

SMART LEAVE MANAGEMENT SYSTEM

BY

SADIKA JAFRIN
ID: 201-16-498

This Report Presented in Partial Fulfillment of the Requirements for the Degree
of Bachelor of Science in Computing and Information System.

Supervised By

Ms. Sonia Nasrin
Lecturer
Department of CIS
Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

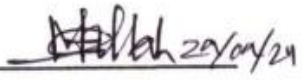
SEPTEMBER 29, 2024



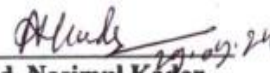
APPROVAL

This Project title “Smart Leave Management System” Submitted by Sadika Jafrin ID No 201-16-498 to the Department of Computing & Information System, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computing & Information System and approved as to its style and contents. The presentation has been held on 29-09-2024.

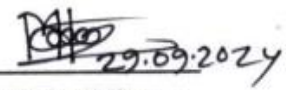
BOARD OF EXAMINERS


Md Sarwar Hossain Mollah
Associate Professor and Head
Department of Computing & Information System
Faculty of Science & Information Technology
Daffodil International University

Chairman


Md. Nasimul Kader
Assistant Professor
Department of Computing & Information System
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner


Md. Mehedi Hassan
Lecturer
Department of Computing & Information System
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

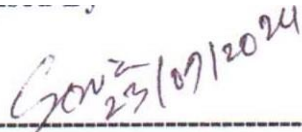

Md. Miftah Uddin
Head of SBU,
Brain Station 23

External Examiner

Declaration

I hereby declare that; this project has been done by me under supervision of Ms Sonia Nasrin, department of Computing and Information System (CIS) of Daffodil International University. I am also declaring that this project or any part of it has never been submitted anywhere else for the award of any educational degree like B.Sc.

Supervised By



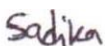
Ms. Sonia Nasrin

Lecturer

Department of CIS

Daffodil International University

Submitted By



Sadika Jafrin

201-16-498

Department of CIS

Daffodil International University

ACKNOWLEDGMENT

Finally, I want to thank DCL for choosing me as an intern trainee since this internship program has taught me a lot. It is believed that a teacher is a student's second parent since they provide them with the proper guidance, allowing them to progress in life.

In my internship program, I had two professors who always advised me in making the best decisions and provided me with advice on how to successfully perform challenging tasks. They never allowed me to struggle alone and were constantly reducing my stress by providing me with mental fortitude. One of them is **Md. Mazharul Islam Masum**, who is my intern trainer, and **Ms. Sonia Nasrin, Lecturer** supervise my academic work.

We are grateful to our fellow classmates at Daffodil International University, who engaged in discussions and provided support during the course of this project.

Dedication

This project is dedicated to Almighty Allah, secondly my Mom, Dad, and my elder sister who have all made it possible for me to complete my BSC Program, I return glory and honor to Almighty God for supporting me, protecting me, and also guiding me. I also thank God for the blessing throughout my program.

ABSTRACT

This research suggests creating an online Human Resource Management System (HRMS) with a particular emphasis on leave administration. The system's goal is to automate and expedite an organization's leave application, approval, and tracking procedures. Through the implementation of streamlined processes and the centralization of leave data, the HRMS aims to lower administrative costs, raise employee satisfaction, and boost overall operational effectiveness. Important functions including automatic notifications, policy enforcement, tracking of leave balances, and payroll system integration will all be included in the system. By implementing an Agile development approach, the system will be improved repeatedly to satisfy the organization's changing requirements. In the end, the HRMS Leave Management System will help create a more transparent, efficient, and well-organized leave management process, which will boost output.

Keywords: HRMS, leave management, automation, employee self-service, workflow, productivity.

TABLE OF CONTENTS

CONTENTS	PAGE
Board of examiners	ii
Declaration	iii
Acknowledgments	iv
Dedication	v
Abstract	vi
CHAPTER	
CHAPTER 1: INTRODUCTION	1
CHAPTER 2: INITIAL STUDY	2-3
2.1 Project Proposal	2
2.2 Background of the project	2
2.3 Problem Area	2
2.4 Possible solution	3
CHAPTER 3: RESEARCH METHODOLOGY	4-6
3.1 Best Features	4
3.2 Limitations	5
3.3 Recommended approach	6
CHAPTER 4: EXPERIMENTAL RESULTS AND DISCUSSION	7-11
4.1 What to use	7
4.2 Why to use	8
4.3 Sections of methodology	9
4.4 Implementation plans	10
CHAPTER 5: PLANNING	12-22
5.1 Impact on Society	12
5.1.1 Work Breakdown Structure (WBS)	12
5.1.2 Resource Allocation	16
5.1.3 Time Duration / Time Boxing	17
5.2 Testing Plan	19

5.3 Risk Management	20
5.3.1 Risk Identification	20
5.3.2 Risk Assessment	21
5.3.3 Steps Taken for Possible Risks	21
5.4 Sustainability Plan	21
5.4.1 Rules Applied to Maintain Quality	21
5.4.2 DSDM Atern Standard Quality Measures	21
5.4.3 Quality Plan and Measuring Meter	22
CHAPTER 6: FEASIBILITY	23-26
6.1 All possible type of feasibility	23
6.2 Cost Benefit Analysis	24
6.3 DSDM – good or not for this project -PAQ	25
6.4 Project Approach Questionnaire (PAQ)	26
CHAPTER 7: FOUNDATION	27-31
7.1 Problem Area Identification	27
7.2 Overall Requirement List	28
7.3 Which Technology to be implemented	29
7.4 Recommendation and Justifications	30
CHAPTER 8: EXPLORATION	32-33
8.1 Activity Diagram	32
8.2 Full System Use Case	33
CHAPTER 9: ENGINEERING	34-41
9.1 Modules of the new system	34
9.2 Use Case	35
9.3 Class Diagram	36
9.4 Peter Chen EERD Diagram	37
9.5 Sequence Diagram	38
9.6 Component Diagram	39
9.7 Deployment Diagram	40
CHAPTER 10: DEPLOYMENT / DEVELOPMENT	42-49
10.1 Core Module Coding Samples	42

10.2 Possible Problem Break Down	47
10.3 Prioritization while developing	48
CHAPTER 11: TESTING	50-52
11.1 Testing Plan	50
11.2 Testing Strategy	51
11.3 Acceptance	52
CHAPTER 12: IMPLEMENTATION	53-54
12.1 Implementation Environment	53
12.2 Coding Standards	53
12.3 Coding Scenario	53
12.4 Load Balancing	54
CHAPTER 13: CRITICAL APPRAISAL AND EVALUATION - HRMS	54-55
13.1 Objectives that Could Be Met	54
13.2 Future Work	54
13.2.1 Why it could not be touched	55
CHAPTER 14: LESSONS LEARNED	56-58
14.1 What Have I Learned	56
14.2 What Problems I Faced	56
14.3 What Solutions Occurred	57
14.4 Overall Reflection	58
CHAPTER 15: CONCLUSION	59-60
15.1 Summary of the project	59
15.2 Goals of the project	59
15.3 What I have done in the documentation (stages/ activities/ plans)	59
15.4 Value of the project	59
15.5 My Experience	60
APPENDIX	61
REFERENCE	62

LIST OF TABLES

TABLES	PAGE NO
Table 5.1: Breakdown Structure of DSMS	12-14
Table 5.2: Resource Allocation List	16
Table 5.3: List of the Time Boxes	17
Table 5.4: Hardware Requirements	18
Table 5.5: Software Requirements	18
Table 6.1: Cost Benefits	25
Table 10.1: Prioritization Requirement	49
Table 11.1: Final Test Report	55

LIST OF FIGURES

FIGURES	PAGE NO
Figure 4.1: Working Flow Diagram	7
Figure 5.1: HRMS Architecture	15
Figure 5.2: Gantt chart	19
Figure 5.3: Risk Management	20
Figure 5.4: DSDM Atern Standard Quality	22
Figure 7.1: Simple Data Flow	28
Figure 8.1: Activity Diagram	33
Figure 8.2: Full System Use Case Diagram	34
Figure 9.1: Use Case	36
Figure 9.2: Class Diagram	37
Figure 9.3: Peter Chen EERD Diagram	38
Figure 9.4: Sequence Diagram	39
Figure 9.5: Component Diagram	40
Figure 9.6: Deployment Diagram	41
Figure 11.1: Testing Plan	50

CHAPTER 1

INTRODUCTION

The Smart Leave Management System provides an all-digital method of managing employee leave. Employees can submit requests for leaves through an easy-to-use interface, which managers can assess and approve, and administrators can keep track of the organization's leave dynamics. In response to this demand, human resource management systems (HRMS) have developed, providing all-inclusive solutions for handling a range of HR tasks. Among these duties, leave administration stands out as an essential element that has a direct bearing on worker happiness and productivity. Organizations can handle employee leave in a completely new way with a Smart Leave Management System (SLMS) integrated within an HRMS. This system guarantees precise tracking, reduced procedures, and increased transparency. A leave management-focused Human Resource Management System (HRMS) provides a comprehensive platform for automating and streamlining leave-related procedures. An HRMS leave management system greatly improves operational efficiency and employee satisfaction by centralizing leave data, automating workflows, and offering real-time visibility into leave balances and approvals. Employees may now easily seek leaves of absence, supervisors can quickly approve or deny requests, and HR administrators can manage leave regulations and produce detailed information thanks to this technology. It assures adherence to company leave standards, cuts down on processing time, and does away with manual paperwork through automation. The goal of this research is to provide a solid HRMS leave management system with comprehensive reporting capabilities, an easy-to-use interface, and workflows that meet the unique requirements of businesses.

CHAPTER 2

INITIAL STUDY

2.1 Project Proposal:

Smart Leave Management System (SLMS) with our HRMS to simplify leave administration processes, raise detail, and improve the manager and employee experience. By automating leave requests, approvals, tracking, and reporting, the SLMS will lower the workload and guarantee adherence to labor laws and business procedures.

2.2. Background

An essential tool for HR administration, a leave management system (LMS) manages many facets of employee leave. The process of requesting, approving, tracking, and managing leaves is automated and made more efficient by the system. This was intended to be avoided by HRMS, which integrated the required applications into software modules. Every software module is a representation of the earlier HR systems. The fact that the different systems were designed and coded consistently to provide the same user experience across all modules is an advantage of integrating them into HRMS. Through the integration of several modules, HRMS functions as a comprehensive, one-stop solution for all HR procedures. For contemporary businesses to effectively manage employee leave, cut down on administrative burdens, and boost overall employee efficiency, a leave management system is an essential tool.

2.3 Problem Area

- **Manual Procedures:**

This is a slow process that can cause delays in approvals, which annoys executives and employees alike, loss of data, and laborious activities.

- **Errors in Leave Balances:**

Spreadsheets showing leave balances are manually updated. This approach may result in errors, such as inaccurate leave balances, that mislead workers and cause payroll problems.

- **Low Visibility:**

Insufficient information results in miscommunication, discontent, and poor preparation.

- **Differential Application of Policy:**

Differences may lead to unfair leave policies and possible legal problems for the business.

- **Restricted Reporting:**

It is difficult to plan the workforce and spot problems like severe absences without accurate reporting.

2.4 Possible solution

Benefits and Impact of a Possible Solution:

- **Automated Processes:** To automate the process of approving and requesting leaves, put in place a Smart Leave Management System (SLMS). By doing this, you'll expedite the procedure, cut down on delays, and lighten the administrative burden.
- **Accurate Leave Balances:** Real-time automatic updates of leave balances will be provided by the SLMS. This minimizes mistakes and confusion by ensuring precise information.
- **Increased Observation:** Establish a self-service site where managers can view team leave schedules and employees can monitor leave balances and request statuses.
- **Improved data reporting:** Detailed data on leave usage and trends will be available through the SLMS. Improved reports aid in the early detection of possible issues and promote well-informed decision-making.

CHAPTER 3

LITERATURE REVIEW

Technology and Literature Review

3.1 Best Features:

Analyzing previous research, publications, and papers on leave management systems, HRMS, and the larger subject of human resource technology are all part of the literature review process.

- **Automated Leave Management:**

Practical assistance, mistakes, and processing time are decreased by automated methods. Research suggests that automated leave management leads to a notable boost in worker productivity and satisfaction.

- **Connectivity with the HRMS:**

Accuracy and accuracy in data management are ensured through seamless connections with other HRMS sections, such as payroll and attendance.

