

Development of Payroll Management System for Lets Learn Coding

A Practicum Report Submitted By

C. M. Abdullah Khan

ID No. 15103195

A practicum report submitted in partial fulfillment of the requirements for the award of Bachelor of Computer Science and Engineering



Department of Computer Science and Engineering

College of Engineering and Technology

IUBAT—International University of Business Agriculture and Technology

Fall 2018

Development of Payroll Management System

for Lets Learn Coding

C. M. Abdullah Khan

A practicum report submitted in partial fulfillment of the requirements for the degree of Bachelor of Computer Science and Engineering (BCSE)

The practicum has been examined and approved,

Prof Dr Md Abdul Haque

Chair and Professor

Dept. of Computer Science and Engineering

IUBAT –International University of Business Agriculture and Technology

Prof Dr.Utpal Kanti Das

Coordinator and Professor

Dept. of Computer Science and

Engineering IUBAT –International University of Business Agriculture and Technology

Md. Mortuza Hossain

Lecturer, Dept. of

Computer Science and Engineering

IUBAT –International University of Business Agriculture and Technology

Department of Computer Science and Engineering

College of Engineering and Technology

IUBAT –International University of Business Agriculture and Technology

Fall 2018

Abstract

Payroll is a critical operation for every organization to pay employee accurately their salary and emoluments on time. For a big organization, the idea of taking control of employees pay calculations is quite daunting. This computer based payroll system aims to explain in simple terms what payroll involves and demystify the payroll processes. Moreover, it is a powerful tool to streamline the time consuming and complex tasks of employee payroll scheduling and planning. This system provides multiple user data access. Each user walks through the entire payroll process as per rights allocated from adding new employee to generate pay slips with clear step by step instructions. Furthermore, the system is flexible to implement changes in pay scales. This Computer based payroll system is capable of keeping a record of employee data including their pay, allowances, deductions on monthly bases so that fresh definitions are reflected from the month onwards, which leaves all the past data intact. This system has been developed using Spring MVC, Spring Security, Spring Core, Spring AOP and database has been designed using Mysql and Redis. The proposed computer based payroll system is advantageous as it provides a user friendly environment. This payroll system increases security and minimizes human calculation errors.

Letter of Transmittal

January 23, 2019

Chairman

Practicum and Placement Committee College of Engineering and Technology (CEAT)
IUBAT-International University of Business Agriculture and Technology 4, Embankment Drive
Road, Sector 10
Uttara Model Town, Dhaka -1230, Bangladesh.

Subject: Letter of Transmittal.

Dear Sir,

With due respect, this is my pleasure to present my project report entitled— Development of Payroll Management System for Lets Learn Coding. I have prepared this report as partial fulfillment of the practicum. I have tried my level best to prepare this project to the required standard.

It was certainly a great opportunity to work on this paper to actualize my theoretical knowledge in the practical arena. Now, I am looking forward or your kind appraisal regarding this practicum report.

I shall remain deeply grateful to you if you kindly go through this report and evaluate my performance. I hope that you would find the report comprehensive and competent augmented.

Sincerely yours,

C.M. Abdullah Khan
ID: 15103195
Program: BCSE

Student's Declaration

I hereby declare that this practicum report titled ‘Development of Payroll Management System’ is my original work. It has never been presented previously or concurrently for any other purpose, reward or degree at IUBAT University or any other institutions either by me or by any other student. I also declare that there is no plagiarism or data falsification and materials used in this report from various sources have been duly cited.

C.M. Abdullah Khan

ID: 15103195

Program: BCSE

Acknowledgement

In the name of ALLAH who is the most merciful and the most graceful. It's my pleasure to take this occasion to thank a few people, who have assisted, encouraged, directed and supported me throughout my practicum program. First of all, I want to thank my parents, who have endowed their immeasurable-innumerable support and encouragement to attain this exquisite event of my life.

I sincerely would like to thank Sumon Molla Selim (Founder and CEO, Lets Learn Coding Ltd.) and Md. Touhidul Islam Foysal (Technical Lead, Lets Learn Coding Ltd.) for guide me through the whole project and giving me the opportunity to complete my internship at Lets Learn Coding Ltd and project at IUBAT UNIVERSITY.

My sincere and outmost thank goes to Prof. Dr. Utpal Kanti Das, Coordinator of Department of Computer Science and Engineering, IUBAT- International University of Business Agriculture and Technology. For his continuous encouragement and contribution gave me the courage, the determination needed to able to finish the internship well.

I will always remain thankful for the advice and suggestion provided my Supervisor Md. Mortuza Hossain Lecturer, Department of Computer Science and Engineering.

This report would not have been possible without the essential and gracious support of many individuals who encouraged me to complete this project on time.

Dedication

This work is dedicated to my honorable advisor Md. Mortuza Hossain sir, this work wouldn't have been possible without his discussion, help and support. It is also dedicated to my parents, who always support me & encouraged me.

Supervisor's Declaration

I certify that the student C. M. Abdullah Khan (ID No.15103195) carried out his practicum work ‘Payroll Management System’ at International University of Business Agriculture and Technology (IUBAT) between September 5, 2018 and December 15, 2018. During this period, he consulted me on a regular basis as required by the department. I, therefore, recommend that his practicum report be accepted for oral examination.

Md. Mortuza Hossain

Lecturer, Dept. of Computer Science and Engineering

IUBAT – International University of Business Agriculture and Technology

Department Certification

On behalf of the Department of Computer Science and Engineering of International University of Business Agriculture and Technology (IUBAT University) we, the undersigned, certify that this practicum report ‘Development of Payroll Management System’ for the award of Bachelor of Computer Science and Engineering (BCSE) degree was duly presented by C. M. Abdullah Khan (ID No.15103195) and accepted by the department.

Prof. Dr. Utpal Kanti Das

Co-Supervisor

Md. Mortuza Hossain

Supervisor

To Whom It May Concern

Table of Content

Practicum Report Title -----	i
Abstract -----	ii
Letter of Transmittal -----	iii
Student's Declaration -----	iv
Acknowledgement -----	v
Dedication -----	vi
Supervisor's Certification -----	vii
Department Certification -----	viii
Organization Certification -----	ix
Table of Contents -----	x-xi
List of Figures -----	xii-xiii
List of Tables -----	xiv
Chapter 1 Organization -----	1-4
1.1. Organization Overview -----	2
1.2. Mission -----	2
1.3. Vision -----	2
1.4. Service provided -----	2
1.5. Organization Structure -----	3
1.6. Location -----	4
Chapter 2 Introduction -----	5-12
2.1. Introduction -----	6
2.2. Background Study -----	6
2.3. Objective -----	7
2.3.1. Board Objective -----	7
2.3.2. Specific Objective -----	7
2.4. Proposed System -----	7
2.5. Methodology -----	7
2.5.1. Data Source -----	8
2.6. Limitation of project -----	8
2.7. Process Model -----	8
2.7.1. Feature and Advantage of Extreme software model -----	10
2.8. Feasibility Study -----	11
2.8.1. Technical Feasibility -----	11
2.8.2. Economical Feasibility -----	12
2.8.3. Operational Feasibility -----	12
Chapter 3 Requirement Engineering -----	13-20
3.1. Requirement Engineering -----	14
3.2. Requirements Analysis -----	14
3.2.1. User Requirement -----	14
3.2.2. Functional Requirement -----	17
3.2.3. Non Functional Requirement -----	18
3.3. Use case Diagram -----	20
Chapter 4 System Planning -----	21-36

4.1. Project Planning -----	22
4.2. Project Estimation -----	22
4.3. Function of Proposed System -----	22
4.4. Function Description -----	23
4.5. Function Point Estimation-----	25
4.5.1. Performance and Environmental -----	30
4.6. Processed Based Estimation-----	31
4.7. Effort Distribution -----	32
4.8. Task Scheduling -----	33
4.9. Project Cost Estimation -----	34
Chapter 5 Analysis -----	37-45
5.1. Activity Diagram -----	37-41
5.2. Swim lame Diagram -----	42-45
Chapter 6 Designing -----	46-53
6.1. Entity Relationship Diagram -----	47-48
6.2. Dataflow Diagram -----	49-53
Chapter 7 Risk Management -----	54-58
7.1. Risk Engineering -----	55
7.2. The RMMM Plan -----	58-
Chapter 8 Interface Design -----	59-70
8.1. Interface Design -----	60-66
8.2. Database Field Design -----	66-70
8.2.1. Database -----	66
8.2.2. Database for Table -----	66
8.2.3. Database for Userinfo -----	66
8.2.4. Database for Salary -----	67
8.2.5. Database for Loan paid Detail -----	67
8.2.6. Database for loan -----	68
8.2.7. Database for Leave User -----	68
8.2.8. Database for Grade -----	69
8.2.9. Database for attendance -----	69
Chapter 9 Quality Assurance and Testing -----	71-78
9.1. Software Testing Strategy -----	72
9.2. System Testing Methodology -----	73
9.3. Testing Design -----	74
Chapter 10 Conclusion & Upcoming Feature -----	79-84
10.1. The Conclusion -----	80
10.2. The Experience -----	81
10.3. Upcoming Feature -----	81
10.4. Glossary -----	83
References -----	84

List of Figures

Figure 1.1:	Organizational Process -----	3
Figure 1.2.:	Company Location -----	4
Figure 2.1:	Incremental Process Model -----	10
Figure 3.1:	Use case diagram -----	20
Figure 4.1:	Effort Distribution -----	33
Figure 5.1.1:	Activity Diagram for Manager -----	39
Figure 5.1.2:	Activity Diagram for Accountant -----	40
Figure 5.2.1:	Manager can enroll new roll user -----	42
Figure 5.2.2:	Admin can Approve or deny loan request -----	43
Figure 5.2.3:	Admin can approve or deny loan request -----	44
Figure 5.2.4:	Admin can give permission to pay salary -----	45
Figure 6.1:	Entity Relationship Diagram -----	48
Figure 6.1.1:	Level 0 DFD -----	49
Figure 6.1.2:	Level 1 DFD -----	49
Figure 6.1.3:	Level 2 Process 1 DFD -----	50
Figure 6.1.4:	Level 2 Process 2 DFD -----	50
Figure 6.1.5:	Level 2 Process 3 DFD -----	51
Figure 6.1.6:	Level 2 Process 4 DFD -----	51
Figure 6.1.7:	Level 2 Process 5 DFD -----	52
Figure 6.1.8:	Level 2 Process 6 DFD -----	53
Figure 8.1.1:	Interface for login -----	60
Figure 8.1.2:	Interface for home -----	60
Figure 8.1.3:	Interface for Registration -----	61
Figure 8.1.4:	Interface enable disable employee -----	61
Figure 8.1.5:	Interface accept/deny leave request -----	62
Figure 8.1.6:	Interface accept/deny loan request -----	62
Figure 8.1.7:	Interface for send loan request -----	63
Figure 8.1.8:	Interface for send leave request -----	63
Figure 8.1.9:	Interface for give attendance -----	64
Figure 8.1.10:	Interface for Admin dashboard -----	64
Figure 8.1.11:	Interface for search information -----	65
Figure 8.1.12:	Interface for salary payment permission -----	65
Figure 8.2.1:	Database -----	66
Figure 8.2.2:	Database table -----	66
Figure 8.2.3:	Database for userinfo -----	67
Figure 8.2.4:	Database for salary -----	67
Figure 8.2.5:	Database for loan paid details -----	68

Figure 8.2.6: Database for loan -----	68
Figure 8.2.7: Database for leave users -----	69
Figure 8.2.8: Database for attendance -----	70
Figure 9.1: Unit test report -----	73
Figure 10.3.1: Github repository -----	82

List of Tables

Table 4.1:	Identifying Complexity (Transaction Function) -----	26
Table 4.2:	Identifying Complexity (Data function) -----	27
Table 4.3:	Unadjusted Function Point Estimation (Transaction Function) -----	28
Table 4.4:	Unadjusted Function Point Estimation (Data Function) -----	30
Table 4.5:	Performance and Environmental Impact -----	30
Table 4.6:	Process Based Estimation -----	31
Table 4.7:	Project Time Scheduling chart -----	34
Table 4.8:	Project Hardware Cost -----	35
Table 4.9:	Project Software Cost -----	35
Table 4.10:	Project Other Cost -----	35
Table 4.11:	Project Personnel Cost -----	36
Table 4.12:	Project Account Cost -----	56
Table 7.1:	Technology Risk -----	56
Table 7.2:	People Risk -----	57
Table 7.3:	Organizational Risk -----	57
Table 7.4:	Tools Risk -----	57
Table 7.5:	Requirement Risk -----	57
Table 7.6:	Estimation Risk -----	74
Table 7.7:	Risk Planning -----	74
Table 9.1:	Testing scenario 1 -----	75
Table 9.2:	Testing scenario 2 -----	75
Table 9.3:	Testing scenario 3 -----	75
Table 9.4:	Testing scenario 4 -----	75
Table 9.5:	Testing scenario 5 -----	76
Table 9.6:	Testing scenario 6 -----	76
Table 9.7:	Testing scenario 7 -----	76
Table 9.8:	Testing scenario 8 -----	77
Table 9.9:	Testing scenario 9 -----	77
Table 9.10:	Quality Assurance Matrix 1 -----	77
Table 9.11:	Quality Assurance Matrix 2 -----	77
Table 9.12:	Quality Assurance Matrix 3 -----	77
Table 9.13:	Quality Assurance Matrix 4 -----	77
Table 9.14:	Quality Assurance Matrix 5 -----	78
Table 9.15:	Quality Assurance Matrix 6 -----	78
Table 9.16:	Quality Assurance Matrix 7 -----	78
Table 9.17:	Quality Assurance Matrix 8 -----	78

Chapter 01

The Organization

1.1 Organizational Overview

Let's Learn Coding is incorporate in late 2012 has emerged as a premier ICT Solution Provider in Bangladesh successfully serving customers around the World. LLC mainly focuses on customized Web/Mobile/Software Development and E-commerce Solution service .

Let's Learn Coding, a trusted name for software service in Bangladesh specialized in providing desktop app, mobile app and web app. Let's Learn Coding is opening the door to educate and empower the youth with the knowledge and power of coding. They provide the best coding and programming related training of the country. They also organize free workshops, seminars and inspirational/motivational talks on a regular basis on different technology topics to promote coding.

1.2 Vision

We firmly believe in the philosophy of 'Make our client happy '. We take pride in a team of highly qualified, skilled and motivated Professionals who are encouraged to lead, innovate and excel. Our team consists of top professionals who share a common vision and passion, providing our clients with critical insights and advise to succeed in today's competitive environment. We believe in delivering Expertise, Excellence Services through our past Experience and providing the highest and best end use of services to our client.

- ✓ To build a trusted IT Companies in Bangladesh
- ✓ To be the largest Software Companies in Bangladesh
- ✓ To be the best choice for people when they like to Software Development
- ✓ To be the largest SEO (search engine optimization) Marketing Company in Bangladesh To be the largest IT company in World.

1.3 Our Mission

is to Inspire, Facilitate and Assist in the growth of all our customers we work with through diversified Automation Systems that will ensure Efficiency, Cost Minimization, Resource Utilization, Quality and Effectiveness. We believe in the following values:

- ✓ Integrity.
- ✓ Innovation.
- ✓ Focus.
- ✓ Efficiency.
- ✓ Customer Satisfaction.

- ✓ Reduced costs that are highly competitive.
- ✓ Cost effective solutions that affect your bottom line in a positive manner.
- ✓ Decrease your time to market by using teams that are more efficient.
- ✓ Superior quality finished application.
- ✓ To consistently enhance our competitiveness and deliver profitable growth.
- ✓ To practice highest standards of corporate governance and be a financially sound company.
- ✓ To be a partner in nation building and contribute towards Bangladesh economic growth.
- ✓ To encourage ideas, talent and value systems and become the customers of choice.
- ✓ To earn the trust and confidence of all customers, exceeding their expectations.

1.4 Organizational structure & Team Structure

The following figure shows the organizational structure of Lets Learn Coding. Founder and chairman is the head of the managing department. Under founder HR and technical director is worked. Under them, others worker is worked.

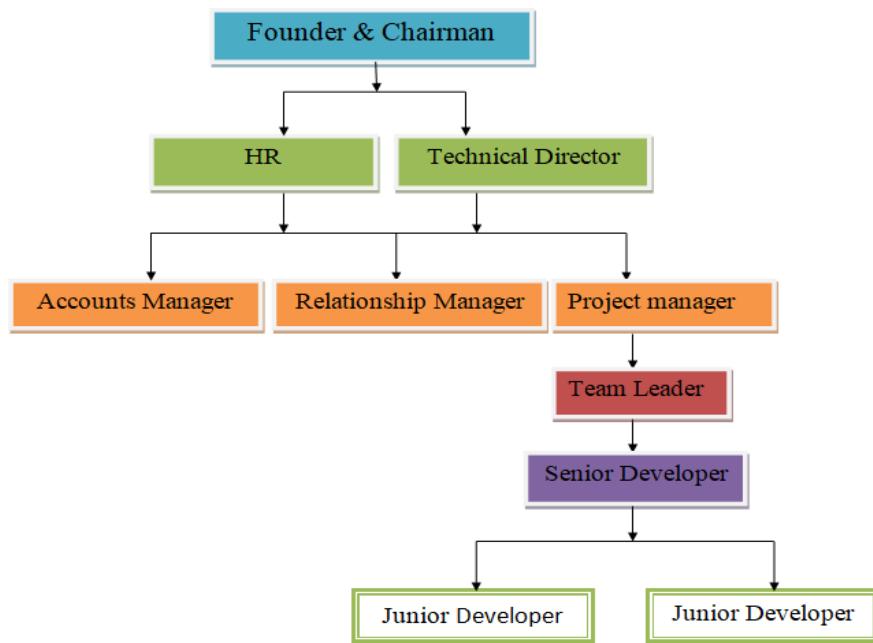


Figure 1.1: Organizational Structure

1.5 Location

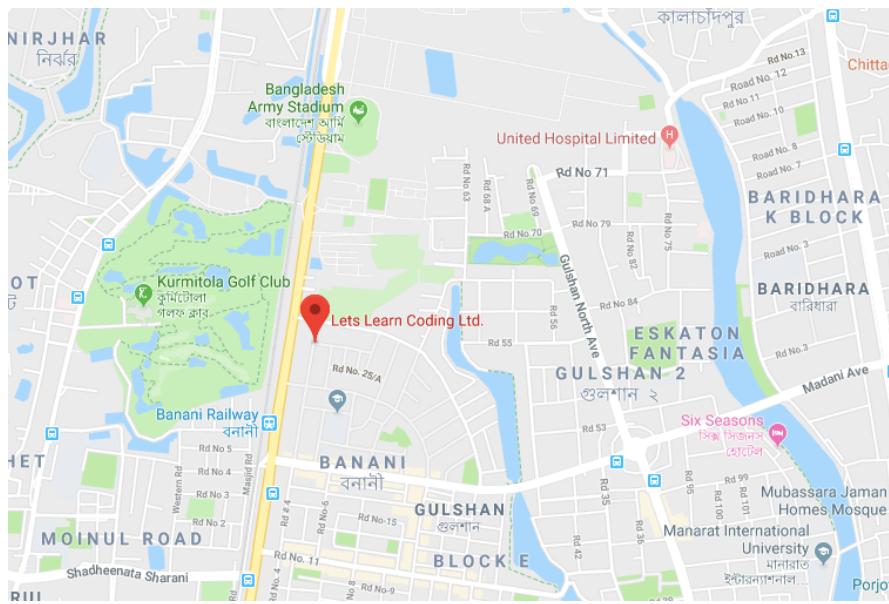


Figure 1.2: Company Location

Address : House 99, Road 4, Block A, Dhaka 1213

Web: letslearncoding.co

Phone: 01979-787771

Chapter 02

Introduction

2.1 Introduction

In an organization, there are several departments and each department has a payroll section to manage its payroll activities. Each section has to perform necessary personnel operations like data collection and preparation, entry, updates, monitoring and reporting of data. Many of these existing practices and procedures need to be reassessed at this time of changing needs, changing demands of employees and changing technologies. So an organization needs an integrated computer based payroll system that would integrate personnel and payroll related data collection, processing in individual departments and payroll audit in a more efficient and streamlined way. With this computer based payroll system, payroll section would be able to keep a record of employees including their personnel data, allowances, deductions, bonus etc. Net pay of each employee is calculated by using the information about his allowances. All these complex calculations have been incorporated in here. Many optional allowances and deductions for the employees are added by specifying the corresponding details. The individual pay slips and deduction vouchers are printed out as a receipt. Salary bonus , allowances, deduction are updated if there is any amendment in salary structure. This proposed computer based payroll system has been developed in accordance with requirements of Let's Learn Coding. Presently a clipper based payroll system is being used at Let's Learn Coding which has to be separately installed in a remote server. This proposed system is secure, easy to use and provides online access to Let's Learn Coding employees from any web enabled computer at any time. The computer based payroll application is a web-based design.

2.2 Background Study

Having an efficient payroll management system also makes it easier to manage employee information, so there's little need to consult various sources for data. Having all the information in one place means that reports generation is a breeze. Information about attendance, leaves, payroll, and the like can be accessed in one place. This is useful for tax compliance and legal documentation, as well as during audits where specific reports are required and must be submitted within tight deadlines

2.3 Objectives

The client uses MS Excel and maintains their records however it is not possible for them to share the data from multiple system in multi user environment, there is lot of duplicate work, and chance of mistake. When the records are changed they need to update each and every excel file. There is no

option to find and print previous saved records. There is no security anybody can access any report and sensitive data also reports of summary. This Payroll Management System is used to overcome the entire problem which they are facing currently, and making complete atomization of manual system to computerized system.

