



Linnæus University

School of Computer Science, Physics and Mathematics

Master Thesis

Salary Management System

Muhammad Azeem
2011-05-12
Subject: Software Technology
Level: Master
Course code: 5DV01E

Masud Ur Rehman
2011-05-12
Subject: Software Technology
Level: Master
Course code: 4DV01E

Acknowledgment

This thesis is regarding development of a web application system for an organization sweden sports academy. We are highly grateful to our supervisor, Professor Dr. Welf Löwe who provided us with an opportunity to work under his kind supervision. We are also grateful to professors Dr.Mathias Hedenborg and Dr.Jonas Lundberg for their kind guidance and support during our thesis project.

We are highly obliged to the CEO of this organization Mr. Gustav Einarsson for giving us the opportunity to work for their organization. We also thank our team leader Mr.Jonas Gustavson for his valuable guidance.

We dedicate this acknowledgment to all our professors and members of our department who gave us knowledge, confidence and shared their ideas during our studies in Linnaeus University. We are thankful to our friends with whom we spent our time and learned a lot, and last but not least we are thankful to our parents and families who gave us the confidence to believe in ourselves.

Abstract

The purpose of developing this software project is to fully automate salary system for an organization sweden sports academy. This software is capable of calculating monthly salaries, tax and social security of employees of that organization. On the basis of certain formulas it will generate files as an output such as bank files and salary slips etc. The system is also capable of calculating tax payable by the organization and social security. The tax file is generated for the tax office. All the data such as, employee information, salaries, social security and tax will be stored in database as a track record.

Key Words: Employee salary calculation, tax calculation, social security

Contents

1	Introduction	1
1.1	Project Outline	1
1.2	Motivation	2
1.3	Problem Identification	2
1.4	Customized Solution	2
1.5	Project Goals	2
1.6	Goals Criteria	3
1.7	Work Division	4
1.8	Report Structure	4
2	Comparison	5
2.1	Why Customized Solution	5
2.1.1	Softwares Available in the Market	5
2.1.2	Analysis of Different Salary Calculators	5
2.2	Employee Salary Management System	7
3	Requirement Analysis	8
3.1	Methodology	8
3.2	The SDLC Waterfall	8
3.2.1	Reasoning to Choose Waterfall Model	9
3.3	System Requirements	10
3.3.1	Functional Requirements	11
3.3.2	Non-functional Requirements	11
3.4	System Use case	12
3.4.1	Team Leader	12
3.4.2	Project Manager	13
3.4.3	System Administrator	14
3.5	System Use cases In Detail	14
3.5.1	Use Case Scenario of Log-in	15
3.5.2	Use Case Scenario of Registration	15
3.5.3	Use case Scenario for Deletion of users	16

3.5.4	Add and Edit Use Case Scenario of Employee	17
3.5.5	Use Case for Activation/Inactivation/Deletion of Employee Accounts	19
3.5.6	Use case for Salary Calculation	20
3.5.7	Use case for Generation of Salary Slip	21
3.5.8	Use case for Bank File	22
3.5.9	Use case for Tax File	23
4	Implementation	24
4.1	DreamWeaver MX 2004	24
4.2	Apache 2.2.11	24
4.3	PHP 5.2.9-2	25
4.3.1	History	25
4.3.2	What is PHP	25
4.3.3	Why PHP	25
4.4	MySQL 5.1.33	26
4.5	phpMyAdmin 3.1.3	26
4.6	Software Development Architecture	26
4.6.1	Three-Tier Architecture	27
4.6.2	Implementation of Three-Tier Architecture in the System	28
4.7	Implementation of the System	29
4.7.1	Implementation of User interfaces	29
4.7.2	Salary Calculation	29
4.7.3	Drop Down Menus	29
4.7.4	Form Validation	30
4.7.5	File Handling	30
4.8	Design and Implementation of Data Base	31
4.8.1	AMGS _ Emp _ Bank _ Detail	34
4.8.2	AMGS _ Banks	35
4.8.3	AMGS _ Emp _ Sal _ Tax	35
4.8.4	AMGS _ Social _ Security	36
4.9	System Description	37
4.9.1	Anatomy of the System	37
4.9.2	Login	37
4.9.3	Employee Registration	38
4.9.4	Salary Calculator	39
4.9.5	Data Verification	40
4.9.6	Generate Employee Pay Slip	40
4.9.7	Employee Salary Slip	41
4.9.8	Employee Salary Report	42
4.9.9	Company/City Salary Report	43

5	Conclusion and Future Enhancements	44
5.1	Conclusion	44
5.2	Future Enhancements	46
5.2.1	Budget Program	46
5.2.2	Visual Representation of Data	46

Chapter 1

Introduction

This chapter provides an introduction to the project, motivation and criteria to achieve the objectives.

1.1 Project Outline

The organization we are working for, is dealing with sports activities within different cities of Sweden and Finland.

The task is to build a salary management system for this organization. Current salary system is manual therefore the organization wants to switch to an automated computerized salary management system. After building this system we have to integrate it with the existing computerized system.

The existing system named as AMG dealing with the client registrations, keeping records of clients, client billing etc. Hence we can say, employee salary system will be a subpart of the existing computerized system.

Employee salary management system is a web application, enabling the organization to handle salaries of employees of sweden sports academy. The managers or team leaders of this organization are able to fill out all necessary information of an employee, i.e., residential address, social security number, banking details, type of salary either fixed, monthly or hourly and other relevant information.

This system should be capable enough to calculate the salaries of employees. On the basis of social security number, it will create a bank file by providing a few parameters to the system and transfer it to the bank by the end of each month. Upon request, the employees can receive their salary slips through email.

Moreover, the system should be able to calculate tax deductions of every employee on monthly basis, throughout offices nationwide, annually. Finally,

the tax files should be sent to tax office.

1.2 Motivation

All calculations such as employee salary, employee tax, to create bank files, organization tax calculations etc, are being done manually at the moment which is a time consuming task. Hence, a system is required that can perform all above said operations automatically. Moreover, the system should be user friendly, flexible, fast and highly secure.

1.3 Problem Identification

The organization is using a manual system to calculate salaries and tax of employees. They also have to calculate and keep record of tax of whole organization and maintain tax files manually. The management of organization also has to deal with bank files manually. All this work requires a lot of paper work, is extremely time consuming job, and accordingly costly as well, as they have to hire more man power.

Since there is always a risk of human errors present in a manual system so the chances of errors are very high and to figure out such errors is also a very lengthy procedure. Therefore, the organization decided to switch from a manual system to an automated computerized salary management system.

The requirement of the organization is to develop a web based application that is able to deal with salary and tax calculation of employees within the organization and maintain its data base. Furthermore this system should be able to generate automatic files such as, salary slips, bank files and tax files in response of queries from the data stored in the database.

1.4 Customized Solution

The organization required a customized solution. We will later on discuss in chapter 2, why is customized solution required? We will briefly study different salary calculation softwares available in the market.

1.5 Project Goals

The system should be capable of performing following functions:

- Store basic information regarding employees of the organization.

- Store salary information of employees (entered by the team leaders in each city) such as, working hours, salary per hour, salary before tax, tax percentage, total amount of tax paid , salary after tax, social security fee, on monthly basis.
- System should be able to generate the following salary information:
 1. Salary calculation for every employee for per month and per annum.
 2. Total salary calculation for each month for the whole company.
 3. Total salary calculation for a year for the whole company.
 4. Total salary of the organization within a city per month and per annum.
 5. Total salary calculation for every domain of the company, for every month and for the whole year.
- Once all team leaders have compiled the information, project manager checks the salary calculations and finally creates a text file to be delivered to the bank. Amendments can not be made in the system after this and the data is stored. The month is set as "frozen".
- Social security fee depends upon employee age. Hence, percentage on the salaries would be variable.
- System should be able to generate the bank files, tax files and employee salary slips automatically.
- Salary slips can be sent to the employees upon request.

1.6 Goals Criteria

The criteria to achieve the goals of our project is as following:

- To generate bank file/invoice
System should be able to generate a bank file and salaries should be delivered to employee accounts.
- Performance and efficiency
The system should perform according to the requirements and provide appropriate and accurate results.

To measure the performance and efficiency of the system we will put load by inserting input by the number of users. We will then check the results and response time of the web pages.

- Delivery on time

We have to deliver the software by the end of October 2010.

1.7 Work Division

Designing, participation in web programming, contribution in database development and parts of report writing is done by Masud Ur Rehman.