- **Criteria for Compliance and Legality:**

Labor rules, industry standards, and data protection rules like GDPR must all be complied with by systems. Research talks about the consequences of non-compliance and the necessity of routine updates and audits.

- **Features of privacy:**

Access management is based on roles to limit access to data. To secure private information, use encrypted data and safe data storage.

- **Worker Self-Service**

Give staff members the freedom to independently monitor status, submit requests, and view leave balances. Self-service manager for examining and approving requests.

- **Compliance Management:**

Integrated compliance to data protection and labor legislation.

routine audits and modifications to guarantee continued compliance.

- **Online Application:**

Adaptable and scalable cloud architecture. lighter IT load and improved remote usability.

3.2 Limitations of Leave Management Software:

While an HRMS leave management system can offer numerous benefits, there are also limitations and challenges that organizations may encounter:

1. **Integration Difficulties:** Strong mechanisms for synchronization are necessary to maintain data consistency across interconnected systems.
2. **Assistance and User Utilization:** Effective programs for training and user uptake are essential for effective execution. Employees and managers could be reluctant to switch from manual processes to automated ones. Thorough training is required to guarantee that every user is able to use the system.
3. **Include initial and ongoing expenses:** It costs money to implement and operate an HRMS leave control system. Upfront Costs: The price of initial setup, adaptation, and instruction may be substantial. Subscription Costs: Cloud-based solutions frequently have a recurring membership cost.
4. **Outage and Technical Problems:** Business operations might be disrupted by technical issues and system outages. System Downtime: Maintenance procedures or server problems may cause system outages, which reduce productivity. Technical Errors: The system's dependability and functionality may be impacted by bugs and technical errors.
5. **Data security and privacy:** Robust security measures are necessary for sensitive candidate data. Put in place access controls, data encryption, and frequent security audits.

6. **Lack of human touch:** An excessive dependence on automation may result in an impersonal and sterile application process. Keep in touch with each other personally via phone conversations, emails, and video interviews.

3.3 Recommended Approach:

1. **Security focus:** Make data security a top priority by putting strong safeguards in place and keeping up of compliance requirements.
2. **User Experience:** Make sure that all devices, including mobile ones, can access the system.
3. **Invest in training:** Give HR personnel and workers thorough instruction on how to use the program in an ethical and successful manner.
4. **Data-driven decision making:** Take advantage of data insights to make well-informed leave decisions while keeping personal preferences and biases in mind.
5. **Ability and capacity:** Select a system that can expand to accommodate the organization's growth. Check that the system is adaptable enough to take organizational and leave policy changes.
6. Transparency in the leaving process should be upheld, and any worries candidates may have regarding the use of technology should be addressed.

The right technology stack, efficient training, and ongoing development are necessary for the successful implementation of an HRMS Leave Management System. Organizations can achieve a successful implementation that boosts productivity, guarantees compliance, and raises user satisfaction by heeding this advice.

CHAPTER 4

METHODOLOGY

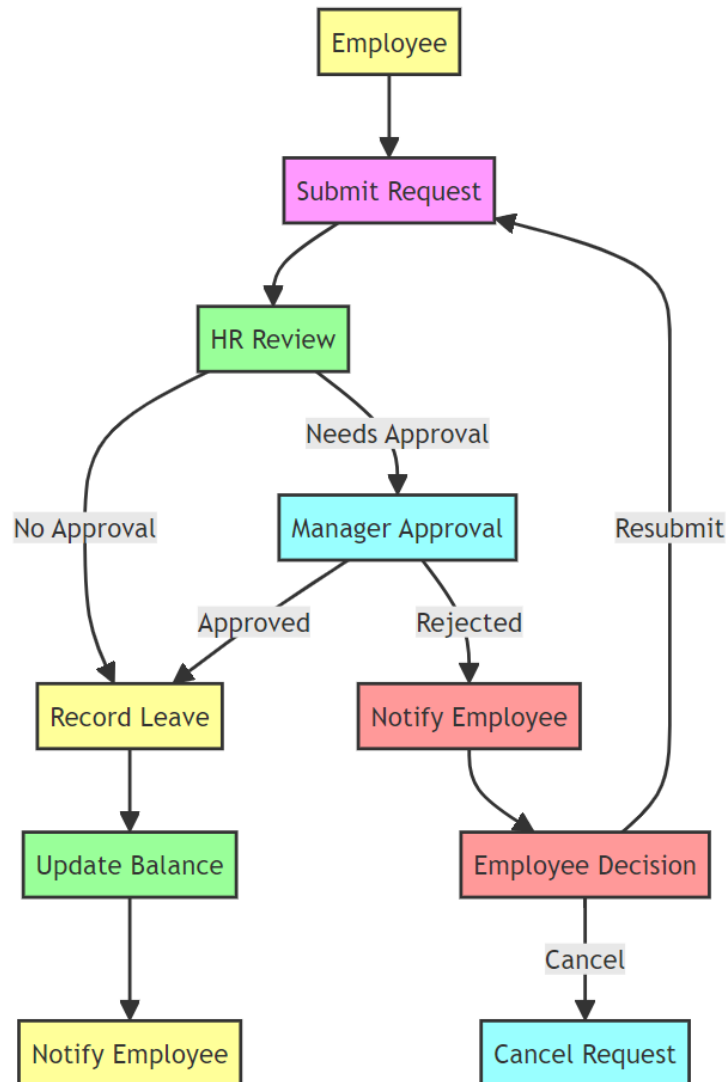


Figure 4.1: Working Flow Diagram

4.1 What to Use:

Technology Stack:

The choice of technology stack is critical for the successful implementation of HRMS leave management software. Considerations should include database management systems, programming languages, frameworks, and any third-party integrations.

- Database: MySQL
- Programming Language: Java, JavaScript
- Frameworks: Spring Boot (Java).

Development Tools:

Select development tools that facilitate collaboration, version control, and project management. Common tools include Git for version control, Jira for project management, and collaborative platforms like Slack or Microsoft Teams.

4.2 Why to Use:

Scalability: The HRMS leave management software's scalability requirements should be supported by the technological stack that is selected. The system should be able to accommodate more data, users, and transactions as the company expands.

Security: When managing sensitive HR data, security is crucial. Data encryption, secure authentication, and access control are examples of industry-standard security measures that should be followed by the selected technology and development processes.

Integration Capabilities: The technologies' ability to be integrated should be taken into account. To guarantee a smooth workflow, the software should interface with databases, third-party apps, and current HR systems with ease.

4.3 Methodology Sections:

Designing the Project and Organizing the Demands: To define the scope, objectives, and requirements of the project.

1. Specify the project's parameters
2. Assemble Materials
3. Documentation Needs
4. Develop a project schedule

Technology Selection: Justify the selection of the chosen technology stack, providing a rationale for each component. Consider factors such as cost, ease of maintenance, community support, and alignment with organizational goals.

Management and Continuous Growth: to keep the system up to date and make constant improvements in response to user input and shifting requirements.

1. Observation
2. Assistance
3. Modifications and Improvements
4. Reward Cycle
5. Record-keeping.

Testing and Quality Assurance: Put into practice a thorough testing plan that includes user acceptability, integration, and unit testing. Before deploying the program, make sure it satisfies quality criteria and is free of serious problems.

Project lifecycle Phase: There are various stages in the project lifecycle for creating an HRMS Leave Management System. These stages adhere to the Agile development methodology's

tenets, guaranteeing adaptability, ongoing input, and iterative advancement. The normal stages of this lifecycle are listed below:

- 1. Initiation Phase:** Specify the goals, high-level criteria, and scope of the project.
- 2. Planning Phase:** Create a thorough project plan and optimize needs.
- 3. Design Phase:** Produce comprehensive designs for the database, user interfaces, and system architecture.
- 4. Development phase:** To create the HRMS Leave Management System using the authorized designs.
- 5. Testing Phase:** Verify that the system satisfies the established requirements and operates as intended.
- 6. Maintenance and Further Growth Phase:** Keep the system up to date and make constant improvements in response to user input and shifting requirements.

4.4. Implementation Plans:

The HRMS Leave Management System implementation strategy guarantees a thorough approach to system deployment, training, and support. By taking these actions, the company can ensure that users are prepared to make the most of the new features, limit disruptions during the system's transition, and accomplish a seamless rollout.

User Support and Feedback: Establish a user support system to address queries and issues promptly. Encourage continuous feedback from users to identify areas for improvement and future enhancements.

Assessment and Analysis: Carry out an assessment to appraise the execution procedure and pinpoint insights gained. Utilize predetermined measures to gauge success, such as business effect, system performance, and user happiness.

User Guide: Produce instructional videos, user manuals, and other training materials.

Organize instruction: Plan training courses for various user groups, such as managers, HR administrators, and employees.

Security Measures: Implement authentication methods (such as single sign-on and multi-factor authentication) as part of security measures. Make sure that data is encrypted both while it is in transit and at rest. Role-based access control can be configured to limit access to sensitive information and features.

Monitoring and Optimization: Implement monitoring tools to track system performance, user engagement, and potential issues. Continuously optimize the system based on user feedback and evolving business needs.

CHAPTER 5

PLANNING

5.1 Project plan

Goal: Create a web-based system that will maintain leave balances, handle employee leave requests, and give HR personnel the administrative tools they need to effectively manage leave records.

Key Feature:

- 1.The ability to submit and track employee leave requests is a key feature.
2. Give staff access to the balance display.
- 3..Dashboard for admin approval/rejection of leaves.
- 4.Email updates on the status of your leave.

Technology Stack:

- **Backend:** Java (Spring Boot), MySQL
- **Others:** Maven, Tomcat

Testing: (Chapter 11)

Activities: All module unit testing

Deliverables:
Case Studies

5.1.1 Work Breakdown Structure (WBS):

The Work Breakdown Structure (WBS) organizes a project's tasks in a hierarchical manner. It breaks down the project into smaller, more manageable components. Tasks can be further divided into subtasks as needed at each stage of the project..

Table 5.1: Breakdown Structure of DSMS

No.	Category	Sub-Category/Tasks	Details/Sub-Tasks
1	User Management	1.1 User Registration	1.1.1 Employee Registration
			1.1.2 Manager Registration
			1.1.3 Administrator Registration
		1.2 User Authentication	1.2.1 Login
			1.2.2 Logout
			1.2.3 Password Reset
		1.3 User Roles and Permissions	1.3.1 Define Roles
			1.3.2 Assign Permissions
2	Leave Application Management	2.1 Leave Application Submission	2.1.1 Leave Request Form
			2.1.2 Attachment Upload (Medical Certificates, etc.)
		2.2 Leave Application Processing	2.2.1 Manager Review
			2.2.2 Manager Approval/Rejection
			2.2.3 Notification System (Email/SMS)
		2.3 Leave Balances	2.3.1 Annual Leave Balance
			2.3.2 Sick Leave Balance
			2.3.3 Other Leave Types Balance (Maternity, Paternity, etc.)
3	Leave Types and Policies	3.1 Leave Types Definition	3.1.1 Annual Leave
			3.1.2 Sick Leave
			3.1.3 Casual Leave
			3.1.4 Maternity Leave
			3.1.5 Paternity Leave
			3.1.6 Compensatory Leave
		3.2 Leave Policy Setup	3.2.1 Accrual Rules
			3.2.2 Carry Forward Rules

			3.2.3 Leave Encashment
4	Approval Workflow	4.1 Workflow Design	4.1.1 Single Level Approval
			4.1.2 Multi-Level Approval
		4.2 Workflow Management	4.2.1 Define Approvers
			4.2.2 Escalation Rules
			5.1.1 Leave Balance Reports
			5.1.2 Leave Utilization Reports
			5.1.3 Absenteeism Reports
		5.2 Custom Reports	5.2.1 Report Builder Tool
			5.2.2 Export to Excel/PDF
		5.3 Analytics Dashboard	5.3.1 Key Performance Indicators (KPIs)
			5.3.2 Visual Analytics (Charts, Graphs)
6	Integration	6.1 HR Management System Integration	6.1.1 Employee Data Sync
		6.2 Payroll System Integration	6.2.1 Leave Encashment Data
		6.3 Calendar Integration	6.3.1 Company Holidays
			6.3.2 Personal Calendars (Google, Outlook)
7	Notifications and Alerts	7.1 Email Notifications	7.1.1 Leave Application Status
			7.1.2 Leave Balance Alerts
		7.2 SMS Notifications	
		7.3 In-App Notifications	
8	Audit and Compliance	8.1 Audit Trails	8.1.1 Leave Application History
			8.1.2 User Activity Logs
		8.2 Compliance Checks	8.2.1 Policy Adherence
			8.2.2 Legal Compliance
9	Security and Data Protection	9.1 Data Encryption	
		9.2 User Data Privacy	
		9.3 Access Control	

Level 1

- Gather user requirements
- Define system's scope and objectives
- Develop project plan and budget
- Sprint Planning

Overview: Refers to the systems development life cycle a set of processes or stages and all stages of the system where a number of steps and the main stage falls below. All the steps and methodologies are:

- Plan
- Analysis
- Design
- Implementation

HRMS Architecture:

HRMS, which is an online intranet system, will be used by four types of employees. These types of people who have different roles can be stated as; admin, manager, HR, or employee. Every user enters the main authentication page and after that, system will grant them authorization.