2.3.1 Broad Objectives

Payroll system software can eliminate errors in the payroll process and eliminate excessive effort involved in calculating employee hours, salary, and bonus. Payroll software is easy to use and often times very affordable for company.

2.3.2 Specific Objectives

The specific objective of our project is to prepare a record of all employees working in a company.

- ✓ Personal Record of all Employees.
- ✓ Salary Calculation of all Employees.
- ✓ Grade Assigning to all Employees.
- ✓ Calculate Monthly total Salary of department.
- ✓ Record Attendance.

2.4 Proposed System

The proposed software will solve all the problems they are facing now. This software is designed such way that it will generate the salary by providing managers permission every month in time. So there not much.

This software also equipped with the facility of checking the employees to whom no salary has been sanctioned. The software built to generate summary of the payroll.

2.5 Methodology

The development process on “Payroll Management System through Incremental Process Model” will complete following the structure described later on Software Analysis & Design.

This study on “Payroll Management System” through Incremental Process Model” is tentative in nature. It aims to development of management System. The variables identified to manipulate through a handy inspection and from primary and secondary data.

2.5.1 Data Sources

For this project in data collection phase we collected two types of data i.e. Primary Data

Primary Data: That are generated by collecting data from organization.

Primary data are generated within the organization. The organizations practical experience, observation, and face-to-face interview with our own web administrators helped us generate the primary data.

Secondary Data

Secondary data are generated by studying different articles, newspapers, research papers and of course information collected via Internet. Data, facts and statistics collected from different web sites and sources made us understand the project better.

2.6 Limitations of project

This project is made for testing the System Analysis and designing subject part. So we get only 14 weeks to make the software. So there are some limitations and boundaries. Here we imply our theoretical knowledge and make a testing software of Payroll Management System. Here we use SDLC model to make software and try to reserve documentation properly. As well we get little time to make the software and try to make it properly as well as professionally. Moreover, many problems have been arisen in the data collection for the software and also have load shading. Now we try to see the proper scenario of this project.

2.7 Process Model

A process of model software alternates sequence for the whole life time. So it is sometimes called the product life cycle. In many cases the initial software requirements are clearly defined, but overall development opportunities the effort precludes a complete linear process, in addition, in the case of this, the organization selected a process model, in addition to providing a limited set of software functionality and then the release of software that can be perfect and extended to a compulsory requirement and that is designed to produce evolutionary model software. The “Incremental process model” is a software development method that is combination of iterative development process model and sequential linear development model.

This phase also includes understanding the system requirements by continuous communication between the customer and the system analyst. At the end of the model, the product is deployed in the identified market.

Design

The Design phase starts with the conceptual design in the baseline incremental and involves architectural analysis, logical design of modules, physical product code conversion and the final test in the subsequent incremental model.

Requirements of Software are first broken down into several modules that can be incrementally constructed and delivered. At any time, the plan is made just for the next increment and not for any kind of long term plans. Therefore, it is easier to modify the version as per the need of the customer. Development Team first undertakes to develop core features (these do not need services from other features) of the system.

Once the core features are fully developed, then these are refined to increase levels of capabilities by adding new functions in Successive versions. Each incremental version is usually developed using an iterative waterfall model of development.

As each successive version of the software is constructed and delivered, now the feedback of the Customer is to be taken and these were then incorporated in the next version. Each version of the software have more additional features over the previous ones.

Code

The Code phase refers to production of the actual software product at every incremental phase. In the baseline incremental, when the product is just thought of and the design is being developed a POC (Proof of Concept) is developed in this phase to get customer feedback.

Then in the subsequent spirals with higher clarity on requirements and design details a working model of the software called build is produced with a version number. These builds are sent to the customer for feedback.

Evaluation and Risk Analysis

Risk Analysis includes identifying, estimating and monitoring the technical feasibility and management risks, such as schedule slippage and cost overrun. After testing the build, at the end of first iteration, the customer evaluates the software and provides feedback.

The following illustration is a representation of the Incremental Model, listing the activities in each phase.

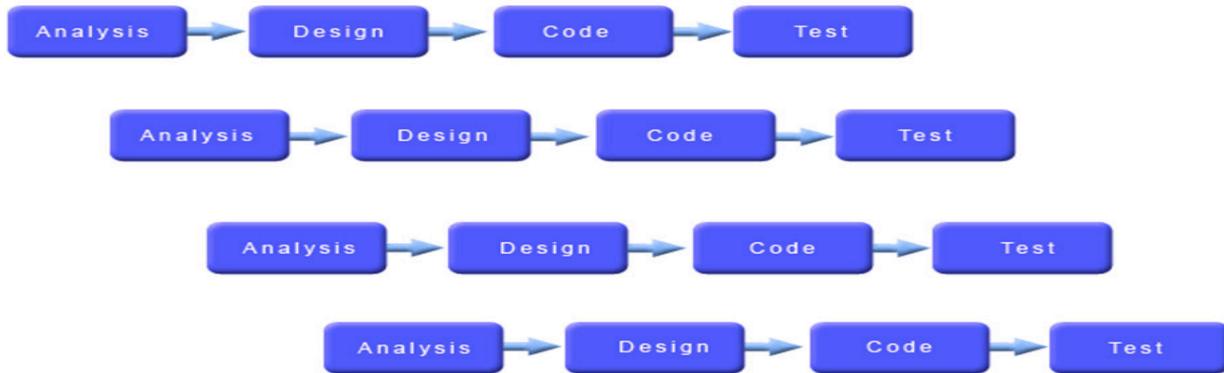


Figure 2.1: Incremental Process Model

Figure 2.1 shows the incremental process model as this project is too much sensitive for costs and security. As here there need to tight security so this analysis and planning will be change any time. Because this model has four phases: analysis phase, design analysis phase, code phase and test phase. So when any requirements will come we easily analysis and edit. In first to last all team members are working at a time. It allows for incremental releases of the product, or incremental refinement through each of the iteration around the model. To use this model I easily analysis the requirement information and in parallel time I will work with my interface, design, get the clients recommendation and thesis about the market value and security. All work will be done at a time and make a good communication with client. This process model always run in thesis based when any problem found its will change, so following this model we need to more time to build a final version of project.

2.7.1 Features and Advantages of Extreme software model

The advantages of this process model describe below

- ✓ Development process is too re usable from other models because here developer and customer have direct interaction to build a system.
- ✓ Customer can see the working product at the early stages of software development life cycle
- ✓ Project can be separated into several parts, and more risky of them can be developed earlier which decreases management difficulties
- ✓ Larger projects / software are created and handled in a strategic way. For my project its too much helpful because this project always run in market analytical way.

- ✓ Risk evaluation is proper so there is no way for overlap the budget estimation and don't think about the market priority.
- ✓ Easily add many more features in a systemic way if it is valuable for market oriented and user friendly
- ✓ Has room for customer feedback and the changes are implemented faster.
- ✓ Risk reduction mechanisms are placed in this system.

2.8 Feasibility Study

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its work ability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus when a new application is proposed it normally goes through a feasibility study before it is approved for development. The document the feasibility of the project that is being design and lists various areas that were considered very carefully during the feasibility study of this project such as technical, economical and operational feasibilities. The following are its features:

2.8.1 Technical feasibility

The system must be evaluated from the technical point of view first. The assignment of this feasibility must bases on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified and outline system, the investigation must go to suggest the type of equipment, required method developing the system, of running the system once it has been designed. Technical issues are raised during the investigation are:

- ✓ Does the existing system technology sufficient for the investigation are?
- ✓ Does the existing technology sufficient for the suggested one?
- ✓ Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within latest technology. Through the technology may become obsolete after some period of time, due to the fact that never version of same software supports older versions, the system may still be used. So there is minimal constant involved with

this project. The system has developed using Spring MVC and Spring Security the project is technically feasible development.

2.8.2 Economical feasibility

The developing system must be justified by the cost and benefit. Criteria to ensure that effort is concentrated on the project, which affect the development of a new system, is the cost it would require. The following are some of the important financial questions asked during preliminary investigation:

- ✓ The cost conducts a full system investigation.
- ✓ The cost of hardware and software.
- ✓ The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also, all the resources are already available, it gives an indication of the system is economically possible for development. Our developed system is economically feasible. When we compare the total cost and benefits from the system that time we see that our client will be beneficial from this system.

2.8.3 Operational feasibility

This includes the following questions:

- ✓ Is there sufficient support for the users?
- ✓ Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and deployed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible. The function is user friendly like manager can easily register employee disable and enable user. User can easily operate the proposed system because the system is user friendly. If the staff of the organization has the basic computer knowledge they could operate the software easily. Every features and the activity that are combined within the system is designed and developed belongs to previous format they had used with a more attractive user interface.

Chapter 03

Requirement Engineering

3.1 Requirement Engineering

Requirement engineering encompasses the tasks that lead to an understanding of what the business impact of the software will be, what the user wants, and how end-users will interact with the software. Requirements engineering is defined in terms of its major activities.

- ✓ Understanding problems
- ✓ Solution determination
- ✓ Specification of a solution that is testable, understandable, maintainable and that satisfies project quality guidelines

Designing and building an elegant computer program that solves the wrong problem serves no one's need. That's why it is important to understand what the customer wants before we begin to design and build a computer-based system.

3.2 Requirement Analysis

Requirement analysis provides the software designer with a representation of information, function and behavior that can be translated to data, architectural, interface and component-level designs.

In the following task phases the requirement analysis was done.

3.2.1 User Requirement

User requirement is most important and first step of software. It decides how the software will behave. From there we get the specification for the software. Define requirement engineering maintain documentation and requirements and this process refers to the process of related engineering and software engineering subfields. Designing and building is an elegant computer program that fixes a wrong problem but it doesn't work for any needs. That's why it is important to understand what the customer wants before we begin to design and build a computer-based system. Essential engineering that works which refers to the business impact of the software, what customers want, and how the end users communicate with the software. Requirements engineering is defined in terms of its major activities-

- ✓ Understanding problems
- ✓ Solution determination
- ✓ Specification of a solution that is testable, understandable, maintainable and that satisfies project quality guidelines

The necessary engineering is closely related to software engineering, which emphasizes the process of users' systems that are designed.

- ✓ User Requirements
- ✓ System Requirements
- ✓ Functional Requirements
- ✓ Non-Functional Requirements

User Management

User requirement:

- ✓ User can login, view profile.

System Requirement:

- ✓ First user login with username and password.
- ✓ If username and password is valid user can view his dashboard.
- ✓ From there he can view profile, send loan request, leave request.

Role Manager:

User Requirement:

1. Manager can able to add employee.
2. Manager can able to set salary allowance.
3. Manager can able to approve/decline loan.
4. Manager can able to approve/decline leave.
5. Manager can able to send salary conformation notice.
6. Manager can able to enable disable user.
7. Manager can able to see all information.

System Requirement

1. Manager can able to add employee.
 - 1.1 Manager can able to see total employee.
 - 1.2 Manager can able to get unique employee id during inset employee.
 - 1.3 Manager can able to get total amount of salary.

2. Manager can able to set salary allowance.
 - 2.1 Manager can able to entry home allowance.
 - 2.2 Manager can able to entry transport allowance.
 - 2.3 Manager can able to entry medical allowance.
3. Manager can able to approve/decline loan.
 - 3.1 Manager can able to accept loan application.
 - 3.2 Manager can able deny loan application.
4. Manager can able to approve/decline leave.
 - 4.1 Manager can able to accept leave application.
 - 4.2 Manager can able to deny leave application.
5. Manager can able to send salary conformation notice.
 - 5.1 can able to send salary permission given notice.
6. Manager can able to enable disable user.
 - 6.1. Manager can able to enable disable user.
7. Manager can able to see all information.
 - 7.1 Manager can able to see all employee attendance in a sheet.
 - 7.2 Manager can able to see all employee attendance in individually.

Role Employee

User Requirement:

1. Employee can able to view profile.
2. Employee can able to view salary details.
3. Employee can able to apply for loan.
4. Employee can able to apply for leave.
5. Employee can able to add attendance.
6. Employee can able to view loan details.

System Requirement

1. Employee can able to view profile.
2. Employee can able to view salary details.
3. Employee can able to apply for loan.
4. Employee can able to apply for leave.

5. Employee can able to add attendance.
6. Employee can able to view loan details.

Role Accountant:

User Requirement:

1. Accountant can able to give bonus.
2. Accountant can able to give salary.
3. Accountant can able to see employee full salary details.

System Requirement

1. Accountant can able to give bonus.
 - 1.1 Accountant can able to give bonus in every eid festival.
2. Accountant can able to give salary.
 - 2.1 Accountant can able to give salary only for enabled users.
3. Accountant can able to see employee full salary details.
 - 3.1 Accountant can able to employee full salary details in paid month.

3.2.2 Functional Requirements:

A functional requirement document defines the functionality of a system or one of its subsystems. It also depends upon the type of software, expected users and the type of system where the software is used.

Manager

1. Add Employee.
2. Disable Employee.
3. View all Employee details.
4. View pending loan list.
5. Approve loan.
6. Decline loan.
7. View Pending leave list.
8. Approve leave.
9. Decline leave.

10. See attendance sheet.

11. Login.

12. Logout.

Employee

1. View personal details.

2. See Loan Details.

3. Apply for loan.

4. See Leave Details.

5. Apply for leave.

6. Input attenance.

7. See salary details.

8. Print pay slip.

9. Login.

10. Logout.

Accountant

1. View personal details.

2. Give salary.

3. Add bonus.

4. Login.

5. Logout.

3.2.3 Non-Function Requirement

Security requirements:

1. Each user has his own account.

2. Users have to login with his own password.

3. Password will be encrypted.

4. User name & password for admin, customer and technician saved in database for security purpose.

5. Usability, reliability & availability requirements must be ensured.

3.3 Use Case Diagram

It's one of the easiest to use case diagrams is to communicate with a user that shows the relationship between user or actor and the various uses related to the user. Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions (use cases) that some system or systems should or can perform in collaboration with one or more external users of the system (actors). Use case diagrams are used to gather the requirements of a system including internal and external influences. In each use, the system's actors should provide some observable and valuable results. A use case diagram does not show the detail of the use cases: it only summarizes some of the relationships between use cases, actors, and systems. Use cases deal only in the functional requirements for a system.

Use case diagrams depict:

- ✓ Use cases: A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse.
- ✓ Actors: An actor is a person, organization, or external system that plays a role in one or more interactions with your system. Actors are drawn as stick figures.
- ✓ Associations: Associations between actors and use cases are indicated in use case diagrams by solid lines. An association exists whenever an actor is involved with an interaction described by a use case.
- ✓ System boundary boxes (optional): You can draw a rectangle around the use cases, called the system boundary box, to indicate the scope of your system.

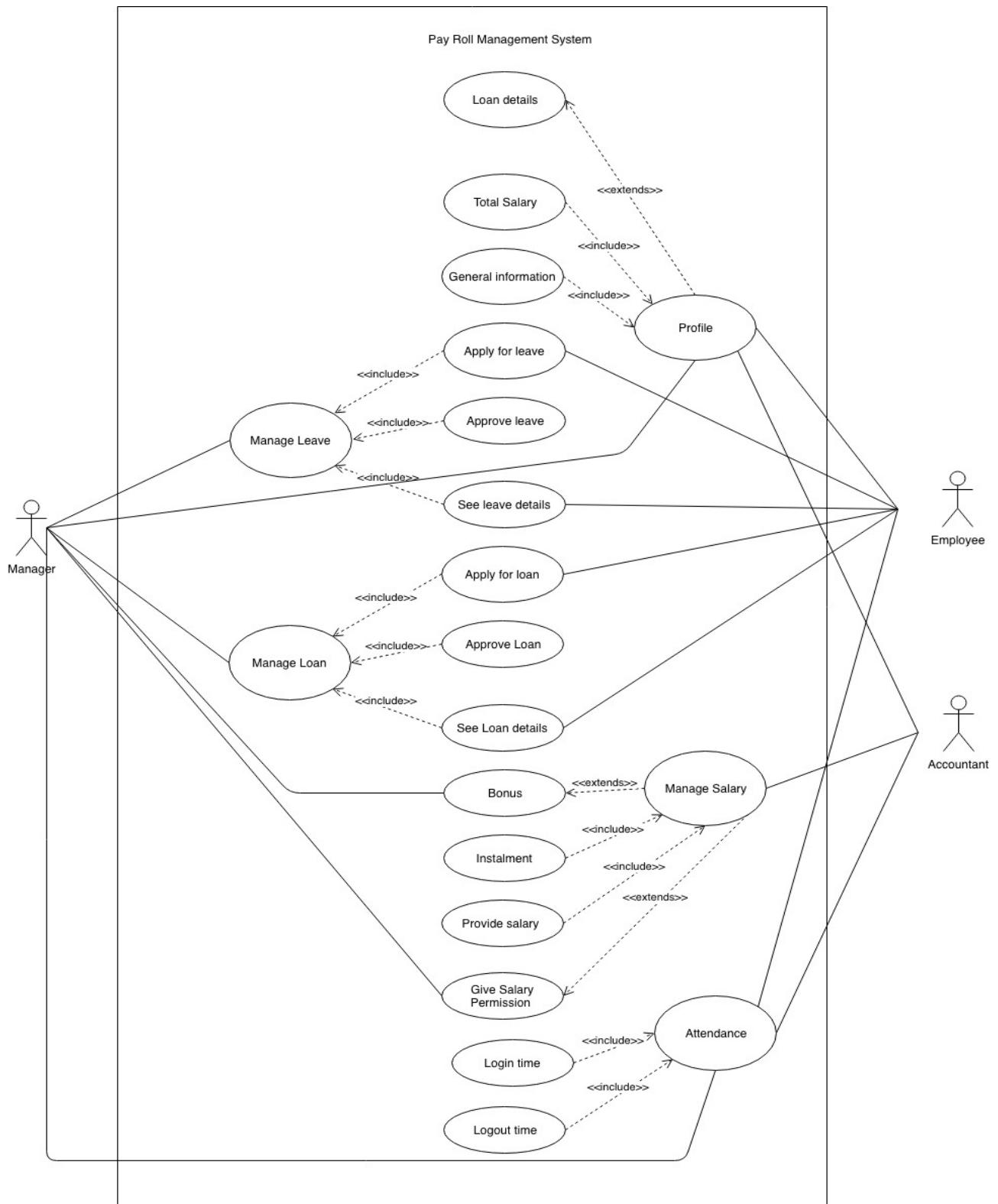


Figure 3.1: Use case diagram

Chapter 04

System Planning

4.1 Project Planning

Before starting any project, the completion of the work will be evaluated, the necessary resources, the time that is going to start from the beginning, and whether it is possible to analyze whether it is possible to analyze the project. Project planning involves breaking down the work into parts and assign these to project team members, anticipate problems that might arise and prepare tentative solutions to those problems. The project plan, which is created at the start of a project, is used to communicate how the work will be done to the project team and customers, and to help assess progress on the project.

The following activities of software project planning that have followed in this project are:

- ✓ Estimation of the software project.
- ✓ Task scheduling.
- ✓ Personal requirements.
- ✓ Resource requirements.
- ✓ Estimation of the software cost.
- ✓ Cost benefit analysis.

4.2 Project Estimation

Accurate estimations ultimately determine the overall success of a software project. Effective project planning and management are difficult to achieve without this. The accuracy of a software project estimate predicated based on a number of things. The accuracy of a software project estimate predicated based on a number of things:

- ✓ Properly estimated the size of the product to build.
- ✓ The ability to translate the size estimation into human effort, calendar time and money.
- ✓ The degree to which the project plan reflects the abilities of the software.
- ✓ The stability of the product requirements and the environment that supports the software engineering effort.

4.3 Functions of proposed system

- ✓ User can login. (F1)
- ✓ Manager can add Employee. (F2)
- ✓ Employee can edit Profile. (F3)
- ✓ Employee attendance entry. (F4)

- ✓ Employee can view attendance sheet. (F5)
- ✓ User can apply for leave. (F6)
- ✓ Manager can approve/decline leave. (F7)
- ✓ User can apply for loan. (F8)
- ✓ Manager can approve/decline loan. (F9)
- ✓ Manager can give permission for bonus. (F10)
- ✓ Applications generate Salary. (F11)
- ✓ Employee can get notification. (F12)

4.4 Function Description

User can login (F1)

Input: username and password.

Output: Login successful or login failed.

Use table of the database: employee.

Manager can add Employee. (F2)

Input: Name, department, designation, email id, password, user role, address, salary.

Output: Add successful.

Use table of the database: employee.

Employee can view Profile. (F3)

Input: password, address.

Output: Edit successfully.

Use table of the database: employee.

Employee attendance entry. (F4)

Input: Click start button .

Output: Successfully added.

Use table of the database: attendance.

Employee can view attendance sheet. (F5)

Input: Month, year.

Output: Attendance sheet.

Use table of the database: attendance.

User can apply for leave. (F6)

Input: leave type, from date, to date, reason.

Output: Save successfully or show error message.

Use table of the database: leave.

Manager can approve/decline leave. (F7)

Input: Accept/Deny.

Output: Save leave data successfully.

Use table of the database: leave.

User can apply for loan. (F8)

Input: Write reason.

Output: Save successfully or show error message.

Use table of the database: loan.

Manager can approve/decline loan. (F9)

Input: Accept/Deny.

Output: Save loan data successfully.