Participation in software engineering, designing, web programming, participation in database development and parts of Report writing is done by Muhammad Azeem.

1.8 Report Structure

"Chapter 1" includes introduction to the organization, project, problem identification and goals criteria.

In *"Chapter 2"* we will compare different softwares available in the market for salary calculation.

In *"Chapter 3"* design structure and requirement analysis of the system will be discussed and sequence diagrams along with use case diagrams will be explained.

"Chapter 4" includes software architecture, database design, implementation and how system works? Technologies and tools used in the development of our system will be briefly explained.

"Chapter 5" covers conclusion and future enhancements in the system .

Chapter 2

Comparison

In this chapter we will compare a system of our own choice with different salary calculation softwares available in the market. On the basis of this comparison we will describe the motivation of developing customized solution.

2.1 Why Customized Solution

There are a variety of software tools available in the market that are able to calculate the salaries of employees. Now the question arises, why is it required to build a customized software?

2.1.1 Softwares Available in the Market

There are many softwares available in the market that are being used by the organizations to calculate the salaries of its employees. We are discussing two softwares here for example.

2.1.2 Analysis of Different Salary Calculators

Salary Calculation Software

This software is able to do calculations of employee salaries on the basis of data provided by the work tracking system. It is basically a combination of two systems, i.e., "work management software" and "Salary calculations". Second one is totally depending on the first one by taking data as input from work management software and calculating the salary on the basis of this data. The software has the following features :

- *"Planning Project"*

- *"Keeping track of working results and helping managers to define work results"*
- *"Helps to Calculate salary from work results"* [1]

Analysis of Salary Calculation Software

- This software is facilitating work management and task scheduling more than the salary calculation. Salary calculation is one of its secondary feature, where as work management is its primary feature, which is not required by the organization sweden sports academy. They are working very well and are satisfied with AMGs, existing employee management system, and are planning, scheduling and task managing.
- This software is not providing any feature to calculate taxes and social securities.
- This software is unable to generate pay/salary slips and bank/invoice slips.
- The company has to purchase the license per user to use this software.
- This software is not providing any interfaces or plug-ins that can be used to merge it with the existing system working with in the organization.
- As with the growth of company, it may require to enhance the system or if company needs any amendments in the system, it would be very hard to achieve such goals as the purchased softwares are delivered always in .Exe form.

Spine Payroll

"This is a very simple, flexible and user-friendly payroll management software specially for India, that takes care of all requirements related to accounting and management of employee payroll. Spine payroll stores complete records of employees, generates pay slips, attendance register, computes all allowances, deductions, generates all statutory and MIS reports." [2]

Analysis of Spine Payroll System

This software, to some extent, might be the alternative of our software but still it is not required by the organization due to the following reasons.

- The organization wants a customized solution which can perform the tasks discussed earlier.

- The organization is already using employee management system which is working perfectly. Therefore it is not wise to purchase another system which has employee management system along with the salary management.
- Spine payroll system is specifically designed for a particular region India according to the requirements of their organizations. In India tax system is different from Eu.
- Spine payroll system is not providing any plug-ins that could be helpful to merge it with the existing system.
- It requires a license per user therefore it is an expensive solution.

2.2 Employee Salary Management System

The System we are going to develop is according to the user requirements which will perform salary calculation as well as tax calculation and file handling such as salary slips along with bank invoices. Moreover, the system will be user friendly and flexible enough to be enhanced according to the needs of the users in future.

Chapter 3

Requirement Analysis

In this chapter, we are going to discuss the requirements, design of the system and analyze them using some appropriate software models.

3.1 Methodology

The most suitable and appropriate software development methodology we found for our software development is *waterfall software development model*, which gave us a clear view about our software and helped us to achieve our goal.

3.2 The SDLC Waterfall

Small and medium size softwares are usually divided into six stages that are interrelated with each other in a top-down approach named as waterfall. The input of one specific stage is the output of previous stage which initializes the next stage. At every step or stage of the model some additional information is also added up into the input of that stage and generating the results. But that additional information is restricted in scope and previous stages are directly traceable from that stage.

The first stage of the waterfall model is *planning stage* in which the most critical task is to clearly mention the high level requirements or goals of the software. It helps to make feasibility and risks associated with the project and also provides the basic project structure.

The next stage is the *requirement definition stage* that takes goals as input from the planning stage and then these goals are defined into a set of one or more requirements. Major functionalities, initial data entities and the operational data areas are defined under this stage.

The input of *design stage* is the out put of the *requirement stage* which is an approved requirements documentation. In this stage the design elements are defined with the help of interviews, prototypes and workshops conducted. These design elements consist of functional hierarchy , business process diagrams, pseudo code and entity-relationship diagram. These design elements provide detail description about the software and each element is related to a specific requirement.

The *development stage* is initialized by the previous design stage. At this stage the code for the design elements of software is written and it provides the functional software components.

Integration and *test stage* is provided the output of the development stage. At this stage the different software functioning components developed in the previous stage are integrated with each other to provide full fledge software project which is providing all the high-level requirements. This software is also passed through different test cases to check the validity, correctness, completeness and hence enables us to achieve our goals regarding our project.

Finally, the *installation* and *acceptance stage* arrives. The software is loaded to the server at site of customers and tested with different test cases once again to check the correct working. If all this shows adequate results satisfying the customer then the software is handed over to the customer formally. [3], [4],[5]

3.2.1 Reasoning to Choose Waterfall Model

As in this model all the phases are in a sequence and are dependent with one another , therefore a phase can not be started until the previous phase is completed and fully documented. This approach is most appropriate for our project, because all the requirements and goals of our project are very clear. Secondly, it is easy to do work in components and waterfall model is providing this approach. After the completion of all the phases individually, they are integrated together. This model is very economical and risk free due to its sequential approach.

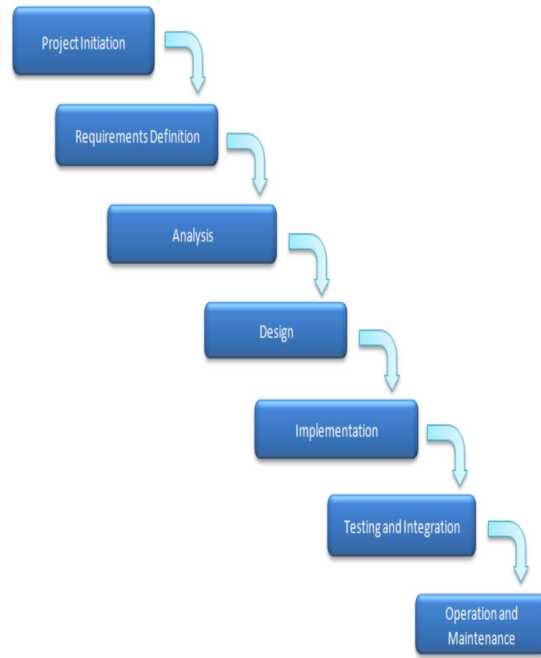


Figure 3.1: Waterfall Model Diagram

3.3 System Requirements

Requirement analysis is an important phase of the system development cycle which provides us all the specification of system in detail which are very essential to build the system and also provides us knowledge about the behavior of the system. Collection of system requirements is a very critical point because the whole system is based on this knowledge and it also provides input to the next following stages of the system development life cycle. We can classify the requirements of the system in two categories as following:

- ***Functional Requirements***
- ***Non Functional Requirements***

3.3.1 Functional Requirements

Functional Requirements specifically defines functionalities of the system, behavior of the system and the goals to achieve it. Functions that describe the behavior of the system are considered as behavior requirements and will be shown in the form of use cases. The importance and description of functional requirements are explained in the following table:

Table 3.1: Functional Requirements

ID NO.	FR	Importance	Description
1	Security	Essential	User authentication is required.
2	Web Interface	Essential	Provides interaction between user and database.
3	Database	Essential	Stores the calculated taxes and salaries.
4	File System	Essential	Stores bank files and tax files.
5	Search	Required	To search out tax files, salary slip and bank files.