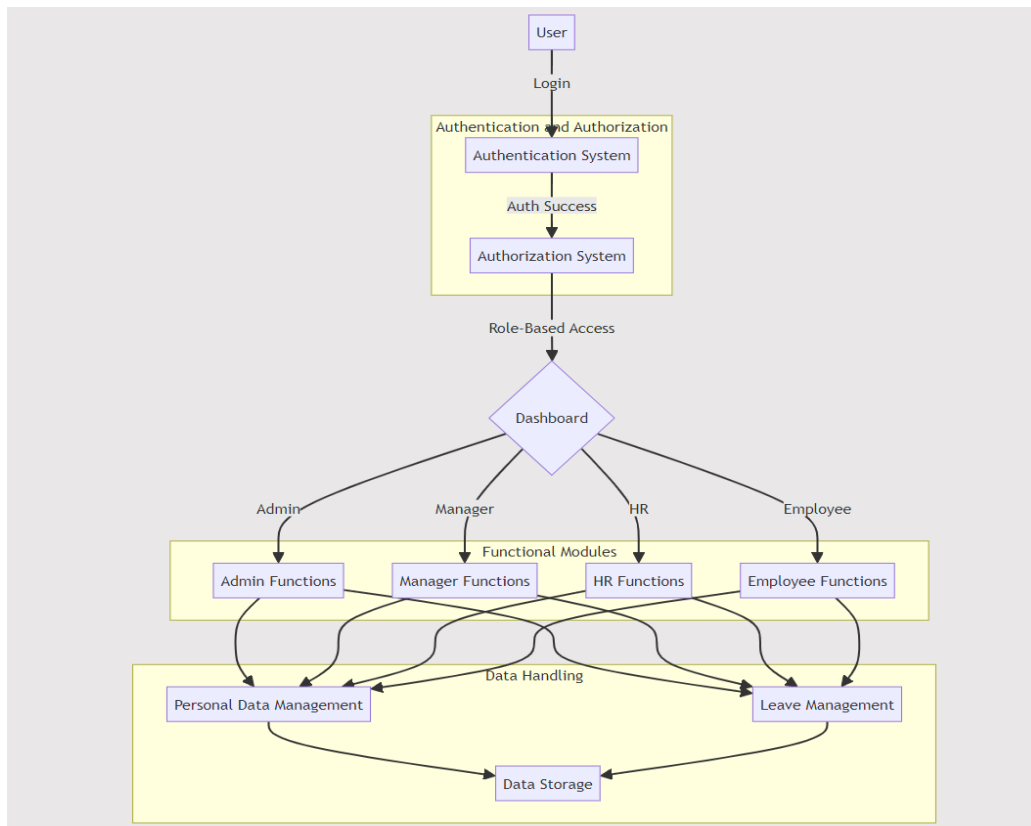


Figure 5.1: HRMS Architecture

Description of procedures and functions: This section will explain the major functions of HRMS along with the data flow. So, the major functionality of the project such as the authentication mechanism, personal data processing, leave system, and be explained step by step.

5.1.2 Resource Allocation

Study Of Current System

Analyzing the current setup in relation to an HRMS (Human Resource Management System) entails evaluating the technologies, processes, and user interactions in order to pinpoint potential for development as well as system strengths and shortcomings.

Time Boxing:

One way that a Time Box differs from standard progress control is that it incorporates project management variables such as the deliverable's scope. However, quality is always the same. The scope, quality, and anticipated completion date of the project must be continuously assessed by the project manager. The deadline cannot be fulfilled if it is not possible to further reduce the delivery's scope or quality.

Hardware Requirements:

These hardware requirements guarantee that the system can efficiently serve its users, ranging from administrators overseeing leave approvals and other administrative duties to regular staff monitoring leave balances. In order to guarantee seamless and effective system operation, the subsequent hardware requirements are advised.

Table 5.4: Hardware Requirements

Requirement	Specification
CPU	Dual Core 5 GHz or later
RAM	8 GB
Storage	Minimum 228 GB

Software Requirements:

Table 5.4: Hardware Requirements

Requirement	Specification
Web Browser	
Integrated Development Environment	IntelliJ IDEA, Visual Studio Code
Database Management System	MySQL

5.1.3 Gantt chart:

This Gantt chart acts as a plan of action for the LMS, making sure that tasks are finished on time and the project moves forward without problems all the way to deployment and beyond.

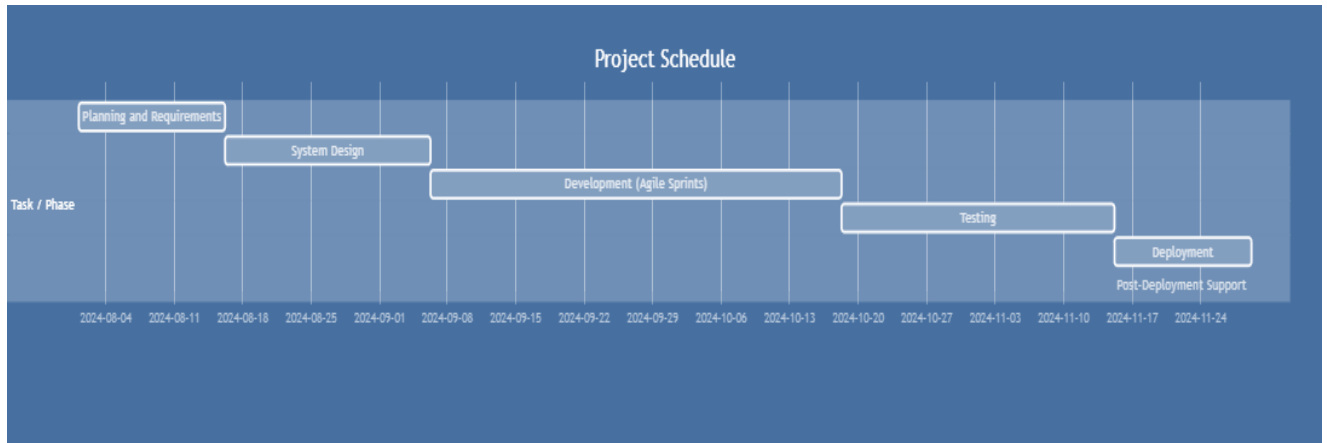


Figure 5.2: Gantt chart

5.2 Testing Plan

a well-defined testing approach guarantees that the system satisfies functional objectives, operates dependably in a range of scenarios, and conforms to user expectations and legal requirements.

System testing aims to verify that each program operates as planned. Systems are tested as individual systems; they are not designed as complete systems. Testing of the system and units must be done by the analyst.

There are various kinds of testing techniques accessible. We have tested our system in a number of ways, such as if the application achieves the objectives for which it was created. Given that the program was intended to be used across a wide network, this was a crucial question that we had to answer.

5.3 Risk Management:



Figure 5.3: Risk Management

5.3.1 Risk Identification:

1. Technical risks:

-include problems with operating systems or browser connectivity, as well as integration with current HR systems.

2. Operational risks:

- include poor backup and recovery protocols and low acceptance rates due to inadequate user training.

3. Organizational Risks:

-Opposition from stakeholders to change and shifts in company objectives that impact project resources.

5.3.2 Risk Assessment:

1. Impact vs. Likelihood Matrix:

- Assessing the potential impact of identified risks on the project objectives against the likelihood of their occurrence.

2. Risk Severity Levels:

- Categorizing risks into low, medium, and high severity levels based on their potential impact on project success.

5.3.3 Steps Taken for Possible Risks:

1. Regular Risk Reviews:

- Conducting regular risk reviews throughout the project lifecycle to identify new risks and reassess existing ones

2. Interaction and Data:

- Keep a risk registry that lists any hazards that have been recognized, together with their evaluations, mitigation plans, and accountable parties.

5.4 Quality Management:

5.4.1 Rules Applied to Maintain Quality

1. Adherence to Standards:

- Strict adherence to industry standards, coding practices, and project management methodologies.

5.4.2 DSDM Atern Standard Quality Measures:

The Agile project delivery framework known as the Dynamic Systems Development Method (DSDM) Atern places a strong emphasis on producing high-quality products through iterative development and frequent stakeholder participation. A set of common quality standards is integrated into the DSDM Atern architecture to guarantee that the finished product satisfies the needs and expectations of all parties involved.

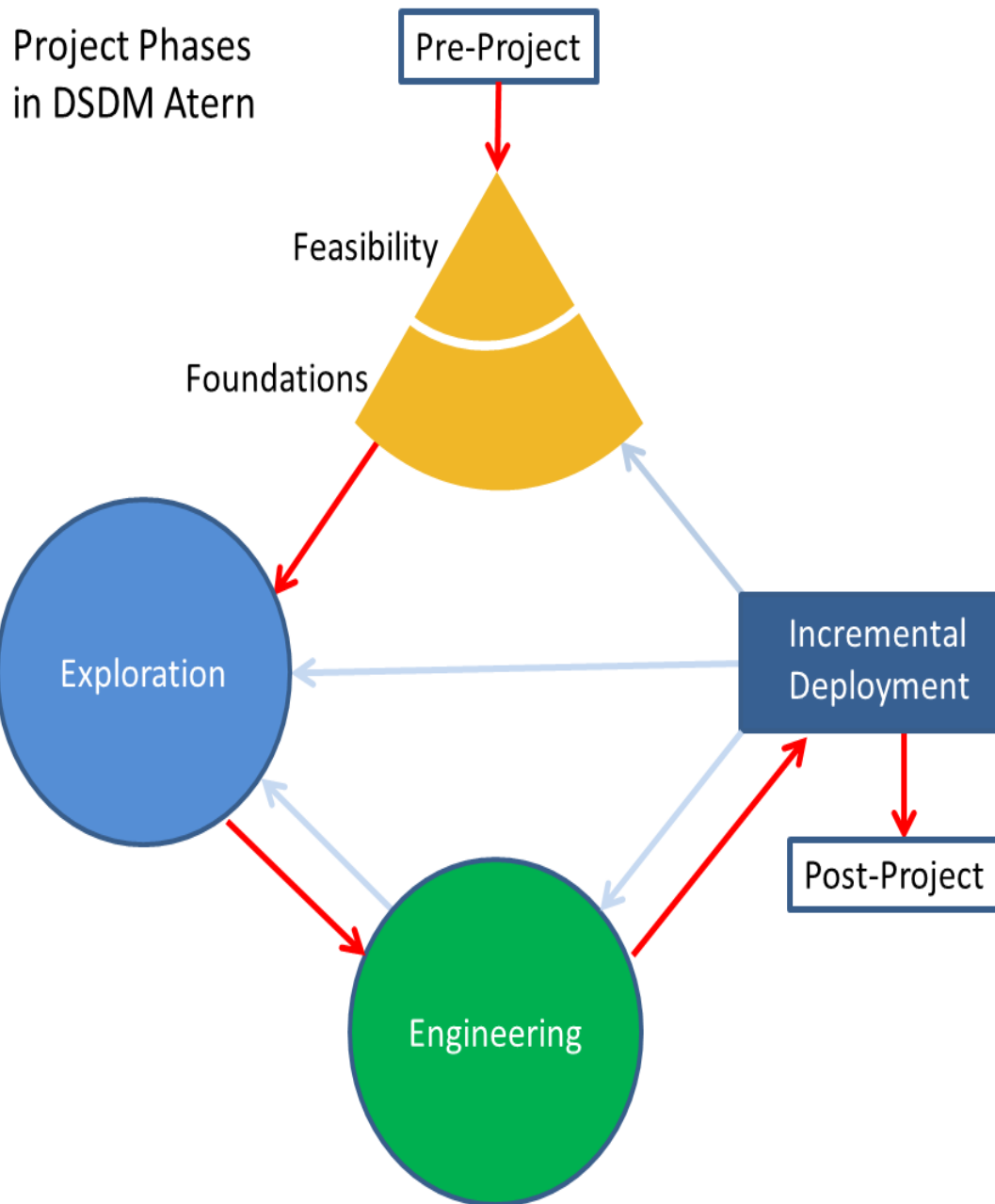


Figure 5.4: DSDM Atern Standard Quality

1. Suitable for Business Purpose:

-Verify that the solution satisfies the goals and objectives of the company and provides measurable business value. Engage stakeholders on a regular basis to ensure that the product fulfills their requirements and expectations.

