

## **Office Management & HR Portal**

Project report submitted in partial fulfillment of the requirement for  
the degree of  
Bachelor of Technology  
in  
**Computer Science and Engineering**

By

Siraj Sharma (181438)

to



Department of Computer Science & Engineering and Information  
Technology

**Jaypee University of Information Technology, Waknaghat,**

**Solan-173234**

**Himachal Pradesh**

## Candidate's Declaration

I hereby declare that the work presented in this report entitled “**Office Management & HR Portal**” in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic of my work carried out over a period from February 2022 to June 2022 under the supervision of **Dr Pankaj Dhiman** (Assistant Professor (Grade-II)).

The matter embodied in the report has not been submitted for the award of any degree or diploma.

Siraj Sharma, 181438

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr Pankaj Dhiman

Assistant Professor (Grade-II)

Computer Science & Engineering and Information  
Technology

Dated:

## **Acknowledgement**

This is a matter of pleasure for me to acknowledge my deep sense of gratitude to Jaypee University of Information Technology for giving me an opportunity to explore my abilities via this internship program. I would like to express my sincere gratitude to our Training and Placement officer, Mr. Pankaj Kumar, and our faculty Coordinator, Dr Nafis U Khan for this opportunity. I also wish to express my gratitude to my internship supervisors, for their valuable guidance and advice towards my internship.

I would like to record my sincere appreciation and gratitude towards all the officials, coaches, trainers, mentors, and my colleagues of VectoScalar Technologies, without whose kind assistance, my internship program would not have been proceeding in a swift direction. The facts and other vital information provided by them have contributed towards making this report as comprehensive as possible. I am indeed thankful to them.

Last but not the least, I would like to express my sincere thanks to all my family members, friends and well-wishers for their immense support and best wishes throughout the internship duration and the preparation of this report and I wish they would continue to contribute towards my well-being.

I believe that this report will be an asset not only for academic institution but will also be useful for all those who are interested to learn about internship experiences in software development and consulting company.

Siraj Sharma, 181438

Jaypee University of Information Technology,

Waknaghat, Solan, HP

# Table of Content

<b>CANDIDATE’S DECLARATION</b>	<b>1</b>
<b>ACKNOWLEDGEMENT</b>	<b>2</b>
<b>TABLE OF CONTENT</b>	<b>3</b>
<b>LIST OF FIGURES</b>	<b>5</b>
<b>LIST OF TABLES</b>	<b>6</b>
<b>ABSTRACT</b>	<b>7</b>
<b>1. CHAPTER–1 INTRODUCTION</b>	<b>8</b>
1.1 Introduction	8
1.2 Problem Statement	9
1.3 Objectives	9
1.4 My Role	10
1.5 Methodology	13
1.6 Organization	14
<b>3. CHAPTER–2 LITERATURES SURVEY</b>	<b>15</b>
2.1 Introduction	15
2.2 Literature Reviews on the Project	15
2.3 Literature Reviews on the Software Methodology	16
<b>3. CHAPTER–3 SYSTEM DEVELOPMENT</b>	<b>17</b>
3.1 Requirements & Analysis	17
3.2 Design	18
3.3 Development	29
<b>4. CHAPTER–4 TESTING &amp; RESULTS</b>	<b>32</b>
4.1 Testing Activities	32

<b>4.2 Test Environment</b>	<b>32</b>
<b>4.3 Scope of Testing</b>	<b>32</b>
<b>4.4 Test Execution</b>	<b>34</b>
<b>4.5 Results</b>	<b>35</b>
<b>5. CHAPTER-5 CONCLUSIONS</b>	<b>37</b>
<b>5.1 Conclusions</b>	<b>37</b>
<b>REFERENCES</b>	<b>38</b>

## **List of Figures**

FIGURE 1: AGILE METHODOLOGY	14
FIGURE 2: SCRUM FRAMEWORK OF AGILE METHODOLOGY	16
FIGURE 3: 3-TIER ARCHITECTURE DIAGRAM	20
FIGURE 4: USE CASE DIAGRAM	26
FIGURE 5: MVC	30
FIGURE 6: ESS PORTAL - REIMBURSEMENTS MODULE	35
FIGURE 7: ESS PORTAL – REIMBURSEMENT MODULE AFTER SELECTING OPTION	35
FIGURE 8: PROJECT MANAGEMENT MODULE IN EMPLOYEE SIDE	36

## **List of Tables**

TABLE 1.1: REIMBURSEMENT STATES	11
TABLE 1.2: NATURE OF EXPENSES	12
TABLE 1.3: PROJECT STATES	12
TABLE 1.4: PROJECT TYPES	13
TABLE 1.5: ACCESS CONTROL VIA ROLES	13
TABLE 3.1: EMPLOYEE TABLE	21
TABLE 3.2: PROJECT TABLE	22
TABLE 3.3: TIMESHEET TABLE	23
TABLE 3.4: REIMBURSEMENT TABLE	24
TABLE 3.5: ADMIN TABLE	24
TABLE 3.6: TASKS TABLE	25
TABLE 3.7: CLIENT TABLE	25
TABLE 3.8: USER LOGIN USE CASE	27
TABLE 3.9: INITIATE REIMBURSEMENT USE CASE	27
TABLE 3.10: APPROVE/REJECT REIMBURSEMENT USE CASE	28
TABLE 3.11: VIEW PROJECTS USE CASE	29
TABLE 3.12: PROJECT MANAGEMENT USE CASE	29
TABLE 4.1: TEST ENVIRONMENT	32
TABLE 4.2: UNIT TESTCASES TABLE	34

## **Abstract**

No company wants to waste its resources and time on daily routine tasks of organization that has no direct relation to organizational productivity but are an essential part of it. **Office Management & HR Portal** System is a combination of tools and processes that helps in managing company's work force (i.e., its employees) from an admin and management viewpoint. **Office Management & HR Portal** is a single point entry to all its employees for efficiently utilizing company's resources. Portal consists of company's intellectual property like details of all its employees. There are several processes developed to tackle other broad categories of portal system. Categories are Admin of employees, a timesheet, reimbursement, leave management, project management, etc. on the online portal for its employees.



# Chapter–1 Introduction

## 1.1 Introduction

**VectoScalar** began its eventful operations into Software/Application Development in 2010. Based in Noida, India we have addressed the growing enterprise software business needs, in a cost & time effective manner. We cover all stages of the application development, quality assurance, and post-project maintenance and support. By adhering to the continuously evolving processes, we have ensured high quality work results and long-lasting relationship with our clients. We build native, hybrid, and cross-platform apps that run on all major operating systems such as iOS and Android as well as Web and Desktop applications.

