



# **ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY**

## **SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTING**

### **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

#### **BSc. SENIOR PROJECT**

#### **Resource Management System for Ethiopian Educational Materials Production and Distribution Enterprise (EMPDE)**

##### **Group Members:**

- |                             |                    |
|-----------------------------|--------------------|
| <b>1. ABREHAM BEKELE</b>    | <b>A/UR4499/09</b> |
| <b>2. AREFAT HYEREDIN</b>   | <b>A/UR5082/09</b> |
| <b>3. BINIYAM GOSSAYE</b>   | <b>A/UR4861/09</b> |
| <b>4. GIRUM GETACHEW</b>    | <b>A/UR4045/09</b> |
| <b>5. GIZEALEW ENDESHAW</b> | <b>A/UR3981/09</b> |

**Advisor: Mr. Shumet F.**

**Adama, Ethiopia**

**August 2021**

## **Approval Page**

Name	ID Number	Signature
1. Abreham Bekele	A/UR4499/09	_____
2. Arefat Hyeredin	A/UR5082/09	_____
3. Biniyam Gossaye	A/UR4861/09	_____
4. Girum Getachew	A/UR4045/09	_____
5. Gizealew Endeshaw	A/UR3981/09	_____

Approved by

Advisor: Mr. Shumet F.

Signature: \_\_\_\_\_

Adama Science and Technology University

Adama, Ethiopia

May 12, 2021

## **Acknowledgment**

The Project Team would like to express their gratitude to Adama Science and Technology University (ASTU) and Ethiopian Educational Materials Production and Distribution Enterprise (EMPDE) for their complete support and efforts that allowed us to work on this project.

We also would like to thanks the Computer Science and Engineering department at ASTU along with the project coordinators for all the support and assistance they have given us.

The team owes a great deal of gratitude to the Project Advisor Mr. Shumet F. for all the useful comments, encouragement, and guidance throughout the project.

Furthermore, we would like to thank and appreciate all parties that have assisted us along the way for their time regarding interviews and feedbacks.

Our thanks and appreciations also go to our families and friends who have supported us in every way they can.

## **Executive Summary**

Our country effortfully trying to build a strong economy has created a competitive market among organizations and businesses of all scales. Recent history and experience show that technology plays a great role in achieving a sustainable national economy. For enterprises to join this path of industriousness, they should acclimate to the fact that digital systems are key to this bright future. Most organizations in Ethiopia do not consider computerizing their business process let alone integrating different sectors of their operation digitally.

Organizations in Ethiopia still under the thumbs of primitive ways have created various problems like performance issues, minimal collaboration, and poor data handling. The government has strategized ICT development as one of its top objectives. However, there is still a lot of work to be done in computerizing and digitizing the fundamental governmental organizations as well as public and private enterprises to bring about a systematic and modern change to the daily business processes of these organizations.

The proposed resource management system for the Ethiopian Educational Materials Production and Distribution Enterprise (EMPDE) plans to alleviate the stated problems by providing an enterprise human resources and finance system. The system would serve the organization by facilitating and integrating various business operations related to human resources and accounting, also creating a better data infrastructure for the enterprise. Moreover, the proposed system plans to account for the existing business practices to ease the transition to a computerized system.

## Table of Contents

<b>Approval Page .....</b>	<b>i</b>
<b>Acknowledgment.....</b>	<b>ii</b>
<b>Executive Summary .....</b>	<b>iii</b>
Table of Contents .....	iv
List of Figures .....	ix
List of Tables.....	xi
Abbreviations .....	xii
<b>1. Chapter One.....</b>	<b>1</b>
1.1. Introduction .....	1
1.2. Background of the Project.....	2
1.3. Background of the Organization .....	4
1.3.1. Mission of the Organization.....	5
1.3.2. Vision of the Organization.....	5
1.3.3. Values of the Organization .....	5
1.3.4. Objectives of the Organization .....	5
1.4. Statement of the Problem .....	6
1.5. Justification of the Project.....	7
1.6. Objective of the Project.....	7
1.6.1. General Objective .....	7
1.6.2. Specific Objective .....	7
1.7. Scope and Limitation .....	8
1.7.1. Scope.....	8
1.7.2. Limitation.....	9
1.8. Feasibility Study.....	9
1.8.1. Technical Feasibility .....	10
1.8.2. Operational Feasibility.....	10
1.8.3. Economic Feasibility .....	10
1.9. Significance of the Project .....	11
1.10. Beneficiaries of the Project.....	11
1.11. Methodology.....	12

1.11.1.	Data Collection Methodology .....	12
1.11.2.	Design Methodology .....	13
1.11.3.	Development Methodology.....	13
1.12.	Development Tools.....	14
1.13.	Testing Plan .....	15
1.14.	Task and Schedule .....	15
1.15.	Cost Analysis.....	16
1.16.	Team Composition .....	16
<b>2.</b>	<b>Chapter Two .....</b>	<b>17</b>
2.1.	Description of Existing System.....	17
2.2.	Major Functions of Existing System.....	18
2.3.	Users of the Current System.....	18
2.4.	Drawbacks of the Current System.....	19
2.5.	Business Model Rule.....	19
2.6.	Review of Related Systems.....	20
<b>3.</b>	<b>Chapter Three: The Proposed System.....</b>	<b>24</b>
3.1.	Overview .....	24
3.2.	Functional Requirement .....	25
3.3.	Non-functional Requirement.....	26
3.3.1.	Design Requirements (DR).....	26
3.3.2.	Usability Requirements (UR) .....	27
3.3.3.	Security Requirements (SR) .....	27
3.3.4.	Reliability Requirements (RR) .....	27
3.3.5.	Performance Requirements (PR) .....	28
3.4.	System Model.....	28
3.4.1.	Scenarios .....	29
3.4.2.	Use case Model .....	35
3.4.2.1.	Use case Diagram.....	38
3.4.2.2.	Use case Description .....	40
3.5.	Object Model.....	69
3.5.1.	Class Diagram.....	69
3.5.2.	Data Dictionary .....	71

3.6.	Dynamic Model.....	72
3.6.1.	Sequence Diagram .....	72
3.6.2.	Activity Diagram .....	77
3.6.3.	State Chart Diagram.....	82
<b>4.</b>	<b>Chapter Four – System Design.....</b>	<b>85</b>
4.1.	Overview of System Design.....	85
4.1.1.	Purpose of the System Design .....	85
4.1.2.	Design Goal .....	85
4.1.2.1.	Performance .....	85
4.1.2.2.	Dependability .....	86
4.1.2.3.	Maintenance .....	87
4.1.2.4.	End-User.....	87
4.2.	Proposed System Architecture .....	88
4.2.1.	System Process.....	90
4.2.2.	Subsystem Decomposition.....	91
4.2.3.	Hardware/Software Mapping.....	94
4.2.4.	Persistent Data Management.....	94
4.2.5.	Component Diagram.....	103
4.2.6.	Database Design.....	104
4.2.7.	Access Control.....	105
4.2.8.	User Interface Design .....	109
<b>5.</b>	<b>Chapter Five: Implementation.....</b>	<b>110</b>
5.1.	Overview .....	110
5.2.	Coding Standards .....	110
5.3.	Prototype .....	112
5.4.	Implementation details .....	112
5.4.1.	Client Side.....	112
5.4.2.	Server Side .....	113
5.5.	Deployment .....	114
<b>6.</b>	<b>Chapter Six: System Testing .....</b>	<b>115</b>
6.1.	Objective .....	115
6.2.	Scope .....	115

6.3.	Resources .....	115
6.3.1.	Roles .....	115
6.3.2.	System.....	117
6.4.	Schedule .....	117
6.5.	Test Case Scenarios and Requirements.....	117
6.5.1.	Data and Database Integrity Testing.....	117
6.5.2.	Functional Testing .....	118
6.5.3.	User Interface Testing.....	119
6.5.4.	Performance Testing .....	119
6.5.5.	Load Testing .....	119
6.5.6.	Security and Access Testing .....	119
6.6.	Test Strategy.....	119
6.6.1.	Data and Database Integrity Testing.....	119
6.6.2.	Function Testing .....	120
6.6.3.	User Interface Testing.....	120
6.6.4.	Performance and Load Testing .....	121
6.6.5.	Security and Access Testing .....	121
6.7.	Estimated Risk.....	122
6.8.	Contingency Plan .....	122
<b>7.</b>	<b>Chapter Seven: User Manual .....</b>	<b>123</b>
7.1.	Demonstration of Tatari System .....	123
7.1.1.	Login Page .....	123
7.1.2.	Administrator Dashboard.....	123
7.1.3.	Company Page .....	124
7.1.4.	Location Page.....	125
7.1.5.	Department, Subdepartment, and Designation Pages .....	125
7.1.6.	Employees Page .....	126
7.1.7.	Employee Details Page .....	127
7.1.8.	Employees Directory Page.....	128
7.1.9.	Employee Last Login Page .....	128
7.1.10.	Roles and Privileges Page .....	128
7.1.11.	Core HR Pages .....	129

7.1.12.	Attendance Pages .....	130
7.1.13.	Work Shift Page .....	131
7.1.14.	Overtime Request Page .....	131
7.1.15.	Leaves Page.....	132
7.1.16.	Payroll Page.....	132
7.1.17.	Generate Payslip Page .....	133
7.1.18.	Vacancy Pages.....	133
7.1.19.	HR Report Pages .....	134
7.1.20.	Ledger Accounts Page.....	134
7.1.21.	Payer/Payees Pages .....	135
7.1.22.	Transaction Pages.....	135
7.1.23.	Invoices Page.....	136
7.1.24.	Fiscal Report Pages .....	137
7.1.25.	Settings Page .....	138
7.1.26.	Employees Dashboard Page .....	138
7.2.	Git Structure and Version Control .....	139
<b>8.</b>	<b>Chapter 8: Conclusion and Recommendation .....</b>	<b>141</b>
8.1.	Conclusion.....	141
8.2.	Recommendation.....	141
<b>References</b> .....		<b>142</b>
<b>Bibliography</b> .....		<b>143</b>
<b>Appendix 1</b> .....		<b>144</b>
<b>Appendix 2 – Letter of Request for Cooperation.....</b>		<b>145</b>
<b>Appendix 3 – Sample Source Codes .....</b>		<b>146</b>

## List of Figures

Figure 1.1 Task and Schedule .....	15
Figure 2.1 ERP System Modules .....	21
Figure 3.1 Tatari System Use case 1 HR module .....	38
Figure 3.2 Tatari System Use case 2 Finance module .....	39
Figure 3.3 Tatari System Class Diagram-1 HR Module .....	69
Figure 3.4 Tatari System Class Diagram-2 Finance Module .....	70
Figure 3.5 Register Organization Sequence Diagram.....	72
Figure 3.6 Submit Attendance Sequence Diagram .....	73
Figure 3.7 Process Payroll Sequence Diagram .....	74
Figure 3.8 Submit Job Application Sequence Diagram .....	75
Figure 3.9 Generate HR Reports Sequence Diagram.....	76
Figure 3.10 Transfer Employee Activity Diagram .....	77
Figure 3.11 Process Deposit Activity Diagram.....	78
Figure 3.12 Register Employee Activity Diagram.....	79
Figure 3.13 Generate HR Report Activity Diagram .....	80
Figure 3.14 Submit Job Application and Manage Job Application Activity Diagram .....	81
Figure 3.15 Register Employee State Diagram.....	82
Figure 3.16 Payroll Processing State Diagram .....	83
Figure 3.17 Promote Employee State Diagram .....	84
Figure 4.1 Hierarchical Model-View-Controller (HMVC) illustration .....	88
Figure 4.2 Tatari System High-level System Architecture .....	90
Figure 4.3 System Process - Tatari HR Module .....	90
Figure 4.4 System Process - Tatari Finance Module .....	91
Figure 4.5 Subsystem Decomposition .....	92
Figure 4.6 Hardware and Software Mapping / Deployment Diagram .....	94
Figure 4.7 Persistent Data Management .....	94
Figure 4.8 Attendance Time Object Mapping .....	95
Figure 4.9 Complaint Object Mapping .....	95
Figure 4.10 Department Object Mapping .....	95
Figure 4.11 Transaction Object Mapping .....	96
Figure 4.12 Designation Object Mapping .....	96
Figure 4.13 Job Application Mapping .....	96
Figure 4.14 Employee Object Mapping .....	97
Figure 4.15 User Roles Mapping .....	97
Figure 4.16 Sub department Object Mapping .....	97
Figure 4.17 Payslips Object Mapping .....	98
Figure 4.18 Job Post Mapping .....	98
Figure 4.19 Leave Object Mapping .....	98
Figure 4.20 Promotion Object Mapping .....	99
Figure 4.21 Resignation Object Mapping .....	99
Figure 4.22 Ledger Account Object Mapping .....	99
Figure 4.23 Warning Object Mapping .....	100
Figure 4.24 Shift Object Mapping .....	100
Figure 4.25 Transfer Object Mapping.....	100
Figure 4.26 Termination Object Mapping .....	101
Figure 4.27 Expense Object Mapping.....	101

Figure 4.28 Deposit Object Mapping.....	101
Figure 4.29 Invoice Object Mapping .....	102
Figure 4.30 Finance Transfer Object Mapping.....	102
Figure 4.31 Payer Object Mapping .....	102
Figure 4.32 Tatari System HR Module Component Diagram .....	103
Figure 4.33 Tatari System Finance Module Component Diagram .....	103
Figure 4.34 Tatari System Database Design (Preliminary) .....	104
Figure 4.35 Tatari System Company Page User Interface .....	109
Figure 4.36 Tatari System Employees Page User Interface.....	109
Figure 7.1 Login Page.....	123
Figure 7.2 Departments Page.....	124
Figure 7.3 Company Page.....	124
Figure 7.4 Locations Pages .....	125
Figure 7.5 Departments Page.....	125
Figure 7.6 Employees Page.....	126
Figure 7.7 Employee Details Page.....	127
Figure 7.8 Employee Details - Qualifications Page .....	127
Figure 7.9 Employees Directory Page .....	128
Figure 7.10 Roles & Privileges Page .....	129
Figure 7.11 Complaints Page .....	129
Figure 7.12 Transfers Page .....	130
Figure 7.13 Attendance Page .....	130
Figure 7.14 Add New Work Shift Page .....	131
Figure 7.15 Overtime Request Page .....	131
Figure 7.16 Leave Page.....	132
Figure 7.17 Employee Pay slip Page .....	132
Figure 7.18 Generate Pay slip Page .....	133
Figure 7.19 Add New Job Page .....	133
Figure 7.20 Employees Report Page.....	134
Figure 7.21 Ledger Accounts Page .....	134
Figure 7.22 Payers Page.....	135
Figure 7.23 Add New Deposit Page.....	135
Figure 7.24 View Transactions Page .....	136
Figure 7.25 Invoices Page.....	136
Figure 7.26 Create Invoice Page .....	137
Figure 7.27 Account Statement Page .....	137
Figure 7.28 Database Backup Page.....	138
Figure 7.29 Profile Picture Change Page .....	138
Figure 7.30 Employee Dashboard Page.....	139
Figure 7.31 Git Repository .....	140
Figure 7.32 Git Branches .....	140

## List of Tables

Table 1.1 Development Tools.....	14
Table 1.2 Cost Analysis .....	16
Table 1.3 Team Composition.....	16
Table 3.1 Outage Time to Impact of Operation Comparison.....	28
Table 3.2 Register Organization Use case description.....	40
Table 3.3 Register Employee Use case description .....	41
Table 3.4 Transfer Employee Use case description.....	42
Table 3.5 Promote Employee Use case description.....	43
Table 3.6 Terminate Employee Use case description .....	44
Table 3.7 Process Resignation Use case description .....	45
Table 3.8 Process Warning Use case description.....	46
Table 3.9 Submit Attendance Use case description .....	47
Table 3.10 Manage Schedule Use case description .....	48
Table 3.11 Generate Timesheet Use case description.....	49
Table 3.12 Process Overtime Use case description .....	50
Table 3.13 Process Leave Use case description.....	51
Table 3.14 Process Payroll Use case description .....	52
Table 3.15 Generate Payslips Use case description .....	53
Table 3.16 Create Job Posts Use case description .....	54
Table 3.17 Submit Job Application Use case description.....	55
Table 3.18 Manage Job Application Use case description .....	56
Table 3.19 Generate HR Reports Use case description .....	57
Table 3.20 Create Ledger Account Use case description .....	58
Table 3.21 Manage Ledger Account Use case description .....	59
Table 3.22 Add Payee Information Use case description .....	60
Table 3.23 Add Payer Information Use case description.....	61
Table 3.24 Process Deposit Use case description .....	62
Table 3.25 Process Expense Use case description .....	63
Table 3.26 Transfer Funds Use case description .....	64
Table 3.27 View Transactions Use case description.....	65
Table 3.28 Create Invoice Use case description .....	66
Table 3.29 Generate Fiscal Reports Use case description .....	67
Table 3.30 View Profile Use case description .....	68
Table 3.31 Request Overtime Use case description .....	68
Table 3.32 Data Dictionary.....	71
Table 4.1 Subsystem Services and Purpose .....	93
Table 5.1 Implementation details.....	114
Table 6.1 Test System Description .....	115
Table 6.2 Human Roles Testing Resources .....	116
Table 6.3 System Resources for Testing.....	117
Table 6.4 Test Schedule .....	117
Table 6.5 Data and Database Integrity Testing.....	120
Table 6.6 Function Testing .....	120
Table 6.7 User Interface Testing.....	121
Table 6.8 Security and Access Control Testing .....	121

## **Abbreviations**

Admin –	Administrator
API –	Application Programming Interface
ASTU –	Adama Science and Technology University
CI -	CodeIgniter
CPU –	Central Processing Unit
CRUD –	Create, Read, Update and Delete
CSV –	Comma-Separated Values
DOM –	Document Object Model
EMPDE –	Ethiopian Educational Materials Production and Distribution Enterprise
ERP –	Enterprise Resource Planning
FDRE –	Federal Democratic Republic of Ethiopia
FR –	Functional Requirement
GUI –	Graphical User Interface
HMVC –	Hierarchical Model-View-Controller
HR –	Human Resources
HTTP –	Hyper Text Transfer Protocol
ICT –	Information Communication Technology
MoFED –	Ministry of Finance and Economic Development
MVC –	Model-View-Controller
PDF –	Portable Document Format
RBAC –	Role-based Access Control
PHP –	Hypertext Preprocessor
RAM –	Random Access Memory
SQL –	Structured Query Language
WOOM –	Web-based Object-Oriented Mode

# **1. Chapter One**

## **1.1. Introduction**

Resource Management System is application software that enables businesses and organizations to utilize a system of integrated programs that can manage a business and automate various backend office operations relating to service, technology, human resources, and finance. This software combines all aspects of a business operation including product planning, manufacturing, sales, and management. It benefits the organization by enhancing efficiency, productivity, minimize operational costs, optimizing resources, and better stock supervision.

One of the largest commercial enterprise software development companies, Oracle [\[1\]](#) defines Enterprise Resource Planning (ERP) as a type of software that businesses and organizations use to manage daily business operations and activities such as finance, procurement, human resources, supply chain operations, and risk management. A comprehensive ERP suite can be seen as a collection of various resource management systems tying together a multitude of business processes in which each system deals with a certain aspect of the business-like accounting, inventory, sales, or others.

Over the years, ERP has evolved; from the 1990s until the early twenty-first century, ERP adoption grew exponentially. The hardware infrastructure required to implement the software was typically on company premises, in a server room. The hardware and software licenses needed capital investments and depreciated over 5 to 10 years. Moreover, organizations nearly always wanted to customize the system to fit their specific goals and needs, entailing additional expense of software maintenance, upgrade, and training [\[1\]](#).

These systems are generally designed with the notion of a single, defined schema that typically has a common database. This ensures that the information communicated across the organization is regulated and relies on common classifications and user experiences. The core functions are then integrated with business processes driven by workflows across various departments such as finance, human resources, manufacturing, marketing while connecting said functions and users.

With the increasing demand from the market, businesses need to foresee their client's needs, to create customer loyalty and boost their business. However, African enterprises are probably prone to being unable to satisfy their customers. There also seem to be some blockades that limit their competitiveness in domestic and international markets. To overcome these limits, a solution might be the implementation of integrated information systems. This information system is anticipated to help organizations meet their strategic goals of development and maintain their role within the global marketplace. Enterprise Resource Planning (ERP) and Resource Management Software are of this kind of systems; to be viewed as the technology that delivers the incorporated business function to the organization by integrating core operations. (Tobie et al., 2016) These systems improve the organization context by integrating all the disparate data into a unique database.

Even though Ethiopia is still lagging compared to many developing countries, ICT penetration and usage is steadily growing. The potential for information systems to increase economic growth and reduce poverty is well-supported, and Ethiopia has begun to embrace information technology use in the whole social, economic, and political structures. The government has strategized ICT development as one of its top objectives. However, there is still a lot of work to be done in computerizing and digitizing the fundamental governmental organizations as well as public and private enterprises to bring about a systematic and modern change to the daily business operations of these organizations.

Some governmental and non-governmental organizations in Ethiopia use resource management software by purchasing them from abroad, which is expensive and also may be incompatible with the business needs here in Ethiopia. Few large companies in Ethiopia such as Ethio telecom, Mesfine Industrial Engineering Pvt. Ltd, Commercial Bank of Ethiopia and MoFED use ERP systems to manage their operations; these systems are very expensive with high failure evidence upon implementation. Other organizations use the common spreadsheet technique to manage their workflow or use the old paper-based method to run their business, that are very inefficient.

The proposed project aims to build a system that can be used to manage the resources of an organization primarily human resources and finance. The system to be built should support the business operations that occur in an organization principally the core human resource processes such as promotions, transfers, resignations, payroll, and the accounting processes such as deposit, expense, fiscal reporting. The system is planned to be scalable, secure, and user-friendly for both technical and non-technical personnel. Furthermore, the proposed system plans to account for the existing business practices in order to ease the transition to a computerized system.

## **1.2. Background of the Project**

Organizations are introducing information and communication technology (ICT) that could add value to their goods, processes, and customer satisfaction in the current dynamic global market. One of these processes for collecting, organizing, storing, and sharing organizational information anytime, anywhere in the adoption, application, and use of homegrown enterprise resource management systems. Although companies have been using enterprise resource management systems and ERP systems, homegrown ERP systems have received less attention than commercial off-the-shelf applications.

Alemu et al. (2015) has described that Over the last few decades, enterprise resource planning (ERP) programs have become a key method for enterprises. An ERP system automates essential business processes and acts as a central repository for all financial and operational data from around the organization. It gathers this information from a number of modules designed to assist different departments, ranging from accounting to supply chain to human resources, in carrying out their duties. An ERP framework must be used to ensure access to an efficient and secure information infrastructure in order for an enterprise to successfully compete in this rapidly growing marketplace. They've basically taken over as the default choice for replacing the legacy system.

Even though organizations have traditionally performed all operations manually, the expanding nature of their activities and globalization is allowing them to adopt, implement, and use information and communication technology. ICT has been empowering organizations globally to perform operations effectively and efficiently. Currently, the exchange of information pertaining to products and services is being simplified with the support of ICT. Organizations have been using both commercial off-the-shelf and custom or tailored software to support their organizational processes and perform competently. Even though these two types of software have their advantages and disadvantages, most researches show that the advantage of tailored software is much more preferable for enterprise resource planning (ERP) purposes than commercial off-the-shelf software.

Zhelyazkov (2012) explains that “ERP systems are comprehensive software packages that seek to integrate the complete range of a business process and functions to present a holistic view of the business from a single information and information technology architecture”.

ERP systems improve interdepartmental collaboration and cooperation because they can construct a single enterprise that manages all departmental functions using a shared database that handles all business transactions (AlQashami and Heba 2015).

Ebizframe [2] indicates that Many Ethiopian businesses have successfully introduced ERP systems. ERP Software, once implemented, brings enormous benefits to the enterprise, such as improved quality, better use of scarce resources, and cost reduction. An ERP Suite is important for integrating and automating an organization's business processes. Enterprise resource management systems, such as ERP in Ethiopia, have aided in exposing Ethiopian businesses to foreign best practices and processes, as well as catalyzing improvements of their competitiveness and performance. Ethiopian organizations are increasingly seeing and realizing the tremendous benefits that a flexible and efficient ERP System can provide, as well as the critical need to begin their own enterprise automation journey with the implementation of a suitable enterprise resource management software solution in their own company.

It is a common misconception that implementing an ERP framework would instantly increase an organization's functionality. The high expectation of achieving overall cost savings and service improvements is highly dependent on how well the selected ERP system matches organizational functionalities and how well the system's tailoring and configuration process suited the organization's business culture, strategy, and structure. In general, an ERP system can boost both backbone and front-end functions of the business at the same time (Liaquat Hossain et al., 2002).

Many companies have recently been confronted with constant demands from rapidly evolving and highly competitive global markets. They must also cater to consumers who demand creative, high-quality products with unique services. Technology advancements, which shorten many product lifecycles, and increased international competition, which drives collaboration to minimize costs and boost production efficiencies, add to the pressures. Companies must have versatile business

information systems that can respond to rapid changes to gain a competitive advantage. Enterprise business applications provide solutions that focus on the consumer by leveraging the accounting, human resources, and supply chain to meet these needs. (Kumar, 2004).

According to the majority of papers and articles, we read, although custom enterprise resource management systems are far more recommended, particularly for developing countries, organizations in Ethiopia have been hesitant to accept homegrown enterprise resource management systems. Enterprise resource management systems must be implemented correctly in Ethiopia in order to automate, review, and regulate a company in real-time across all departments and locations. The consistency of the product can also be much superior to the manual labor-intensive methods that were previously used.

### **1.3. Background of the Organization**

The organization we have selected and contacted to model and test the different ERP modules that are going to be developed as a result of this project is the **Ethiopian Educational Materials Production and Distribution Enterprise (EMPDE)**.

The FDRE Council of Ministers under Regulation No. 50/1999 [3] authorized the establishment of the organization known as Educational Materials Production and Distribution Enterprise (EMPDE), this regulation cites that the operations and functions of the institution are governed according to the Public Enterprises Proclamation. The enterprise has its head office in Addis Ababa, Ethiopia around Gurd Shola, Megenagna. It also has branches all over the country that handle the distribution and other related tasks. The enterprise has an operational authorized capital of 150 million Birr. It has 337 full-time employees and over 1200 part-time contractual staff nationwide.

Educational Materials Production and Distribution Enterprise have defined its objectives as follows:

- Production of educational materials, Domestic and Foreign purchase;
- To give training and professional consultancy service;
- To increase profit and create a stable market;
- To engage in other related activities for the attainment of its objectives.

‘Educational Materials’ shall mean Textbooks, Visual Aids, Charts, Maps, Science and Vocational Instruments, Children’s Toys, different forms, and other materials applied in the educational sphere.

The enterprise provides its services to most government universities, secondary and primary schools. It receives orders of procurement from various parts of the country with the official capacity to get different educational materials that would help them in their daily activities. The organization's mission, vision, values, and objectives [4] are presented as follows.

### **1.3.1. Mission of the Organization**

To provide quality products and services to customers using modern technology, skilled manpower, and effective working procedures that assure the sustainability of the organization.

### **1.3.2. Vision of the Organization**

To make the enterprise one of the leading producers and distributors of learning and teaching materials as well as printing products in the sector.

### **1.3.3. Values of the Organization**

By strengthening the best experiences and working culture which developed through time and by avoiding harmful practices; the future products and services depend on the following values:

- Competitiveness and Excellence
- Accountability and Transparency
- Effectiveness and Vigilance
- Recurrent capacity building works and efficiency
- Give positive response and respect to complaints and comments which arise from customers
- Getting ready to educate oneself and change by accepting the working procedure from the heart and being firm and consistent for the implementation.

### **1.3.4. Objectives of the Organization**

- Prepare, publish and print student textbooks, teacher guides, reference materials, charts and maps.
- Produce science, technical, vocational and other educational equipment as well as repair and maintain them.
- Produce furniture and chalk for schools and universities
- Produce materials for kindergarten
- Purchase from abroad and locally per the government procurement regulations and distribute the materials among various beneficiary institutions throughout the country.
- Carry out the research necessary for the production and procurement of improved materials.

## **1.4. Statement of the Problem**

The workplace at EMPDE can be characterized as almost entirely a paper-based environment. Stationery products such as paper, pen, and copiers are used. Handwritten documents, table and list form papers are the basic mechanisms to collect, disseminate and keep information records. This paper-based operation is costly with stationery expenses accumulating over time. The organization also acknowledged that this paper-based manual technique has put a strain on their performance as it is time-consuming to maintain, find and file records accordingly. This paper environment is unsafe as well as inefficient, it is exposed to various damages such as water damage, rendering the record unusable.

The Human Resources department has disclosed that among the major problems it faces are improper record-keeping with paper or spreadsheet, lack of vacancy announcement platform, and lack of analysis and reporting tools for employee and organizational information which has led to limited performance and statistics. Attendance of employees is the other problem that is maintained in which their names have to be hand-written and listed out in paper; which has to be updated almost every week as the paper runs out of space. This process has created an inefficient way to monitor employees' work habits, punctuality, and performance. With the very large number of employee turnover mostly technical and manual labor workers, the enterprise has had to announce vacancies using third parties such as newspapers, websites, and paper posts. This has clumped the schedule of the organization with the time for contacting potential third parties and the vetting process once candidates apply.

The Finance department also disclosed that the critical problem is that as there is no integrated system of communication between other departments such as human resources, the transaction processing most of the time suffers from unreliable and inconsistent information as each department keeps a spreadsheet of their records. Financial records such as receipts, purchase orders, payslips, bills, and alike are the other paper-based outputs that different departments of the organization produce, which are all difficult to manage, organize and store. These financial records even the one electronically generated being in paper creates the problem of processing them in bulk at a time to complete a fiscal year, puts strain and delay on audits and reports as well as opens doors for maleficence and discrepancies.

The major problems that the cases above have discussed are the paper-based manual hand-written environment, time delays, resource wastages, outsourcing critical tasks to third-parties, expensive financial expenses for stationeries, unsafe information archive mechanisms, lack of organized data for reporting and statistics as well as inefficient attendance monitoring and fiscal record keeping have created an unproductive workplace. Overall, the enterprise faces a lot of problems including the ones discussed above. The work environment portrays inefficiency, poor performance, inorganization, and poor record-keeping practices.

## **1.5. Justification of the Project**

Workwise [5] indicates that numerous organizations have obsolete processes in place that never change merely because it has always been there like spreadsheets and paper-based activities. This can complicate and candidly waste time in the long run. Also implementing various native systems for different operations with no way to communicate information across brings similar problems. There's no reason to be reluctant to change, particularly if that change can create a more suitable and uplifting stream of work for every party involved. Transitioning to a modular integrated resource management system allows businesses to keep track of everything from employees, finances, inventory, and orders all in one place.

Lack thereof an integrated resource management system in place that can communicate different branches of an enterprise, especially one of a larger size, there is so much critical information that goes unknown, left behind, and not shared across. There is also the case of high miscellaneous expenses that would otherwise be avoided if a unified information system is in place. So, one of the key benefits of such a resource management system being implemented is that it would facilitate the communication of information throughout the enterprise while reducing operational costs. It has been disclosed that the enterprise is open and willing to work with the team and assisting through every phase of the project. The swift response from the organization to collaborate and readiness to deploy such a system further proves the need for this project.

In turn, the motivation for the project can be considered as the inefficient workplace and operations that are observed in EMPDE and other different organizations which can be resolved by using a computerized resource management system like some ERP modules that suit the business needs of the organization. Moreover, the expensiveness of procuring or developing as well as deploying such a system indicates that there is a demand for the product, it just happens to be effortful in actually realizing it. Hence, the project as well as the motive behind it is justified; since the project plans to deliver an affordable, customizable, scalable, and localized resource management system as a product.

## **1.6. Objective of the Project**

The general and specific objectives of the project are presented as follows.

### **1.6.1. General Objective**

The general objective of this project is to design and develop a web-based resources management system for the Ethiopian Educational Materials Production and Distribution Enterprise (EMPDE).

### **1.6.2. Specific Objective**

The specific objectives of this project in order to meet the general objective are:

- To gather the requirements of the system using various data collection methods,
- To study and analyze the currently available related technologies and systems in the enterprise resource planning and management area;

- To review different articles about the design, development, and implementation of various ERP systems and modules.
- To design and develop a database to hold various records of the enterprise such as employee, transactions, and attendance which can be managed with a system in place.
- To design and develop a web-based human resources management system module:
  - ✓ Develop an employee information management system
  - ✓ Develop a core HR processing system
  - ✓ Develop a job application and recruitment system
- To design and develop a web-based finance management system module:
  - ✓ Develop an account or general ledger management system
  - ✓ Develop a financial transaction record (payer, payee, balance, deposit) system
  - ✓ Develop a payroll management and payment invoice system
- To evaluate and test the systems mentioned above separately.
- To integrate the two modules mentioned above into a single application suite.
- To implement and test the system with real work environment demonstration at EMPDE.
- To document the overall procedure and findings that occurred while analyzing, modeling, designing, developing, and testing the products.

## 1.7. Scope and Limitation

### 1.7.1. Scope

This project will focus on developing a web-based enterprise resource management system that mainly comprises the human resource management system and finance system for organizations. This system will let users (administrators, managers, employees, accountants) interact with their corresponding day-to-day operations. It will facilitate core HR (Human Resource) and Finance/Accounting processes that an organization uses daily by providing different features that would provide means to view, manage, monitor, track, record, and report organizational activities of the HR and Finance department.

The HR management system module in this proposed system includes the following features:

- Employee Information Management (Personnel Tracking) – generally considered one of the most fundamental features of human resources software. It serves as a database for all employee details, including names and addresses. Job names, pay, and so on.
- Core HR processes – activities like Transfers, Resignation, Termination. Promotion and Warning features are included in this module.
- Reporting – Collect and report on employee and applicant data to analyze any trend that may appear.
- Payroll – Cut paychecks, generate payslips, track salaries, bonuses and deductions. Note that some features like this one will incorporate with the finance/accounting module.
- Time and Attendance Tracking – enables employees to clock in and out, tracks attendance and absences, generates timesheets along with HR calendar.

- Recruitment/ Applicant Tracking – a centralized database for recruiters to store and access vacancy announcements, applicant information throughout the hiring process/

The Accounting/Finance module in this proposed system includes the following functionalities:

- General Ledger – a history of all financial transactions that can be used to compile financial statements and monitor both inbound and outbound funds.
- Accounts Payable – a list of all the money a company has paid or owes to its vendors or other creditors.
- Accounts Receivable – a list of all money owed to the company that has yet to be paid.
- Billing and Invoices – allows for sales, purchases to be billed with invoicing payments.
- Financial Reporting – allows for tracking funds, measure performance, and optimize profitability. These reports often include expense, income, deposit, transfer reports.
- Expense Management – to keep track of all work-related expenses such as travels, maintenance.

### **1.7.2. Limitation**

The resource management system covers a multitude of services in the human resources and finance sector. However, due to time, lockdown, financial, and other restrictions, the proposed system faces the following limitations:

- Basic computer knowledge, as well as some experience in operating such systems, is required to work on the system.
- Due to the pandemic and limited time, requirement and information gathering proved to be difficult. There wasn't satisfactory contact and communication to the model organization for said reasons.
- The system may require a brief maintenance period for any failures or customization needs.
- The system is not a distributed system; it operates using a central server using on-premise network infrastructure that keeps track of all activities of the system which limits the capability and scalability of the system.
- To operate the system outside the local environment for example from a different branch at a different location, an internet connection is required to communicate with the server.
- The system does not consist of Inventory management, Project management, Customer Relationship Management, or Supply Chain management due to various constraints like time. However, given the proposed modules are developed successfully, these modules will be considered to be developed and integrated into the system in the future.

## **1.8. Feasibility Study**

A feasibility study is essential to evaluate the cost and benefits of the project. Based on the feasibility study decision is taken on whether to proceed, pivot, iterate or cancel the project. The project aims at supporting resource management systems to reduce risks and handle the implementation process in a holistic and integrated way, highlighting the critical interaction between aspects concerning technical and organizational issues; strengthening its feasibility. Aspects of technical, operational, and economical feasibility are discussed as follows.

### **1.8.1. Technical Feasibility**

As the system in question is a multi-dimensional and practical area with a real work environment, it would prove to be challenging in investigating and developing the system which manages different parts of the work environment after assessing and determining which aspects of the work can potentially be used computerized and integrated into a web application. The proposed project will entail the use of web development in developing an integrated system that can attain the given objectives, the team is capable of designing and developing such a system given enough background information and research on enterprise structure and activities. Hardware including desktops and laptops along with software tools such as CodeIgniter web framework, Xampp local server are easily obtained and ready for the upcoming design and development phase. The project team is capable of working with said resources drawing experiences from various coursework and projects before. Thus, it is technically feasible to develop the described system with achieving the goals in mind and practicality.

### **1.8.2. Operational Feasibility**

Operationally speaking, the project tends to bring about a product to enterprises that will also be developed with EMPDE's assistance and backing which would make it more realistic in bringing about a better and relatable product. The system would provide services as per the requirements of the enterprise. It is the goal of this project to bring about a product that is easy to use with minimal technical expertise along with low overhead for maintenance and updates after deployment. The product would then be available for different organizations, proven and tested that it holds general standards and qualifications. The product would be tailored to the needs, standards, and qualifications of the enterprise to be tested on; with general operational scopes and aspects of any organization in mind and maintained throughout the development, which would enable it to be operationally fitting to generally any workplace environment from small to large businesses and organizations. As it is one of the goals to develop an operation-capable product that can potentially and successfully perform in the workplace, it can be said that the project is operationally feasible in terms of attaining the objectives outlined.

### **1.8.3. Economic Feasibility**

Costs of developing the product along with the entire procedures of the project would not be very high, as the development uses a systematic study of different ERP system modules, basic on-site survey, practical background knowledge in database design, web development, and system development with readily available hardware and software as well as open-source materials; compared to the various and significant resources that are being wasted and spent there now daily the proposed system will bring about a great change and difference with what is being used now. ERP systems on the market even individual modules like Human Resource Management System are expensive as they are developed abroad and sold with foreign currency to us. The product

would immensely reduce overall costs that an enterprise spends unnecessarily with the current system or procured one from a foreign vendor, the proposed system would mitigate by saving those costs along with boosting efficiency and productivity. It can be said the project is economically feasible in terms of cost and benefit as well as providing an affordable product.

### **1.9. Significance of the Project**

The project's main goal is to address the numerous productivity and performance problems that concern many organizations by developing a product, namely a resource management system, that would streamline many of an organization's operations, resulting in a more effective and competitive work climate.

These system's modules will support a specific company by increasing productivity and profitability, standardizing business processes, improving teamwork and workflow, and improving monitoring and planning activities. A company will be able to better manage and control its assets, such as human resources and finances, with greater ease, flexibility, scalability, and security, all while staying within a reasonable budget.

Many governmental and non-governmental entities in need of a human resources and finance management system, like the model organization, will greatly benefit from this project once it is completed with a working product. Some of these significances are presented below:

- ✓ To improve efficiency and productivity of small to large scale organizations;
- ✓ Enhance the business process and business reporting activities of various organizations;
- ✓ To save unnecessary costs and lower operational cost that are now being spent for many reasons which would be minimized upon implementation of the system;
- ✓ Automate the manual process, such as invoice creation, routine communication, financial report generation, timesheet generation;
- ✓ Improve accessibility and security of the organizational information which would then be recorded and accessed consistently and reliably;
- ✓ Reduce third-party dependencies in outsourcing and involvement of many external vendors
- ✓ Provide structured human resources and financial information system along with features mentioned which would greatly benefit enterprises, by also reducing risk with data integrity
- ✓ Enhance the very workflow and integration of various departments and operations of an organization.

### **1.10. Beneficiaries of the Project**

According to Terillium [6], an Oracle Partner; the type of business that would benefit the most from such resource management systems or ERP system are businesses engaged in manufacturing, distribution, construction, industrial services, healthcare, and alike; due to the versatile nature of ERP software which allows companies across industries to implement solutions based on their business needs.

The primary beneficiary of this project is the Ethiopian Educational Materials Production and Distribution Enterprise (EMPDE) which is a national enterprise engaged in the manufacturing and distribution industry that could benefit from such a system that the project aims to develop. The human resources department, accounting department, high-level to low-level employees of the enterprise as well as the government would benefit from the system.

Small to medium companies and organizations in a related sector that need a resource management system for their activities are also among the project's beneficiaries. With minimal configuration and customization, the system is built as easy-to-use, inexpensive, and scalable will greatly enhance the goals of the system being deployed and implemented in any company with a similar structure and architecture. The project aims to support businesses by lowering costs and increasing efficiency for businesses who want to use the system in their industry.

## **1.11. Methodology**

The systematic, theoretical analysis of the techniques used in a field of research is known as methodology. It entails a theoretical examination of a body of methods and concepts associated with a field of study. It is the mechanism and method for gathering and analyzing data to generate useful information. As a result, we've decided to use the following approaches to collect data, design, develop and test our system.

### **1.11.1. Data Collection Methodology**

The data collection methodology utilized for this project is discussed as follows. We have used both primary and secondary data collection methods. As primary data collection method, the qualitative approaches such as observation and open-ended question interviews were used. As secondary data collection method, we have collected technical organization documents, related publications and other open-source materials.

One of our team members has observed and experienced first-hand what the environment at Educational Materials Production and Distribution Enterprise Addis Ababa Head Office is like and what he has described from his preliminary observation is the work environment has a lot of full-time and temporary employees. This workplace can be characterized as almost entirely a paper-based workplace with a slight spreadsheet activity at several points. They use paper products, hand-written documents, and stationaries for their daily operations; which costs a lot of money when it accumulates over time. It was observed that using paper for every aspect of their work has constrained their productivity and performance.

Another valuable part of the project is to conduct a deep on-site study and survey of the workplace to determine the potential aspects to solve using this project which would involve informal interviews with different administration bodies of the enterprise and also employees. These open-ended interviews have greatly helped in determining the design structure and possible features to integrate into the web application suite. The interviews are accompanied with requesting and

acquiring different documents/records such as paper forms and lists used in the office to carry out different tasks, as well as formats for invoices issued for any sale or purchase; which are all inputs for the digital forms to be designed on the system.

One of the methodologies we utilized while working on this project is the analysis of other available (maybe opensource) human resource management systems, financial management system as well as ERP systems to gain a deep understanding of available systems, their workflows, functionalities, features, limitations and alike. This would help us understand what the product would have to look like, what it has to include, and the overall steps and procedures to get there.

A literature review is also another important method to acquiring vital information and data about resource management systems as well as ERP systems. This would the enterprise, prior researches into building a related system, and digital infrastructure that the enterprise uses and also related articles and documents would be studied.

### **1.11.2. Design Methodology**

Upon results obtained from the data gathering and analysis as well as reviews, system design and modeling will follow an Object-Oriented Design approach. It would be used to model the system's functions in an organized and object-oriented manner. This phase would allow to design and refine UML models such as use-case, sequence, and activity diagrams identified, study their interaction with other objects and create scenarios that best fit the environment and processes involved.

### **1.11.3. Development Methodology**

Application development will then follow with the survey and reviews in the arsenal, application suite development will follow a component-based or modular-development approach, as the system is at first modularized and can be developed separately. Functionalities will be maintained as best possible through the development cycle. The different types of implementation and deployment stages will be accompanied by relevant expert opinions from university advisors.

The primary development methodology we plan to utilize for this project is the Agile Software Development Methodology. Because it fulfills the demands of the project as best possible; it would account for the quick development to deliver a working product which we would need as time is very limited for this project's design and analysis phase. The method also reduces risk and welcomes changing requirements later on. It promotes the team to meet regularly and discuss the progress. The system will be developed in increments of modules that would be identified from the features specified. We selected it also as it maintains simplicity in both systems being developed and in the development process.

Testing, Deployment, and Documentation will also be used as project methodology to bring about a reliable and well-documented product. The developed web application suite will go through several tests, with every step documented for future work and also for product specification.

## 1.12. Development Tools

The following development tools are considered to be used in different phases of the project.

**Hardware:** Desktops, Laptops, USB Thumb drives

	Software	Use
<b>Operating System</b>	Windows 7/10, Kali Linux 2020	<i>Development environment operating system</i>
<b>Diagram Tools</b>	Enterprise Architect, Microsoft Visio, Draw.io	<i>To design and draw different types of diagrams to illustrate various database structures and activities</i>
<b>Documentation</b>	Microsoft Word 2020, Microsoft Excel 2020, Microsoft Publisher 2020	<i>To document the project and the product, product specification, statistics, and technical paper documentation</i>
<b>Programming Languages</b>	HTML, CSS, JavaScript, PHP	<i>To develop the different modules of the web-based system</i>
<b>Database</b>	MySQL running on MariaDB's XAMPP server	<i>To handle the various kinds of records that need to be kept on a database,</i>
<b>Framework</b>	CodeIgniter PHP Framework, along with custom modules	<i>To develop and showcase the different systems that are the results of the project.</i>
<b>Browsers</b>	Mozilla Firefox, Google Chrome	<i>To develop, debug and build the different systems of the web application suite.</i>
<b>Editors and IDEs</b>	Sublime Text, Atom, Visual Studio Code	<i>To write, debug and execute the source codes and interfaces of the system being developed.</i>
<b>Version Control</b>	GitHub	<i>To collaborate and maintain the entire history and version of the codebase and other related resources of the project</i>
<b>Collaboration and Communication Platform</b>	Slack, Telegram Private Group, Jira Software	<i>For communication, task tracking, and collaboration of the team members &amp; advisor.</i>

Table 1.1 Development Tools

## 1.13. Testing Plan

In order to deliver a quality product as well as reduce project risk; testing is a vital phase for which we plan to run a range of tests throughout the requirement analysis, design and modeling, development as well as deployment phase. The following types of tests are planned to be carried out through each phase:

Requirement Testing: where we plan to test cases, conditions, and requirements derived from the analysis phase. It would include functional and non-functional tests.

Unit Testing: This test focuses on a minimal system component, feature, or module to determine and verify if it performs as expected and identified from the analysis and design phase. This would include running white-box and black-box tests, to determine if the system is operational, error-free, and optimized for better performance.

Integration Testing: this is where we plan to test how modules would operate after they are combined, where modules are tested as a group after they are integrated into the system. It occurs after unit testing is carried out successfully.

System Testing and Acceptance Testing would also be part of this testing phase to determine if the product as a whole performs as needed and that it can be accepted by the test model organization which is Educational Materials Production and Distribution Enterprise at Addis Ababa, Ethiopia. Alpha and Beta tests are included in this particular testing phase.

## 1.14. Task and Schedule

This project upon approval of the proposal by the Senior Project Committee will be conducted between February 25<sup>th</sup>, 2021, and May 5<sup>th</sup>, 2021. The timeline for the semester project depends on the First Semester's schedule for the Academic year 2020/2021. The complete implementation phase which consists of the development, testing, and product delivery is primarily planned to be commenced on the second semester of the academic year. Due to the limited time allocated for the project, all tasks and objectives may not be completely fulfilled.

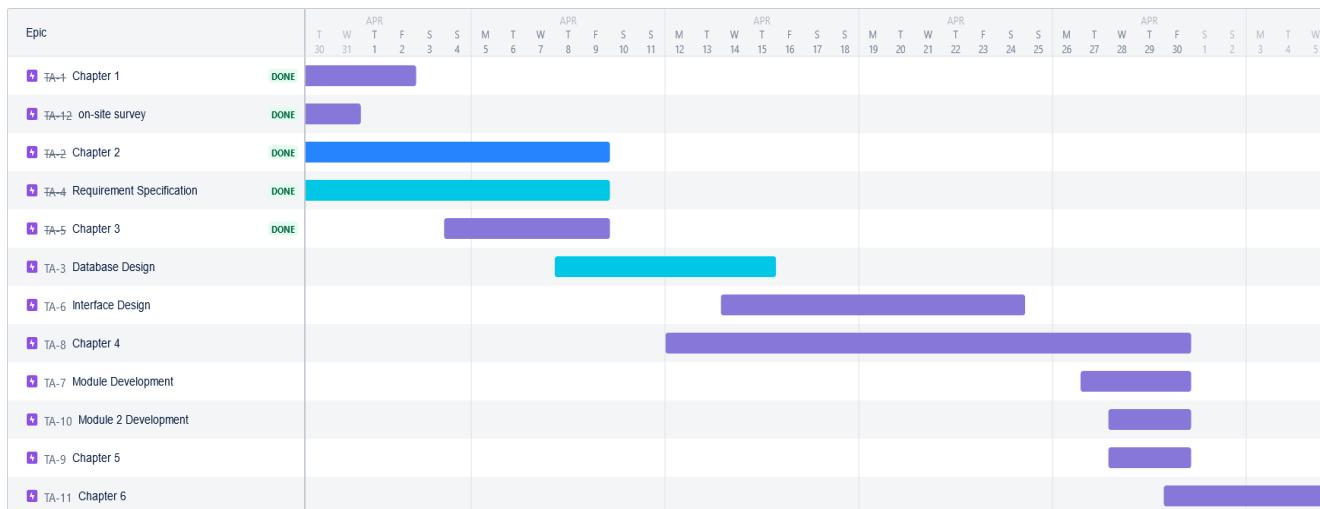


Figure 1.1 Task and Schedule (Courtesy of Jira Software Project Task Management Board)

## 1.15. Cost Analysis

Although there is no official budget/cost allocated for this project, here is an estimated budget/cost for resources required for the project:

Resource	Amount	Cost	Total Price
<i>Printing and Stationery</i>	-	300 ETB	300 ETB
<i>Laptop</i>	1	12,000 ETB	22,000 ETB
<i>USB Drives</i>	2	250 ETB	500 ETB
<i>Microsoft Office Suite</i>	1	70 USD (2800 ETB)	2800 ETB
<i>Microsoft Windows 10 Pro OS</i>	1	200 USD (8000 ETB)	8000 ETB
<i>Transportation</i>	-	500 ETB	500 ETB
<i>Internet</i>	~100 Hours	1500 ETB	1500 ETB
<i>Miscellaneous</i>	-	500 ETB	500 ETB
<b>Total Cost</b>			36,100 ETB

Table 1.2 Cost Analysis

## 1.16. Team Composition

No.	Name	Email	Major Responsibilities
1	Abreham Bekele	abrehambk54@gmail.com	Requirement Analysis, UI/ UX Design, Graphics, Development, Testing
2	Arefat Hyeredin	arefat.hyeredin@gmail.com	Team Leader, Requirement Analysis, Design, Development, Implementation, Testing, Documentation
3	Biniyam Gossaye	biniyamgossaye@gmail.com	Requirement Gathering, Design, Development, Testing, Documentation
4	Girum Getachew	girumgetachewmain@gmail.com	Requirement Gathering and Analysis, Design, Implementation, Testing
5	Gizealew Endeshaw	giz863939@gmail.com	Requirement Gathering and Analysis, Design, Development, Testing
<b>Advisor:</b>		Mr. Shumet F.	