5.4.3 Quality Plan and Measurement Metrics.:

1. Quality Metrics:

- To guarantee continual progress, define and monitor quality measures like as defect density, test coverage, and customer satisfaction. Embracing change, retaining high standards, and managing risks effectively are essential elements of developing an HRMS recruitment system. The project guarantees smooth progress by proactive risk identification, assessment, and mitigation. Accepting change within the Agile framework promotes adaptation and flexibility, which guarantees that the system is in line with changing corporate requirements. The implementation of continuous testing and standardization in quality management techniques serves as the foundation for a solid and dependable.

CHAPTER 6

FEASIBILITY

6.1 All Possible Types of Feasibility

The feasibility of software can be tested in two dimensions:

1. Practicality of Operation:

This phase raises questions about who will utilize the initiative and how it will operate. We must investigate the issues with the current system to determine whether they warrant fixing. The user's time and effort in analyzing employee data will be greatly reduced by this project, which will also offer more capabilities. It is therefore operationally possible.

2. Technical feasibility:

Evaluates if the project or solution can be carried out using the technical know-how and available technology. Assesses the necessary gear, software, and expertise. takes into account probable technical issues and their fixes.

6.2 Cost-Benefit Analysis (CBA):

Cost Analysis:

Leave Module:

- Submit Leave Request
- Review Leave Request
- Leave Balance Management
- Notification System
- Leave Policy Configuration
- Reporting and Analytics

1. Training Costs:

- Costs associated with training HR professionals and other users on the new HRMS software.

2. Integration Costs:

- Expenses related to integrating the HRMS software with existing systems and databases

Project Name: Leave Management System (LMS)

Total Cost:

Table 6.1: Cost Benefits

Item	Year 1	Year 2	Year 3	Total
Web Based Application	150000	0	0	150000
Domain & Hosting	20000	20000	20000	60000
Software	2500	2500	2500	7500
Internet	3500	3500	3500	10500
Training	7000	0	0	7000
Development	2000	2000	2000	6000
Maintenance	12000	12000	12000	36000
Total	197000	40000	40000	277000

6.3 DSDM – Is it suitable for this project? - PAQ

Positive Aspects:

1. Agile Approach:

DSDM facilitates incremental and iterative development, allowing teams to swiftly adjust to shifting priorities and objectives.

2. User Involvement:

Maintaining user involvement lowers the chance of delivering an unusable product by ensuring that the project remains in line with user expectations and demands.

3. Focus on Business Needs:

This approach places a strong emphasis on meeting stakeholder needs and creating goods that are valuable and pertinent to the business.

4. Incremental Delivery:

When product components are sent incrementally on a regular basis, problems can be identified early and feedback can be provided promptly, potentially producing higher-quality results.

Future Work:

1. Resource Intensive:

- DSDM requires active and continuous involvement from users and stakeholders, which can be resource-intensive for busy HR professionals.

2. Adaptation Challenges:

- If the organization is not accustomed to Agile practices, there might be challenges in adapting to the DSDM approach.

3. Change Management:

- DSDM welcomes changes, and managing change effectively is crucial. This may require a robust change management strategy.

6.4 Project Approach Questionnaire (PAQ):

1. Project Scope:

-Exist any features that are specifically left out of the project's scope?

2. Stakeholder Engagement:

- Have key stakeholders been identified, and is there a plan for continuous engagement throughout the project?

3. Resource Availability: including manpower available for successful project implementation?

4. Risk Assessment: - Have potential risks, such as data security and system integration challenges, been thoroughly assessed?

5. User Training: - Is there a well-defined plan for training HR professionals and other users on the new HRMS software?

6. Feedback Mechanism: - Is there a mechanism in place for gathering user feedback during and after the software development process?

The PAQ helps assess the readiness and viability of the project, considering various aspects such as scope, stakeholder engagement, resource availability, and risk management. The responses to these questions provide insights into potential challenges and areas that may require additional attention during the project lifecycle.

CHAPTER 7

FOUNDATION

7.1 Problem Area Identification:

- **Manual procedures:**

For the purpose of handling leave requests, approvals, and tracking, many organizations still use manual procedures. This frequently requires spreadsheets, emails, and paper forms, which can result in errors and delays. They may cause erroneous leave balances, postponed approvals, and misunderstandings between management and staff.

- **Limited Availability:**

Workers might not have simple access to their leave policies, leave balances, or the option to request time off from a distance. Insufficient accessibility has the potential to exacerbate employee annoyance, diminish openness, and cause delays in the leave administration procedure.

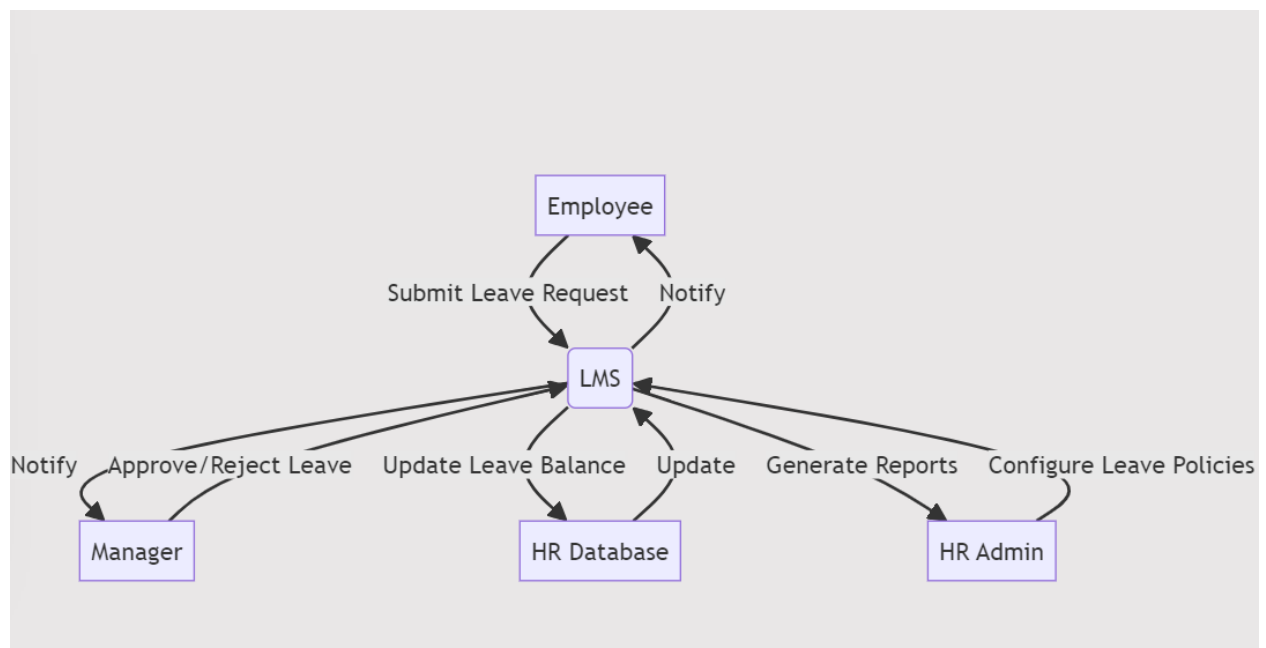


Figure 7.1: Simple Data Flow

7.2 Overall Requirement List

Here's a list of functional and non-functional requirements for an HRMS leave system.

Functional Requirements (what the system does):

1. Authentication and Verification of Users:

- It must be possible for users to log in with secure credentials.
- Different user roles (employee, manager, HR, admin) should be supported by the system with the relevant rights.

2. Leave Request Management:

- Workers must be able to request leaves of absence, detailing the dates and kinds of leaves they need.
- Leave requests must be visible to managers for approval or rejection.

3. Leave Balance Management:

- As soon as a leave request is approved, the system need immediately automatically update leave balances.
- It is imperative that workers have access to their present leave balances.

4. Leave Policy Configuration:

- Leave policies, including accrual rates, carryover regulations, and maximum leave limitations, must be configurable by HR.
- Employee leave balances must be modified per rules.

5. Notification System:

- Managers and staff should receive messages from the system about balance adjustments, approvals, denials, and requests for leaves of absence.
- Both email and in-app messaging are options for notifying users.

Non-Functional Requirements

1. Performance:

- Fast loading times and responsiveness
- Ability to handle large volumes of data
- High uptime and availability

2. Security:

- Protect sensitive candidate and employee data
- Comply with data privacy regulations
- Implement role-based access control,

3. Scalability

- As time goes on, the system will need to be able to handle more users and data.
- support for new leave categories and regulations as the company develops.

4. Usability:

- in order to reduce the requirement for training, the user interface must be simple to use and intuitive.
- Desktops, tablets, and smartphones should all be able to access the system.

7.3 Which Technology to be implemented

The specific requirements of the system, the development teams experience, the necessity for scalability, and compatibility with the current infrastructure all play a role in the technology selection for a Leave management system.

Java Spring Boot:

A framework built on Java that makes it easier to create apps that are suitable for production. It works great for creating reliable and scalable web apps. Convention over Configuration: Spring Boot reduces the requirement for developers to define specific configurations by adhering to the convention over.

- **Convention over Configuration:**

Spring Boot reduces the requirement for developers to define specific configurations by adhering to the convention over configuration paradigm. Because of its sensible defaults, developers may

concentrate more on business logic.

- **Embedded Web Server:**

With the help of Spring Boots embedded web server—often called Tomcat, Jetty, or Undertow—deploying apps is made simple and eliminates the need for external servers.

- **Auto-Configuration:** Based on the dependencies included in the project, Spring

Boot uses auto-configuration to automatically set up application components.

Developers will need to spend less time manually configuring systems as a result.

- **Micro services Support:** With built-in capability for generating standalone,

independently deployable services, it is ideal for designing micro services

architectures.

- **Spring Ecosystem Integration:** Because of Spring Boots smooth integration with the larger spring ecosystem, developers can use a variety of spring projects for messaging, security, data access, and other purposes.

- **Spring Boot Dev Tools:** A collection of tools called Dev Tools, which include live reloading, automated program restarts, and improved debugging assistance, are available to make development easier.

7.4 Recommendation and Justifications:

Justification:

A well-liked Java-based framework called Spring Boot makes it easier to create apps that are ready for production. It offers tools like embedded servers, auto-configuration, and a range of modules to help developers create applications that are effective and scalable

Spring Boot is a good option for Java development because of its large ecosystem and community support. I used MySQL to manage the database and the Java Spring Boot Framework to run the backend. Due to the fact that large libraries don't need to be loaded repeatedly, it becomes extremely safe as well as quick.