Use table of the database: loan.

Manager can give permission for bonus. (F10)

Input: Enable bonus button.

Output: Save successfully or show error message.

Use table of the database: salary.

Applications generate Salary. (F11)

Input: Basic salary, salary allowance, bonus.

Output: Total Salary.

Use table of the database: salary, Loan and leave.

Employee can get notification. (F12)

Input: username, password.

Output: total salary details in a notification.

Use table of the database: salary, grade, leaveusers, loanpaiddetails, loan, attendence, userinfo.

4.5 Function Point Estimation

Function point based estimation focuses on information domain values rather than software values. Function points are computed by comparing five information domain characteristics. The Five Components of Function Points-[5]

Data Functions

- ✓ Internal Logical Files
- ✓ External Interface Files

Transactional Functions

- ✓ External Inputs
- ✓ External Outputs
- ✓ External Inquiries

Number of external inputs – Each user input that provides distinct application-oriented data to the software is counted inputs should be distinguished from inquiries.

Number of external outputs – Each user output that provides application-oriented information to the user is counted.

Number of external inquiries – An inquiry defined as an on-line input those results in the generation of some immediate software response in the form of an on-line output. Each distinct inquiry counted.

Number of Internal Logical files – Each logical internal file is a logical grouping of data that resides within the application's boundary and is maintained via external inputs.

Numbers of external interfaces – All machine-readable interfaces that used to transmit information to another system counted.

The weights of the domains are fixed, which are provided in appropriate table location. Weights can be divided into three categories according to the functionality of the system. They are simple, average and complex. The total system is a complex system but the part of the total system. Once these data has collected, a complexity value is associated with each count. To find out the FP count the following formula is used,

Value Adjustment Factor (VAF) = $(0.65 + (.01 \times TDI))$
Adjusted Function Point Count (AFP) = UFP X VAF
Effort for PHP = AFP x Productivity

Unadjusted Function Point Contribution for Transaction Function

Here Table 4.1 shows the Unadjusted Function Point Contribution for Transaction Function for Payroll Management system.

Table 4.1 : Unadjusted Function Point Contribution for Transaction Function

Transaction Function	Fields/File Involvement	FTRs	DETs
User can login. (F1)(EQ)	Fields: username, password, authority File Name: userinfos	1	3
Manager can add Employee. (F2)(EI)	Fields: username, password, authority, fullname, address, email, phone, enabled, joiningDate, grade_id File Name: userinfos, grade	2	10
Employee can edit Profile. (F3)(EI)	Fields: password File Name: userinfos	1	1
Employee attendance entry. (F4)(EI)	Fields: logintime, ipaddress, userinfo_id File Name: attendance, userinfo	2	3
Employee can view attendance sheet. (F5) (EI)	Fields: workinghours File Name: attendance, userinfo	2	1
User can apply for leave. (F6)(EI)	Fields: leavetype, amount, status, userinfo_id, reason, entryfrom, entryto, total_leave_days. File Name: leaveusers, userinfo	2	8
Manager can approve/ decline leave. (F7)(EI)	Fields: leavetype, amount, status, userinfo_id, reason, entryfrom, entryto, total_leave_days. File Name: leaveusers, userinfo	2	8
User can apply for loan. (F8)(EI)	Fields: placedate, amount, status, userinfo_id, reason File Name: loan, userinfo	2	5
Manager can approve/ decline loan. (F9)(EI)	Fields: placedate, amount, status, userinfo_id, reason, approvedate File Name: loan, userinfo	2	6
Manager can give permission for bonus. (F10)(EI)	Fields: pendingbonus, userssalary File Name: isApprovedByAdmin	1	2

Applications generate Salary. (F11)(EO)	Fields: username, authority, workinghours, joiningDate, ipaddress, status, loan_id, userinfo_id, total_leave_days, status, loan_id, basic, medicalallowence, houserent, transport, lunch, study, leavetype, File Name: salary, greade, leaveusers, loanpaiddetails, loan, attendence, userinfo.	7	18
Employee can get notification. (F12)(EO)	Fields: username, authority, workinghours, status, loan_id, userinfo_id, total_leave_days, status, loan_id, study, leavetype, File Name: salary, greade, leaveusers, loanpaiddetails, loan, attendence, userinfo.	7	11
Manager can send mail. (F13)(EI)	Fields: placedate, amount, status, userinfo_id, reason, approvedate File Name: loan, userinfo	2	6

Table 4.2: Identifying Complexity Data function

Transaction Function	Fields/File Involvement	FTRs	DETs
Registration (ILF)	Fields: username, password, authority, fullname, email, phone, password, enabled, joiningDate, grade_id. File Name: userinfos, greade.	2	10
Attendance (ILF)	Fields: logintime, logouttime, userinfo_id, ipaddress, workinghours. File Name: userinfos, logintime.	2	5
Accept / Deny Leave (ILF)	Fields: status. File Name: leaveusers, userinfos.	2	1
Accept / Deny Loan (ILF)	Fields: status. File Name: loan, userinfo_id.	2	1
Leave (ILF)	Fields: reasone, status, leavetype, userinfo_id, entryfrom, entryto, total_leave_days File Name: leaveusers, userinfos.	2	7
Loan (ILF)	Fields: placedate, approvedate, amount, status, userinfo_id, reason File Name: loan, userinfo	2	5

Salary (ILF)	Fields: id, datemonthyear, Loan_id, grade_id, userinfo_id, leaveusers_id, totalsalary. File Name: salary, greade, leaveusers, loanpaiddetails, loan, attendence, userinfo.	7	7
loan paid details (ILF)	Fields: id, datetime, paidamount, loan_id. File Name: loanpaiddetails, loan	2	3

Table 4.3 : Unadjusted Function Point Estimation (Transaction Function)

Transaction Function	Fields/File Involvement	FTRs	D E T S	Complexity	UFP
User can login. (F1)(EQ)	Fields: username, password, authority File Name: userinfos	1	3	Low	3
Manager can add Employee. (F2)(EI)	Fields: username, password, authority, fullname, address, email, phone, enabled, joiningDate, grade_id File Name: userinfos, grade	2	10	Average	4
Employee can edit Profile. (F3)(EI)	Fields: password File Name: userinfos	1	1	Low	3
Employee attendance entry. (F4)(EI)	Fields: logintime, ipaddress, userinfo_id File Name: attendence, userinfo	2	3	Low	3
Employee can view attendance sheet. (F5) (EO)	Fields: workinghours File Name: attendance, userinfo	2	1	Low	4
User can apply for leave. (F6) (EI)	Fields: leavetype, amount, status, userinfo_id, reason, entryfrom, entryto, total_leave_days. File Name: leaveusers, userinfo	2	8	Average	4

Manager can approve/ decline leave. (F7)(EI)	Fields: leavetype, amount, status, userinfo_id, reason, entryfrom, entryto, total_leave_days. File Name: leaveusers, userinfo	2	8	Average	4
User can apply for loan. (F8) (EI)	Fields: placedate, amount, status, userinfo_id, reason File Name: loan, userinfo	2	5	Low	3
Manager can approve/ decline loan. (F9)(EI)	Fields: placedate, amount, status, userinfo_id, reason, approvedate File Name: loan, userinfo	2	6	Average	4
Manager can give permission for bonus. (F10) (EI)	Fields: pendingbonus, userssalary File Name: isApprovedByAdmin	1	2	Low	3
Applications generate Salary. (F11) (EO)	Fields: username, authority, workinghours, joiningDate, ipaddress, status, loan_id, userinfo_id, total_leave_days, status, loan_id, basic, medicalallowence, houserent, transport, lunch, study, leavetype, File Name: salary, greade, leaveusers, loanpaiddetails, loan, attendence, userinfo.	7	18	High	7
Employee can get notification. (F12)(EO)	Fields: username, authority, workinghours, status, loan_id, userinfo_id, total_leave_days, status, loan_id, study, leavetype, File Name: salary, greade, leaveusers, loanpaiddetails, loan, attendence, userinfo.	7	11	High	7
Manager can send mail. (F13)(EI)	Fields: placedate, amount, status, userinfo_id, reason, approvedate File Name: loan, userinfo	2	6	High	6
Total					55

Table 4.4: Unadjusted Function Point Estimation (Data Function)

Transaction Function	FTRs	DETs	Complexity	UFP
Registration (ILF)	2	10	Low	7
Attendance (ILF)	2	5	Low	7
Accept / Deny Leave (ILF)	2	1	Low	7
Accept / Deny Loan (ILF)	2	1	Low	7
Leave (ILF)	2	7	Low	7
Loan (ILF)	2	5	Low	7
Salary (ILF)	7	7	Low	7
loan paid details (ILF)	2	3	Low	7
Total				56

4.5.1 Performance and Environmental Impact

Here Table 4.5. shows the Performance and Environmental Impact of Payroll management System.

Table 4.5: Performance and Environmental Impact

GSC	DI
Data Communications	3
Distributed Data Processing	0
Performance	2
Heavily Used Configuration	2
Transaction Rate	0
Online Data Entry	2
End-user Efficiency	2
Online Update	3
Complex Processing	3
Reusability	4
Installation Ease	4
Operational Ease	2
Multiple Site	3

Facilitate Change	4
Total Degree of Influence (TDI)	35

$$\text{Value Adjustment Factor (VAF)} = (0.65 + (0.1 \times \text{TDI})) = (0.65 + (0.1 \times 35)) = 1$$

$$\text{UFP} = \text{UFP (Data Function)} + \text{UFP (Transaction Function)} = 55 + 56 = 111$$

$$\text{Adjusted Function Point Count} = \text{UFP} \times \text{VAF} = 111 \times 1 = 111$$

$$\text{Effort's for JAVA} = \text{AFP} \times \text{Proximity}$$

$$= 111 \times 10.6 \text{ [Productivity of JAVA is 10.6]}$$

$$= 1176.6 \text{ person hours / 8 hours}$$

$$= 147 \text{ person days / 24 days}$$

$$= 6.125 \text{ persons month / 2 months}$$

$$= 3.0625 \text{ months for 2 persons}$$

4.6 Process Based Estimation

In process-based estimation, process is decomposed into a relatively small set of tasks and the effort required to accomplish each task is estimated. Process based estimation begins with a delineation of software functions obtained from the project scope. A series of software process activities must be performed for each function.[6]

Table 4.6 : Process based estimation table:

Activity	CC	Planning	Risk Analysis	Engineering		Construction		CE	Total
Function				Analysis	Design	Code	Test		
Manager can add new employee	0.13	0.33	0.96	1.37	0.24	3.33	0.13	n/a	6.49
User can login	0.27	1.56	0.57	1.27	0.33	3.66	0.27	n/a	7.93
Manager can give permission for	0.25	0.96	0.41	2.13	1.12	3.78	0.25	n/a	8.9
Employee can update Profile	0.33	1.12	0.37	2.27	1.42	3.96	0.33	n/a	9.86

Employee Apply for loan	0.66	2.96	0.47	1.25	1.29	2.57	0.66	n/a	9.86
Employee apply for leave	0.78	0.67	0.46	0.33	1.33	0.33	0.41	n/a	4.31
Manager can accept leave	0.96	0.24	0.27	1.66	1.56	2.56	0.37	n/a	7.62
Manager Can deny leave	0.57	0.33	0.39	0.78	1.96	0.96	0.27	n/a	5.26
Manager can see total salary	0.41	0.12	0.66	0.96	5.12	2.12	0.78	n/a	10.11
Manager can monitor system	0.37	0.42	0.66	1.57	4.96	4.96	0.96	n/a	13.9
Accountant pay salary	0.27	0.29	0.78	1.41	3.67	8.77	0.57	n/a	15.76
Total	5%	0%	6%	15%	23%	37%	5%	n/a	100

4.7 Process based estimation pie chart:

The Use Case Points (UCP) has been a method that often used as a reference to calculate effort estimation, the amount of worker and time required in software development project. In Reality, software project planning has not been only effort estimation, but also cost estimation to estimate how much cost of resource allocation spent in software development. Result of the effort estimation using UCP is the number of effort required to develop the software as a whole. In fact, the determination of the cost of software development is based on the cost per activity of pay rate different in each phase of software development. Therefore, it requires the distribution of effort per activity to get the cost of software development.

Effort distribution particularly

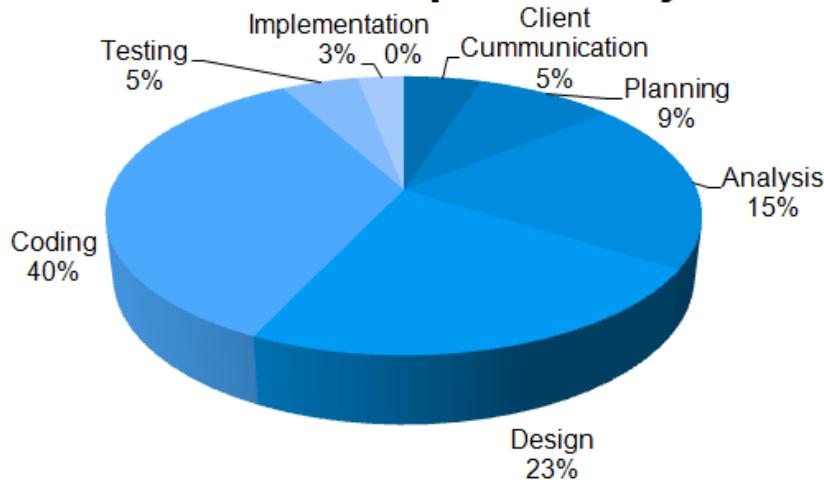


Figure 4.1 Effort Distribution

4.8 Task Scheduling

Project scheduling is an activity of distributing the estimated efforts within the planned project duration. There are some basic rules for project scheduling. They are as follows
Compartmentalization – The project must compartmentalize into a number of manageable activities and tasks.

Interdependency: The interdependency of each compartmentalized activity or task must be determined. Some tasks must occur in sequence while others can occur in parallel.

Time allocation: Each task to be scheduled must allocated some number of work units.

Effort validation: Every project has a defined number of staff members. It should ensure that no more than the allocated number of people has scheduled at any given time.

Defined responsibilities: Every task that is scheduled should assign to a specific team member.

Defined outcomes: Every task that is scheduled should have a defined outcome. The outcome is normally a work product or a part of a work product.

Table 4.7: Project Time Scheduling chart

Detail Activity	First Month				Second Month				Third Month				Fourth Month			
	1 st	2 ⁿ	3 ^r	4 ^t	1 ^s	2 ⁿ	3 ^r	4 ^t	1 ^s	2 ⁿ	3 ^r	4 ^t	1 ^s	2 ⁿ	3 ^r	4 ^t
<i>Week</i>	d	d	h	t	d	d	h	t	d	d	h	t	d	d	h	
Project Selection																
Proposal Submission to instructor																
Analysis the institution activity																
Introduce Prototype to client																
Collect client requirement																
Requirement understanding & analysis																
Making conceptual design & dataflow																
Divide the requirement into activity																
Define dependencies																
Select the types of resources																
Design the database																
Design the ER diagram																
Select the tools and software																
Design the interface																
Meet with Clients & show the interface																
Write Complete code																
Testing the input and testing the output																
testing the syntax error																
Install the system to clients hardware																

4.9 Cost Estimation

Personnel cost:

Number of days in a year = 365

Number of government holidays in a year =24

Number of weekly holidays in a year =52

Total number of working days to develop the project = $365-(52+24)$ =289 days

Total number of working days per months to develop the project = $289/12$ =24.083 days

Organization working hours per day = 8 hours

Organization working hours per month= $24.083*8= 192.66$ hours

Employee Cost:

Here Table 4.8 shows the Employee Cost of Payroll management System.

Table 4.8: Employee Cost

S a l a r y Management	O f	Working week	Person	Money
System Analyst	3		1	30000/=
System Designer	2.5		2	40000/=
Coder	4		4	28000/=
Schedule Monitor	5.5		2	20000/=
Budget controller	4.9		2	20000/=
System Maintenance	3.4		1	10000/=
Total				148000/=

Hardware and Software Cost:

Here Table 4.9 shows the hardware and software of Payroll management System.

Table 4.9: Hardware and Software Cost

Hardware/Software Name	Pieces	Price
Desktop	3	70000/=
Laptop	2	40000/=
Printer	3	20000/=
Windows	1	8000/=
Total		138000/=

Other Cost:

Here Table 4.10 shows the hardware and software cost of Payroll management System.

Table 4.10: Others Cost

Name	Price
Transport	1200/=
House rent	2000/=
Electric bill	1500/=

Extra	500/=
Total	5200/=

Total Cost:

Hear Table 4.11 shows the total cost of Payroll management System.

Table 4.11: Total cost

Cost Name	Price
Employer	148000/=
Hardware And Software	138000/=
Other	5200/=
Total	291200/=

Chapter 05

Analysis

5.1 Activity Diagram:

Activity diagrams describe the workflow behavior of a system. Activity diagrams are similar to state diagrams because activities are the state of doing something. The diagrams describe the state of activities by showing the sequence of activities performed. Activity diagrams can show activities that are conditional or parallel.

The basic purpose of activity diagrams is similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

Activity diagram is suitable for modeling the activity flow of the system. An application can have multiple systems. Activity diagram also captures these systems and describes the flow from one system to another. This specific usage is not available in other diagrams. These systems can be database, external queues, or any other system.

It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

The purpose of an activity diagram can be described as –

- ✓ Draw the activity flow of a system.
- ✓ Describe the swim len from one activity to another.
- ✓ Describe the parallel, branched and concurrent flow of the system.

5.1.1 Activity Diagram for Manager:

Figure 5.1.1 is the activity diagram for manager. Here we can see activity of admin from the beginning process like login. Manager can view pending leave and loan request and also can approve or decline it. Beside that he can send notification each and every employee and manage employee like add and edit employee.

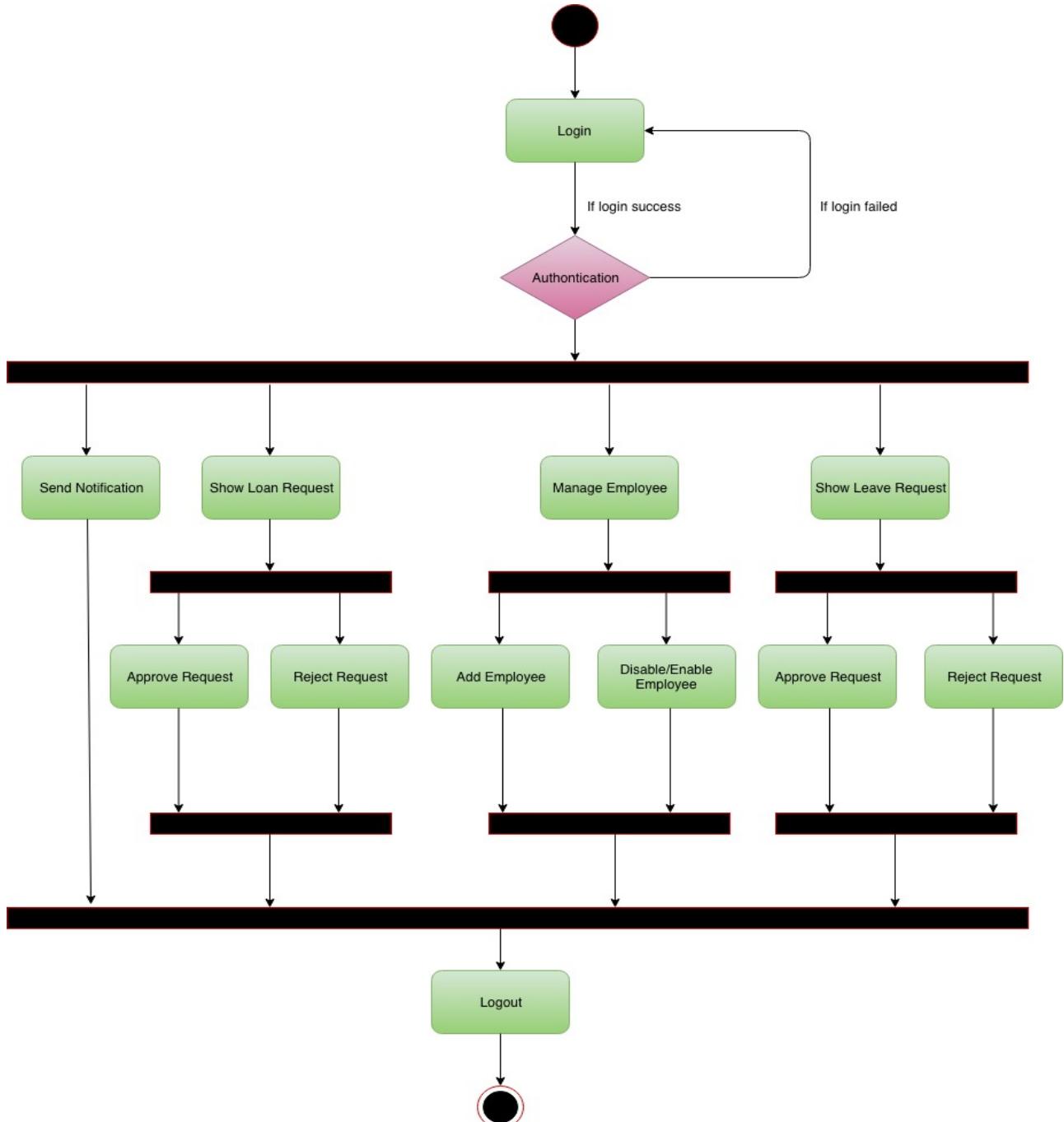


Figure 5.1.1 Activity Diagram for Manager

5.1.2 Activity Diagram for Accountant:

Figure 5.1.2 is the activity diagram Accountant. Here we can see the details accountant can do. He can login in the system and can view the salary status also can provide salary.