3.3.2 Non-functional Requirements

Non-functional requirements are also known as quality of a system. Hence, it provides us knowledge regarding the operations instead of behavior or functionalities, contradicting with the FR in this manner. Non-FR are described in the system architecture helping us to achieve the quality goals and improves the functionalities of the system. Non-functional requirements are as following:

Table 3.2: Non-functional Requirements

ID NO.	Non FR	Importance	Description
1	Performance	Essential	Performance of system should be adequate and fast.
2	User Friendly Environment	Essential	System interface should be simple and user friendly.
3	Application Maintenance	Essential	Maintenance and documentation should be done thoroughly.
4	Paging	Essential	System should be capable of splitting large amount of data using data paging to present information in a user friendly way.
5	Application Scalability	Required	System should be flexible and expendable for future use.
6	Platform Independence	Required	System should be capable to work in any environment.

3.4 System Use case

In this section we will discuss the use case of salary management system. Uses cases show how the users interact with the system. There are three actors in our system, team leaders, project managers and administrator. Each one has its own responsibilities and level of access to the system. The figure 3.2 is showing a general overview of use cases and its general description.

3.4.1 Team Leader

Team leader is responsible to maintain basic information of his team members such as name, address, date of birth, joining date, social security number etc. A team leader will be assigned a login and password for the system. After authentication he/she will be the only person authorized to add and edit the basic information of the employees . He/she will be able to calculate the salaries of the particular employees on the basis of given parameters.

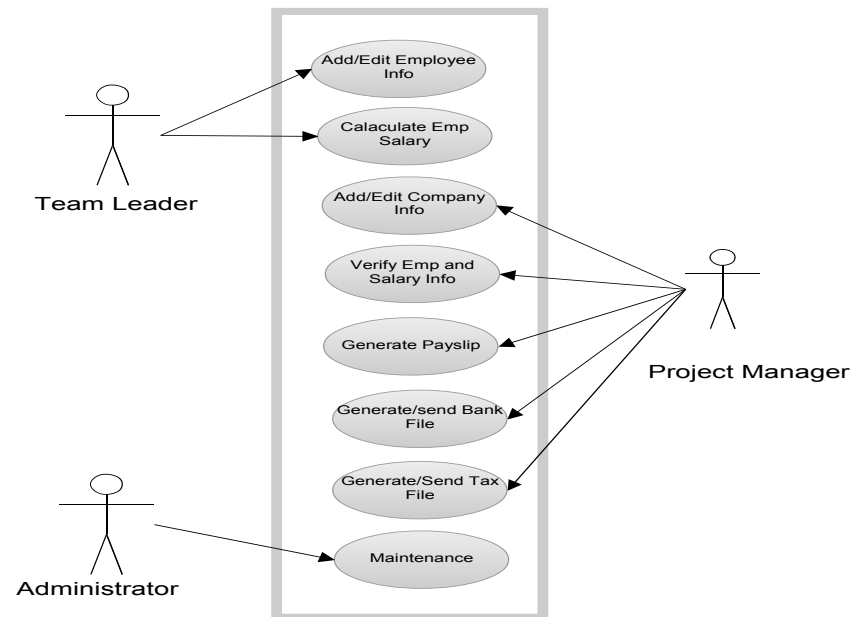


Figure 3.2: Use Case Diagram

3.4.2 Project Manager

Project manager is the main actor in our system. After passing through the authentication process the he has the access to the system and is authorized to do the following actions:-

1. Addition of necessary information, updates and news etc.
2. Verification of employee data and salary calculation entered by the team leaders.
3. To generate salary slips through the system.
4. To generate bank files and send them to the bank.
5. To calculate tax payable by employees and organization on the basis of formula provided by the tax office.
6. To generate tax files and send them to the tax office.

3.4.3 System Administrator

The last actor is system administrator. Maintenance is one of his major responsibilities. He checks the security logs, handles the security issues, checks the application logs, handles the error messages of the application, handles the database errors and web server problems etc.

3.5 System Use cases In Detail

The figure 3.3 is showing the use cases of the system in detail and will be discussed further each use case individually.

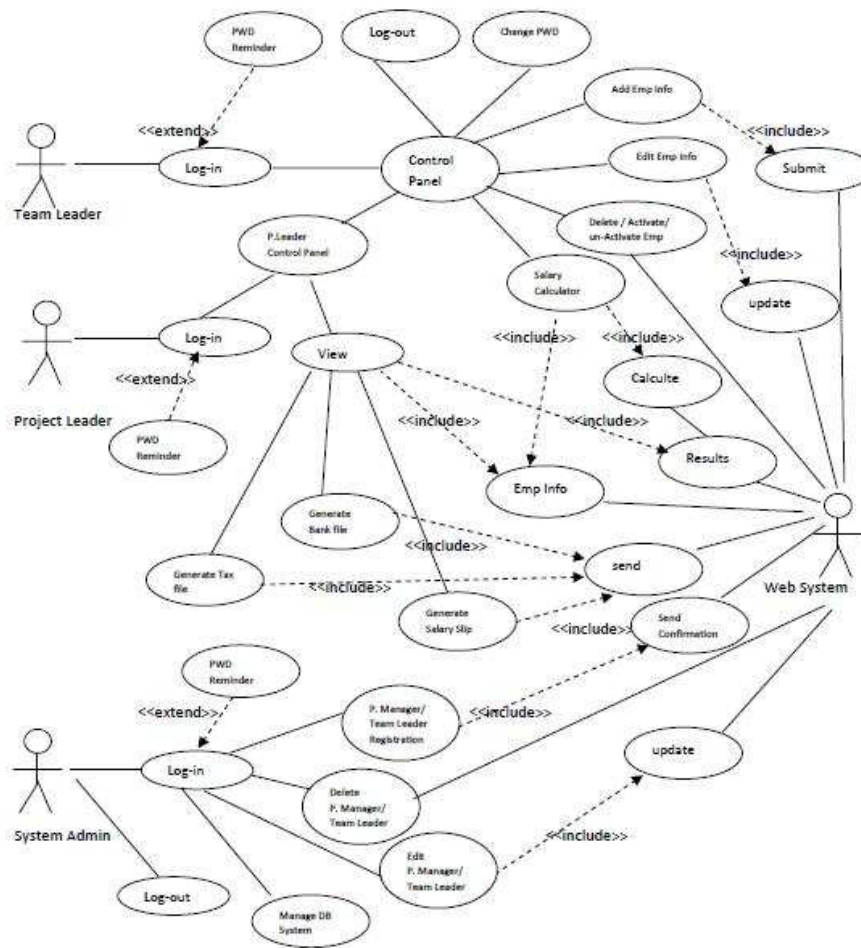


Figure 3.3: Use Case Diagram in Detail

3.5.1 Use Case Scenario of Log-in

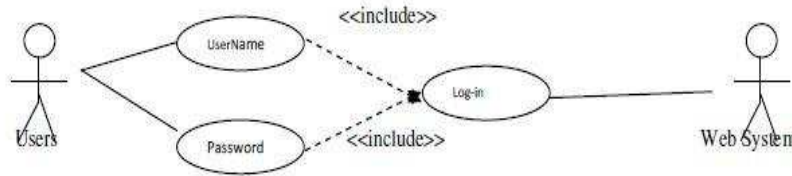


Figure 3.4: Use Case Diagram of Login

Scenario

Actors (administrator, project manager, team leader) want to enter into the system to perform the specific operations.

Basic Flow

When any of the actors want to log-in, the system will ask the user name and password. On entering the log-in parameters the system will authenticate them. After validation it will authorized the actor to enter.

Alternative Flow

An error is generated if the log-in parameters are not valid.

Pre-Condition

User must be registered into the system.

Post-Condition

The user will be entered into the system successfully.

3.5.2 Use Case Scenario of Registration



Figure 3.5: Use Case Diagram of User Registration

Scenario

Registration is required to enter and use the system. After logged-in the administrator is authorized only to register the actors (project manager and team leader) into the system or edit their accounts. On completion of the user registration process a confirmation mail will be sent into the email accounts of the actors containing the log-in and password of the user.

Basic Flow

In case of user registration, system administrator will enter the required information about the actors (project manager/team leader) including the log-in name, password and create the new user accounts.

In case of edition of user accounts information, administrator will re-enter some information and update of existing users.

Alternative Flow

In case of creating new user account, if the log-in name already exists or if the entered information is not in a valid format, an error message is generated.

In case of editing user accounts, system will check the format of re-entered information. If it is invalid an error message is generated.

The system administrator then can re-enter the information in basic flow or cancel the process.

Pre-Condition

In case of new user accounts, no entries should be existing in the system for that user.

In case of editing user accounts, a user should be registered into the system.

Post-Condition

In case of new user entries system will create a new user in the database. System will update the existing user accounts in the database, in case of editing user accounts.