Table 1.3 Team Composition

## **2. Chapter Two**

### **2.1. Description of Existing System**

EMPDE is a national government organization with several branches throughout the country and customers from all over the nation. It is common to see companies in Ethiopia and Africa with vast resources and production capacity without a proper computerized resource management system in place.

The current existing system at Ethiopian Educational Materials Production and Distribution Enterprise (EMPDE) is a manual, paper-based system. A manual paper-based system is a record-keeping system using pen and paper and, in some cases, printed documents with tables and forms to be filled out by hand. There is limited use of spreadsheets in the organization which primarily use Microsoft Office products such as Microsoft Excel which are printed or copied and shared physically.

The Organization has several departments such as human resources, accounting/finance, general services, production/manufacturing, distribution, technical, and maintenance. Focusing on the human resources and finance department, it can be described as the management goes from top-to-down as instructions and strategies are passed down the chain. Human resources department has human resource officers along with the leader of the department HR Administrator. Finance Department has several senior and junior accountants as well as cashiers, under the leadership of the Head of Finance.

The human resources department uses daily and weekly attendance sheets to monitor attendance. They use paper forms to process basic HR functions such as terminations, complaints, resignations, leaves, and promotions. It has a bunch of blank spaces that have to be filled out by the HR officer or blank pages where reports have to be handwritten. They also use excel spreadsheets to keep records of some activities but are not structured or consistent. The department also announces vacancies on newspaper and websites, and accept applications in postal mail or physically at the office. The department also approves payroll aligned with the timesheet manually.

The Finance department utilizes a few accountings software such as an outdated Peachtree Accounting software to process transactions including excel spreadsheets. They keep manual receipts, bills, and invoices which have to be stamped and signed accordingly. They have cash register machines and spreadsheets to process expenses but no other digitally kept financial data.

At EMPDE, the work environment is almost entirely paper-based. They run their business on paper items, handwritten notes, and stationery, all of which add up to a lot of money over time. As businesses are bombarded with piles of reports, files, and documents, their attempts to analyze data, maintain compliance, and manage accounts become increasingly futile. It is observed that the filing/record-keeping system in place involves archives full of paper documents that hold records of transactions, employee files, daily operations, and other related work tasks due to the lack of a digital filing system.

## **2.2. Major Functions of Existing System**

Major functions of the existing paper-based and spreadsheet system at EMPDE include the following:

- Register employee information
- Perform core HR process on employee records
- Keep attendance of employees
- Announce vacancies
- Process payroll
- Record job applicant's information
- Receive client's orders
- Keep records of inventory items and stock capacity
- Keep financial records of transactions
- Keep procurement, expenses, and purchase records
- Keep sales records
- Issue invoices and bills for sales, purchases, and payments

## **2.3. Users of the Current System**

Mainly, users of the current system are the various departments at EMPDE like any other organization which is summarized as follows:

- High-level Administration
  - Upper administration uses the existing system to monitor the overall operations of the different departments and branches of the enterprise.
- Human Resources Department
  - HR administrators along with mid-level HR managers use the existing system to keep records of employees, keep attendance of employees and announce vacancies.
- Accounting/Finance Department
  - Head of Accounting along with senior and junior accountants use the existing system to process payroll, sales, issue invoices and keep financial records.
- Production Department
  - The production department keeps track of products, inventory, and stock using the current system.
- Distribution Department
  - The distribution department keeps track of orders, supply, and delivery using the current system.
- Staff/Employee
  - Employees use the existing system to register information, fill out attendance timesheets and receive salaries.
- Clients/Customers
  - Clients use the current system to place orders and process payments.

## **2.4. Drawbacks of the Current System**

As previously discussed in the Statement of the Problem section on Chapter One, the current paper-based manual system at EMPDE has various systemic drawbacks. The work environment generally portrays inefficiency, inorganization, and lack of structure. The major drawbacks that the cases above have discussed the current system are time delays, resource wastages, outsourcing critical tasks to outside parties, expensive financial costs for stationeries, transportations, and inefficient attendance monitoring have created an unproductive workplace. The following list provides some of the drawbacks identified from the existing system:

- Storage space being cluttered over time with paper and copies of physical documents;
- Recurring cost of stationery supplies like packs of paper, ink, pens, and printing;
- Editing or updating on a printed paper is highly messy, leaves the paper in bad shape;
- Collaboration between different departments is limited as information is in physical form which cannot be easily shared or distributed, which shows a lack of data accessibility;
- Paper records and archives holding them are vulnerable to damage if exposed to adverse weather conditions or disasters;
- Error and delays resulting from paperwork-related mishaps;
- The time-consuming cycle of filing, searching, and locating (misplaced) paper documents;
- Paper documentation makes access to sensitive data all too easy, leading to poor data privacy;
- There is no easy way to generate reports and reviews of operations and performance;
- Creating manual invoices takes long overhead time which is also prone to manual error;
- Traditional bookkeeping methods prevent the organization from developing a full picture of its financial performance.

## **2.5. Business Model Rule**

As the system in question is intended for organizations, enterprises, businesses, and companies it coheres with an organizational structure of most business models which is the hierarchical model structure.

A company's chain of command, which usually runs from senior management and executives to general employees, is referred to as a hierarchical structure. In other words, this arrangement is appropriate for organizations with a single leader and a hierarchy of subordinates.

After that, the staff is split into groups, with each group reporting to their respective manager. The department manager receives a briefing from the general manager. As a result, the organization is divided into tiers, with the highest level wielding the most power. Pyramids are often used to represent hierarchical structures. The method, as an organizational business model, refers to the enterprise structure's hierarchical vertical model for decision making and operational intelligence.

The following are the business rules of the system in general.

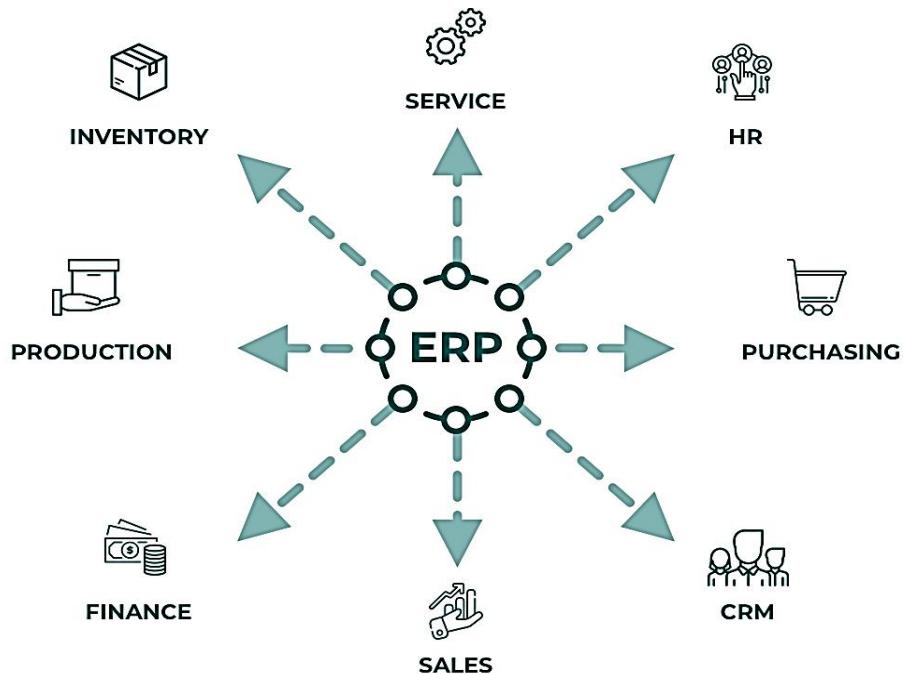
1. The system administrator with the approval and authority of high-level administration of the organization is the one with complete access and rights to the entire system modules.
2. The system administrator is responsible for assigning the proper rights and permissions of the system to all other users of the system based on the particular needs of that user.
3. The human resources management module of the system should operate and perform as per the rules and regulations of the FDRE Labor Proclamation No. 377/2003 and the Federal Civil Servants Proclamation 1064/2017.
4. The accounting/finance module of the system should operate and perform as per the rules and regulations of the Financial Administration Proclamation 1996 of the FDRE.
5. Department Heads such as the Human Resources Administrator and Head of Accounting/Finance should acquire proper credentials from the system administrator to access the system.
6. Users of the system receive access authorization from an immediate upper administration.
7. Employees shall receive credentials from the human resources department after successfully registering in the system as an employee.
8. System access level follows the hierarchical structure of the organization.
9. To operate the system, one requires a certain level of background and experience in IT and basic reading and writing skills of the English language.
10. An internet connection is required to connect multiple branches of the enterprise to headquarters.
11. The system keeps logs of every login, activity, and transaction.
12. The system must provide a backup of the database periodically.

## **2.6. Review of Related Systems**

This section reviews some related systems to the proposed system as well as some design and development approaches.

Until recently, most businesses had disparate organizational operations. Unintegrated systems are only capable of supporting the activities of a single functional area. As a result, an organization will have separate structures to handle manufacturing, distribution, and the supply chain, each with its hardware, software, and data and information processing methods. For example, the purchasing department issued purchase orders without having current stock details.

ERP systems are mainly used by businesses to automate their business processes. Enterprise resource planning software tools deal with resource management, which is as rational as it sounds. Each ERP system must be tailored to a company's particular needs in order to effectively address specific areas of performance that need to be improved.



*Figure 2.1 ERP System Modules (Photo Credit: Vilmate, <https://vilmate.com/wp-content/themes/vilmate/images/2019/12/ERP-application-development-Structure-ERP.jpg> )*

Furthermore, ERP applications must make it easier for these departments to communicate with one another. All solutions would have one thing in common: all system elements, referred to as modules, must be loosely connected in order to integrate with other systems already in use by an organization. As a result, using modules or creating new ones would be part of the process of developing custom ERP applications.

ERP systems collect data in a single, robust database and make it accessible to scalable applications that serve all aspects of a company's value chain across roles, business divisions, and geographic areas (Poston & Grabski, 2001). These systems have become the de facto operating standards for businesses, and they represent standardized, multi-level configurable and scalable solutions that implement best practices and embody a set of assumptions on how businesses work in general. By the use of a data dictionary, ERP systems can share the same data description across all modules.

ERP systems connect a variety of business processes and allow data to flow between them. ERP systems reduce data duplication and provide data integrity with a common source of truth by obtaining an organization's shared transactional data from various sources. Modern ERP systems may be hosted in a public or private cloud, on-premise, or in a range of hybrid scenarios overlapping environments.

D.M. Bahssas and Dr. Adnan M. Albar (2015) concluded that nowadays, the internet has had a significant impact on the lives of businesses and organizations, necessitating the reshaping of business models in order to gain a strategic edge in global manufacturing. With the addition of a modern internet model, Enterprise Resource Planning (ERP) became even more important. The Web-based Object-Oriented Model (WOOM) is a modern ERP model that uses internet components and technologies as object-oriented models for easier, faster implementation and customization. Web-based ERP is a new generation of ERP developed by WOOM. Besides this, WOOM created an ERP framework that is defined in terms of objects that are represented using an object-oriented method and are based on real-world entities like organizations, events, or individual roles. Web-based ERP systems provide internet connectivity and real-time data, resulting in more effective and reliable business operations.

Vilmate Software [7] discusses that when it comes to the company's software requirements, deciding whether to build or purchase is a difficult task. If you go with building, you'll have to figure out how to create web-based ERP software, but if you go with buying, you'll have to figure out what that software might look like. Purchasing an off-the-shelf ERP solution would almost certainly save you money, which is a significant advantage over customizing one. The lack of customization, on the other hand, is the disadvantage of purchasing apps. This drawback eventually begins to outweigh the cost benefits, particularly as the business grows.

As the system is a multi-dimensional with a lot of parts moving around; very deep understanding of the organizational aspect and design approaches must be considered to result in a successful product. Thus, the system is going to be modeled after a thorough requirement analysis of the organization and other system features. It is being approached to align with the organizational goals along with good practices for designing and developing a modular system.

The following section presents a comparative review on some off-the-shelf products of ERP software collected from Capterra [8] – an online marketplace for software and technology vendors and BetterBuys [9] – software review solutions for businesses.

- **Aquilon Software:** Licensing is available on-premise and, in the cloud, and all pricing is dependent on the number of users. The cost of an on-premise license ranges from \$1,500 to \$3,000 per user, depending on the modules chosen. The cloud licensing choice normally ranges from \$75 to \$150 per user per month, depending on the modules chosen.
- **Sage Intacct:** This product is a good fit for large-scale financial management and e-commerce ventures. Clients can add or request extra functionality if required, and the program has a wide range of reporting and accountability features. For a simple annual subscription, pricing starts at \$9,000 per user.
- **Sap:** Sap is obviously the market leader, with over 300,000 clients worldwide. The robust flexibility of Sap's ERP modules – the modules adapt to a variety of business infrastructures – is the key to this resounding success. Sap is also a good option for those who are new to operational management. While this is one of the most effective resources available, it is also one of the most expensive: an annual subscription costs \$420 per user per month.
- **NetSuite:** NetSuite, which is owned by Oracle, is another common brand among large businesses. NetSuite is well-known for its business intelligence modules, as well as its

excellent accounting management. NetSuite's monthly base license costs \$999 plus \$99 per user.

- ***Microsoft Dynamics 365***: Individual modules can be purchased depending on a company's unique business needs. The Sales Professional module, for example, costs \$65 per month per user, while the Sales Enterprise module costs \$95 per month per user.
- ***Workwise***: Workwise ERP is a full-featured, flexible enterprise resource planning (ERP) platform that includes eight suites and over 45 applications built specifically for manufacturers who make-to-order, engineer-to-order, configure-to-order, build-to-stock. It enables users to make data-driven decisions that can reduce turnaround times, improve supply chain management, simplify planning and execution, and improve communication. It's ideal for lean manufacturing operations. It provides affordable, flexible pricing based on the implementation and buying preferences of its customers starting from \$30 per user.
- ***ERPNext*** is an open-source ERP system that can be used by companies in a variety of industries, including manufacturing, distribution, retail, education, healthcare, agriculture, and non-profits. Using a centralized dashboard, administrators can handle accounting, onboarding, payroll, manufacturing, distribution, procurement, and CRM processes. Managers may also use it to schedule projects and keep track of activities in real-time. ERPNext's monthly pricing starts at \$25.00 per feature. A free version is available. A free trial of ERPNext is available.
- ***Odoo***: Customer relationship management (CRM), e-commerce, payroll, billing, inventory management, project management, warehouse management, financial management, manufacturing, and purchasing are all included in Odoo's advanced open-source enterprise resource planning (ERP) program. These modules are designed to interact with one another effectively and seamlessly. Odoo's monthly pricing starts at \$20.00 per customer. A free version is available. Odoo provides a free trial for up to 15 days.

In comparison to the proposed system of this project, there is a long road ahead to develop more modules and build a more comprehensive ERP system over time. In the meantime, the project aims to develop an affordable, customizable, localizable, and easy-to-use ERP system module as a product in which it would be chosen over the systems discussed above as they are very expensive and modeled for different structures of organizations. They are also licensed systems that you have to pay for periodically or per feature. Even the opensource systems discussed are not complete but a demonstration of the entire system to boost sales and pricing.

Most of the systems discussed above have so many features and options, there may be a steeper learning curve for new users. The system being developed in this project's case tends to bring about a cheaper, marketable, and affordable product to Ethiopia with the consideration of a light-weight and shallow learning curve; as well as customizable as needed for other organizations apart from the test-model enterprise with a little overhead. Localization is also one of the design considerations planned to be implemented which this system uniquely provides to the Ethiopian industry and market.

### **3. Chapter Three: The Proposed System**

#### **3.1. Overview**

The proposed system is a web-based system that can manage the organizational day-to-day business activities and resources such as human resources management and accounting. The system will consist of two primary modules with various features. The features are described in chapter one; the features section of this documentation. Most of the system will be an internal system to be accessed and operated by authorized personnel employed by the organization, as it is an enterprise system that handles back-office tasks and operations. The system can manage multiple branches, departments, and sub-departments of an enterprise with a hierarchical decision-making system. The system may have separate sign-in pages for different employees, clients, and administrators.

The employees will have access to the system with privileges and access granted by the upper echelon of the organization based on their position and responsibilities. Job applicants can easily apply for the position they want, and their application will be viewed and vetted by the human resources department all using the system. Clients will have a simple page to place orders to the enterprise. Administrators such as HR managers and Head of Finance will have higher level access than ordinary employees with more access and privilege and a broader view of the business domain. The system administrator/super admin will have access to the entire system and be able to monitor, manage and control every aspect of the system's features and data.

The web application suite would primarily integrate two modules namely the Human Resource Management System and the Finance/Accounting System. Each module would have its particular features that would uphold the objectives of the project.

The proposed system would register the organization where the system would be implemented. The organizational information may consist of the name, departments, branches, locations, license records, and other details about the organization.

The human resources module would register employees of the organization with necessary details, which would then manage these employee records. It shall perform essential HR processes such as promotions, terminations, resignations, transfers and warnings. It shall perform schedule management and attendance monitoring activities. It shall also perform payroll-related activities as well as HR-related reporting activities. It also announces and manages vacancies within the organization, receives job applications from candidates.

The finance module would register the organization's general accounting ledger information with its different accounts. It performs deposit and expense management processes. It shall provide fiscal reporting as well as billing and invoices service for various transactions. It shall also provide interfaces for the above-mentioned process where users can enter, update, delete, view, search, and filter records based on their particular access and needs.

The organization upon implementation of this proposed system would benefit greatly by improving efficiency and performance, enhance the business reporting as well as the business process with automation and improved accessibility and security.

### **3.2. Functional Requirement**

The following section describes the data required and the functional requirements that shall be performed in this proposed system. After the requirement gathering and analysis of those requirements as well as some other related systems, the following functional requirements are specified. These functional requirements include the ongoing daily, weekly, monthly, and annual business operations and transactional functions of the organization. The requirements are listed in their order of priority and importance for the human resources management system module and accounting/finance management module of the System respectively.

#### **Functional Requirements (FR)**

FR-01: The System shall register the organizational information along with its branches, departments, and sub-departments.

##### **a) Human Resources Module Functional Requirements**

FR-02: The System shall allow the recording and management of employee information.

FR-02.1: System shall record an employee's information and login credentials.

FR-02.2: System shall display information about all and individual employees.

FR-02.3: System shall update and delete employee information.

FR-02.4: System shall search for an employee record.

FR-03: The System shall perform core HR (human resource) process.

FR-03.1: System shall perform the transfer of an employee to another department.

FR-03.2: System shall perform the promotion process of an employee.

FR-03.3: System shall perform the resignation process of an employee.

FR-03.4: System shall perform the termination process of an employee.

FR-03.5: System shall allow warnings to be added to an employee record.

FR-03.6: System shall receive and manage complaint actions that are raised.

FR-04: The System shall perform time and attendance tracking operations.

FR-04.1: System shall provide a means to track attendances of employees.

FR-04.2: System shall create, display, and update schedules and work shifts.

FR-04.3: System shall produce a date-wise or monthly attendance timesheet.

FR-04.4: System shall process overtime requests.

FR-04.5: System shall process different leave types for employees.

FR-05: The System shall have the ability to create payroll from underlying data.

FR-05.1: System shall process and record payments made to employees.

FR-05.2: System shall generate payslips for a particular payroll type.

FR-06: The System shall perform vacancy applicant tracking and recruitment operations.

FR-06.1: System shall create and monitor job posts for vacant positions.

FR-06.2: System shall provide a means to receive job application information.

FR-06.3: System shall display all job applicants and their information.

FR-07: The System shall produce consolidated HR reports such as attendance reports, payslips reports, employees' reports, and leave reports.

##### **b) Finance Module Functional Requirements**

FR-08: The System shall allow the recording and management of financial information.

FR-08.1: System shall create records of general ledger accounts of the enterprise.

- FR-08.2: System shall display information about all and individual accounts.
- FR-08.3: System shall allow authorized users to maintain and search accounts.
- FR-09: The System shall allow the recording and management of deposit information for the Account Receivable.
- FR-09.1: The System shall provide the capability to process the payer information for the Accounts Receivable.
- FR-10: The System shall allow the recording and management of expense information for the Accounts Payable.
- FR-10.1: The System shall provide the capability to process the payee information for the Accounts Payable.
- FR-11: The System shall process the transfer of funds information from one account to another.
- FR-12: The System shall display and search all financial transaction information.
- FR-13: The System shall produce invoices for different payments and sales activities.
- FR-14: The System shall produce consolidated Financial reports such as expense reports, income reports, transfer reports, and account statement reports.

### **3.3. Non-functional Requirement**

The following section describes good standards and peripheral requirements intended for the betterment of the proposed system. In addition to the main system modules, the proposed system includes some services and features to accommodate the previous work practices and future modules planned to be developed and integrated into the system. Some of these services and features:

- ✓ Import and export spreadsheets and documents that have been used by the existing system; as transition to a fully computerized system at once will prove to be difficult.
- ✓ Generate reports and summary statements from the data entered and manipulated across the system and departments which will provide a broader view of the business.
- ✓ Development of pluggable and customizable system endpoints to enhance system capabilities and features later in the development stage.
- ✓ Implement standardized and good practices of system development to build a dynamic, flexible and scalable system that can be used not only at the test company EMPDE but also any other organization with the related business model.

After analyzing various features and capabilities of different related systems, the following non-functional requirements are specified in their respective domain.

#### **3.3.1. Design Requirements (DR)**

- DR-01: The System shall have a common relational database core that allows integration of data and transactions between all operational, financial, and other functions within the System.
- DR-02: The System shall have a graphical user interface (GUI) implemented as a Web-based interface with an acclaimed administrative dashboard, navigation, charts, themes, and tools.
- DR-03: The System shall run on PCs that are using Windows 7 or newer operating systems.
- DR-04: The System shall use Microsoft Office 2007 or newer productivity software for any ad hoc data or file exchange.

DR-05: The System shall run on any web browser including newer versions of Firefox, Google Chrome, and Microsoft Edge.

DR-06: The System shall create export files and have the capability to import files between the system and the existing spreadsheets.

DR-07: The System shall have administrator and user security functionality to include:

- a. Setting up a new user
- b. Updating an existing user's information
- c. Restricting user access to certain roles

### **3.3.2. Usability Requirements (UR)**

UR-01: The user should be able to create reports using parameters (such as for specific date or time ranges and be able to customize and filter out various reports).

UR-02: The System shall be available in local languages such as Amharic and Oromiffa.

UR-03: The System should be able to create charts and graphs from underlying data.

UR-04: The System report selects should have limiting filters to reduce the number of records returned and displayed.

UR-05: Reports should be exportable to different desktop tools (including but not limited to Microsoft Excel)

UR-06: The System should have the ability to customize forms and reports with the ability to add and remove user fields and layouts as well as assign constant values.

### **3.3.3. Security Requirements (SR)**

SR-01: Overall system security should be authentication-based and also meet the standard security guidelines as personal and financial data is stored within the system.

SR-02: The System should disable all redirects from the System website and access to any external system or Internet.

SR-03: The System shall ensure that employee application-level security will follow these guidelines:

- a. Must use a permission-based scheme to limit access to application functions and data.
- b. Should provide password management functionality that supports password policies including:
  - i. Password expiration and renewal
  - ii. Password complexity
  - iii. Must provide a way to grant/revoke user permission without affecting the application data.

### **3.3.4. Reliability Requirements (RR)**

The possibility that the System will be able to process work correctly and absolutely without being aborted is referred to as reliability. In the event of a failure, recoverability refers to the ability to restore operation and data.

RR-01: The System should capture audit trails of all changes made to any data such as date and time of change, a user making change.

In the event of a module failure (becomes unusable) for some time the Impact on the enterprise would be as follows:

<b>Length of Time of Outage</b>	<b>Impact on Operations and Productivity</b>
One Hour	Some Impact on processing operations and transactions.
One Day	Medium to Large Impact to process transactions and operations
One Week	Very Large to Catastrophic Impact.

Table 3.1 Outage Time to Impact of Operation Comparison

- RR-02: The Minimum acceptable level of reliability for system modules would be two (2) days. Beyond this timeframe, an organization's ability to conduct business would be significantly affected. Back-up operational processes (relying on manual methods) would be deployed but the results of these processes would need to be re-entered into the system once it again becomes available.
- RR-03: The System should have the means to create a backup of the database at any particular time, which would be used as a recovery point for the database in case of any failure or data loss.

### 3.3.5. Performance Requirements (PR)

- PR-01: The current preference is that accessing any operational or transactional screen and updating data fields should take no more than 3 seconds.
- PR-02: System performance should be measured using up to 100 concurrent users.

## 3.4. System Model

A requirements-driven alignment approach combines business requirements specifications with a model that specifies ERP device capabilities. This necessitates that the system capabilities model reflects the full range of options and features available in the system in a way that allows them to be matched to business requirements. (Soffer et al., 2003)

ERP systems with embedded enterprise models, such as SAP and Baan, reflect their functionality and "best practice" solutions. The model built into an ERP framework may be used to match the system with the specifications. As a result, using the same modeling methodology to define the specifications makes sense. Certain schemes, on the other hand, do not have a business model, while others use various modeling principles to solve different aspects of the organization.

The proposed system is modeled to be scaled up to an ERP system with two primary modules developed for this particular project. The modules are specified above in the functional requirements section. The proposed system tends to bring about a product with the proposed system modules, named '**Tatari**' System.

Tatari (tatarī / ታጥሪ) is an Amharic word that can be translated as hardworking, diligent, or industrious. It was selected as a product or system name as it captures the essence and value of the project and the proposed system, that is to create a productive and efficient work environment, maximizing the productivity of the organization.

The following sections discuss the scenarios, use cases, and various model diagrams to describe the proposed system from different perspectives. It would greatly enhance the analysis, design, and development phases of the project.

### **3.4.1. Scenarios**

To describe possible scenarios and interactions between different users and the proposed system, the following actors are used in the scenario descriptive paragraphs:

- Abel is the System Administrator employed by the organization in the IT Department.
- Hana is the HR Administrator that oversees the fundamental human resources process.
- Daniel is an assistant human resources administrator that aids the HR administrator Hana.
- Fekadu is the Head of Finance that manages the accounting operations of the organization.
- Eleni is an employee of the organization that works in the manufacturing department.
- Amir is an employee of the organization that works at the distribution department.
- Dawit is an accountant that works in the finance department of the organization.
- Solomon is the General Manager of the organization that works in Administration.
- Rebeca is the candidate that has applied for a position at the organization.
- Desta is an experienced electrician that is looking for a job that suits his position.
- Hailu is an employee that works in the general services department of the organization.
- Abebe is an employee of the organization working as a security guard.
- Hawi is a junior accountant at the organization who subs as a cashier.

The scenarios are written in as per the guidelines of a descriptive paragraph of a non-technical manner for the simple understanding of any stakeholder or individual. In all the subsequent scenarios discussed below, the following preliminary assumptions are made:

- The user has access to a computer device; laptop or desktop (preferably).
- The user is on-premise of the organization (unless otherwise specified).
- The user has access to the enterprise network, or Internet connection (for branch organizations connecting to the central server via the Internet).
- The user is aware of the enterprise portal in which the Tatari System resides.
- The user as an employee has proper log-in credentials to the system with specific permissions and access levels.
- The user has logged in to the system with said credentials unless otherwise indicated.

#### ***Scenario 1: Register a new Employee***

Hana chooses to enter the employee registration page to register the new applicant, Rebeca. Hana is shown the employee registration page which consists of a form with different fields. Hana enters the employee information such as name, gender, date of birth, address, department, designation, etc. by asking Rebeca and referring to the job application documents. System validates the employee information. System creates a record of the new employee. System displays the new employee record among the list of employees.

#### ***Scenario 2: Transfer an Employee***

Hana chooses to enter the employee transfer page to transfer the employee Amir from distribution to the customer service department. Hana is shown the transfers page which consists of a form for transfers and a list of previous transfers. Hana is prompted to enter the employee and department

details of the transfer. Hana enters the required information. The system verifies the entered information. Systems create a record for the transfer. System displays the new transfer record among the list of transfers.

### ***Scenario 3: Promote an Employee***

Hana chooses to enter the employee promotion page to promote the employee Eleni to Head of Manufacturing. Hana is shown a list of previous promotions and an option to enter a new one. Hana chooses to add a new promotion; She is then prompted to fill out a form with the details of the promotion. Hana enters the required information for the promotion such as designation, promotion position and title and other details. System verifies the information provided. System creates a record of the promotion. System displays the new promotion record among the list of promotions.

### ***Scenario 4: Submit a warning***

Hana is presented with a complaint with proof against the employee Dawit and Hana after consideration of the facts and evidence, chooses to enter the Warnings page on the system. Hana then selects the option to add a new warning. She is then presented with a form to be filled out about the employee, the subject, and other details; which she fills out along with the proper evidence. System validates that information. System creates a record of the warning. System displays the new warning among the list of warnings.

### ***Scenario 5: Terminate an Employee***

Hana as an HR Administrator is faced with the difficult decision of terminating one of her employees - Hailu. She has discussed with her colleagues and decided that Hailu's behavior is unacceptable. Hana then chooses to enter the employee termination page. System has displayed a list of past terminations and an option to add a new termination. Hana chooses to add new termination. System displays a form to be filled out for a proper termination of an employee. Hana enters the details of the employee and the termination reason. System validates the information. System creates a record of that particular termination. System displays the recent termination on the list of past terminations.

### ***Scenario 6: Clock-in/Clock-out an attendance***

Fekadu is a good employee with an exemplary record. He arrived at work early and wants to clock in for the day. He chooses to enter the attendance page. System provides the right details and options to clock-in for the day. Fekadu clocks in, and submits the initial attendance record. At the end of his shift/day, he again goes to the same page and chooses the clock-out option; submitting the final attendance record for the day. System creates an attendance record for the particular employee. System shows an attendance summary.

### ***Scenario 7: Employee Resigning***

Abebe is a disgruntled employee of the organization that recently came across a better-paying position at another company. Abebe wanted to resign from the organization, he goes to see the HR administrator Hana. He explains his situation and informs her about his decision to resign. Hana

then logs into the system. She chooses to enter the resignation page. System shows past resignations and an option to enter a new one. She chooses a new resignation operation and fills out a form to process a resignation. She asks for some details of the resignation and enters it onto the form and submits it. System validates the information. System creates a record of the resignation. System shows the newly added resignation among the list of resignations.

#### ***Scenario 8: Update an Employee Record***

Amir who works at the distribution department wanted to update his employee record. He wanted to change his telephone number and home address on file. He goes and sees the assistant HR administrator Daniel. He explains his need to update his employee information. Daniel then logs onto the system and chooses to enter the employee records page. He then selects Amir's record and chooses to edit the record. Daniel asks for the new telephone number and home address and fills out the record with the new information. Daniel then updates the employee record. A similar scenario can be considered an employee himself/herself logs into the system and update their record without contacting the human resources department.

#### ***Scenario 9: Create a night-shift schedule***

The General Manager Solomon after rigorous discussion with other departments decides to create a night shift, given the market demand and performance. Solomon browses the system and logs in. Solomon chooses to enter the Manage Schedule option. Solomon then chooses to add a new shift to the work schedule. System displays a form to create a new shift. Solomon enters the details of the new night shift with the proper weekdays and hours. Solomon then saves the form. System validates the information. System creates the new night-shift schedule record. System displays the new shift among the list of existing work schedules.

#### ***Scenario 10: Generate Monthly Timesheet***

Hana wanted to generate a monthly timesheet to submit to upper management. Hana browses the Attendance page. Hana chooses to enter the Monthly Timesheet option on the Attendance page. System displays attendance records as a table and options to export them as various reports. Hana fills out the parameters for the report and generates the monthly timesheet as a PDF. System generates the timesheet report as PDF. Hana then prints the PDF and shares it with upper management.

#### ***Scenario 11: Request Overtime for Sunday***

Dawit as an accountant faces long hours at the end of the month. Dawit worked overtime on Sunday last week. He wanted to request overtime for that instance. Dawit has logged into the system and chooses to enter the Request Overtime option. System provides a form to be filled out with the details of the overtime. Dawit fills out the form with the overtime details and saves it. System validates the information and creates a record for that overtime request. System provides the overtime requests to the HR administrator Hana. Hana then approves the requests for overtime payment. System processes the overtime and updates the payroll.

### ***Scenario 12: Request Sick Leave***

Eleni was feeling sick and saw a doctor. She was directed to take a few days off to rest up. She then wanted to ask for sick leave. Eleni calls the HR administrator Hana and explains the situation. Hana then enters the Leaves option on the system. System provides a list of leaves and an option to add a new one. Hana selects add new leave. System provides a leave form. Hana enters the details and reason for the leave and saves the form. System validates the information. System creates a new leave record. System displays the new sick leave among the list of leaves.

### ***Scenario 13: Generate Payroll and Payslips***

Hana as an HR administrator has the responsibility to generate payroll at the end of the month for the employees' salary payment. Hana works closely in communication with Finance to generate and approve payroll. Hana chooses to enter the payroll option. System shows the payroll status of employees for the month. Hana then approves the salary payment along with overtime, deductions, and alike. Hana can choose to submit the payroll either individually or in bulk. After successful generation of payroll, the system provides a payslips generation option. Hana then chooses to generate payslips for the payments on the payroll. System provides the latest status of the payroll and payslips.

### ***Scenario 14: Announce Vacancy***

The organization has a vacant electrician position. Hana wanted to announce a vacancy for the position on the organization's public website. Hana chooses to enter the Job Posts option. Hana then selects add new job post. System provides a form to be filled out to create a job post. Hana fills out the details of the electrician vacancy position and submits the form. System validates the given information. System creates the job post record. System displays the job post among the list of job posts. System displays the job post on the organizational public webpage.

### ***Scenario 15: Apply for a Job***

Desta came across the electrician vacancy post on the organization's public webpage. Desta then chooses to apply for the position. System provides a form for the job application. Desta fills out the form and provides proper documentation with the application. He then submits the application form. System validates the given information. System creates a record for the job application received.

### ***Scenario 16: Recruit Candidates***

Hana has waited a few days after announcing the electrician vacancy and wanted to see the job applications received. She chooses to enter the recruitment page on the system. She chooses the Candidates option. System displays a list of candidates/applicants with their submitted job applications. She then sees the details of the applications received for the position which would assist her in deciding for hiring the applicant.

### ***Scenario 17: Generate Attendance Report***

Hana wanted to generate a monthly timesheet for an employee to present as evidence for a litigation case against the company. She chooses to enter the Generate HR reports option. System displays a form to be filled out that is used to generate the report. She fills out the form with the specific details she needs for her case. She chooses to generate an attendance report with the required details. System generates the attendance report. System provides options of different formats to export the report. She chooses to print the generated attendance report.

### ***Scenario 18: Create a manufacturing ledger account***

As business and market share rises, management decides to create a new and separate account for the manufacturing department. This decision is passed down to the finance department. Head of Finance Fekadu receives this order and executes the following. He chooses to enter the Accounts option on the system. System shows a list of existing accounts. Fekadu chooses to add a new account as per his orders. System provides a form to create a new ledger account. Fekadu fills out the form with the necessary details of the manufacturing ledger account and saves the form. System validates the provided information. System displays the new manufacturing ledger account record among the list of other accounts.

### ***Scenario 19: Create a cashier profile***

The organization needs a cashier at one of their branches. Hawi was ordered to work at the outpost as a cashier for some time. Head of Finance Fekadu wants to create a payee profile for Hawi as she is going to be working as a cashier. Fekadu chooses to enter the Payee page. System shows a form to be filled out to create a payee profile. Fekadu enters the Hawi's details so that she can successfully process the operation. System validates the given information. System shows the new payee/cashier profile record of Hawi on the active payee list.

### ***Scenario 20: Create a client profile***

The organization has a repeat customer and the Head of finance wanted to create a client profile for this customer. Fekadu chooses to enter the Payer page. System shows a form to be filled out to create a payer profile. Fekadu enters the details of the client so that they can successfully process the operation. System validates the given information. System shows the new payer/client profile record on the payer list.

### ***Scenario 21: Process a budget deposit***

A new fiscal year is about to start, and the organization received an approved budget from the board. The Head of finance is tasked with processing the new budget as a deposit to the organization. Fekadu chooses to enter the Deposit's page. Fekadu selects the add new deposit option. System displays a form necessary to process a deposit. Fekadu enters the details of the approved budget and submits it as a deposit. System validates the provided information. System shows the new budget deposit on the deposit list.

### ***Scenario 22: Process a fuel expense***

Transportation is crucial for distribution and fuel is basically what runs it. Fuel expense is usually processed daily. It is one of the responsibilities of Accountant Dawit to process expenses. Dawit chooses to enter the Expense page. Dawit chooses the add new expense option to add the fuel expense for today. System provides a form to be filled with the expense details. Dawit fills the form with the details of the fuel expense. System validates the given information. System shows the new fuel expense among the list of previous expense records.

### ***Scenario 23: Transfer funds from reserve to manufacturing account***

The manufacturing department has used up most of their funds for the month and requested an extra operational fund from the reserved funds. With the approval from higher management, Fekadu the Head of finance is tasked with transferring funds from the reserve to the manufacturing account. Fekadu chooses to enter the Transfers page. System shows a form with fields required for transfer. Fekadu enters the details of the reserve funds and the manufacturing account and submits a transfer for the approved amount. System validates the information. System shows the transfer records.

### ***Scenario 24: View the latest transactions***

Fekadu has been away for a few days and wanted to see the latest transactions that have occurred in his absence. Fekadu chooses to enter the Transactions page. System shows a list of all transactions committed recently. Fekadu observes the recent transactions and their details.

### ***Scenario 25: Create an Invoice for an item sold***

Dawit processes items sold in the day and wanted to create invoices for those payments. Dawit chooses to enter the Invoices option. Dawit selects to add a new invoice. System shows an invoice form with details of payment like amount, item, tax, unit price, discount, and date. Dawit fills out the form and submits it. System validates the given information. System shows the invoice record which Dawit prints out keeps for his manual record.

### ***Scenario 26: Generate an expense report***

At the end of every month, the Head of Finance Fekadu presents an expense report for upper management. Fekadu wanted to generate an expense report for the month. Fekadu chooses to enter the Fiscal Reports page. System shows the different kinds of reports that can be generated. Fekadu chooses the expense report. System displays a form to be filled out as parameters for the report. Fekadu enters the details of the current month. System validates the given information. System generates the expense report for the month. Fekadu saves the report as a PDF with plans to use it later.

### **3.4.2. Use case Model**

This particular section elaborates on the use case diagram and cases related to the proposed system and its corresponding modules. To show the interaction between the system and actors, the following use cases have been utilized. The following use cases are grouped separately as per module (Human Resources and Finance respectively) but viewed as a unit after being integrated into a single system.

#### ***Human Resources Module Use case***

- Register Organization
  - Register Employee
  - Transfer Employee
  - Promote Employee
  - Terminate Employee
  - Process Resignation
  - Process Warning
  - Manage Employee Record
  - Submit Attendance
  - View Profile
  - Request Overtime
  - Manage Schedule
  - Generate Timesheet
  - Process Overtime
  - Process Leave
  - Process Payroll
  - Generate Payslips
  - Create Job posts
  - Submit Job Applications
  - Manage Job Applications
  - Generate HR Reports
- 

#### ***Finance Module Use case***

- Create Ledger Account
- Add Payee Information
- Add Payer Information
- Process Expense
- Transfer Funds
- View Transactions
- Manage Ledger Account
- Process Deposit
- Create Invoice
- Manage Invoice
- Generate Fiscal Reports

To depict the entities and people interacting with the system, the following actors have been identified. The actual actors which include most of the employees of the organization at different levels of the business, clients, and other stakeholders are numerous; so, the use case is only specified for major actors for the sake of simplicity. The remaining actors are implicitly indicated and covered with the Roles and Permissions feature and approach which entails assigning a certain functionality/use case to a particular user/actor that absolves it from being explicitly represented.

- Human Resources Officer-1
  - Description: This user works in the Human Resources Department of the organization as an HR officer. It uses the human resources system module in the capacity allowed by the higher administration. It interacts with the system to perform his/her daily tasks which are common HR processes.
  - Activity: Register an employee; Promote, Transfer, Terminate an Employee; Process Resignation, Warnings, Overtime, Leave
- Human Resources Administrator
  - Description: This user manages the entire Human Resources Department of the organization. It uses the human resources system module to the maximum capacity and permission allowed. It interacts with the system to perform daily core HR tasks as well as some other additional fundamental HR operations and reporting.
  - Activity: Register an employee; Promote, Transfer, Terminate an Employee; Process Resignation, Warnings, Overtime, Leave; Manage Employee Records, generate timesheet, Manage Schedule, Process payroll, generate payslips, Create Job posts, Manage Job Applications, Generate HR reports
- Accountant-1
  - Description: This user works in the Finance/Accounting department of the organization. An accountant can be of different levels such as a Junior or Senior accountant. It uses the finance system module in the capacity allowed by the higher administration. It interacts with the system to perform daily accounting tasks.
  - Activity: Add payee and payer information, Process deposit and expense, create invoices.
- Head of Finance/Accounting
  - Description: This user oversees the whole accounting department of the enterprise. It uses the finance system module to the maximum capacity and permission allowed. It interacts with the system to perform daily accounting operations as well as some other fundamental financial tasks and reporting.
  - Activity: Add payee and payer information, Process deposit and expense, create invoices, Create and maintain ledger accounts, Transfer funds, view transactions, Manage Invoices, Generate Fiscal reports.
- System Administrator
  - Description: This is a superuser of the system with the highest-level administrative access to the system and its data. It manages and maintains the entire system. This user has more technical skills in the field and possibly works in the IT department of the organization, given proper clearance and access by the administration.

- Activity: Maintain system, view overall system logs and data, Manage organizational and departmental detail, Grant and Revoke access to all employees and users with roles and permission assignment scheme.
- Employee-1
  - Description: This user is an employee of the organization working at a particular department which interacts with the system in a very small manner. As the department that he/she works on doesn't have a module of its own. It initiates some system functionalities like overtime, resignation, leaves but is not considered to be acting as the actor.
  - Activity: Submit attendance, view employee profile, view organization structure
- Candidate-1
  - Description: This user is a job applicant that applied for a position at the organization. It interacts with the system to apply for a vacancy and to be considered for the recruitment process.
  - Activity: Apply for a vacancy

The following use case diagrams illustrates the general interaction of the actors with the system and its use cases. It is presented in the following section of this chapter.

### 3.4.2.1. Use case Diagram

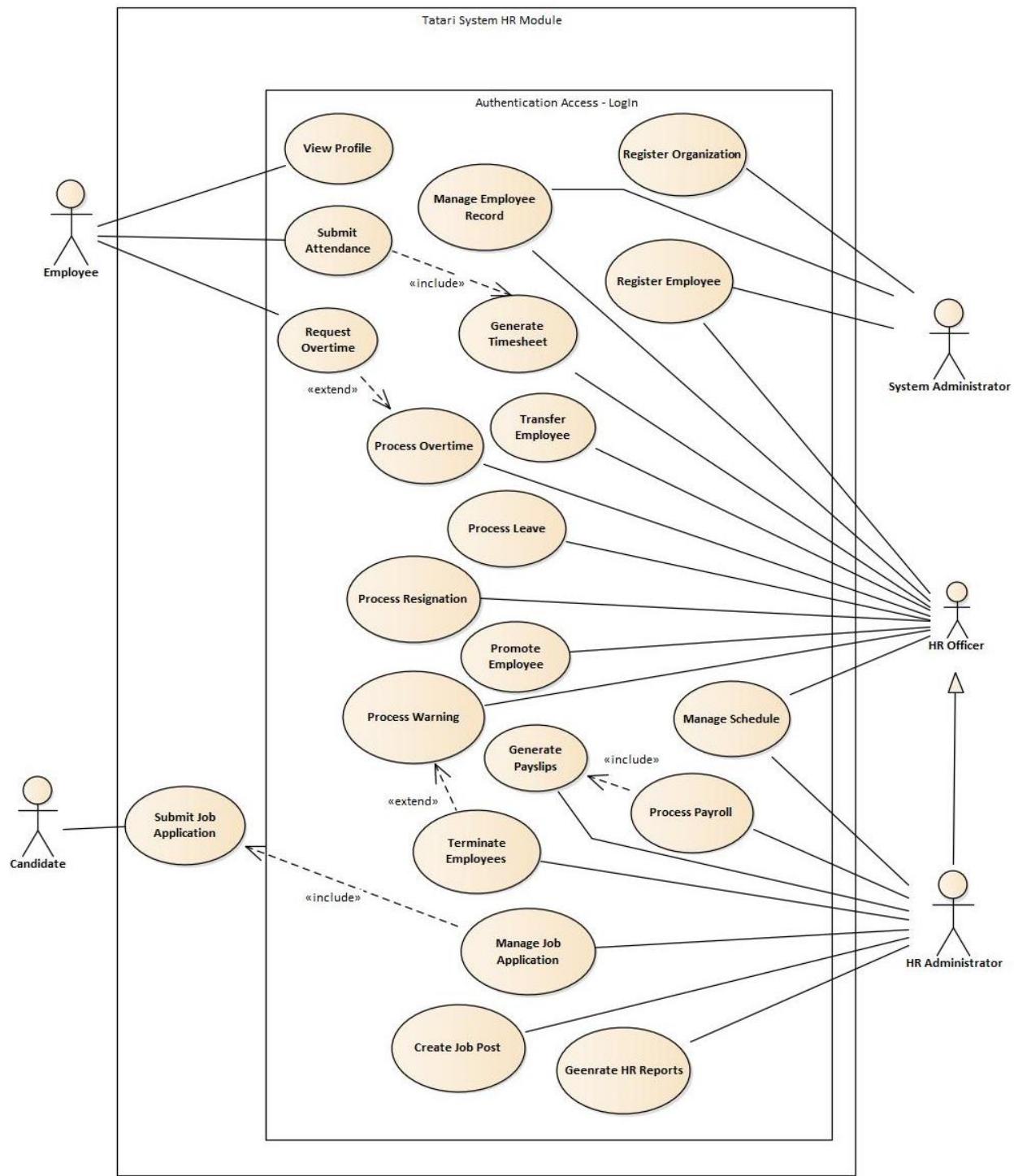


Figure 3.1 Tatari System Use case 1 HR module

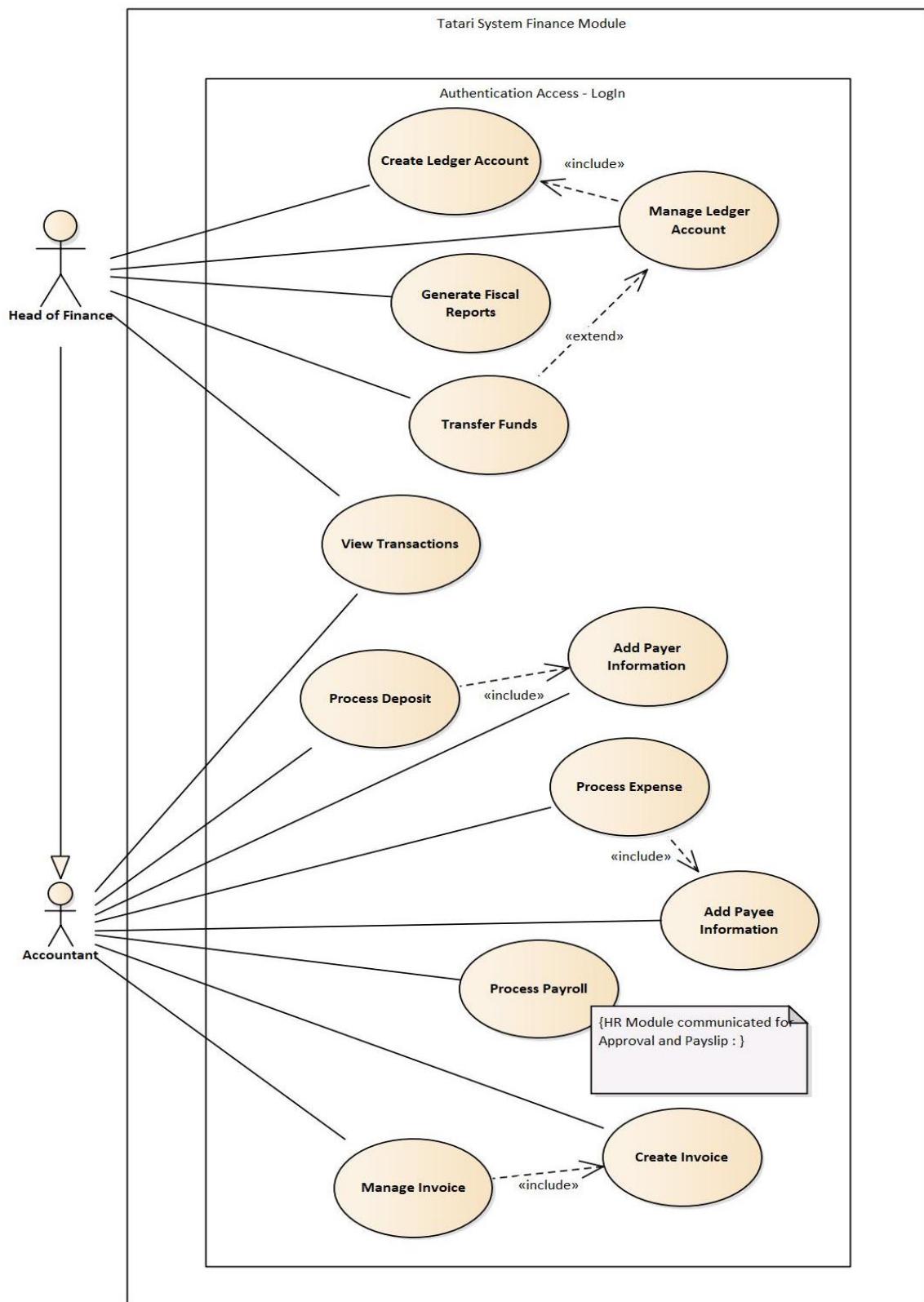


Figure 3.2 Tatari System Use case 2 Finance module

### 3.4.2.2. Use case Description

A comprehensive explanation of the use case diagram and a step-by-step interaction between the system and the actors is presented in this section along with any other required details.