### **Industry Expertise**

#### ***AWS Service:***

- Expertise in Web Hosting and Application Deployment. Leveraging deep skills to turn idea into a creative user experience.

#### ***iOS Applications:***

- Great experience in building high class and beautiful iOS applications for both iPhone and iPad

#### ***Web Development:***

- Innovative web solutions to deliver unparalleled high performance. Create awesome transformative web experience

#### ***Android Applications:***

- Developing engaging mobile experiences, while taking care of everything from design to development for your big ideas

#### ***Chrome app:***

- Good expertise in creating a better-quality Chrome app, which can be a better solution of any desktop application

#### ***Amazon Alexa:***

- Great experience in Amazon's Alexa voice service. Provides skills that enable customers to create better experience

### ***Hybrid Mobile Applications:***

- Deep skills, robust process, and flexibility to develop paramount hybrid solutions for IONIC and METEOR mobile applications

### ***Desktop Applications:***

- We provide cutting-edge solutions for Desktop and Native applications for all operating systems

### ***Beacon:***

- Give users better proximity experiences by providing a strong context signal for their devices in the form of Bluetooth

## **1.2 Problem Statement**

No company wants to waste its time and resources on daily menial tasks of organization that has nothing to do with organization productivity but are an essential part of it.

So, we needed to develop a tool which can manage company's employees/project details in an automated manner which is currently being managed manually using excel sheets.

The tool that can track employee daily, lock employee's work hours, attendance, reimbursements to determine the employee's performance on a particular project and to get realistic billable amount from the client.

## **1.3 Objectives**

employee's work details along with effort spend per day at one place (in-line) for multiple projects and helps supervisors to monitor supervises task status and time spent on task.

This application has mainly followed parts: –

- Employee
- Timesheet
- **Reimbursement**
- Leave Management System
- **Project Management**

**Users of the System:** –

**Employee** is the normal user of the tool and has following rights in the application: –

- Log in with their "Username" and "Password" to the system

- Edit their profile
- Search and view other employees
- Fill in the timesheet
- Initiate reimbursement request
- Can see projects he/she has been assigned, etc.

**Admin** has all rights throughout the application. In addition to employee rights, they have extra rights in the application: –

- Login to the system using their admin “Username” and “Password”
- Add new employees or edit existing employees
- Admin should be able to approve or reject the employee’s timesheet the reimbursements
- Admin generate invoice based on the employee work effort spend

**Functionality:** –

- An employee should be able to login
- An admin has permissions to view all the modules
- An employee can access modules that are accessible to him
- Admin can configure system, add employees, and update details
- In employee details, we can view basic information of an employee
- An employee can view holidays
- An employee can add time sheets for the work they have done
- An employee can view their past leave summary
- In EIM, we can view employee HR Records, Personal Records
- An employee can apply for their reimbursement

## 1.4 My Role

My responsibility is to create and test the modules: –

- Reimbursement Management
- Project Management

**Reimbursement:** –

- Employees can instigate reimbursements for the expenses they have incurred
- Submit reimbursement for approval, in case rejected can edit their reimbursement

- Reimbursement is approved or rejected by respective “Reimbursement Supervisor” as assigned by the admin
- Admin should be able to approve or reject the employee’s reimbursement

***Reimbursement States: –***

Reimbursement must be in one of the following states: –

<b>States</b>	<b>Description</b>
Draft	Initial state when user can edit
Submit	Submitted to admin
Under admin process	Admin will process
Approved by admin	If admin approves
Rejected	If admin rejects
Under Finance process	Finance will process
Approved	If finance approves
Paid	If reimbursement paid to user

*Table 0.1: Reimbursement States*

***Nature of Expenses: –***

<b>Types</b>	<b>Description</b>
Meals	Bills for meals with value cap
Conveyance	Bills for conveyance
Hotel	Bills for hotel
Flight	Bills for flight
Birthday Vouchers	Can redeem vouchers

Others	Bills for other expenses
--------	--------------------------

*Table 0.2: Nature of Expenses*

### **Project Management**

- This module manages the information about the client-projects
- Admin will have the privileges to add new project and configure project based on project types
- Admin can add employees to project & upload scanned contracts
- Employees can see in what projects they have been assigned by the admin

#### ***Project States: –***

Project must be in one of the following states:

<b>States</b>	<b>Description</b>
Open	Admin has privileges to change the status of projects
Closed	Admin has privileges to change the status of projects

*Table 0.3: Project States*

#### ***Project Types: –***

Project must be in one of the following types: –

<b>Types</b>	<b>Description</b>
Retainer ship	A fixed monthly amount paid irrespective of man days
Fixed	A fixed monthly amount paid based on number of man days
Timesheet	Depending on number of hours

Retainer ship + Timesheet	Combination of Retainer ship & Timesheet
---------------------------	--

*Table 0.4: Project types*

### **Access Control via Roles**

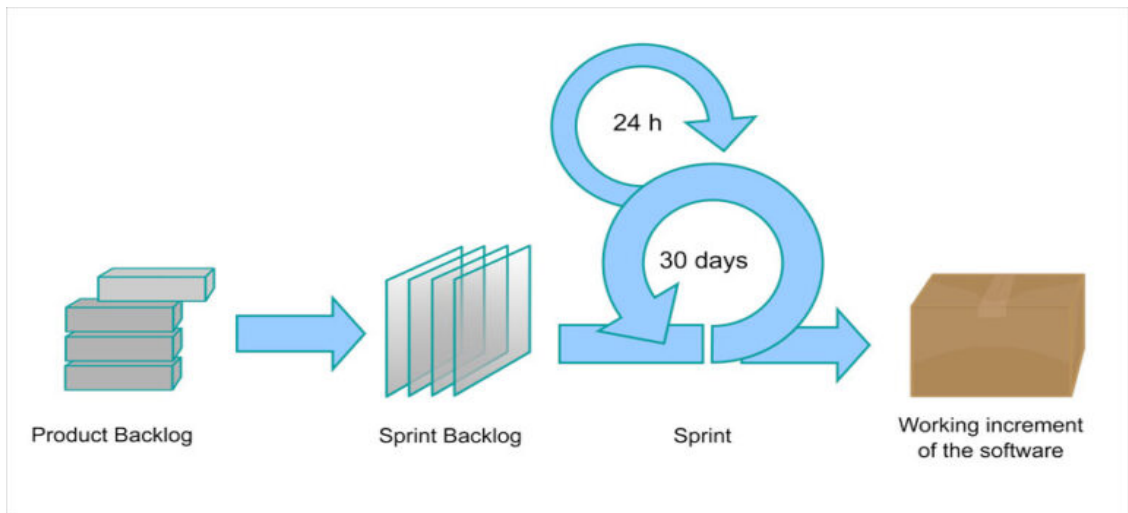
<b>Types</b>	<b>Privileges</b>
Employee	Employee can fill timesheet & raise reimbursements
Supervisor	First level approver. Supervisor can approve timesheet & reimbursements
Admin	After supervisor approval, admin approval comes
Finance	After admin approval, finance role comes. Finance pays the amount to the employee. Finance can generate invoice
Director	Have all the required privileges