3.5.3 Use case Scenario for Deletion of users

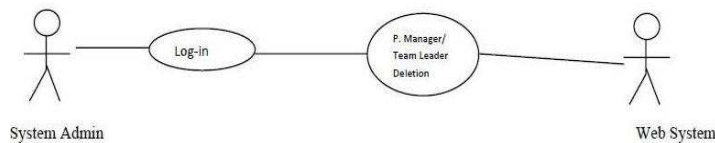


Figure 3.6: Use Case Diagram of User Deletion

Scenario

Administrator wants to delete the existing user accounts.

Basic Flow

Administrator can delete the existing user accounts after logging in when ever it is necessary.

Alternative Flow

System will generate a message to confirm deletion of existing user account or give an alternative to cancel the process without deletion.

Pre-Condition

User account must exist.

Post-Condition

System will delete user account and update the database.

3.5.4 Add and Edit Use Case Scenario of Employee

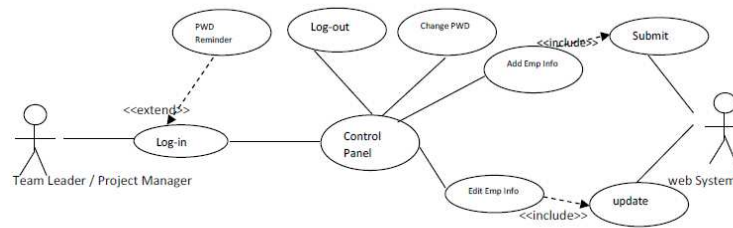


Figure 3.7: Use Case Diagram of Add/Edit Employee

Scenario

After logging in the actors enter into the interface called control panel from where they can perform the following functions :

- Log-out directly from the system
- Change of password
- Add information of a new employee
- Edit information of the existing employee

Basic Flow

In case of log-out, actors simply quit from the system.

In case of change in password, actor has to put his old password and type a new password. After authentication, password will be changed successfully.

In case of adding a new employee information, actor will create a new employee account and add all the necessary information provided in the employees registration form into the system.

In case of editing employee information, actors will access the existing user information, make some changes and then update the data in the system.

Alternative Flow

In case of log-out, there will be not be an alternative flow.

In case of change in password, system will match the old password entered with the password saved in the database and also cross check the new entered password. If old password or new entered passwords does not match, the system will generate an error and ask for correction.

Actor will remain on the same page in case of the error, from where they can go back to main control panel by canceling the process or trying again to change the password.

In case of adding an employee information, system will check if the log-in name already exists. If all the required fields in the form are not entered or the formate is incorrect e.g, email ID, the system will generate an error. Actor will remain on the same page.

In case of editing employee information, system will check the entered information is in correct format. If not then an error will be generated. Actor will remain on the same page.

Pre-Condition

Actor must be logged-in in order to log-out.

In case of change password, actor must be logged-in to the system.

In case of adding a new employee, no previous entries for that employee should be existing.

In case of editing employee information, a user account exists in the system.

Post-Condition

In case of logging-out, system will quite the user from control panel interface and take him into log-in interface.

In case of change in password, if the user performs the basic flow successfully, the new password will be updated into the system.

In case of adding a new employee information, the employee will be registered successfully into the system and all the necessary information

regarding him will be stored into the database.

In case of editing employee information, specific information of that employee will be updated into the database.

3.5.5 Use Case for Activation/Inactivation/Deletion of Employee Accounts

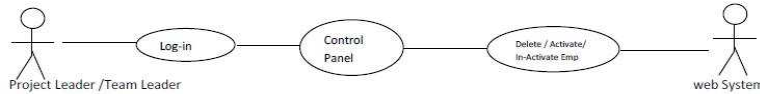


Figure 3.8: UCD of Activation/Inactivation/Deletion of Employee Accounts

Scenario

In case, if an employee temporarily or permanently quits from the job or rejoins, then the authorized users will change his status in the system.

Basic Flow

After logging-in, the user will search for the employee from the database by entering his ID and performing the following functions :

- The user will activate the account, if the employee wants to rejoin the organization after some time or change his location.
- In-Activate the account if employee wants to quite temporarily.
- Delete the employee account if an employee quits permanently.

Alternative Flow

The system will generate an error if the entered ID is not valid and the user will remain on the same page.

Pre-Condition

User must be logged-in and authorized to perform task. Account of an employee must exist.

Post-Condition

If the basic flow is performed successfully the status of an employee account will be changed and updated in the database.

3.5.6 Use case for Salary Calculation

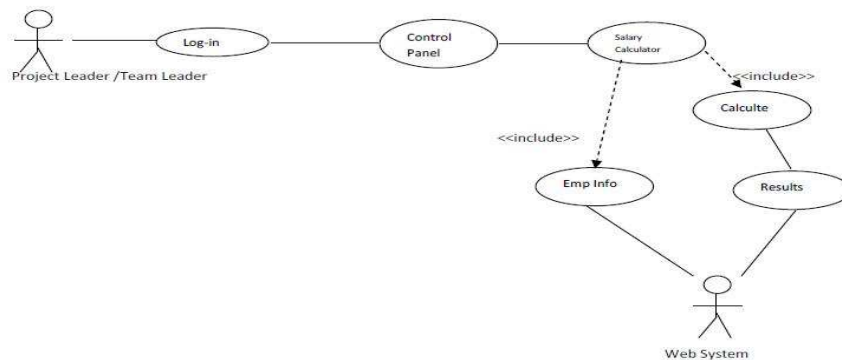


Figure 3.9: Use Case Diagram of Salary Calculation

Scenario

User wants to calculate the salary of an employee.

Basic Flow

After logged-in to system the user will select salary calculator from control panel to calculate the salary of an employee. In salary calculator the user will enter the ID of an employee. A procedure calls in response and gathers the required information about the user from the database, adding it to the salary calculator.

User will enter the following fields:

- In case of hourly based salary, enters hours worked per day.
- In case of fixed salary per month ,enters it.
- Percentage of tax.

- Percentage of social security.

Following calculation will be performed:

- Salary of an employee will be calculated.
- Tax of an employee will be calculated.

The results of an employee will be stored.

Alternative Flow

If entered salary data is incorrect the system will generate an error and user will remain in the salary calculator interface.

Pre-Condition

Employee account should be activated.

Post-Condition

Calculations performed in response to the basic flow, will be entered into the database.

3.5.7 Use case for Generation of Salary Slip

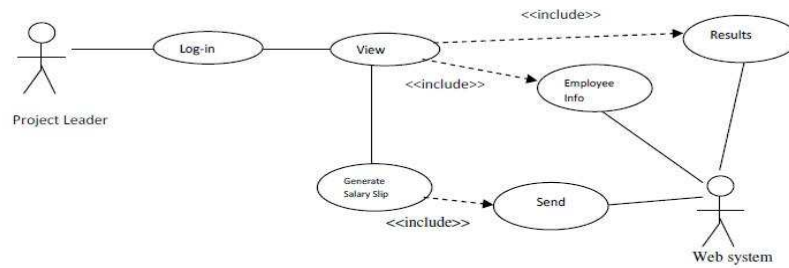


Figure 3.10: Use Case Diagram of Salary Slip Generation

Scenario

User wants to generate salary slips of the employees.

Basic Flow

Project manager will select employees one by one, mention the salary month being paid and submit to server. After input validation, server will then generate the payslips and send it to employees.

Alternative Flow

System will generate error message of wrong inputs.

Pre-Condition

Salary calculation of an employee for that month should be in the database.

Post-Condition

System will generate payslip.

3.5.8 Use case for Bank File

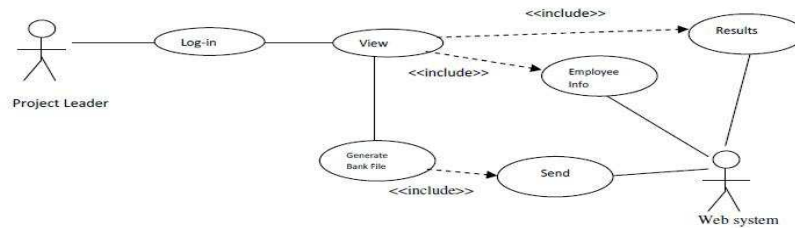


Figure 3.11: Use Case Diagram of Generation of Bank File

Scenario

User wants to generate the bank slip/invoice.