Use case ID: 1      Functional Requirement ID: FR-01	
<b>Use case name</b>	Register Organization
<b>Participating Actor</b>	System Administrator (Admin)
<b>Description</b>	System administrator creates an organizational profile upon which the entire system will be based upon.
<b>Precondition</b>	<ul style="list-style-type: none"> <li>• System administrator has access to a computer device; laptop or desktop (preferably).</li> <li>• System administrator has access to the enterprise network, or Internet connection (for branch organizations via the Internet).</li> <li>• System administrator is aware of the enterprise portal in which the Tatari System resides.</li> <li>• System administrator has proper log-in credentials to the system with the highest access level.</li> </ul>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. System administrator browses to the Tatari system enterprise portal.</li> <li>2. System administrator logs in to the system by submitting proper credentials.</li> <li>3. A system dashboard is shown with general statistics, sidebar and navigation.</li> <li>4. Admin clicks on the organization option.</li> <li>5. System displays organization page.</li> <li>6. Admin clicks on Add New Organization.</li> <li>7. System displays a form to be filled out to create an organization profile.</li> <li>8. Admin fills out the form with details about the organization like name, location, and departments.</li> <li>9. Admin click the save button.</li> <li>10. System validates the given information.</li> <li>11. System displays the organization on the organization page.</li> </ol>
<b>Postcondition</b>	An organizational profile/account is created.
<b>Alternate Flow of events</b>	<ul style="list-style-type: none"> <li>• Incorrect login credentials display login error.</li> <li>• If improper or empty required fields are found on the form, system displays an error message.</li> <li>• The organization might already be registered.</li> </ul>
<b>System Derives</b>	Update and delete organizational information from the database.
<b>Business Rule</b>	1
<b>Frequency of Use</b>	Once at system installation and updated as necessary.

Table 3.2 Register Organization Use case description

<b>Use case ID:</b> 2	<b>Functional Requirement ID:</b>	<b>FR-02.1</b>
<b>Use case name</b>	Register Employee	
<b>Participating Actor</b>	Human Resources Administrator (HR Admin)   HR Officer-1	
<b>Description</b>	HR Admin creates a profile/account for an employee which the employee can use to log in and perform tasks.	
<b>Precondition</b>	<ul style="list-style-type: none"> <li>• HR Admin has access to a computer device; laptop or desktop.</li> <li>• HR Admin has access to the enterprise network, or Internet connection (for branch organizations via the Internet).</li> <li>• HR Admin is aware of the enterprise portal in which the Tatari System resides.</li> <li>• HR Admin has proper log-in credentials to the system with the proper permission and access level of the HR module.</li> </ul>	
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. HR Admin browses to the Tatari system enterprise portal.</li> <li>2. HR Admin logs in to the system by submitting proper credentials.</li> <li>3. A system dashboard of the human resources module is shown with general statistics, sidebar and navigation.</li> <li>4. HR Admin clicks on the employee's option.</li> <li>5. System displays employee's page.</li> <li>6. HR Admin clicks on Add New Employee.</li> <li>7. System displays a form to be filled out to create an employee profile.</li> <li>8. Admin fills out the form with details about the employee like name, gender, date of birth, department, contact information.</li> <li>9. HR Admin clicks the save button.</li> <li>10. System validates the given information.</li> <li>11. System displays the employee record on the employee list page.</li> </ol>	
<b>Postcondition</b>	Employee account and record are created.	
<b>Alternate Flow of events</b>	<ul style="list-style-type: none"> <li>• Incorrect login credentials display login error.</li> <li>• If improper or empty required fields are found on the form, system displays an error message.</li> <li>• Employee might already be registered.</li> </ul>	
<b>System derives</b>	Manage employee records (FR-02.2, FR-02.3, FR-02.4) Search, update and delete operations of employee records are covered by system.	
<b>Business Rule</b>	3	
<b>Frequency of Use</b>	Depends on the number of employees and is updated as necessary.	

Table 3.3 Register Employee Use case description

<b>Use case ID:</b> 3	<b>Functional Requirement ID:</b>	<b>FR-03.1</b>
<b>Use case name</b>	Transfer Employee	
<b>Participating Actor</b>	Human Resources Administrator (HR Admin)   HR Officer-1	

<b>Description</b>	HR Admin transfers an employee from branch to branch or from one department to another department of the organization.
<b>Precondition</b>	<ul style="list-style-type: none"> <li>• HR Admin has access to a computer device; laptop or desktop.</li> <li>• HR Admin has access to the enterprise network, or Internet connection (for branch organizations via the Internet).</li> <li>• HR Admin is aware of the enterprise portal in which the Tatari System resides.</li> <li>• HR Admin has proper log-in credentials to the system with the proper permission and access level of the HR module.</li> </ul>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. HR Admin browses to the Tatari system enterprise portal.</li> <li>2. HR Admin logs in to the system by submitting proper credentials.</li> <li>3. A system dashboard of the human resources module is shown with general statistics, sidebar and navigation.</li> <li>4. HR Admin clicks on the employee's transfer option.</li> <li>5. System displays employee's transfer page.</li> <li>6. HR Admin clicks on Add new.</li> <li>7. System displays a form to be filled out to transfer an employee.</li> <li>8. Admin fills out the form with details about the transfer like name, department, date and location.</li> <li>9. HR Admin clicks the save button.</li> <li>10. System validates the given information.</li> <li>11. System displays the transfer on the employee transfer list.</li> </ol>
<b>Postcondition</b>	Transfer record for an employee is created.
<b>Alternate Flow of events</b>	<ul style="list-style-type: none"> <li>• Incorrect login credentials display login error.</li> <li>• If improper or empty required fields are found on the form, system displays an error message.</li> <li>• Employee might already be transferred.</li> </ul>
<b>System derives</b>	Search, update and delete operations of the transfer record are covered by the system.
<b>Business Rule</b>	3
<b>Frequency of Use</b>	Depends on the workload and transfer need of the organization.

Table 3.4 Transfer Employee Use case description

Use case ID: 4	Functional Requirement ID:	FR-03.2
<b>Use case name</b>	Promote an Employee	
<b>Participating Actor</b>	Human Resources Administrator (HR Admin)   HR Officer-1	
<b>Description</b>	HR Officer promotes an employee to a higher position based on direction from the HR admin.	

<b>Precondition</b>	<ul style="list-style-type: none"> <li>• HR Officer has access to a computer device; laptop or desktop.</li> <li>• HR Officer has access to the enterprise network, or Internet connection (for branch organizations via the Internet).</li> <li>• HR Officer is aware of the enterprise portal in which the Tatari System resides.</li> <li>• HR Officer has proper log-in credentials to the system with the proper permission and access level of the HR module.</li> </ul>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. HR Officer browses to the Tatari system enterprise portal.</li> <li>2. HR Officer logs in to the system by submitting proper credentials.</li> <li>3. A system dashboard of human resources module is shown with general statistics, sidebar and navigation.</li> <li>4. HR Officer clicks on the employee's promotion option.</li> <li>5. System displays employee's promotion page.</li> <li>6. HR Officer clicks on Add new.</li> <li>7. System displays a form to be filled out to promote an employee.</li> <li>8. HR Officer fills out the form with details about the transfer like name, designation, date and title.</li> <li>9. HR Officer clicks save button.</li> <li>10. System validates the given information.</li> <li>11. System displays the promotion on the employee promotion list.</li> </ol>
<b>Postcondition</b>	Promotion record of an employee is created.
<b>Alternate Flow of events</b>	<ul style="list-style-type: none"> <li>• Incorrect login credentials display login error.</li> <li>• If improper or empty required fields are found on the form, system displays an error message.</li> <li>• Employee might already be promoted.</li> </ul>
<b>System derives</b>	Search, update and delete operations of the promotion record are covered by the system.
<b>Business Rule</b>	3
<b>Frequency of Use</b>	Depends on performance and promotion need of the organization.

Table 3.5 Promote Employee Use case description

Use case ID: 5	Functional Requirement ID:	FR-03.4
<b>Use case name</b>	Terminate Employee	
<b>Participating Actor</b>	Human Resources Administrator (HR Admin)	
<b>Description</b>	HR Admin terminates an employee and renders his/her credentials inactive.	

<b>Precondition</b>	<ul style="list-style-type: none"> <li>• HR Admin has access to a computer device; laptop or desktop.</li> <li>• HR Admin has access to the enterprise network, or Internet connection (for branch organizations via Internet).</li> <li>• HR Admin is aware of the enterprise portal in which the Tatari System resides.</li> <li>• HR Admin has proper log-in credentials to the system with the proper permission and access level of the HR module.</li> </ul>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. HR Admin browses to the Tatari system enterprise portal.</li> <li>2. HR Admin logs in to the system by submitting proper credentials.</li> <li>3. A system dashboard of human resources module is shown with general statistics, sidebar and navigation.</li> <li>4. HR Admin clicks on the employee's termination option.</li> <li>5. System displays employee's termination page.</li> <li>6. HR Admin clicks on Add new.</li> <li>7. System displays a form to be filled out to terminate an employee.</li> <li>8. Admin fills out the form with details about the termination like name, department, and date.</li> <li>9. HR Admin clicks save button.</li> <li>10. System validates the given information.</li> <li>11. System displays the termination on the employee termination list.</li> </ol>
<b>Postcondition</b>	Termination record of an employee is created.
<b>Alternate Flow of events</b>	<ul style="list-style-type: none"> <li>• Incorrect login credentials display login error.</li> <li>• If improper or empty required fields are found on the form, system displays an error message.</li> <li>• Employee might already be terminated.</li> </ul>
<b>System derives</b>	Search, update and delete operations of the termination record are covered by the system.
<b>Business Rule</b>	3
<b>Frequency of Use</b>	Depends on the performance, character, and termination need of the company.

Table 3.6 Terminate Employee Use case description

Use case ID: 6	Functional Requirement ID:	FR-03.3
<b>Use case name</b>	Process Resignation	
<b>Actor</b>	Human Resources Administrator (HR Admin)	
<b>Description</b>	HR Admin processes the resignation request of an employee and renders the employee's account inactive.	

<b>Precondition</b>	<ul style="list-style-type: none"> <li>• HR Admin has access to a computer device; laptop or desktop.</li> <li>• HR Admin has access to the enterprise network, or Internet connection (for branch organizations via Internet).</li> <li>• HR Admin is aware of the enterprise portal in which the Tatari System resides.</li> <li>• HR Admin has proper log-in credentials to the system with the proper permission and access level of the HR module.</li> </ul>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. HR Admin browses to the Tatari system enterprise portal.</li> <li>2. HR Admin logs in to the system by submitting proper credentials.</li> <li>3. A system dashboard of human resources module is shown with general statistics, sidebar and navigation.</li> <li>4. HR Admin clicks on the employee's resignation option.</li> <li>5. System displays employee's resignation page.</li> <li>6. HR Admin clicks on Add new.</li> <li>7. System displays a form to be filled out to resign an employee.</li> <li>8. Admin fills out the form with details about the resignation like name, department, and date.</li> <li>9. HR Admin clicks the save button.</li> <li>10. System validates the given information.</li> <li>11. System displays the resignation on the employee resignation list.</li> </ol>
<b>Postcondition</b>	Resignation record of the employee is created.
<b>Alternate Flow of events</b>	<ul style="list-style-type: none"> <li>• Incorrect login credentials display login error.</li> <li>• If improper or empty required fields are found on the form, system displays an error message.</li> <li>• Employee might already resign.</li> </ul>
<b>System Derives</b>	Search, update and delete operations of the resignation and exit record are covered by the system.
<b>Business Rule</b>	3
<b>Frequency of Use</b>	Depends on the employee's need for resignation and Organizational factors

Table 3.7 Process Resignation Use case description

Use case ID: 7	Functional Requirement ID:	FR-03.5   FR-03.6
Use case name	Process Warning	
Participating Actor	Human Resources Administrator (HR Admin)   HR Officer-1	
Description	HR Admin receives complaints and processes warning against an employee to be included put on the Employee record.	

<b>Precondition</b>	<ul style="list-style-type: none"> <li>• HR Admin has access to a computer device; laptop or desktop.</li> <li>• HR Admin has access to the enterprise network, or Internet connection (for branch organizations via the Internet).</li> <li>• HR Admin is aware of the enterprise portal in which the Tatari System resides.</li> <li>• HR Admin has proper log-in credentials to the system with the proper permission and access level of the HR module.</li> </ul>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. HR Admin browses to the Tatari system enterprise portal.</li> <li>2. HR Admin logs in to the system by submitting proper credentials.</li> <li>3. A system dashboard of human resources module is shown with general statistics, sidebar and navigation.</li> <li>4. HR Admin clicks on the employee's warning option.</li> <li>5. System displays employee's warning or complaint page.</li> <li>6. HR Admin clicks on Add new.</li> <li>7. System displays a form to be filled out to warn an employee or submit a complaint.</li> <li>8. Admin fills out the form with details about the complaint or warnings like name, subject, and date.</li> <li>9. HR Admin clicks save button.</li> <li>10. System validates the given information.</li> <li>11. System displays the warning or complaint among the list of warnings or complaints.</li> </ol>
<b>Postcondition</b>	A complaint is successfully submitted into the system or a warning has been added to the employee record.
<b>Alternate Flow of events</b>	<ul style="list-style-type: none"> <li>• Incorrect login credentials display login error.</li> <li>• If improper or empty required fields are found on the form, system displays an error message.</li> </ul>
<b>System derives</b>	Search, update and delete operations of the warning or complaint record are covered by the system.
<b>Business Rule</b>	3
<b>Frequency of Use</b>	Depends on the organizational factors and employee performance.

Table 3.8 Process Warning Use case description

Use case ID:	8	Functional Requirement ID:	FR-04.1
<b>Use case name</b>	Submit Attendance		
<b>Participating Actor</b>	Employee-1		

<b>Description</b>	An employee of the organization marks the attendance by clocking in and clocking out on the system. Or HR Officer imports a spreadsheet of the attendance record into the system.
<b>Precondition</b>	<ul style="list-style-type: none"> <li>Employee-1 has access to a device; laptop or mobile.</li> <li>Employee-1 has access to the enterprise network.</li> <li>Employee-1 is aware of the enterprise portal in which the Tatari System resides.</li> <li>Employee-1 has proper log-in credentials to the system with the proper permission.</li> </ul>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>Employee-1 browses to the Tatari system enterprise portal.</li> <li>Employee-1 logs in to the system by submitting proper credentials.</li> <li>An employee dashboard is displayed with some information, sidebar and attendance submission functionality.</li> <li>Employee clicks on the attendance marker to clock-in at the beginning of the shift and clock-out at the end of shift.</li> <li>System validates the given information.</li> <li>System adds the attendance record to the timesheet.</li> <li>System notifies of either a clock-in or clock-out.</li> <li>System displays an attendance overview for Employee-1.</li> </ol>
<b>Postcondition</b>	An attendance record with clock-in and clock-out time is recorded and added to the timesheet of the employee.
<b>Alternate Flow of events</b>	<ul style="list-style-type: none"> <li>Incorrect login credentials display login error.</li> <li>Double clock-in and clock-out on the attendance marker.</li> <li>For importing bulk attendance from a spreadsheet, file import errors might arise.</li> </ul>
<b>Includes</b>	Update, search and delete of attendance records are covered by the system from HR Administrator side. Importing bulk attendance records from a spreadsheet is also covered from an HR Administrator's scope.
<b>Business Rule</b>	3, 8
<b>Frequency of Use</b>	Daily for employee attendance, as necessary by HR Admin to import collected records over some time.

Table 3.9 Submit Attendance Use case description

Use case ID: 9 Functional Requirement ID:		FR-04.2
<b>Use case name</b>	Manage Schedule	
<b>Actors</b>	Human Resources Administrator (HR Admin)   HR Officer-1	
<b>Description</b>	A working shift schedule and timetable for the weekdays and weekends is set to be used as a guide for employee attendance. Also, day-offs such as national holidays are introduced for attendance exemption.	

<b>Precondition</b>	<ul style="list-style-type: none"> <li>• HR Admin has access to a computer device; laptop or desktop.</li> <li>• HR Admin has access to the enterprise network, or Internet connection (for branch organizations via the Internet).</li> <li>• HR Admin is aware of the enterprise portal in which the Tatari System resides.</li> <li>• HR Admin has proper log-in credentials to the system with the proper permission and access level of the HR module.</li> </ul>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. HR Admin browses to the Tatari system enterprise portal.</li> <li>2. HR Admin logs in to the system by submitting proper credentials.</li> <li>3. A system dashboard of human resources module is shown with general statistics, sidebar and navigation.</li> <li>4. HR Admin clicks on the manage schedule option.</li> <li>5. System displays shift schedule and holiday page.</li> <li>6. HR Admin clicks on Add news.</li> <li>7. System displays a form to be filled out to create a new shift schedule and add day-offs.</li> <li>8. Admin fills out the form with details about the schedule like day and time or specific dates for holiday.</li> <li>9. HR Admin clicks save button.</li> <li>10. System validates the given information.</li> <li>11. System displays the shift schedule or holiday schedule.</li> </ol>
<b>Postcondition</b>	A new shift schedule record or holiday day-off schedule record is created.
<b>Alternate Flow of events</b>	<ul style="list-style-type: none"> <li>• Incorrect login credentials display login error.</li> <li>• If improper or empty required fields are found on the form, system displays an error message.</li> <li>• Schedule might already exist.</li> </ul>
<b>System derives</b>	Update, search and delete options of a shift schedule or day-offs are covered by the system.
<b>Business Rule</b>	3
<b>Frequency of Use</b>	Depends on the shift requirement and calendar of the organization.

Table 3.10 Manage Schedule Use case description

Use case ID: 10	Functional Requirement ID: FR-04.3
<b>Use case name</b>	Generate Timesheet
<b>Participating Actor</b>	Human Resources Administrator (HR Admin)   HR Officer-1
<b>Description</b>	A timesheet of either date-wise or month-based is generated as a report from attendance records of employees.

<b>Precondition</b>	<ul style="list-style-type: none"> <li>• HR Admin has access to a computer device; laptop or desktop.</li> <li>• HR Admin has access to the enterprise network, or Internet connection (for branch organizations via the Internet).</li> <li>• HR Admin is aware of the enterprise portal in which the Tatari System resides.</li> <li>• HR Admin has proper log-in credentials to the system with the proper permission and access level of the HR module.</li> </ul>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. HR Admin browses to the Tatari system enterprise portal.</li> <li>2. HR Admin logs in to the system by submitting proper credentials.</li> <li>3. A system dashboard of human resources module is shown with general statistics, sidebar and navigation.</li> <li>4. HR Admin clicks on the date-wise or monthly attendance timesheet option.</li> <li>5. System displays a daily attendance record or monthly timesheet.</li> <li>6. HR Admin fills out a date to get a date-wise attendance record.</li> <li>7. HR Admin clicks on the Get button.</li> <li>8. System displays a date-wise attendance record of an employee.</li> <li>9. Admin chooses the month to generate a timesheet of an employee for that month.</li> <li>10. HR Admin clicks the Get button.</li> <li>11. System validates the given information.</li> <li>12. System generates a timesheet with the required information.</li> <li>13. System displays the generated timesheet.</li> </ol>
<b>Postcondition</b>	A date-wise or monthly timesheet of an employee is generated and displayed which can be exported to various forms and reports.
<b>Alternate Flow of events</b>	<ul style="list-style-type: none"> <li>• Incorrect login credentials display login error.</li> <li>• If improper or empty required fields are found on the form, system displays an error message.</li> </ul>
<b>System derives</b>	Search and filter option for the generated timesheet is available and covered by the system.
<b>Business Rule</b>	3
<b>Frequency of Use</b>	Daily or Monthly, depends on the need of the attendance report.

Table 3.11 Generate Timesheet Use case description

<b>Use case ID:</b> 11		<b>Functional Requirement ID:</b>	<b>FR-04.4</b>
<b>Use case name</b>	Process Overtime		
<b>Participating Actor</b>	Human Resources Administrator (HR Admin)   HR Officer-1		
<b>Description</b>	This is used to receive and process overtime requests by employees for longer work hours outside of the original shift schedule.		
<b>Precondition</b>	<ul style="list-style-type: none"> <li>• HR Admin has access to a computer device; laptop or desktop.</li> <li>• HR Admin has access to the enterprise network, or Internet connection (for branch organizations via the Internet).</li> <li>• HR Admin is aware of the enterprise portal in which the Tatari System resides.</li> <li>• HR Admin has proper log-in credentials to the system with the proper permission and access level of the HR module.</li> </ul>		
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. HR Admin browses to the Tatari system enterprise portal.</li> <li>2. HR Admin logs in to the system by submitting proper credentials.</li> <li>3. A system dashboard of human resources module is shown with general statistics, sidebar and navigation.</li> <li>4. HR Admin selects Overtime Request option.</li> <li>5. System displays Overtime page.</li> <li>6. HR Admin clicks on Add New button.</li> <li>7. System displays a form to be filled out about the details of the overtime.</li> <li>8. HR Admin enters the information about the overtime such as date, reason, in-time and out-time.</li> <li>9. HR Admin clicks save button on the form.</li> <li>10. System validates the given information.</li> <li>11. System creates an overtime request.</li> <li>12. System displays the request for the particular employee and among the list of overtime requests for HR Admin.</li> </ol>		
<b>Postcondition</b>	An overtime request record has been created and processed for other needs.		
<b>Alternate Flow of events</b>	<ul style="list-style-type: none"> <li>• Incorrect login credentials display login error.</li> <li>• If improper or empty required fields are found on the form, the system displays an error message.</li> </ul>		
<b>System derives</b>	Update, search and delete options of the overtime request record are covered by the system.		
<b>Business Rule</b>	3		
<b>Frequency of Use</b>	Depends on the organization schedule as well as employee's need and performance.		

Table 3.12 Process Overtime Use case description

<b>Use case ID:</b> 12		<b>Functional Requirement ID:</b>	<b>FR-04.5</b>
<b>Use case name</b>	Process Leave		
<b>Participating Actor</b>	Human Resources Administrator (HR Admin)   HR Officer-1		
<b>Description</b>	An employee leave request is created and processed such as sick leave, bereavement leave.		
<b>Precondition</b>	<ul style="list-style-type: none"> <li>• HR Admin has access to a computer device; laptop or desktop.</li> <li>• HR Admin has access to the enterprise network, or Internet connection (for branch organizations via Internet).</li> <li>• HR Admin is aware of the enterprise portal in which the Tatari System resides.</li> <li>• HR Admin has proper log-in credentials to the system with the proper permission and access level of the HR module.</li> </ul>		
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. HR Admin browses to the Tatari system enterprise portal.</li> <li>2. HR Admin logs in to the system by submitting proper credentials.</li> <li>3. A system dashboard of human resources module is shown with general statistics, sidebar and navigation.</li> <li>4. HR Admin selects the Manage Leaves option.</li> <li>5. System displays Employee Leaves page.</li> <li>6. HR Admin clicks on Add New button.</li> <li>7. System displays a form to be filled out about the details of the leave.</li> <li>8. HR Admin enters the information about the overtime such as dates, reason and leaves type.</li> <li>9. HR Admin clicks save button on the form.</li> <li>10. System validates the given information.</li> <li>11. System creates a leave request record.</li> <li>12. System displays the leave request for the particular employee and among the list of leave requests for HR Admin.</li> </ol>		
<b>Postcondition</b>	A leave request record has been created and processed for other needs.		
<b>Alternate Flow of events</b>	<ul style="list-style-type: none"> <li>• Incorrect login credentials display login error.</li> <li>• If improper or empty required fields are found on the form, system displays an error message.</li> </ul>		
<b>System derives</b>	Update, search and delete options of the leave request record are covered by the system.		
<b>Business Rule</b>	3		
<b>Frequency of Use</b>	Depends on the employee's need and other situational factors.		

Table 3.13 Process Leave Use case description

<b>Use case ID:</b> 13		<b>Functional Requirement ID:</b>	<b>FR-05.1</b>
<b>Use case name</b>	Process Payroll		
<b>Participating Actor</b>	Human Resources Administrator (HR Admin)   Head of Finance		
<b>Description</b>	HR Administrator initiates Payroll for employees which Head of Finance processes salary payment.		
<b>Precondition</b>	<ul style="list-style-type: none"> <li>• HR Admin and Head of Finance have access to a computer device; laptop or desktop.</li> <li>• HR Admin and Head of Finance has access to the enterprise network, or Internet connection (for branch organizations via the Internet).</li> <li>• HR Admin and Head of Finance are aware of the enterprise portal in which the Tatari System resides.</li> <li>• HR Admin has proper log-in credentials to the system with the proper permission and access level of the system.</li> </ul>		
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. HR Admin browses to the Tatari system enterprise portal.</li> <li>2. HR Admin logs in to the system by submitting proper credentials.</li> <li>3. A system dashboard of human resources module is shown with general statistics, sidebar and navigation.</li> <li>4. HR Admin selects Payroll option.</li> <li>5. System displays Employee Payroll page.</li> <li>6. HR Admin initiates the process to payroll either individually or in bulk.</li> <li>7. HR Admin clicks Pay button or Bulk Payment option.</li> <li>8. System displays a notification about the payment process.</li> <li>9. System calculates payment and validates the information.</li> <li>10. System creates a payroll record for individual or bulk payment.</li> <li>11. System displays the payment record on the list of payrolls.</li> </ol>		
<b>Postcondition</b>	Payroll is initiated and processed, and the record is stored on the payroll record.		
<b>Alternate Flow of events</b>	<ul style="list-style-type: none"> <li>• Incorrect login credentials display login error.</li> <li>• If improper payment is initiated, displays payment not processed error.</li> </ul>		
<b>System derives</b>	View and delete option for the payroll record. System implicitly calculates the monthly salary, allowances, loan, deductions and overtime payments for the payroll.		
<b>Business Rule</b>	3, 4		
<b>Frequency of Use</b>	Monthly, depends on the need for payroll processing of the organization.		

Table 3.14 Process Payroll Use case description

<b>Use case ID: 14 Functional Requirement ID:</b>		<b>FR-05.2</b>
<b>Use case name</b>	Generate Payslips	
<b>Actors</b>	Head of Finance   Human Resources Administrator	
<b>Description</b>	Head of HR or Finance generates payslips for processed payments on the payroll either in single or in bulk.	
<b>Precondition</b>	<ul style="list-style-type: none"> <li>• Head of Finance or HR Admin has access to a computer device; laptop or desktop.</li> <li>• Head of Finance or HR Admin has access to the enterprise network, or Internet connection (for branch organizations via the Internet).</li> <li>• Head of Finance or HR Admin is aware of the enterprise portal in which the Tatari System resides.</li> <li>• Head of Finance or HR Admin has proper log-in credentials to the system with the proper permission and access level of the system.</li> </ul>	
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. HR Admin browses to the Tatari system enterprise portal.</li> <li>2. HR Admin logs in to the system by submitting proper credentials.</li> <li>3. A system dashboard of human resources module is shown with general statistics, sidebar and navigation.</li> <li>4. HR Admin selects Generate Payslips option on the Payroll option.</li> <li>5. System displays Payroll records with Payslips Generate option either individually or in bulk.</li> <li>6. HR Admin initiates the process to generate payslips either individually or in bulk.</li> <li>7. HR Admin clicks the pay slip button or Bulk Payslips option.</li> <li>8. System displays details about the payment and approval levels for the payment.</li> <li>9. System generates payslips for individual or bulk payment process.</li> <li>10. System provides a download or print option for the generated payslips.</li> <li>11. System displays the pay slip.</li> </ol>	
<b>Postcondition</b>	Payslips are processed with levels of approval and generated in a printable format.	
<b>Alternate Flow of events</b>	<ul style="list-style-type: none"> <li>• Incorrect login credentials display login error.</li> <li>• If an improper payslip is generated, displays payslips not processed error.</li> <li>• If payroll doesn't hold the payslip record needed to be generated</li> </ul>	
<b>System derives</b>	View, Download, Delete and other options for the generated payslips are covered by the system.	
<b>Business Rule</b>	4	
<b>Frequency of Use</b>	Monthly, depends on the payroll processing frequency.	

Table 3.15 Generate Payslips Use case description

<b>Use case ID: 15      Functional Requirement ID:</b>		<b>FR-06.1</b>
<b>Use case name</b>	Create Job Posts	
<b>Actor</b>	Human Resources Administrator (HR Admin)	
<b>Description</b>	HR Admin creates job posts/announcements for vacant positions in the organization, so that people suited for the position would see and apply for the position.	
<b>Precondition</b>	<ul style="list-style-type: none"> <li>• HR Admin has access to a computer device; laptop or desktop.</li> <li>• HR Admin has access to the enterprise network, or Internet connection (for branch organizations via Internet).</li> <li>• HR Admin is aware of the enterprise portal in which the Tatari System resides.</li> <li>• HR Admin has proper log-in credentials to the system with the proper permission and access level of the system.</li> </ul>	
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. HR Admin browses to the Tatari system enterprise portal.</li> <li>2. HR Admin logs in to the system by submitting proper credentials.</li> <li>3. A system dashboard of human resources module is shown with general statistics, sidebar and navigation.</li> <li>4. HR Admin clicks on the employee's recruitment option and then Job posts.</li> <li>5. System displays Job posts page.</li> <li>6. HR Admin clicks on Add New.</li> <li>7. System displays a form to be filled out to create a new job post, fields include position, description, experience and number.</li> <li>8. Admin fills out the form with details about the vacancy.</li> <li>9. HR Admin clicks save button.</li> <li>10. System validates the given information.</li> <li>11. System displays the job post on the list of open jobs and also on the vacancy page of the organization for the public.</li> </ol>	
<b>Postcondition</b>	A new job post is created and displayed on the vacancy page of the organization.	
<b>Alternate Flow of events</b>	<ul style="list-style-type: none"> <li>• Incorrect login credentials display login error.</li> <li>• If improper or empty required fields are found on the form, the system displays an error message.</li> <li>• Job post might already exist.</li> </ul>	
<b>System derives</b>	Search, update and delete options for the job posts are covered by the system.	
<b>Business Rule</b>	3	
<b>Frequency of Use</b>	Depends on the organizational needs and factors.	

Table 3.16 Create Job Posts Use case description

<b>Use case ID:</b> 16	<b>Functional Requirement ID:</b>	<b>FR-06.2</b>
<b>Use case name</b>	Submit job application	
<b>Actor</b>	Candidate-1	
<b>Description</b>	Candidate applies for a job post and provides necessary information for the vacancy.	
<b>Precondition</b>	<ul style="list-style-type: none"> <li>• Candidate-1 has access to a computer device; laptop or desktop.</li> <li>• Candidate-1 has access to the Internet connection.</li> <li>• Candidate-1 is aware of the enterprise public site and the Vacancy page.</li> </ul>	
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. Candidate-1 browses the organizational public website.</li> <li>2. Candidate-1 selects the vacancy option and system shows the available list of job posts by the organization.</li> <li>3. Candidate-1 selects his/her desired and qualified position from the job posts.</li> <li>4. System shows a form to be filled out for the job application.</li> <li>5. Candidate-1 enters the required information for the job application.</li> <li>6. Candidate-1 applies for the vacancy.</li> <li>7. System validates the provided information on the form.</li> <li>8. System displays successful submission of the job application.</li> </ol>	
<b>Postcondition</b>	System receives the job application submission and creates a record of the application for the candidate.	
<b>Alternate Flow of events</b>	<ul style="list-style-type: none"> <li>• If improper or empty required fields are found on the form, the system displays an error message.</li> </ul>	
<b>Business Rule</b>	3	
<b>Frequency of Use</b>	Depends on the organizational needs and number of vacant positions.	

Table 3.17 Submit Job Application Use case description

<b>Use case ID:</b> 17	<b>Functional Requirement ID:</b>	<b>FR-06.1   FR-06.3</b>
<b>Use case name</b>	Manage Job Applications	
<b>Actor</b>	Human Resources Administrator (HR Admin)	
<b>Description</b>	HR admin views and manages the list of candidates for the particular position/vacancy and their job application details.	
<b>Precondition</b>	<ul style="list-style-type: none"> <li>• HR Admin has access to a computer device; laptop or desktop.</li> <li>• HR Admin has access to the enterprise network, or Internet connection (for branch organizations via the Internet).</li> <li>• HR Admin is aware of the enterprise portal in which the Tatari System resides.</li> </ul>	

	<ul style="list-style-type: none"> <li>• HR Admin has proper log-in credentials to the system with the proper permission and access level of the system.</li> </ul>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. HR Admin browses to the Tatari system enterprise portal.</li> <li>2. HR Admin logs in to the system by submitting proper credentials.</li> <li>3. A system dashboard of human resources module is shown with general statistics, sidebar and navigation.</li> <li>4. HR Admin clicks on the employee's recruitment option and then Candidates.</li> <li>5. System displays Candidate's page.</li> <li>6. System displays a list of candidates and their submitted job application.</li> <li>7. HR Admin can view job applications in detail.</li> </ol>
<b>Postcondition</b>	System manages (view, search, delete) the candidate's job application information record.
<b>Alternate Flow of events</b>	<ul style="list-style-type: none"> <li>• Incorrect login credentials display login error.</li> <li>• If improper or empty required fields are found on the form, the system displays an error message.</li> </ul>
<b>System derives</b>	Search, view, filter and delete operations for the candidate records are covered by the system.
<b>Business Rule</b>	3
<b>Frequency of Use</b>	Depends on the organizational needs and number of vacant positions.

Table 3.18 Manage Job Application Use case description

Use case ID: 18	Functional Requirement ID:	FR-07
<b>Use case name</b>	Generate HR Reports	
<b>Actor</b>	Human Resources Administrator (HR Admin)	
<b>Description</b>	HR Admin uses this to generate various human-resource-related reports in a different format.	
<b>Precondition</b>	<ul style="list-style-type: none"> <li>• HR Admin has access to a computer device; laptop or desktop.</li> <li>• HR Admin has access to the enterprise network or Internet connection</li> <li>• HR Admin is aware of the enterprise portal in which the Tatari System resides.</li> <li>• HR Admin has proper log-in credentials to the system with the proper permission and access level of the system.</li> </ul>	

<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. HR Admin browses to the Tatari system enterprise portal.</li> <li>2. HR Admin logs in to the system by submitting proper credentials.</li> <li>3. A system dashboard of human resources module is shown with general statistics, sidebar and navigation.</li> <li>4. HR Admin selects on HR Reports option.</li> <li>5. System displays different types of HR reports that can be generated.</li> <li>6. HR Admin clicks on a particular kind of report like attendance, payslips, leave or employee report.</li> <li>7. System displays a form to be filled out to set the parameters of the report such as dates and employee.</li> <li>8. Admin fills out the form with details about the report.</li> <li>9. HR Admin clicks Generate button.</li> <li>10. System validates the given information.</li> <li>11. System produces the required kind of report with the given parameters which can be exported in a different format.</li> </ol>
<b>Postcondition</b>	System generated various kinds of human resource reports with the given parameters.
<b>Alternate Flow of events</b>	<ul style="list-style-type: none"> <li>• Incorrect login credentials display login error.</li> <li>• If improper or empty required fields are found on the form, the system displays an error message.</li> </ul>
<b>System derives</b>	Different kinds of report export methods would be available by the system such as Excel, CSV, PDF.
<b>Business Rule</b>	3
<b>Frequency of Use</b>	Depends on the need to generate reports and organizational factors.

Table 3.19 Generate HR Reports Use case description

Use case ID: 19	Functional Requirement ID:	FR-08.1
<b>Use case name</b>	Create Ledger Account	
<b>Participating Actor</b>	Head of Finance	
<b>Description</b>	Head of Finance/Accounting creates a general ledger account for all the financial and transactional needs of the organization.	
<b>Precondition</b>	<ul style="list-style-type: none"> <li>• Head of Finance has access to a computer device; laptop or desktop.</li> <li>• Head of Finance has access to the enterprise network, or Internet connection (for branch organizations via the Internet).</li> <li>• The Head of Finance is aware of the enterprise portal in which the Tatari System resides.</li> </ul>	

	<ul style="list-style-type: none"> <li>• Head of Finance has proper log-in credentials to the system with the highest permission and access level of the Finance Module of the system.</li> </ul>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. Head of Finance browses to the Tatari system enterprise portal.</li> <li>2. Head of Finance logs in to the system by submitting proper login credentials.</li> <li>3. A system dashboard of the financial module is shown with general statistics, sidebar and navigation.</li> <li>4. Head of Finance clicks on the Accounts option.</li> <li>5. System displays a form to be filled out to create a new ledger account and a list of accounts.</li> <li>6. Head of Finance fills out the form with details about the account like name, account number and banking information.</li> <li>7. Head of Finance clicks the save button.</li> <li>8. System validates the given information.</li> <li>9. System displays the new account record among the account list.</li> </ol>
<b>Postcondition</b>	A new general ledger account record has been created.
<b>Alternate Flow of events</b>	<ul style="list-style-type: none"> <li>• Incorrect login credentials display login error.</li> <li>• If improper or empty required fields are found on the form, system displays an error message.</li> <li>• The account might already exist.</li> </ul>
<b>Business Rule</b>	4
<b>Frequency of Use</b>	System initialization, and as required to maintain and update the account information and details.

Table 3.20 Create Ledger Account Use case description

Use case ID: 20	Functional Requirement ID:	FR-08.2   FR-08.3
Use case name	Manage Ledger Account	
Participating Actor	Head of Finance	
Description	The Head of Finance views the balance and other details as well as manages the ledger account by updating and deleting as necessary.	

<b>Precondition</b>	<ul style="list-style-type: none"> <li>• Head of Finance has access to a computer device; laptop or desktop.</li> <li>• Head of Finance has access to the enterprise network, or Internet connection (for branch organizations via the Internet).</li> <li>• Head of Finance is aware of the enterprise portal in which the Tatari System resides.</li> <li>• Head of Finance has proper log-in credentials to the system with the highest permission and access level of the Finance Module of the system.</li> </ul>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. Head of Finance browses to the Tatari system enterprise portal.</li> <li>2. Head of Finance logs in to the system by submitting proper login credentials.</li> <li>3. A system dashboard of financial module is shown with general statistics, sidebar and navigation.</li> <li>4. Head of Finance clicks on the Accounts option.</li> <li>5. System displays a form to be filled out to create a new ledger account and a list of accounts.</li> <li>6. Head of Finance views the account balance and other information. He/she updates or deletes the account as needed.</li> <li>7. Head of Finance clicks the save button or delete confirmation.</li> <li>8. System validates the given information.</li> <li>9. System displays the latest account record details among the account list.</li> </ol>
<b>Postcondition</b>	A ledger account record is searched, viewed, edited or deleted.
<b>Alternate Flow of events</b>	<ul style="list-style-type: none"> <li>• Incorrect login credentials display login error.</li> <li>• If improper or empty required fields are found on the form, system displays an error message.</li> <li>• Account might already exist.</li> </ul>
<b>Business Rule</b>	4
<b>Frequency of Use</b>	Depends on organizational factors and financial trends.

Table 3.21 Manage Ledger Account Use case description

Use case ID: 21	Functional Requirement ID:	FR-10.1
Use case name	Add Payee Information	
Participating Actor	Head of Finance   Accountant-1	
Description	Accountant adds information about a payee or receiver such as accountant, teller, or cashier.	

<b>Precondition</b>	<ul style="list-style-type: none"> <li>• Accountant-1 has access to a computer device; laptop or desktop.</li> <li>• Accountant-1 has access to the enterprise network, or Internet connection (for branch organizations via Internet).</li> <li>• Accountant-1 is aware of the enterprise portal in which the Tatari System resides.</li> <li>• Accountant-1 has proper log-in credentials to the system with the appropriate permission and access level of the Finance Module of the system.</li> </ul>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. Accountant-1 browses to the Tatari system enterprise portal.</li> <li>2. Accountant-1 logs in to the system by submitting proper login credentials.</li> <li>3. A system dashboard of financial module is shown with general statistics, sidebar and navigation.</li> <li>4. Accountant-1 clicks on the Payee option.</li> <li>5. System displays a form to be filled out to add a new payee and a list of payees.</li> <li>6. Accountant-1 fills out the form with details about the payee like name, telephone number.</li> <li>7. Accountant-1 clicks the save button.</li> <li>8. System validates the given information.</li> <li>9. System displays the new payee record among the payees list.</li> </ol>
<b>Postcondition</b>	A new payee record has been created.
<b>Alternate Flow of events</b>	<ul style="list-style-type: none"> <li>• Incorrect login credentials display login error.</li> <li>• If improper or empty required fields are found on the form, system displays an error message.</li> </ul>
<b>System derives</b>	Search, update and delete actions on the payee record are covered by the system.
<b>Business Rule</b>	4
<b>Frequency of Use</b>	Depends on the number of cashiers and tellers.

Table 3.22 Add Payee Information Use case description

Use case ID:	22	Functional Requirement ID:	FR-09.1
<b>Use case name</b>	Add Payer Information		
<b>Participating Actor</b>	Head of Finance   Accountant-1		
<b>Description</b>	Accountant adds information about a payer or client such as customer or financier.		

<b>Precondition</b>	<ul style="list-style-type: none"> <li>• Accountant-1 has access to a computer device; laptop or desktop.</li> <li>• Accountant-1 has access to the enterprise network, or Internet connection (for branch organizations via the Internet).</li> <li>• Accountant-1 is aware of the enterprise portal in which the Tatari System resides.</li> <li>• Accountant-1 has proper log-in credentials to the system with the appropriate permission and access level of the Finance Module of the system.</li> </ul>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. Accountant-1 browses the Tatari system enterprise portal.</li> <li>2. Accountant-1 logs in to the system by submitting proper login credentials.</li> <li>3. A system dashboard of financial module is shown with general statistics, sidebar and navigation.</li> <li>4. Accountant-1 clicks on the Payer option.</li> <li>5. The system displays a form to be filled out to add a new payer and a list of payers.</li> <li>6. Accountant-1 fills out the form with details about the payer like name, telephone number.</li> <li>7. Accountant-1 clicks the save button.</li> <li>8. System validates the given information.</li> <li>9. System displays the new payer record among the payers list.</li> </ol>
<b>Postcondition</b>	A new payer record has been created.
<b>Alternate Flow of events</b>	<ul style="list-style-type: none"> <li>• Incorrect login credentials display login error.</li> <li>• If improper or empty required fields are found on the form, the system displays an error message.</li> </ul>
<b>System derives</b>	Search, update and delete actions on the payer record are covered by the system.
<b>Business Rule</b>	4
<b>Frequency of Use</b>	Depends on the number of clients and customers.

Table 3.23 Add Payer Information Use case description

Use case ID: 23	Functional Requirement ID: FR-09
<b>Use case name</b>	Process Deposit
<b>Participating Actor</b>	Head of Finance   Accountant-1
<b>Description</b>	Accountant-1 processes and manages a deposit made to the organization.

<b>Precondition</b>	<ul style="list-style-type: none"> <li>• Accountant-1 has access to a computer device; laptop or desktop.</li> <li>• Accountant-1 has access to the enterprise network, or Internet connection (for branch organizations via Internet).</li> <li>• Accountant-1 is aware of the enterprise portal in which the Tatari System resides.</li> <li>• Accountant-1 has proper log-in credentials to the system with the appropriate permission and access level of the Finance Module of the system.</li> </ul>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. Accountant-1 browses to the Tatari system enterprise portal.</li> <li>2. Accountant-1 logs in to the system by submitting proper credentials.</li> <li>3. A system dashboard of financial module is shown with general statistics, sidebar and navigation.</li> <li>4. Accountant-1 clicks on the Deposit option.</li> <li>5. System displays the deposit page.</li> <li>6. Accountant-1 clicks on Add new deposit.</li> <li>7. System displays a form to be filled out to process a deposit.</li> <li>8. Accountant-1 fills out the form with details about the deposit like account, amount, payer, and date.</li> <li>9. Accountant-1 clicks the save button.</li> <li>10. System validates the given information.</li> <li>11. System displays the new deposit record on the deposit list.</li> </ol>
<b>Postcondition</b>	A deposit record has been created.
<b>Alternate Flow of events</b>	<ul style="list-style-type: none"> <li>• Incorrect login credentials display login error.</li> <li>• If improper or empty required fields are found on the form, the system displays an error message.</li> </ul>
<b>Include</b>	Search, update and delete actions on the deposit record are covered by the system.
<b>Business Rule</b>	4
<b>Frequency of Use</b>	Depends on organizational factors and market indicators.

Table 3.24 Process Deposit Use case description

Use case ID: 24	Functional Requirement ID:	FR-10
<b>Use case name</b>	Process Expense	
<b>Participating Actor</b>	Head of Finance   Accountant-1	
<b>Description</b>	Accountant-1 processes and manages an expense made by the organization.	

<b>Precondition</b>	<ul style="list-style-type: none"> <li>• Accountant-1 has access to a computer device; laptop or desktop.</li> <li>• Accountant-1 has access to the enterprise network, or Internet connection (for branch organizations via the Internet).</li> <li>• Accountant-1 is aware of the enterprise portal in which the Tatari System resides.</li> <li>• Accountant-1 has proper log-in credentials to the system with the appropriate permission and access level of the Finance Module of the system.</li> </ul>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. Accountant-1 browses to the Tatari system enterprise portal.</li> <li>2. Accountant-1 logs in to the system by submitting proper credentials.</li> <li>3. A system dashboard of financial module is shown with general statistics, sidebar and navigation.</li> <li>4. Accountant-1 clicks on the Expense option.</li> <li>5. System displays the expense page.</li> <li>6. Accountant-1 clicks on Add new expense.</li> <li>7. System displays a form to be filled out to process an expense.</li> <li>8. Accountant-1 fills out the form with details about the expense like account, amount, payee, and date.</li> <li>9. Accountant-1 clicks the save button.</li> <li>10. System validates the given information.</li> <li>11. System displays the new expense record on the deposit list.</li> </ol>
<b>Postcondition</b>	An expense record has been created.
<b>Alternate Flow of events</b>	<ul style="list-style-type: none"> <li>• Incorrect login credentials display login error.</li> <li>• If improper or empty required fields are found on the form, system displays an error message.</li> <li>• Expense over the balance displays over expense error.</li> </ul>
<b>System derives</b>	Search, update and delete actions on the expense record are covered by the system.
<b>Business Rule</b>	4
<b>Frequency of Use</b>	Depends on the business factors and market indicators.

Table 3.25 Process Expense Use case description

Use case ID: 25	Functional Requirement ID:	FR-11
<b>Use case name</b>	Transfer Funds	
<b>Participating Actor</b>	Head of Finance	

<b>Description</b>	Head of Finance transfers funds from one ledger account to the other for any number of operational reasons.
<b>Precondition</b>	<ul style="list-style-type: none"> <li>• Head of Finance has access to a computer device; laptop or desktop.</li> <li>• Head of Finance has access to the enterprise network, or Internet connection (for branch organizations via Internet).</li> <li>• Head of Finance is aware of the enterprise portal in which the Tatari System resides.</li> <li>• Head of Finance has proper log-in credentials to the system with the highest permission and access level of the Finance Module of the system.</li> </ul>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. Head of Finance browses to the Tatari system enterprise portal.</li> <li>2. Head of Finance logs in to the system by submitting proper login credentials.</li> <li>3. A system dashboard of financial module is shown with general statistics, sidebar and navigation.</li> <li>4. Head of Finance clicks on the Transfer option.</li> <li>5. System displays a form to be filled out to transfer funds between ledger accounts.</li> <li>6. Head of Finance fills the form with the account information and amount to transfer views the account balances of the accounts.</li> <li>7. Head of Finance clicks the save button.</li> <li>8. System validates the given information.</li> <li>9. System displays a notification about the successful transfer of funds.</li> </ol>
<b>Postcondition</b>	Record for the transfer of funds between account is created.
<b>Alternate Flow of events</b>	<ul style="list-style-type: none"> <li>• Incorrect login credentials display login error.</li> <li>• If improper or empty required fields are found on the form, system displays an error message.</li> <li>• Transfer between the same accounts shows an error notification.</li> <li>• Amount of transfer exceeding balance shows an error.</li> </ul>
<b>Business Rule</b>	4, 11
<b>Frequency of Use</b>	Depends on the organizational factors and market indicators.

Table 3.26 Transfer Funds Use case description

<b>Use case ID:</b> 26	<b>Functional Requirement ID:</b>	<b>FR-12</b>
<b>Use case name</b>	View Transactions	
<b>Participating Actor</b>	Head of Finance	
<b>Description</b>	Head of Finance can view and search through the overall financial transaction records of the organization.	
<b>Precondition</b>	<ul style="list-style-type: none"> <li>• Head of Finance has access to a computer device; laptop or desktop.</li> <li>• Head of Finance has access to the enterprise network, or Internet connection (for branch organizations via Internet).</li> <li>• Head of Finance is aware of the enterprise portal in which the Tatari System resides.</li> <li>• Head of Finance has proper log-in credentials to the system with the highest permission and access level of the Finance Module of the system.</li> </ul>	
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. Head of Finance browses to the Tatari system enterprise portal.</li> <li>2. Head of Finance logs in to the system by submitting proper login credentials.</li> <li>3. A system dashboard of financial module is shown with general statistics, sidebar and navigation.</li> <li>4. Head of Finance clicks on the Transactions option.</li> <li>5. System displays a list of all financial transactions made with other supplementary information.</li> <li>6. Head of Finance can search through the transactions to find a specific record.</li> </ol>	
<b>Postcondition</b>	Transaction records are displayed. They can be sorted and searched through.	
<b>Alternate Flow of events</b>	<ul style="list-style-type: none"> <li>• Incorrect login credentials display login error.</li> </ul>	
<b>System derives</b>	Search and sort option for the displayed list of transaction records are covered by the system.	
<b>Business Rule</b>	4, 11	
<b>Frequency of Use</b>	Depends on the business factors and need for transaction records.	