CHAPTER 8

EXPLORATION

8.1 Activity Diagram

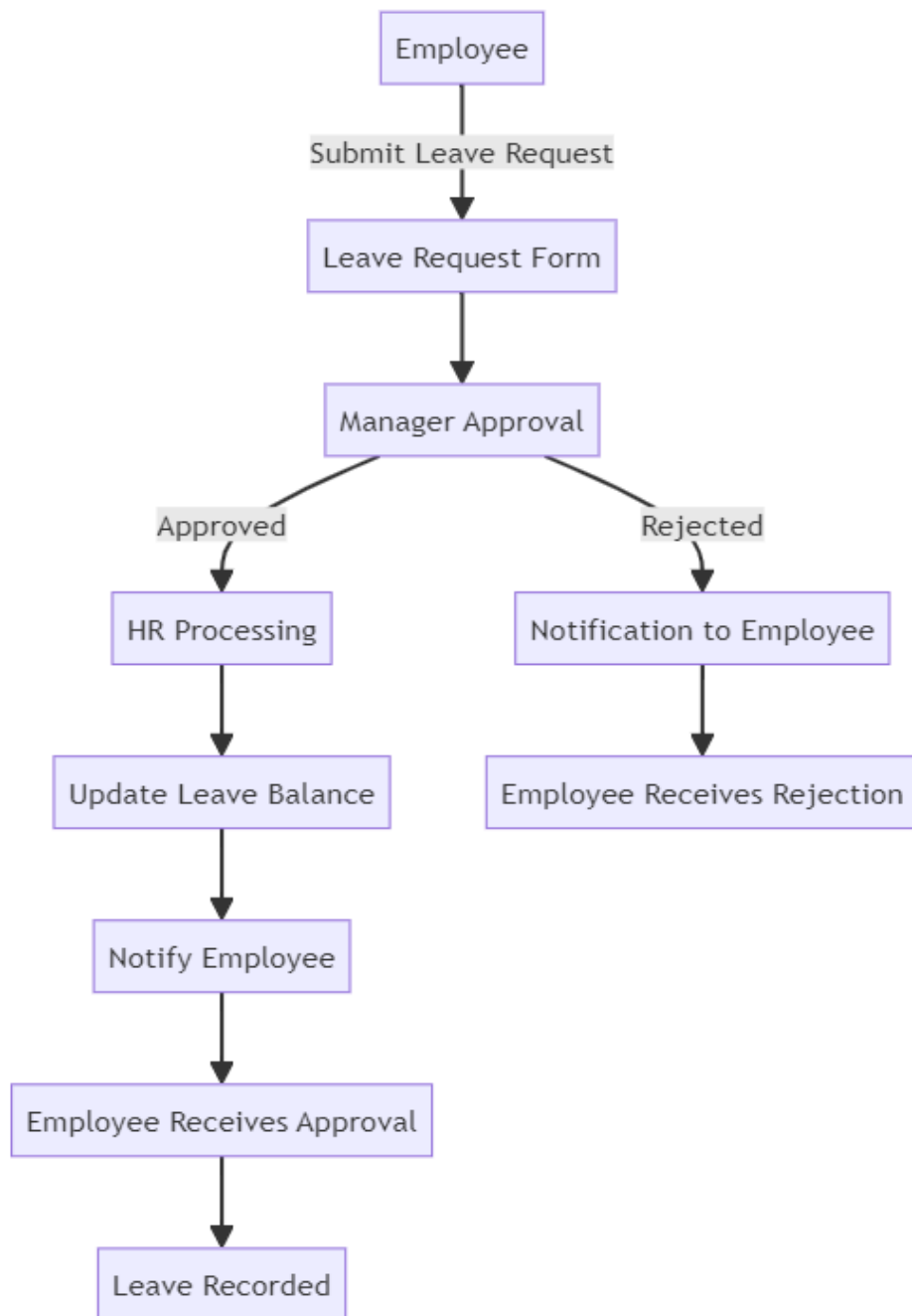


Figure 8.1 "Activity Diagram"

8.2 Full System Use Case

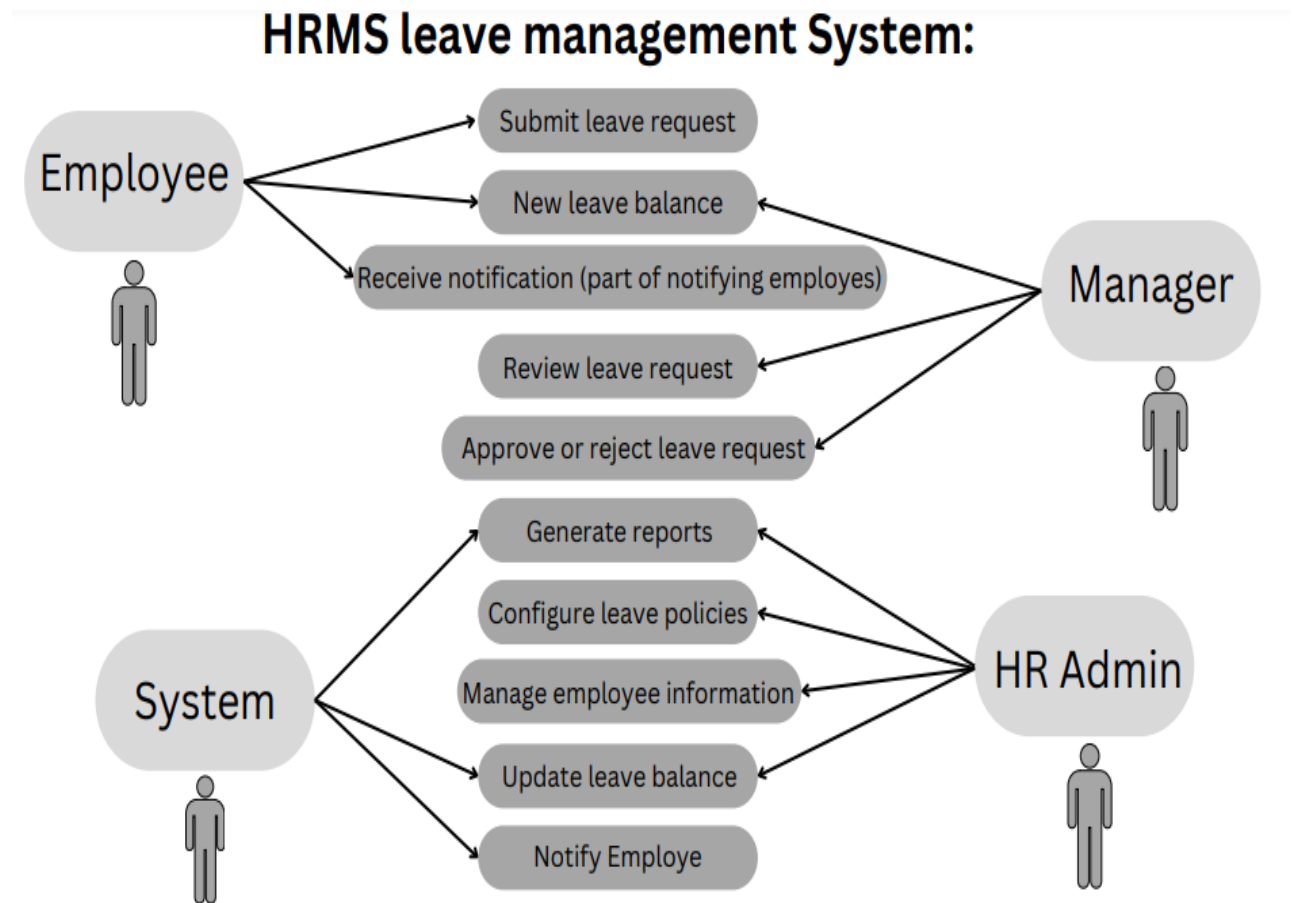


Figure 8.2: Full System Use Case Diagram

CHAPTER 9

ENGINEERING

9.1 Modules of the new system:

The Leave Management Service has a ton of new modules available. Here, I'll go over some of the core modules:

User Authentication and Access Control, Employee Self-Service, Leave Request Management, Leave Types and Policies, Leave Calendar, Manager Dashboard, HR Administration, Notifications and Alerts, Mobile Access, etc.

1. User Management Module:

- Employee Profiles:

Fill in both personal and professional information by creating and maintaining employee profiles.

- Role management:

Establish permissions in accordance with defined roles (such as employee, manager, and HR).

- Access control and secure login procedures comprise authentication and authorization.

2. Leave Request Module:

- Employees can use this interface to request leaves, indicating the kind, dates, and purpose of their request.
- Attached send: the ability to attach supporting documentation (such as health records).

3.Approval Workflow Module:

- Manager Dashboard: A viewable and editable interface for managers to accept or reject leave requests.

- Approval Hierarchy: Assistance with several tiers of approval procedures.
- Notification Mechanism: auto-reminders, approval, rejection, and application for leave alerts.

4. Mobile Access Module:

- Make sure the system is accessible and mobile-friendly across a range of devices by using responsive design.
- Optional mobile application designed to improve accessibility and user experience.

5. Administration Module:

- Network Setting: Module updates, policy management, and system setting configuration tools for administrators.
- Back and Restore: To avoid data loss, regularly do data backups and use recovery techniques.