Basic Flow

User will select the company and city along with the date of salary payment and submit it to the server. Server will then generate bank file. Create its link to download and send it to the bank.

Alternative Flow

If the basic flow is not performed appropriately, System will generate an error message.

Pre-Condition

Salary calculations of the employees should be verified and stored in the database to generate bank file.

Post-Condition

Bank file will be generated and stored in the database.

3.5.9 Use case for Tax File

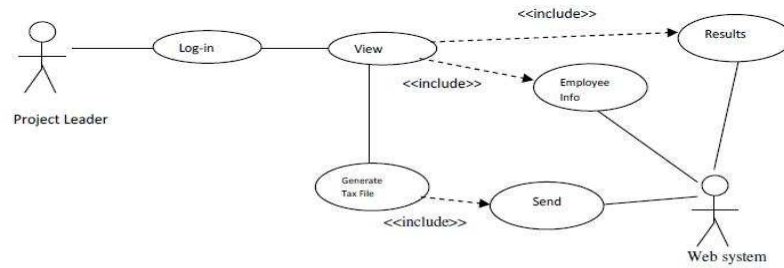


Figure 3.12: Use Case Diagram of Generation of Tax File

Scenario

User wants to generate the tax file for tax office.

Basic Flow

User will generate the tax file by the user interface for a particular month or year. Tax will be shown in the file according to the rules given by the tax office.

Alternative Flow

System will show the error message if database server is unavailable or down.

Pre-Condition

Tax should be calculated and stored in the database.

Post-Condition

Tax file will store in the database. [16]

Chapter 4

Implementation

In this chapter we will describe elements of development phase, technologies, tools that are required in the completion and implementation of our software project.

4.1 DreamWeaver MX 2004

DreamWeaver MX 2004 is a very powerful tool . It enables a user, either a beginner or professional, to develop a web application and to manage it. It has many beautiful features like support for PHP, java scripting, ASP. Net, HTML, XML, MySQL drag and drop feature. It provides GUI based layout tools, code editor and application development features.

Dreamweaver also has a powerful feature of CSS-based design and integration support. As from the above discussion we have come to a conclusion that dreamweaver is providing powerful features, hence we have choosed dreamWeaver MX 2004. [6],[7]

4.2 Apache 2.2.11

Apache web server 2.0 is widely used as HTTP(web) server which is an enhanced version of the previous ones. The reason of its popularity is due to its robustness, rich features, open source code and platform independence. The main reason to choose the apache is due to its platform independence which enables our system to run on any environment under any operating system. [8],[9]

4.3 PHP 5.2.9-2

4.3.1 History

PHP is a structured programming language, composition of variables, classes and functions like c++. It was first introduced in 1994 by Rasmus Lerdorf. It had some loose free ware scripts, based on perl and dubbed. At that time PHP was an abbreviation of personal home page. Now a days it called hypertext preprocessor. With the release of PHP version 3 in 1998, it appeared as a very popular and strong web developing tool in the web developer society.

PHP V.3 supports a lot of features such as, support to data bases of different platforms like ODBC, Oracle, mySQL, images, XML and host of many other technologies. Zend encoder is another powerful feature due to which PHP becomes more popular amongst the professional web developers. Therefore it enables the PHP source code to be encrypted. PHP V.5 is currently in use which is providing strong features of OOP. [10]

4.3.2 What is PHP

PHP is an open source server side scripting language which needs three things the php parser (CGI or Server module), a web server and a web browser to perform work. Another feature in it which is the reason of its popularity is platform independence, i.e., php can be used for all major operating systems including linux, many unix variants, microsoft windows etc.

It also supports most of the web servers today such as apache, microsoft IIS, netscape and many others. PHP supports for procedural, OOP with many code libraries as well as SAX and DOM standards. PHP also generates automatically and saves a variety of out put files other than HTML including PDF files, images, XML files and many more. More over a very strong and significant feature of PHP is its support for a wide range of databases. PHP can instantiate java objects to use them as PHP objects. [11], [12], [13]

4.3.3 Why PHP

PHP is popular as it is simple and easy to use, even for people with no programming experience can use it for designing and developing web applications. It also provides stability, compatibility (platform independence), working efficiently with a variety of web servers. PHP is an open source, it does not need any license and is available free.

We are using MySQL database and XML which is perfectly supported by PHP. We chose PHP because we found it to be a flexible, user friendly ,platform independent and it supports for a variety of web servers. [14]

4.4 MySQL 5.1.33

MySQL is an open source, multi-threaded, relational database management system created by Micheal Monty in 1995, it can be used by anyone for personal use. There are several reasons of its popularity among other freely available open source database systems such as availability, reliability, scalability, speed, security, performance. It is providing a strong back end support to web applications.

Moreover MySQL is providing a lot of built-in functions and also a variety of storage engines which are responsible to handle the queries between the back end storage database and a users SQL statement. That is why we choosed it in our project. [15], [17]

4.5 phpMyAdmin 3.1.3

PHP Myadmin is an open source tool that is used to administrate MySQL over world wide web. It supports most of the MySQL features such as managing databases, tables, fields and setting the privileges to the users, management of stored procedures and triggers . It can import data from SQL and CVS and can export it into various formats like CSV, SQL, PDF, XML etc.

More over it can administrate multiple servers, perform search operations in the database, generate complex quires using query-by-example(QBE) and a wide range of many other features.

As discussed above php MyAdmin is a useful tool for the administration of a database and since we are using MySQL in our project so, found phpMyAdmin very useful for our project. [18]

4.6 Software Development Architecture

In software engineering a system can be divided into multiple-layers either physically or logically providing a client-server environment known as n-tier architecture. The purpose of dividing the system is to provide a model for software developers developing flexible, reusable and more secure applications.

We applied 3-tier architecture in the development of our web application due to reliability, re-usability, flexibility and security. Making change in one layer has no effect on other layers, so, less work is required for maintenance and error correction.

4.6.1 Three-Tier Architecture

A 3-tier architecture comprises of following layers :-

- *Presentation/Client Tier*
- *Middle Tier*
- *Data Tier*

Presentation/Client Tier

This layer is on the top of application development hierarchy which enables a client to put some requests by putting a URL in the web browser or enter some data into a form. This layer fetches the results according to the request from the other tiers by communicating with them and presents it to the user.

Middle Tier

The layer that works as an agent between the data storage and presentation tier is known as middle tier. This layer also performs logical processing of the application and provides all the functions. In web applications, the middle tier consists of a web server which is responsible to get the request from client tier. It then sends it to data tier after doing appropriate processing on the request generated by the presentation tier, giving back results to the client tier.

Data Tier

In 3-tier architecture database server is considered as a third layer of the architecture. At this level data is stored and retrieved according to the requirements and requests sent by the presentation tier through middle tier. This layer is independent of other tiers due to which the performance and scalability of the system increases.

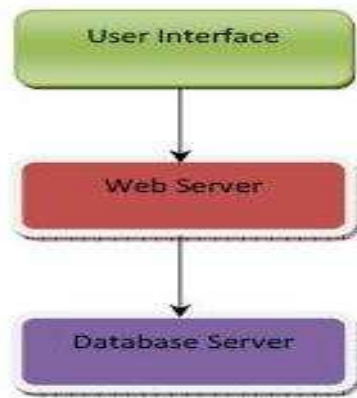


Figure 4.1: Three Tier Architecture of the System

4.6.2 Implementation of Three-Tier Architecture in the System

User Interface

It is easy to use and easy to understand user interface. User interacts with the web server through interface and performs different tasks like employee registration or salary calculation etc. Web browser is required to run the user interface.

Web Server

In our project apache is used as web server. Web server is used for secure communication between user interface and database server. Web server receives the request from client and sends them to the database. Similarly, it takes data from the database against the clients request and forwards it to the client.

Database Server

MySQL server is used in this project to handle database operations. The client does not directly interact with database, rather it goes through web server in order to keep the database secure from an unauthorized access. It is also called a third tier in the three tier architecture as implemented in this project.

4.7 Implementation of the System

This section will describe how the system is implemented using different web techniques.

4.7.1 Implementation of User interfaces

User interfaces are front end of any system with whom users interact to communicate with the system. There are two major users of this system that are team leaders and project managers. Hence we can divide the interface of our system into two major interfaces. Team leaders user interface and project managers interface.

After login into the system the users of the system are able to view only their authorized section, i.e., a Team leader can access the employee registration, employee management and salary calculation interfaces only. Where as the project managers can access interface which generates different reports other than the access of team leader interface.