Table 3.27 View Transactions Use case description

<b>Use case ID:</b> 27	<b>Functional Requirement ID:</b>	<b>FR-13</b>
<b>Use case name</b>	Create Invoice	
<b>Participating Actor</b>	Head of Finance   Accountant-1	
<b>Description</b>	Accountant-1 creates an invoice/bill for a recorded statement for a purchased item required for a project.	

<b>Precondition</b>	<ul style="list-style-type: none"> <li>• Accountant-1 has access to a computer device; laptop or desktop.</li> <li>• Accountant-1 has access to the enterprise network, or Internet connection (for branch organizations via Internet).</li> <li>• Accountant-1 is aware of the enterprise portal in which the Tatari System resides.</li> <li>• Accountant-1 has proper log-in credentials to the system with the appropriate permission and access level of the Finance Module of the system.</li> </ul>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. Accountant-1 browses to the Tatari system enterprise portal.</li> <li>2. Accountant-1 logs in to the system by submitting proper credentials.</li> <li>3. A system dashboard of financial module is shown with general statistics, sidebar and navigation.</li> <li>4. Accountant-1 clicks on the Invoices option.</li> <li>5. System displays the invoice page.</li> <li>6. Accountant-1 clicks on Create new Invoice.</li> <li>7. System displays a form to be filled out to produce an invoice.</li> <li>8. Admin fills out the form with details about the invoice like invoice number, amount, item, tax, unit price, discount and date.</li> <li>9. Accountant-1 clicks submit invoice button.</li> <li>10. System validates the given information.</li> <li>11. System displays the new invoice record on the invoice list.</li> </ol>
<b>Postcondition</b>	Invoice record for payment/purchase is created.
<b>Alternate Flow of events</b>	<ul style="list-style-type: none"> <li>• Incorrect login credentials display login error.</li> <li>• If improper or empty required fields are found on the form, system displays an error message.</li> <li>• Tax and subtotal calculation error arise error notification.</li> </ul>
<b>System derives</b>	Search, update and delete options are covered by the system implicitly.
<b>Business Rule</b>	4
<b>Frequency of Use</b>	Depends on the business factors such as demand.

Table 3.28 Create Invoice Use case description

Use case ID: 28	Functional Requirement ID:	FR-14
<b>Use case name</b>	Generate Fiscal Reports	
<b>Participating Actor</b>	Head of Finance	
<b>Description</b>	Head of Finance uses this to generate various financial-related reports in a different format.	

<b>Precondition</b>	<ul style="list-style-type: none"> <li>• Head of Finance has access to a computer device; laptop or desktop.</li> <li>• Head of Finance has access to the enterprise network, or Internet connection (for branch organizations via Internet).</li> <li>• Head of Finance is aware of the enterprise portal in which the Tatari System resides.</li> <li>• Head of Finance has proper log-in credentials to the system with the highest permission and access level of the Finance Module of the system.</li> </ul>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. Head of Finance browses to the Tatari system enterprise portal.</li> <li>2. Head of Finance logs in to the system by submitting proper credentials.</li> <li>3. A system dashboard of financial module is shown with general statistics, sidebar and navigation.</li> <li>4. Head of Finance selects on Report's option.</li> <li>5. System displays different types of Fiscal reports that can be generated.</li> <li>6. Head of Finance clicks on a particular kind of report like account, expense, income or transfer reports.</li> <li>7. System displays a form to be filled out to set the parameters of the report such as dates and employees.</li> <li>8. Head of Finance fills out the form with details about the report.</li> <li>9. Head of Finance clicks Generate button.</li> <li>10. System validates the given information.</li> <li>11. System produces the required kind of report with the given parameters which can be exported in a different format.</li> </ol>
<b>Postcondition</b>	System generated various kinds of fiscal reports with the given parameters.
<b>Alternate Flow of events</b>	<ul style="list-style-type: none"> <li>• Incorrect login credentials display login error.</li> <li>• If improper or empty required fields are found on the form, system displays an error message.</li> </ul>
<b>System derives</b>	Different kinds of report export methods would be available by the system such as Excel, CSV, PDF.
<b>Business Rule</b>	4
<b>Frequency of Use</b>	Depends on the need to generate reports and organizational factors.

Table 3.29 Generate Fiscal Reports Use case description

Use case ID: 29	Functional Requirement ID:	FR-02.2
<b>Use case name</b>	View Profile	
<b>Actor</b>	Employee-1	

<b>Description</b>	An employee of the organization would view his/her profile and work history with the organization. They will be provided with a summary of their time with the company.
<b>Precondition</b>	<ul style="list-style-type: none"> <li>Employee-1 has access to computer and the enterprise network.</li> <li>Employee-1 is aware of the enterprise portal in which the Tatari System resides and has logged in to the system.</li> </ul>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>An employee dashboard is displayed with some information, sidebar and profiles options.</li> <li>Employee clicks on the profile icon.</li> <li>System displays a profiles page with different tabs about employee's information, work history, leaves and payments.</li> <li>Employee click the different options to view various aspects of his/her work profile.</li> </ol>
<b>Postcondition</b>	Employee's profile containing work history, salary, requests and other related information are displayed.
<b>Alternate Flow</b>	<ul style="list-style-type: none"> <li>Incorrect login credentials display login error.</li> </ul>
<b>Includes</b>	Searching through one's individual work history, updating some minor details are covered by the system. Viewing a summary of payments and attendance is also included here in the view profile function.
<b>Frequency of Use</b>	Depends on the employee's need to view his/her work profile.

Table 3.30 View Profile Use case description

Use case ID:	30	Functional Requirement ID:	FR-04.4
<b>Use case name</b>	Request Overtime		
<b>Actor</b>	Employee-1		
<b>Description</b>	An employee of the organization would able to put in a request for overtime work hours using the system.		
<b>Precondition</b>	<ul style="list-style-type: none"> <li>Employee-1 has access to computer and the enterprise network.</li> <li>Employee-1 knows enterprise portal and is logged in to system.</li> </ul>		
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>An employee dashboard is displayed with summaries and detail.</li> <li>Employee clicks on the request overtime option from sidebar.</li> <li>System displays an overtime requests list page with an option to request a new one.</li> <li>Employee fills out a form with date, time and reason for the overtime request and submits the form.</li> <li>System displays the new request among the list and its status.</li> </ol>		
<b>Postcondition</b>	Employee requested an overtime work hour and its record is created.		
<b>Alternate Flow</b>	<ul style="list-style-type: none"> <li>Incorrect login credentials display login error.</li> </ul>		
<b>Includes</b>	Updating the overtime request is covered by the system.		
<b>Frequency of Use</b>	Depends on the employee's need to work overtime and performance.		

Table 3.31 Request Overtime Use case description

## 3.5. Object Model

### 3.5.1. Class Diagram

As per the object-oriented design methodology, the following class diagrams show the static structure of the system objects with their classes, attributes, operations and the relationship among objects; that were identified from the above-discussed use cases and scenarios.

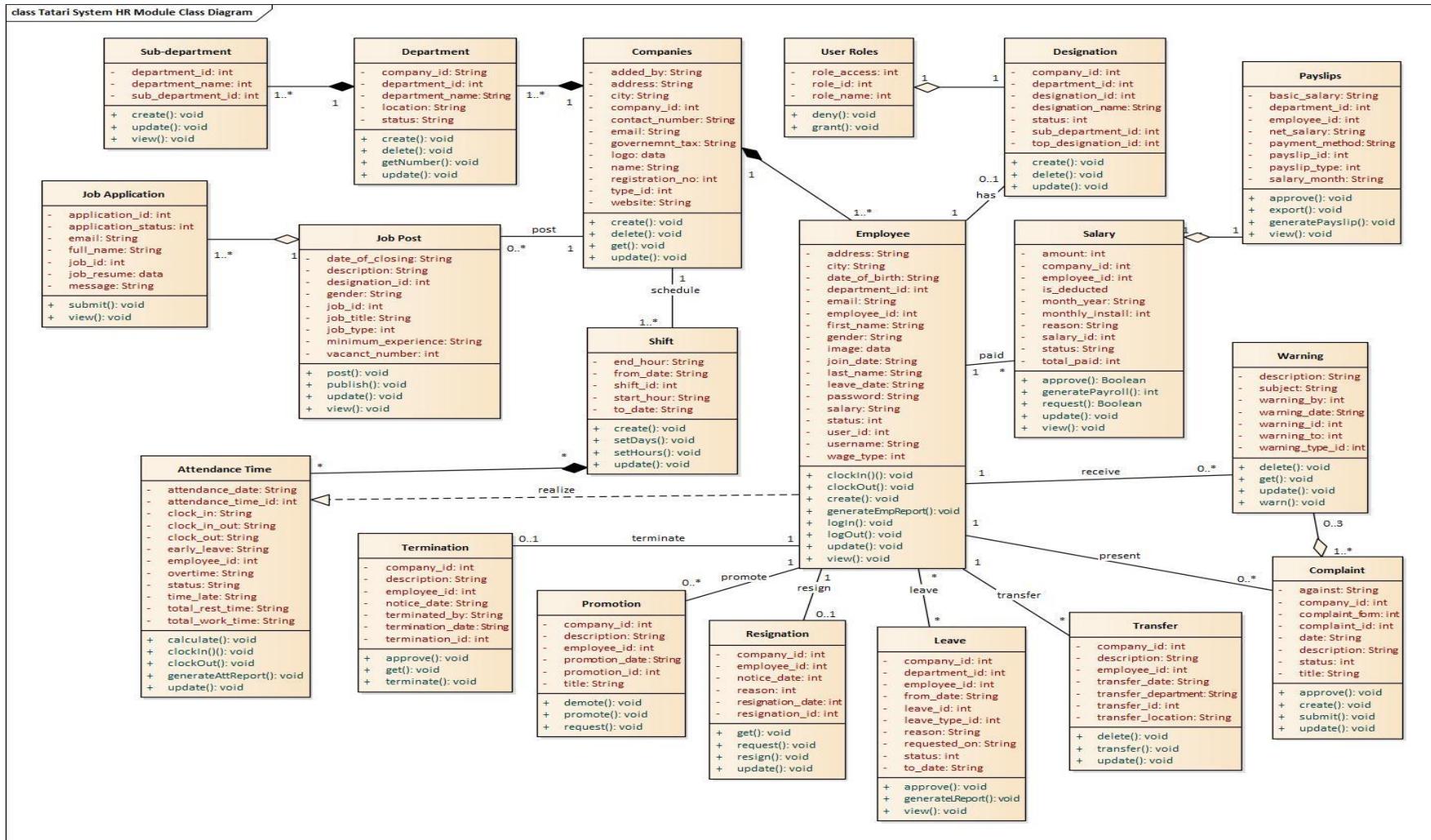


Figure 3.3 Tatari System Class Diagram-I HR Module

class Tatari System Finance Module Class Diagram

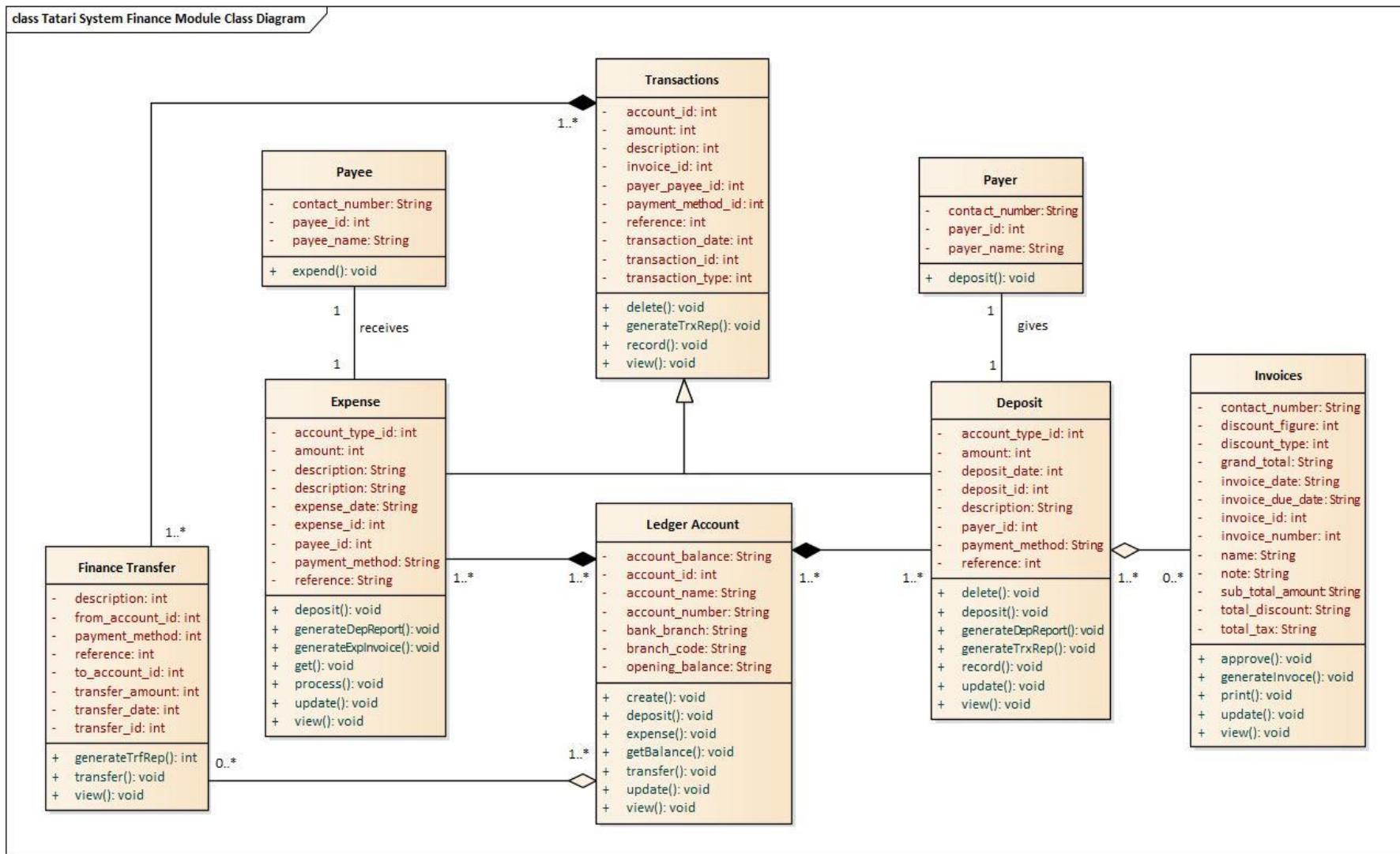


Figure 3.4 Tatari System Class Diagram-2 Finance Module

### 3.5.2. Data Dictionary

The following table shows the detailed features of data type in our database design.

Data Item	Data Type	Data Format	Number of Character	Description	Example	Validation
<b>Id</b>	Number	Digits	5-11 (Length differs for different table)	Unique identifier for user, role, salary, transaction	51239	Must be valid id address for system tables (should be unique for a table)
<b>Account Number</b>	Number	Digits	13	A general ledger account number that uniquely identifies the account	1000212332323	Must be unique in the account list
<b>InvoiceID</b>	Text	Alphanumeric character string	9	A string to uniquely identifies an invoice	Inv723115	Must be unique for every invoice
<b>Company Name</b>	Text	Alphabet Character String	100	Name of the organization	Educational Materials Production and Distribution Enterprise	Must be valid and officially recognized name of the organization
<b>Transaction ID</b>	Text	Alphanumeric character string	12	Unique identifier for every transaction	TrD323212118	Must be valid and unique for transactions
<b>Attendance Date</b>	Date	YYYY-MM-DD	10	An identifier for a particular day of the year.	2020-05-07	Must be a valid date
<b>Clock-in time</b>	Time	HH:MM	5	An identifier for a specific time of the day	04:15	Must be a valid time
<b>Image</b>	Text	Path of image file	100	The absolute path of the folder and directory of image file	Server/assets/images/emp2212.jpg	Valid directory, folder and file path

Table 3.32 Data Dictionary

## 3.6. Dynamic Model

### 3.6.1. Sequence Diagram

One of the most widely used interaction diagrams is Sequence Diagram. It depicts the interaction between objects in a sequential manner. It shows the order of interactions that take place. From the analysis phase of use cases and flow of events, the following sequence diagrams portray the interaction between actors, lifelines, views, controllers and databases by passing messages around.