9.2 Use Case:

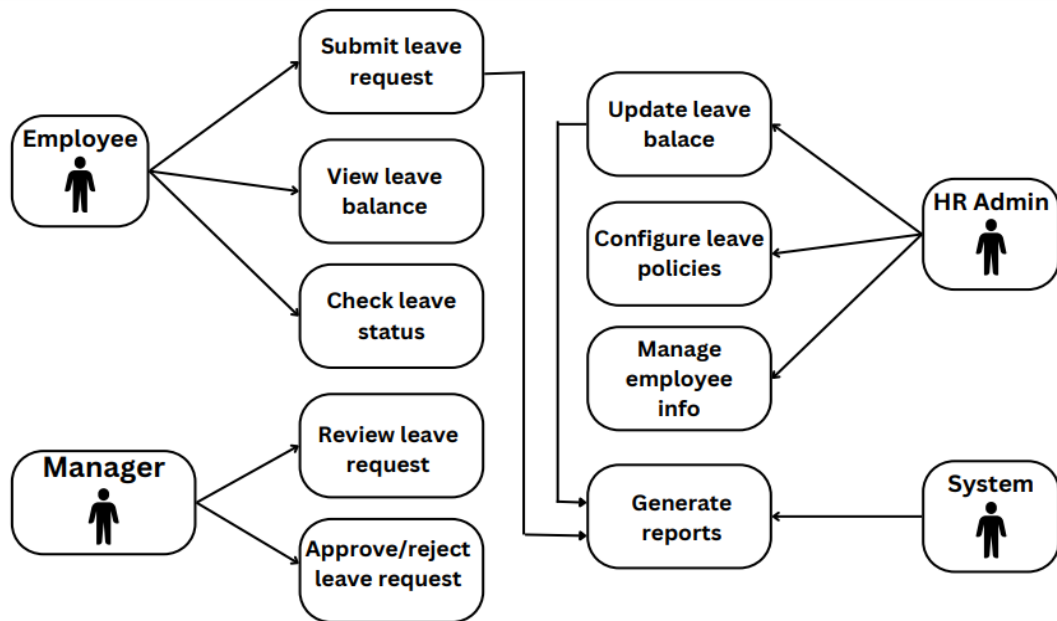


Figure 8.3: Use Case

9.3 Class Diagram:

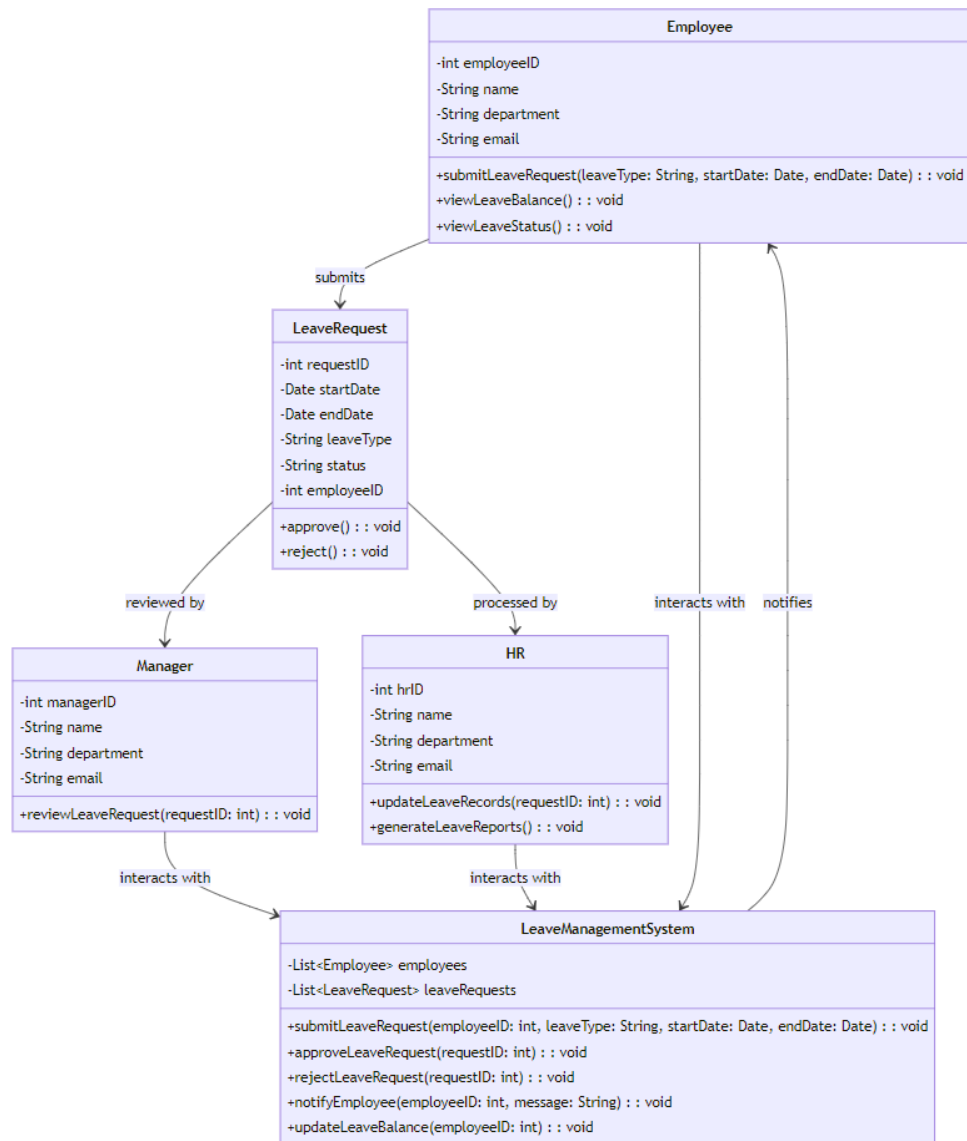


Figure 9.2: Class Diagram

9.4 Peter Chen EERD Diagram:

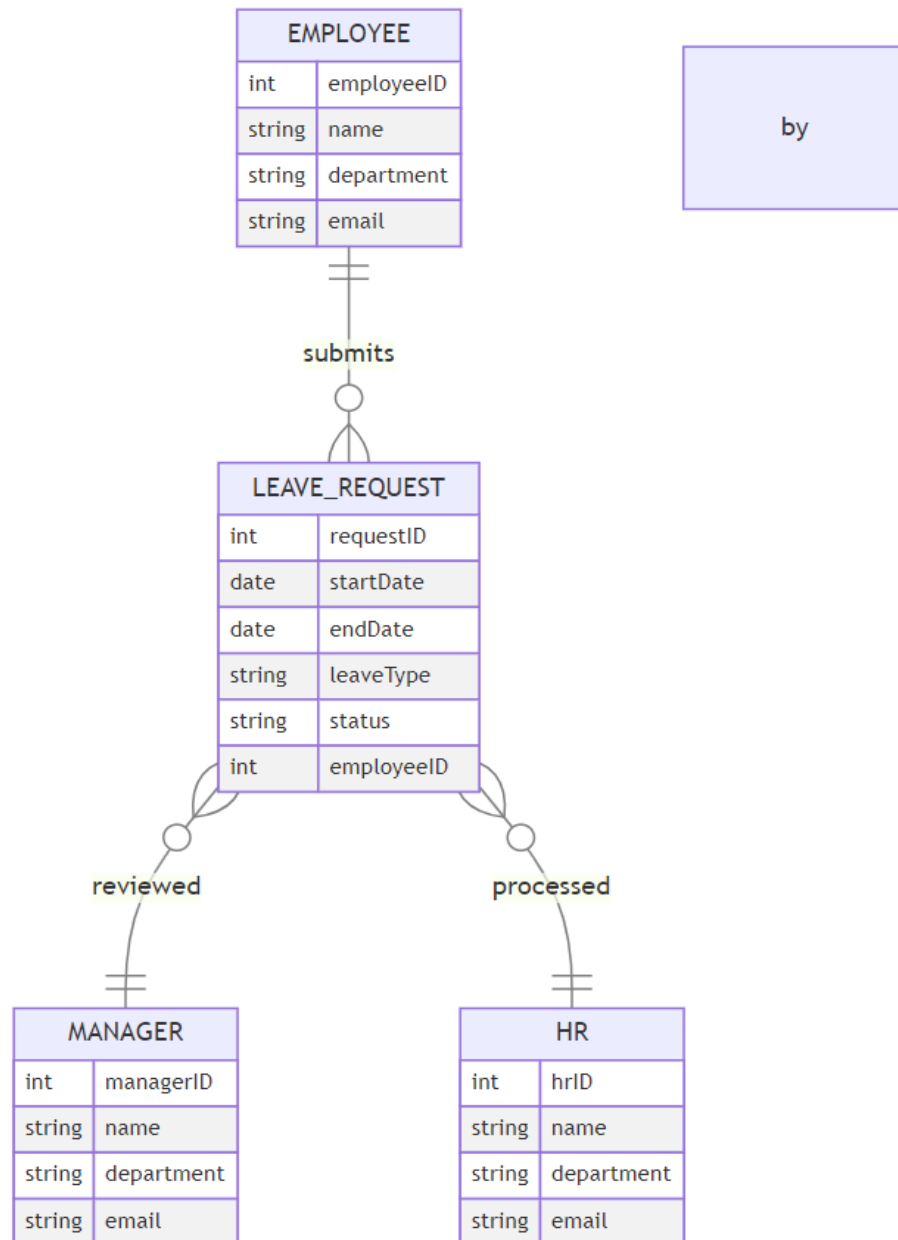


Figure 9.3: Peter Chen EERD Diagram

9.5 Sequence Diagram

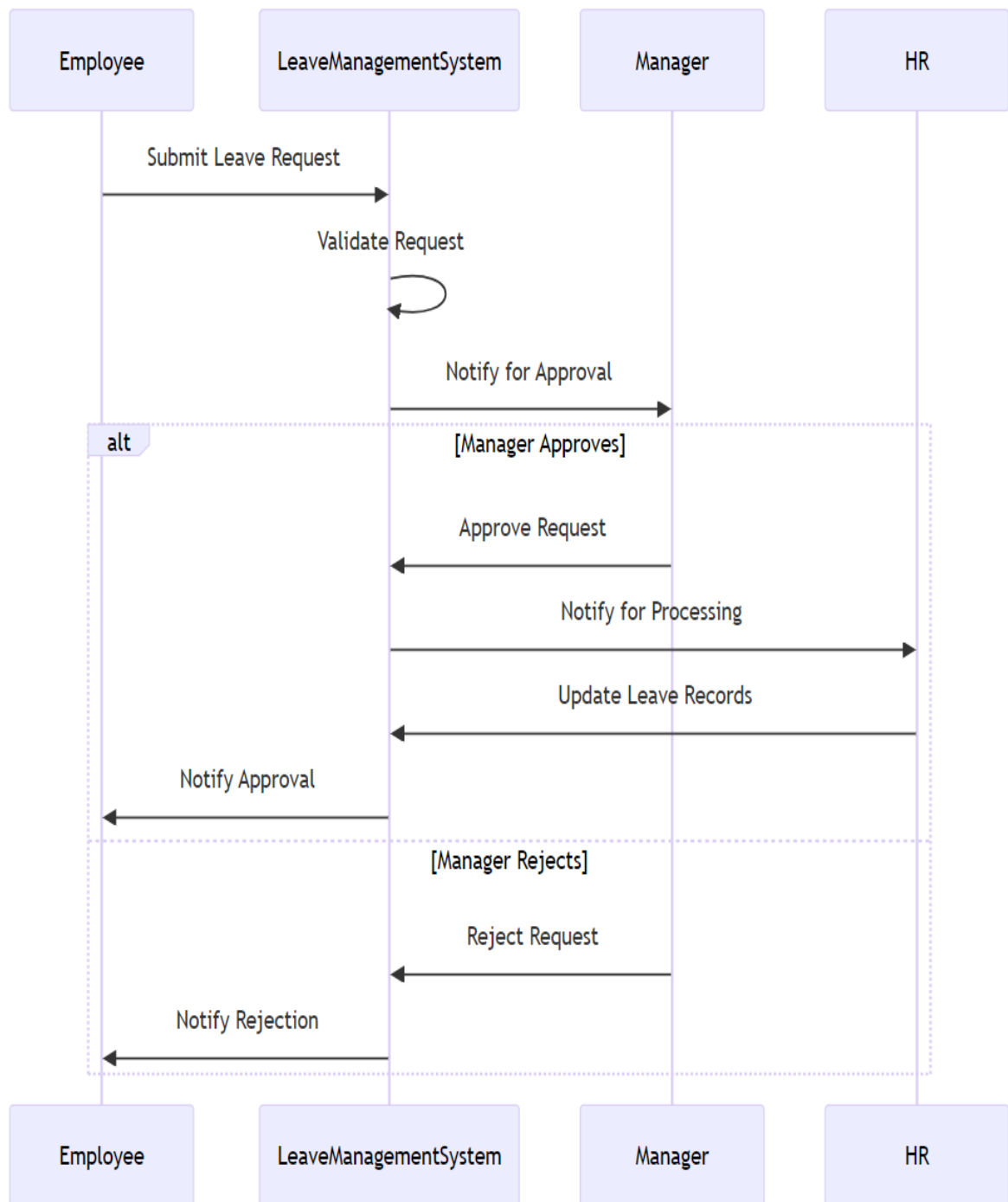


Figure 9.4: Sequence Diagram

9.6 Component Diagram

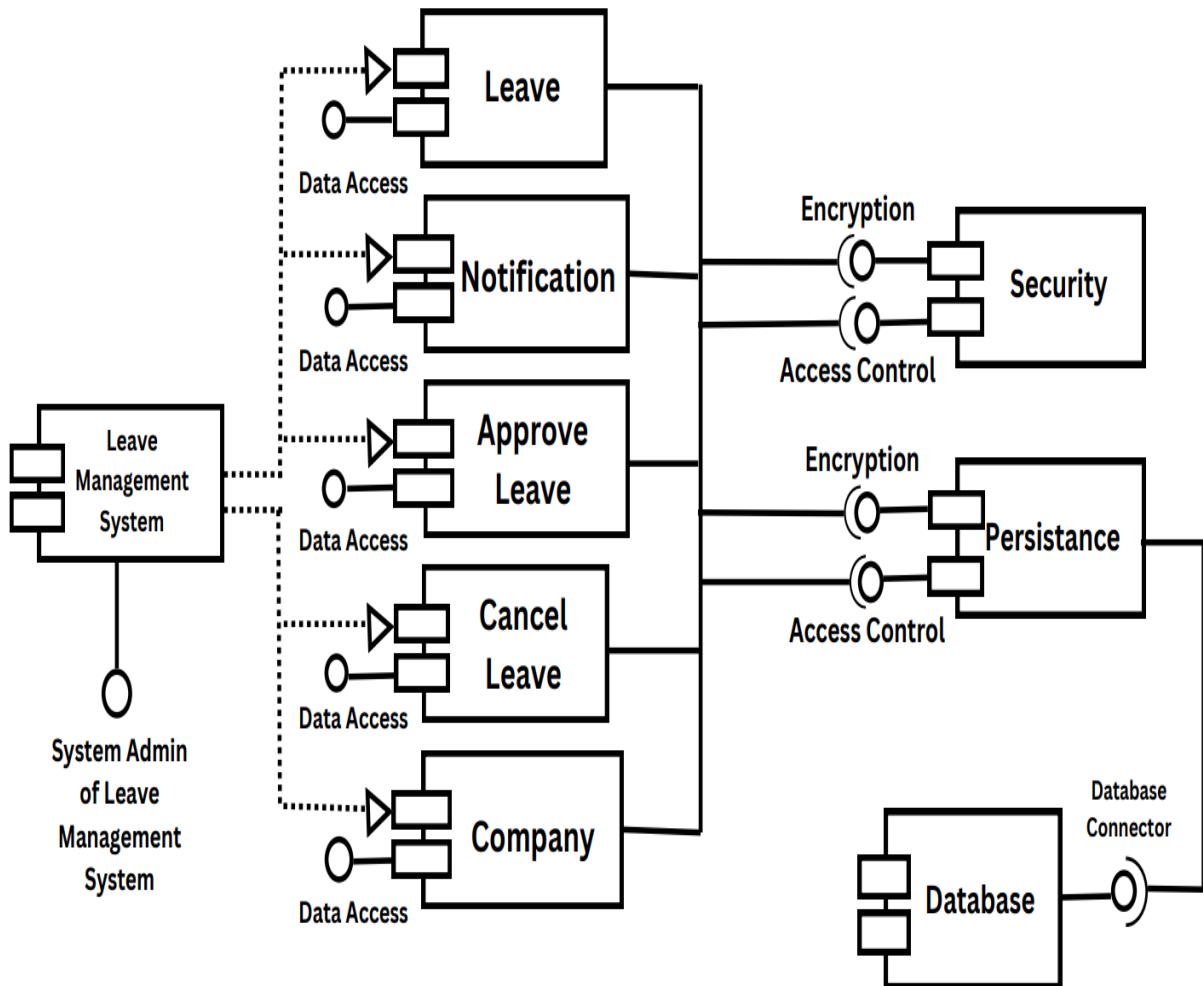


Figure 9.5: Component Diagram

9.7 Deployment Diagram

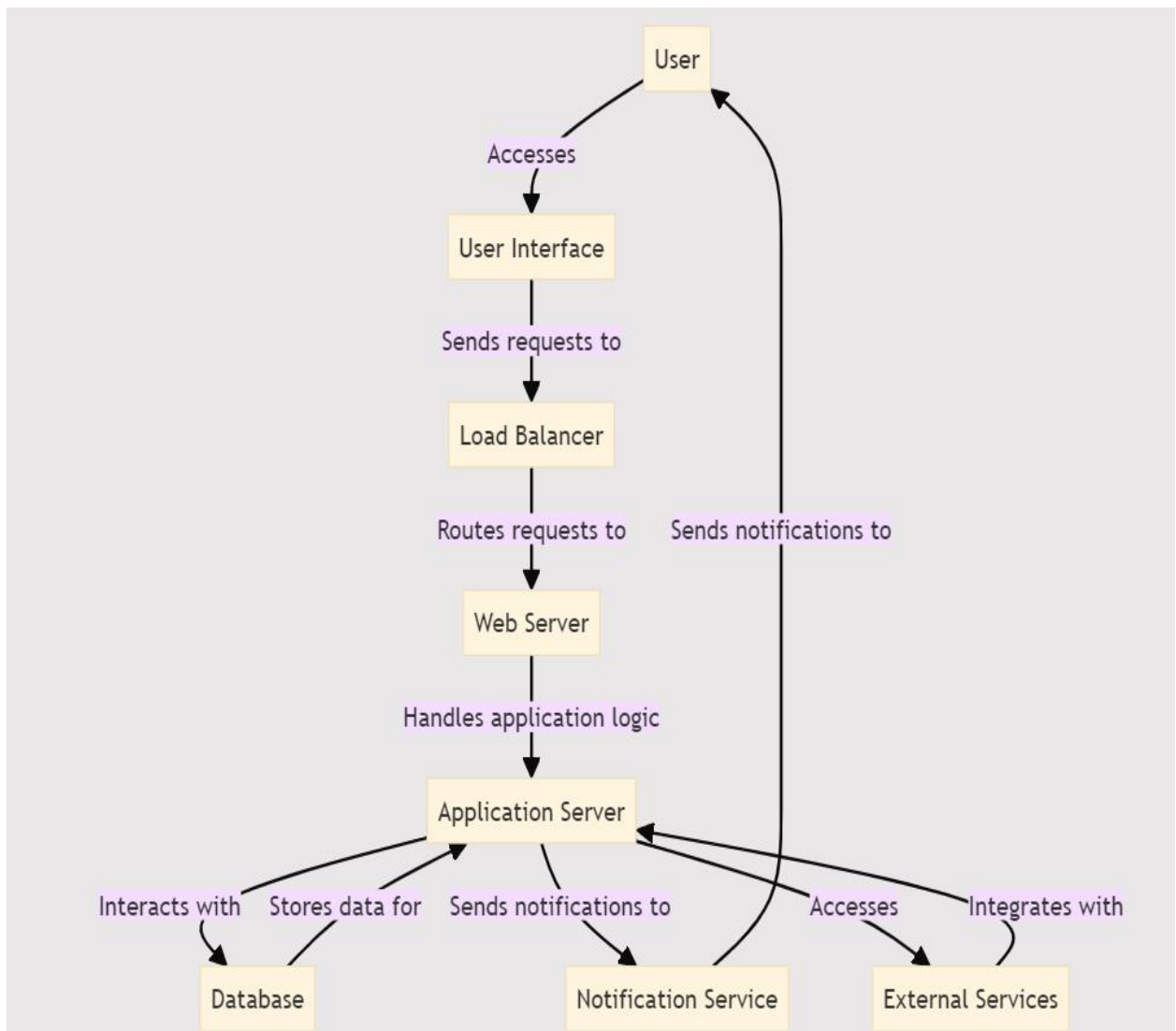


Figure 9.6: Deployment Diagram

CHAPTER 10

DEPLOYMENT / DEVELOPMENT

10.1 Coding:

I'm working on is fairly large, I'm concentrating on the Leave Management Module at this time. Requests for and approvals of employee leaves are handled by this module. The Leave Management Module i working on is a part of a larger HR management system. This module is crucial for managing employee leave requests, approval workflows, and tracking leave balances.

HRMS Project backend

```
1 package com.HRMSProjectBackend;
2
3 > import ...
4
5
6 @SpringBootTest
7 class HrmsProjectBackendApplicationTests {
8
9     @Test
10    void contextLoads() {
11    }
12
13 }
14
```

1. ManageSentLeaves (leave request)

Function: This procedure handles submitted leave requests.

```
package com.axelor.apps.hr.service.leave;

import com.axelor.apps.base.AxelorException;
import com.axelor.apps.base.db.repo.TraceBackRepository;
import com.axelor.apps.hr.db.Employee;
import com.axelor.apps.hr.db.LeaveLine;
import com.axelor.apps.hr.db.LeaveRequest;
import com.axelor.apps.hr.db.repo.LeaveRequestRepository;
import com.axelor.apps.hr.exception.HumanResourceExceptionMessage;
import com.axelor.i18n.I18n;
import com.google.inject.Inject;
import com.google.inject.persist.Transactional;

public class LeaveRequestManagementServiceImpl implements LeaveRequestManagementService {
    protected LeaveLineService leaveLineService;

    @Inject
    public LeaveRequestManagementServiceImpl(LeaveLineService leaveLineService) {
        this.leaveLineService = leaveLineService;
    }

    @Override
    @Transactional(rollbackOn = {Exception.class})
    public void manageSentLeaves(LeaveRequest leave) throws AxelorException {
        Employee employee = leave.getEmployee();

        LeaveLine leaveLine = leaveLineService.getLeaveLine(leave);

        if (leaveLine == null) {
            throw new AxelorException(
```

```

@Override
@Transactional(rollbackOn = {Exception.class})
public void manageSentLeaves(LeaveRequest leave) throws AxelorException {
    Employee employee = leave.getEmployee();

    LeaveLine leaveLine = leaveLineService.getLeaveLine(leave);

    if (leaveLine == null) {
        throw new AxelorException(
            leave,
            TraceBackRepository.CATEGORY_CONFIGURATION_ERROR,
            I18n.get(HumanResourceExceptionMessage.LEAVE_LINE),
            employee.getName(),
            leave.getLeaveReason().getName());
    }
}

```

2. ManageValidateLeaves(LeaveRequest leave)

Function: This procedure handles and verifies leave applications.

```

@Override
@Transactional(rollbackOn = {Exception.class})
public void manageValidateLeaves(LeaveRequest leave) throws AxelorException {
    Employee employee = leave.getEmployee();

    LeaveLine leaveLine = leaveLineService.getLeaveLine(leave);

    if (leaveLine == null) {
        throw new AxelorException(
            leave,
            TraceBackRepository.CATEGORY_CONFIGURATION_ERROR,
            I18n.get(HumanResourceExceptionMessage.LEAVE_LINE),
            employee.getName(),
            leave.getLeaveReason().getName());
    }

    if (leave.getInjectConsumeSelect() == LeaveRequestRepository.SELECT_CONSUME) {
        leaveLine.setQuantity(leaveLine.getQuantity().subtract(leave.getDuration()));
        if (leaveLine.getQuantity().signum() == -1 && !employee.getNegativeValueLeave()) {
            throw new AxelorException(
                leave,