Different web development techniques are used to develop and implement these interfaces such as XHTML, CSS, java scripting, ajax toolkit. With the help of these techniques we developed interactive and user-friendly interfaces.

4.7.2 Salary Calculation

Salary calculation interface is available for both team leader and project manager. This interface contains two drop down menus which are implemented using ajax. PHP code is used to fetch the data from the database after selecting particular employee from the list of employees that belong to a city which is selected from the drop down menu of city. After the fields filled out automatically the user can fill some fields manually that are optional such as allowances or deductions if any. PHP code is working behind the *calculate* button which will be activated and give result to user in response. After calculating the salary could be stored in the database or can be calculated again.

4.7.3 Drop Down Menus

All the drop down menus in the interfaces are generated dynamically using ajax and CSS. PHP and HTML code is used to implement these drop down menus which will interact with the database and fetch the required data from the database to drop down menus.

4.7.4 Form Validation

At user interfaces/front end of the system, different forms are available. The data entered into the fields of these forms by the users is validated by using the java scripting at the client side. After validation using the PHP code on client side interface, data is sent to the server side interface. Finally the data is sent to the database system to be stored.

4.7.5 File Handling

Different kind of files are required to generate such as salary slips, bank files/invoices or tax file. These files generate on server side interface by using the PHP code. This code interacts with the database, gets the required data and fills out the fields of the form which is developed using ajax. This form is then converted into the files with .pdf extension and can be saved into the system and can be sent via email. The whole process is done with the help of PHP code.

amgs_domains	
domain_ID	
domain	
comment	
language_ID	
userAgreement	
logoTop	
logoTopInvoice	
currencyAbbr	
VATInfo	
roundOffNearest	
spacingUnderInvoiceRows	
invoiceOurReference	
shortDateFormat	
includeAddressOnInvoiceTop	
domainOwner	
domainOwnerNoReplyEmail	
autoCreatePublicCity	
autoIncludeStandardInvoice...	
invoicePayCond	

Figure 4.3: Relational Table for Domain

AMGS _ City

The figure 4.4 is showing the relational table for cities, amgs _ city is capable of keeping the records of cities in its domain. City and city_ ID are representing the name and id for that city.

City_ ID is working as primary key (PK) in the table whereas country_ ID is working as foreign key(FK).

Creator_ ID is the id of administrator responsible to make any changes in the list of cities. *CreatecampResult* keeps the records of a camp in a city.

amgs_city	
city_ID	
city	
addDate	
changedDate	
creator_ID	
createcampResult	
city_type	
country_ID	

Figure 4.4: Relational Table for Cities

AMGS _ City _ User

Table in figure 4.5 contains data about the user (team leader) responsible for the data of a particular city. City_ ID and creator_ ID is working as foreign key (FK) and city_ use_ ID is working as primary key (PK) .



Figure 4.5: Relational Table for City Users

AMGS _ Employee

Table shown in figure 4.6, keeps the basic record of employees such as social security number, employment date, first name, last name, address, post code, city, country, phone number, mobile number, email address and status of the employment. Employee id's are assigned to each member of the organization, working as primary key (PK). whereas city is working as foreign key (FK).

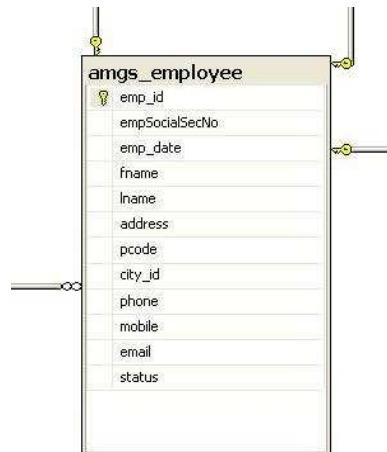



Figure 4.6: Relational Table for Employee

AMGS _ Emp _ Salary _ Info

Employee salary information after calculation will be stored in table showing in figure 4.7 . Each salary calculation is given a unique id, working as primary key(PK). Employee id (emp_ id) is working as foreign key(FK) ,used to identify a specific employee for whom the salary is being calculated. It (emp _ id) also identifies the status of the employee, i.e, if the employee is working on hourly or monthly wages.

Hours and *salPerHr* will keep the record of salaries of employees working on hourly basis. The salaries of those employees, working on monthly wages taking a fixed amount, will be calculated and stored under column *salPerMon*. The allowances added to the salaries or any deduction will be stored under *allowances* and *deductions*.



amgs_emp_salary_info_id
emp_id
salPerHr
hours
salPerMon
allowance
allowance_desc
deduction
deduction_desc
socialSecPerc
socialSec
salBefTax
taxPerc
tax
salAfterTax
comments
salDate
status

Figure 4.7: Relational Table for Employee Salary Information

4.8.1 AMGS _ Emp _ Bank _ Detail

Table shown in figure 4.8 will keep the records of bank accounts of employees while bank_ detail_ id will keep the record of the details regarding banks. Every bank is given a unique id (bank_ id) which is the primary key. Emp_ id is the foreign key (FK) in this table. Individual account details of the employees will be stored under columns *acctNo* and *clrNo*.



Figure 4.8: Relational Table for Bank Details

4.8.2 AMGS _ Banks

Table shown in figure 4.9 will keep the record regarding banks such as bank name under the column bank. Bank_ id is the foreign key whereas bank_ id and bank together are set as the primary key (PK) .



Figure 4.9: Relational Table for Banks

4.8.3 AMGS _ Emp _ Sal _ Tax

Figure 4.10 is showing a relational table stores all tax records upon employee salaries, after certain calculation. Emp_ sal_ tax_ id is set as the primary key (PK). Emp_ id has a relationship with amgs_ employee table therefore it is working as foreign key (FK). It can be used to access the data about any employee to calculate his/her tax.

Salary and salary _ type has a relationship with the amgs_ emp_ salary_ info which can be used to access the information about the salary type and amount of salary, both are working as foreign key (FK).

And on the basis of information taken from the amgs_ emp_ salary_ info table and amgs_ employee table we are able to calculate the percentage of the tax payable by the employee.

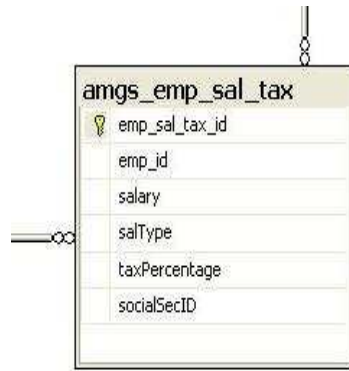


Figure 4.10: Relational Table for Employee Salary Tax

4.8.4 AMGS _ Social _ Security

Table shown in figure 4.11 will keep record of social security of the employees such as employee social ids, formulas for social security and percentage to calculate social security for the employees. Social_id is the foreign key (FK)



Figure 4.11: Relational Table for Social Security

4.9 System Description

In this section we will describe how *employee salary management system* is functioning. This system will be integrated with the *client management system* named as AMGS that is already functioning within the organization.

Employee salary management system is a web based system so at the front end, users of the system can access different graphical forms which can only be filled by the authorized. The data is given as the input to the forms and after passing through certain query, it will check and process the data and finally store it in the database which is working at back end of the system. And whenever the need arises the data can be fetched out from the database.

This system calculates the monthly or hourly salaries of its employees on the basis of certain parameters. It is also capable of calculating the tax for each individual and generate different kind of files, such as bank files or employee salary slips etc.

Moreover, the system is able to calculate the total amount of salaries, social security and tax of all the employees in one city for each month. It is also capable of calculating the salary and social security payable by the organization for a whole year. It then calculates the total amount of tax payable by the organization for every year.

All information is stored in the database of the system for the record of the organization and can be viewed any time it is needed.

4.9.1 Anatomy of the System

In this section we will explain the overall working of the system and functions. Each function will be discussed and explained individually as following:

After Logging in to the system a welcome screen appears where the user can view all the functions that he can be performed under the heading *Contents*, such as, logout, employee registration, employee management, salary calculator, bank file, employee pay slip, employee salary report, company/city salary report. All these functions are explained below.

4.9.2 Login

As the user enters in to the system the login screen appears, figure shown in 4.12. The user can view all the employees in the form of a list of a particular city, by selecting the city from the list of cities. User can view some basic information such as, person number, first and last name. To view the detail information about a specific employee the user can go to detail information

page of that employee by just clicking on the view detail option. The user can also view the functions that he can perform under the heading contents.