#### 3.6.1.1. Register Organization Sequence Diagram

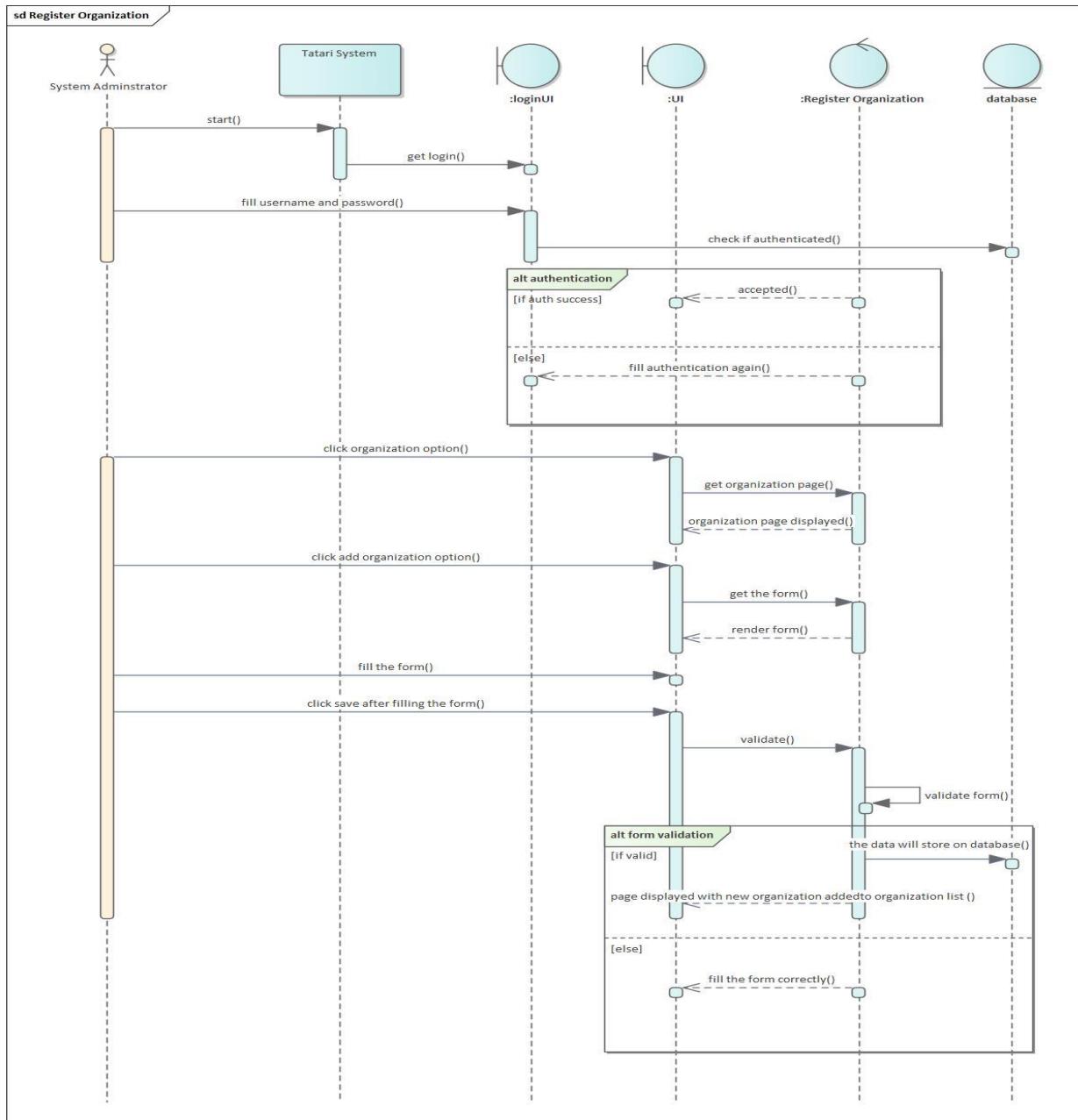


Figure 3.5 Register Organization Sequence Diagram

### 3.6.1.2. Submit Attendance Sequence Diagram

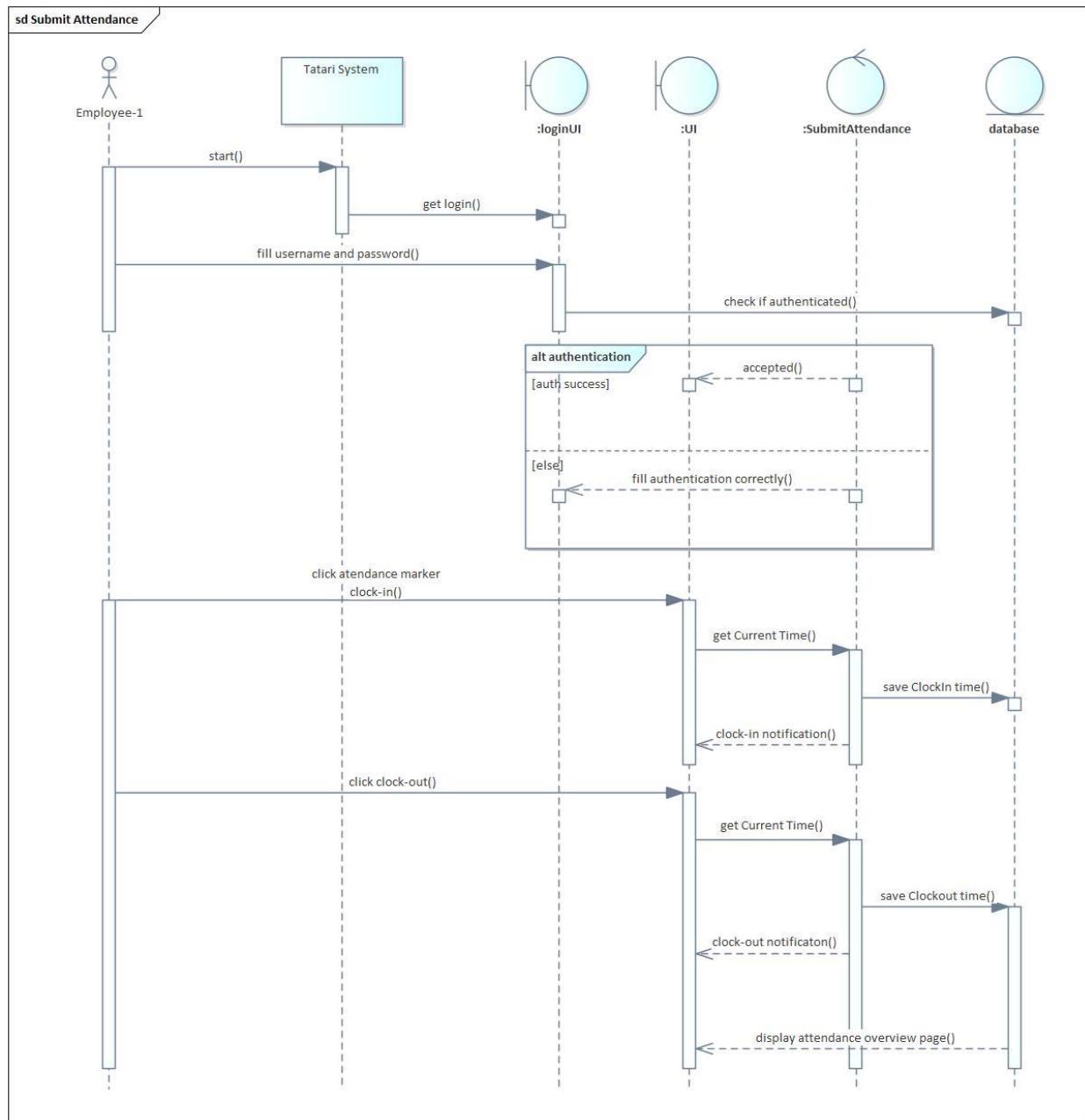


Figure 3.6 Submit Attendance Sequence Diagram

### 3.6.1.3. Process Payroll Sequence Diagram

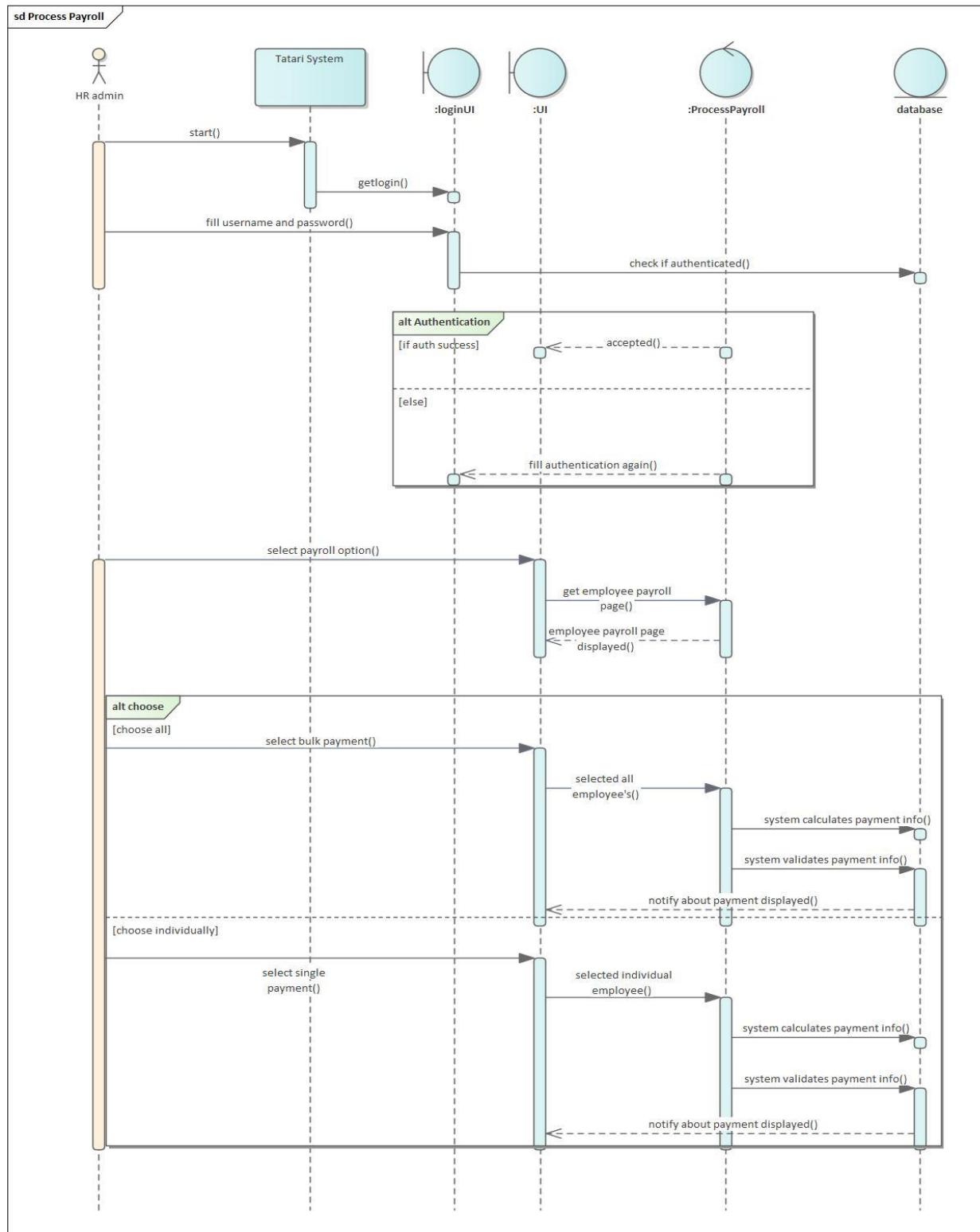


Figure 3.7 Process Payroll Sequence Diagram

### 3.6.1.4. Submit Job Application Sequence Diagram

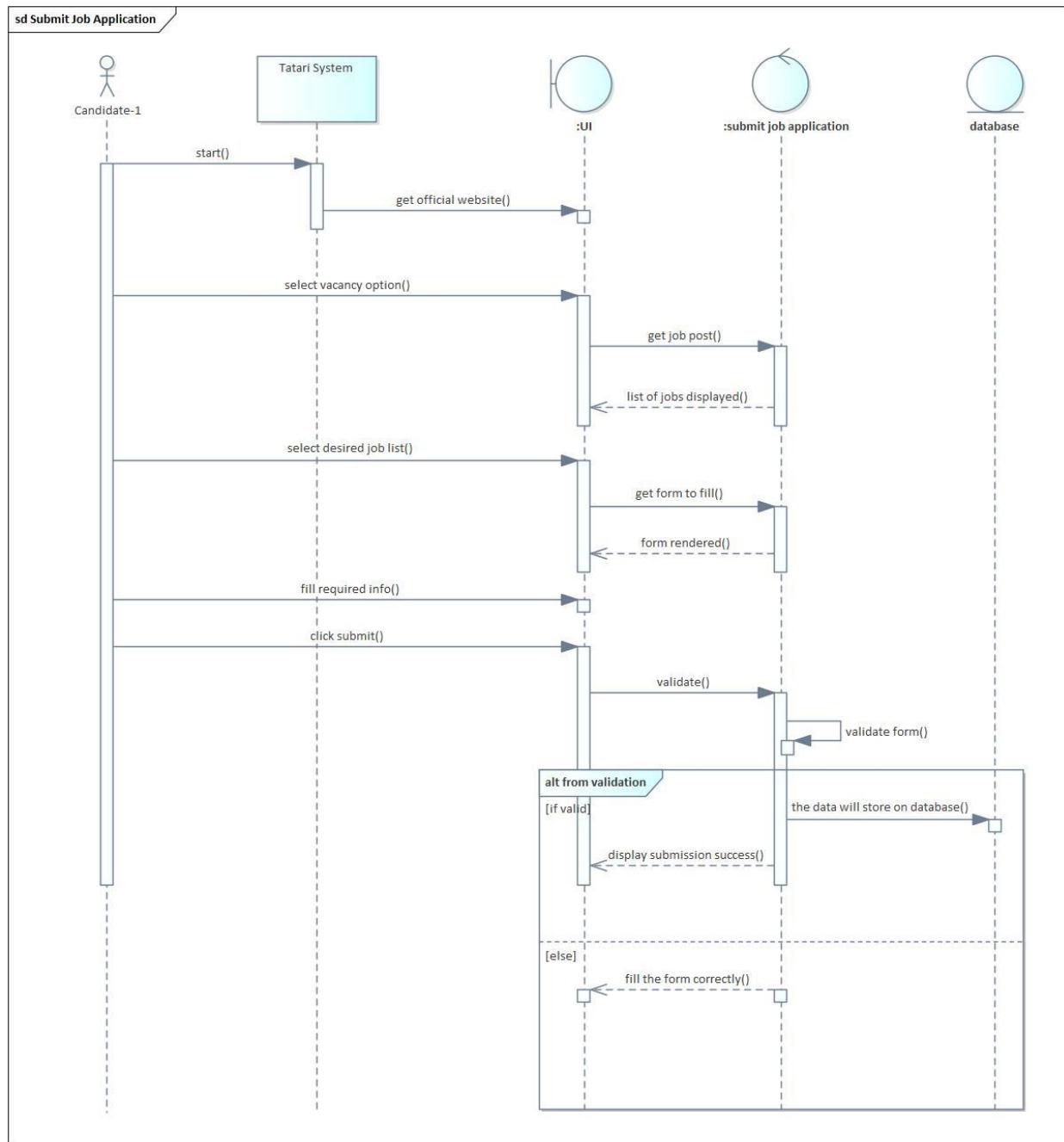


Figure 3.8 Submit Job Application Sequence Diagram

### 3.6.1.5. Generate HR Reports Sequence Diagram

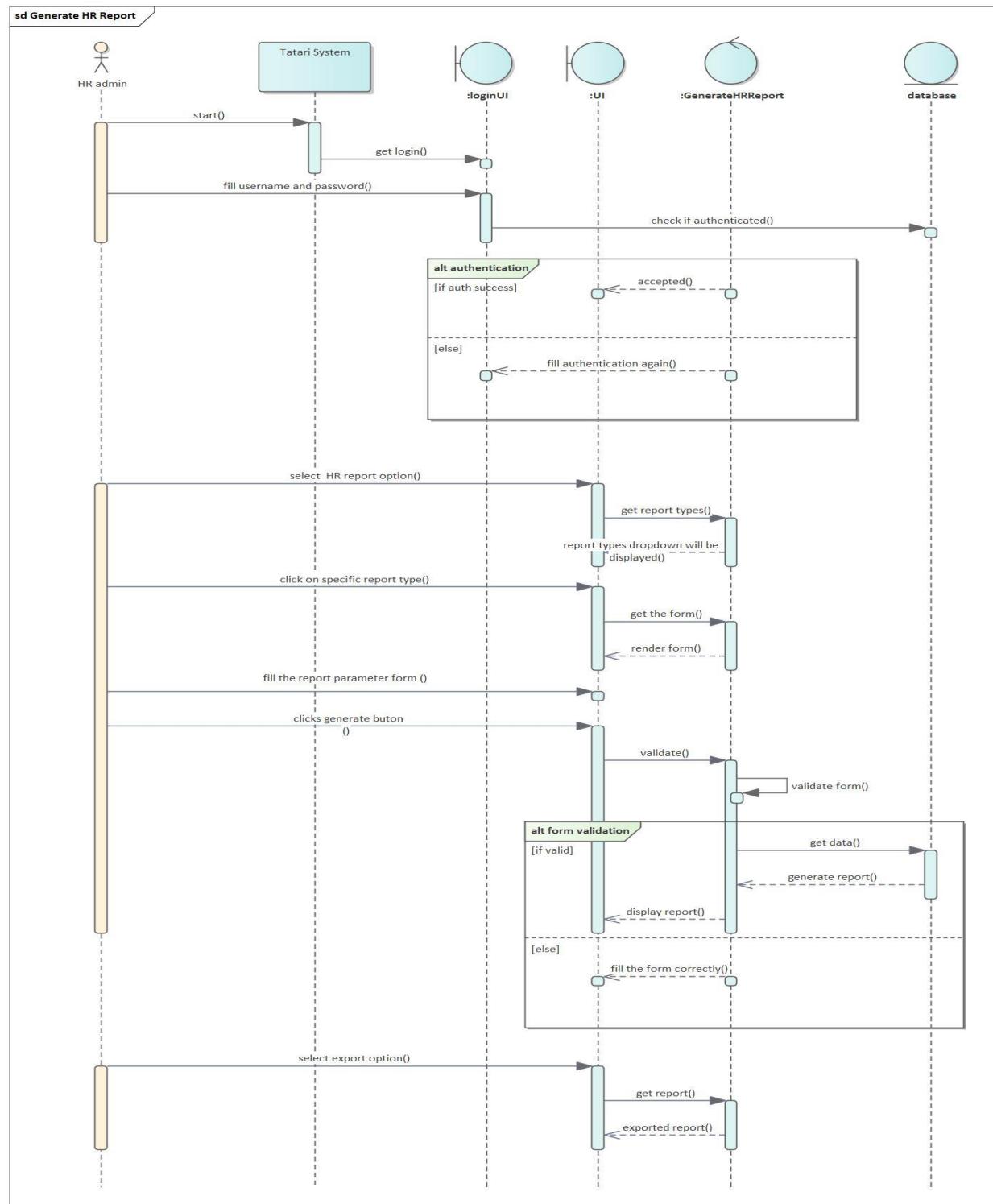


Figure 3.9 Generate HR Reports Sequence Diagram

### 3.6.2. Activity Diagram

#### 3.6.2.1. Transfer Employee Activity Diagram

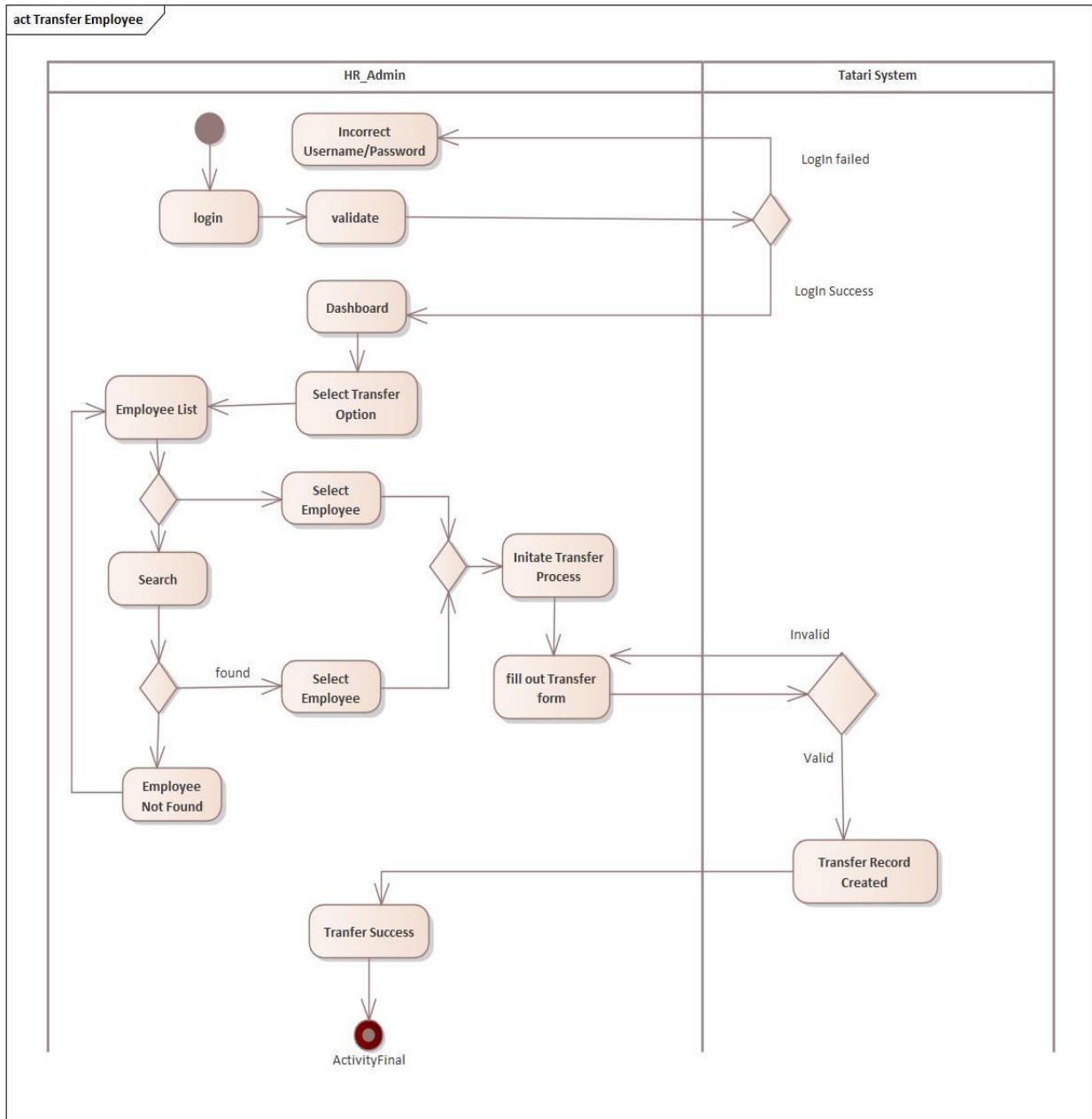


Figure 3.10 Transfer Employee Activity Diagram

### 3.6.2.2. Process Deposit Activity Diagram

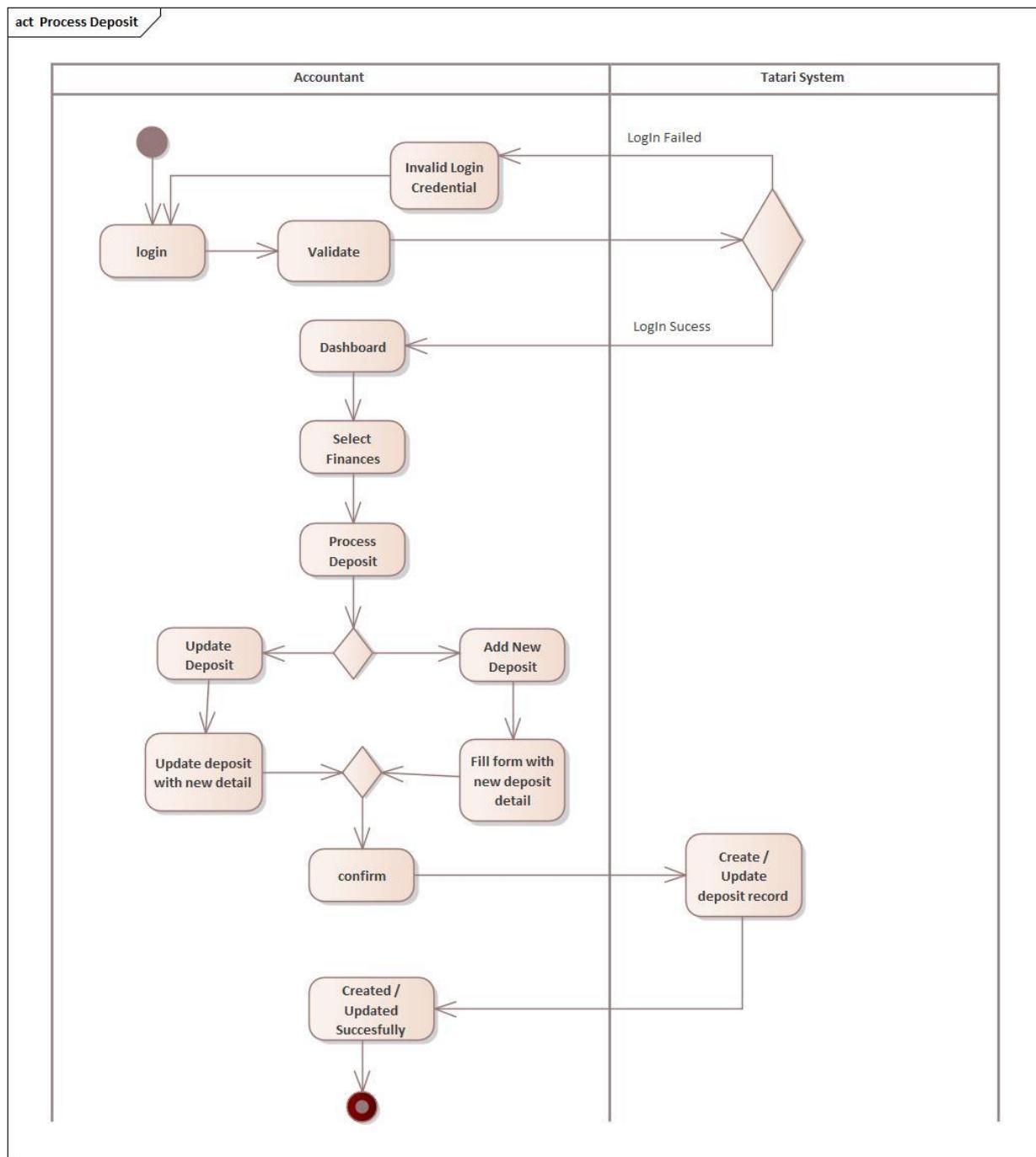


Figure 3.11 Process Deposit Activity Diagram

### 3.6.2.3. Register Employee Activity Diagram

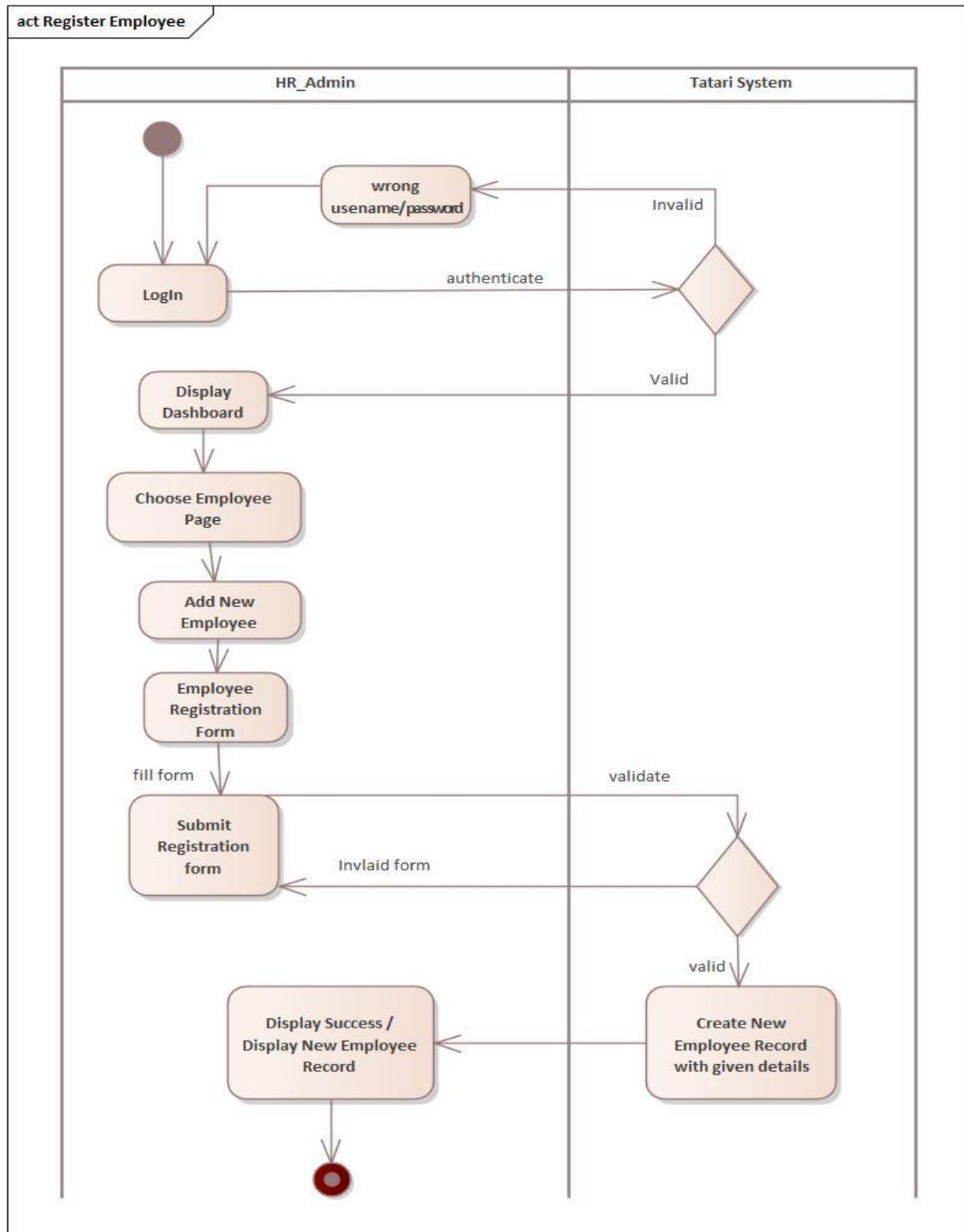


Figure 3.12 Register Employee Activity Diagram

### 3.6.2.4. Generate HR Report Activity Diagram

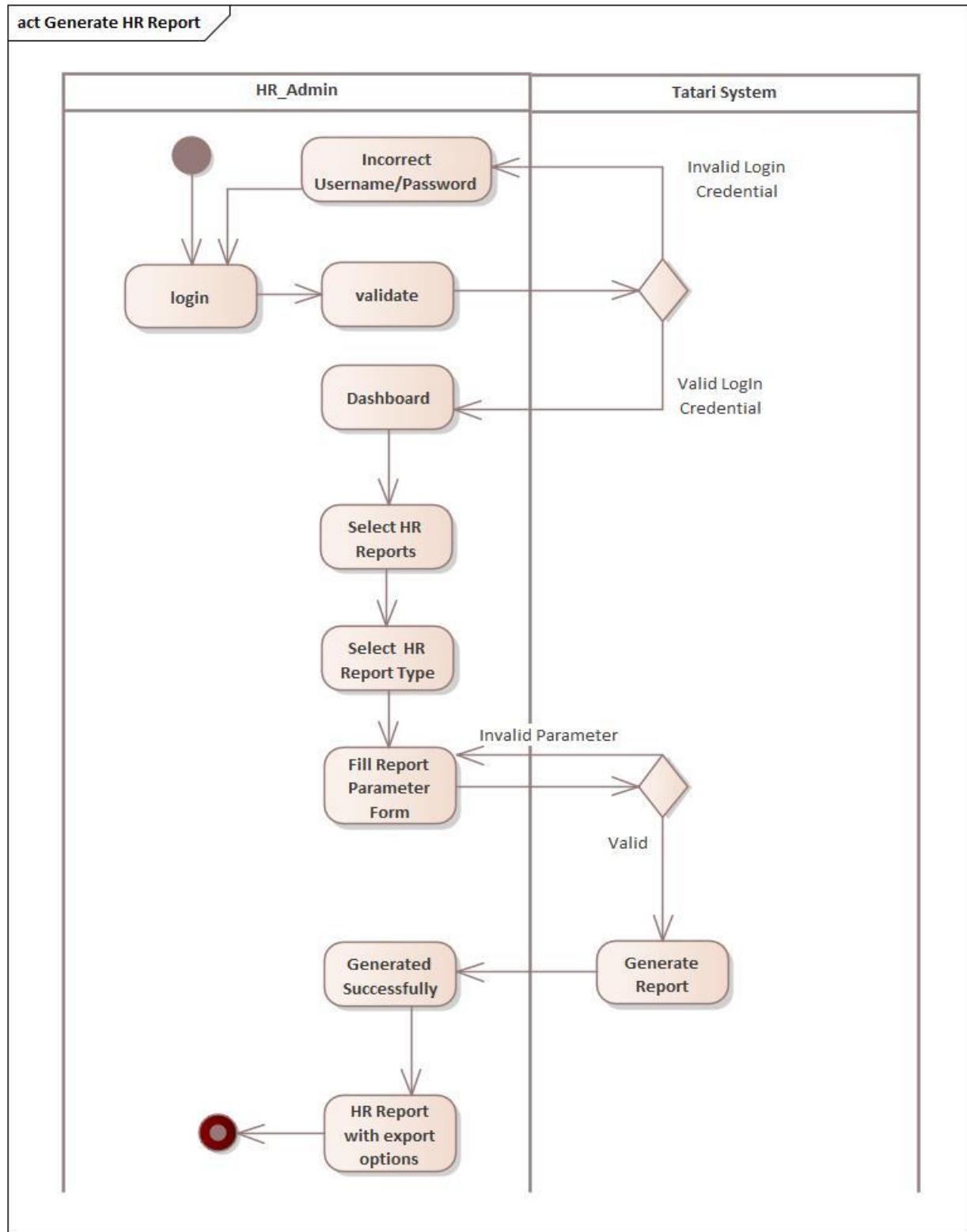


Figure 3.13 Generate HR Report Activity Diagram

### 3.6.2.5. Submit Job Application and Manage Job Application Activity Diagram

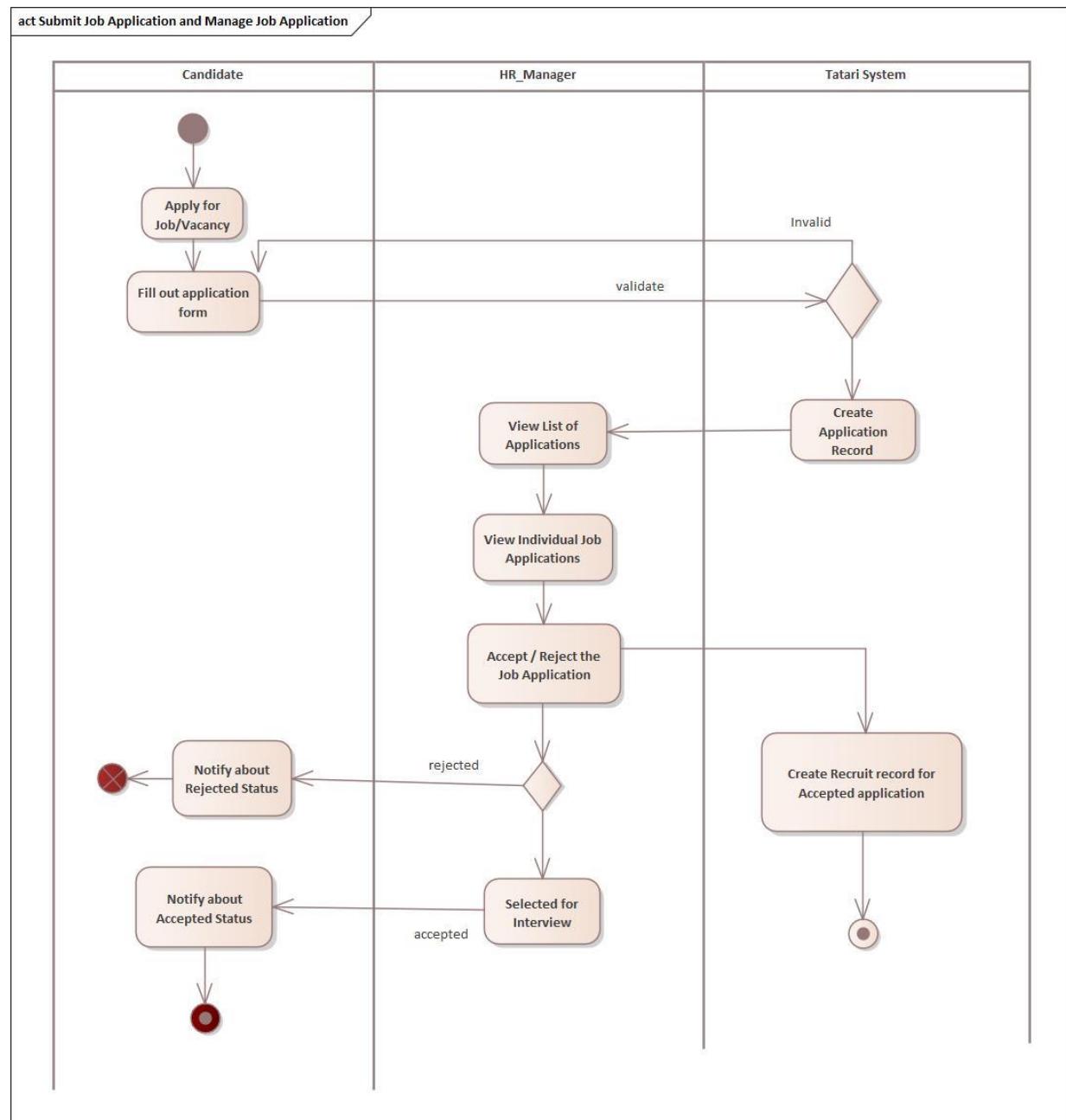


Figure 3.14 Submit Job Application and Manage Job Application Activity Diagram

### 3.6.3. State Chart Diagram

#### 3.6.3.1. Register Employee State Diagram

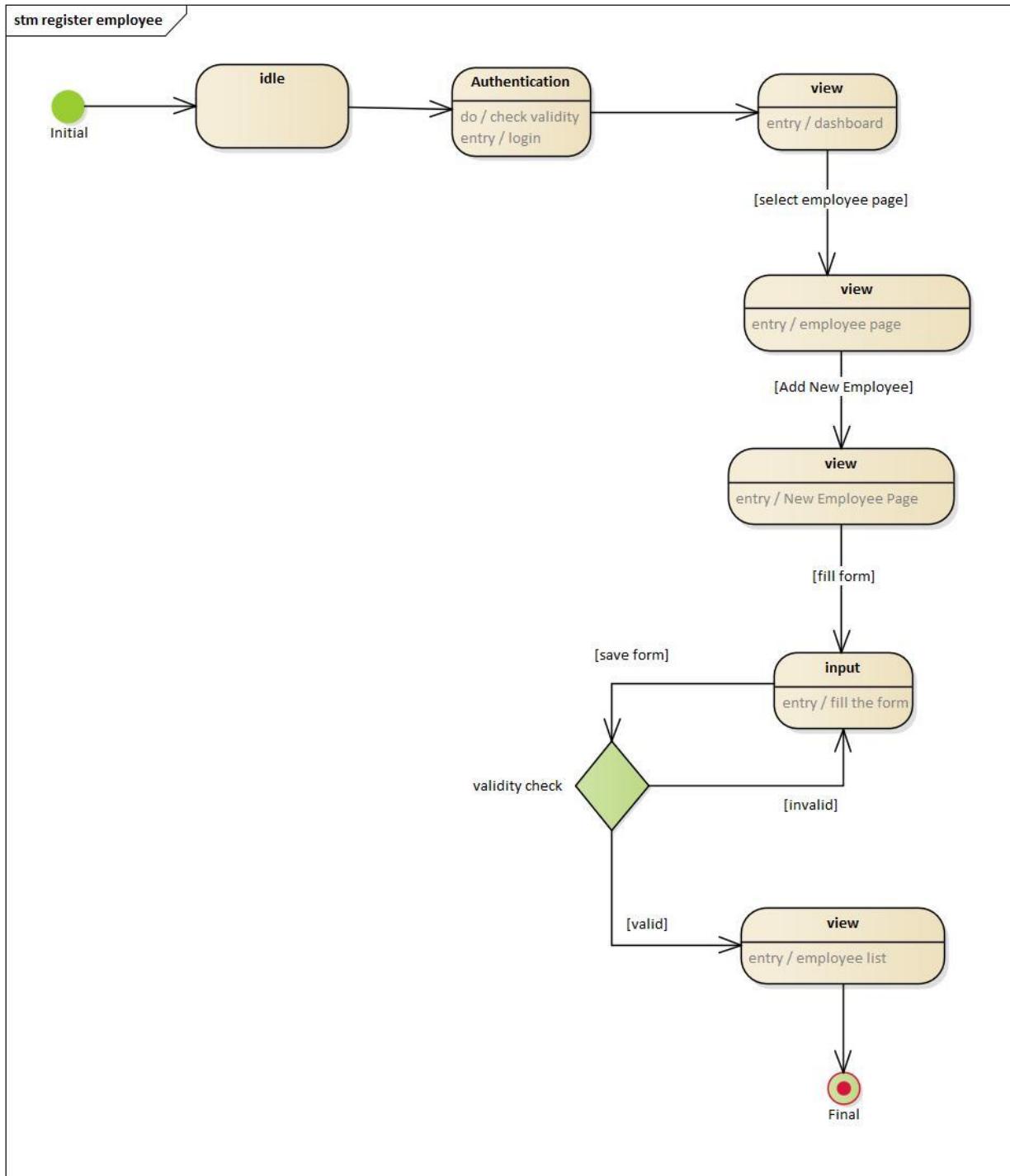


Figure 3.15 Register Employee State Diagram

### 3.6.3.2. Payroll Processing State Diagram

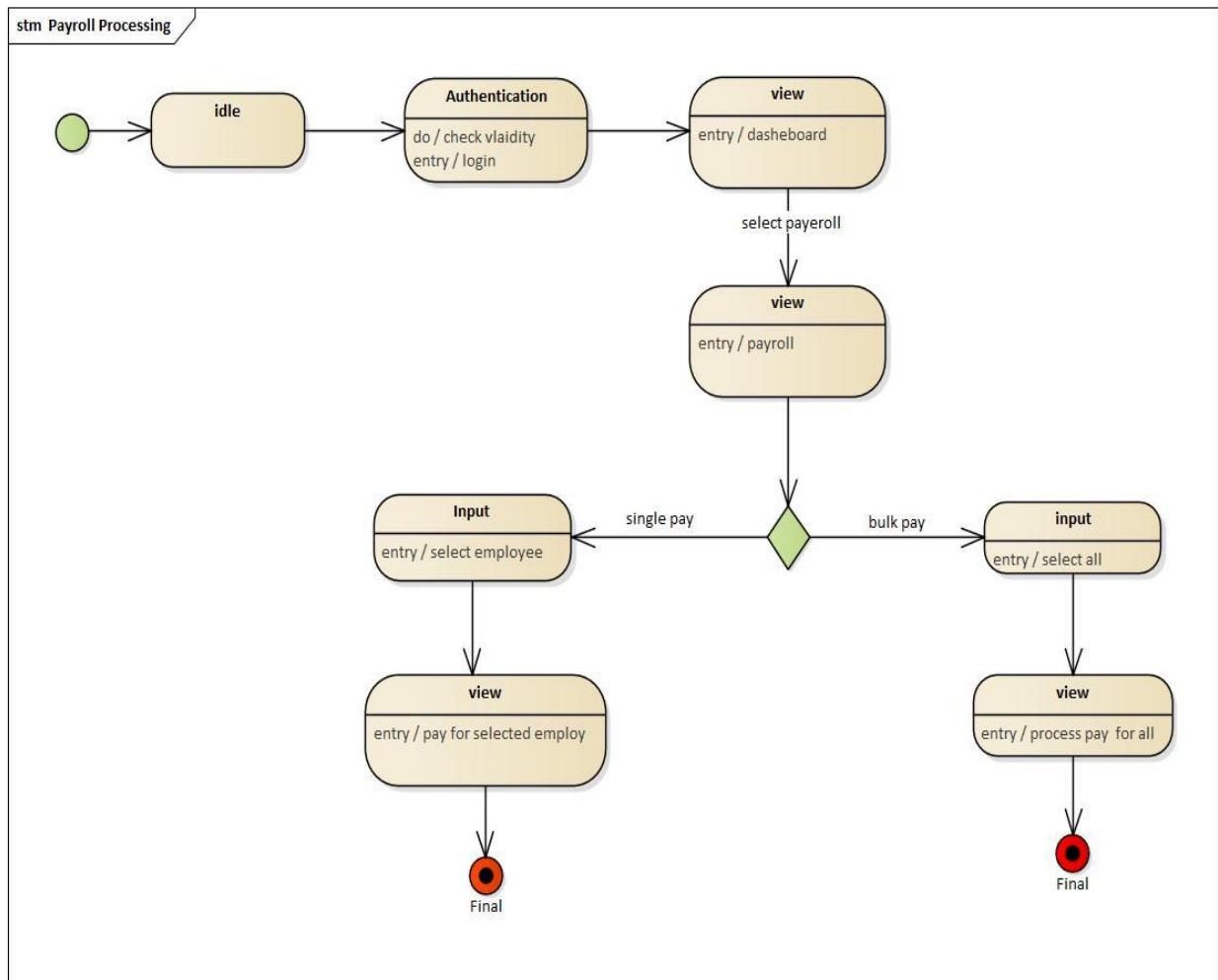


Figure 3.16 Payroll Processing State Diagram

### 3.6.3.3. Promote Employee State Diagram

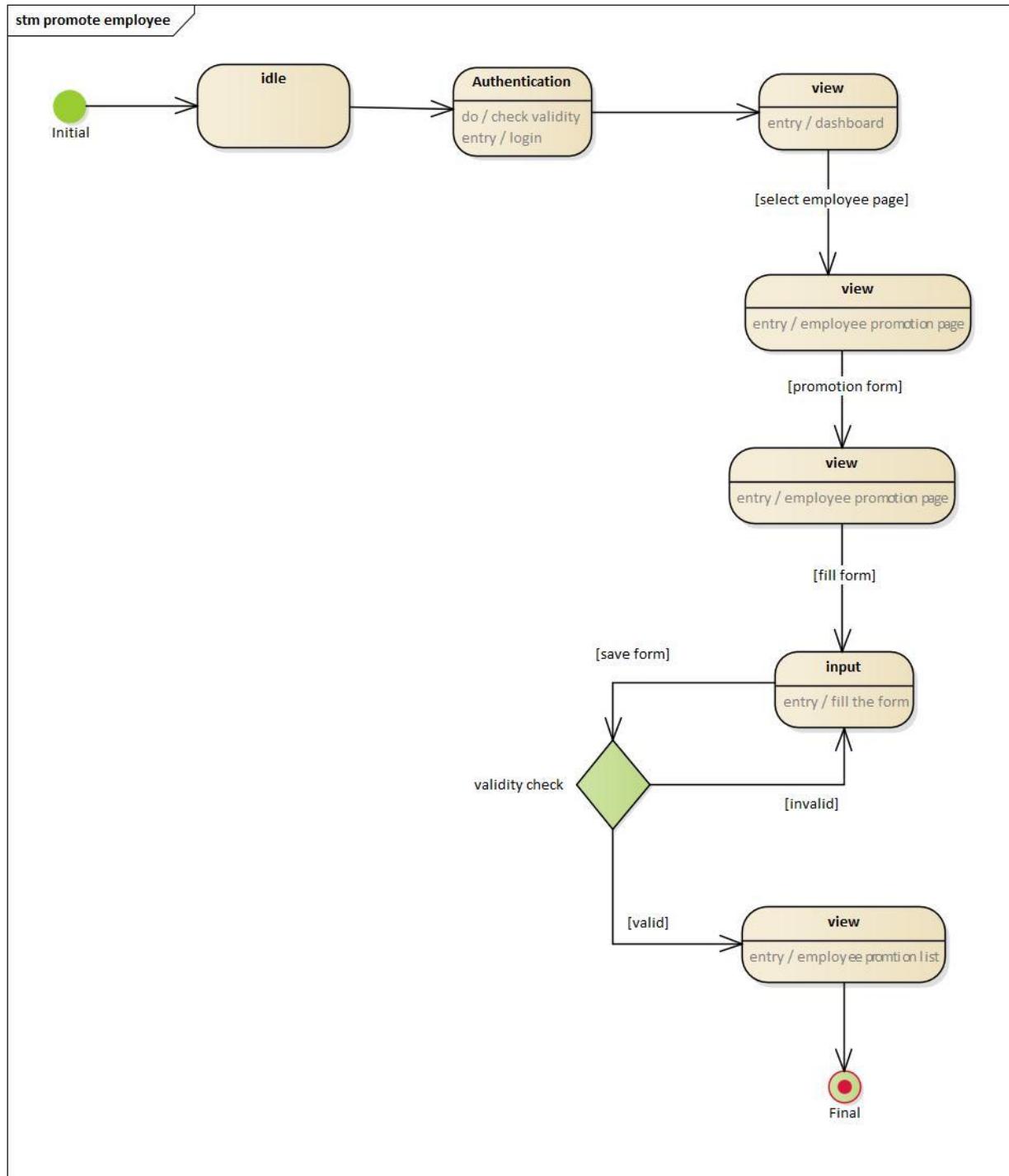


Figure 3.17 Promote Employee State Diagram

## **4. Chapter Four – System Design**

### **4.1. Overview of System Design**

The process of translating software specifications into a representation of system modules, interfaces, and data required for the implementation phase is known as software design. The System Design Document lays out how the software system will be built to meet the specifications. It is the primary source of code creation material, so it must provide all a programmer needs to write code. There are two steps for this. the first step is to create a preliminary design that defines the overall system architecture and data architecture. More complex data structures are described and algorithms are built in the second stage, referred to as the detailed design stage.

The proposed enterprise resource management system plans to improve the current problems that are faced in the enterprise and other similar organizations. Based on the described requirements and models portrayed in the preceding chapters, this section of the documentation aims to map those requirements into an actual design of the system. This chapter will generally explore the system design by identifying the architecture, process, and components of the system and their interaction.

#### **4.1.1. Purpose of the System Design**

System design aims to provide a complete image of the proposed system's architecture. Here, the requirement specifications are translated into system features, interfaces, processes, and data outputs. This system design expresses the requisite building blocks that are necessary to build and realize the system. It also provides the basic implementation aspects of the system.

This system should adhere to some kind of enterprise structure and organizational principles since it is essentially an enterprise resource management system focused on human resources and finance. As a result of the systematic design approaches that will be used, the system will be able to become more adaptive to a higher level of operation, refactoring, improvements, and maintainability.

#### **4.1.2. Design Goal**

The design goal specifies the major qualities that a system needs to achieve. The design goal here is to identify the qualities the system should focus on. These qualities are mostly referred to from the non-functional requirements. When designing this enterprise resource management system, there are several factors to consider for development. The importance of each consideration should be based on the system's priorities and objectives.

##### **4.1.2.1. Performance**

The system should perform with a practical standard of processing time and memory. It should respond with an acceptable range of time and memory. Tasks performed on the system should get processed and returned as quickly as possible in terms of response time and processing memory.

- **Response Time:** The system should have an average response time of one second. Depends on the available processing power, network bandwidth, and hardware specification. Page render and reload are handled using the CodeIgniter framework cache and dynamic instantiation which manipulates the virtual DOM swiftly.

- **Memory Space:** The system should operate optimally on a client's device with a minimum of 2GB RAM and a Dual Core CPU Processor. The on-premise server where the system will be deployed should have the following minimum requirements:
  - 4GB RAM
  - Quad-Core CPU Processors
  - 2TB Hard disk
  - Network speed of 1MBps

#### **4.1.2.2. Dependability**

The dependability of the proposed enterprise resource management system is focused on the framework, libraries, tools, and techniques that are utilized to deliver the functionalities and services. It explores the following aspects:

- **Security:** CodeIgniter fundamentally handles or prevents common security problems such as SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF). [\[10\]](#) The system is an enterprise system handling sensitive corporate data is equipped with a standard authentication system with login credentials. The system would also consist of roles and permissions-based access control mechanism where user accounts with specific roles can be created dynamically by enabling or disabling the module along with creating, view, update, delete and approve permissions.
- **Robustness:** this criterion is considered when an error occurs on the system. When there is a sudden termination of processes, the system will handle it by notifying the user as well as following the exception flow of events tasked by the robust programming built-in. CodeIgniter is the framework that is used to develop the backend of the system which consists of a robust programming approach. The error functions in CodeIgniter are basic procedural interfaces that are available globally in the application, unlike most other systems. This method eliminates the need to think about class/function scoping when triggering error messages.
- **Reliability:** The system should provide reliable service throughout all business hours and maintenance periods. The system is planned to be deployed on Apache Webserver. It has been regarded as the most reliable and popular HTTP server in the entire Internet space since April 1995. Apache's load balancing feature enhances its reliability. It's also worth noting that it was served by over 305 million sites in 2016. [\[11\]](#)
- **Availability:** The system should be available on the enterprise portal and network where it will be deployed on-premise of the enterprise. An internet connection is required to access the system outside the local enterprise network which relies on the national network grid.
- **Fault Tolerance:**
  - When sections of the system or subsystem fail, the system will be able to retain its functionalities and services due to robust programming and error handling.
  - Furthermore, the Apache webserver is equipped with load balancing and bandwidth throttling features that handle incoming traffics and respond accordingly.

#### **4.1.2.3. Maintenance**

The system should be maintainable to handle future updates and upgrades.

- **Modifiability:** If a particular feature or functionality needs to be modified, that particular feature should be updated separately without causing issues for other functionalities. This is achieved through the utilization of a modular development approach which is used in this system design. This would enable the system to be maintained in the future.
- **Adaptability:** In order to adapt the changing environment and platform, the system should not specifically rely on a certain trend. Because of its simplicity, CodeIgniter framework is a popular choice among developers. It's simple at its core, and it's complemented by a powerful collection of libraries that make it ideal for full-fledged web development.
- **Extensibility:** To include a new feature in the system, the feature should be developed as a distinct subsystem or module where it can be integrated into the original system with little overhead. Moreover, the modular-based approach along with CodeIgniter's extensibility through the use of native or custom libraries, helpers, or class extensions or hooks enhances the extensibility and scalability of the system. This is an essential design consideration as the system is going to be developed and extended over time, with more modules being integrated.
- **Portability:** The system is designed to be applicable in various environments as it is a web-based system, it is supported by any platform with a web browser. The system is also designed to be an open system where it can be deployed in any similarly structured organization other than the model organization EMPDE and deliver the same functions and services there as well.
- **Flexibility:** The system should bring about a well-thought-out plan to perform a certain task in more ways than one, accommodating optional methods. For instance, provide a search and filter option, perform a single and bulk operation, custom fields option, and parameter assignment options.

#### **4.1.2.4. End-User**

An end-user is ultimately the one who uses the system or who is the consumer or client of the product.

- **Usability:** The system will be interactive, easy-to-use, and understandable; accompanying a simple user-friendly interface with navigation, descriptive names, sidebars, themes, and settings. The system will provide a consistent design for usability. The system would also provide its services in local languages such as Amharic and Afaan Oromo which increases its usability. The dashboard is also another aspect to enhance usability which delivers statistics, trends, and updates graphically and interactively.
- **Ease of Learning:** The system will keep standard layouts and patterns which would simplify the learning process. To create a shallow learning curve and facilitate the transition from the legacy system, the design considers easing the learning process with best practices and approaches.
- **Low cost:** The system will be sold at a reasonable cost which would be very less than any other foreign commercial off-the-shelf or custom-built systems.

## 4.2. Proposed System Architecture

The high-level structures of a software system, as well as the discipline of designing such structures and systems, are referred to as software architecture. Each structure is made up of software elements, their relationships, and the properties of both the elements and the relationships. A software system's architecture is a metaphor, comparable to the architecture of a building. It serves as a blueprint for the framework and the ongoing project, setting out the tasks that the design teams must complete.

The proposed enterprise resource management system follows the variation of Model-View-Controller (MVC) architecture which is known as Hierarchical Model-View-Controller (HMVC). The HMVC pattern is an evolution of the MVC pattern, which is currently used in the majority of web applications. It was created in response to the scalability issues that were evident in MVC-based applications. The traditional Model, View, and Controller triad will be layered into a "hierarchy of parent-child MCV layers," according to a solution presented on the JavaWorld website in July 2000.

Developers have developed many frameworks and design trends with varying degrees of success to mitigate the risk associated with creating a robust client tier. The Model-View-Controller (MVC) pattern is still one of the most common. When it comes to the control of GUI components, however, the standard MVC scope falls short (widgets).

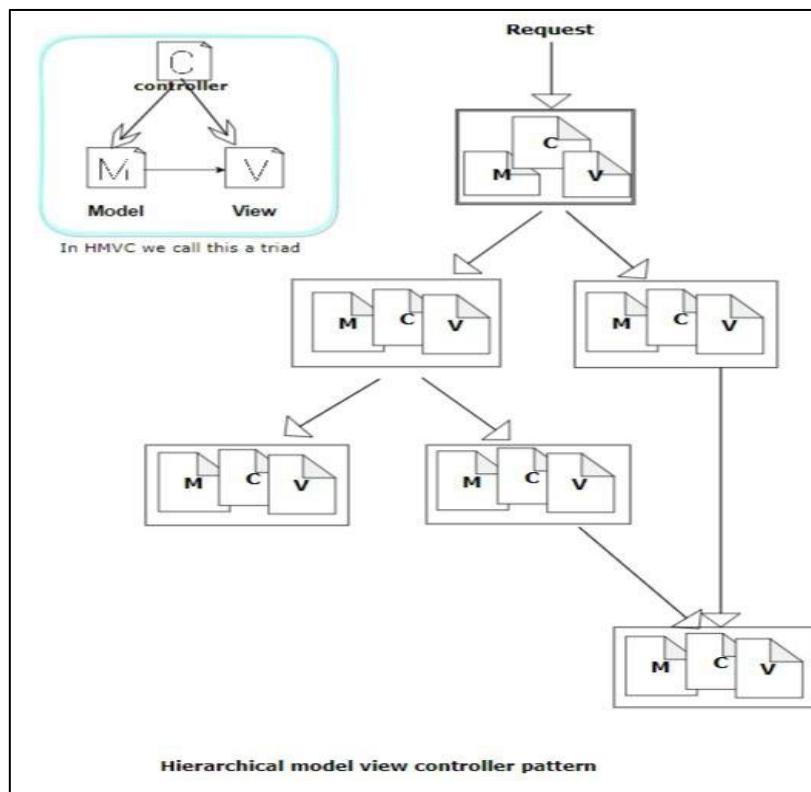


Figure 4.1 Hierarchical Model-View-Controller (HMVC) illustration

Model is usually based on real-world scenarios. This code may either carry raw data or describe the app's critical components. View consists of all the functions that communicate directly with the user. This is the code that determines how your user sees and interacts with your software, as well as how it looks. Controller connects the Model and the View, collecting user feedback and deciding what to do about it. It's the application's brain, and it connects the model and the vision.

The nuances of data processing, event management, and application flows are not handled by MVC. The HMVC (Hierarchical-Model-View-Controller) paradigm, which is an extension of the MVC triad, aims to address some of these issues. Thus, the following are some of the benefits we considered while using the HMVC Pattern in our design and development cycle: [\[12\]](#)

- Organization: having a folder for each of the related triads reduces the amount of work that needs to be done.
- Reusability: because of the architecture, virtually any piece of code can be reused.
- Modularization: reducing dependencies between different sections of an application.
- Extensibility: increases the application's extensibility without losing its ease of use.

Each triad works independently of the others. A triad's controllers may request access to another triad. Both of these points make it possible to distribute the application through several locations if necessary. The layering of MVC triads also enables more detailed and robust application development. Although MVC only allows for one controller per request, HMVC allows for multiple controllers to be combined with multiple triads. It simplifies the testing of disparate parts of the framework while ensuring the application's enhancement by allowing existing code to be reused.

The HMVC architectural pattern in CodeIgniter addresses the scalability problems that affect the MVC pattern. Users end up duplicating and swapping templates, making their widget frameworks or library files or using custom helpers because the MVC does not address content structure issues.

In a generic MVC web application, the URL request is handled by a routing portion. This request will be mapped to a Controller action, which will invoke business logic before being transferred to the View for rendering. This is fine for simple applications, but as your application becomes more complex, you'll find that you'll need to share features between controllers. This is one of the reasons why HMVC is chosen as a system architecture for the design and development of the proposed system.

Tatari System follows a three-tier high-level system architecture which consists of:

- ✓ Client Tier – this layer is a collection of system users which include system administrator, head of department, employees, and candidates.
- ✓ Business Logic / Application Tier – is the layer where the system with all its functions and business logic resides. It is used to serve the clients with the services they require.
- ✓ Data Tier – is the layer where all the data and collected enterprise information by the system is stored centrally.

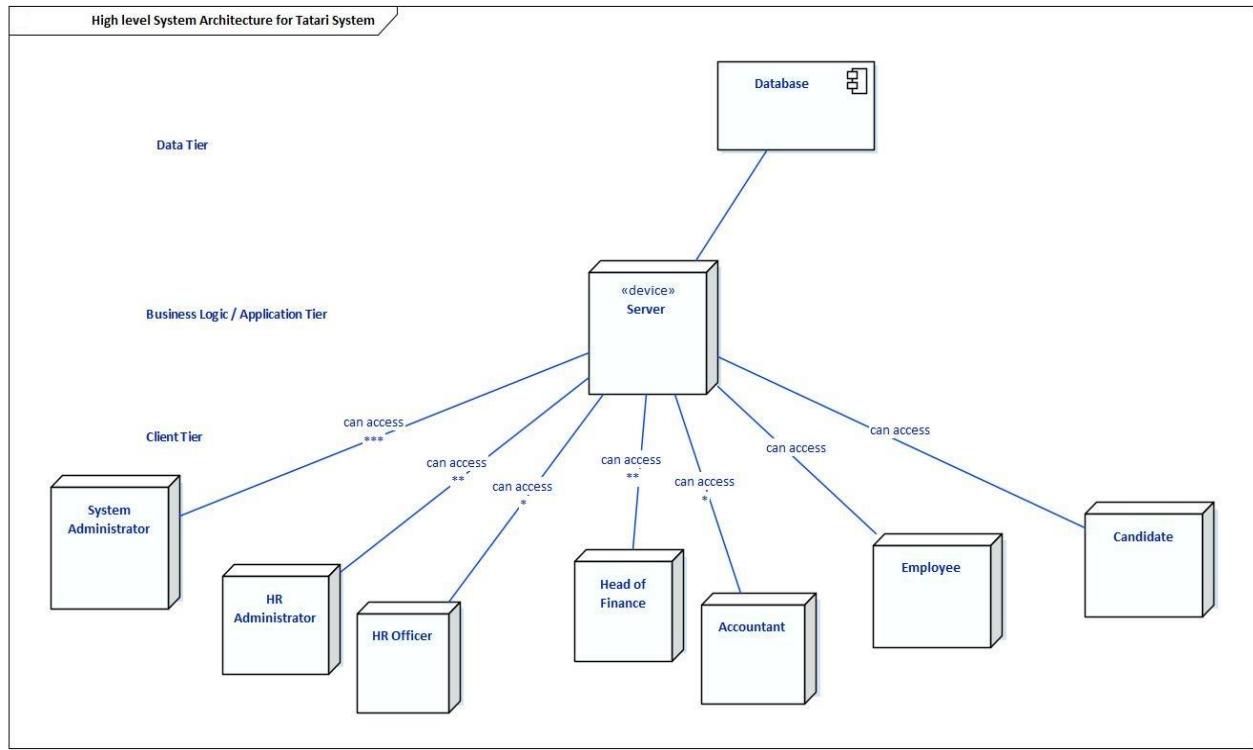


Figure 4.2 Tatari System High-level System Architecture

#### 4.2.1. System Process

The following diagrams illustrate of the system process from the Human Resources module perspective of the Tatari system.

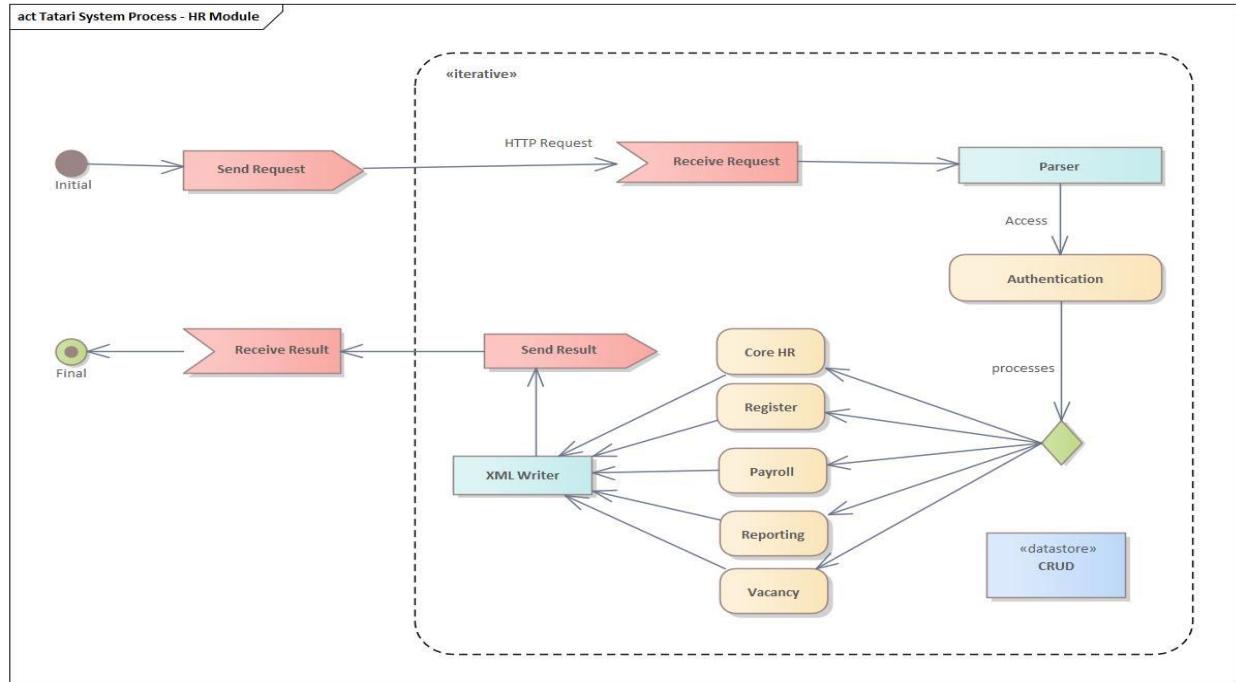


Figure 4.3 System Process - Tatari HR Module

The following diagrams illustrate of the system process from the Finance module perspective of the Tatari system.

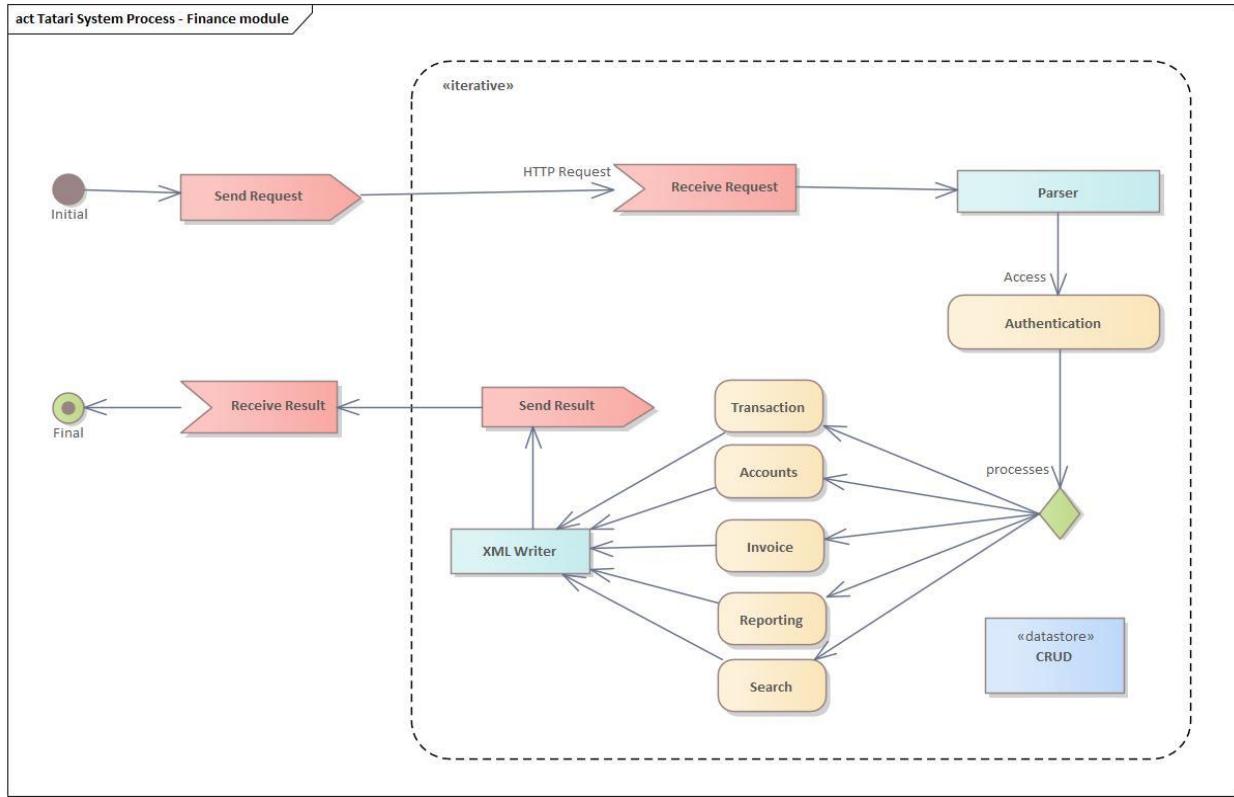


Figure 4.4 System Process - Tatari Finance Module

#### 4.2.2. Subsystem Decomposition

The degree to which components of a system depend on one another is referred to as coupling. The system becomes more reusable and modular as the number of interdependent components decreases. We intended to create a system that was loosely coupled to enhance the scalability of the system, as there are plans to integrate more modules into the system.

Layers and partitions are often used to break down a complex and large system such as this enterprise resource management system into subsystems. Partitions divide a system vertically into many isolated (or weakly-coupled) subsystems that provide services at the same abstraction level. Decomposing the system is beneficial for increasing performance and reducing system complexity. The object model's classes and objects serve as the seeds for the subsystems. There are fifteen (15) major subsystems in this system.

The core/basic subsystems that manage the overall schema and integrity of the system are the ones found at the center of the following diagram such as Organization, Access/Role, Settings and Authentication. The other subsystems found on the right and left side are for the HR and Finance module respectively.

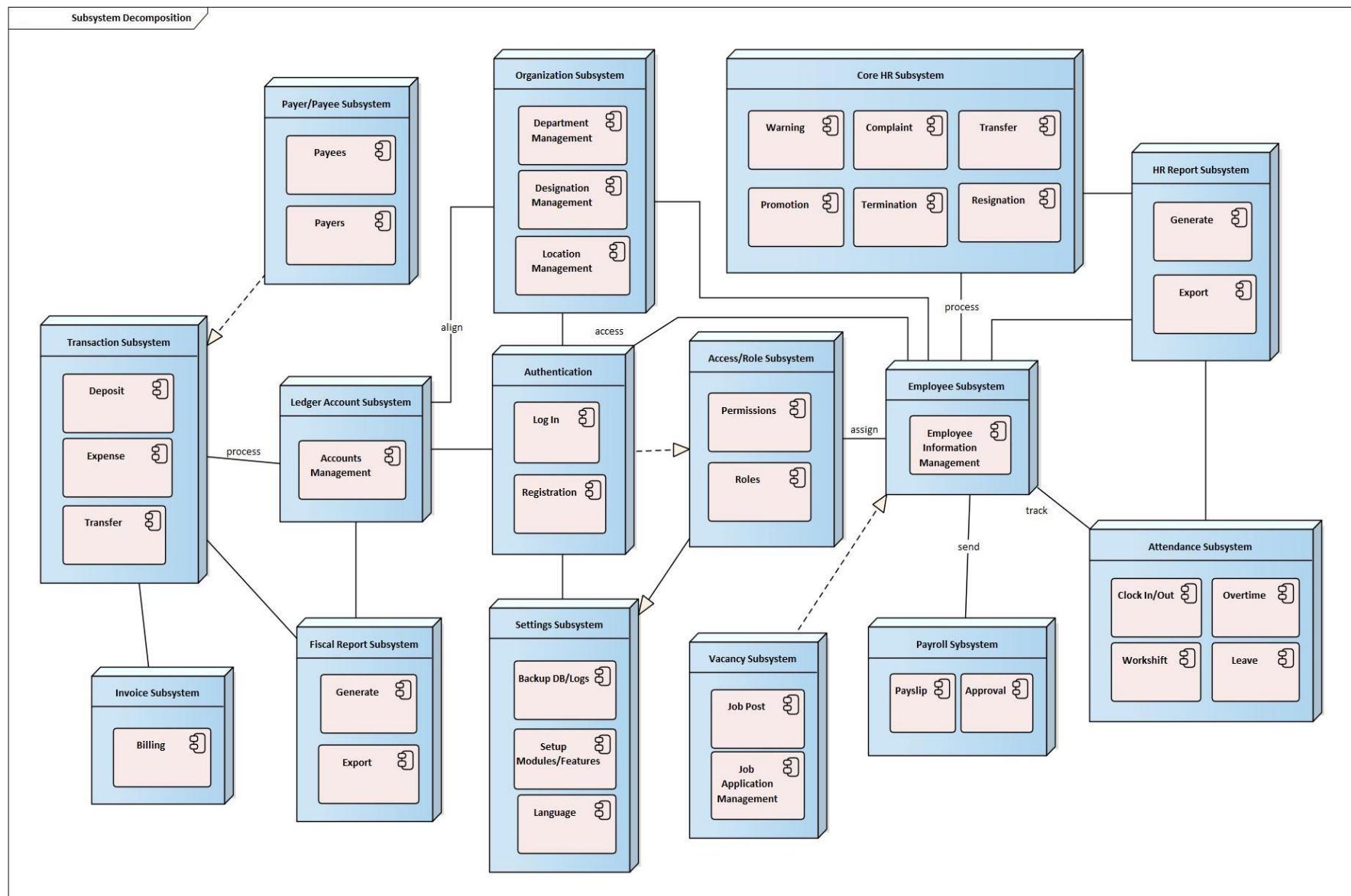


Figure 4.5 Subsystem Decomposition

The following table describes the purposes and services of the subsystems along with their classes.

	<b>Subsystem</b>	<b>Purpose / Services</b>	<b>Class / Object</b>
<b>Core Subsystems</b>	Organization Subsystem	To create the organizational structure and departments that it runs	Company, Department, Sub-department, Designation
	Authentication Subsystem	To authenticate users into the system with login credentials and user accounts	Employee, User roles
	Access/Role Subsystem	To manage roles and permissions of system users with proper access control	Employee, Designation, User roles
	Settings Subsystem	To manage system settings like modules, backup, language, custom fields	System data, backup, logs, languages
<b>HR Module Subsystems</b>	Employee Subsystem	To manage employee information and records with user accounts and profiles	Employee, Designation
	Core HR Subsystem	To handle core HR processes such as termination, transfer, resignation, promotion	Promotion, Termination, Warning, Resignation, Transfer, Complaint, Warning
	Payroll Subsystem	To process payroll and other salary payment operations	Salary, Payroll, Payslips
	Attendance Subsystem	To track attendance and work schedules, employees	Attendance, Shift, Leave
	Vacancy Subsystem	To manage job post, job applications and recruitment of candidates	Job post, Job applications, Candidates
	HR Report Subsystem	To generate consolidated HR reports based on collected data from the user and system	Attendance, Employee, Payroll, Leave
<b>Finance Module Subsystem</b>	Ledger Account Subsystem	To manage general ledger accounts of the organization	Ledger Account
	Transaction Subsystem	To process financial transactions such as deposit, expense and transfers	Finance Transfer, Deposit, Expense
	Invoice Subsystem	To generate invoices and bills for transactions and processed payments	Invoices
	Payer/Payee Subsystem	To register and manage payer and payee or cashier and client information	Payer, Payee
	Fiscal Report Subsystem	To generate consolidated fiscal reports based on financial transactions	Deposit, Expense, Transfer

Table 4.1 Subsystem Services and Purpose

#### 4.2.3. Hardware/Software Mapping

To provide its services, the proposed enterprise resource management system will make use of several hardware devices and software applications. The servers are equipped with Apache web servers to host PHP-based web applications, which are the foundation of the CodeIgniter project. Data storage and manipulation are handled by MariaDB MySQL database servers, which operate the database operation. A load balancer enterprise server with Apache, which also serves static content and functions with a firewall, is used to manage the operations of these servers. To access Tatari System from the client's perspective, all that is needed is a web browser.

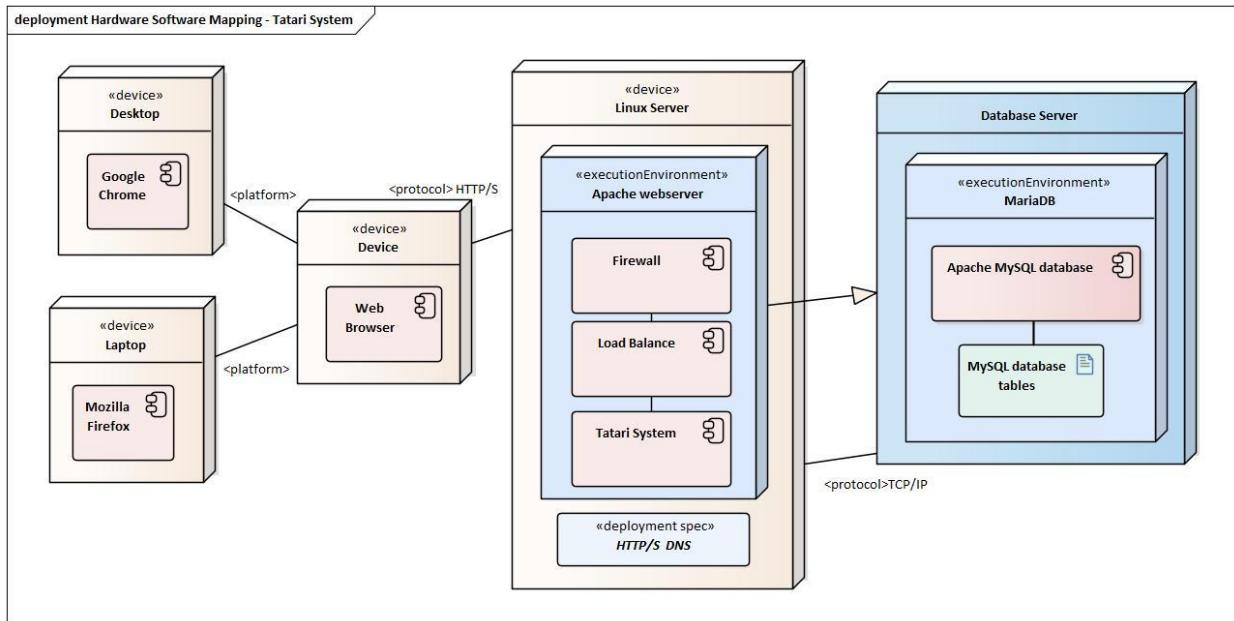


Figure 4.6 Hardware and Software Mapping / Deployment Diagram

#### 4.2.4. Persistent Data Management

This section provides a mapping of the class diagram's classes and objects that were identified in the requirement analysis phase into a relational database format.