*Table 0.5: Access Control via Roles*

## **1.5 Methodology**

### **Methodology Used**

The project in question employs an agile iterative development technique, in which requirements and solutions because of engagement with the client. Agile development refers to the process for creating desktop software. It is more robust and efficient in a shorter time frame than other models, and it involves face-to-face contact as well as technical personnel and consumers as team members. Project managers and business analysts are used in agile desktop development, with a focus on clearly defined goals, organization, and iterative delivery. Agile development guarantees that the product is completed successfully at the conclusion of each iteration.



[This Photo](#) by Unknown Author is licensed under [CC BY](#)

*Figure 1: AGILE Methodology*

## **1.6 Organization**

The material presented in this report is organized into five chapters. After this introductory chapter, chapter 2 describes the “Literature Survey” for understanding the concepts, analyzing the research, finding connections, and planning the implementation.

Chapter 3 provides the details about the architecture used, analysis (functional and reliability requirements), design (use case of the project, flow diagram development architecture), development (it contains the system model and DFD models).

Chapter 4 presents the “Performance analysis” of the system in the different testing phases of the system.

Finally, Chapter 5 “Conclusions” compiles the overall work, future scope, and the application in which the developed system can be used.

### **3. Chapter–2 Literatures Survey**

#### **2.1 Introduction**

The examination of the literature related to the Employee Management System is summarized in this chapter. It investigates the project's theories, concepts, approaches, methodologies, and procedures. Existing technologies that are related to the creation of an EMS are discussed.

#### **2.2 Literature Reviews on the Project**

A human resource management system (HRMS) is a set of systems and procedures that sit at the crossroads of HRM and IT. It combines HRM as a subject, as well as core HR activities and procedures, with the realm of IT, while data processing system programming developed into procedures and software packages of ERP. [1]

A corporation or organization with a big number of people manages a larger amount of data. Without a more advanced tool for storing and retrieving data, this task might be overwhelming. The evolution of HR technology may be used to investigate the various degrees of complexity. Paper-based platforms, initial PC technology, digital libraries, and Internet technology are the four stages of development for these aspects. [2]

HR and other divisions of the organization are becoming more aware of the advantages of digitization. To gain productivity, the focus has turned to digitizing as many activities as feasible.

The future of the internet will be about quick access to current, correct data, and the way to obtain this data across many platforms will provide businesses a strategic advantage. HR is projected to abandon its position as the only owner of HR data, allowing managers and employees to use Web-based technologies to address their own problems. The proposed system will not imply a reduction in HR personnel. The new system will allow HR experts to concentrate on translating information into usable that can be utilized by the business to make decisions; it will be a case of HR and IT collaborating to make the most of technology. Since they use technology successfully, high-performing firms spend 25% less on HR than their rivals, according to recent research even by Hackett Group, a workflow advising firm. [3]

Employee self-service and manager self-service are the most prominent Web-based HR tools nowadays. These systems have allowed firms to delegate responsibility for seeing and



modifying data to individual employees, transforming the way employees obtain information and interact with HR departments. [4]

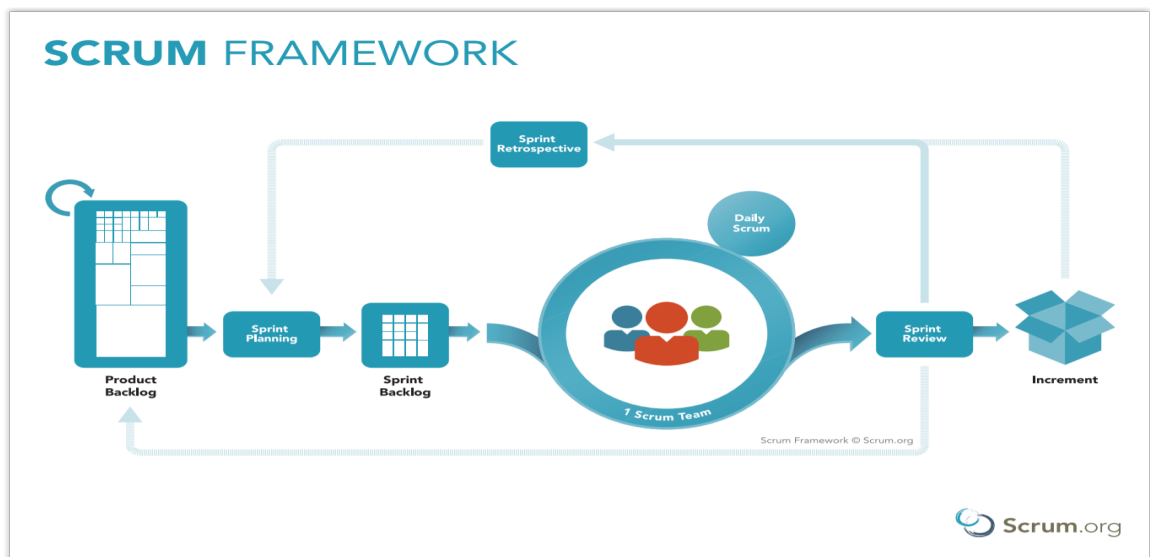
### 2.3 Literature Reviews on the Software Methodology

Software development methodology is a set of methods, techniques, tools, and documentation that assist developers in the implementation of a system. [5]

There are several software development methodologies, each of which is chosen based on a variety of project-related factors, such as time, cost, incorporating requirement changes during in the project development, system complexity, customer-developer communication, software critically, and the size of the development team. These generic models aren't meant to be exact representations of software processes. Rather, they are process abstractions that may be used to describe various software development methodologies. They may be thought of as process frameworks that can be customized and expanded to build more particular software engineering processes. Here is selected model:

#### Scrum Methodology

Scrum is an agile framework that assists businesses in facilitating team cooperation and simplifying difficult projects by specifying what the Scrum team must achieve, how they should complete the tasks, but not in detail. In essence, it is not a rigid methodology because it merely gives the structure of the methodology, letting scrum teams to develop their own, stressing cooperation. [6]