The image shows a user login interface. On the left, there is a 'CONTENTS' button and a 'Login' link. On the right, under the heading 'Welcome To Sweden Sport Academy', there are input fields for 'User ID:' and 'Password:'. Below these fields are 'Submit' and 'Reset' buttons, and a link for 'Forget Password ?'.

Figure 4.12: User Login

4.9.3 Employee Registration

Figure 4.13 is showing an employee registration form. It is viewed by clicking on employee registration from the contents. This function is used to register a new employee. The user can store all necessary information such as, name, person number, address in detail, bank details, contact information, information about salary, tax, social security and the status of employment, in the database.



The image shows an employee registration form. On the left, under the 'CONTENTS' heading, there is a list of links: Logout, Employee Registration, Employee Management, Salary Calculator, Bank File, Employee Pay Slip, Employee Salary Report, and Company/City Salary Report. The main form area, titled 'Welcome To Sweden Sport Academy', contains the following fields: PNo: 197901267865, Emp Date: 2010-06-10, First Name: Naeem, Last Name: Akhtar, Address: nyagatan, Post Code: 5555, City: V?xj? Sport Academy, Country: swedenportacademy.com, Phone: 111111, Mobile: 555555, Email: naeem@yahoo.com, Tax (%): 10, Salary Type: Hourly, Salary: 120, Bank Name: Swed Bank, Clearing No: 5555, Account No: 95845545, and Social Security: 1983 and before. At the bottom of the form are 'Submit' and 'Reset' buttons.

Figure 4.13: Employee Registration

4.9.4 Salary Calculator

Figure 4.14 is showing a view of salary calculator. It is accessible through the salary calculator from the contents. The user will select an employee from the list of employees from a city. He will fill in necessary information i.e, working hours, allowances, deductions, percentage of the tax and some comments. This will be the input to the system and the output is provided by the system as soon as the user will click on the *calculate* button at the bottom.

The screenshot displays a web application titled "Welcome To Sweden Sport Academy". On the left, a "CONTENTS" sidebar lists navigation options: Logout, Employee Registration, Employee Management, Salary Calculator (highlighted), Bank File, Employee Pay Slip, Employee Salary Report, and Company/City Salary Report. The main content area features a form for calculating salary. At the top, it includes a dropdown for "Stockholm Sport Academy" and a "SELECT Employee" dropdown with a "Fill" button. The form contains several input fields: "Salary Per Hour:", "Hours:", "Salary Per Month:", "Allowances:" (with a value of 0), "Allowances Desc:", "Deductions:" (with a value of 0), "Deductions Desc:", "Salary before Tax", "Tax (%)", "Salary after tax:", and "Comments:". Each of the "Desc:" fields has a small up/down arrow icon. At the bottom of the form are three buttons: "Calculate", "Submit", and "Reset".

Figure 4.14: Salary Calculator

4.9.5 Data Verification

In this form the employees belonging to a certain city can be viewed by clicking the drop down list of cities. The employees can be selected by clicking the check boxes and the data entered will be verified as soon as the verify button is clicked. It will be processed through the queries working and linked with the verify button.

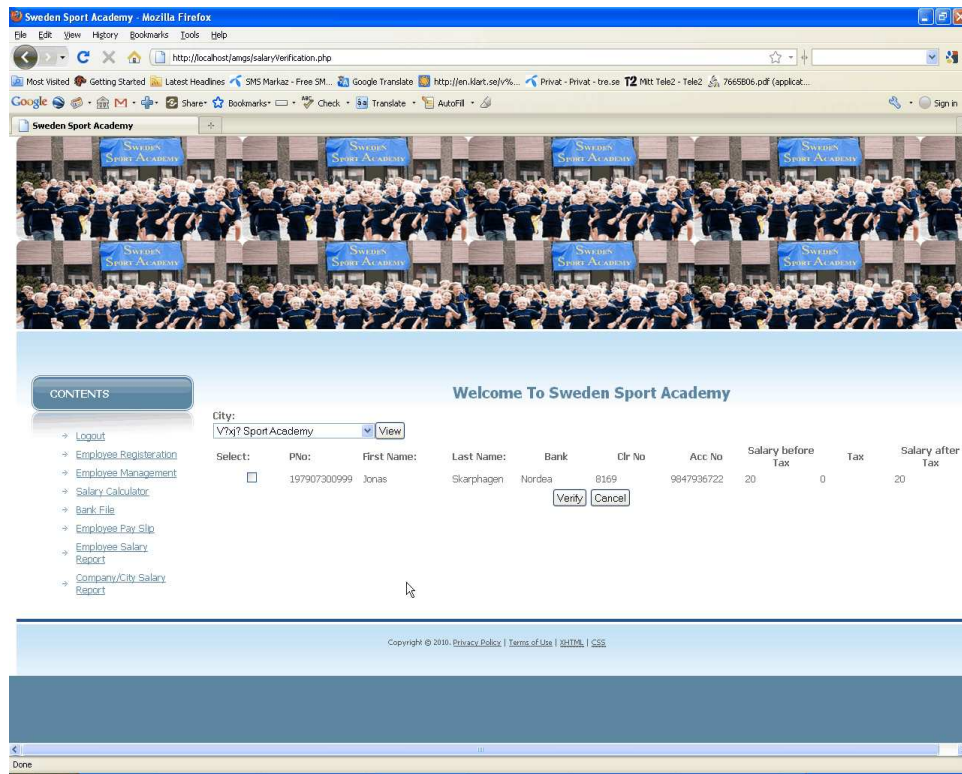


Figure 4.15: Emp Data Verification

4.9.6 Generate Employee Pay Slip

Figure 4.16 is a view of generating pay slip for an employee. Once a city is clicked from the drop down list, the list of all the employees can be accessed. The user will select an employee, a month and fill-in the year to generate the pay slip. As soon as the user clicks, *generate pay slip* button, the system will run some quires. After doing some final checks the output is generated in the form of pay slip.

Rest button is used to reset the form so that the pay slip for another employee can be generated.



City: Uppsala Sport Academy Employee: SELECT Employee Month: March Year: 2011

Generate PaySlip Reset

Figure 4.16: Generate Employee Pay Slip

4.9.7 Employee Salary Slip

Figure 4.17 is showing a view of salary slip of an employee. It contains all the necessary information such as, employee name, address, person number, working hours, organization name and number, total salary per month or per hour before tax, amount of tax, salary after tax and date of transfer of salary. The salary slips will be sent to the employees.



Salary Slip October

Employee: Jonas Skarphagen
Personal No: 197907300999
Address: Växjö
35252 V7xj7 Sport Academy
Employer: swedensportacademy.com
Org No: Not Available
Period: October
Hours: 1
Salary Per Hour: 20
Salary Per Month: 0
Salary Sent To Bank: 25 October

Salary	20
Tax	-0
Salary After Tax	20

Figure 4.17: Employee Pay Slip

4.9.8 Employee Salary Report

All the data about an employee including salary information and salary history is available by clicking on the employee salary report. As soon as the user clicks the *submit* button, a report regarding an employee arises on the screen. The report includes the personal information, salary, tax and bank details as shown in figure 4.18.

The report also shows a detailed history of the employee salary, tax information, social security for a whole year and for the previous months as a track record stored in the database of the system.

Welcome To Sweden Sport Academy

CONTENTS

- Logout
- Employee Registration
- Employee Management
- Salary Calculator
- Bank File
- Employee Pay Slip
- Employee Salary Report
- Company/City Salary Report

City: Employee:

Pin: 197907300999

Emp Date: 2010-09-01

First Name: Jonas

Last Name: Skarphagen

Address: Vaxjo

Post Code: 35252

City: Vaxjo Sport Academy

Country: Sweden

Phone: 123456

Mobile: 123456

Email: jonaa@hotmail.com

Bank Name: Nordea

Clearing No: 8169

Account No: 9847936722

Salary History

Salary per hour	20
Hours:	
Monthly Salary:	0
Allowances:	0
Allowances Desc:	
Deductions:	0
Deductions Desc:	
Social Security (%):	0
Social Security :	0
Tax (%):	0
Tax :	0
Salary after Tax:	20
Comments:	Testing

Year Salary Tax Social Security

2010	20	0	0
------	----	---	---

Months Salary Tax Social Security

October	20	0	0
---------	----	---	---

Copyright © 2010. Privacy Policy | Terms of Use | HTML | CSS

Figure 4.18: Employee Salary Report

4.9.9 Company/City Salary Report

User can access the company or city salary report form by clicking on the company/city salary report as shown in figure 4.19. The user can select either the organization or can select a city from the drop down lists. Certain queries will run as soon as the *submit* button is clicked and a report is generated.