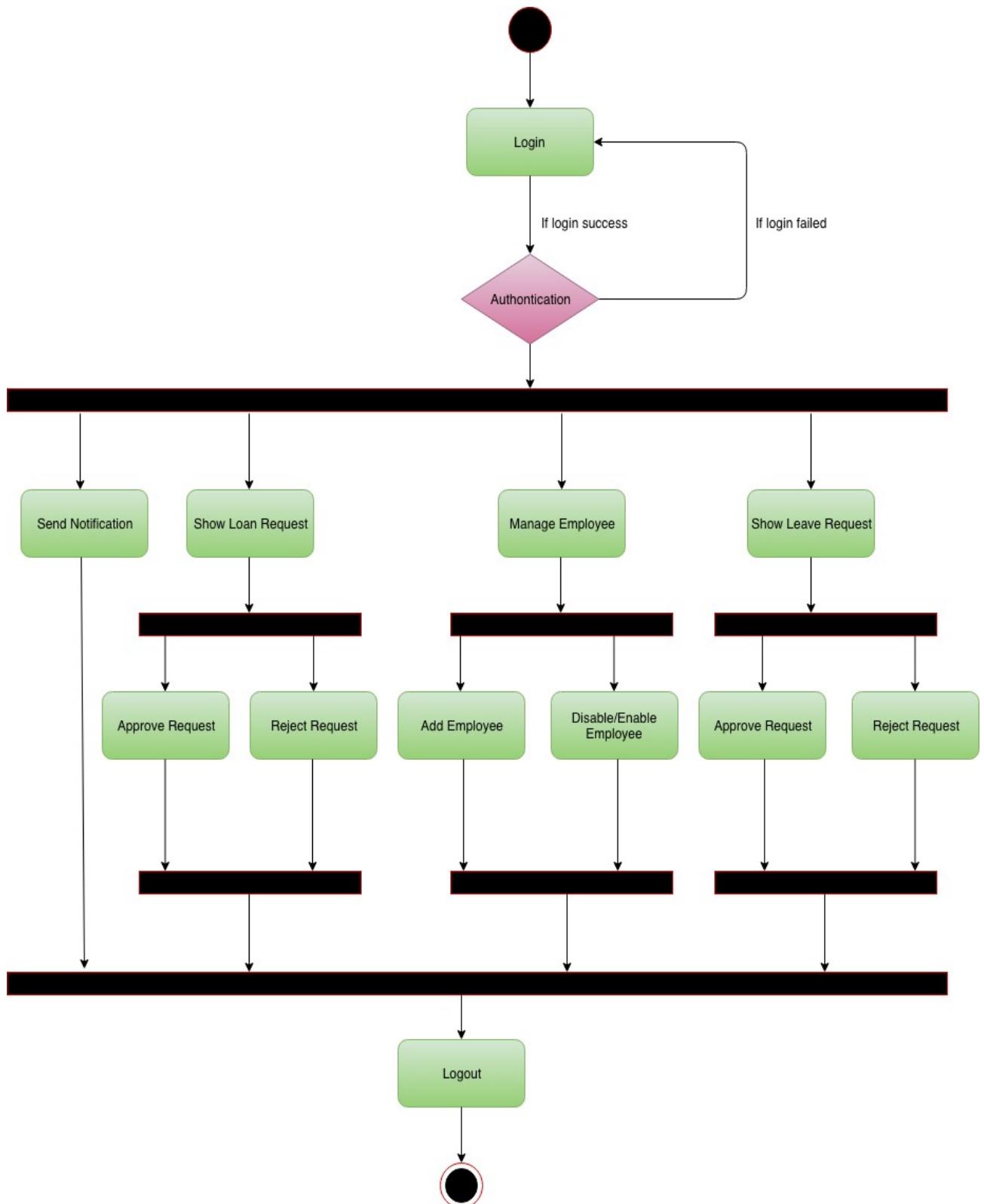


Figure 5.1.2 Activity Diagram for Accountant

5.1.3 Activity Diagram for Employee:

Figure 5.1.3 is the activity diagram for employee. Here we can see to access the software a employee have to login in to the system then he can view his profile and can print his pay slip Also can see the attendance sheet of all employee and enter attendance. He also can apply for leave and loan that manager will approve later.

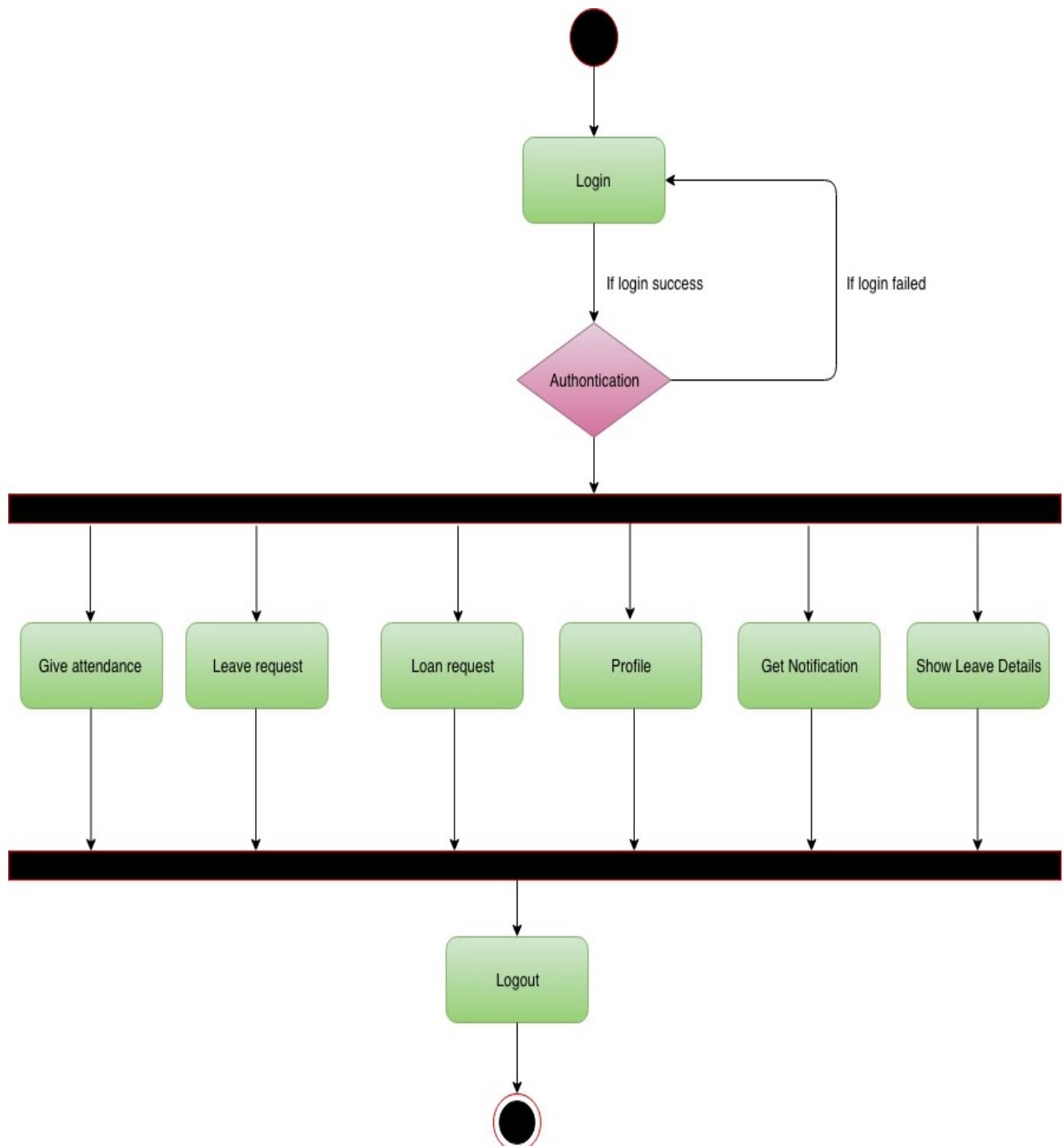


Figure 5.1.3 Activity Diagram for Employee

5.2 Swim lane Diagram

5.2.1 Manager can enroll new roll user

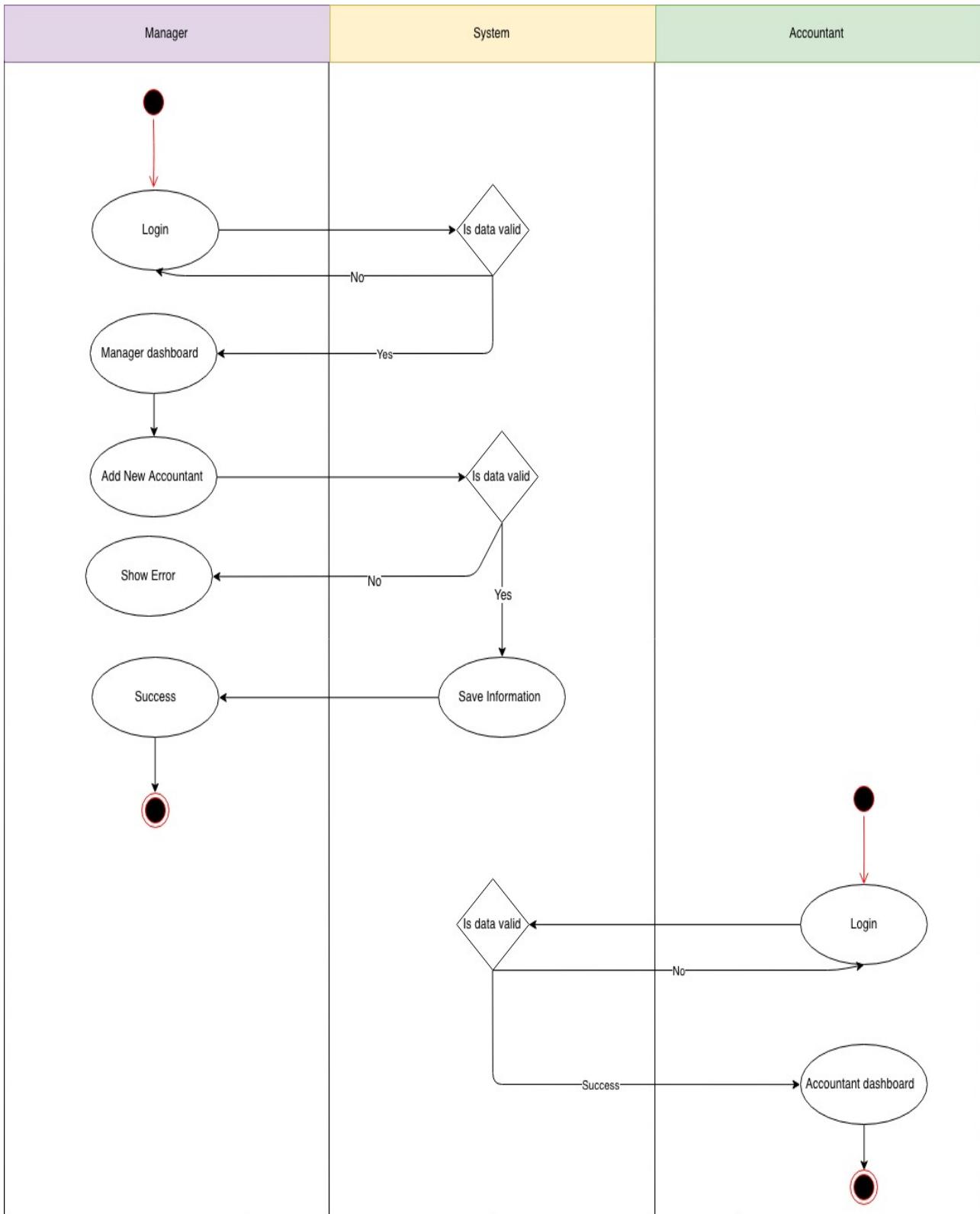


Figure 5.2.1 Manager can enroll new roll user

5.2.2 Admin can approve or deny loan request

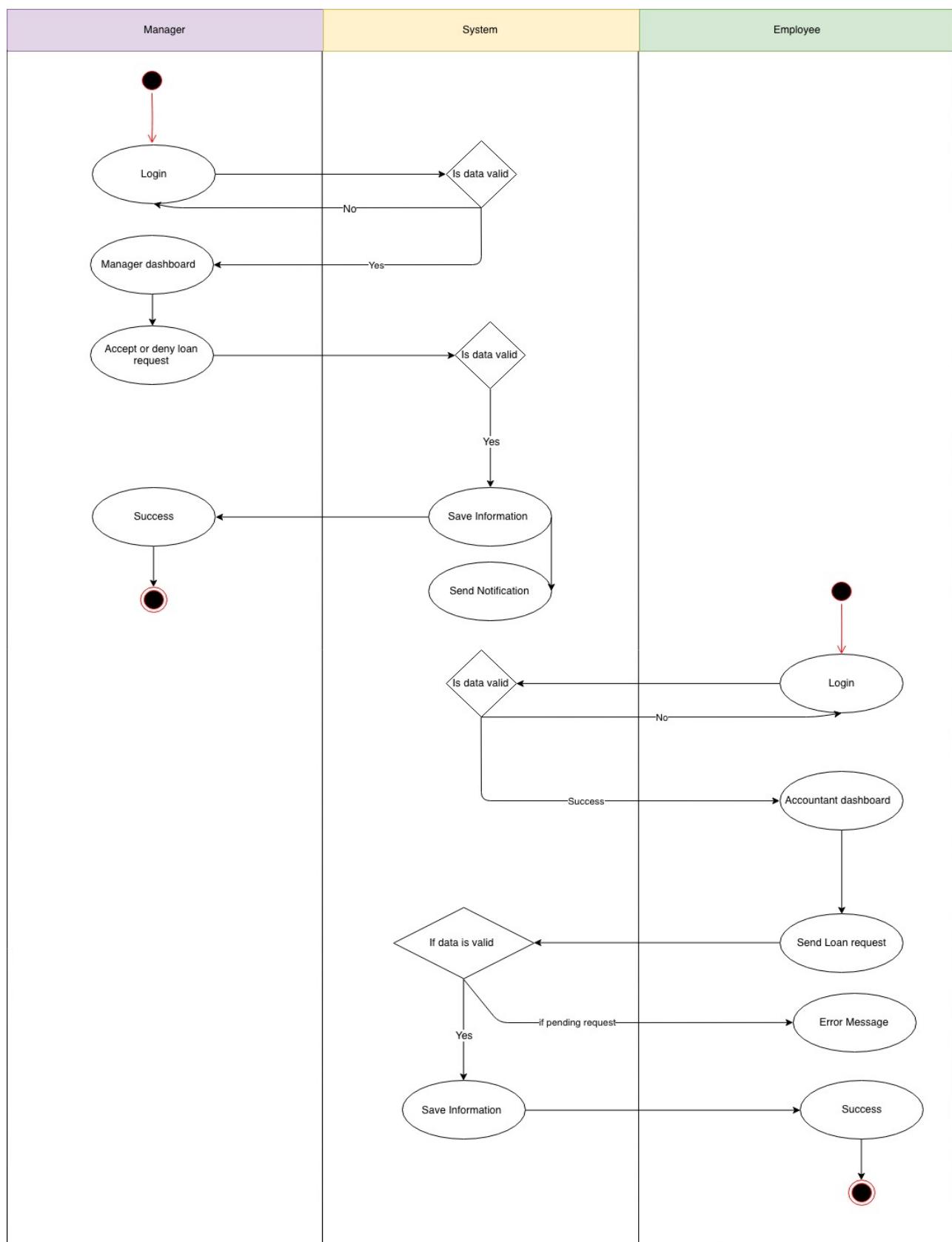


Figure 5.2.2 Admin can approve or deny loan request

5.2.3 Admin can approve or deny leave request

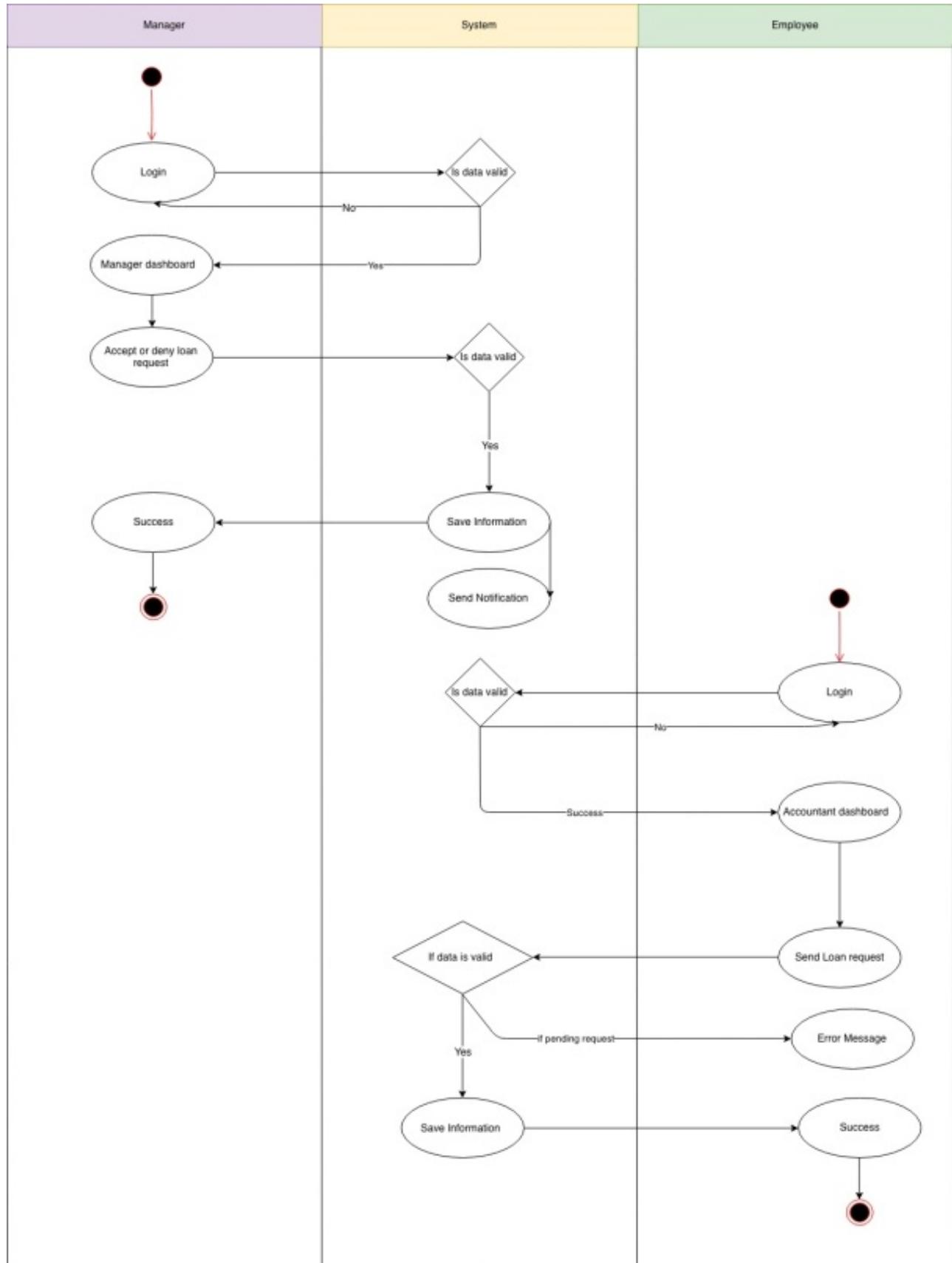


Figure 5.2.3 Admin can approve or deny leave request

5.2.4 Admin can give permission to pay salary

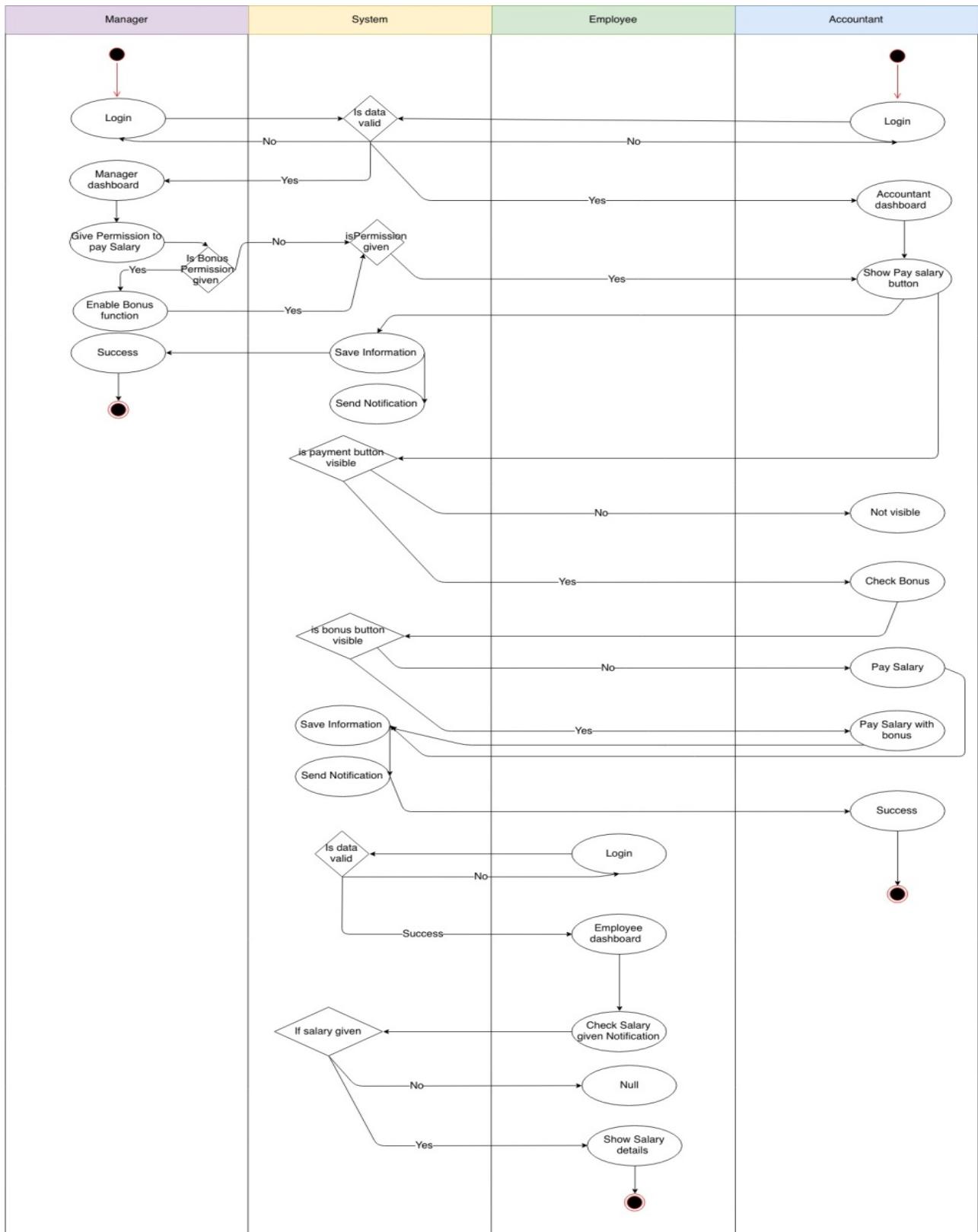


Figure 5.2.4 Admin can give permission to pay salary

Chapter 06

Designing

6.1 Entity Relationship Diagram

The Entity Relationship Diagram (ERD) enables a software engineer to specify the data objects that are input and output from a system, the attributes that define the properties of these objects and their relationship. It provides an excellent graphical representation of the data structures and relationship. They provide a clear view of the logical structure of data within the boundary of interest and allow the engineer to model the data without considering the physical form. Some of the basic terms used

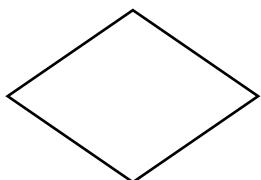


in ERD described below:

Entity: An entity is an object with physical existence or may be an object with conceptual existence. For example, a car, a student, an employee, an applicant.

