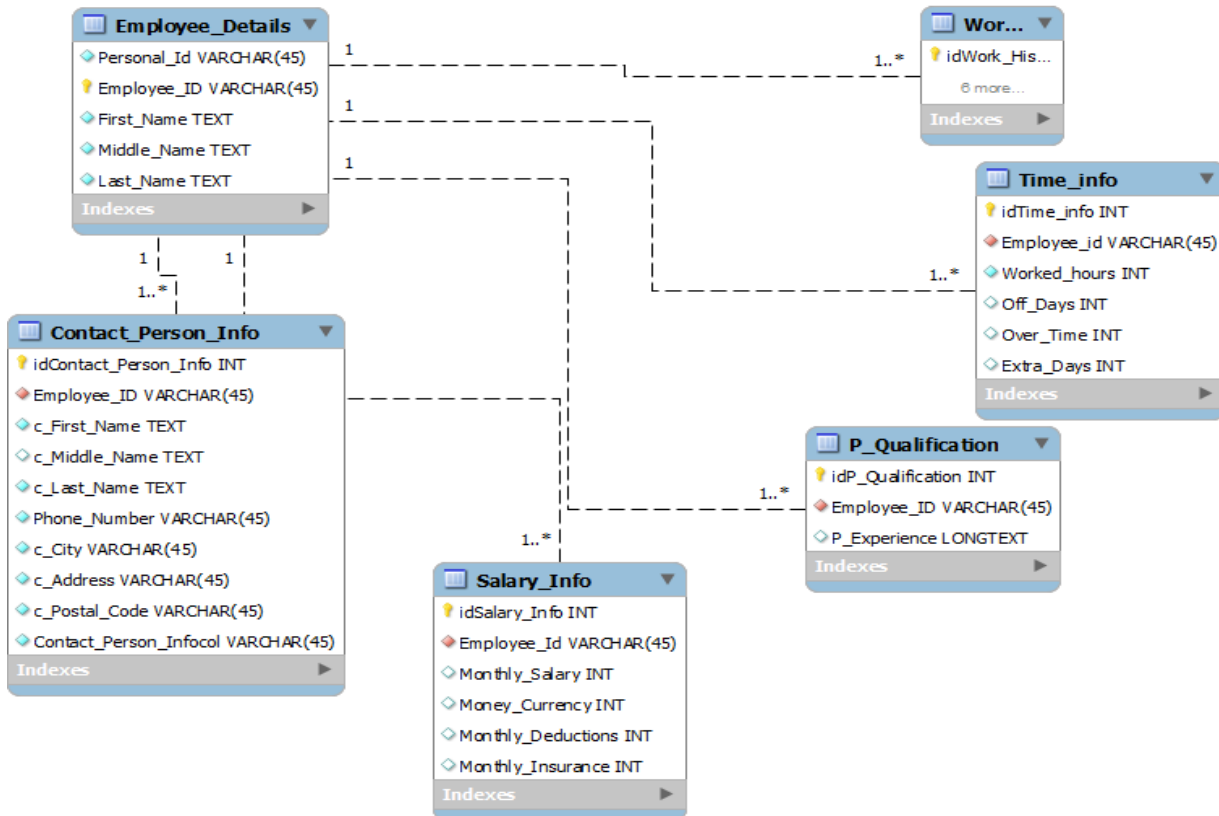
Time information object:

This contains the worksheet of the employees. That is how many hours they have worked in a month:

NAME	MEANING	VALUE
Job_id	License plate number of the car an employee worked on	20 characters
Worked hours	How many hours an employee has worked	10 numbers
Off hours	Number of hours employee has been off	10 numbers
Off days	Number of days employee has been off	10 numbers
Over time	Extra hour put in by the employee	10 numbers
Extra days	Extra days put in by the employee	10 numbers
Start day	Day employee started to work on a specific car	date
End day	Day employee finished to work on the car	date

Detailed interaction between subsystems:

The ERD (See Figure 1.5) gives a brief description of how the tables would be linked together in the system, in order to make data retrieval and updates easier for the user.