This report includes the details of the salaries, total amount of tax and social security paid by the organization for the whole year such as year 2009 and year 2010 and for each month in a year as shown in the results.

The screenshot shows a web browser window titled "Sweden Sport Academy - Mozilla Firefox". The address bar shows the URL "http://localhost/amgs/compSalReport.php". The page features a header with the Sweden Sport Academy logo and a navigation menu on the left. The main content area is titled "Welcome To Sweden Sport Academy" and contains a form for selecting an organization and a city, with a "Submit" button. Below the form, a table displays salary, tax, and social security data for the years 2009 and 2010, and for the month of July.

Year	Salary	Tax	Social Security	Month	Salary	Tax	Social Security
2009	8500	1500	4500				
2010	26440	4380	9240	July	8500	1500	4500

Figure 4.19: Company/City Salary Report

Chapter 5

Conclusion and Future Enhancements

This chapter is based upon the conclusion of what we have done so far and how the system can be further enhanced with an increase in requirements.

5.1 Conclusion

The criteria set for goals is to generate files, deliver on time, performance and efficiency issues of the system.

The major goal we have achieved is to generate the bank files/invoices for the banks with salary information so that the banks are able to transfer the salaries of the employees into their accounts.

After developing the system, we calculated the salary and generated a bank file. The bank file was then sent to the bank and was practically done successful(see section 4.9.7).

To check the performance and efficiency of the system we used the WAPT 7.1 tool. It is a load, stress and performance testing tool for web sites and web applications. We created basic testing scenarios and hence got some meaningful results shown in figure 5.1. These results are showing that this web application is efficient and performing very well.

Summary

Profile	Successful sessions	Failed sessions	Successful pages	Failed pages	Successful hits	Failed hits	Total KBytes sent	Total KBytes received	Avg Response time, sec (with page elements)
Profile1	100	0	1800	0	1800	0	1248	11497	0.03(0.03)

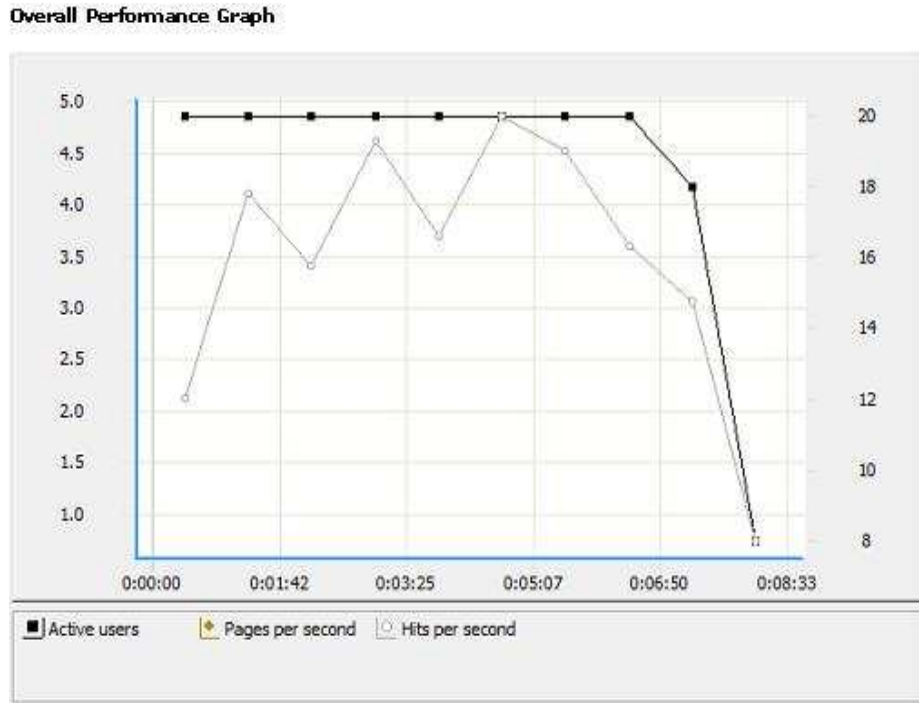


Figure 5.1: Overall Performance Graph

Our target was to deliver the system by the end of October 2010. Whereas, we finalized it by 15th October 2010 and delivered well on time.

The system we developed is more customized and specifically designed according to the requirements of our client and is able to integrate with the existing AMG's system to form one complete computerized system for the organization.

The system is able to calculate employees salaries according to certain parameters, such as age. It is also able to generate specific files for the organization such as employee salary slips and bank files.

The system is also able to calculate tax payable by the organization on the basis of certain formulas provided by the tax authorities according to the Swedish law. It generates and sends tax files to tax office, moreover, the system is more cost effective as compare to other salary management systems.

As this system is integrated with the existing system so does not need new resources to run the system. Therefore, it is easy to handle single system

instead of two.

Maintenance is easy and requires no further cost, further enhancement is possible according to the requirements of the organization which is not possible in the case of using existing salary management softwares available in the market.

Hence the system is developed successfully and is in proper function according to our goals and criteria.

5.2 Future Enhancements

5.2.1 Budget Program

This salary management program can be further enhanced by a budget program in future. In budget program every team leader will have support to manage and utilize specific amount of money in a better way. With this amount he will manage everything like stadium rent, office expenditures and employees salaries.

5.2.2 Visual Representation of Data

Graphs and charts can be added into future version of salary management program. Through these charts or graphs management can see the change in the salary of an employee or over all increment in salaries of employees. Management can also analyze the tax paid by company in different time periods.

Bibliography

- [1] <http://www.taskmanagementsoft.com/purchase/> "Last visiting date 20th March 2011"
- [2] <http://www.spinetechnologies.com/products/payrollindex.htm> "Last visiting date 20th March 2011"
- [3] <http://www.elucidata.com/refs/sdlc.pdf> "Last visiting date 20th March 2011"
- [4] <http://www.relativitycorp.com/projectmanagement/article10.html> "Last visiting date 20th March 2011"
- [5] <http://www.freetutes.com/systemanalysis/sa2-waterfall-software-life-cycle.html> "Last visiting date 20th March 2011"
- [6] <http://site.ebrary.com/lt.ltag.bibl.liu.se/lib/linkoping/docDetail.action?docID=10043887> "Last visiting date 20th March 2011"
- [7] <http://oit.wvu.edu/training/files/dw2004mx.pdf> "Last visiting date 20th March 2011"
- [8] <http://httpd.apache.org/docs/> "Last visiting date 20th March 2011"
- [9] <http://www.serverwatch.com/tutorials/article.php/3074841/IIS-vs-Apache-Looking-Beyond-the-Rhetoric.htm> "Last visiting date 20th March 2011"
- [10] <http://www.developer.com/lang/php/article.php/900521/Learning-PHP-The-Whats-and-the-Whys.htm> "Last visiting date 20th March 2011"
- [11] Luke Welling; Laura Thomson (2008). *PHP and MySQL Web Development, Fourth Edition*, Addison-Wesley Professional "Last visiting date 20th March 2011"
- [12] <http://se.php.net/manual/en/intro-what-is.php> "Last visiting date 20th March 2011"
- [13] <http://se.php.net/manual/en/intro-what-cando.php> "Last visiting date 20th March 2011"

- [14] <http://www.developer.com/lang/php/article.php/900521/Learning-PHP-The-Whats-and-the-Whys.htm> "Last visiting date 20th March 2011"
- [15] Russell Dyer (2008). *MySQL in a Nutshell, 2nd Edition*, O'Reilly Media, Inc. "Last visiting date 20th March 2011"
- [16] Tsui, Frank F. *Essentials of software engineering, second edition*, Jones and Bartlett Publishers. "Last visiting date 20th March 2011"
- [17] <http://www.mysql.com/why-mysql/topreasons.html> "Last visiting date 20th March 2011"
- [18] http://www.phpmyadmin.net/home_page/index.php "Last visiting date 20th March 2011"



Linnæus University

School of Computer Science, Physics and Mathematics

SE-351 95 Växjö / SE-391 82 Kalmar

Tel +46-772-28 80 00

dfm@lnu.se

Lnu.se