An entity represented by a rectangle.

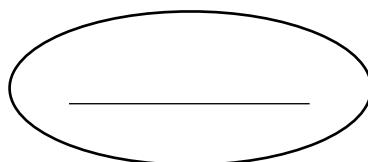
Relationship: A relationship is a logical linkage between two or more entities which describes how



the entities are associated with each other. A relationship described by a diamond.

Attribute: Attribute is a piece of information that describes a particular entity.

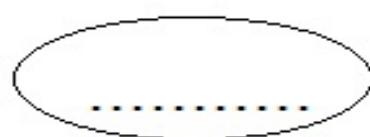
Primary Key: A primary key is an attribute or collection of attributes that allow us to identify an entity uniquely.



Foreign key: A foreign key is an attribute of a relation, which refers to an existing attribute of another relationship.

Relationship Cardinality

Relationship cardinality refers to the number of entity instances involved in the relationship.



The cardinality ratios are:

- ✓ 1:1 (One to One)
- ✓ 1:M (One to Many)
- ✓ M:M (Many to Many)

Here in Figure 6.1 we can see the ER Diagram of Payroll Management System. This diagram shows all the entities. Label with number shows the cardinality ratio and double line shows the total participation.[7] entity and attribute as well as participation and cardinality ration. Here employee ,manager, accountant ,loan, leave ,salary ,attendance are entity and no one is weak entity.

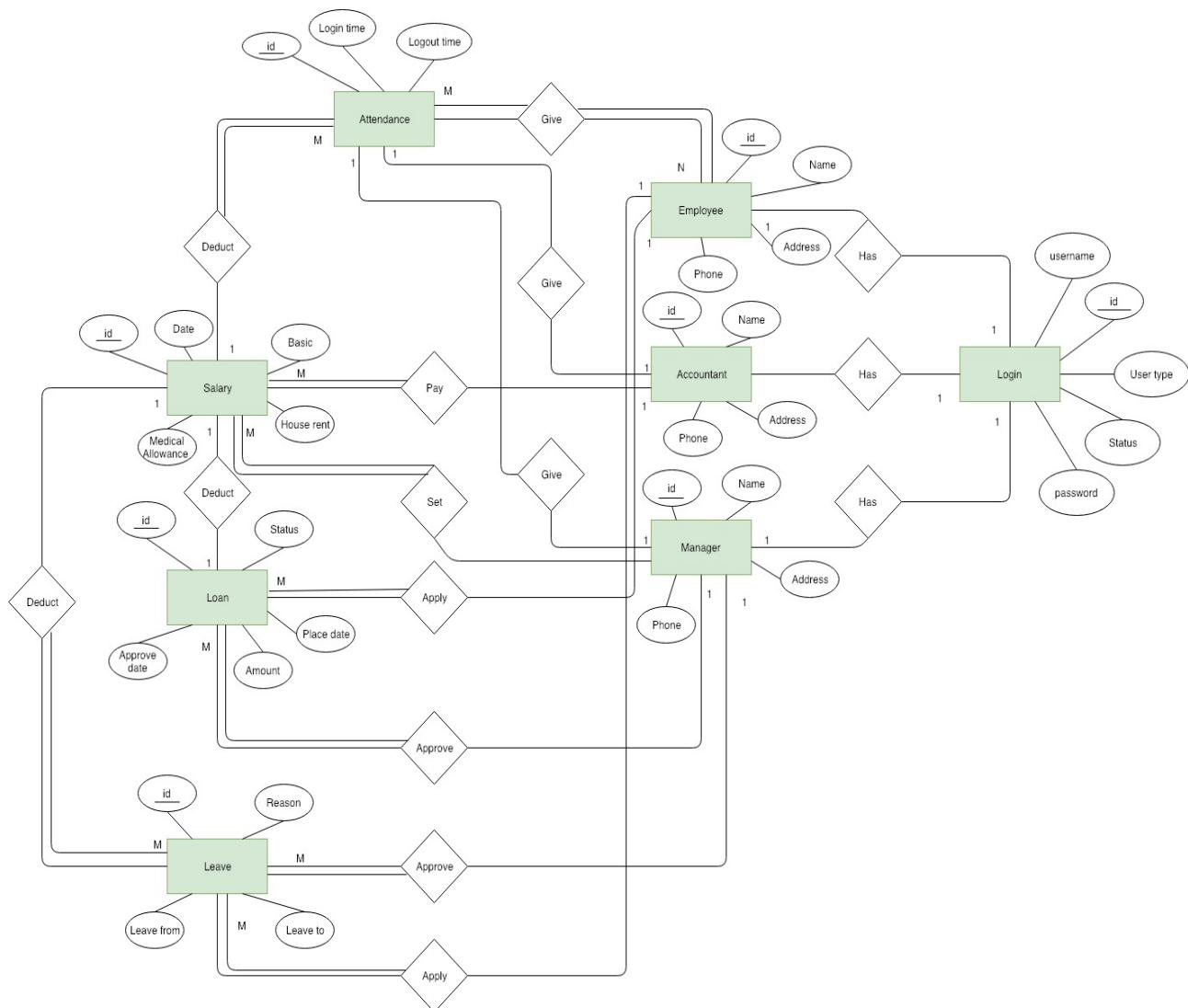


Figure 6.1 Entity Relationship Diagram

6.2 Data Flow Diagram

6.2.1 Level 0 DFD

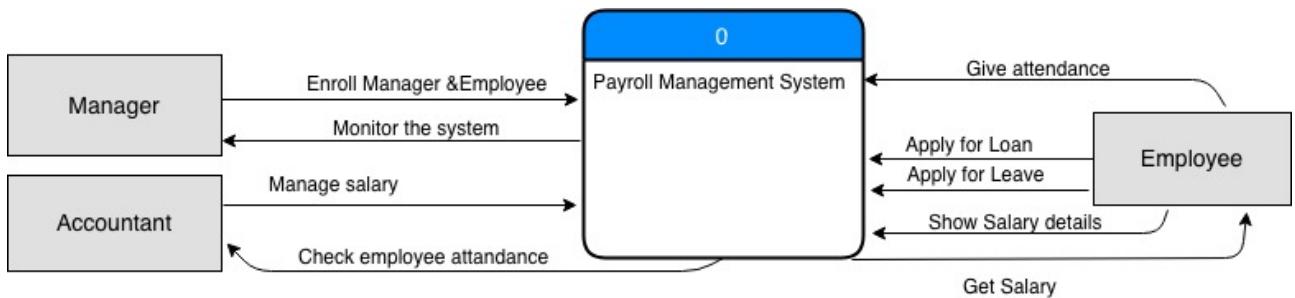


Figure 6.2.1 Level 0 DFD

6.2.2 Level 1 DFD

Figure 6.2.2 is the level 1 dfd of payroll management system. Here we have describe the context level diagram more briefly. Here we have divided the full PMS in 6 different module .and the figure shows how data are passing among themselves form module to database and module to employee ,manager ,accountant.

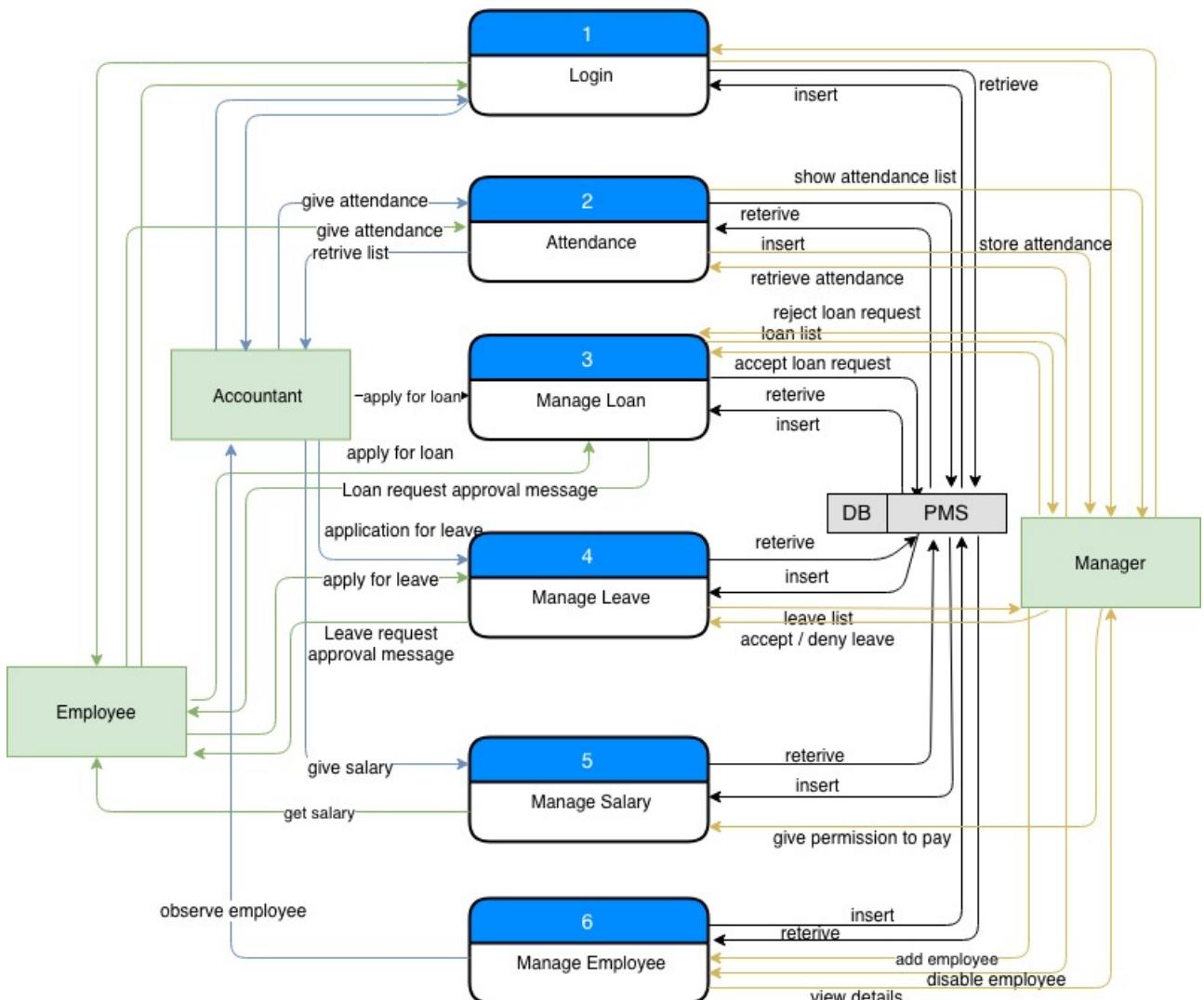


Figure 6.2.2 Level 1 DFD

6.2.3 Level 2 DFD of Process 1 (Login)

Figure 6.2.3 is the process 1 of level 2dfd of Payroll Management System. In this figure we can see Employee, Manager and Account giving input their login information and database is checking whether the information is valid or not if not show error message .

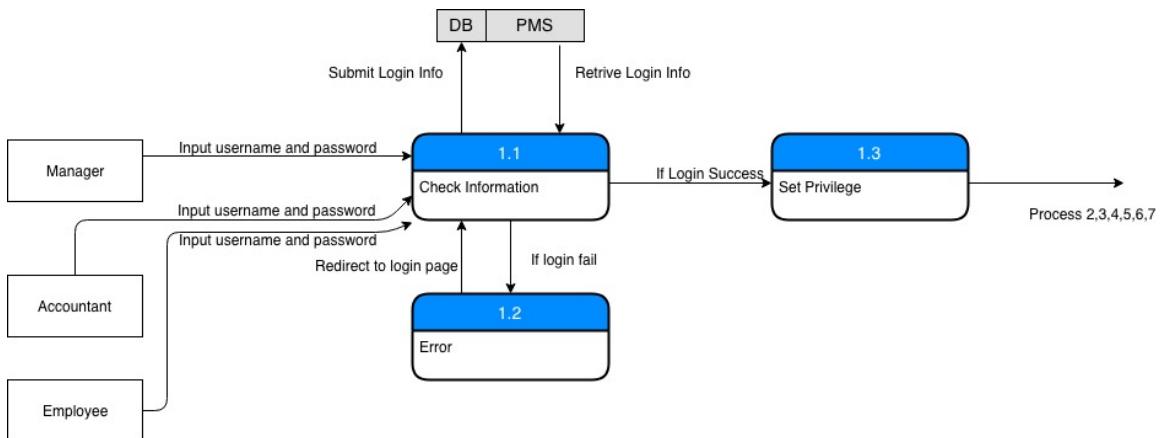


Figure 6.2.3 Level 2 of process 1 (login)

6.2.4 Level 2 DFD of Process 2 (Manage Employee)

Figure 6.2.4 shows level 2 Dfd of process 2 (Manage Employee).Here We can see process 2 is divided into four sub section add employee , employee details and disable employee. Here manager can add employee and disable employee , employee and accountant can edit there details but not all information ,some specific information. Database save all the input and edit information and retrieve where needed.

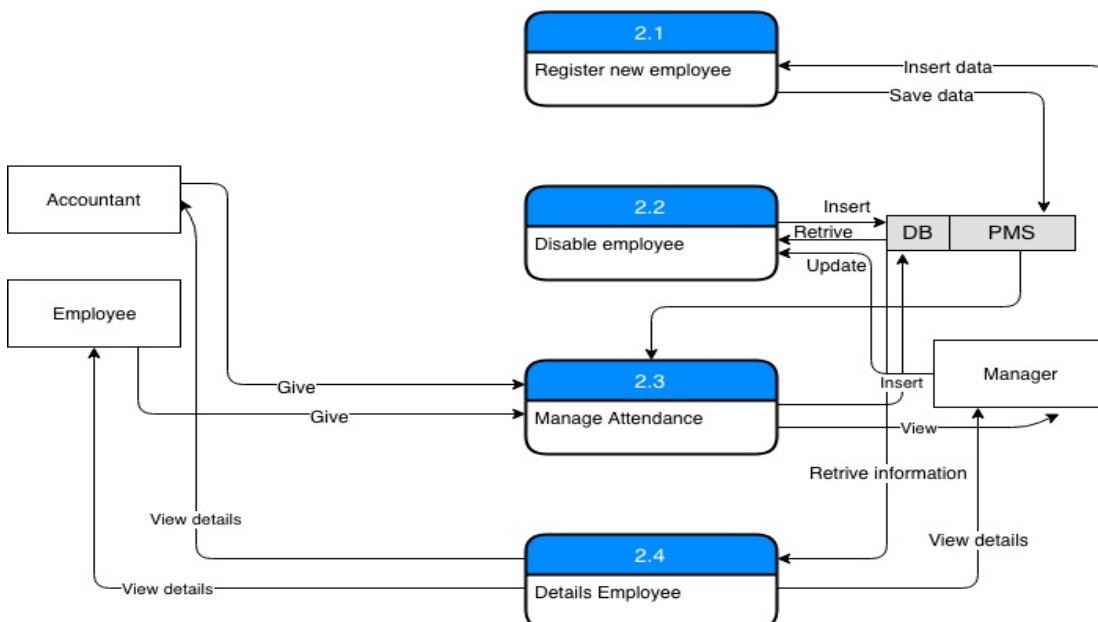


Figure 6.2.4 Level 2 of process 2 (Manage Employee)

6.2.5 Level 2 DFD of Process 3 (Manage Loan)

Figure 6.2.5 shows the level 2 dfd of process 3 (Manage Loan). In this diagram we can see the data flow of loan management .Loan management is divided into 3 subsection apply for loan , loan details ,approve / decline loan. Here employee, manager and account can also apply for loan .Manager can see pending loan list and approve or decline it. Database save all the information and retrieve where needed.

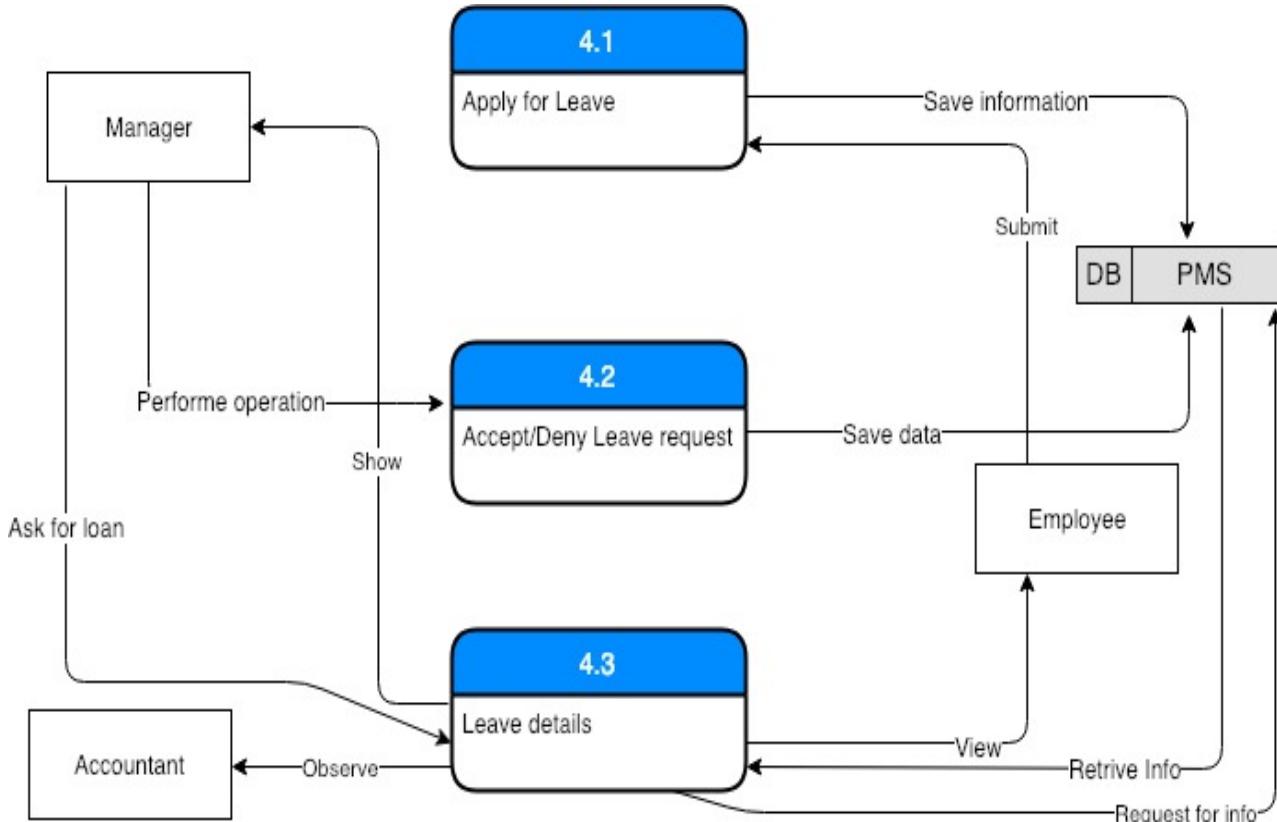


Figure 6.2.5 Level 2 DFD of Process 3 (Manage Loan)

6.2.6 Level 2 DFD of Process 4 (Manage Leave)

Figure 6.2.6 shows the level 2 dfd of process 3 (Manage Leave). Leave management is divided into four subsection apply for leave, leave details, approve / decline leave. Here employee, manager and account can also apply for leave .Manager can see pending leave list and approve or decline it. Database save all the information and retrieve where needed.