                TraceBackRepository.CATEGORY_CONFIGURATION_ERROR,
                I18n.get(HumanResourceExceptionMessage.LEAVE_ALLOW_NEGATIVE_VALUE_EMPLOYEE),
                employee.getName());
        }
        if (leaveLine.getQuantity().signum() == -1 && !leave.getLeaveReason().getAllowNegativeValue()) {
            throw new AxelorException(
                leave,
                TraceBackRepository.CATEGORY_CONFIGURATION_ERROR,
                I18n.get(HumanResourceExceptionMessage.LEAVE_ALLOW_NEGATIVE_VALUE_REASON),
                leave.getLeaveReason().getName());
        }
        leaveLine.setDaysValidated(leaveLine.getDaysValidated().add(leave.getDuration()));
    } else {
        leaveLine.setQuantity(leaveLine.getQuantity().add(leave.getDuration()));
    }
}

```

3. ManageRefuseLeaves (leave request)

```
@Override
@Transactional(rollbackOn = {Exception.class})
public void manageRefuseLeaves(LeaveRequest leave) throws AxelorException {
    Employee employee = leave.getEmployee();

    LeaveLine leaveLine = leaveLineService.getLeaveLine(leave);

    if (leaveLine == null) {
        throw new AxelorException(
            leave,
            TraceBackRepository.CATEGORY_CONFIGURATION_ERROR,
            I18n.get(HumanResourceExceptionMessage.LEAVE_LINE),
            employee.getName(),
            leave.getLeaveReason().getName());
    }
}
```

Function: This procedure handles leave requests and denies them.

4. ManageCancelLeaves(LeaveRequest leave):

Function: This procedure handles leave requests and cancels them. status and type of the leave request.

```
@Override
@Transactional(rollbackOn = {Exception.class})
public void manageCancelLeaves(LeaveRequest leave) throws AxelorException {
    Employee employee = leave.getEmployee();

    LeaveLine leaveLine = leaveLineService.getLeaveLine(leave);

    if (leaveLine == null) {
        throw new AxelorException(
            leave,
            TraceBackRepository.CATEGORY_CONFIGURATION_ERROR,
            I18n.get(HumanResourceExceptionMessage.LEAVE_LINE),
            employee.getName(),
            leave.getLeaveReason().getName());
    }

    if (leave.getStatusSelect() == LeaveRequestRepository.STATUS_VALIDATED) {
        if (leave.getInjectConsumeSelect() == LeaveRequestRepository.SELECT_CONSUME) {
            leaveLine.setQuantity(leaveLine.getQuantity().add(leave.getDuration()));
        } else {
            leaveLine.setQuantity(leaveLine.getQuantity().subtract(leave.getDuration()));
        }
        leaveLine.setDaysValidated(leaveLine.getDaysValidated().subtract(leave.getDuration()));
    }
}
```

10.2 Possible Problem Break Down:

The task will be finished effectively and skillfully. Consequently, the following list of potential problems is provided.

Certainly! Let's break down potential problems and prioritize them during the development of the HRMS Leave System:

1. Integration Challenges:

- Problem Breakdown
- Difficulty integrating the HRMS Leave Management software with existing HR or organizational systems.

2. User Adoption Resistance:

- Problem Breakdown
- Resistance or reluctance from HR professionals to adopt the new system.

3. Data Security Risks:

- Problem Breakdown
- Identifying vulnerabilities and ensuring the secure storage of sensitive HR data

4. Performance Bottlenecks:

- Problem Breakdown:
- Identifying and resolving potential performance issues, especially during peak usage.

5. Incomplete Feature Set:

- Problem Breakdown:
- Some critical features identified during requirements gathering are missing or incomplete

6. Scalability Concerns:

- Problem Breakdown:
- Ensuring the system can scale effectively with a growing number of users and data.

7. Training Program Gaps:

- Problem Breakdown:
- Inadequate training programs for HR professionals and other users.

8. User Feedback Collection:

- Problem Breakdown:
- Lack of a structured mechanism for gathering and incorporating user feedback during development.