[This Photo](#) by Unknown Author is licensed under [CC BY-SA](#)

Figure 2: Scrum Framework of Agile Methodology

## **3. Chapter–3 System Development**

### **3.1 Requirements & Analysis**

The analysis included these steps: –

- Built the employee requirements
- Performed interviews, group discussions, brain storming sessions
- Identified the project team
- Established the development environment
- Identify the client requirements
- Set the project scope and schedule

**Functional Requirements: –**

- All users must login with their username and password
- Role Based Security Needed
- User can view and manage his / her profile on the portal
- Role based security should be there
- Timesheet should be submitted by employee at the end of the week which includes employee weekly status
- Employee can send Reimbursement request for the expenses he/she have incurred

**Reliability Requirements: –**

- The application's state should be consistently saved by the system

**Hardware Requirement: –**

- **Architecture** x86 or x86-64bit hardware architecture
- **Processing Power** Core 2 Duo 2.4GHz processor or faster
- **Memory** 4GB of RAM

**Software Requirement: –**

- **Library** ReactJS
- **Runtime Environment** NodeJS
- **Database** MySQL
- **Web Browser** Google Chrome

## **3.2 Design**

Design phase commences after the requirements finalized and frozen. The design phase attempts to uncover various entities involved in the system and their associated behaviour and also the interfaces that would be provided by the system. Data Flow Diagrams for the system are developed.

### **System Interfaces**

The following interfaces are part of the overall system: –

#### ***Login Page***

- The user can authenticate using this interface
- Authentication is based on a form
- This individual is in charge of authentication and authorisation

#### ***Home Page***

- This is the page that appears once the user logs in
- This page is accessible to both employee and admin user

#### ***Employees***

- User / Admin clicks on Employee Tab
- This page displays employee details

#### ***Timesheet***

- Timesheet is used to eliminate manual processes of time tracking
- Timesheet management is beneficial to both employees & project managers to track time spent on each task in a project
- Employee can create, save and submit timesheet
- Supervisor can view and can either approve or reject timesheet
- Admin should be able to reject timesheets post approval from Supervisor
- Timesheet will either be in one of the states: –
  - Draft
  - Submitted
  - Pending
  - Rejected
  - Approve

- Done

### ***Reimbursement***

- Employee can fill reimbursement and submit to supervisor
- Supervisor can either approve or reject reimbursement
- Reimbursement will either be in one of the states: –
  - Draft
  - Submitted
  - Under Admin process
  - Approved by admin
  - Rejected
  - Under Finance process
  - Approved
  - Paid

### ***Project Management***

- Project Management module manages the information about client projects
- Admin will have privilege to add new project & configure project based on project types
- Admin can add employees to project & upload scanned contracts

### ***Leave Management System***

- Employee can apply leave, view their leave summary

### ***Report***

- Admin / Finance can generate reports
- Reports can be generated either client-wise, project-wise

### **Architecture used**

#### **3-Tier Architecture used with defined layers**

Layered architecture is concerned with the division of related functionality inside an application into discrete layers that are layered vertically on top of one another. Layer communication is transparent and loosely connected.

Layering the application appropriately helps to support a strong separation of concerns that, in turn, supports flexibility and maintainability. At the most higher level of abstraction, the

logical architecture of any system may be viewed as a collection of interacting components organised into layers.

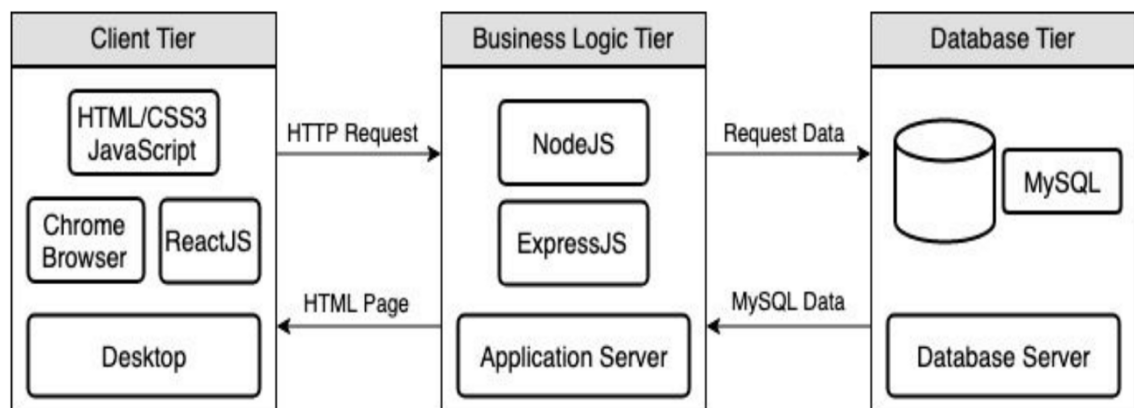


Figure 3: 3-Tier Architecture Diagram

### ***Presentation Layer***

The Presentation Tier (UI) is the topmost layer of an application where the user interacts. This layer mostly handles user input validation and constraint processing. UI layer interface with controller classes in layered architecture of application.

### ***Business Logical Layer***

It normally comprises of modules, some of which may provide effective engagement that other callers may use. Business Logic Layer performs the fundamental function of the software and encapsulates the required business logic and validations that must be implemented on the input data. It manages the functionality of an application by doing thorough processing.

### ***Data Access Layer***

The Data Access Layer enables access to data housed within the system's limits as well as data accessible by other networked systems, which may be accessed via services. The data layer offers generic APIs that the business layer modules can use.