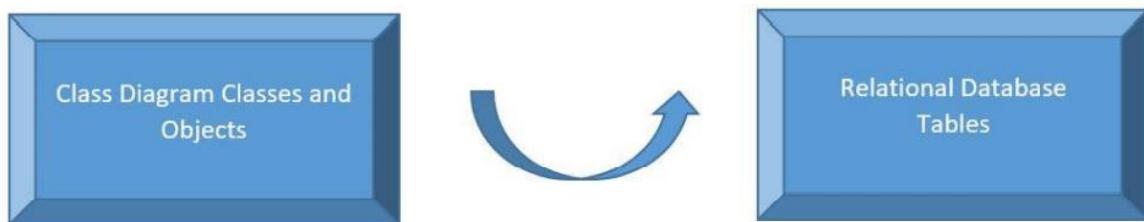


Figure 4.7 Persistent Data Management

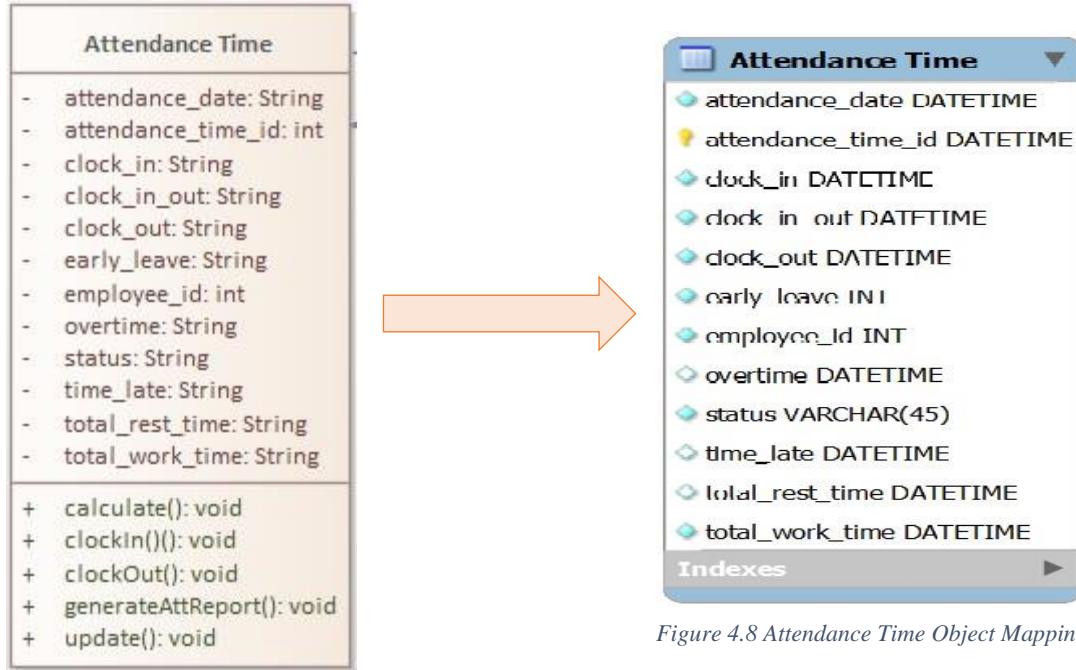


Figure 4.8 Attendance Time Object Mapping

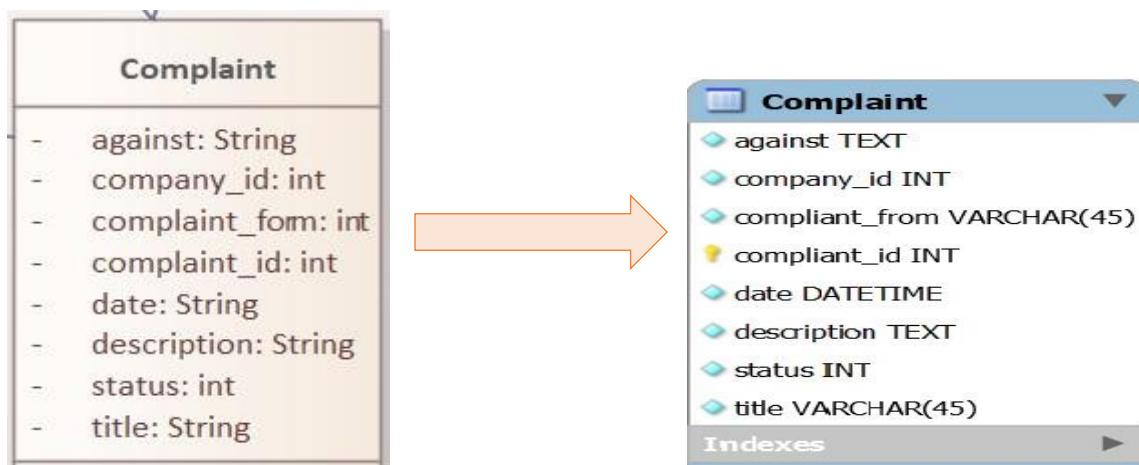


Figure 4.9 Complaint Object Mapping

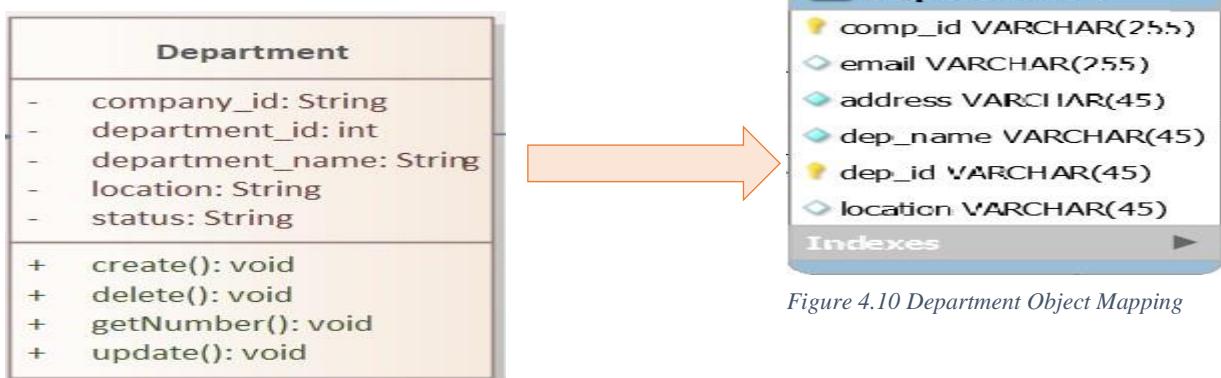


Figure 4.10 Department Object Mapping

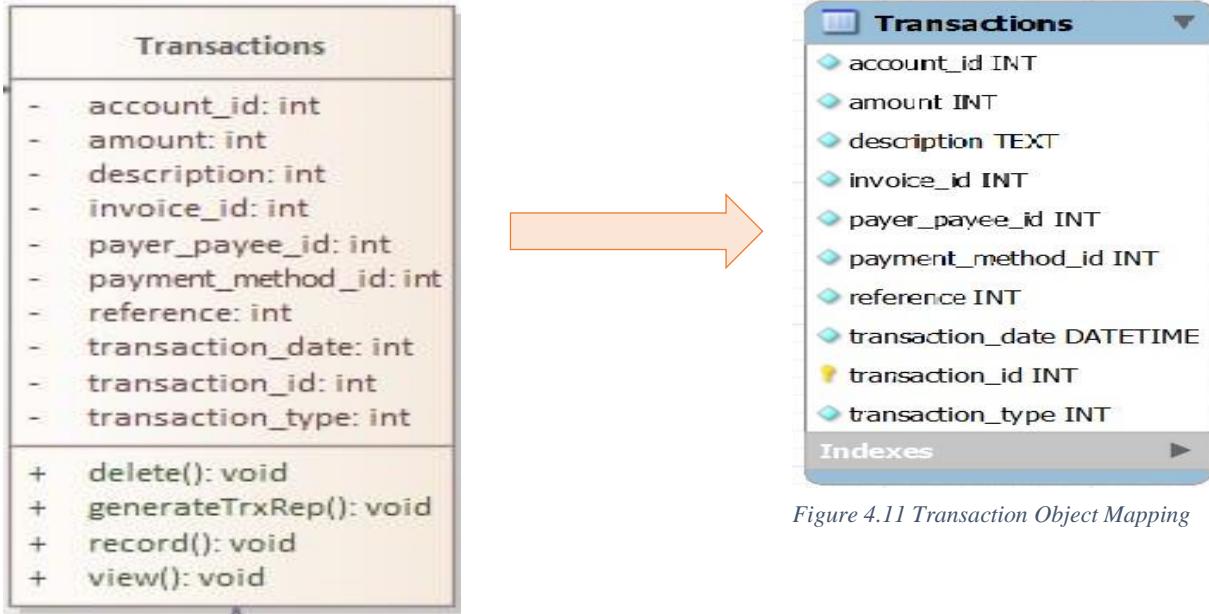


Figure 4.11 Transaction Object Mapping

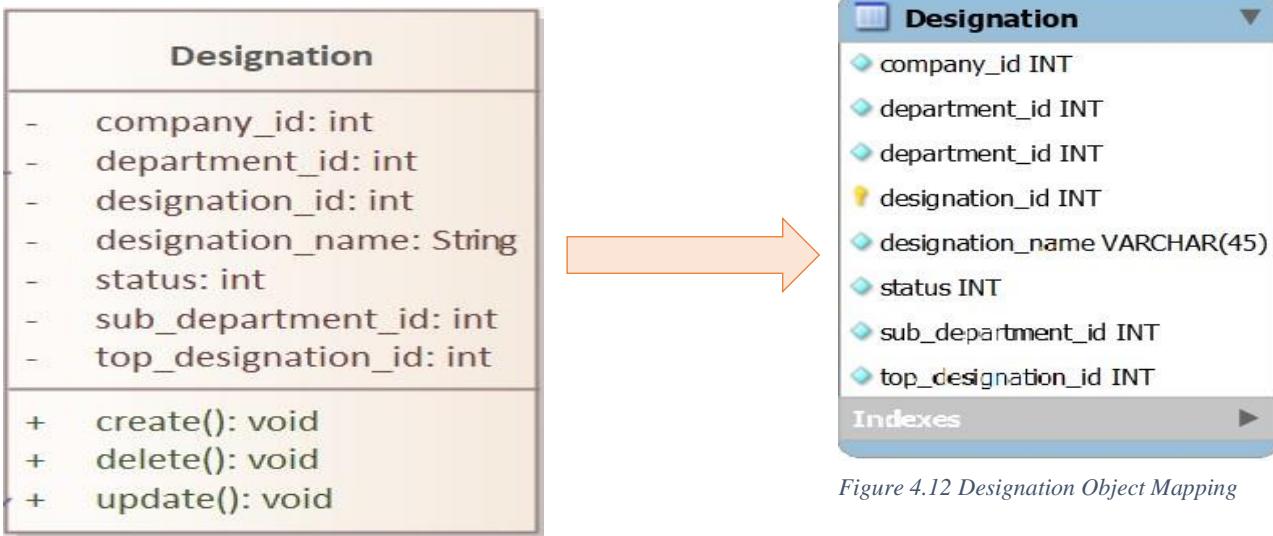


Figure 4.12 Designation Object Mapping

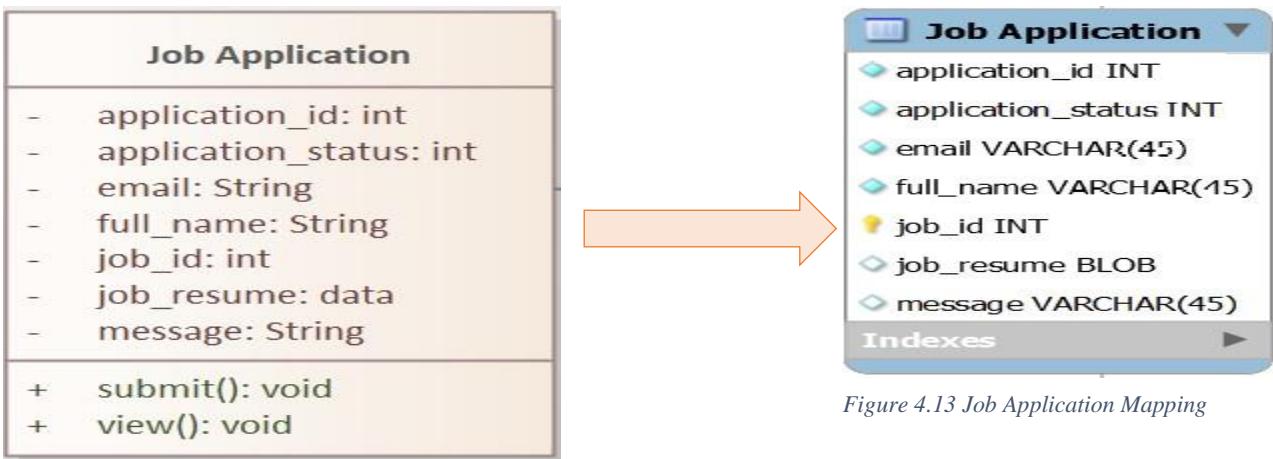


Figure 4.13 Job Application Mapping

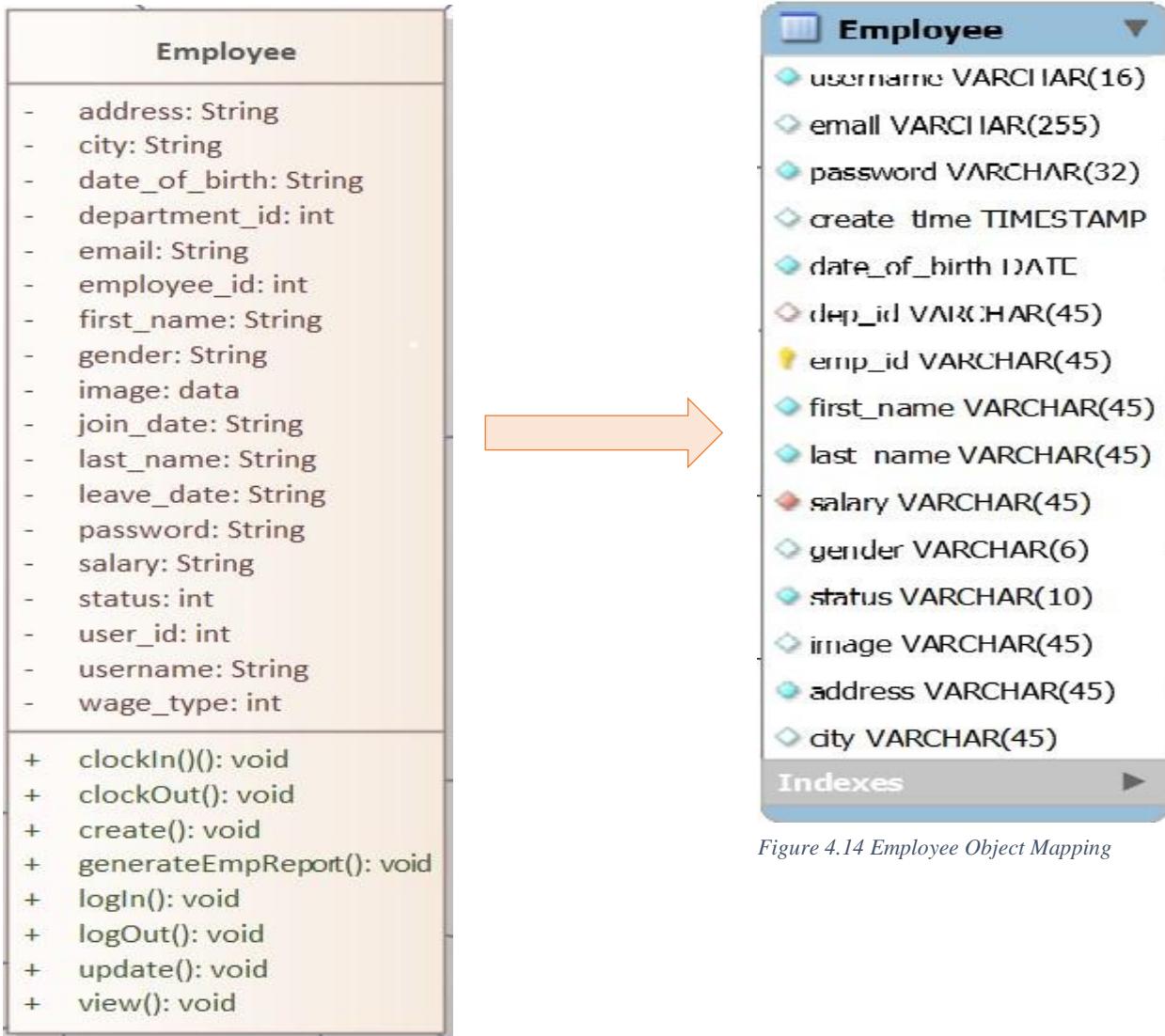


Figure 4.14 Employee Object Mapping

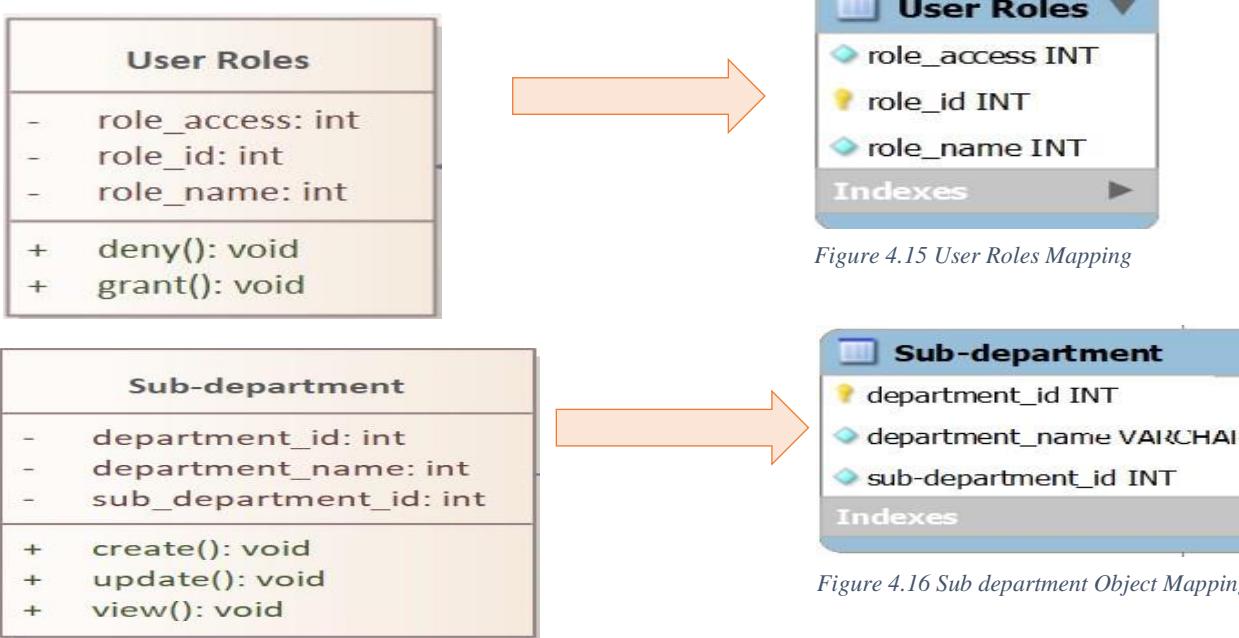


Figure 4.15 User Roles Mapping

Figure 4.16 Sub department Object Mapping

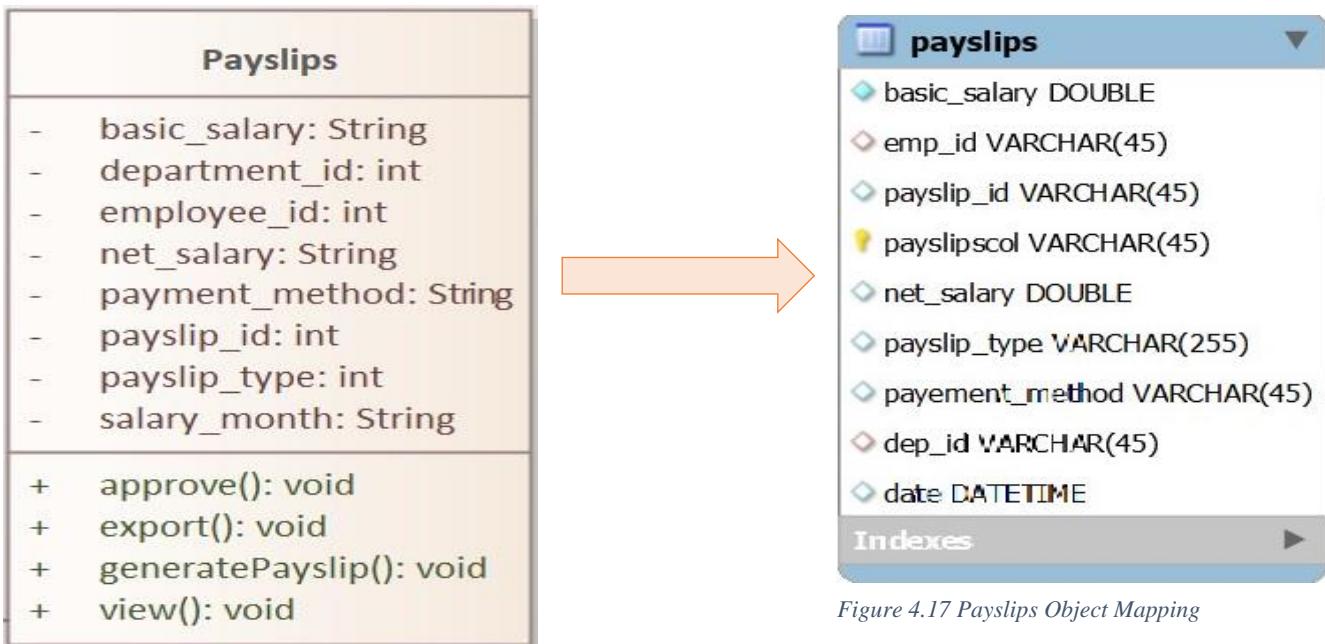


Figure 4.17 Payslips Object Mapping

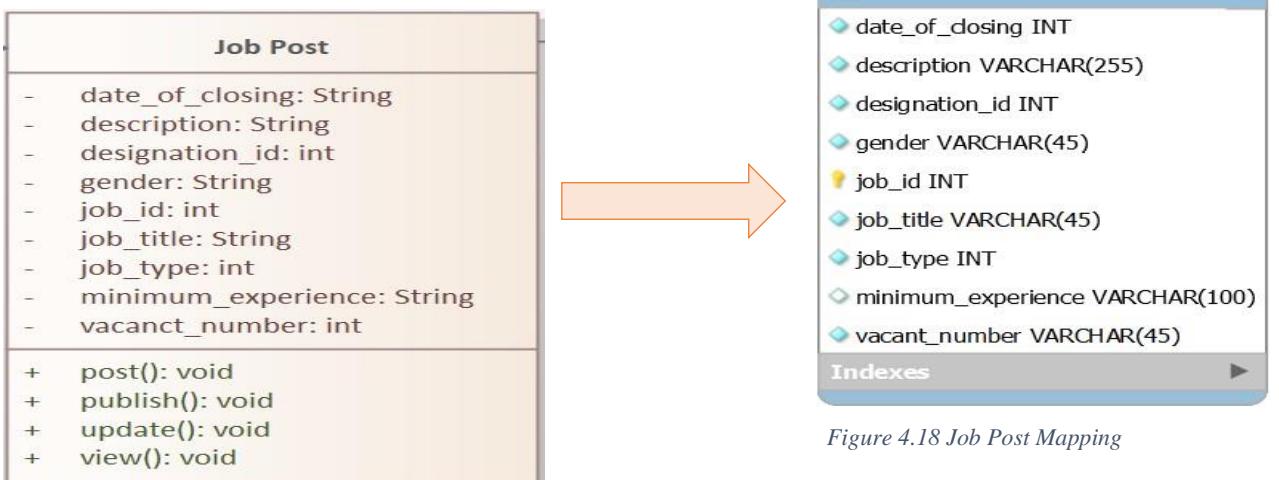


Figure 4.18 Job Post Mapping

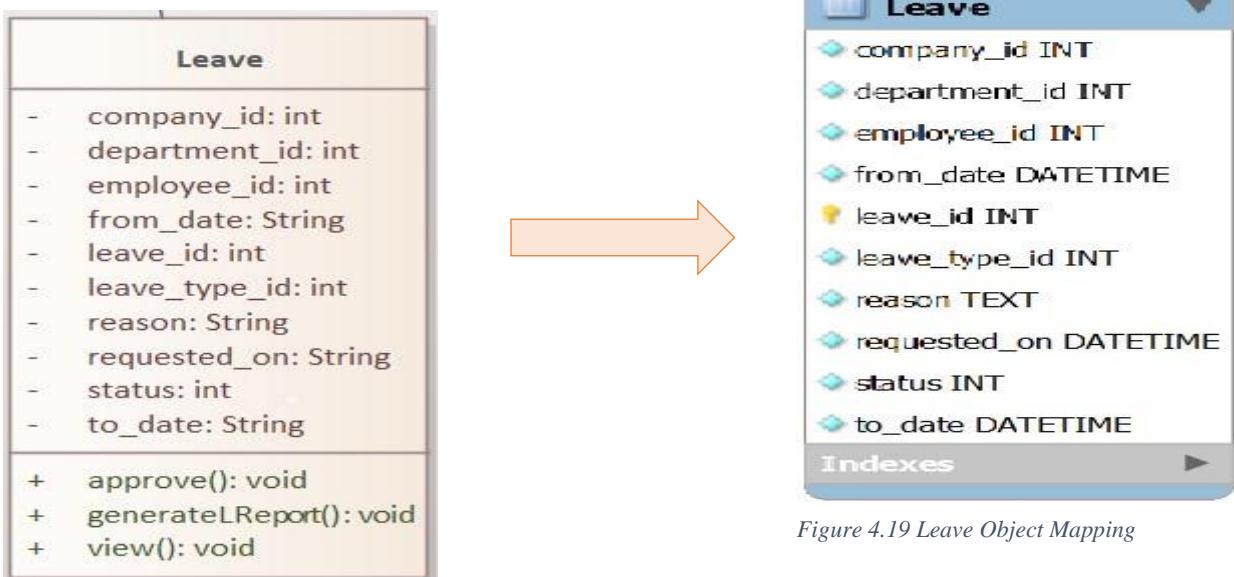


Figure 4.19 Leave Object Mapping

Promotion	
-	company_id: int
-	description: String
-	employee_id: int
-	promotion_date: String
-	promotion_id: int
-	title: String
+	demote(): void
+	promote(): void
+	request(): void



Promotion	
◆	company_id INT
◆	description TEXT
◆	employee_id INT
◆	promotion_date DATETIME
◆	Promotion_id INT
◆	title VARCHAR(45)
Indexes	

Figure 4.20 Promotion Object Mapping

Resignation	
-	company_id: int
-	employee_id: int
-	notice_date: int
-	reason: int
-	resignation_date: int
-	resignation_id: int
+	get(): void
+	request(): void
+	resign(): void
+	update(): void



Resignation	
◆	company_id INT
◆	employee_id INT
◆	notice_date DATETIME
◆	reason TEXT
◆	Resignation_id INT
Indexes	

Figure 4.21 Resignation Object Mapping

Ledger Account	
-	account_balance: String
-	account_id: int
-	account_name: String
-	account_number: String
-	bank_branch: String
-	branch_code: String
-	opening_balance: String
+	create(): void
+	deposit(): void
+	expense(): void
+	getBalance(): void
+	transfer(): void
+	update(): void
+	view(): void



Ledger Account	
◆	account_balance FLOAT
◆	account_id INT
◆	account_name VARCHAR(45)
◆	account_number INT
◆	bank_branch VARCHAR(45)
◆	branch_code VARCHAR(50)
◆	opening_balance VARCHAR(45)
Indexes	

Figure 4.22 Ledger Account Object Mapping

Warning	
-	description: String
-	subject: String
-	warning_by: int
-	warning_date: String
-	warning_id: int
-	warning_to: int
-	warning_type_id: int
+	delete(): void
+	get(): void
+	update(): void
+	warn(): void



Warning	
◆	description TEXT
◆	subject VARCHAR(15)
◆	warning_by INT
◆	warning_date DATETIME
◆	warning_id INT
◆	warning_to INT
◆	warning_type_id INT
Indexes	

Figure 4.23 Warning Object Mapping

Shift	
-	end_hour: String
-	from_date: String
-	shift_id: int
-	start_hour: String
-	to_date: String
+	create(): void
+	setDays(): void
+	setHours(): void
+	update(): void



Shift	
◆	end_hour DATETIME
◆	from_date DATETIME
◆	shift_id INT
◆	start_hour DATETIME
◆	to_date DATETIME
Indexes	

Figure 4.24 Shift Object Mapping

Transfer	
-	company_id: int
-	description: String
-	employee_id: int
-	transfer_date: String
-	transfer_department: String
-	transfer_id: int
-	transfer_location: String
+	delete(): void
+	transfer(): void
+	update(): void



Transfer	
◆	comp_id VARCHAR(45)
◆	description VARCHAR(255)
◆	emp_id VARCHAR(45)
◆	transfer_date DATETIME
◆	transfer_dep VARCHAR(45)
◆	transfer_id VARCHAR(45)
◆	transfer_location VARCHAR(45)
Indexes	

Figure 4.25 Transfer Object Mapping

Termination	
-	company_id: int
-	description: String
-	employee_id: int
-	notice_date: String
-	terminated_by: String
-	termination_date: String
-	termination_id: int
+	approve(): void
+	get(): void
+	terminate(): void



Termination	
◆	company_id INT
◆	description TEXT
◆	employee_id INT
◆	notice_date DATETIME
◆	terminated_by VARCHAR(45)
◆	termination_date DATETIME
!	termination_id INT
Indexes	

Figure 4.26 Termination Object Mapping

Expense	
-	account_type_id: int
-	amount: int
-	description: String
-	description: String
-	expense_date: String
-	expense_id: int
-	payee_id: int
-	payment_method: String
-	reference: String
+	deposit(): void
+	generateDepReport(): void
+	generateExpInvoice(): void
+	get(): void
+	process(): void
+	update(): void
+	view(): void



Expense	
◆	account_type_id INT
◆	amount INT
◆	description TEXT
◆	expense_date DATETIME
!	expense_id INT
◆	payee_id INT
◆	payment_method VARCHAR(45)
◆	reference TEXT
Indexes	

Figure 4.27 Expense Object Mapping

Deposit	
-	account_type_id: int
-	amount: int
-	deposit_date: int
-	deposit_id: int
-	description: String
-	payer_id: int
-	payment_method: String
-	reference: int
+	delete(): void
+	deposit(): void
+	generateDepReport(): void
+	generateTrxRep(): void
+	record(): void
+	update(): void
+	view(): void



Deposit	
◆	account_type_id INT
◆	amount INT
◆	deposit_date DATETIME
!	deposit_id INT
◆	description TEXT
◆	payer_id INT
◆	payment_method VARCHAR(45)
◆	reference TEXT
Indexes	

Figure 4.28 Deposit Object Mapping

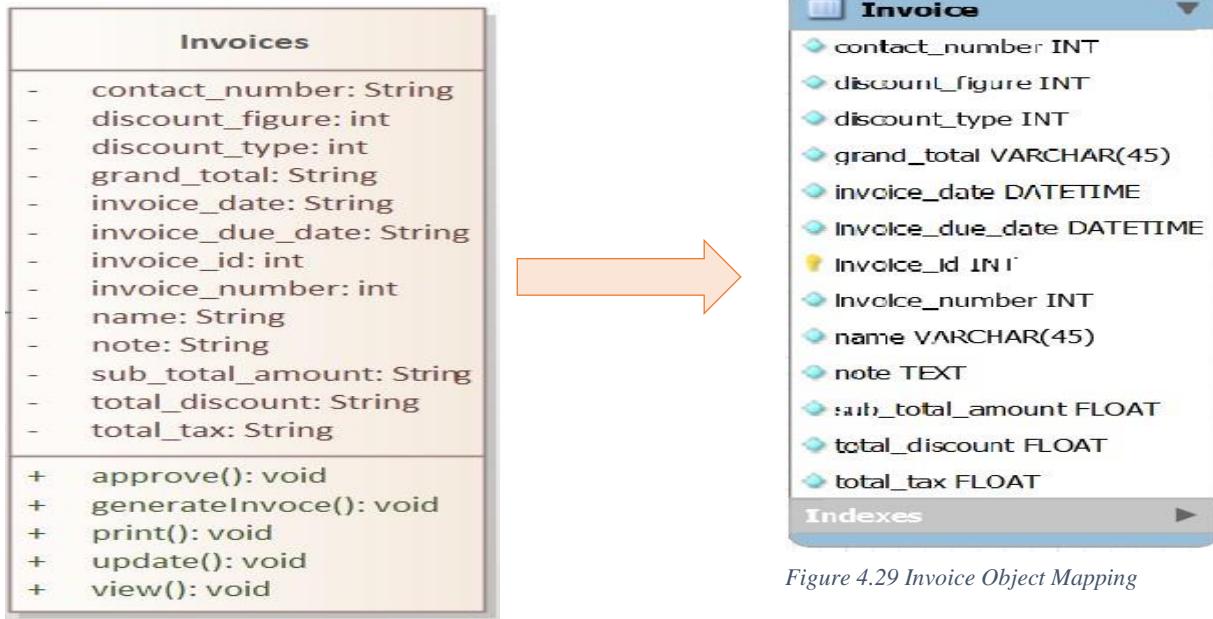


Figure 4.29 Invoice Object Mapping

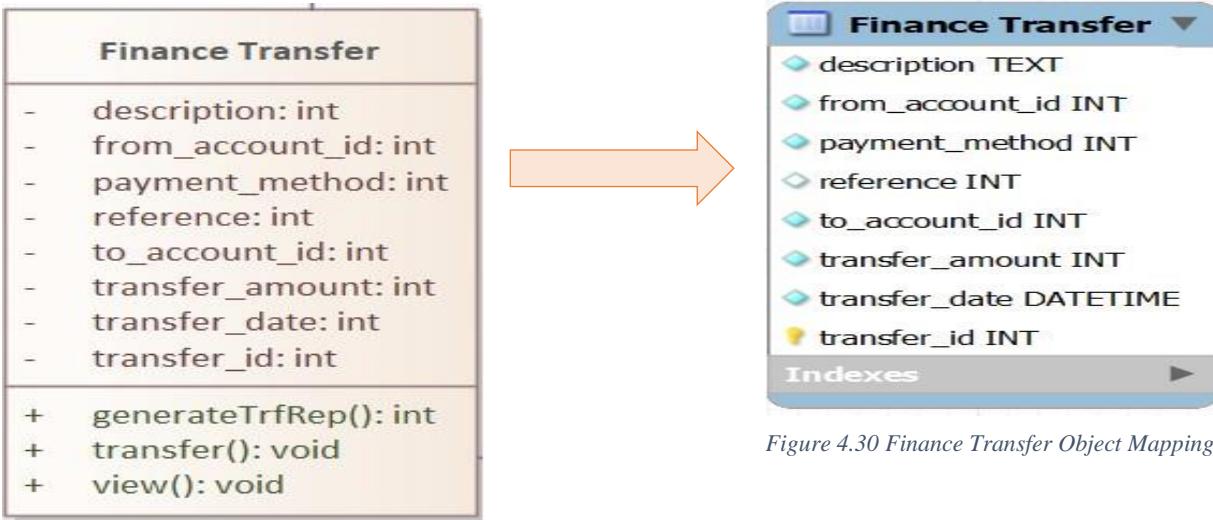


Figure 4.30 Finance Transfer Object Mapping

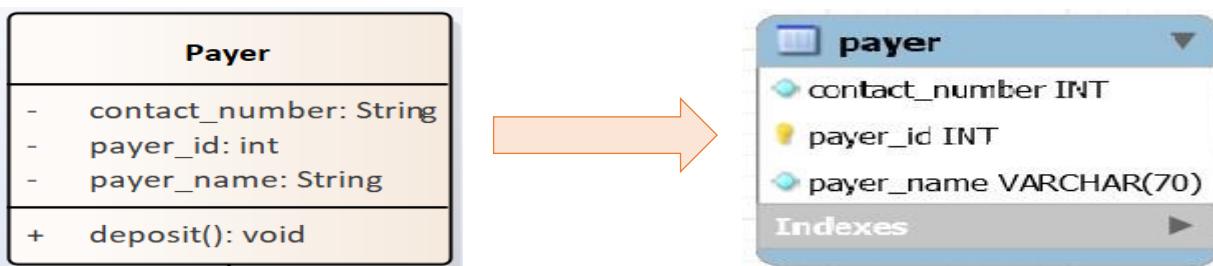


Figure 4.31 Payer Object Mapping

#### 4.2.5. Component Diagram

The following diagrams are component diagrams for individual modules of the Tatari System. It depicts the flow of data, access, and interfaces for communication with databases and security.

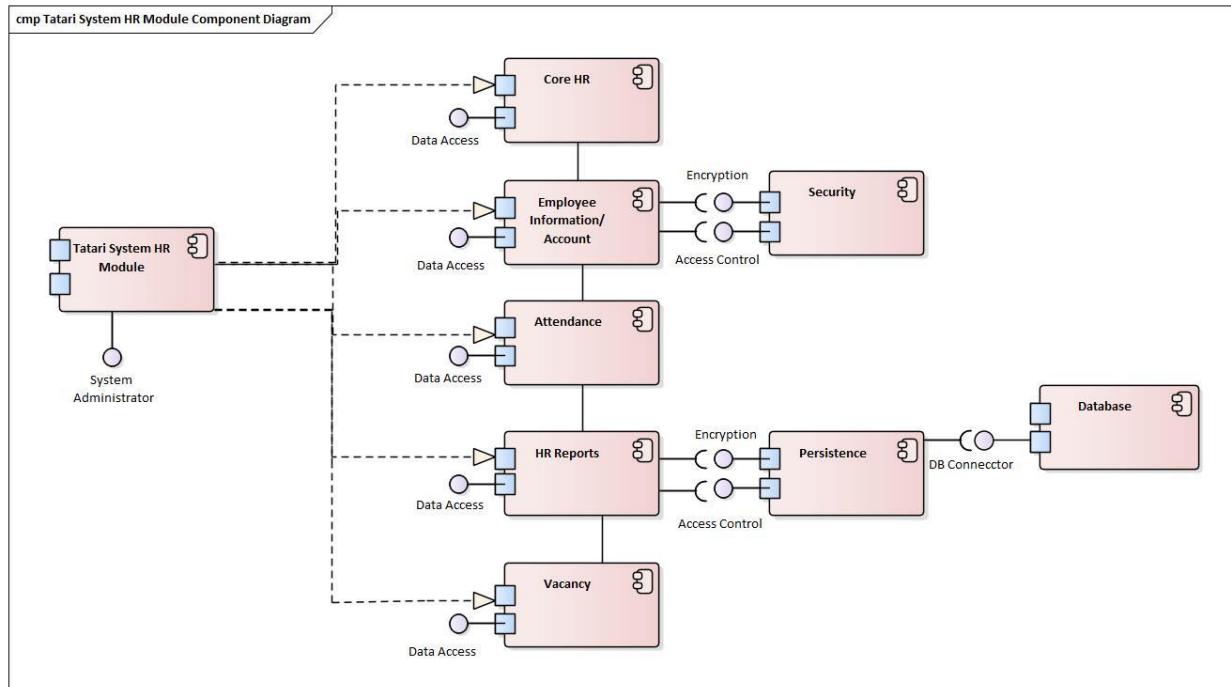


Figure 4.32 Tatari System HR Module Component Diagram

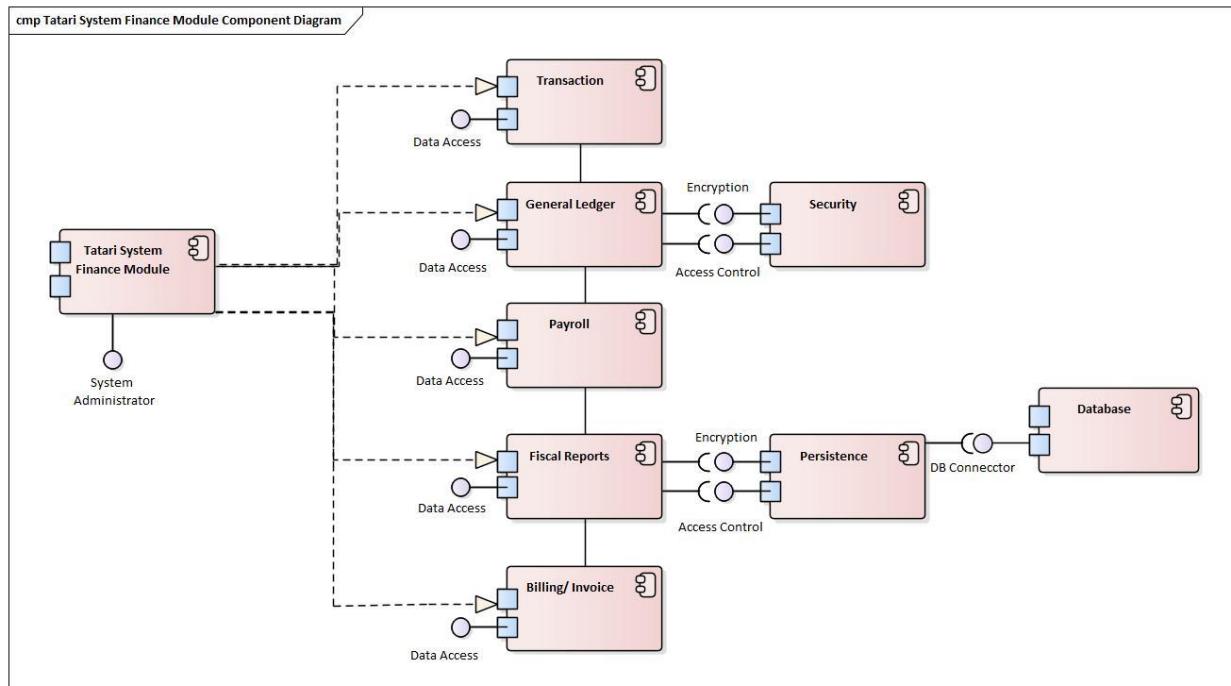


Figure 4.33 Tatari System Finance Module Component Diagram

#### 4.2.6. Database Design

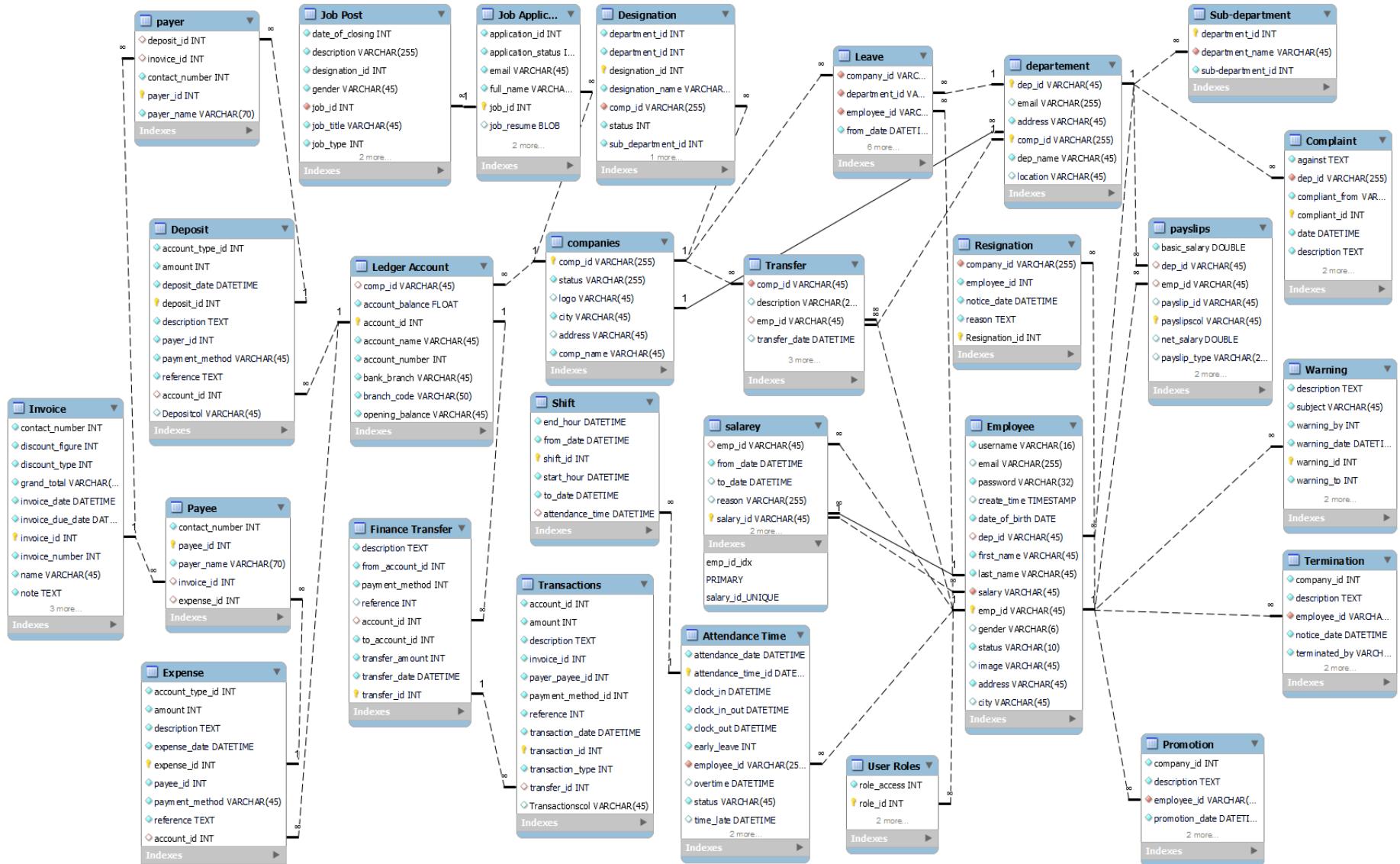


Figure 4.34 Tatari System Database Design (Preliminary)

#### **4.2.7. Access Control**

In a multiuser system such as this enterprise resource management system, access control is one of the fundamental design considerations for system security. Tatari system implements the access control method known as Role-based access control (RBAC). Role-based access control is a method of restricting system access to authorized users.

RBAC (role-based access control) is a policy-independent access-control system based on roles and privileges. Role-permissions, user-roles, and role-role relationships are all RBAC components that make user assignments easy. RBAC can be used in large organizations with hundreds of users and thousands of permissions to make security administration easier. [13]

Roles are generated for different job functions within an organization. Special functions are given permissions to perform specific operations. Members of staff (or other system users) are given specific positions, and those roles grant them the approvals they need to execute specific system functions. Since users do not receive permissions directly from the system, but rather through their position (or positions), managing individual user rights becomes as simple as assigning appropriate roles to the user's account; this simplifies typical operations like introducing a user or changing a user's department.

RBAC is characterized by three primary rules:

1. Role assignment: Only a user who has chosen or been assigned a role may exercise permission.
2. Role Authorization: A user's active role must be approved. This rule, like rule 1, guarantees that users can only take on positions for which they have permission.
3. Permission Authorization: A user may only exercise a permission if it is approved for the user's active position. This law, along with rules 1 and 2, guarantees that users can only exercise permissions for which they have been granted.

Roles can be incorporated in a hierarchy where higher-level roles subsume permissions owned by sub-roles, and additional restrictions can be added.

The proposed system adheres to the Role-based access control principles in a hierarchical manner which is the business logic model followed by the model enterprise as well as many other organizations. This role-based access control will be developed as a subsystem to manage the different roles and their permissions across the system. It is going to consist of a dynamic roles and permissions assignment feature for any number of roles that would be created that arise from various positions from the organization. The system architecture HMVC along with CodeIgniter's libraries are ideal and would be utilized to develop this particular security feature as a subsystem.

This access control system can also implement defined roles with access based on the system model, which would include the different actors that have already been identified. The organization can add any number of roles as they see fit in the future for any number of positions. This also facilitates system extensibility, as more modules are integrated into the system, roles and permissions management will stay intact as it already accounts for scalability with its dynamic roles and permissions access control mechanism.

The following section presents the pre-defined permissions and privileges for the system actors that have been preliminarily identified without accounting for any number of roles that might arise in the enterprise environment.

- a) System Administrator: This user has virtually complete access to the system since the development is primarily made from this perspective. However, following the business model rule and the user's defined responsibilities, it has the following privileges:
  - Log In/Log Out
  - Create Organization
  - View, Update and Delete Organization
  - Create Department
  - View, Update and Delete Department
  - Enable/Disable Module
  - Add Employee
  - View, Update and Delete Employee
  - Create Role
  - Assign Permission
  - View, Update and Delete Role and Permission
  - View System Logs
  - Backup Database
- b) Human Resources Administrator: This user has complete access to the Human Resource module of the system. As per the business model and defined responsibilities by the organization, it has the following privileges:
  - Log In/Log Out
  - Add Employee
  - View, Update and Delete Employee
  - Promote Employee (Add, View, Update and Delete Promotion Record)
  - Transfer Employee (Add, View, Update and Delete Transfer Record)
  - Resign Employee (Add, View, Update and Delete Resignation Record)
  - Terminate Employee (Add, View, Update and Delete Termination Record)
  - Manage Complaint (Add, View, Update and Delete Complaint Record)
  - Manage Warning (Add, View, Update and Delete Warning Record)
  - Generate Timesheet
  - Manage Attendance (Update, View and Import Attendance Record)
  - Manage Work shift (Add, View, Update and Delete Work shift)
  - Manage Overtime (Add, View, Update and Delete Overtime Record)
  - Manage Leave (Add, View, Update and Delete Leave Record)
  - Approve Payroll
  - Generate Payslips (Create, View and Delete Payslips)
  - View, Search, Filter Payroll
  - Manage Job Post (Create, View, Update and Delete Job Post)
  - Manage Job Application (View, Confirm, Reject and Delete Job Application)
  - Generate HR Reports

c) Human Resources Officer: This user has partial access to the Human Resource module of the system. As per the business model and defined responsibilities by the organization, it has the following privileges:

- Log In/Log Out
- Add Employee
- View and Update Employee
- Promote Employee (Add, View, Update and Delete Promotion Record)
- Transfer Employee (Add, View, Update and Delete Transfer Record)
- Manage Complaint (Add, View, Update and Delete Complaint Record)
- Manage Attendance (Update, View and Import Attendance Record)
- Manage Overtime (Add, View, Update and Delete Overtime Record)
- View Resignation Record
- View Termination Record
- View Warning Record
- Generate Timesheet
- View Work shift schedule
- View Leave Record
- View Payslips
- View, Search, Filter Payroll

d) Head of Finance: This user has complete access to the Finance module of the system. As per the business model and defined responsibilities by the organization, it has the following privileges:

- Log In/Log Out
- Create Accounts
- View, Update and Delete Accounts
- Manage Deposit (Add, View, Update and Delete Deposit Records)
- Manage Expense (Add, View, Update and Delete Expense Records)
- Manage Transfer (Add, View, Update and Delete Transfer Records)
- View Transactions
- Manage Payees (Add, View, Update and Delete Payee Records)
- Manage Payers (Add, View, Update and Delete Payer Records)
- Manage Payroll (Verify, View and Process Payroll)
- Create Invoices
- View, Update, Delete Invoices and View Invoice Payments
- Generate Fiscal Reports

e) Accountant: This user has partial access to the Finance module of the system. As per the business model and defined responsibilities by the organization, it has the following privileges:

- Log In/Log Out
- View Accounts

- Manage Deposit (Add, View, Update and Delete Deposit Records)
  - Manage Expense (Add, View, Update and Delete Expense Records)
  - View Transactions
  - Manage Payees (Add, View, Update and Delete Payee Records)
  - Manage Payers (Add, View, Update and Delete Payer Records)
  - Create Invoices
  - View, Update, Delete Invoices and View Invoice Payments
- f) Employee: This user has limited access to the system. It interacts with the system in a small role, however, realizes most core HR process as it is the one who initiates or requests them.
- Log In/Log Out
  - Clock-in Attendance
  - Clock-out Attendance
  - View Profile
  - Change Password
  - Request Overtime
  - View Work History
  - View Payslips
- g) Candidate: This user has external access to the system with a very limited role, which is just submitting a Job Application, which then gets managed by Human Resources, and the candidate gets notified if the application has been selected for further processing.