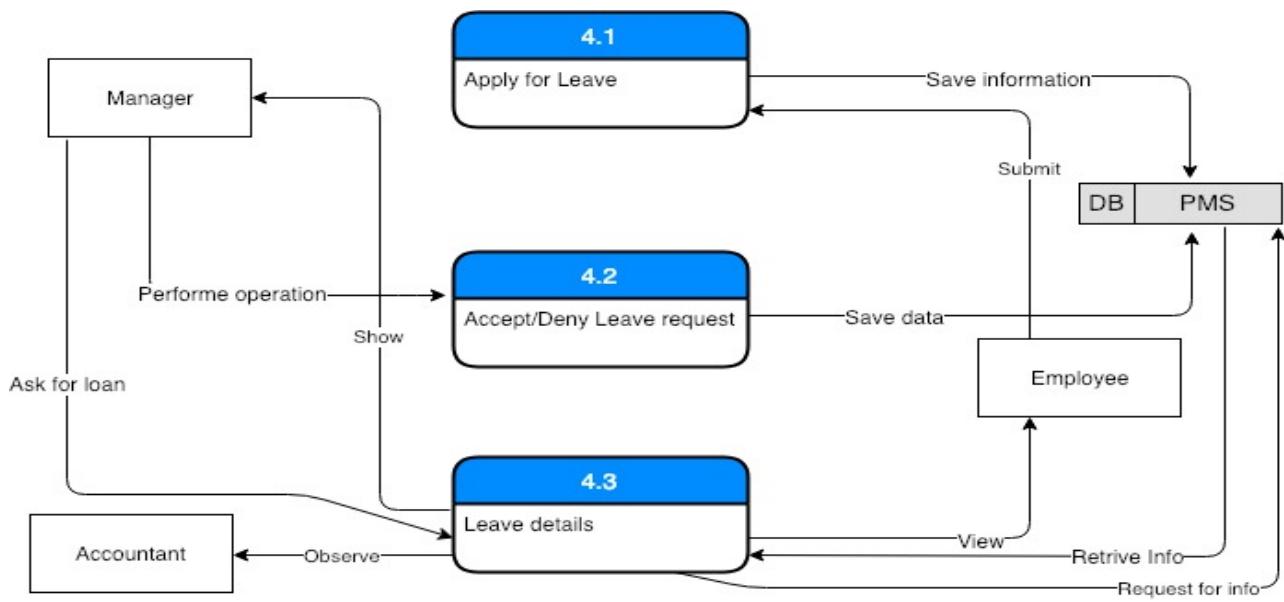


Figure 6.2.6 Level 2 DFD of Process 4 (Manage Leave)

6.2.7 Level 2 DFD of Process 5 (Manage Salary)

Figure 6.2.7 shows the level 2 dfd of process 5 (Manage salary) which is the most complex part of project. Here Accountant set the salary allowance and provident fund percentage .Accountant inserts the bonus and deduction. Algorithm inside the system process the total salary of a month.

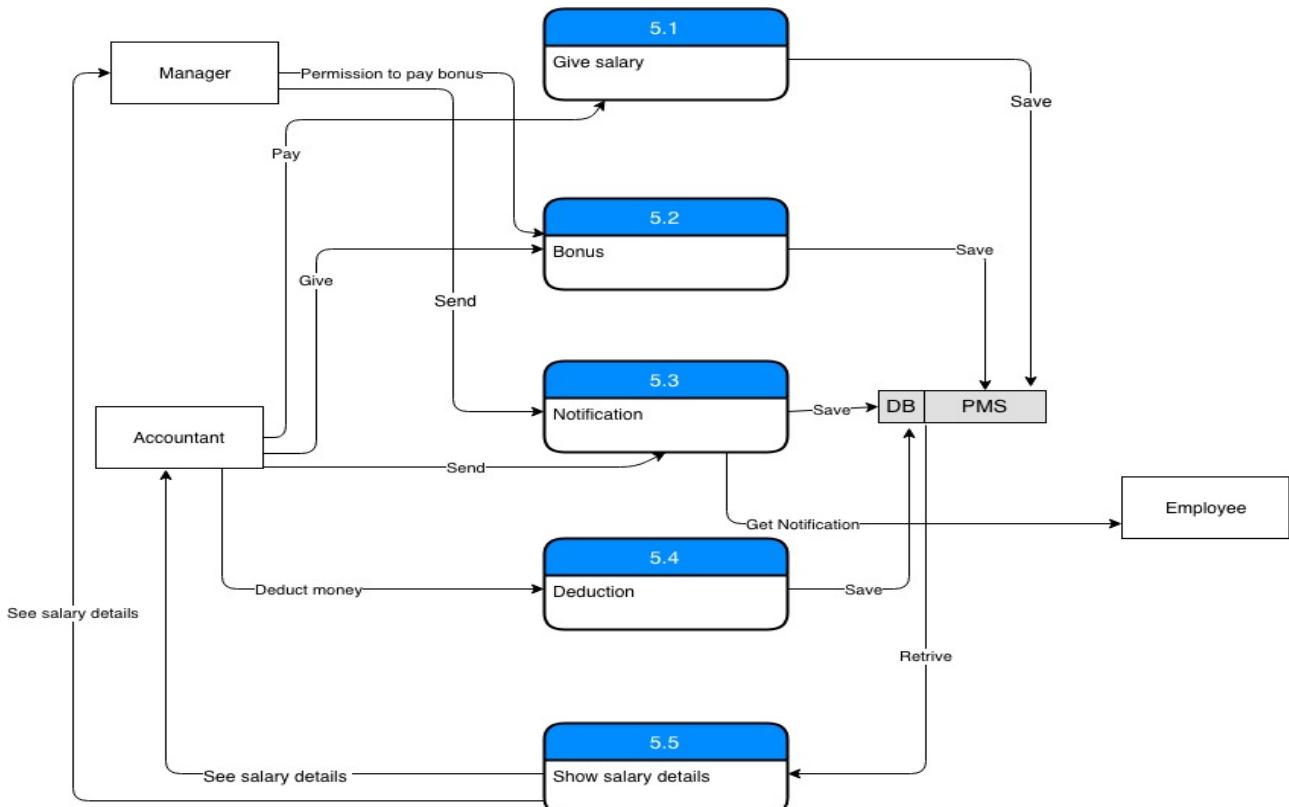


Figure 6.2.7 Level 2 DFD of Process 5 (Manage Salary)

6.2.8 Level 2 DFD of Process 6 (Attendance)

Figure 6.2.8 shows the level 2 dfd of process 6 (Attendance) .Here all employee can give attendance. Database save the attendance information and retrieve via an attendance sheet which is visible by all. Here Only manager can edit attendance information.

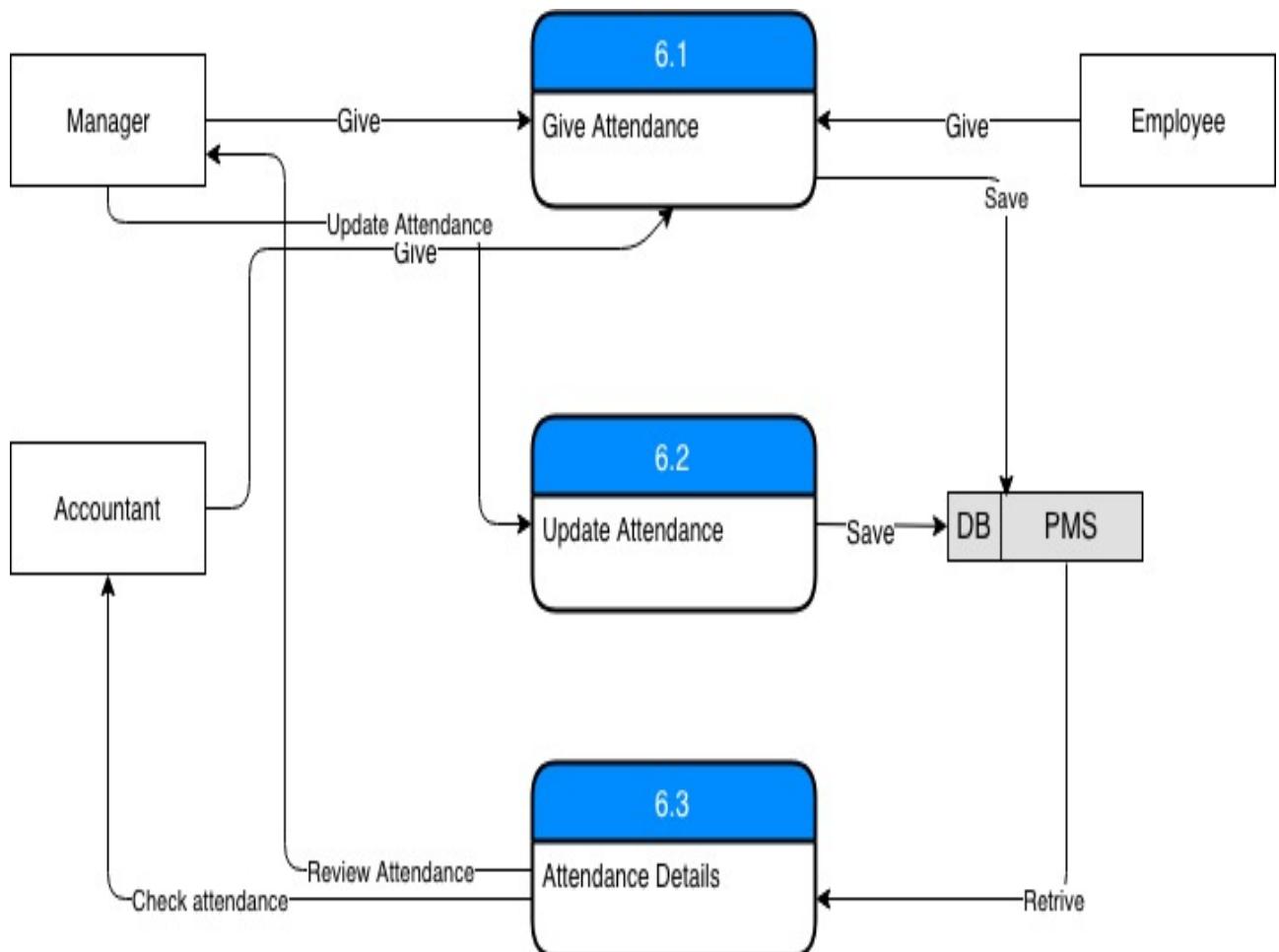


Figure 6.2.8 Level 2 DFD of Process 6 (Attendance)

Chapter 07

Risk analysis

7.1 Risk Engineering

A risk is a serious problem that might or might not happen. It is necessary to analyze the potential risks in a project. If the risks of a software project are not properly analyzed and estimated, many problems can plague the software project. Risk analysis and management are a series of steps that help a software team to understand and manage uncertainty. To establish a risk management model the following phases are followed:

Identification: Risk identification is the process of detecting potential risks or hazards through data collection. A range of data collection and manipulation tools and techniques exists. The team is using both automated and manual techniques to collect data and begin to characterize potential risks to Web resources. Web crawling is one effective way to collect information about the state of Web pages and sites.

Classification: Risk classification is the process of developing a structured model to categorize risk and fitting observable risk attributes and events into the model. The team combines quantitative and qualitative methods to characterize and classify the risks to Web pages, Web sites, and the hosting servers.

Assessment: Risk assessment is the process of defining relevant risk scenarios or sequences of events that could result in damage or loss and the probability of these events. Rosenthal describes the characteristics of a generic standard for risk assessment as "transparent, coherent, consistent, complete, comprehensive, impartial, uniform, balanced, defensible, sustainable, flexible, and accompanied by suitable and sufficient guidance."

Analysis: Risk analysis determines the potential impact of risk patterns or scenarios, the possible extent of loss, and the direct and indirect costs of recovery. This step identifies vulnerabilities, considers the willingness of the organization to accept risk given potential consequences, and develops mitigation responses.[8]

Implementation: Risk management implementation defines policies, procedures, and mechanism to manage and respond to identifiable risks. The implemented program should balance the value of assets and the direct and indirect costs of preventing or recovering from damage or loss.

To take comprehensive care of a web based system we must consider the following points:

1. Hardware and software environment including any upgrades to the operating system and Web server, the installation of security patches, the removal of insecure services, use of firewalls, etc.

2. Administrative procedures such as contracting with reputable service providers, renewing domain name registration, etc.
3. Network configuration and maintenance including load balancing, traffic management, and usage monitoring.
4. Backup and archiving policies and procedures including the choice of backup media, media replacement interval, number of backups made and storage location.

1. Project risks: These risks threaten the project plan. If these risks become real, it is likely that the project schedule will slip and that costs will increase. Project risks identify potential budgetary, schedule, personnel, resource, customer and requirement problems and their impact on the software project.

2. Technical risks: These risks threaten the quality and timeliness of the software to be produced. If a technical risk becomes a reality, implementation may become difficult or impossible. Technical risks identify potential design, implementation, interface, verification and maintenance problems. Moreover, specification ambiguity, technical uncertainty, technical obsolescence are also risk factors.

3. Business risks: These risks threaten the viability of the software to be built. The business risks can be market risks, building a system that no one really wants. Strategic risks, building a system that no longer fits into the overall business strategy for the company.

Assess probability and seriousness of each risk is:

Table 7.1: Technology Risk

Technology Risks	Probability	Effects
1.1) Unintentional defects in software may arise.	Low	Insignificant
1.2) The software we are using in our project may have any defect.	Low	Insignificant
1.3) Database may transaction many files at a time.	Low	Insignificant

Table 7.2: People Risk

People Risks	Probability	Effects
2.1) Insufficient number of staffs.	Low	Insignificant

2.2) Not all staffs are enough experienced.	Moderate	Tolerable
---	----------	-----------

Table 7.3: Organisational Risk

Organizational Risks	Probability	Effects
3.1) Not having sufficient funds may cause problem.	Low	Insignificant
3.2) Loss of service providing from organization.	Very Low	Insignificant

Table 7.4: Tools Risk

Tools Risks	Probability	Effects
4.1) Wrong tools may cause problems.	Low	Insignificant
4.2) Insufficient number of tools.	Low	Insignificant

Table 7.5: Requirement Risk

Requirement Risk	Probability	Effect
5.1) With time user requirements may change which may force to change the whole system.	Low	Insignificant
5.2) The process of managing requirement changes may increase system budget.	Moderate	Tolerable

Table 7.6: Estimation Risk

Estimation Risks	Probability	Effects
6.1) Our estimated time may not be correct.	Moderate	Tolerable
6.2) Estimated budget may increase.	Moderate	Tolerable
6.3) Estimated resources may not be enough for our project.	Low	Tolerable

7.2 The RMMM Plan

1. **Risk Mitigation:** Proactive planning for risk avoidance.
2. **Risk Monitoring:** Assessing whether predicted risks occur or not, ensuring preventive steps are being properly applied, collect information for future risk analysis, attempt to determine which risks caused which problem.
3. **Risk Management:** Actions to be taken in the event that mitigation steps have failed and the risk has become a live problem

Type of Impact:

1. Catastrophic,
2. Marginal,
3. Tolerable,
4. Critical.

Type of Probability:

- ✓ Very Low (<10%)
- ✓ Low (10%-25%)
- ✓ Moderate (25%-50%)
- ✓ High (50%-75%)
- ✓ Very high (75%-100%)

Project Risks: Threaten the project plan. In my system the bellow mentioned projects risks I Needed manage. Table VIII shows the risk of change Requirements.

Chapter 08

Interface Design

8.1 Interface for homepage

Url : <http://localhost:8081/PayrollManagementSystem/>

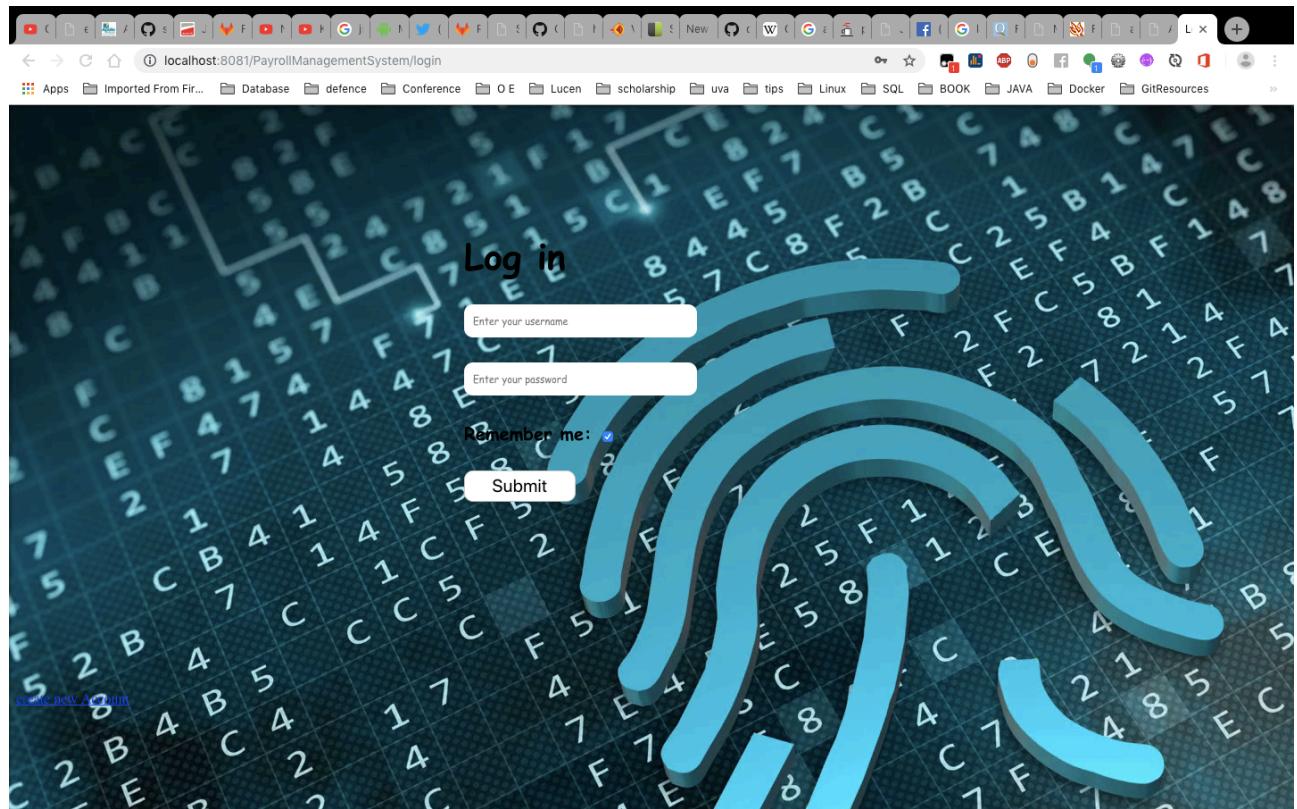


Figure 8.1.1 Login

Url : <http://localhost:8081/PayrollManagementSystem/home>

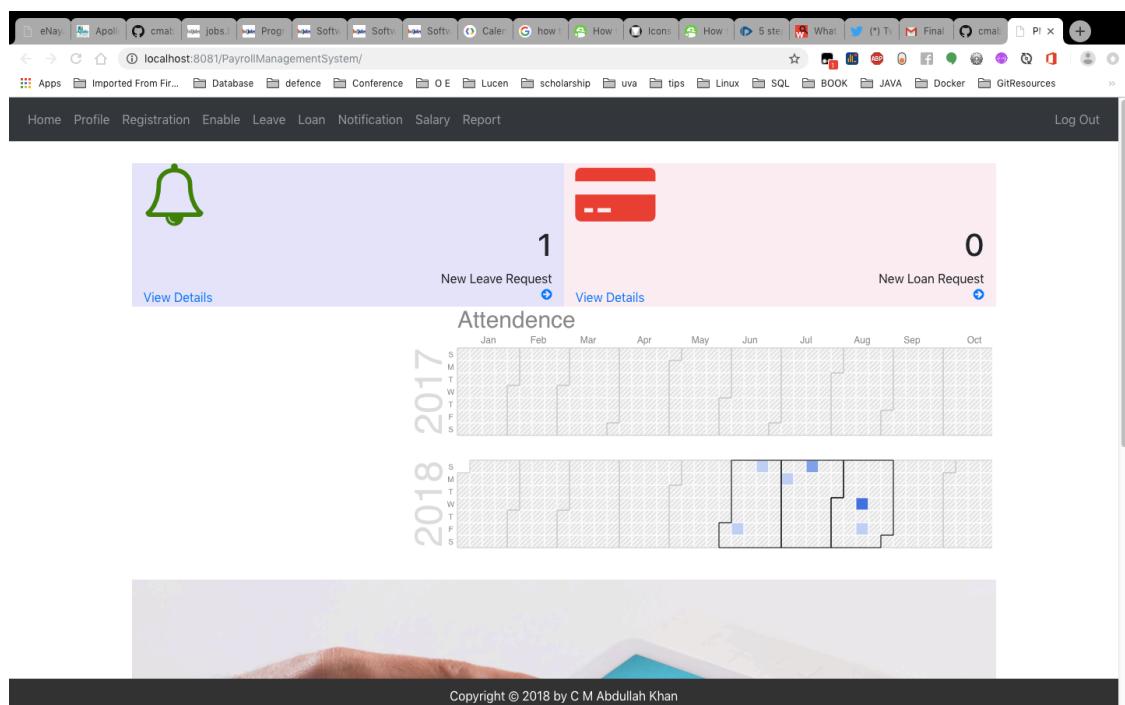


Figure 8.1.2 Home

Url : <http://localhost:8081/PayrollManagementSystem/registration>

The screenshot shows a web browser window with the URL <http://localhost:8081/PayrollManagementSystem/registration>. The page has a dark header bar with links for Home, Profile, Registration, Enable, Leave, Loan, Notification, Salary, Report, and Log Out. Below the header is a form with fields for Full Name, Address, Email, Phone, User Name, and Password. There is also a section for Authentication with a radio button for ADMIN. At the bottom of the page is a copyright notice: Copyright © 2018 by C M Abdullah Khan.