## **Logical Database Design**

### ***Employee Table***

Field	Control	Description	Constraint
First Name	Textbox	First Name	Mandatory

Last Name	Textbox	Last Name	Mandatory
Date of Birth	Date Picker	DOB	Mandatory
Contact Email	Textbox	Email Id	Mandatory, Validation for email-di
Contact Mobile#	Textbox	Mobile Number	Mandatory, Input only numbers
Designation	Select Box	Profile	Mandatory
Department	Select Box	Department	Mandatory
Address	Textbox	Permanent Address	Mandatory
Blood Group	Textbox	Blood Group	Optional, Input only characters
Date of joining	Date Picker	Date of Joining	Mandatory
Remarks	Textbox	Remark	Optional

*Table 3.1: Employee Table*

***Project Table***

<b>Field</b>	<b>Control</b>	<b>Description</b>	<b>Constraint</b>
Project Name	Textbox	Project Name	Mandatory
Project Type	Select Box	Type of Projects	Mandatory
Description	Textbox	Project Details	Optional
Customer	Dropdown	Customer/Clients	Mandatory
Billing Method	Select Box	Type of Billing Method	Mandatory

Team Lead	Select Box	Employee (Team Lead)	Mandatory
Team Details	Checkbox	Number of Employees	Mandatory
Number of Employees	Textbox	Total Number of Employee	Mandatory, Input only numbers
Notes	Textbox	Textbox	Optional
Contract	File	Scanned image of contract	Optional
Engagement date	Text field	Date	Optional
Duration	Number	Duration details for project	Based on project type
Per Man day rate	Number	Per Man day rate	Based on project type
Agreement	File	Employee	Optional

*Table 3.2: Project Table*

***Timesheet Table***

<b>Field</b>	<b>Control</b>	<b>Description</b>	<b>Constraint</b>
Project Name	Select Box	Project Name	Mandatory
Work Details	Textbox	Descriptions	Mandatory
Time Spent	Textbox	HH:MM	Mandatory, Input only Numbers
Start Date	Date Picker	Start Date	Mandatory, Input only back-date by 'x' no. of days.

End Date	Date Picker	End Date	Mandatory. Input date more than start-date
Submitted Date	Date Picker	Timesheet Submit date	Mandatory. Current Date

*Table 3.3: Timesheet Table*

***Reimbursement Table***

<b>Field</b>	<b>Control</b>	<b>Description</b>	<b>Constraint</b>
Project Name	Select Box	Project Name	Mandatory
Employee Name	Textbox	Employee Name	Disabled
Reimbursement Type	Select Box	Reimbursement Type	Mandatory
Reimbursement Title	Textbox	HH:MM	Mandatory, Input only Numbers
Amount	Textbox	Start Date	Mandatory, Input only back-date by 'x' no. of days
Submission date	auto	Timestamp of the submission	Mandatory
Travel Period From/ to	Date Picker	Users start and end date for travel	Mandatory. Current Date



No of days	Input box	Days of travel period	Enter manually
Per diem per day	Input box	Amount for the per diem (for single day)	Enter manually
Currency	Input box	Currency to be converted	Auto fill
Rate	Input box	Rate for the currency conversion	Enter manually

*Table 3.4: Reimbursement Table*

***Admin Table***

<b>Field</b>	<b>Control</b>	<b>Description</b>	<b>Constraint</b>
Full Name	Textbox	Full Name	Mandatory
Email	Textbox	Email	Mandatory

*Table 3.5: Admin Table*

***Tasks Table***

<b>Field</b>	<b>Control</b>	<b>Description</b>	<b>Constraint</b>
Status	Textbox	Status	Mandatory
Update By	Text Box	Updated by	Mandatory
Approved By	Textbox	Approved by	Mandatory
Deleted By	Textbox	Deleted by	Mandatory

End Date	Date Picker	End date	Mandatory
Start Date	Date Picker	Start date	Mandatory
Comment	Textbox	Comment	Mandatory
Updated At	Select box	Updated at	Mandatory
Created By	Textbox	Created by	Mandatory

*Table 3.6: Tasks Table*

***Client Table***

<b>Field</b>	<b>Control</b>	<b>Description</b>	<b>Constraint</b>
First Name	Textbox	First name	Mandatory
Last Name	Textbox	Last name	Mandatory
Project Details	Textbox	Project details	Mandatory

*Table 3.7: Client Table*

## Use Case Description

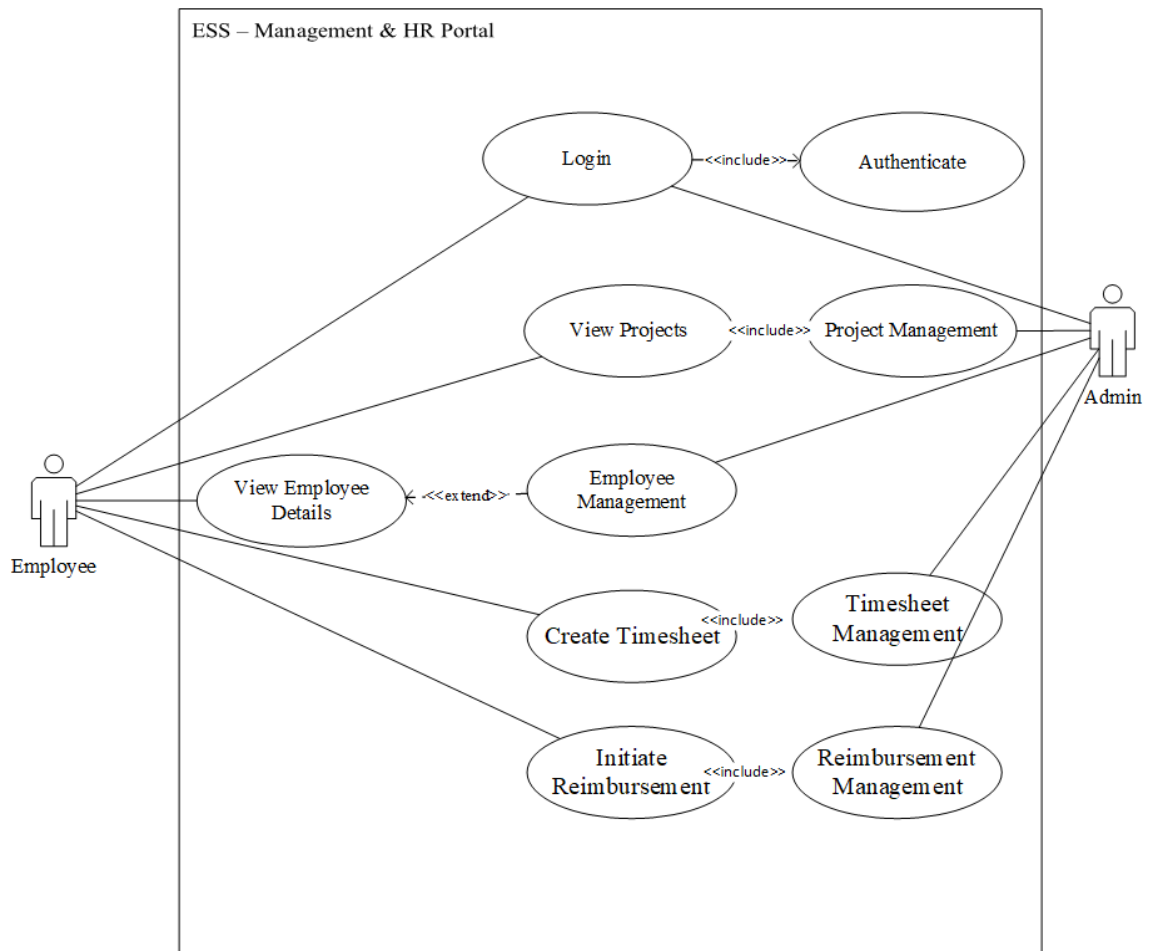


Figure 4: Use Case Diagram

### User Login

Use Case Name	Login
Purpose	To login into the system
Actors	Admin, Employee
Preconditions	User must have proper credentials
Post Conditions	User redirects to the dashboard