Conclusion:

The data dictionary described the attributes and the data type. In this chapter, each class and its attributes are explained in detail using a data dictionary. The ERD explains the relationship between the different classes and attributes that are needed to implement the user interface. In the next chapter the object oriented design, which is also known as low level design will be discussed.

OBJECT ORIENTED DESIGN (OOD) OR LOW LEVEL DESIGN (LLD)

Introduction:

In the previous chapter, each object was described and documented in terms of a data dictionary. In this chapter, the Object Oriented Design (OOD) will be described. The data types for the attributes, the algorithms and implementation particulars of each class etc will be explained. The OOD will present all the classes mentioned in the OOA in terms of pseudo-code. The state diagram and sequence diagram will also be explained in this chapter.

The inner details of class attributes (data types) and methods (functions):

The ERD (See figure 1.5) defined in chapter 4 shows all classes/tables that will be used inside the system. Also data types were well-defined in the data dictionary and functions are pronounced within the ERD.

State Diagram:

The state diagram (See figure 1.6) portrays the dynamic performance of login and application functions of the system. Both the employee and the administrator log in using the same application page. So both the users enter the username and password, and then the system authenticates the entries. If valid the user is recognized by the system or else the user is rejected. The user then selects an option and if he is privileged to use the option then the system confirms it. The state diagram below explains it all:

The event diagram or sequence diagram:

(See figure 1.7) Demonstrates the sequence of activities where the system is in operation.

Pseudo code:

CONCLUSION:

This chapter undertook the pseudo-code showed an outline of programs, written in a form which is clear and which will be easy to convert into a programming language. It also covered the state diagram and sequence diagram.

CODE DOCUMENTATION

7.1 Introduction

In the previous chapter the data types for the attributes, the algorithms and implementation details of classes were defined. Also, the state and sequence diagrams of the system have been explained. In this chapter a sketch of the entire system that helps to view that all activities performed in the system. The code is fully documented. For each PHP file, we defined MySql query which helps to retrieve the information from the database according our user requirements. The code also has comments to explain the algorithm.

7.2 System Operations

The following sketch (see Figure 1.8) represents all the operations between the users and the system. For every user, the operations are numbered according to the order in which they occur. The number of the operation will be used in the code documentation (the number of the action will be reflected as a comment in the code). The code documentation is provided according to the type of the user. For example the Login code will be given for all users (Employee and Admin)

7.3 Registration code:

```
<?php
```

```
/** ***** AUTHOR:  
    PADUA GLORIA EMAIL: 2916574@uwc.ac.za
```

```
***** This page is to  
    register a new employee(reggy.php)***
```

```
*****This page comes after the Admin  
    has hit the submit button on the html page
```

```
*****/
```

```
// connect to the database
```

```
$con=mysql_connect("localhost","root","");
```

```
if(! $con)
```

```
{
```

```
    die('could not connect: '.mysql_error());
```

```
}
```

```
//Select a database
```

```
mysql_select_db("ems", $con);
```

```
//Insert values from admin into the database
```

```
$sql="INSERT INTO work_experience  
(Personal_id,First_name,Middle_name,Last_name,Cell_phone,Add  
ress,Postal_code,Qualification,Experience,Type_of_employee,Mari  
tal_status,Gender,Date,Contact_person,C_cellphone,email)
```

```
VALUES($_POST[personal'],$_POST[first'],$_POST[middle'],$_  
_POST[last'],$_POST[cell'],$_POST[add'],$_POST[code'],$_POS  
T[quali'],$_POST[exp'],$_POST[type'],$_POST[status'],$_POST[  
gender'],$_POST[date'],$_POST[person'],$_POST[phone'],$_PO  
ST[email]);
```

```
$random = substr(number_format(time() * rand(),0,"",0,10);
```

```
$sql="INSERT INTO login(Employee_Id>Password)  
VALUES($random,$_POST[pass]);
```

```
if(!mysql_query($sql,$con))
```

```
{
```

```
die('Error: ' .mysql_error());
```

```
}
```

```
echo "1 record added";
```

```
mysql_close($con);
```

```
?>
```

Update Coding:

<?php

```
/** ***** *update.php*****
```

```
***** *Author: Padua Gloria *Email: 2916574@uwc.ac.za  
*****
```

```
***** this page(update.php) comes when the user login as  
Admin.*****
```

It has the button Submit which helps the Admin to update the any table and a view Button which helps him to find out the employee details have been updated.