9. Technological Obsolescence:

- Problem Breakdown:
- The chosen technology stack may become obsolete or unsupported.

To tackle these possible issues, careful planning, efficient project management, and an emphasis on user-centric design are necessary. Prioritizing these problems during the development of the HRMS Recruitment System ensures that critical issues are addressed promptly, reducing the risk of project delays or failure. Regular reassessment and adaptability to changing circumstances are crucial throughout the development lifecycle.

10.3 Prioritization while developing

This table (MoSCoW, Eisenhower Matrix, Value vs. Effort Matrix) offers an organized method of prioritizing utilizing several frameworks. These criteria can be used to efficiently prioritize activities and features while developing your leave management system.

Table 10.1: Prioritization Requirement

Priority Criteria	Examples of Leave Management System (LMS)
MOSCOW Method	
Must-Have	Basic leave request submission, approval workflows
Should-Have	Policy configuration, team calendar integration
Could-Have	Advanced reporting, mobile app support
Won't-Have	Social media integration, gamification features
Eisenhower Matrix	
Urgent and Important	Security updates, compliance with labor laws
Important, Not Urgent	Enhanced reporting, user interface refinements
Urgent, Not Important	UI aesthetic updates, minor feature enhancements
Not Urgent, Not Important	Experimental features, non-critical UI tweaks
Value vs. Effort Matrix	
High Value, Low Effort	Quick wins like leave balance visibility
High Value, High Effort	Complex approval workflows, deep integrations
Low Value, Low Effort	Minor UI improvements, email notifications
Low Value, High Effort	Advanced analytics features, complex policy configurations

CHAPTER 11

Testing Plan

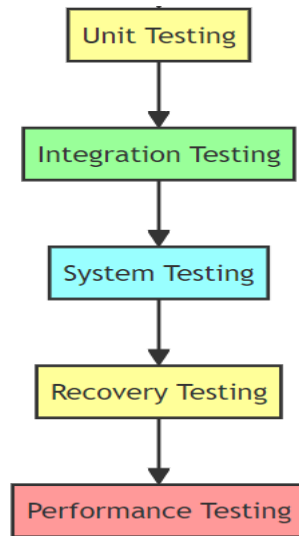


Figure 11.1: Testing Plan

Black-box testing:

Also known as functional testing. Software testing techniques where by the internal working of the item being tested are not known by the tester. For example, in a black box test on software design, the tester only knows the inputs and what the expected outcomes should be not how the program arrives at those outputs. The tester does not ever examine the programming code and does not need any further knowledge of the program other than its specification.

White Box Testing:

Also known as glass box, structural, clear box and open box testing. A software testing technique where by explicit knowledge of the internal workings of the item being tested are used to select the test data. Unlike black box testing, white box testing uses specific knowledge of programming code to examine outputs. The test is accurate only if the tester knows what the program is supposed to do.

Test cases completed: 75

Test cases that pass: 75

Test cases are unsuccessful: 0

Pass: 100%

Failure: 0%

Note: Developed based on user requirements and successful testing phase.

Defect

One of the most important items in the test report is the defect. The report ought to contain the following data.

Total number of bugs: 10 (open, closed, and resolved).

Number of bugs:

- Open: 10
- Resolved: 10
- Closed: 10

Sorting by importance and severity: resolve

Test Report- (4)

	Test	
--	-------------	--

Test Cycle

System test

EXECUTED	PASSED	50
	FAILED	0
Pending		0
In Progress		0
Blocked		0
(Sub-Total) Test Passed		0
Pending + In Progress+ Blocked + Test Executed		50

Table 11.1: Final Test Report

CHAPTER 12

IMPLEMENTATION

12.1 Coding Scenario:

We utilized the API Call GraphQL query language and Python programming. REST is not used in favor of GraphQL.

All of the functionality has been modularized so that we may use it as needed. We have also accessed employee attachments through the Google Drive API. Those documents are saved, and only the administrator has access to them.

Our inability to succeed when 500 employees utilized the HRMS at once was a significant obstacle. For load balancing, we have employed Nginx. In terms of https support, our HRMS software is likewise secure.

12.2 Load Balancing:

Load balancing involves optimizing a system to handle its users' impact. The number of users accessing the system at any given time and its duration of use are factors that contribute to the system's load balance and load equalization..

CHAPTER 13

CRITICAL APPRAISAL AND EVALUATION - HRMS

13.1 Objectives that could be met:

There is no such thing as a flawless system. Updates are always possible. I've tried to complete all necessary tasks so that a university may operate this system and use it efficiently. Nonetheless, some conditions might be met.

13.2 Future Work:

1. Comprehensive Reporting and Analytics:

- Why Not Touched: Focused on core functionalities during the initial release.
- What Could Have Been Done: Allocate additional resources for advanced analytics implementation.

2. Including software and documentation:

- Software: - Success Rate: Successfully deployed the HRMS recruitment system.
- How Much Better: Continuous monitoring and improvement for system performance and security.
- Why Not Done: Limited post-deployment monitoring due to resource constraints.
- Improvement: Allocate resources for ongoing system maintenance and improvement.
- Documentation:
- Success Rate: Comprehensive documentation for system users and administrators.
- How Much Better: Regular updates to documentation based on user feedback and system enhancements.

- Why Not Done: Limited resources for continuous documentation updates.
- Improvement: Establish a documentation update schedule and assign responsible individuals for regular reviews and revisions.

13.2.1 Why it could not be touched

For this segment, I did some field research. If we wish to develop online transactions, we need to do work linked to APIs. This cannot be developed in the timeframe and money provided.

CHAPTER 14

LESSONS LEARNED

14.1 What Have I Learned:

This leave Management system, a web-based program. As a trainee DCL intern, I provided backend development. I learned a number of things from working on this project that I will find very helpful going forward. My initial experience with software architecture was with third-party designs. I learned a few new, undiscovered design. During the project, I researched and integrated methods for using templates to create designs and improve usability. I am now prepared for the system development following the completion of the design phase..

I am ready for the system to be developed after the design is complete. Any system must first have its database designed before it can be constructed. If a suitable database architecture is not put into practice, The importance of a comprehensive project closure process, including knowledge transfer, post-implementation reviews, and documentation, was underlined during the closing phase. There was an extra benefit in recognizing accomplishments and holding an open project closure meeting with stakeholders.

Nonetheless, this system has given me more experience with a sizable database and multiple data tables.

I helped install the system on the working server when the project was completed. I had never installed a system on a shared server before now. Working on this project has been a fantastic experience, and I have learned a lot.

14.2 What Problems I Faced:

I worked as a developer as well as a designer on this Leave System project. Over the course of the project, understanding the architecture takes a lot of time and complexity. I have to confront and resolve issues iteratively. Stated differently, issues that crop up in one time box get fixed in this other time box and task. I've already begun creating the system, but I was told to stick to the templates. However, the reason it took so long to finish was that I had no knowledge of how to make the templates for the design. I also needed to learn new things about. It became obvious that I would need to establish a large database with multiple data tables as soon as I started working. But I had never dealt with a database this huge before.

- Resource limits:

Throughout the development process, there were manpower and time limits that presented difficulties. This had an effect on the scope and depth of features that were first intended to be included.

- Modification Complexity:

Changing the system to satisfy particular business needs without incurring undue expenses or delays.

- User Utilization and Training:

Overcoming aversion to change and making sure users are properly instructed to operate the new system.

-Adaptation to Changing Requirements:

Regular modifications to system requirements were necessary due to the dynamic nature of recruitment processes. Although adaptability was a plus, it also made it harder to keep the project scope steady.

14.3 What Solutions Occurred:

There is a fix for every problem that can be applied to fix it. I have completed the system and come up with a solution for every issue I mentioned earlier. First and foremost, I developed a way to use software architecture that was already produced by others. During the design

process, I discovered some new and surprising basic ideas.. A database needs to be constructed before any systems can be developed.

- **Thorough Documentation:** To promote knowledge transfer, thorough documentation—including system manuals and training materials was ensured.
- **Transparent Communication:** resolved expectations with stakeholders in a transparent manner, highlighting project milestones that were reached.

14.4 Overall Reflection:

Upon contemplating the complete project lifetime, I acknowledge that project management is inherently iterative. The success of the project was greatly influenced by the team's capacity for adaptation, learning from setbacks, and pro-actively implementing fixes. Future initiatives will always be built on the foundation of continuous improvement, which makes sure that every project serves as a springboard for improving project management techniques and producing better result.

CHAPTER 15

CONCLUSION

HRMS Leave Management System Developments

15.1 Summary of the project

The development and implementation of the HRMS recruitment system mark a significant milestone in advancing our organizational capabilities in human resource management. This conclusion reflects on key achievements, challenges faced, and the broader impact on HR processes and organizational efficiency.

Employee leave requests are handled effectively via the Leave Management System. Employees can apply for leave through the system, while leaders can accept, deny, or cancel requests for leave. Ensuring that every procedure linked to leaves is automated, it lessens the workload and streamlines operations

15.2 Goals of the project

Since I took part in a DCL internship program, my organization has determined the goal of my project. In the software development industry, a systems goal is established by the needs of the customer.

The main goal of this project is to streamline manual processes and reduce labor costs.

1. Automate the management of leave:

- Make the procedure for submitting, reviewing, denying, and withdrawing leave requests as simple as possible.
- Using automation, lessen the administrative load on managers and HR departments.

2. Interface That's Easy to Use:

- Provide an interface that is simple to use and intuitive for managers and staff alike.

- Make sure users only need a minimal amount of training to navigate and operate the system efficiently.

3. Detailed Testing:

- Carry out comprehensive security, performance, integration, and unit testing.
- Make sure the system is rigorously tested to verify its dependability and compliance with performance standards.

15.3 What I have accomplished in the documentation (stages/activities/plans): I have included a thorough account of all I accomplished for the project in this documentation. I began by writing the system introduction in the documentation, after which I started this project & basic investigation and progressively finished all the procedures required for this system. This documentation was created using Microsoft Word for all tables, and a third-party tool was used to create any figures or diagrams.

15.4 Value of the project

Work that is done entirely by hand is more difficult. If all of the job can be entirely automated, time and hassle might be saved. The same amount of labor can be done more quickly and with less effort when technology is used. It takes a lot of work and time to manually maintain all of an organization's client information.

15.5 My experience

Developing software and writing code can be rewarding and difficult at the same time. There are times when things come together and I feel accomplished, but I also run across unforeseen difficulties that call for problem-solving abilities. Creating software for Leave Management System is a complex process that calls for innovative thinking, technological difficulties, and a never-ending learning curve. It's a journey that blends the necessity for adaptability in the face of changing requirements and technologies with the gratification of making a positive contribution to a crucial process.

APPENDIX:

Applicant: A person who applies for HRMS leave management system

LMS: Leave Management System

Actor: A person or system that interacts with the Leave Management system

Use Case: A description of how a system will be used by its actors (users).

Oracle

<https://www.oracle.com/human-capital-management/hrms/>

Spring Boot

<https://spring.io/projects/spring-boot/>

MySQL

<https://www.mysql.com/>

REFERENCE:

1. Risk Management. (2007). Retrieved from <https://www.pacs.org.uk/risk-management#:~:text=%EF%BB%BFManagement&text=PACS%20IWMS%20provides%20a%20fully,manage%20performance%20and%20measure%20compliance>.
2. https://en.wikipedia.org/wiki/Dynamic_systems_development_method
3. Rungta, K. (2008). Test Report. Retrieved from Guru99: <https://www.guru99.com/how-test-reports-predict-the-success-of-your-testing-project.html>
4. <https://code-projects.org/employee-leave-management-project-report-in-php-css-js-and-mysql-free-download/>

Plagiarism report:

201-16-498

ORIGINALITY REPORT

10%

SIMILARITY INDEX

12%

INTERNET SOURCES

1%

PUBLICATIONS

7%

STUDENT PAPERS

PRIMARY SOURCES

1

dspace.daffodilvarsity.edu.bd:8080

Internet Source

3%

2

Submitted to Daffodil International University

Student Paper

3%

3

Submitted to University of Northumbria at
Newcastle

Student Paper

2%

4

m.mu.edu.sa

Internet Source

1%

5

Submitted to University of Greenwich

Student Paper

1%

Exclude quotes On

Exclude matches < 1%

Exclude bibliography Off