Basic Flow	<ul style="list-style-type: none"> <li>• User enters a username and password to login into the system</li> <li>• Password validates, if username and password is valid, user is successfully logged in the system</li> <li>• If username and password is not valid, user can't be able to logged-in to the system</li> <li>• User redirects to the dashboard page</li> </ul>
Alternate Flows	If username and password is not valid, user can't be able to logged-in to the system. User remains in the login page

*Table 3.8: User Login Use Case*

***Initiate Reimbursement***

<b>Use Case Name</b>	<b>Initiate Reimbursement</b>
Purpose	User can Initiate a reimbursement request
Actors	Supervisor, Employee
Preconditions	User must login into the system
Post Conditions	Reimbursement status changed and reimbursement request is sent to supervisor for approval
Basic Flow	<ul style="list-style-type: none"> <li>• User clicks on initiate reimbursement button</li> <li>• Selects project, enters reimbursement amount</li> <li>• Save and submit reimbursement request</li> </ul>
Alternate Flows	N/A

*Table 3.9: Initiate Reimbursement Use Case*

***Approve or Reject Reimbursement Request***

<b>Use Case Name</b>	<b>Approve or Reject Reimbursement</b>
Purpose	Whether reimbursement will approve or reject
Actors	Admin, Supervisor
Preconditions	User must request reimbursement and send for approval
Post Conditions	Timesheet status will be changed
Basic Flow	<ul style="list-style-type: none"> <li>• User submits a reimbursement request for approval</li> <li>• Supervisor views pending reimbursement</li> <li>• Supervisor can either approve or reject reimbursement.</li> <li>• Reimbursement status will be changed</li> <li>• If reimbursement rejects, employee need to edit reimbursement and again send for approval</li> </ul>
Alternate Flows	Admin can also reject reimbursement

*Table 3.10: Approve/Reject Reimbursement Use Case*

***View Projects***

<b>Use Case Name</b>	<b>View Projects</b>
Purpose	To View Ongoing Projects assigned to an Employee
Actors	Employee
Preconditions	User must be logged in.

Post Conditions	N/A
Basic Flow	<ul style="list-style-type: none"> <li>• User clicks on view projects</li> <li>• User can then view projects assigned to him/her.</li> </ul>
Alternate Flows	N/A

*Table 3.11: View Projects Use Case*

### ***Project Management***

<b>Use Case Name</b>	<b>Project Management</b>
Purpose	To add new projects and to assign employee to a project
Actors	Admin
Preconditions	Admin must be logged in
Post Conditions	N/A
Basic Flow	<ul style="list-style-type: none"> <li>• Admin clicks on create project</li> <li>• Admin can then assign projects to employees</li> </ul>
Alternate Flows	<ul style="list-style-type: none"> <li>• Admin clicks on manage project</li> <li>• Admin can then assign projects to employees</li> </ul>

*Table 3.12: Project Management Use Case*

### **3.3 Development**

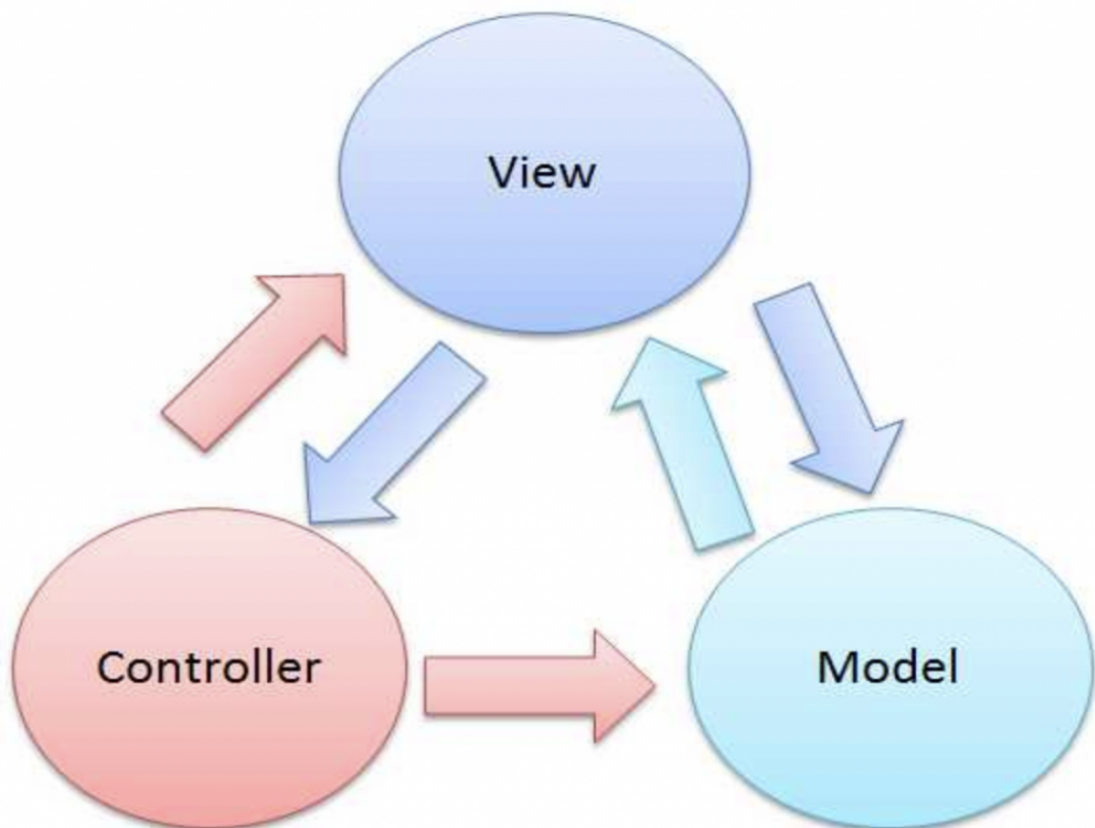
The system is installed on the a server with the Tomcat component. The user interacts with the server via a web browser, which makes an HTTP / HTTPS http get request, which responds upon parsing the requested page. As a result, the system is built as a Client-Server

paradigm. To obtain the data, the browser establishes a connection with the server, which responds with the required data sets.