Full Name :
Enter your Full name

Address :
Enter your address

Email :
Enter your email

Phone :
0

User Name :
Enter your username

Password :
[Redacted]

Confirm Password :
[Redacted]

Authentication
 ADMIN

Copyright © 2018 by C M Abdullah Khan

Figure 8.1.3 Registration

Url : http://localhost:8081/PayrollManagementSystem/disable_enable_user

The screenshot shows a web browser window with the URL http://localhost:8081/PayrollManagementSystem/disable_enable_user. The page has a dark header bar with links for Home, Profile, Registration, Enable, Leave, Loan, Notification, Salary, Report, and Log Out. The main content area displays the message "You can enable or disable any user". It includes a text input field for "Enter your username", a radio button group for "Enable" (selected) and "Disable", and a green "Submit" button. At the bottom of the page is a copyright notice: Copyright © 2018 by C M Abdullah Khan.

You can enable or disable any user

Enter your username

Enable Disable

Submit

Copyright © 2018 by C M Abdullah Khan

Figure 8.1.4 Enable Disable employee

Url : http://localhost:8081/PayrollManagementSystem/ad_leave

The screenshot shows a web browser window with the URL localhost:8081/PayrollManagementSystem/ad_leave. The page displays a table of leave requests. The columns are: Fullname, Email, Reason, Leave Type, Entry From, Entry To, Total Leave Days, Ignore Leave, Delete, and Accept. One row is visible, showing C. M. Abdullah Khan with email a.kium.khan@gmail.com, reason Sick, leave type regular, entry from 2018-12-13T00:00, entry to 2018-12-14T00:00, total leave days 2, ignore leave 4, and buttons for Delete and Accept.

Copyright © 2018 by C M Abdullah Khan

Figure 8.1.5 Accept /Deny Leave request

Url : http://localhost:8081/PayrollManagementSystem/ad_loan

The screenshot shows a web browser window with the URL localhost:8081/PayrollManagementSystem/ad_loan. The page displays a table of loan requests. The columns are: ID, Reason, User ID, Amount, Delete, and Accept. One row is visible, showing ID 1022, reason Study Loan, user ID 2026, amount 277200.0, and buttons for Delete and Accept.

Copyright © 2018 by C M Abdullah Khan

Figure 8.1.6 Accept /Deny Loan request

Url : http://localhost:8081/PayrollManagementSystem/loanreq

The screenshot shows a web browser window with the URL localhost:8081/PayrollManagementSystem/loanreq. The page contains a message: "You can apply for Loan , You can get Business loan = salary*6 , Medical loan = salary*3, Study loan = salary*2". Below this is a large empty input field. At the bottom left is a radio button group for "Medical" (selected), "Study", and "Business". A green "Submit" button is at the bottom right.

Copyright © 2018 by C M Abdullah Khan

Figure 8.1.7 Send Loan request

Url : http://localhost:8081/PayrollManagementSystem/leavereq

The screenshot shows a web browser window with the URL localhost:8081/PayrollManagementSystem/leavereq. The page has a header "Please Enter Date Range what you wanted to leave from given calendar". It features two date input fields: "Select Date From" and "Select Date To", both with placeholder text "dd/mm/yyyy". Below these is a text area for "Enter Your Leave Reason". At the bottom left is a radio button group for "Regular" (selected), "Study", and "Medical". A green "Submit" button is at the bottom right. A small inset window in the bottom right corner shows a preview of the leave request form.

Copyright © 2018 by C M Abdullah Khan

Figure 8.1.8 Send Leave request

Url : http://localhost:8081/PayrollManagementSystem/attendance

The screenshot shows a web application interface for managing payroll. At the top, there is a navigation bar with links to Home, Profile, Attendance, Leave, Loan, and Notification, along with a Log Out button. Below the navigation bar, the title "Your Last One Week Login Activity" is displayed. A table lists the login times and working hours for each entry:

Login Time	Working Hours
2018-12-15T12:42:26	0
2018-12-17T15:18:10	0
2018-12-17T20:08:56	0
2018-12-17T20:13:05	0
2018-12-17T20:22:38	0
2018-12-17T20:27:37	0
2018-12-17T20:30:03	0
2018-12-17T20:30:08	0
2018-12-17T20:30:13	0

Below the table, a section titled "Please Give Your Attendance" contains a green "Check In" button. At the bottom of the page, a copyright notice reads "Copyright © 2018 by C M Abdullah Khan".

Figure 8.1.9 Give Attendance

Url : http://localhost:8081/PayrollManagementSystem/report

The screenshot shows a web application interface for managing payroll reports. At the top, there is a navigation bar with links to Home, Profile, Registration, Enable, Leave, Loan, Notification, Salary, and Report, along with a Log Out button. Below the navigation bar, the title "Leave Status Quick Overview" is displayed. A table shows the count of leaves by type:

Leave Type	count(id)	Details
regular	14	Details
study	2	Details

Below the table, a section titled "Please Enter Date Range what you wanted to search from given calendar" includes fields for "Select Date From" and "Select Date To" (both dd/mm/yyyy), and radio buttons for "Attendance", "Salary", and "Loan". A green "Submit" button is present.

At the bottom of the page, a section titled "Search Based on Id" includes a text input field for "Enter your username", a date range selector for "Select Date From" and "Select Date To" (dd/mm/yyyy), and radio buttons for "Attendance", "Salary", and "Loan". A green "Submit" button is present. A copyright notice at the very bottom reads "Copyright © 2018 by C M Abdullah Khan".

Figure 8.1.10 Search Information

Url : http://localhost:8081/PayrollManagementSystem/permission_for_pay_salary

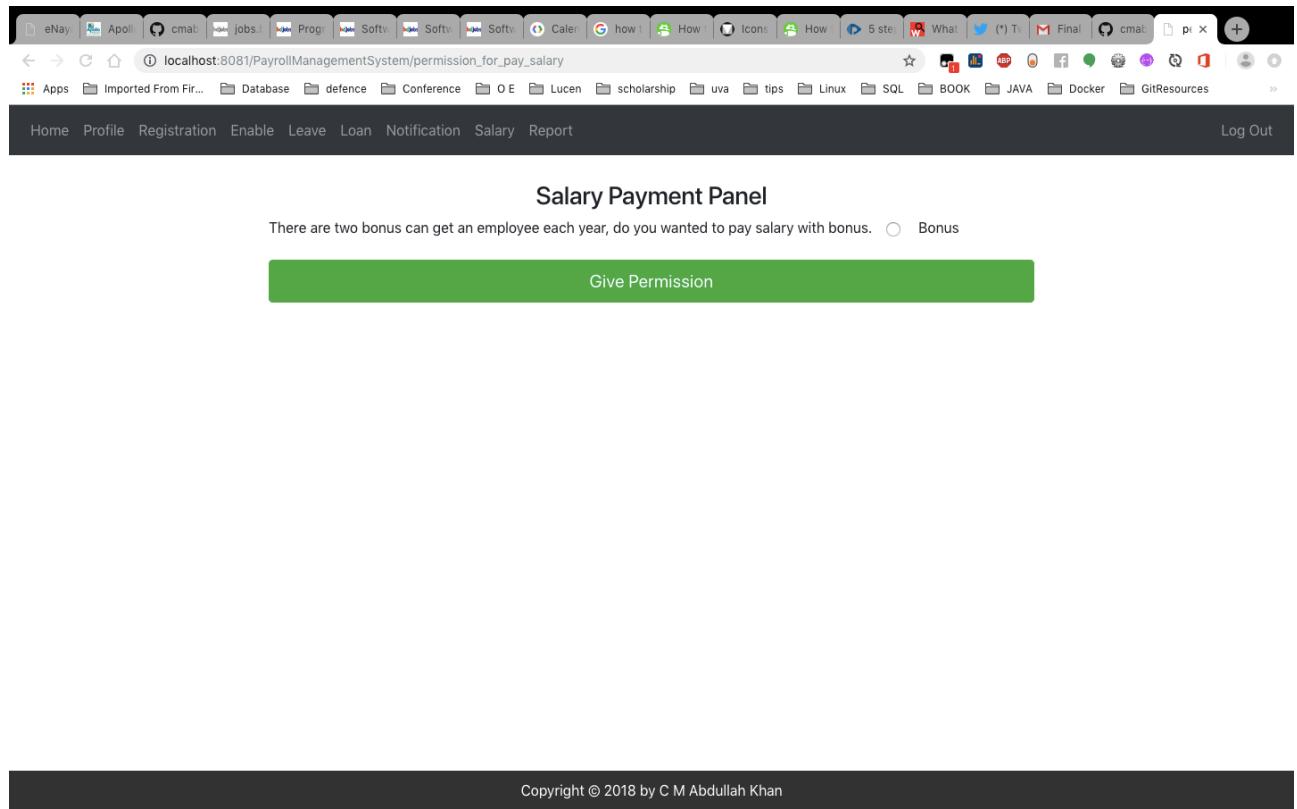
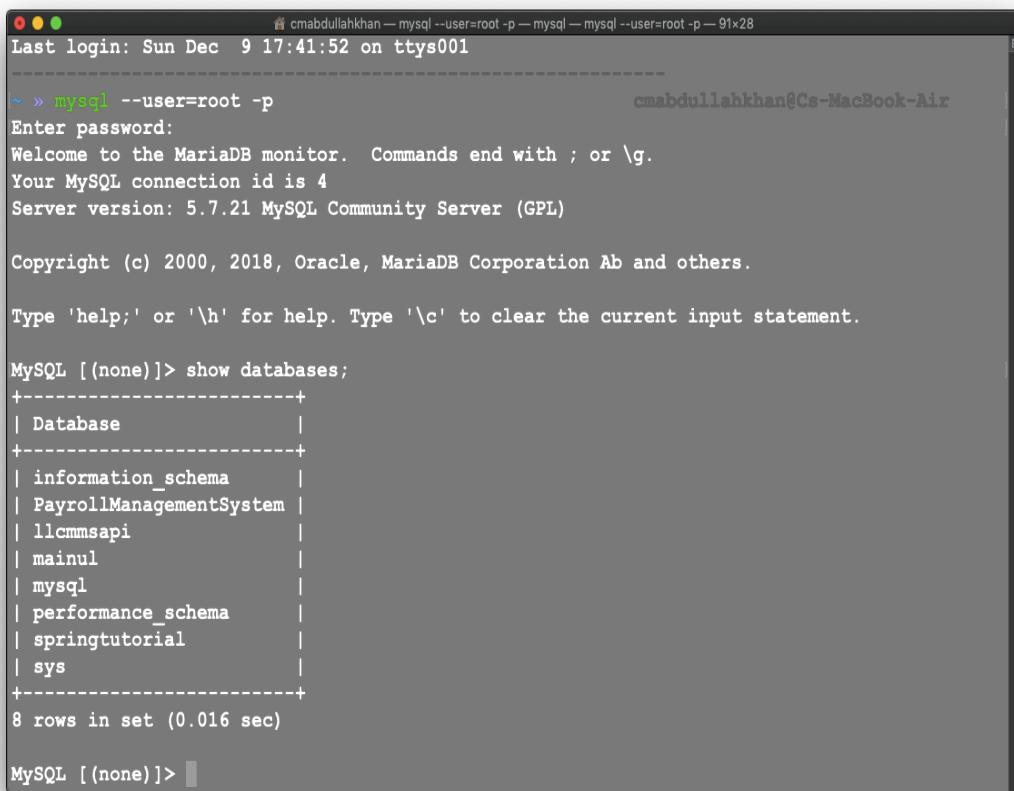


Figure 8.1.11Salary Payment Panel

8.2 Database Design

8.2.1 Database



```
cmabdukkhan: ~ cmabdukkhan — mysql --user=root -p — mysql — mysql --user=root -p — 91x28
Last login: Sun Dec  9 17:41:52 on ttys001

~ » mysql --user=root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.21 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

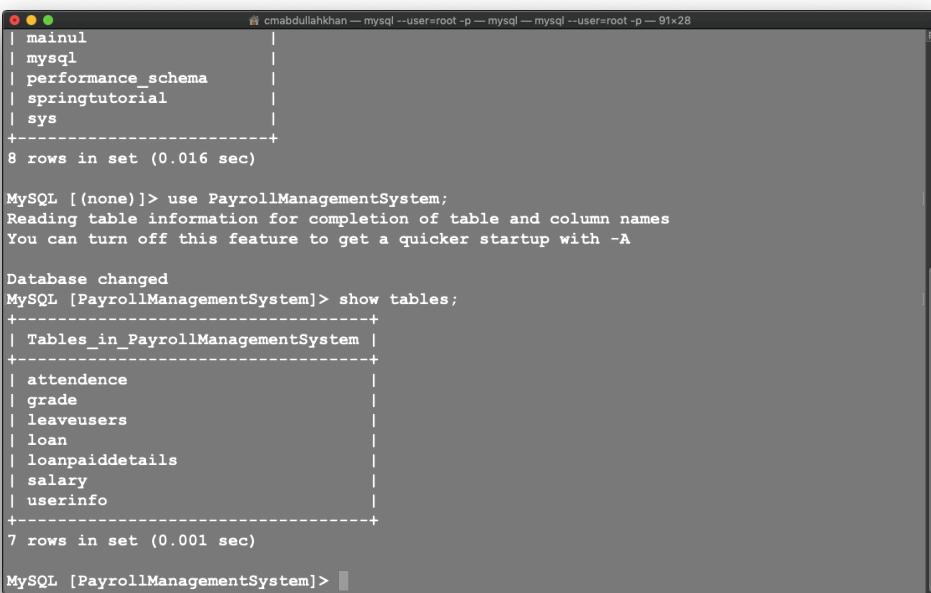
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases;
+-----+
| Database      |
+-----+
| information_schema |
| PayrollManagementSystem |
| llcmmmsapi |
| mainul |
| mysql |
| performance_schema |
| springtutorial |
| sys |
+-----+
8 rows in set (0.016 sec)

MySQL [(none)]>
```

Figure 8.2.1 Database

8.2.2 Database Table



```
cmabdukkhan: ~ cmabdukkhan — mysql --user=root -p — mysql — mysql --user=root -p — 91x28
| mainul      |
| mysql       |
| performance_schema |
| springtutorial |
| sys         |
+-----+
8 rows in set (0.016 sec)

MySQL [(none)]> use PayrollManagementSystem;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [PayrollManagementSystem]> show tables;
+-----+
| Tables_in_PayrollManagementSystem |
+-----+
| attendence |
| grade      |
| leaveusers |
| loan       |
| loanpaiddetails |
| salary     |
| userinfo   |
+-----+
7 rows in set (0.001 sec)

MySQL [PayrollManagementSystem]>
```

Figure 8.2.2 Database Table

8.2.3 Database for UsersInfos

```
cmabdullahkhan — mysql --user=root -p — mysql — mysql --user=root -p — 91x28
| grade
| leaveusers
| loan
| loanpaiddetails
| salary
| userinfo
+-----+
7 rows in set (0.001 sec)

MySQL [PayrollManagementSystem]> describe userinfo;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id | int(11) | NO | PRI | NULL | auto_increment |
| username | varchar(45) | NO | | NULL |
| fullname | varchar(45) | YES | | NULL |
| address | varchar(250) | YES | | NULL |
| email | varchar(45) | NO | | NULL |
| phone | int(14) | NO | | NULL |
| password | varchar(100) | NO | | NULL |
| authority | varchar(45) | YES | | NULL |
| enabled | tinyint(4) | YES | | NULL |
| joiningDate | datetime | YES | | NULL |
| grade_id | int(11) | NO | MUL | NULL |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0.010 sec)

MySQL [PayrollManagementSystem]>
```

Figure 8.2.3 Database for UsersInfos

8.2.4 Database for salary

```
cmabdullahkhan — mysql --user=root -p — mysql — mysql --user=root -p — 91x28
| username | varchar(45) | NO | | NULL | |
| fullname | varchar(45) | YES | | NULL | |
| address | varchar(250) | YES | | NULL | |
| email | varchar(45) | NO | | NULL | |
| phone | int(14) | NO | | NULL | |
| password | varchar(100) | NO | | NULL | |
| authority | varchar(45) | YES | | NULL | |
| enabled | tinyint(4) | YES | | NULL | |
| joiningDate | datetime | YES | | NULL | |
| grade_id | int(11) | NO | MUL | NULL | |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0.010 sec)

MySQL [PayrollManagementSystem]> describe salary;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id | int(11) | NO | PRI | NULL | auto_increment |
| datemonthyear | datetime | YES | | NULL | |
| Loan_id | int(11) | NO | MUL | NULL | |
| grade_id | int(11) | NO | MUL | NULL | |
| userinfo_id | int(11) | NO | MUL | NULL | |
| leaveusers_id | int(11) | NO | MUL | NULL | |
| totalsalary | int(11) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.001 sec)

MySQL [PayrollManagementSystem]>
```

Figure 8.2.4 Database for salary

8.2.5 Database for loan paid details

```
cmabdullahkhan — mysql --user=root -p — mysql — mysql --user=root -p — 91x28
11 rows in set (0.010 sec)

MySQL [PayrollManagementSystem]> describe salary;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+
| id    | int(11) | NO   | PRI  | NULL    | auto_increment |
| datemonthyear | datetime | YES  |      | NULL    |             |
| Loan_id | int(11) | NO   | MUL  | NULL    |             |
| grade_id | int(11) | NO   | MUL  | NULL    |             |
| userinfo_id | int(11) | NO   | MUL  | NULL    |             |
| leaveusers_id | int(11) | NO   | MUL  | NULL    |             |
| totalsalary | int(11) | YES  |      | NULL    |             |
+-----+-----+-----+-----+-----+
7 rows in set (0.001 sec)

MySQL [PayrollManagementSystem]> describe loanpaiddetails;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+
| id    | int(11) | NO   | PRI  | NULL    | auto_increment |
| datetime | datetime | YES  |      | NULL    |             |
| paidamount | float   | YES  |      | NULL    |             |
| loan_id | int(11) | NO   | MUL  | NULL    |             |
+-----+-----+-----+-----+-----+
4 rows in set (0.001 sec)

MySQL [PayrollManagementSystem]>
```

Figure 8.2.5 Database for loan paid details

8.2.6 Database for loan.

```
cmabdullahkhan — mysql --user=root -p — mysql — mysql --user=root -p — 91x28
7 rows in set (0.001 sec)

MySQL [PayrollManagementSystem]> describe loanpaiddetails;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+
| id    | int(11) | NO   | PRI  | NULL    | auto_increment |
| datetime | datetime | YES  |      | NULL    |             |
| paidamount | float   | YES  |      | NULL    |             |
| loan_id | int(11) | NO   | MUL  | NULL    |             |
+-----+-----+-----+-----+-----+
4 rows in set (0.001 sec)

MySQL [PayrollManagementSystem]> describe loan;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+
| id    | int(11) | NO   | PRI  | NULL    | auto_increment |
| placedate | datetime | NO   |      | NULL    |             |
| approvedate | datetime | YES  |      | NULL    |             |
| amount | int(11) | YES  |      | NULL    |             |
| status | int(11) | YES  |      | NULL    |             |
| userinfo_id | int(11) | NO   | MUL  | NULL    |             |
| reason | varchar(450) | YES  |      | NULL    |             |
+-----+-----+-----+-----+-----+
7 rows in set (0.001 sec)

MySQL [PayrollManagementSystem]>
```

Figure 8.2.6 Database for loan.