The above-discussed roles and permissions are derived from the actors and primary system model. Yet, any number of roles can be created by the organization with different permissions using the Role/Permission subsystem. System Administrators can create a new role with permissions and privileges that suit the position. Human Resources can then assign these roles to employees as they see necessary. This would allow for a dynamic access-controlled environment to be more flexible and extensible.

For instance, if the organization decides to give certain access to the General Services department, the System administrator is contacted to create such a role. A role called General Services will be created with privileges like view attendance reports, view staff directory, and create work shift schedule. Human resources can then assign this role to an existing or new employee, this employee will then have the mentioned privileges on the system. This feature allowed for a non-preidentified user/actor to interact with the system in a restricted and defined manner without affecting the general structure and architecture of the system.

Another instance, if the finance department hired a cashier (different from an accountant) to only process deposits. The system administrator can create a cashier role, giving access only to the Deposit feature. Then the cashier can add, update and view deposits. This role has not affected the accountant role, it simply creates a subset role from an accountant and assigned it to a lower-level role which is a cashier. This emanated from a hierarchical business model and HMVC architecture.

#### 4.2.8. User Interface Design

The screenshot shows the 'Company' page of the Tatari System. At the top, there's a header with the title 'Company' and a breadcrumb navigation 'Home / Company'. Below the header is a section titled 'Add New Company' with a blue 'Add New' button. Underneath is a table titled 'List All Companies' with columns: Action, Company, Email, City, Country, and Added By. A single company entry is listed: 'Educational Materials Production and Distribution Enterprise' (Type: Corporation, Contact#: +251114533232, Website: empde.com.et), added by 'System Admin' from 'Addis Ababa, Ethiopia'. The table includes sorting icons for each column. At the bottom, it says 'Showing 1 to 1 of 1 entries' and has navigation buttons for 'Previous', '1', and 'Next'.

Figure 4.35 Tatari System Company Page User Interface

The screenshot shows the 'Employees' page of the Tatari System. At the top, there's a header with the title 'Employees' and a breadcrumb navigation 'Home / Employees'. Below the header is a section titled 'Add New Employee' with a blue 'Add New' button. Underneath is a table titled 'List All Employees' with columns: Action, Name, Company, Contact, and Role. A single employee entry is listed: 'System Admin' (Name: Tatarisystem, Company: Educational Materials Production and Distribution Enterprise, Location: Addis Ababa, Department: IT Department, Designation: Software Developer), with contact information (Email: tataris\_admin@tatari.com, Phone: 251912123423) and role 'Super Admin Active'. The table includes sorting icons for each column. At the bottom, it says 'Showing 1 of 1 entries' and has navigation buttons for 'Previous', '1', and 'Next'.

Figure 4.36 Tatari System Employees Page User Interface

## **5. Chapter Five: Implementation**

### **5.1. Overview**

An implementation is the computer programming and deployment of a technical specification as a program, software component, or other computer systems. In Object-oriented programming, when a class implements an interface, the class is an implementation of the interface which consists of the methods specified by the specification.

In this section of the documentation, though it would not present a complete implementation section due to limited time and scheduling; it will discuss the general path and approaches that are going to be followed to develop the proposed system into a product based on the analysis, specification, design, and model that have been described in earlier chapters. The complete implementation document, as well as system testing, will be included in the following documentation that is planned to be commenced in the second semester of the academic year as per the scheduling.

### **5.2. Coding Standards**

Coding conventions are a collection of standards for programming style, procedures, and methods for each component of a program written in a particular programming language. File structure, indentation, comments, declarations, statements, white space, naming conventions, programming practices, programming principles, programming rules of thumb, architectural best practices, and so on are generally covered by these conventions. There are standards for the structural consistency of applications. Following these guidelines is strongly recommended for software developers who want to improve the readability of their source code and make software maintenance easier [\[14\]](#).

The implementation phase of this project while developing the system will follow some general coding standards [\[15\]](#) as well as several PHP style guides recommended by CodeIgniter [\[16\]](#). The following section presents a review of some of the standards and styles that will be used.

#### **a) Standard Header description for Modules**

For clear description and maintenance of the code, we will begin the header of different modules such as the models, views, controller files with a standard header detail. It will consist of information like:

- Name of the module
- Author of the module
- Date of module creation
- Update history
- Summary of module objective
- Functions and variables supported or accessed

### b) Indentations

Appropriate indentation is essential to enhance code readability and reusability. Proper use of white spaces as well as layout is important to preserve the quality of the code. We will attempt to maintain:

- Proper indentation and spacing on a nested block of codes.
- No whitespace should come before the opening PHP tag or follow the closing PHP tag. Output is protected, so whitespace in files can cause output to begin before CodeIgniter outputs its content, leading to errors and an inability for CodeIgniter to send proper headers.
- Exercise Allman-style indenting.
- The beginning and end of the code will have proper indentation with block and parenthesis.
- Spacing after a comma between function arguments.
- Braces starting in a new line of braces following code will be exercised as required.

### c) Naming standards

We will exercise proper naming standards for local variables, global variables, constants, and functions throughout the code as best possible. This will improve code readability, maintainability as well as understandability across the development team or other party. The naming standards will also follow the case-sensitive naming for global and local variable declarations of CodeIgniter.

- Variables and functions will use underscores (\_) to name multiple separate words usually with lower case letters.
- Class files must be named beginning with an upper case Somelibrary or Some\_library, while any other file name (configurations, views) should be in all lowercase like config.php
- Functions will often be verbs that describe the task performed by that particular function.
- Variables will often be nouns that describe the data being held or communicated.
- Avoid the use of digits in any naming of entities as it might cause confusion and runoffs.

### d) Length of functions

A single source code file might consist of thousands of lines with dozens of functions inside. The file length is not of much importance but the length of functions should be managed carefully, as a long function might be hard to understand. Lengthy functions should be divided into smaller chunks that would achieve the same goal. Keeping functions as short and as simple as possible will be exercised, however long the source code in a file would be.

### e) Commenting

Code should be well documented inside the source code as well as outside. A well-documented code would be simpler to maintain and also update. Commenting is one of the principal methods used to document source code. Code should have comments wherever necessary. It not only assists programmers in describing the flow and purpose of the code, but it can also be extremely useful when returning to our code months later or hunting bugs. DocBlocks standard of phpDocumentor comment standards will be exercised in all general comment requiring areas of the source code.

### **5.3. Prototype**

A preliminary prototype is going to be developed for the proposed system for this semester as per the scheduling and full system implementation and testing will be done in the second semester of the academic year. The initial prototype will consist of user interface pages of several systems features with CRUD (Create, Read, Update and Delete) control functionality.

The prototype will mostly have a backend system where users log in and access system services from a secure enterprise portal. The system will have a standard administrative user interface that is the AdminLTE dashboard and template. System services are going to be laid out in sidebars, navigations, and settings. Features are going to be listed out in the sidebar with all necessary options. Forms are going to be used to receive user input. Datatables will be used to list out data items stored in the database.

MySQL server will be used as a database server, it will contain the relational database along with the tables, data, and relations of the system. The prototype will follow a model-view-controller design in a hierarchical manner which is HMVC to accommodate the modular, scalable, and access-control architecture. CodeIgniter will be used to develop the API that delivers data along with controllers. Libraries and helpers will be used as a third party for pdf generation, email, and loaders.

### **5.4. Implementation details**

The system development is mainly focused on two implementation groups which are client-side and server-side.

#### **5.4.1. Client Side**

This is the part of the system that runs on the user device when it's accessed from the web server. As system users might be using various devices from desktops, laptops to smart phones and tablets. Thus, the system would be developed in such a way that it can accommodate those range of devices by optimizing it as much as possible. This would be achieved by systematically reducing the overall size of the web pages as well as data rendered on those pages. CodeIgniter is ideal for this task as it offers simple routing methods, and cache management; minimizing loading time.

A recent web browser is installed and running on the client-side. It will be used to access the system from the enterprise portal or the localhost at the time of development. The device can be a desktop computer or a laptop for a better user experience with moderate performance specifications.

Client-side consists of frontends particularly for administrators and employees. It can be characterized as interactive ground with dashboards, navigations, sidebars, forms and Datatables. AdminLTE would be used as a boiler plate admin template along with Datatables to handle the various features and functionalities discussed above.

### 5.4.2. Server Side

Server-side can be considered where the system resides and where the clients are getting served based on their requests. Here also the framework CodeIgniter plays an important role along with HMVC architecture to provide a reliable service.

At the time of deployment, the system with all its contents, database, and data will be served on an enterprise server along with a database server. It will provide the system services through a defined portal and then to a web page of the web browser. The server would perform some generic operations related to serving the system such as:

- User authentication from both client and server-side.
- Database querying operations
- Write data and files onto system and database
- Process user input from forms and queries
- Store the system data and logs
- Store files and data from the user in an organized manner

Details about the implementation of the prototype will be verbosely discussed in subsequent documentation. However, as an introductory piece, the following table will be regarding the details of the implementation phase of the project and the components that are planned to be used in the development of the prototype.

Component Name	Implementation Detail
<b>HMVC Router</b>	An open-source library extending from CI_Router (CodeIgniter's native router), runs the route mapping function for the HMVC model. It also validates path segments, sets default controllers, parses routes
<b>HMVC Loader</b>	An open-source library extending from CI_Loader (CodeIgniter's native loader), loads controllers and adds the current module to all paths. It is also used to load classes, controllers, models, configs, widgets, languages, and helpers.
<b>Base Model</b>	A base model with a series of CRUD functions (CI's query builder), validation-in-model support, event callbacks.
<b>Authentication Controller</b>	An authentication controller extending from the base controller and CI security, to handle the authentication process of users such as login, logout, and reset password.
<b>Employee Controller</b>	An extension of the base controller which manages the employee model of the system.
<b>Employee Model</b>	An extension of CI base model to get employee records as well as write, update and delete employee records and related details.

<b>Roles Controller</b>	Extending from the Role model, it controls the entire role and permission scheme hierarchically and dynamically.
<b>Roles model</b>	Extending from the CI base model, it manipulates the role data inside the database.
<b>Autoloader</b>	Loads packages, models, and libraries automatically for fast access and delivery. To keep the framework as lightweight as possible only the absolute minimal resources are loaded by default.
<b>Authentication View</b>	It consists of the login page and forgot passwords page as a user interface, it is driven by the authentication controller to authenticate the user.
<b>Cache controller</b>	A cache management driver from CodeIgniter modified for an HMVC design pattern.
<b>Datatables</b>	A library that enhances the accessibility of data in HTML tables; it uses AJAX to do the ordering, filtering, searching, and paging data.
<b>AdminLTE</b>	An open-source WebApp template for admin dashboards and control panels. It is a responsive HTML template that is based on the CSS framework Bootstrap 3.
<b>TCPDF</b>	TCPDF is a free and open-source software PHP class for generating PDF documents. TCPDF is the library used to generate payslips and invoices in our case.
<b>PHPMailer</b>	PHPMailer is a code library to send emails safely and easily via PHP code from a web server, It is used to send emails of different templates like forgot password in our case.

Table 5.1 Implementation details

## 5.5. Deployment

The system is planned to be deployed on-premise of the model organization EMPDE. On-premise deployment is the installation and release of a system locally on-site of the enterprise using the company's servers and computers.

On-premise systems are much easier to maintain and modify. Many organizations, particularly in niche industries like specialized manufacturers with unique processes, value the opportunity to customize to their particular needs and requirements. Provided that, the project considered to be a scalable and flexible one which aligns with this principle.

Apache has been selected as it supports a variety of features required for the optimization of the system. It implements as compiled modules extending the core functionality which aligns with HMVC architecture. It also provides a range of components from authentication schemes for server-side programming languages such as PHP which is the language used by CodeIgniter. It also apparatuses TLS/SSL, proxy modules and URL rewriting modules as well as custom filter and log files. Compression (gzip), caching and load balancing are key considerations which were fulfilled by it.

## 6. Chapter Six: System Testing

System testing is the process of testing how the systems' functional and non-functional requirements are performing their functions correctly and effectively. The system testing in this case tests the Tatari System which focuses primarily on the backend of the system.

### 6.1. Objective

The primary aim of this section of the document is to plan and execute particular test cases and methods in the testing phase of the system development. The main objectives are:

- List out the recommended test cases and requirements.
- Differentiate which features and functionalities to test and which not to test.
- Specify test strategies and methods to implement during the testing phase.
- List out the various deliverable elements of the testing activity.
- Identify the required resources and provide an estimate of the testing activity.

### 6.2. Scope

This test documentation is intended for the following system and version.

System Name	Tatari System
System Description	Tatari System is an enterprise resource management system that consists of human resources management and finance/accounting modules which are modeled after the organization Ethiopian Educational Materials Production and Distribution Enterprise (EMPDE); that can be repurposed for any similar enterprise for generic use.
System Version	1.3.0

Table 6.1 Test System Description

The testing plan focuses on performing tests on the developed modules and features. Activities commenced for this phase include:

- Test the functionality of the features.
- Test the operability of the two integrated modules.
- Confirm correct data creation at the database.

### 6.3. Resources

Various resources will be used to perform the testing activity of this system. The main aspects are role (human roles) and system (hardware and software) resources.

#### 6.3.1. Roles

The following table presents the specific roles played by individuals involved in the testing phase. It defines the specific roles and responsibilities of individuals as well as the tasks performed to achieve the completion of the testing stage.

<b>Role</b>	<b>Recommended Minimum Human Resources</b>	<b>Responsibility</b>
<b>Test Supervisor</b>	Arefat Hyeredin	<p>Provide oversight and monitoring of the entire testing activity</p> <p>Responsibilities:</p> <ul style="list-style-type: none"> <li>▪ Prepare required resources</li> <li>▪ Provide technical direction and assistance</li> <li>▪ Reporting of the testing activity</li> </ul>
<b>Testcase Designer</b>	Arefat Hyeredin Gizealew Endeshaw	<p>Specify, prioritize and design test cases</p> <p>Responsibilities:</p> <ul style="list-style-type: none"> <li>▪ Generate Testcases</li> <li>▪ Generate Test suite</li> </ul>
<b>System Testers</b>	Biniyam Gossaye Girum Getachew	<p>Perform the testing according to the test cases</p> <p>Responsibilities:</p> <ul style="list-style-type: none"> <li>▪ Execute manual tests</li> <li>▪ Log test results</li> <li>▪ Recover from errors</li> <li>▪ Document issues and errors found</li> </ul>
<b>System Administrator</b>	Arefat Hyeredin	<p>Provide proper access to the system modules as per assignment and explain system in general.</p> <p>Responsibilities:</p> <ul style="list-style-type: none"> <li>▪ Host and maintain system in the test period</li> <li>▪ Log system calls and errors during the test period</li> <li>▪ Monitor and direct system paths and directories to testers</li> </ul>
<b>Database Administrator</b>	Gizealew Endeshaw	<p>Ensure database is accessible with proper privileges.</p> <p>Responsibilities:</p> <ul style="list-style-type: none"> <li>▪ Monitor test data entry, update, and removal</li> <li>▪ Administer database</li> </ul>
<b>Test Analyst</b>	Abreham Bekele Girum Getachew	<p>Focus on business problems and solutions of the system.</p> <p>Responsibilities:</p> <ul style="list-style-type: none"> <li>▪ Ensure functional readiness of the system</li> <li>▪ Run and troubleshoot tests to find defects</li> </ul>
<b>QA Tester</b>	Biniyam Gossaye	Ensure the software product works seamlessly and meets quality standards.

Table 6.2 Human Roles Testing Resources

### 6.3.2. System

The table below shows the system resources involved in the testing phase of Tatari System.

Resources	Description
Desktops	Microsoft Windows 7,8,10 Operating System   Corei5 Processor   4GB RAM   x64-based PC
Laptops	Microsoft Windows 10 Operating System   Corei5 Processor   8GB RAM   x64-based PC
Browsers	Google Chrome, Mozilla Firefox, Brave Browser, Microsoft Edge
Web Server	Apache HTTP Server, Ngrok (exposes local servers behind NATs and firewalls to the public internet over secure tunnels.)
Database	MySQL running on MariaDB's XAMPP server

Table 6.3 System Resources for Testing

### 6.4. Schedule

The testing schedule for Tatari System is as follows. It has been hastened due to the limited time, however, fundamental testing procedures are planned and performed accordingly. Person days (pd) unit is used to estimate the test activities required for every task in the testing procedure.

Task	Effort (pd)	Scheduled Dates	Responsibility
Test planning	1	August 21, 2021	Arefat Hyeredin
Test Design	1	August 22, 2021	Gizealew Endeshaw
Unit Testing	2	August 22-23,2021	Girum Getachew, Abreham Bekele, Biniyam Gossaye
Integration testing	1	August 23,2021	Arefat Hyeredin
Reporting	1	August 24, 2021	All Team Members

Table 6.4 Test Schedule

### 6.5. Test Case Scenarios and Requirements

The section below discusses those components (use cases, functional requirements, non-functional requirements, features) that have been identified as targets for the testing phase. The following subsections present what is tested and what is not tested.

#### 6.5.1. Data and Database Integrity Testing

- Verify access to Tatari System Database.
- Verify access to Tatari Database tables and records.
- Verify access with proper privileges is granted to authorized users.
- Verify accurate retrieval of records from database at insertion, deletion and update states.
- Verify persistence of data for auto-generated fields and auto-loaded forms.
- Validate each value is successfully recorded in the database tables.
- Verify the data in the data tables can be edited, deleted and filtered.

### **6.5.2. Functional Testing**

This testing type tests the mainline functions of the system, it involves basic usability testing and accessibility testing.

- Verify model organization registration
  - Insert, update and delete organizational details.
  - Insert, view, update and delete organizational location, department, subdepartment and designation
- Verify test user – employee registration
  - Insert, view, update and delete employee details such as basic information, emergency contacts, qualification, work experience, bank account, and documents
- Verify Core HR processes applied on employee records
  - Insert, view, update and delete core HR actions such as transfers, promotions, resignations, warnings, complaints and terminations
- Verify Attendance submission and tracking of employees
  - Receive and process clock-in and clock-out operations of employees
  - Create, view, update and delete work schedule, leaves and overtime requests
  - View, update and delete daily and monthly attendance timesheet records
- Verify Payroll creation and Payslip generation from underlying salary data of employees
  - Insert, view, update and delete salary record for an employee
  - View and Filter payroll records for both paid and unpaid salary
  - Generate payslip for processed salary payment either individually or in bulk
- Verify vacancy creation and job applications
  - Create, view, update and delete job posts.
  - View, manage and delete job applications
- Verify the accurate generation of HR reports
  - Generate HR reports such as employee report, attendance report, leave report and payslip reports from underlying employee information
- Verify test account – ledger accounts creation
  - Insert, view, update and delete ledger accounts with their details
  - View total balance of accounts and each account balance
- Verify payer/payee record creation
  - Insert, view, update and delete payer record
  - Insert, view, update and delete payee record
- Verify financial transactions are created and recorded
  - Create, view, update and delete deposit records
  - Create, view, update and delete expense records
  - Create, view, update and delete transfer records
  - View all transactions that occur on ledger accounts
- Verify creation and payment processing of invoices
  - Create, view, update, delete and export invoices/bills
  - View payment history for paid invoices

- Verify the accurate generation of Fiscal reports
  - Generate Fiscal reports such as account statements, expense reports, income reports and transfer reports from underlying financial transaction information

#### **6.5.3. User Interface Testing**

- Verify ease of use, navigation, and consistency across a sample set of user interface views
- Verify screen views conform to a general dashboard, icon and other GUI standards
- Verify notifications, menu items, buttons, and color standards are met

#### **6.5.4. Performance Testing**

- Verify response time for login authentication
- Verify response time for data insertion of various components
- Verify response time for search and filter operations on data tables
- Verify response time for file upload in forms
- Verify response time for report (exported) file generation

#### **6.5.5. Load Testing**

Non-functional testing such as load testing and automated tests are not executed. Unfortunately, tests such as acceptable concurrent user volume, processor memory consumption, server request processing time are not performed due to limited time.

#### **6.5.6. Security and Access Testing**

- Verify login operation from local PC
- Verify login operation from remote PC
- Verify login, locked state and password recovery operations
- Verify access control method is applied based on the role-based access mechanism

### **6.6. Test Strategy**

Test Strategy presents the recommended approach to the testing of the system. The previous section on Test Requirements described what will be tested; this explains how it will be tested. The main deliberations for the test strategy are the techniques to be used and the condition for knowing when the testing is completed. In addition to the considerations provided for each test below, testing should only be performed using recognized, controlled databases in secured settings.

#### **6.6.1. Data and Database Integrity Testing**

The database and operations on the database should be tested as isolated systems. These systems should be tested without the applications (as the interface to the data).

<b>Test Objective</b>	Ensure Database access methods function properly and without data corruption.
<b>Technique</b>	<ul style="list-style-type: none"> <li>▪ Invoke each database access method seeding each with valid and invalid data (or requests for data).</li> <li>▪ Inspect the database to ensure the data has been populated as intended, all database events occurred properly, or review the returned data to ensure that the correct data was retrieved (for the correct reasons)</li> </ul>
<b>Completion Criteria</b>	All database access methods function as designed and without any data corruption.

*Table 6.5 Data and Database Integrity Testing*

### **6.6.2. Function Testing**

Testing of the system should focus on any target requirements that can be traced directly to use cases, functional requirements, and business rules. The goals of these tests are to verify proper data acceptance, processing, and retrieval, and the proper implementation of the business rules. This type of testing is based upon black-box techniques, that is, verifying the application (and its internal processes) by interacting with the application via the GUI and analyzing the output (results). Identified below is an outline of the testing recommended for each application:

<b>Test Objective</b>	Ensure proper application navigation, data entry, processing, and retrieval.
<b>Technique</b>	<p>Execute each use case, use case flow, or function, using valid and invalid data, to verify the following:</p> <ul style="list-style-type: none"> <li>▪ The expected results occur when valid data is used.</li> <li>▪ The appropriate error/warning notifications are displayed when invalid data is used.</li> </ul>
<b>Completion Criteria</b>	<ul style="list-style-type: none"> <li>▪ All planned tests have been performed.</li> <li>▪ All identified defects have been addressed</li> </ul>

*Table 6.6 Function Testing*

### **6.6.3. User Interface Testing**

User Interface testing verifies a user's interaction with the system. The goal of UI Testing is to ensure that the User Interface provides the user with the appropriate access and navigation through the functions of the applications. In addition, UI Testing ensures that the objects within the UI function as expected and conform to corporate or industry standards.

<b>Test Objective</b>	Verify the following: <ul style="list-style-type: none"> <li>▪ Navigation through the application properly reflects business functions and requirements, including a screen to screen, field to field, and use of access methods.</li> </ul>
-----------------------	--

	<ul style="list-style-type: none"> <li>▪ Screen objects and characteristics, such as menus, size, position, state, and focus conform to standards.</li> </ul>
<b>Technique</b>	Create/modify tests for each screen view to verify proper navigation and object states for each application window and object.
<b>Completion Criteria</b>	Each screen was successfully verified to remain consistent with the benchmark version or within the acceptable standard.

Table 6.7 User Interface Testing

#### 6.6.4. Performance and Load Testing

Performance testing measures response times, transaction rates, and other time-sensitive requirements. The goal of Performance testing is to verify and validate the performance requirements have been achieved. Performance testing is usually executed several times, each using a different "background load" on the system. Load testing measures subjects the system-under-test to varying workloads to evaluate the system's ability to continue to function properly under these different workloads. The goal of load testing is to determine and ensure that the system functions properly beyond the expected maximum workload. Regrettably, the team has not been able to run these tests due to limited time and a tight schedule. Rest assured, these tests would be performed shortly to complete the testing phase.

#### 6.6.5. Security and Access Testing

This is where the implemented Role-based access control, as well as Security Testing, is performed. It focuses on two vital areas of security:

- Function / Data security, including access to the Data or Business Functions, and
- System Security, including logging into / accessing the system. Function security ensures that, based upon the desired security, users are restricted to specific functions or are limited in the data that is available to them. System security ensures that only those users granted access to the system are capable of accessing the system and only through the appropriate privileges.

<b>Test Objective</b>	<ul style="list-style-type: none"> <li>▪ Function / Data Security: Verify that users can access only those functions/data for which their user type is provided permissions.</li> <li>▪ System Security: Verify that only those users with access to the system are permitted or authorized to access them.</li> </ul>
<b>Technique</b>	Create/modify tests for each screen view to verify proper navigation and object states for each application window and object.
<b>Completion Criteria</b>	Each screen was successfully verified to remain consistent with the benchmark version or within the acceptable standard.

Table 6.8 Security and Access Control Testing

## **6.7. Estimated Risk**

The primary risks that may hinder the testing and verification process are as follows:

- Limited time for preparation and completion of system development, testing, and documentation
- Not delivering product on scheduled deliverable deadlines
- Limited technical skills to perform required testing
- Inability to commence test by testers such as team member incapacitated
- Inadequate budget to conduct testing
- Unavailability of testing resources, and
- Inaccessibility to automated testing platforms to simulate multiple user activities.

## **6.8. Contingency Plan**

The contingency plan to remedy the aforementioned risks in the above section are:

- Conduct testing with least and available budget as possible
- Compile progress and showcase to stakeholders even if the testing is not fully completed
- Perform load and stress testing as well as automated tests at the earliest convenience
- Allocate more hours to developing, testing, and documenting outside of schedule to meet deadlines
- Consider using third-party libraries, packages, and frameworks to comply with standards and deadlines
- Quickly troubleshoot errors and document every defect for further analysis

## 7. Chapter Seven: User Manual

### 7.1. Demonstration of Tatari System

#### 7.1.1. Login Page

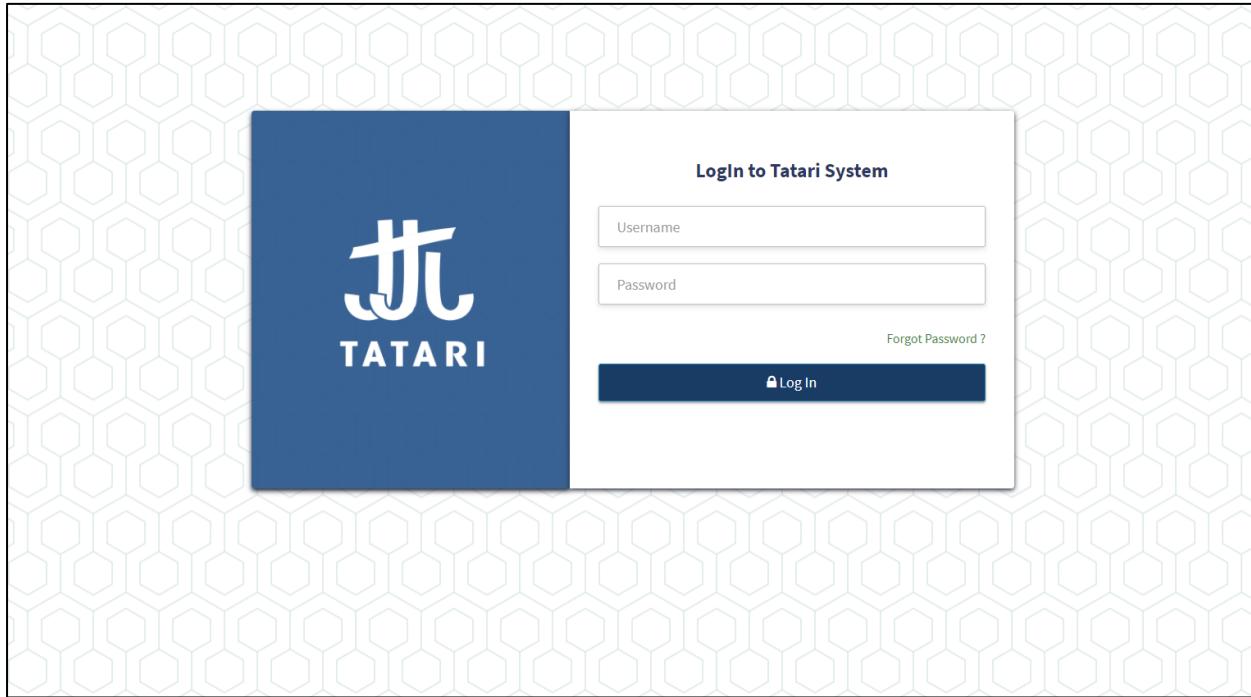


Figure 7.1 Login Page

This is the login page for Tatari System – an enterprise resource management system with human resources and finance management modules with various features. This login page would be used to authenticate system administrator and system users (employees) with their username and password into the system based on a pre-configured access control and privilege mechanism. It also consists of a forgot password and password recovery mechanism.

#### 7.1.2. Administrator Dashboard

System administrator has a dedicated dashboard showing key trends and standings of the organization such as total employees, quick links, employee-designation chart, employee-department chart and financial data. It also has a sidebar on the left side that has all the built-in features and modules categorized for easy access. Sidebar consists of the logged in user information as well as all the navigation tabs to the employee, attendance, payroll, transaction etc. screen views. Furthermore, the header has a section on the right corner that has language option and account options to lock user, logout, see profile page and change password.

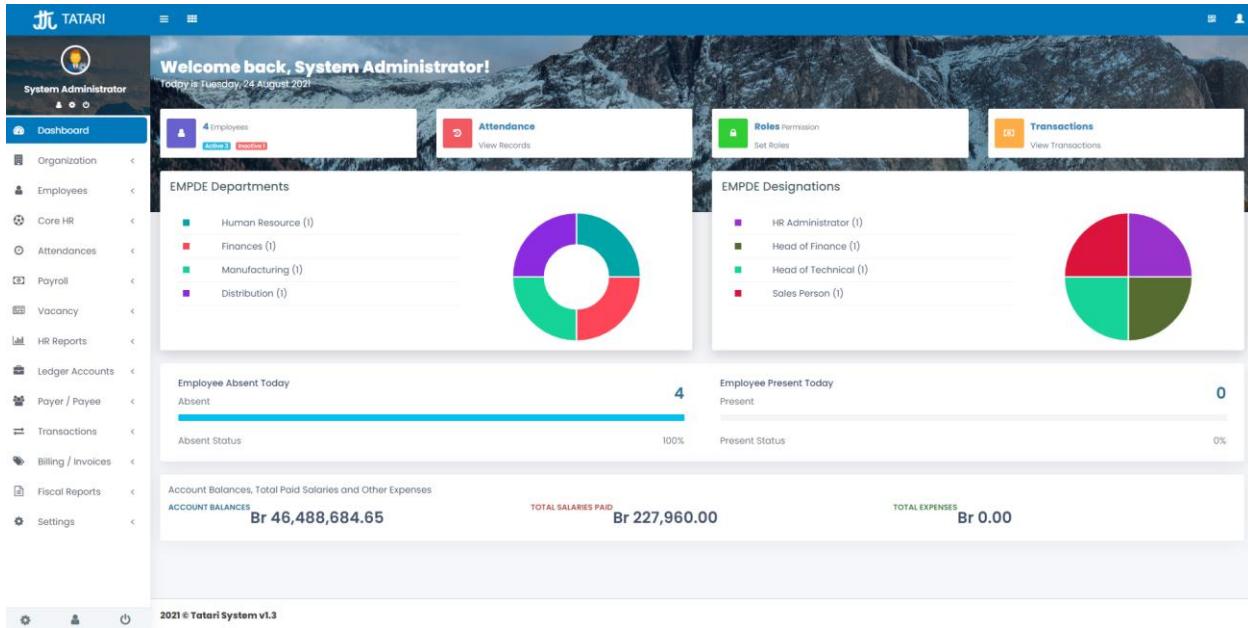


Figure 7.2 Departments Page

### 7.1.3. Company Page

The organization tab on the left sidebar contains options leading to Company, Location, Department, Sub department and Designation pages.

The Company page contains a retractable form to add a company with different fields necessary to register a certain company. It also shows a datatable with all the details of the registered company with an action to view, update and delete a company equipped with a modal.

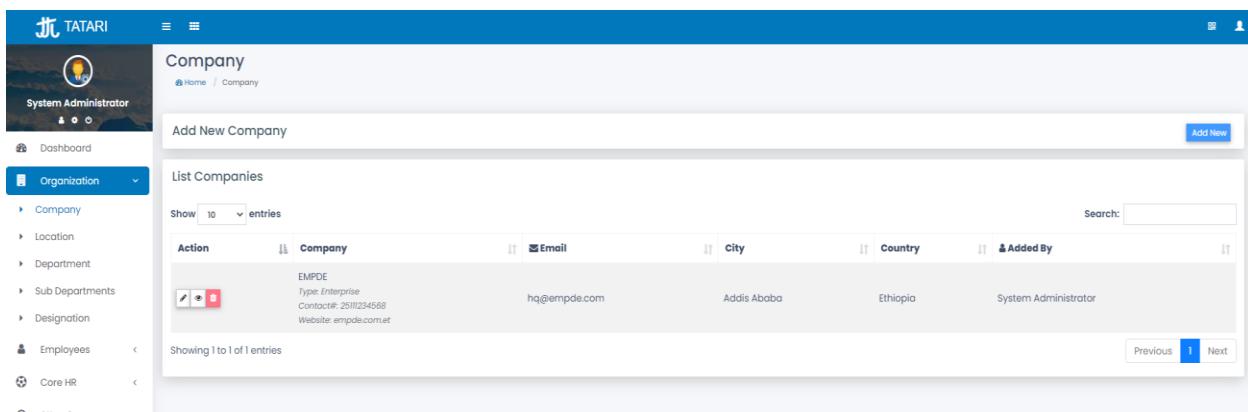


Figure 7.3 Company Page

### 7.1.4. Location Page

The Location page contains a retractable form to add the location of a registered company with different fields required to register a certain company location. It also shows a datatable with all the details of the inserted company locations with an action to view, update and delete locations.

The screenshot shows the 'Locations' page in the TATARI system. On the left, there is a sidebar with a 'System Administrator' profile picture and a navigation menu under 'Organization' containing links for Company, Location, Department, Sub Departments, Designation, Employees, Core HR, Attendances, Payroll, Vacancy, HR Reports, Ledger Accounts, Payer / Payee, and Transactions. The main content area has a header 'Locations' with a 'Home / Locations' breadcrumb. Below the header is a section titled 'Add New Location' with a 'Add New' button. The main area is titled 'List All Locations' and includes a search bar and a table with columns: Action, Location Name, Location Head, City, Country, and Added By. The table lists four entries: Addis Ababa Head Quarters (Head: Abebe Balacha, City: Addis Ababa, Country: Ethiopia, Added By: System Administrator), Bahir Dar (Head: Abebe Balacha, City: Bahir Dar, Country: Ethiopia, Added By: System Administrator), Hawassa Branch (Head: Semira Umer, City: Hawassa, Country: Ethiopia, Added By: System Administrator), and Mekelle (Head: Semira Umer, City: Mekelle, Country: Ethiopia, Added By: System Administrator). A message at the bottom says 'Showing 1 to 4 of 4 entries'. Navigation buttons for 'Previous' and 'Next' are at the bottom right.

Figure 7.4 Locations Pages

### 7.1.5. Department, Subdepartment, and Designation Pages

The Department page contains a form on the left side and a data table on the right. The form on the left has fields to enter details about a particular department. On the right side of the screen, it has a data table that lists recorded departments and an option to update and delete a particular department record.

The screenshot shows the 'Departments' page in the TATARI system. The left sidebar is identical to the one in Figure 7.4. The main content area has a header 'Departments' with a 'Home / Departments' breadcrumb. Below the header is a section titled 'Add New Department' with fields for Name, Company, Location, and Department Head, and a 'Save' button. To the right is a section titled 'List All Departments' with a search bar and a table with columns: Action, Department Name, Location, and Company. The table lists four entries: Human Resource (Head: Semira Umer, Location: Addis Ababa Head Quarters, Company: EMPDE), Finances (Head: Abebe Balacha, Location: Addis Ababa Head Quarters, Company: EMPDE), Manufacturing (Head: Abebe Balacha, Location: Addis Ababa Head Quarters, Company: EMPDE), and Distribution (Head: Abebe Balacha, Location: Addis Ababa Head Quarters, Company: EMPDE). A message at the bottom says 'Showing 1 to 4 of 4 entries'. Navigation buttons for 'Previous' and 'Next' are at the bottom right.

Figure 7.5 Departments Page

Similarly, the Subdepartment and Designation pages have the same layout and structure with their own properties. The data tables have a feature to sort, filter, and search records; which is pretty handy for a large number of records.

### 7.1.6. Employees Page

The employee's tab on the left sidebar contains options leading to Employee, Employee Directory, Role & Privileges, and Employee Last Login pages.

The Employees page has a stat section at the top with active/inactive status and employee gender statistics. Then it contains a retractable form with fields required to register an employee. After the above two sections, it has a data table listing registered employees with their basic details and an option to delete and view more details about a particular employee.

The screenshot shows the 'Employees' page in the Tatari System. At the top, there is a stats bar with four colored boxes: green (3 Active), red (1 Inactive), blue (75% Male), and orange (25% Female). Below this is a retractable form titled 'Add New Employee' with fields for Name, Company, Contact, and Role. The main area is titled 'List Employees' and displays a table of four employees:

Action	Name	Company	Contact	Role
	System Administrator ID: tataris Shift: Basic Schedule	EMPDE Location: Addis Ababa Head Quarters Department: Finances Designation: Head of Finance	tataris administrator@tataris.com 12345678900	Super Admin Active
	Abebe Bolachio ID: obobe Shift: Basic Schedule	EMPDE Location: Addis Ababa Head Quarters Department: Human Resource Designation: HR Administrator	obobe obefat2005@gmail.com 1232	Employee Active
	Semira Umer ID: 123244 Shift: Basic Schedule	EMPDE Location: Addis Ababa Head Quarters Department: Manufacturing Designation: Head of Technical	semira semira@gmail.com 251534535	Basic Account Inactive
	Dereje Yalow ID: dereje Shift: Basic Schedule	EMPDE Location: Addis Ababa Head Quarters Department: Distribution Designation: Sales Person	dereje dereje@yahoo.com 092123445	Employee Active

At the bottom, there is a footer with the text '2021 © Tatari System v1.3'.

Figure 7.6 Employees Page

### 7.1.7. Employee Details Page

The Employee Details page shows a horizontal tab with options such as General, Core HR, Set Salary, Leaves, Payslips, and Profile Picture. The vertical tab list on General Employee Details has options to manage different details about a particular employee such as Basic Information, Emergency Contacts, Qualifications, Work Experience, Bank Account, Document, and Change Password; each with a separate form and data table that loads on selection without page reloads.

**Employee Details**

Home / Employee Details

General Core HR Set Salary Leaves Payslips Profile Picture

**Basic Information**

First Name \* Abebe Last Name \* Balacha

Employee ID \* abebe Username \* abebe Email \* arefat2005@gmail.com

Company \* EMPDE Location \* Addis Ababa Head Quarters Department \* Human Resource

Designation \* HR Administrator Date of Joining \* 2021-05-02 Date of Leaving

Role \* Employee Gender Male Marital Status Single

Contact Number \* 1232 Status Active Office Shift Basic Schedule

Date of Birth \* 1998-03-28 Leave Category Annual Leave Sick Leave

State State City City P.O.Box

Address Piazza Taytu Hotel around Dz

Figure 7.7 Employee Details Page

**Employee Details**

Home / Employee Details

General Core HR Set Salary Leaves Payslips Profile Picture

**Basic Information**

**List All Qualification**

Action	School/University	Time Period	Education Level
<input checked="" type="checkbox"/> Addis Ababa University	Aug-18-2021 - Aug-23-2021	University Degree	

Showing 1 to 1 of 1 entries

**Add New Qualification**

School/University \* School/University Education Level High School Diploma

Time Period \* from to Professional Certification Professional Certification

Description Description

**Save**

Figure 7.8 Employee Details - Qualifications Page

Core HR tab on horizontal tab list shows details of a particular employee in relation to transfer, promotion, complaint and warning with a list of records on a datatable.

Set Salary tab page presents vertical tab list of different salary types and requirements such as Allowance, Commission, Loan, Deductions, Overtime and other payments. These can be shown separately in their own views with their own form to set their details and datatable listing the related records.

Leaves page in employee details shows leaves applied by the employee. Payslip page in employee details shows a datatable list of processed salary payments from payroll. Profile picture tab page presents an option to choose a new profile or remove the current one.

### 7.1.8. Employees Directory Page

Employees Directory page displays a filter option at the top to filter out the entire employees' directory list in a card view with their pictures. It has an option to view more details of an employee which takes the user to the employee details page.

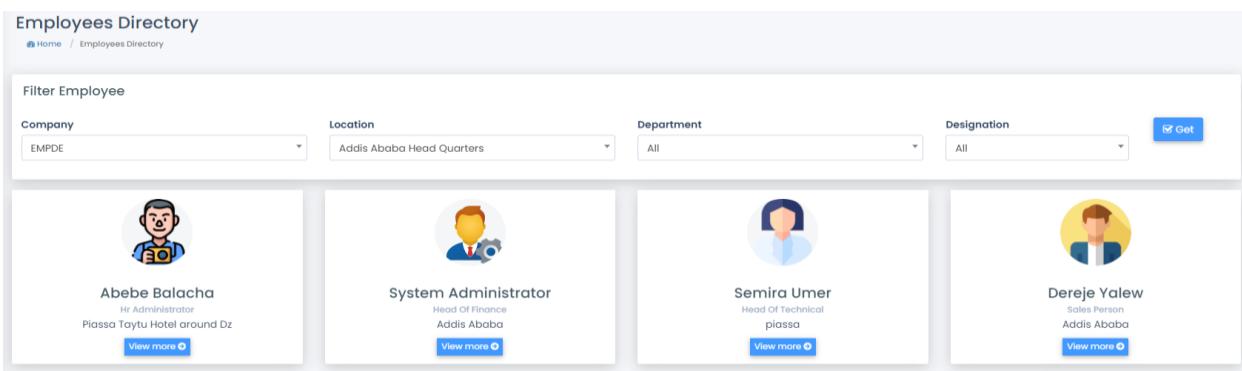
The screenshot shows a web-based application titled "Employees Directory". At the top, there is a navigation bar with "Home" and "Employees Directory". Below the navigation is a "Filter Employee" section with dropdown menus for "Company" (set to "EMPDE"), "Location" (set to "Addis Ababa Head Quarters"), "Department" (set to "All"), and "Designation" (set to "All"). A blue "Get" button is to the right of the filters. Below the filter section is a grid of four employee profiles, each in its own card. 1. Abebe Balacha: HR Administrator, Piazza Taytu Hotel around Dz. 2. System Administrator: Head Of Finance, Addis Ababa. 3. Semira Umer: Head Of Technical, piazza. 4. Dereje Yalew: Sales Person, Addis Ababa. Each card includes a "View more" button.

Figure 7.9 Employees Directory Page

### 7.1.9. Employee Last Login Page

Employee Last Login Page presents the system administrator with a datatable list of employees logs of last login date-time information and other status, that can be searched and sorted accordingly.

### 7.1.10. Roles and Privileges Page

This is one of the most vital features that manages and applies the role-based access control system to a certain created role. It has a retractable form to define role name and access type with an expanding feature list on the right containing checkboxes to select specific and general features to assign as a privilege to a role. Selecting a general feature will grant all subsequent features as a privilege to that role. If you select the module for an employee, an employee will only have access to their related record of that particular feature. If you select a company view, an employee will have access to all record of his/her organization of that particular feature. At the bottom, it shows a datatable listing available roles with an option to update and delete a certain role.

**Add New Role**

**Role Name \***  
Role Name

**Select Access \***  
Select Access

**Features**

- +  Organization
- +  Employees
- +  Core HR
- +  Attendances
- +  Payroll
- +  Vacancy
- +  HR Reports
- +  Ledger Accounts
- +  Payees and payers
- +  Transactions
- +  Billing / Invoices
- +  Fiscal Reports
- +  Settings
- Theme Settings

**Remark:**

- 1- If you select a company view, an employee will have access to all record of his/her organization of that particular feature.
- 2- If you select the module for an employee, an employee will only have access to their related record of that particular feature.

**List All Roles**

Action	Role ID	Role Name	Menu Permission	Added Date
	1	Super Admin	All Menu Access	May-28-2021
	2	Employee	Custom Menu Access	May-21-2021
	3	Basic Account	Custom Menu Access	Aug-21-2021

Showing 1 to 3 of 3 entries

Figure 7.10 Roles & Privileges Page

### 7.1.11. Core HR Pages

There are 6 different Core HR pages i.e., Transfer Page, Resignation Page, Promotions Page, Complaints Page, Warnings Page and Termination Page. Each of these pages have a similar layout in view and they perform their specific core HR-related operation on employee records. These pages have a retractable form with fields required to be filled out to process one of the particular operations. They each also have a datatable listing their records accordingly with possible actions on those records such as update, view and delete as well as search and sort operations.

**Complaints**

**Add New Complaint**

**List All Complaints**

Action	Complaint From	Complaint Against	Company	Complaint Title	Complaint Date
	Abebe Balacha Description: Noise around machines <span>Pending</span>	1. Abebe Balacha	EMPDE	Noise	May-18-2021
	Semira Umer Description: dfg <span>Pending</span>	1. Semira Umer	EMPDE	dfg	Aug-25-2021
	Semira Umer Description: dfg <span>Pending</span>	1. Abebe Balacha	EMPDE	dfg	Aug-10-2021
	Semira Umer Description: sdf <span>Accepted</span>	1. Abebe Balacha	EMPDE	sdf	Aug-25-2021
	Semira Umer Description: fdf <span>Pending</span>	1. Abebe Balacha	EMPDE	fdf	Aug-17-2021

Showing 1 to 5 of 5 entries

Figure 7.11 Complaints Page

The screenshot shows the 'Transfers' page. At the top, there's a header with a logo and navigation links. Below it, a form titled 'Add New Transfer' contains fields for 'Company' (dropdown), 'Employee to Transfer' (dropdown), 'Transfer Date' (text input), 'Description' (text area), 'Transfer To (Department)' (dropdown), and 'Transfer To (Location)' (dropdown). A 'Save' button is at the bottom right. Below the form is a section titled 'List All Transfers' with a table. The table has columns: Action, Employee Name, Company, Transfer Date, and Status. One entry is shown: Abebe Balacha, EMPDE, May-06-2021, Pending. There are buttons for CSV, Excel, PDF, and Print, along with a search bar.

Figure 7.12 Transfers Page

### 7.1.12. Attendance Pages

4 pages are particularly dedicated to attendance management, they are Daily Attendance Page, Monthly Timesheet Page, Date-wise Attendance Page, and Update Attendance Page. These pages show different views of the attendance records as per their assigned names.

Daily Attendance Page shows a daily attendance report data table with employees' status, total work, overtime and clock in/clock out timestamps.

The monthly Timesheet Page shows a data table listing days of the month horizontally with an employee's attendance status such as absent, leave, and present.

The screenshot shows the 'Attendance' page. At the top, there are filters for 'Location' (dropdown) and 'Date' (text input with value '2021-08-24'). A 'Get' button is next to the date input. Below is a section titled 'Daily Attendance Report' with a table. The table has columns: HR Information (Employee, Company, Date, Status, Clock IN, Clock Out, Late, Early Leaving, Overtime, Total Work, Total Rest) and Attendance Report (Employee, Company, Date, Status, Clock IN, Clock Out, Late, Early Leaving, Overtime, Total Work, Total Rest). Two entries are listed: Abebe Balacha (EMPDE, Aug-24-2021, Absent) and Dereje Yalew (EMPDE, Aug-24-2021, Absent). There are buttons for Previous and Next, along with a search bar.

Figure 7.13 Attendance Page

Date-wise Attendance page is a more sophisticated daily attendance where dates can be filtered from start to end date parameters and employees to view such specified attendance status.

Update Attendance page shows a form on the left to specify which attendance record to update and the datatable on the right displays that particular record to update. It also has an extended feature to add new attendance if necessitated by the administrator or other body.

### 7.1.13. Work Shift Page

Work shift page has a retractable form at the top with fields to create a new schedule by defining in and out times for the days of the week. It has a datatable with a list of shifts created previously which can be edited and deleted accordingly.

Figure 7.14 Add New Work Shift Page

### 7.1.14. Overtime Request Page

Overtime request page has a datatable showing a list of overtime requests from employees. It also has an option to add a new overtime request on behalf of employees with a modal. It also has an option to approve, update and delete these overtime request records.

Figure 7.15 Overtime Request Page

### 7.1.15. Leaves Page

The screenshot shows the 'Leave' page with a header 'Leave' and a breadcrumb 'Home / Leave'. Below is a section titled 'Add Leave' with a 'Add New' button. Under 'List All Leave', there's a search bar and a table with columns: Action, Leave Type, Department, Employee, Request Duration, and Applied On. The table contains three entries:

Action	Leave Type	Department	Employee	Request Duration	Applied On
	Sick Leave Reason: Medical Reasons - COVID-19 Positive 	Human Resource	Abebe Balacha	Jul-29-2021 to Jul-30-2021 Total Days: 2	Jul-29-2021
	Annual Leave Reason: rest 	Human Resource	Abebe Balacha	Aug-19-2021 to Aug-21-2021 Total Days: 3	Aug-20-2021
	Annual Leave Reason: leave 	Manufacturing	Semira Umer	Aug-24-2021 to Aug-25-2021 Total Days: 2	Aug-20-2021

At the bottom, it says 'Showing 1 to 3 of 3 entries' and has navigation buttons for 'Previous' and 'Next'.

Figure 7.16 Leave Page

Like most pages of the Tatari system, the Leaves page has a retractable form at the top that can be used to process a leave request of an employee. And it has a datatable listing out leave records of employees. It also has an option to edit and view records as well as view more details on a particular leave, which is called Leave Details page.

### 7.1.16. Payroll Page

Payroll page shows a datatable with a list of processed payroll payments with a filter mechanism that can specify department and month for the payment. It also has an option to view further details on the payment and a payslip download option. Payslips page has details about the payment to the employee and approval mechanism, with a payslip PDF generation option.