**Model View Controller**, or **MVC** as it is more often known, is a framework that is designed used to create online applications. MVC pattern is formed up of the three components listed below: –

- **Model:** The pattern's lowest level, which is in charge of data maintenance
- **View:** This is in charge of showing everything and a part of the collected data
- **Controller:** The software code that governs the relationships in between Model and the View

MVC is widely used because it separates the logic from of the user interface layer and allows for the separation of responsibilities. The Controller accepts all application requests and afterwards work with model to generate any data required by the View. The View generates a finished presentable response using the data supplied by the Controller. The MVC concept may be represented visually as follows.



[This Photo](#) by Unknown Author is licensed under [CC BY-SA-NC](#)

*Figure 5: MVC*

### **The Model**

The model is in charge of the application's data management. It reacts to the view's request and also to the controller's commands to update itself.

### **The View**

It refers to the display of data in a certain format, which is prompted by a controller's choice to show the data. They are script-based template solutions that are simple to combine with AJAX.

### **The Controller**

The controller is in charge of responding to user inputs and interacting with data model objects. The controller receives the input, validates it, and then executes the business activity that changes the structure of the database schema.



## 4. Chapter–4 Testing & Results

### 4.1 Testing Activities

#### Project Initiation

Testing targets were set during this phase. The key duties in terms of testing were to structure the approach to testing and to begin building up a test environment if it did not already exist.

#### Development Team Testing

A whole-team method was used, in which employees with testing abilities were successfully integrated into the development team, and the team was in charge of the most of the testing. This method is effective.

#### End-of-lifecycle Testing

End-of-lifecycle testing, when an external test team confirms that the system is ready for production, was a significant aspect of the release effort.

### 4.2 Test Environment

Hardware	Software	Supporting Tools
Windows, Mac	Chrome, Safari, Edge, Firefox & Internet Explorer	Postman, Jest

*Table 4.1: Test Environment*

### 4.3 Scope of Testing

#### Modules to be Tested

- Reimbursement Module
- Project Management Module

#### Types of Testing

##### *Testing performed at module / project level*

- Unit Testing
- Individual units or components of software are tested

- Unit testing is a process of testing individual sections of a programme to see if they perform as planned
- For Unit Testing, Third Party Libraries should be used by developer like “Jest”
- Jest is a popular test framework, it runs test cases and reports any errors encountered during the process
- Test coverage report will be generated for verify whether test cases cover all lines of code or not

*Methodology used*

The developer solely performed unit testing in the development environment. The developers evaluated their code to ensure that the units under test behaved as intended.

*Tools used*

- Jest (Third Party Libraries)

*Testcases*

<b>Test Id</b>	<b>Test Case Name</b>	<b>Test Case Description</b>
1.	Login with correct credentials only.	With correct username and password, the user should be easily logged in.
2.	Error on invalid username and password	While retrieving username and password, if it doesn't match to the existing data values then, an Error should be displayed
3.	Connect user and site admin via Mail.	User should be able to send mails to the site admin.
4.	Add/Update project/clients' information.	While retrieving information, new information, if added, and existing information, if updated, should reflect in the database.

5.	Delete project/client's information.	While retrieving information, if existing information, if deleted, the subsequent changes should reflect in the database.
----	--------------------------------------	---

*Table 4.2: Unit Testcases Table*

### ***Integration Testing***

- Individual units are integrated and tested collectively

#### *Methodology used*

The bottom up technique was utilised for integration testing, in which the bottom level modules are tested first, followed by test results of higher tier modules. The approach was continued until the top-level component of the hierarchy was tested. Individual units are integrated and tested collectively.

#### *Tools used*

Not applicable (Manual testing was carried out)

### ***Non-functional Testing***

- To examine a software application's non-functional features (performance, usability, reliability, and so on)

#### *Methodology used*

Browser compatibility, performance of the system and daily usability was tested

#### *Tools used*

- Web Browser Dev Tool (Network performance, lighthouse)

## **4.4 Test Execution**

The tests have been run on individual components as part of unit testing. Each functionality input was classified as valid or invalid, and a composite of which was evaluated on the relevant functional modules. The outcomes of such inputs are detailed in the testcase results section. Each need was analysed as well as its implementation with relevant functionality was supplied.

## 4.5 Results

The outcome was quite decent; it met the majority of the customer criteria; the user is simple to browse; and even inexperienced users can easily navigate the web application. The validation on the client side strong. There is some problems with Google authentication during login which is a key flaw. System is able to generate structured reports bases on specific data that HR cares about.