This page also has a link "Balance Statement" which shows the payment statement.

```
**/
```

```
$con=mysql_connect("localhost","root","");
```

```
if(! $con)
```

```
{
```

```
die('could not connect: '.mysql_error());
```

```
}
```

```
// connecting to the database.
```

```
mysql_select_db("ems", $con);
```

```
//Intializing the values
```

```
$tbl=$_POST['table'];
```

```
$id=$_POST['id'];
```

```
$
```

```
// update data in mysql database
```

```
$sql="UPDATE $tble SET '$name'='$value' WHERE Employee_Id='$id'",
```

```
$result=mysql_query($sql);
```

```
// if successfully updated.
```

```
if($result){
```

```
echo "Successful";
```

```
echo "<BR>";
```

```
echo "<a href='list_records.php'>View result</a>";
```

```
}
```

```
else {
```

```
echo "ERROR";
```

```
}
```

```
?>
```

QUESTIONNAIRE

1. Do you have a timesheet for your workers? How does it work?

.....
.....
.....

2. Do you have stock?

.....
.....
.....

3. How do you monitor/ manage your stock?

.....
.....
.....

4. What kind of system do you use to allocate workload or jobs to the employees?

.....
.....
.....

5. Who is in charge of this whole system?

.....
.....
.....
.....

6. Do you have a timesheet for your workers? How does it work?

.....
.....
.....

7. Do you have stock?

.....
.....
.....

8. How do you monitor/ manage your stock?

.....
.....
.....

9. What kind of system do you use to allocate workload or jobs to the employees?

.....
.....
.....

10. Who is in charge of this whole system?

.....
.....
.....
.....

REFERENCES

- All about PHPMyAdmin http://www.phpmyadmin.net/home_page/index.php
- All about Apache http://www.webopedia.com/TERM/W/Web_server.html
- All about PHP <http://php.net/manual/en/intro-what-is.php>

BIBLIOGRAPHY

- Webslave/advantages of apache.* (2011, October). Retrieved march 15, 2012, from webslave:
<http://www.webslave.ws/apache/the-advantages-of-apache/>
- Dvoski, D. D. (2007). *Installing, configuring and developing with XAMPP.*
- Lerdorf, R. (2002). *PHP advanced tutoriala.* Veracruz.
- mpkrishi.org. (n.d.). *mpkrishi.org.* Retrieved March 04, 2012, from mpkrishi:
<http://www.mpkrishi.org/krishinet/hindisite/Pmis.pdf>
- Oracle and affiliates. (2012). *MySQL and PHP Reference.* Oracle.
- PHPMyAdmin and affiliates. (n.d.). *PHPMyAdmin.* Retrieved from phpmyadmin.net:
<http://www.phpmyadmin.net>
- Viteb. (2012). *employee-management.* Retrieved March 2012, from Viteb: <http://www.viteb.com/employee-management>
- williams, L. (2006). *White box testing.* Retrieved March 04, 2012, from agile.csc.ncsu.edu:
<http://agile.csc.ncsu.edu/SEMaterials/WhiteBox.pdf>

QUESTIONNAIRE

1. Do you have a timesheet for your workers? How does it work?

.....
.....
.....
.....

2. Do you have stock?

.....
.....
.....
.....

3. How do you monitor/ manage your stock?

.....
.....
.....
.....

4. What kind of system do you use to allocate workload or jobs to the employees?

.....
.....
.....
.....

5. Who is in charge of this whole system?

.....
.....
.....
.....
.....

PHP, vii

UML, vii

INDEX