The screenshot shows the 'Employee Payslip' page with a header 'Employee Payslip' and a breadcrumb 'Home / Employee Payslip'. The main area displays a payslip for 'Abebe Balacha' for January 2021. It includes sections for 'Payment Details' (Monthly Payslip), 'Employee ID: #abebe', 'Employee Name: Abebe Balacha', 'Payslip NO: 0', 'Phone: 1232', 'Joining Date: May-02-2021', 'Department: Human Resource', 'Designation: HR Administrator', 'Status: Final Approval', and a 'Save' button.

On the right, there are two tables: 'Earnings' and 'Deductions'.

Earnings	
Monthly Payslip:	Br 10,000.00
Total Allowance:	Br 1,233.00
Commissions:	Br 0.00
Total Loan:	Br 0.00
Total Overtime:	Br 16.00
Statutory deductions:	Br 0.00
Other Payment:	Br 0.00
Paid Amount:	Br 11,233.00

Figure 7.17 Employee Pay slip Page

### 7.1.17. Generate Payslip Page

This page has two filter forms at the top to load the datatable with particular employee and month data or process bulk payment option. The datatable below based on the above-set parameters renders records of employee's payment details with options to generate payslip.

The screenshot shows the 'Generate Payslip' page. At the top, there are two filter sections: 'Generate Payslip' and 'Bulk Payment'. The 'Generate Payslip' section includes dropdowns for 'Company' (All Companies), 'Employee' (All Employees), and 'Select Month' (2021-08), with a 'Search' button. The 'Bulk Payment' section includes dropdowns for 'Company' (All), 'Location' (All), 'Department' (All), and 'Select Month' (2021-08), with a checked 'Bulk Payment' checkbox. Below these is a table titled '2021-08' showing employee payment details:

Action	Name	Payroll Type	\$ Salary	\$ Net Salary	Status
[Edit]	Dereje Yalew (EMPDE) ID: dereje	Monthly Payslip	Br 0.00	Br 0.00	UnPaid
[Edit]	Abube Balacha (EMPDE) ID: abebe	Monthly Payslip	Br 10,000.00	Br 10,496.00	UnPaid
[Edit]	Semira Umer (EMPDE) ID: 1231244	Monthly Payslip	Br 12,300.00	Br 12,300.00	UnPaid

At the bottom, it says 'Showing 1 to 3 of 3 entries' and has 'Previous' and 'Next' buttons.

Figure 7.18 Generate Pay slip Page

### 7.1.18. Vacancy Pages

The Vacancy Feature has two primary pages which are Job Posts and Job Candidates. It is used to announce a vacancy position for the company and also view received job applications.

Job Posts page has a retractable form that adds a new job post with its necessary details. It also has a datatable that lists out job posts that have been created with an option to update and delete those records.

The screenshot shows the 'Add New Job' page under the 'Vacancy' section. On the left is a sidebar with navigation links like Dashboard, Organization, Employees, Core HR, Attendances, Payroll, and Vacancy (which is selected). The main area has a form for 'Add New Job' with fields for Job Title, Department, Publish to FrontEnd?, Status, Gender, Number of Positions, Date of Closing, Minimum Experience, and Short Description. There is also a 'Long Description' rich text editor. A 'Save' button is at the bottom right. Below the form is a table titled 'List All Jobs' showing one record:

Action	Position	Department	Posted Date	Status	Closing Date
[Edit]	IT Specialist Applicants	Human Resource	Aug-10-2021	Published	Aug-25-2021

At the bottom, it says 'Showing 1 to 1 of 1 entries' and has 'Previous' and 'Next' buttons.

Figure 7.19 Add New Job Page

Job Candidates page has datatable to view, approve, delete and download job applications from the candidates/applicants.

### 7.1.19. HR Report Pages

This particular feature has 4 major pages that each generate a different kind of HR-related reports from the underlying employee data. These are Employees Report, Attendance Report, Payslip Reports and Leave Reports. These pages however different their functionality is, they all have the same layout. They each have report filter form to specify parameters on the right and an exportable data table on the right listing the generated records of any selected report type and format.

The screenshot shows the 'Employees Report' page. On the left, there is a 'Report Filters' sidebar with dropdowns for 'Company' (EMPDE), 'Department' (All), and 'Designation' (All), and a 'Get' button. The main area is titled 'View Employees Report' and contains a table with columns: ID, Name, Email, Department, Designation, and Status. The table has four entries:

ID	Name	Email	Department	Designation	Status
I231244	Semira Umer	semira@gmail.com	Manufacturing	Head of Technical	Inactive
abebe	Abebe Balachha	orefat2005@gmail.com	Human Resource	HR Administrator	Active
dereje	Dereje Yalew	dereje@yahoo.com	Distribution	Sales Person	Active
tatari	System Administrator	administrator@tatari.com	Finances	Head of Finance	Active

At the bottom, it says 'Showing 1 to 4 of 4 entries' and has 'Previous' and 'Next' buttons.

Figure 7.20 Employees Report Page

### 7.1.20. Ledger Accounts Page

Ledger Accounts page is the first page from the Finance module, it has a form on the left to create a new ledger account with its details. It also has a datatable on the right that lists all available ledger accounts with an option to update and delete them. One page that is closely related to this, but that is a separate page is the Account Balances page displaying balances of each account and total balance in a datatable view.

The screenshot shows the 'Ledger Accounts' page. On the left, there is a 'Add New Account' form with fields for 'Account Name' (Account Name), 'Initial Balance' (Initial Balance), 'Account Number' (Account Number), 'Bank Name' (Bank Name), and 'Bank Branch' (Bank Branch). At the bottom is a 'Save' button. The main area is titled 'List All Ledger Accounts' and contains a table with columns: Action, Ledger Accounts, Account No., Bank Name, Balance, and Bank Branch. The table has three entries:

Action	Ledger Accounts	Account No.	Bank Name	Balance	Bank Branch
	EMPDE Government Total Budget	100032343322	CBE	Br 32,042,656.00	Megenogna
	EMPDE Government Approved Budget	100032345456	CBE	Br 12,404,028.65	Pissa
	EMPDE Operational	10003223223	DAS	Br 2,042,000.00	Bole Main

At the bottom, it says 'Showing 1 to 3 of 3 entries' and has 'Previous' and 'Next' buttons.

Figure 7.21 Ledger Accounts Page

### 7.1.21. Payer/Payees Pages

Payer and Payees pages are similar and simple pages with a small form to record the detail of a payee or a payer which would later be used to process deposits and expenses. They also show registered payees and payers on a datatable on the right with an option to update, search, delete and filter those records.

The screenshot shows the 'Payers' page. On the left, there is a form titled 'Add New Payer' with fields for 'Payer Name' and 'Contact Number', and a 'Save' button. On the right, there is a table titled 'List All Payers' showing three entries:

Action	Payer	Contact Number	Create At
<a href="#">Edit</a>	Bahir Dar University	251232234234	Aug-19-2021
<a href="#">Edit</a>	Selale University	+25134232323	Aug-19-2021
<a href="#">Edit</a>	Addis Ababa Institute of Technology	+251923223233	Aug-19-2021

Showing 1 to 3 of 3 entries

Figure 7.22 Payers Page

### 7.1.22. Transaction Pages

There are 3 different transaction pages i.e., Deposit Page, Expense Page, and Transfer Page. Each of these pages has a similar layout in view and they perform their specific financial-related operation on ledger accounts. These pages have a retractable form with fields required to be filled out to process one of the, particularly mentioned operations. They each also have a datatable listing their records accordingly with possible actions on those records such as update, view and delete as well as search and sort operations.

The screenshot shows the 'Add New Deposit' page. On the left, there is a form with fields for 'Account', 'Amount' (2021-08-24), 'Category', 'Payer', 'Payment Method', and 'Ref#'. On the right, there is a table titled 'List All Deposit' showing two entries:

Action	Account	Payer	Amount	Category	Ref#	Payment	Date
<a href="#">Edit</a>	EMPDE Operational	Selale University	Br 32,000.00	Direct Deposit	I232214	Bank Transfer	Aug-10-2021
<a href="#">Edit</a>	EMPDE Government Total Budget	--	Br 30,446.20	--	Invoice Payments	Cheque	Aug-21-2021

Showing 1 to 2 of 2 entries

Figure 7.23 Add New Deposit Page

View Transactions page is a page that displays a datatable of a list of all kinds of transactions whether it is deposit, expense or transfer in a sortable, searchable and structured manner.

The screenshot shows a table titled "List All Transactions" with the following data:

Date	Ledger Accounts	Amount	Ref#
Aug-02-2021	EMPDE Government Approved Budget	Br 12,344.00	rwerqw45345
Aug-02-2021	EMPDE Government Total Budget	Br 12,344.00	rwerqw45345
Aug-10-2021	EMPDE Government Total Budget	Br 55,000.00	asd
Aug-10-2021	EMPDE Operational	Br 55,000.00	asd
Aug-10-2021	EMPDE Government Total Budget	Br 15,000.00	asd1213
Aug-10-2021	EMPDE Operational	Br 15,000.00	asd1213
Aug-10-2021	EMPDE Operational	Br 32,000.00	1232214
Aug-16-2021	EMPDE Operational	Br 20,000.00	234234dsfdf
Aug-16-2021	EMPDE Government Approved Budget	Br 12,000.00	12a1212lbcc
Aug-17-2021	EMPDE Government Approved Budget	Br 15,000.00	CBE1288912VV

Figure 7.24 View Transactions Page

### 7.1.23. Invoices Page

Invoices page displays a general stat of invoice records at the top, with a list of invoice records in a datatable. It has an option to create an invoice which leads to a separate form that can be used to create an invoice with all the necessary details for billing information.

The screenshot shows a summary of invoices and a table of all invoices:

Paid Invoices	Unpaid Invoices	Paid Amount	Due Amount
1	0	Br 30,446.25	Br 0.00

**List All Invoices**

Action	Invoice #	Issued to	Total	Invoice Date	Due Date	Status
	INV-0001	DS Importers	Br 30,446.25	Aug-11-2021	Aug-24-2021	

Figure 7.25 Invoices Page

View Invoices page is a page that shows the generated invoice with an option to pay the bill as well as export it as a PDF. Once an invoice is paid, it will be shown in a separate page which is called Invoice Payments displaying a list of invoice payment records.

The screenshot shows the 'Create Invoice' page. At the top, there are fields for 'Invoice Number' (INV-0004), 'Issued to', 'Invoice Date', and 'Due Date'. Below this, there are two rows for items. Each row has fields for 'Item Name', 'Tax Type' (VAT (15%)), 'Tax Rate' (0), 'Qty/Hr' (1), 'Unit Price' (0), and 'Subtotal' (0). A blue '+ Add Item' button is visible. To the right, there are summary fields for 'Sub Total' (Br 0), 'TAX' (Br 0), and a discount section with 'Discount Type' (Flat), 'Discount' (0), and 'Discount Amount' (0). At the bottom, there is a 'Grand Total' (Br 0) and a 'Submit Invoice' button.

Figure 7.26 Create Invoice Page

### 7.1.24. Fiscal Report Pages

This particular feature has 4 major pages that each generate a different kind of fiscal-related reports from the underlying financial data. These are Account Statement, Expense Report, Income Reports and Transfer Reports. These pages however different their functionality is, they all have the same layout. They each have report filter form to specify parameters at the top and an exportable datatable at the bottom listing the generated records of any selected report type and format.

The screenshot shows the 'Account Statement' page. At the top, there is a dropdown for 'EMPDE Government Total Budget', date range fields for '2021-08-17' to '2021-08-26', and a 'Get' button. Below this, there is another 'Account Statement' section with a 'Show 10 entries' dropdown and buttons for 'Copy', 'CSV', 'Excel', 'PDF', and 'Print'. There is also a 'Search:' input field. The main area contains a table with columns 'Date', 'Description', 'Credit', and 'Debit'. The table data is as follows:

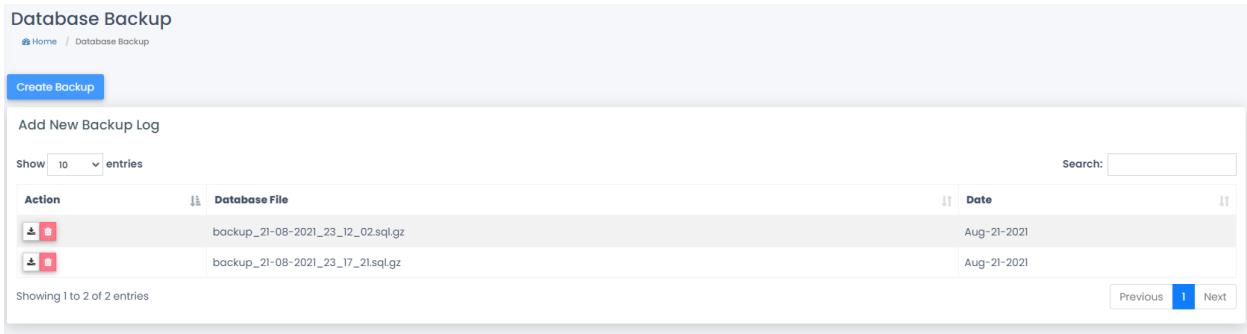
Date	Description	Credit	Debit
Aug-17-2021	transfer Management Decision for Approval	Br 15,000.00	Br 0.00
Aug-21-2021	income Invoice Payments	Br 30,446.20	Br 0.00
		<b>Credit: Br 45,446.20</b>	<b>Debit: Br 0.00</b>

At the bottom, it says 'Showing 1 to 2 of 2 entries' and has 'Previous' and 'Next' buttons.

Figure 7.27 Account Statement Page

### 7.1.25. Settings Page

However, not so much configuration and setting pages have not been developed. Database Backup page can be seen as one that can create a backup of the database from the current log. Other feature that can be considered as settings is the profile page. On the top right corner, with logout and lock account, there is a profile option which leads to account settings page of an authenticated user. Languages Setting, although at early stages of translation is included in a limited manner to showcase the feature.

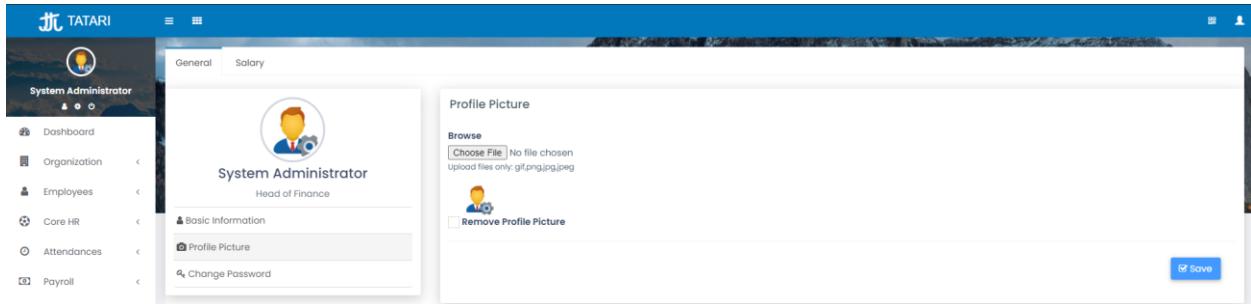


The screenshot shows a 'Database Backup' page with a header 'Home / Database Backup'. A blue 'Create Backup' button is at the top left. Below it is a table titled 'Add New Backup Log' with columns 'Action', 'Database File', and 'Date'. Two entries are listed:

Action	Database File	Date
	backup_21-08-2021_23_12_02.sql.gz	Aug-21-2021
	backup_21-08-2021_23_17_21.sql.gz	Aug-21-2021

At the bottom, it says 'Showing 1 to 2 of 2 entries' with 'Previous' and 'Next' buttons.

Figure 7.28 Database Backup Page



The screenshot shows a 'Profile Picture' change page for a 'System Administrator' user. The sidebar on the left shows 'General' and 'Salary' tabs. The main area displays the user's profile picture and basic information. It includes fields for 'Choose File' (no file chosen), 'Upload files only: gif,png,jpg,jpeg', and 'Remove Profile Picture'. A blue 'Save' button is at the bottom right.

Figure 7.29 Profile Picture Change Page

### 7.1.26. Employees Dashboard Page

When an employee logs in with a registered account detail, he/she will land on an employee dashboard displaying a clock in/clock out card that marks attendance hours for that particular logged-in employee. It also shows personal details card of the authenticated employee on the side. An employee can choose to update its profile to some extent like change basic information, change password or profile picture.

The sidebar on the left for an authenticated employee would not show all available features of the system, rather display the assigned features based on the role and privilege granted by the system administrator which is governed by the business rule and role-based access control.

Another thing to consider here is that for example when an employee has access to overtime requests in an individual level, that employee can only send an overtime request for himself/herself, one cannot process that request for other employees unless specifically granted by the administrator; for like HR officer.

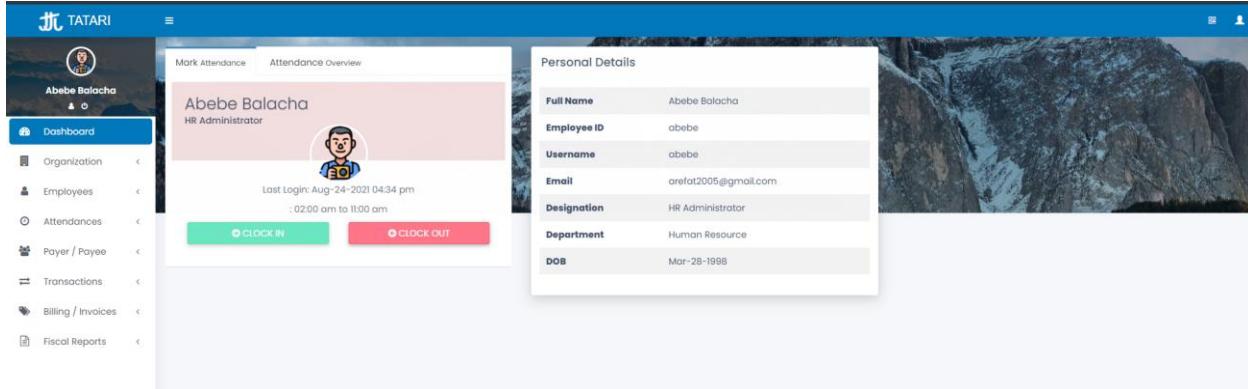


Figure 7.30 Employee Dashboard Page

## 7.2. Git Structure and Version Control

The system development is accompanied by git version control and GitHub repository system. Git tracks changes to a file which is very convenient when developing with a team, it also saves branches and different release versions. It is an ideal method for parallel system development work, which has been utilized for this project. The current GitHub [repository](#) is at the GitHub organization. The repository is a part of an [organization](#) created on GitHub that has documents, systems and prototype repositories. [GitHub Link: <https://github.com/TatariSystem/tatari-v1>] The primary system repository has several branches that can be categorized as:

- Master Branch
- Features Branch
- Staging Branch

**Master Branch:** is the main repository branch where all approved changes are merged and it is the current stable state of the system. It is the primary repository for the development which holds a working state and tested codebase for the system.

**Features Branch:** are pieces of works by different team members pushing and pulling requests to be merged to the master branch. They are a few with all of them designated with their corresponding features and notes. They are later pulled to the master branch after testing and approval.

**Staging Branch:** is a branch that is used for transition and middle-state testing; to be later merged to master branch. It is mostly used to fix errors and compare notes with the team.

This screenshot shows a GitHub repository page for the project 'TatariSystem/overtime'. The repository has 11 branches and 1 tag. The 'Code' tab is selected, displaying a list of files and their commit history. The README file is also visible. On the right side, there are sections for About, Releases, Packages, and Contributors.

**About**  
This is preliminary project starter module for Tatari Enterprise Resource Management System modules development.

**Releases**  
1 tags

**Packages**  
No packages published  
Publish your first package

**Contributors** 5

**Languages**  
PHP 93.2% JavaScript 4.8% Other 2.0%

Figure 7.31 Git Repository

Here are some of the branches that are active and being worked on, as discussed in the above section. There are different feature branches and module releases included in the repository.

This screenshot shows the GitHub branches page. The 'Active' tab is selected, displaying a list of active branches. Each branch entry includes the branch name, last update time, commit count, pull request status, and merge history.

Branch	Last updated	Commits	Pull Requests	Status
report	Updated 7 hours ago by unknown	0   1	#9	Open
overtime	Updated 3 days ago by Girum-Getachew	1   0	#7	Merged
amharic-launguge-module	Updated 4 days ago by giz86	2   1	#8	Open
job_candidates-job_post	Updated 6 days ago by BiniyamB	4   0	#6	Merged
attendance-hr-module	Updated 8 days ago by arefathi	7   0	#5	Merged
resignation-termination-feature	Updated 28 days ago by Girum-Getachew	9   0	#4	Merged
promotion-transfer	Updated last month by unknown	11   0	#3	Merged
complaint-warning	Updated last month by giz86	13   0	#2	Merged
designation-location-features	Updated last month by BiniyamB	15   0	#1	Merged

Figure 7.32 Git Branches

## **8. Chapter 8: Conclusion and Recommendation**

### **8.1. Conclusion**

The proposed system is an enterprise resource management system that primarily consists of human resources and financial management system. It is modeled after the test organization Ethiopian Educational Materials Production and Distribution Enterprise. Its main goal is to provide a means to facilitate daily business operations of a company and transition them into a computerized system. It simplifies the business process that have been identified in the requirement elicitation phase. It eases the entire institutional interaction between employees, departments and tasks. It also creates a better data infrastructure for the enterprise. Furthermore, the proposed system plans to account for the existing business practices to ease the transition to a computerized system.

On this project we attempted to specify the statement of the problem which is the need for a computerized resource management system for the company. After detail investigation into the company and other related system, we have defined the business operations that can be designed into the system, more so incase of human resources; there was limited time and access to model the finance department, as we have modelled it using standard finance operations and ideals. This led to us defining the general objective of the project which is to design and develop a web-based resources management system for the Ethiopian Educational Materials Production and Distribution Enterprise (EMPDE). This will be done by implementing the functional requirements like company and employee registration, core HR processes, vacancy posts, payroll processing, ledger account management, financial transactions, billing and invoice, and reporting; while maintaining security, usability, reliability and business rules.

Some of the difficulties that hindered the project is the limited access to departmental internal workings to model processes, restricted movement due to the COVID-19 pandemic, and less system development, documentation and testing time.

### **8.2. Recommendation**

During the development of the system and the entire project timeline, the project team and advisor has faced and tackled various challenges. Credit to the past experiences and the one here developed while designing, modelling and implementing such a similar project, we would suggest to focus on the requirement gathering because it was the most fundamental stance that based the entire project. We would like to invite any team or individual who would like to work more on this system to add more features, include more modules that would bring it closer to becoming a full-fledged ERP system. We would also like to state our open interest for the model organization or any other similar company to deploy and test the system in a real work environment, which would provide more insight and feedback for further work.

## References

1. Oracle Australia *What is ERP? Oracle*. Retrieved March 16, 2021, from <https://www.oracle.com/au/erp/what-is-erp/>
2. ebizframe Eastern Software Systems. *ERP in Ethiopia*. Retrieved April 21, 2021, from <https://ebizframe.com/erp-in-ethiopia/>
3. *Ethiopian Educational Materials Production and Distribution Enterprise Establishment Proclamation*. Retrieved March 18, 2021 from <https://chilot.me/wp-content/uploads/2011/11/reg-no-50-1999-educational-materials-production-and-distri1.pdf>
4. *EMPDE*. Education Material Production and Distribution Enterprise. Retrieved March 24, 2021, from <http://www.empde.com.et/index.htm>
5. Workwise Software. *Five Risks Businesses face by not using ERP software*. Retrieved March 25, 2021, from <https://www.workwisellc.com/blog/five-risks-businesses-face-not-using-erp-software/>
6. Terillium. *Benefits of ERP: Advantages, Disadvantages & Selecting an Enterprise Resource Planning System*. Retrieved March 29, 2021, from <https://terillium.com/benefits-of-erp/>
7. Vilmate Blog. *A Complete Guide to ERP Application Development*. Retrieved March 30, 2021, from <https://vilmate.com/blog/custom-erp-software-development/>
8. Capterra. *Best ERP Software Reviews*. Retrieved April 4, 2021, from <https://www.capterra.com/enterprise-resource-planning-software/#top-20>
9. BetterBuys. *Top ERP Solutions (for Manufacturing)*, Retrieved April 5, 2021, from <https://www.betterbuys.com/erp/top-erps-for-manufacturing/>,  
<https://www.betterbuys.com/erp/>
10. CodeIgniter Documentation. *CodeIgniter Security*. Retrieved April 26, 2021, from <https://www.codeigniter.com/userguide3/general/security.html>
11. Hostry. *Apache-reliable and flexible web server*. Retrieved April 26, 2021, from <https://hostry.com/blog/apache-reliable-http-server/>
12. Barry Cogan. *HMVC: An Introduction and Application*. Retrieved April 26, 2021, from <https://code.tutsplus.com/tutorials/hmvc-an-introduction-and-application--net-11850>
13. Wikipedia, the Free Encyclopedia. *Role-based Access Control*. Retrieved April 28, 2021, from [https://en.wikipedia.org/wiki/Role-based\\_access\\_control](https://en.wikipedia.org/wiki/Role-based_access_control)
14. Wikipedia, the Free Encyclopedia. *Coding Conventions*. Retrieved April 29, 2021, from [https://en.wikipedia.org/wiki/Coding\\_conventions](https://en.wikipedia.org/wiki/Coding_conventions)
15. Sayan Kumar Pal, Geeks for Geeks. *Coding Standards and Guidelines*. Retrieved April 29, 2021, from <https://www.geeksforgeeks.org/coding-standards-and-guidelines/>
16. CodeIgniter User Guide. *PHP Style Guide*. Retrieved April 29, 2021, from <https://codeigniter.com/userguide3/general/styleguide.html>

## Bibliography

- Abiot Sinamo Boltena, Jorge Marx Gomez (2012). *A Successful ERP Implementation in an Ethiopian Company: A case study of ERP Implementation in Mesfine Industrial Engineering Pvt. Ltd.* Conference on ENTERprise Information Systems, University of Oldenburg, Oldenburg, Germany.
- Alemu Samuel, Mesfin Belachew, Mesfin Kifle (2015). *ERP Implementation Framework: The Case of Ethiopia.* Addis Ababa University, Addis Ababa, Ethiopia.
- AlQashami A, Heba M (2015). *Critical success factors (CSFs) of enterprise resource planning (ERP) system implementation in Higher Education Institutions (HEIs): concepts and literature review.* Computer science & information technology, fourth international conference on advanced information technologies and applications (ICAITA 2015). Dubai, UAE.
- D.M. Bahssas, Dr. Adnan M. Albar, (2015). *Enterprise Resource Planning (ERP) Systems: Design, Trends and Deployment.* The International Technology Management Review, Vol. 5, King Abdulaziz University, Jeddah, Saudi Arabia.
- Kumar, V. (2004). *Enterprise resource planning Concepts and Practices, Second Edition.* Prentice Hall of India private limited, New Delhi-110 001.
- Liaquat Hossain and Mohammad A. Rashid (2002). *The Evolution of ERP Systems: A Historical Perspective.*
- Poston, R., & Grabski, S. (2001). *Financial Impacts of Enterprise Resource Planning Implementations.* International Journal of Accounting Information Systems.
- Soffer P., Boaz Golany, and Dov Dori (2003). *ERP Modeling: A Comprehensive Approach.* Information Systems.
- Tadele Kebede Jember (2014). *An Assessment of the Practices and Challenges of Business Process Reengineering (BPR) Implementation: The Case of Educational Materials Production and Distribution Enterprise (EMPDE).* St. Mary's University. School of Graduate Studies. Addis Ababa, Ethiopia.
- Tobie A. M., R. A. Etoundi and Jean Zoa (2017). *A Literature Review on ERP implementation in African Countries.* University of Yaoundé I, Cameroon.
- Wagaw, Mekonnen (2017). *Acceptance of homegrown enterprise resource planning (ERP) systems in Ethiopia.* Appl Inform 4, 6 <https://doi.org/10.1186/s40535-017-0034-5>
- Zhelyazkov GI (2012). *The impact of ICT systems on road transport SMEs in Australia.*

## **Appendix 1**

**ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY**  
**COMPUTER SCIENCE AND ENGINEERING DEPARTMENT**  
**SENIOR PROJECT TITLE – RESOURCE MANAGEMENT SYSTEM FOR EMPDE**

***Open ended Questions used for the Survey and Interview***

**Questions**

1. What is the main purpose, function and responsibility of the IT Department at EMPDE?
2. Describe some of the systems that are in use currently at EMPDE? (It can be paper-based, spreadsheet or computerized systems)
3. What are the main departments that are found in EMPDE? (like Human Resource, Accounting...)
4. Where in the organization do you think a computerized web resource management system can be the most productive?
5. The team is preparing to develop an enterprise resource management system module (like Human Resource Management System, Finance/Accounting System), system that can be implemented for testing at the enterprise. How significant and operational do you think these systems can be at your organization?
6. How close do you think the employees of the organization are to the ICT world? Can you explain the case for the usability of such system if implemented?
7. What kind of forms do you use to register employees, keep attendance, issue invoices and bills, keep inventory of the production and distribution? Can you share/attach a sample format for those paper forms and table used in these daily activities? (Photos If possible)
8. Why do you think there is no computerized resource management system deployed at the enterprise until now?
9. We understand that the organization uses spreadsheets and papers to enter and keep record of different data. Can you specify the column headers on those spreadsheets? (this would be helpful in designing the database and digital forms)
10. Finally, given the description of our project; can you suggest or explain any method, material or direction to help us develop such a system?

*(For documentation purposes only)*

Name - \_\_\_\_\_ Position- \_\_\_\_\_ Date- \_\_\_\_\_

## Appendix 2 – Letter of Request for Cooperation



### አዳማ ስኞችና ቁኜርና የኝርጓት ADAMA SCIENCE & TECHNOLOGY UNIVERSITY

1888

022-110-0025

Fax: 0251-022-112-01-50

E-mail: socedean@astu.edu.et

ASTU, P.O. Box 1888, Adama - Ethiopia

አዳማ - ኢትዮጵያ  
Adama - Ethiopia

አንድዎች መረጃዎች የሚረዳ እና የማኅኑ ደረሰኑ

በ: 18 ዓመት 2013  
የፍር: 212-1-4/መ/80/2052/13

ለአካላት የሚፈጸም እና ከምደቻቸው  
ተ/ቤት አ.

P/C E&E ተዘግኝ

የጥቅ: የሰነድ ሲሆን የሚከታተሉት

በአዲት ለደገኘ ቁኜርና የኝርጓት በአዲት ማኅኑ ደረሰኑ እና ከምደቻቸው የሚፈጸም ሌሎች  
የሚፈጸም ተ/ቤት የሚከታተሉት የሚፈጸም የሚፈጸም የሚፈጸም የሚፈጸም የሚፈጸም የሚፈጸም

የሚፈጸም የሚፈጸም የሚፈጸም የሚፈጸም የሚፈጸም የሚፈጸም የሚፈጸም የሚፈጸም  
የሚፈጸም የሚፈጸም የሚፈጸም የሚፈጸም የሚፈጸም የሚፈጸም የሚፈጸም የሚፈጸም

1. ተግራ አረፍተኛ የዚሁ የሚፈጸም የሚፈጸም የሚፈጸም የሚፈጸም
2. ተግራ በኋላ የሚፈጸም የሚፈጸም
3. ተግራ የፌዴራል የሚፈጸም
4. ተግራ የኢትዮጵያ አንቀጽ የሚፈጸም
5. ተግራ አስተያየት የሚፈጸም

ከለም: 20  
R/C E&E ተዘግኝ  
የሚፈጸም የሚፈጸም የሚፈጸም የሚፈጸም  
የሚፈጸም የሚፈጸም የሚፈጸም



ገልጻ  
- ለት/ቤት ተዘግኝ  
- ለብንዱ ተ/ቤት  
አስተያየት

## Appendix 3 – Sample Source Codes

### A. Attendance Model

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Attendance_model extends CI_Model
{
    public function __construct()
    {
        parent::__construct();
        $this->load->database();
    }

    public function get_office_shifts() {
        return $this->db->get("tat_office_shift");
    }

    public function get_company_shifts($company_id) {
        $sql = 'SELECT * FROM tat_office_shift WHERE company_id = ?';
        $binds = array($company_id);
        $query = $this->db->query($sql, $binds);
        return $query;
    }

    public function all_leave_types() {
        $query = $this->db->get("tat_leave_type");
        return $query->result();
    }

    public function get_company_leaves($company_id) {
        $sql = 'SELECT * FROM tat_leave_applications WHERE company_id = ?';
        $binds = array($company_id);
        $query = $this->db->query($sql, $binds);
        return $query;
    }

    public function get_multi_company_leaves($company_ids) {
        $sql = 'SELECT * FROM tat_leave_applications where company_id IN ?';
        $binds = array($company_ids);
        $query = $this->db->query($sql, $binds);
        return $query;
    }

    public function attendance_time_check($employee_id) {
        $sql = 'SELECT * FROM tat_attendance_time WHERE employee_id = ?';
        $binds = array($employee_id);
        $query = $this->db->query($sql, $binds);

        if ($query->num_rows() > 0) {
            return $query->result();
        } else {
            return false;
        }
    }

    // Check if CheckIn is available
    public function attendance_first_in_check($employee_id,$attendance_date) {

        $sql = 'SELECT * FROM tat_attendance_time WHERE employee_id = ? and attendance_date = ? limit 1';
        $binds = array($employee_id,$attendance_date);
        $query = $this->db->query($sql, $binds);
    }
}
```

```

        return $query;
    }

    public function attendance_first_in($employee_id,$attendance_date) {
        $sql = 'SELECT * FROM tat_attendance_time WHERE employee_id = ? and attendance_date = ?';
        $binds = array($employee_id,$attendance_date);
        $query = $this->db->query($sql, $binds);

        return $query->result();
    }

    // Check if CheckOut is available
    public function attendance_first_out_check($employee_id,$attendance_date) {

        $sql = 'SELECT * FROM tat_attendance_time WHERE employee_id = ? and attendance_date = ? order by time_attendance_id desc limit 1';
        $binds = array($employee_id,$attendance_date);
        $query = $this->db->query($sql, $binds);

        return $query;
    }

    public function attendance_first_out($employee_id,$attendance_date) {

        $sql = 'SELECT * FROM tat_attendance_time WHERE employee_id = ? and attendance_date = ? order by time_attendance_id desc limit 1';
        $binds = array($employee_id,$attendance_date);
        $query = $this->db->query($sql, $binds);

        return $query->result();
    }
?>

```

## B. Attendance Controller

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Attendance extends MY_Controller {
    public function __construct() {
        parent::__construct();
        $this->load->model("Attendance_model");
        $this->load->model("Employees_model");
        $this->load->model("Tat_model");
        $this->load->model("Department_model");
        $this->load->model("Designation_model");
        $this->load->model("Roles_model");
        $this->load->model("Location_model");
    }

    public function output($Return=array()){
        header("Access-Control-Allow-Origin: *");
        header("Content-Type: application/json; charset=UTF-8");
        exit(json_encode($Return));
    }
}

```

```

public function index() {
    $session = $this->session->userdata('username');
    if(empty($session)){
        redirect('admin/');
    }
    $month_year = $this->input->post('month_year');
    if(isset($month_year)): $title = date('F Y', strtotime($month_year)); else: $title = date('F Y');
endif;
    $data['title'] = $this->lang->line('tat_employees_monthly_timesheet').' | '.$title;
    $data['breadcrumbs'] = $this->lang->line('tat_monthly_timesheet');
    $data['path_url'] = 'timesheet_monthly';
    $data['get_all_companies'] = $this->Tat_model->get_companies();
    $data['all_employees'] = $this->Tat_model->all_employees();
    $role_resources_ids = $this->Tat_model->user_role_resource();
    if(in_array('10',$role_resources_ids)) {
        if(!empty($session)){
            $data['subview'] = $this->load->view("admin/attendance/timesheet_monthly", $data, TRUE);
            $this->load->view('admin/layout/layout_main', $data);
        } else {
            redirect('admin/');
        }
    } else {
        redirect('admin/dashboard');
    }
}
public function attendance()
{
    $session = $this->session->userdata('username');
    if(empty($session)){
        redirect('admin/');
    }
    $data['title'] = $this->lang->line('dashboard_attendance').' | '.$this->Tat_model->site_title();
    $data['breadcrumbs'] = $this->lang->line('dashboard_attendance');
    $data['path_url'] = 'attendance';
    $data['all_office_shifts'] = $this->Location_model->all_office_locations();
    $role_resources_ids = $this->Tat_model->user_role_resource();
    if(in_array('28',$role_resources_ids)) {
        if(!empty($session)){
            $data['subview'] = $this->load->view("admin/attendance/attendance_list", $data, TRUE);
            $this->load->view('admin/layout/layout_main', $data);
        } else {
            redirect('admin/dashboard/');
        }
    } else {
        redirect('admin/dashboard');
    }
}

```

```

public function date_wise_attendance()
{
    $session = $this->session->userdata('username');
    if(empty($session)){
        redirect('admin/');
    }
    $data['title'] = $this->lang->line('left_date_wise_attendance').' | '.$this->Tat_model->site_title();
    $data['all_employees'] = $this->Tat_model->all_employees();
    $data['get_all_companies'] = $this->Tat_model->get_companies();
    $data['breadcrumbs'] = $this->lang->line('left_date_wise_attendance');
    $data['path_url'] = 'date_wise_attendance';
    $role_resources_ids = $this->Tat_model->user_role_resource();
    if(in_array('29',$role_resources_ids)) {
        if(!empty($session)){
            $data['subview'] = $this->load->view("admin/attendance/date_wise", $data, TRUE);
            $this->load->view('admin/layout/layout_main', $data);
        } else {
            redirect('admin/');
        }
    } else {
        redirect('admin/dashboard');
    }
}

public function update_attendance()
{
    $session = $this->session->userdata('username');
    if(empty($session)){
        redirect('admin/');
    }
    $data['title'] = $this->lang->line('left_update_attendance').' | '.$this->Tat_model->site_title();
    $data['breadcrumbs'] = $this->lang->line('left_update_attendance');
    $data['path_url'] = 'update_attendance';
    $data['get_all_companies'] = $this->Tat_model->get_companies();
    $data['all_employees'] = $this->Tat_model->all_employees();
    $role_resources_ids = $this->Tat_model->user_role_resource();
    if(in_array('30',$role_resources_ids)) {
        if(!empty($session)){
            $data['subview'] = $this->load->view("admin/attendance/update_attendance", $data, TRUE);
            $this->load->view('admin/layout/layout_main', $data);
        } else {
            redirect('admin/');
        }
    } else {
        redirect('admin/dashboard'); } }
}

```

### C. Daily Attendance View

```
<?php /* Daily Attendance view */?>
<?php $session = $this->session->userdata('username');?>
<?php $get_animate = $this->Tat_model->get_content_animate();?>
<?php $user_info = $this->Tat_model->read_user_info($session['user_id']);?>
<div class="box mb-4 <?php echo $get_animate;?>">
    <div class="box-body">
        <div class="row">
            <div class="col-md-12">
                <?php $attributes = array('name' => 'attendance_daily_report', 'id' => 'attendance_dai-ly_report', 'autocomplete' => 'off', 'class' => 'add form-tat');?>
                <?php $hidden = array('user_id' => $session['user_id']);?>
                <?php echo form_open('admin/attendance/attendance_list', $attributes, $hidden);?>
                <?php
                    $data = array(
                        'type'      => 'hidden',
                        'name'       => 'date_format',
                        'id'         => 'date_format',
                        'value'      => $this->Tat_model->set_date_format(date('Y-m-d')),
                        'class'      => 'form-control',
                    );
                    echo form_input($data);
                ?>
                <div class="row">
                    <?php if($user_info[0]->user_role_id==1){ ?>
                        <div class="col-md-4">
                            <div class="form-group">
                                <label for="name"><?php echo $this->lang->line('left_location');?></label>
                                <select name="location_id" id="location_id" class="form-control" data- plugin="select_tat" data-placeholder="<?php echo $this->lang->line('left_location');?>">
                                    <option value="0"><?php echo $this->lang->line('tat_acc_all');?></option>
                                    <?php foreach($all_office_shifts as $elocation) {?>
                                        <option value="<?php echo $elocation->location_id?>"><?php echo $elocation->location_name?></option>
                                    <?php } ?>
                                </select>
                            </div>
                        </div>
                    <?php } else {?>
                        <input type="hidden" value="0" name="location_id" id="location_id" />
                    <?php } ?>
                    <div class="col-md-4">
                        <div class="form-group">
                            <label for="first_name"><?php echo $this->lang->line('tat_e_details_date');?></label>
```

```

        <input class="form-control attendance_date" placeholder=<?php echo $this->lang-
>line('tat_select_date');?>" readonly id="attendance_date" name="attendance_date" type="text"
value=<?php echo date('Y-m-d');?>">
    </div>
</div>
<div class="col-md-4">
    <div class="form-group"> &nbsp;
        <label for="first_name">&nbsp;</label><br />
        <button type="submit" class="btn btn-primary save"><?php echo $this->lang-
>line('tat_get');?></button>
    </div>
</div>
<?php echo form_close(); ?> </div>
</div>
</div>
<div class="box <?php echo $get_animate;?>">
    <div class="box-header with-border">
        <h3 class="box-title"><?php echo $this->lang->line('tat_daily_attendance_report');?></h3>
    </div>
    <div class="box-body">
        <div class="box-datatable table-responsive">
            <table class="datatables-demo table table-striped table-bordered" id="tat_table">
                <thead>
                    <tr>
                        <th colspan="2"><?php echo $this->lang->line('tat_hr_info');?></th>
                        <th colspan="9"><?php echo $this->lang->line('tat_attendance_report');?></th>
                    </tr>
                    <tr>
                        <th style="width:120px;"><?php echo $this->lang->line('tat_employee');?></th>
                        <th style="width:100px;"><?php echo $this->lang->line('left_company');?></th>
                        <th style="width:100px;"><?php echo $this->lang-
>line('tat_e_details_date');?></th>
                        <th style="width:100px;"><?php echo $this->lang-
>line('dashboard_tat_status');?></th>
                        <th style="width:100px;"><?php echo $this->lang-
>line('dashboard_clock_in');?></th>
                        <th style="width:100px;"><?php echo $this->lang-
>line('dashboard_clock_out');?></th>
                        <th style="width:100px;"><?php echo $this->lang->line('dashboard_late');?></th>
                        <th style="width:100px;"><?php echo $this->lang-
>line('dashboard_early_leaving');?></th>
                        <th style="width:100px;"><?php echo $this->lang-
>line('dashboard_overtime');?></th>

```

```

        <th style="width:100px;"><?php echo $this->lang-
>line('dashboard_total_work');?></th>
        <th style="width:100px;"><?php echo $this->lang-
>line('dashboard_total_rest');?></th>
    </tr>
</thead>
</table>
</div>
</div>
</div>
```

## D. Finance Model

```

<?php
/**
 * Finance Model: This model contains models and schema for the Finance (2nd) Module of Tatari
 * System.
 * It will manage the ledger account along with transactions on different accounts.
 *
*/
defined('BASEPATH') OR exit('No direct script access allowed');

class Finance_model extends CI_Model {

    public function __construct() {
        parent::__construct();
        $this->load->database();
    }
    // Read from DB - Finance related tables
    public function get_bankcash() {
        return $this->db->get("tat_finance_bankcash");
    }
    public function get_deposit() {
        return $this->db-
>query("SELECT * from tat_finance_transaction where transaction_type = 'income'");
    }

    public function get_expense() {
        return $this->db-
>query("SELECT * from tat_finance_transaction where transaction_type = 'expense'");
    }

    public function get_invoice_payments() {
        return $this->db->query("SELECT * from tat_finance_transaction where invoice_id != ''");
    }
```

```

public function get_transfer() {
    return $this->db->get("tat_finance_transfer");
}

public function get_transaction() {
    return $this->db-
>query("SELECT * from tat_finance_transaction order by transaction_id desc");
}

public function all_payees()
{
    $query = $this->db->query("SELECT * from tat_finance_payees");
    return $query->result();
}

public function get_bankwise_transactions($id)
{
    $sql = "SELECT transaction_date,dr_cr,amount,account_id,transaction_type,description,IF(dr_cr='dr',amount,NULL) as debit,IF(dr_cr='cr',amount,NULL) as credit FROM tat_finance_trans action WHERE account_id='$id'";
    $binds = array($id);
    $query = $this->db->query($sql, $binds);
    return $query;
}

public function get_payers()
{
    return $this->db->get("tat_finance_payers");
}

public function get_payees()
{
    return $this->db->get("tat_finance_payees");
}

public function all_bank_cash()
{
    $query = $this->db->query("SELECT * from tat_finance_bankcash");
    return $query->result();
}

public function all_payers()
{
    $query = $this->db->query("SELECT * from tat_finance_payers");
    return $query->result();
}
?>
```

## E. Finance Controller

```
<?php
/*
*
* Finance Module Controller: here are some of the controllers that receive data from view and
process it though the model regarding
* the finance features such as ledger account management, deposit, expense, transfer and all
transactions along with the payer and payee
* information.
*
*/
defined('BASEPATH') OR exit('No direct script access allowed');
class Finance extends MY_Controller
{

    public function output($Return=array()){
        header("Access-Control-Allow-Origin: *");
        header("Content-Type: application/json; charset=UTF-8");
        exit(json_encode($Return));
    }
    public function __construct() {
        parent::__construct();
        $this->load->model('Finance_model');
        $this->load->model('Expense_model');
        $this->load->model('Invoices_model');
        $this->load->model('Employees_model');
        $this->load->model('Department_model');
        $this->load->model('Tat_model');
    }
    public function read() {
        $session = $this->session->userdata('username');
        if(empty($session)){
            redirect('admin/');
        }
        $data['title'] = $this->Tat_model->site_title();
        $id = $this->input->get('bankcash_id');
        $result = $this->Finance_model->read_bankcash_information($id);
        $data = array(
            'bankcash_id' => $result[0]->bankcash_id,
            'account_name' => $result[0]->account_name,
            'account_balance' => $result[0]->account_balance,
            'account_number' => $result[0]->account_number,
            'branch_code' => $result[0]->branch_code,
        )
    }
}
```

```

        'bank_branch' => $result[0]->bank_branch,
        'created_at' => $result[0]->created_at
    );
}

if(!empty($session)){
    $this->load->view('admin/finance/dialog_accounting', $data);
} else {
    redirect('admin/');
}
}

public function account_balances() {

    $session = $this->session->userdata('username');
    if(empty($session)){
        redirect('admin/');
    }
    $system = $this->Tat_model->read_setting_info(1);
    if($system[0]->module_finance!='true'){
        redirect('admin/dashboard');
    }
    $data['title'] = $this->lang->line('tat_acc_account_balances').' | '.$this->Tat_model->site_title();
    $data['breadcrumbs'] = $this->lang->line('tat_acc_account_balances');
    $data['path_url'] = 'finance_account_balances';
    $role_resources_ids = $this->Tat_model->user_role_resource();
    if(in_array('73',$role_resources_ids)) {
        if(!empty($session)){
            $data['subview'] = $this->load-
>view("admin/finance/account_balances", $data, TRUE);
            $this->load->view('admin/layout/layout_main', $data);
        } else {
            redirect('admin/');
        }
    } else {
        redirect('admin/dashboard');
    }
}

public function accounts_ledger() {

    $id = $this->uri->segment(4);
    $bac_id = $this->Finance_model->read_transaction_by_bank_info($id);
    if(is_null($bac_id)){
        redirect('admin/finance/transactions');
    }
    $system = $this->Tat_model->read_setting_info(1);
}

```

```

        $data['title'] = $this->lang->line('tat_acc_ledger_account').' | '.$this->Tat_model-
>site_title();
        $data['breadcrumbs'] = $this->lang->line('tat_acc_ledger_account');
        $data['path_url'] = 'finance_bankwise_transactions';
        $session = $this->session->userdata('username');
        $role_resources_ids = $this->Tat_model->user_role_resource();
        if(!empty($session)){
            if(in_array('4',$role_resources_ids)) {
                $data['subview'] = $this->load-
>view("admin/finance/ledger_account_list", $data, TRUE);
                $this->load->view('admin/layout/layout_main', $data);
            } else {
                redirect('admin/dashboard');
            }
        } else {
            redirect('admin/');
        }
    }

    public function ledger_accounts() {

        $system = $this->Tat_model->read_setting_info(1);
        $data['title'] = $this->lang->line('tat_acc_ledger_account').' | '.$this->Tat_model-
>site_title();
        $data['breadcrumbs'] = $this->lang->line('tat_acc_ledger_account');
        $data['path_url'] = 'tat_ledger_accounts';
        $session = $this->session->userdata('username');
        $role_resources_ids = $this->Tat_model->user_role_resource();
        if(!empty($session)){
            if(in_array('4',$role_resources_ids)) {
                $data['subview'] = $this->load-
>view("admin/finance/full_ledger_account_list", $data, TRUE);
                $this->load->view('admin/layout/layout_main', $data);
            } else {
                redirect('admin/dashboard');
            }
        } else {
            redirect('admin/');
        }
    }

    public function transactions() {

        $session = $this->session->userdata('username');
        if(empty($session)){

```

```

        redirect('admin/');
    }
    $system = $this->Tat_model->read_setting_info(1);
    if($system[0]->module_finance!='true'){
        redirect('admin/dashboard');
    }
    $data['title'] = $this->lang->line('tat_acc_view_transactions').' | '.$this->Tat_model->site_title();
    $data['breadcrumbs'] = $this->lang->line('tat_acc_view_transactions');
    $data['path_url'] = 'finance_transactions';
    $role_resources_ids = $this->Tat_model->user_role_resource();
    if(in_array('78',$role_resources_ids)) {
        if(!empty($session)){
            $data['subview'] = $this->load-
>view("admin/finance/transaction_list", $data, TRUE);
            $this->load->view('admin/layout/layout_main', $data);
        } else {
            redirect('admin/');
        }
    } else {
        redirect('admin/dashboard');
    }
}

```