8.2.7 Database for Leave Users.

```
cmabdullahkhan — mysql --user=root -p — mysql — mysql --user=root -p — 91x18
7 rows in set (0.001 sec)

MySQL [PayrollManagementSystem]> describe leaveusers;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| id | int(11) | NO | PRI | NULL | auto_increment |
| reasone | varchar(45) | YES | | NULL | |
| status | int(11) | YES | | NULL | |
| leavetype | varchar(45) | YES | | NULL | |
| userinfo_id | int(11) | NO | MUL | NULL | |
| entryfrom | datetime | YES | | NULL | |
| entryto | datetime | YES | | NULL | |
| total_leave_days | int(11) | YES | | NULL | |
+-----+-----+-----+-----+-----+
8 rows in set (0.001 sec)

MySQL [PayrollManagementSystem]>
```

Figure 8.2.7 Database for Leave Users

8.2.8 Database for grade.

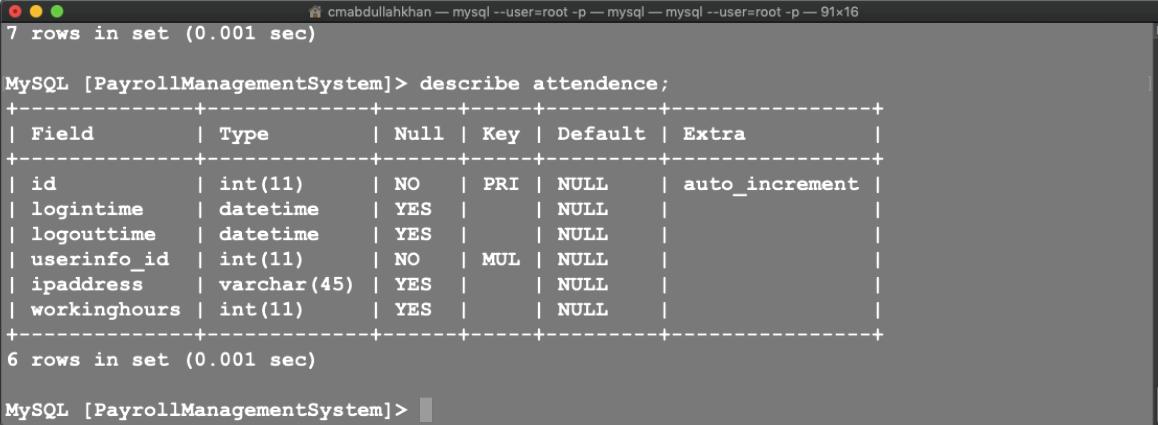
```
cmabdullahkhan — mysql --user=root -p — mysql — mysql --user=root -p — 91x18
8 rows in set (0.001 sec)

MySQL [PayrollManagementSystem]> describe grade;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| id | int(11) | NO | PRI | NULL | |
| basic | int(11) | YES | | NULL | |
| medicalallowence | int(11) | YES | | NULL | |
| houserent | int(11) | YES | | NULL | |
| transport | int(11) | YES | | NULL | |
| lunch | int(11) | YES | | NULL | |
| study | int(11) | YES | | NULL | |
+-----+-----+-----+-----+-----+
7 rows in set (0.001 sec)

MySQL [PayrollManagementSystem]>
```

Figure 8.2.7 Database for grade.

8.2.8 Database for attendance.



The screenshot shows a terminal window titled "cmabdullahkhan — mysql --user=root -p — mysql — mysql --user=root -p — 91x16". The command "describe attendence;" is run, resulting in 6 rows in set (0.001 sec). The output is a table describing the columns of the "attendence" table:

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
logintime	datetime	YES		NULL	
logouttime	datetime	YES		NULL	
userinfo_id	int(11)	NO	MUL	NULL	
ipaddress	varchar(45)	YES		NULL	
workinghours	int(11)	YES		NULL	

MySQL [PayrollManagementSystem]>

Figure 8.2.8 Database for attendance.

Chapter 09

Quality Assurance and Testing

According to the common process framework (CPF), the software testing is the final activity that has to initiate after testing. Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation.

The objectives of software testing are:

- ✓ Testing is a process of executing a program with the intent of finding an error.
- ✓ A good test case is one that has a high probability of finding an as-yet-undiscovered error.
- ✓ A successful test is one that uncovers an as-yet-undiscovered error.

The design of tests for software can be challenging as the initial design of the product itself.

Software can be tested in one of two ways:

- ✓ Knowing the specified function that the software has been designed to perform, tests can be conducted that demonstrate each function fully while at the same time searching for errors in each function. This approach is known as black-box testing.
- ✓ Knowing the internal workings of software, tests can be conducted to ensure that internal operations are performed according to specifications and all internal components have been adequately exercised. This approach is known as white-box testing.

9.1 Software Testing Strategy

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of a software. The strategy provides a road map that describes the steps to be conducted as part of testing.

Testing strategy that will be followed in this software project –

- ✓ Unit testing
- ✓ Integration testing
- ✓ Validation testing

The first step in software testing is unit testing. Unit testing concentrates on each unit of the software as implemented in source code. Unit testing focuses on each component individually. The unit test is white-box oriented. Thus, unit testing of this library software will be done after completion of every module or component.

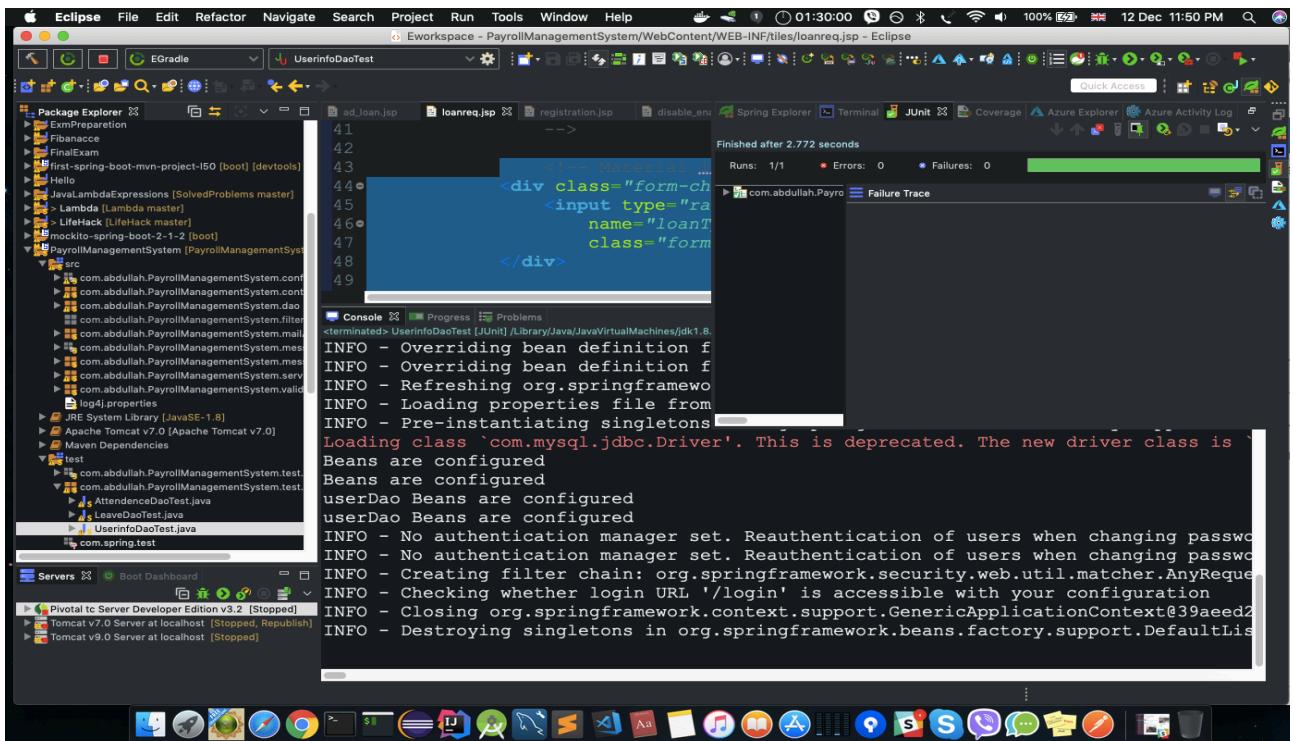


Figure 9.1 Unit test report

The next step is integration testing. Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective of integration testing is to take unit tested components and build a program structure that has been dictated by design.

The integration testing strategy that has been chosen for this project is top down testing. Black-box testing method is the most prevalent for integration testing. Top down integration strategy will be used to perform integration testing. Top down integration will be done by breadth-first manner. Breadth-first integration incorporates all components directly subordinate at each level, moving across the structure horizontally.

After the software has been integrated, a set of high order tests are conducted. Hence, the validation criteria that have been mentioned in requirements engineering should be tested. Validation testing provides final assurance that software meets all functional, behavioral and performance requirements. The black-box testing method is exclusively used in validation.

9.2 System Testing Methodology

✓ Black-box Testing

Black-box testing which is also known as behavioral testing focuses on the functional requirements of the software. It enables the software engineer to derive sets of input conditions that will fully

exercise all functional requirements for a program. Black-box testing method will be applied to test the modules of LMS.

✓ White-box Testing

White-box testing, which also known as glass-box testing, is a test case design method that uses the control structure of the procedural design to derived test cases. Using white-box testing methods, software engineer can derive test cases that-

1. guarantee that all independent paths within a module have been exercised at least once
2. exercise all logical decisions on their true and false sides
3. execute all loops at their boundaries and within their operational bounds
4. Exercise internal data structures to ensure their validity.

The modules that contain some complex calculations or decision making code such as check the availability of the library item will be tested using white-box method.

9.3 Testing Design

Table 9.3.1: Testing scenario 1

Testing scenario No: 1	
Scenario	Manager Login testing scenario of our system
Input's	ID, password of Manager for Login
Desired Output's	When enter ID, password then get access level define.
Actual Output's	For login our system work correctly
Verdict	Getting result from Desired Output's and Actual Output's decided this system is successful for login.

Table 9.3.2: Testing scenario 2

Testing scenario No: 2	
Scenario	Employee Login testing scenario of our system
Input's	ID, password of Employee for Login
Desired Output's	When enter ID, password then get access level define.
Actual Output's	For login our system work correctly
Verdict	Getting result from Desired Output's and Actual Output's decided this system is successful for login.

Table 9.3.3: Testing scenario 3

Testing scenario No: 2	
Scenario	Accountant Login testing scenario of our system
Input's	ID, password of Accountant for Login
Desired Output's	When enter ID, password then get access level define.
Actual Output's	For login our system work correctly
Verdict	Getting result from Desired Output's and Actual Output's decided this system is successful for login.

Table 9.3.4: Testing scenario 4

Testing scenario No: 4	
Scenario	Manager can accept or reject leave request.
Input's	Leave request status updated by the manager
Desired Output's	Applicants get the notification by the manager.
Actual Output's	For update status flag our system work correctly
Verdict	The process is worked correctly and successfully.

Table 9.3.5: Testing scenario 5

Testing scenario No: 4	
Scenario	Manager can accept or reject Loan request.
Input's	Loan request status updated by the manager
Desired Output's	Applicants get the notification by the manager.
Actual Output's	For update status flag our system work correctly
Verdict	The process is worked correctly and successfully.

Table 9.3.6: Testing scenario 6

Testing scenario No: 6	
Scenario	Manager can give salary payment permission.
Input's	Press pay salary button
Desired Output's	Accountant get payment permission notification...

Actual Output's	Accountant payment function enable...
Verdict	The process is worked correctly and successfully.

Table 9.3.7: Testing scenario 7

Testing scenario No: 7	
Scenario	Customer can add product to cart
Input's	Select product, give quantity
Desired Output's	Total price of product will be shown and order will be added in the cart
Actual Output's	Customer can see cart details
Verdict	The process is worked correctly and successfully.

Table 9.3.8: Testing scenario 8

Testing scenario No: 7	
Scenario	Customer can give product review
Input's	Select product, fill form
Desired Output's	Confirmation message will show to customer
Actual Output's	Customer can see the product review
Verdict	The process is worked correctly and successfully.

Table 9.3.9: Testing scenario 9

Testing scenario No: 9	
Scenario	Customer must pay for order
Input's	Select order, give transaction
Desired Output's	Total price of order is shown and give a pending confirmation message
Actual Output's	Customer can see confirmation message
Verdict	The process is worked correctly and successfully.

4. Quality Assurance Matrix

Table 9.3.10: Quality Assurance Matrix 1

Quality Assurance scenario no: 1	
Requirement	The system worked according to requirement.
P r o v i d e d outputs	My requirement was Manager enters the system using ID and password.
Decision	This system working correctly. We hope our system will work in future.

Table 9.3.11: Quality Assurance Matrix 2

Quality Assurance scenario no: 2	
Requirement	The system worked according to requirement.
P r o v i d e d outputs	My requirement was manager can register new employee
Decision	This system working correctly. We hope our system will work in future.

Table9.3.12: Quality Assurance Matrix 3

Quality Assurance scenario no: 3	
Requirement	The system worked according to requirement.
P r o v i d e d outputs	My requirement was manager can accept or deny leave request.
Decision	This system working correctly. We hope our system will work in future.

Table9.3.13: Quality Assurance Matrix 4

Quality Assurance scenario no: 4	
Requirement	The system worked according to requirement
Provided outputs	My requirement was manager can accept or deny loan request.
Decision	This system working correctly. We hope our system will work in future.

Table9.3.14: Quality Assurance Matrix 5

Quality Assurance scenario no: 5	
Requirement	The system worked according to requirement

Provided outputs	My requirement was manager send salary given permission notification.
Decision	This system working correctly. We hope our system will work in future.

Table9.3.15: Quality Assurance Matrix 6

Quality Assurance scenario no: 6	
Requirement	The system worked according to requirement
P r o v i d e d outputs	My requirement was accountant can give salary with bonus (if Bonus given)
Decision	This system working correctly. We hope our system will work in future.

Table 9.3.16: Quality Assurance Matrix 7

Quality Assurance scenario no: 7	
Requirement	The system worked according to requirement
Provided outputs	My requirement was employee can send loan request
Decision	This system working correctly. We hope our system will work in future.

Table9.3.17: Quality Assurance Matrix 8

Quality Assurance scenario no: 8	
Requirement	The system worked according to requirement
Provided outputs	My requirement was employee can send leave request
Decision	This system working correctly. We hope our system will work in future.

Chapter 10

Conclusion

1. Conclusion

The goal of this project “Development of Payroll System Implementation Lets Learn Coding” was to a web based payroll system using Java Core, Spring MVC, Spring AOP, Spring Security and JavaScript. This computer based Payroll system is accessible on the internet and calculates, maintains and records the payroll information of employees. This Application will help to automate payroll system of an organization. Multiple authorized users will be able to login and logout from a web browser. Login checks (username, password) are controlled by administrator. Administrator will have total web based control to completely customize the payroll system. Manager will be able to authenticate new employees, view reports while the operator may calculate pay and can only view reports. The system is user friendly. Whenever there is an error in entering data, it immediately shows an error. The application is equipped with tools for updating salary records, add new allowances, deduction and bonus and many other features that are easy to be operated by users. Every individual element in system can be added, viewed and updated online as per individual's rights. The system has also provision for full salary history including all payroll elements and changes that have been implemented. The prototype computer based payroll system is complete in itself and ready to be implemented but changes and growth in requirements will be a reality on every software project so there is need to timely update them. The same applies to this payroll system. The four years of undergraduate engineering studies gives a student theoretical and practical knowledge. Using that knowledge and observing live operational system, the practicum program clarifies those subjects' matters to another level blessed with practical working skills. Considering this fact, it gives us an immense pleasure to say that our practicum was a successful event. Lets Learn Coding gives us the opportunity of working in a professional working environment. During the internship period I have tried my level best to make my system efficient. I followed the lessons, methods, tools and techniques that I have learned during my study period at IUBAT. Successful software development is a blend of standard development practices, proper theoretical knowledge and the developer's creativity.

10.2 The Experience

Working in this project was a big opportunity for me. The following will indicate my learning from this project

- ✓ The designing & development strategy of a online based application.
- ✓ The analyzing strategy of software.
- ✓ Database management.
- ✓ New environment of programming languages.
- ✓ Hands-on experience about system security management.
- ✓ Learn new unit test tools.
- ✓ Amazing experience to work with security system.
- ✓ Familiar with deployment

10.3 Upcoming Features

i. Employee will get salary automatically through bKash account.

Right now employee have to get their salary via hand-cash. After development this system employee can get salary via bkash or visa card.

ii. More Loan features will be available .

Right now three type's of loan available.

iii. Password Security will strong.

Right now system takes any password. In future password will 3 stage. Like latter, number and symbol.

iv. Open Source Contribution.

This project is completely open source. This project is released under GNU GENERAL PUBLIC LICENSE Version 3. [<https://github.com/cmabdullah/PayrollManagementSystem/>]

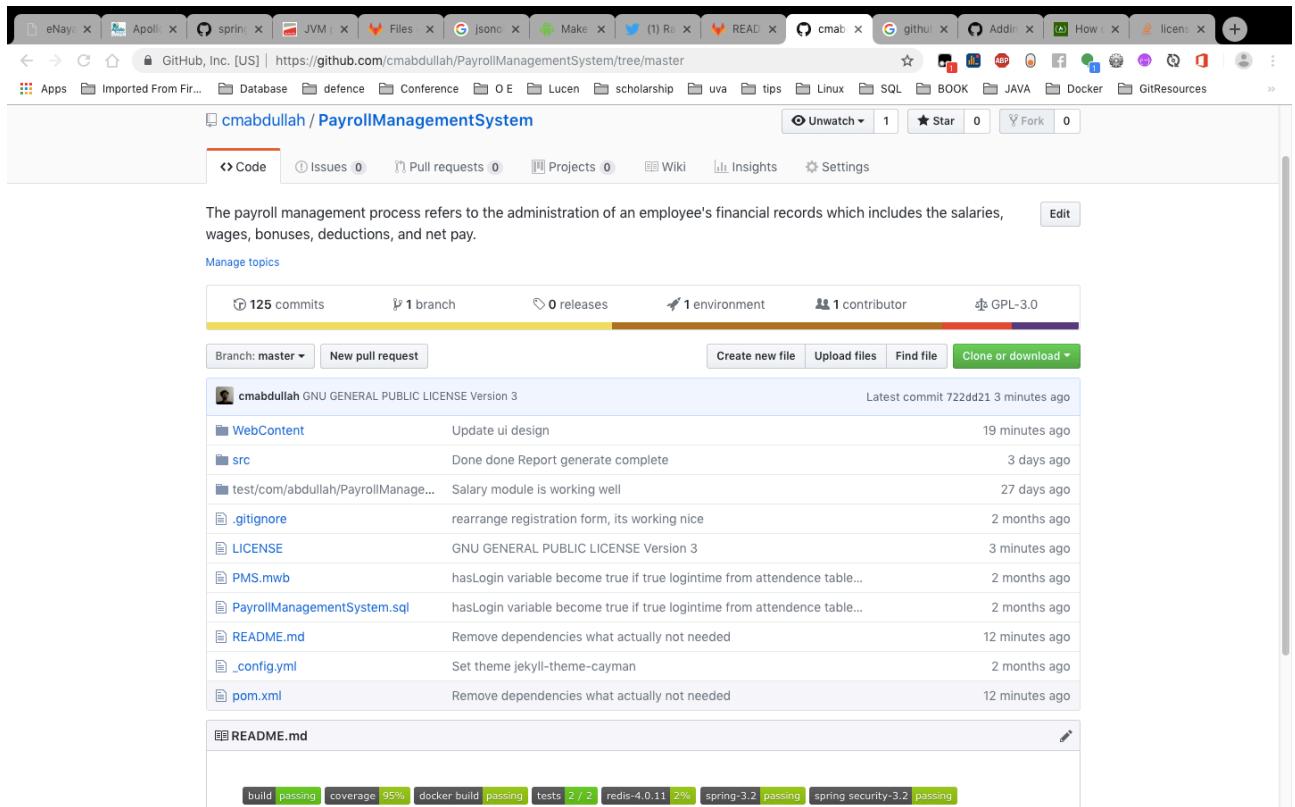


Figure 10.3.1 Github repository.

10.4 Glossary

Attribute - Attributes are the particular properties which are used to describe the entity.

CC-Customer Communication

CE- Customer Evaluation

DFD- Data Flow Diagram

DFD-Data Function

DET-Data Element

EI-External Input

EO-External Output

EIF-External Interface File

Entity - An entity is an object with physical existence or may be an object with conceptual existence.

Entity Relationship (ER) diagrams - The Entity Relationship diagram describes data as entities, relationships and attributes. ERD is known as the graphical representation of the database.

Estimation - A project planning activity that attempts to project effort project effort and cost for a project.

Function Point (FP) - A measure of the utility delivered by an application.

GSC-General System Characteristics

PMS- Payroll Management System.

ISP- Internet Service Provider

ILF-Internal Logical File

IT-Information Technology

References

1. Spring security by robert winch
2. Silberschatz, Abraham, Korth, Henry F., and Sudrashan S. Database System Concepts. 4th ed. Boston: McGraw Hill, 2002.
3. Kendall, E. Kendall. System Analysis and Design. 4thEd. New Delhi: Prentice Hall. 1999.
4. O'Brien, James and M. Markas, George. Management Information System. 8th International ed.2008.
5. Miller, Dave. Data Communication and Networks. 1st ed.2006.
6. S. Pressman, Roger, Software Engineering A Practitioner's Approach, 6th edition, 2008
7. Bell, "Underwater Archaeological Survey Report Template: A Sample Document for Generating Consistent Professional Reports," Underwater Archaeological Society of Chicago, Chicago, 2012.

Websites

1. <https://docs.spring.io/spring-security/site/docs/3.0.x/reference/taglibs.html>
2. https://www.tutorialspoint.com/redis/redis_java.htm
3. <https://www.lucidchart.com/pages/ER-diagram-symbols-and-meaning>