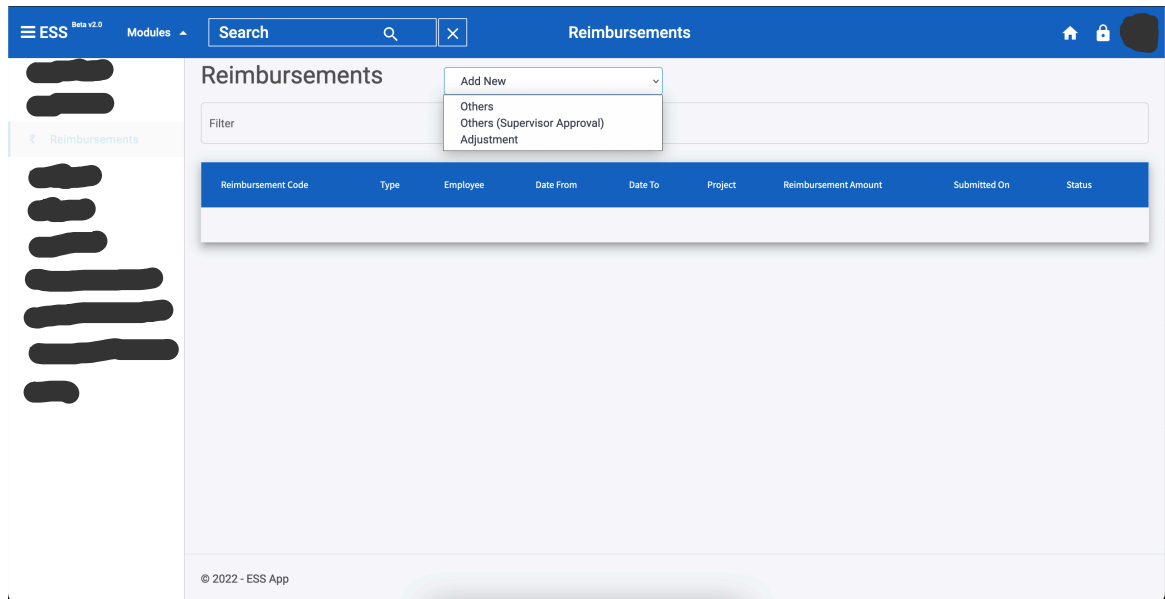


Figure 6: ESS Portal - Reimbursements Module

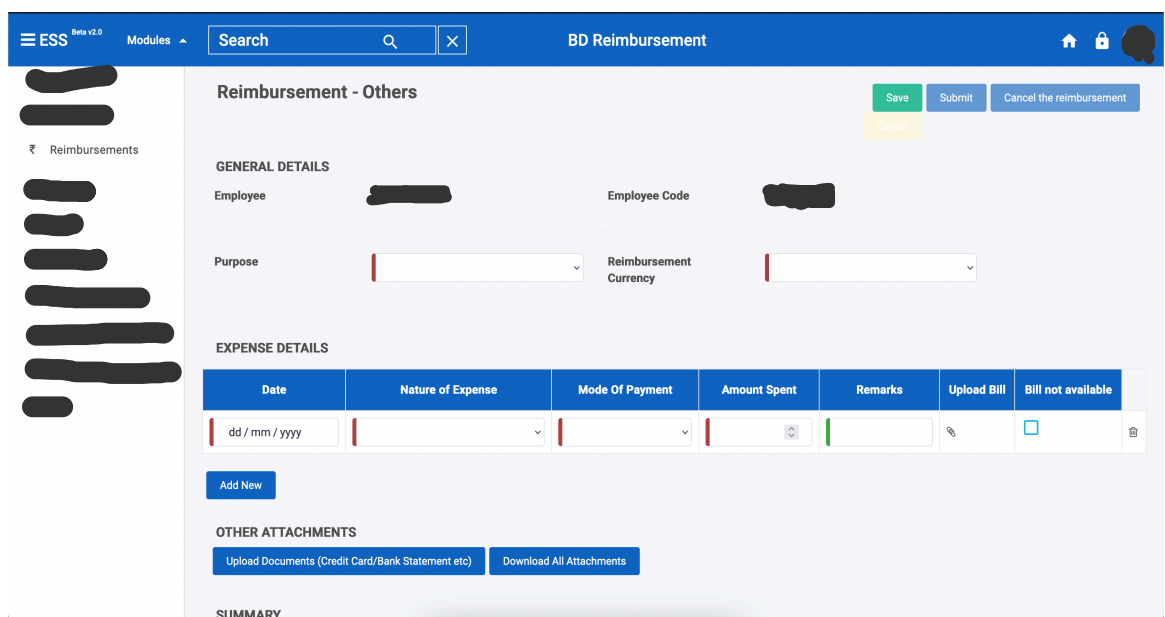
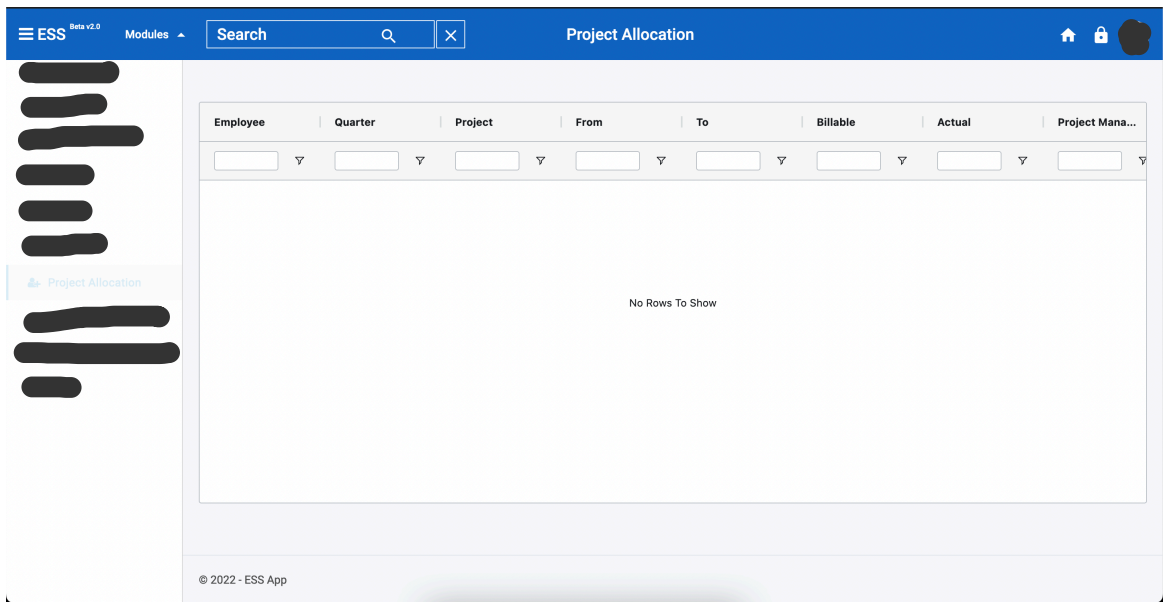


Figure 7: ESS Portal – Reimbursement Module after selecting option



*Figure 8: Project Management Module in Employee Side*

## **5. Chapter–5 Conclusions**

### **5.1 Conclusions**

- This project aims to develop an automated system which will provide employees a functionality to manage company resources
- The employees can create timesheet and update about their assigned tasks and keep themselves and supervisors updated about the employee status and project as well.

## References

- [1] J. Bulmash, "Human Resource Management and Technology", Chapter 3.
- [2] "TECH HRM (Human Resource Management System)," 4 11 2013. [Online]. Available: <http://www.techjetsolutions.com/brochure/TECHHRM.pdf>.
- [3] R. Broderick and J. Boudreau, "Human Resource Management, Information Technology, and the Competitive Edge," *CAHRS Working Paper Series*, vol. 6, 1992.
- [4] "Centralized Employee Information," Zoho, 08 07 2014. [Online]. Available: <https://www.zoho.com/people/employee-management-system.html>.
- [5] I. Sommerville, "Software Engineering", 9th Edition, Addison-Wesley, 2011.
- [6] J. Campbell, "Scrum Methodology: Breaking Down the Scrum Framework," [scrumexplainer.com](http://scrumexplainer.com), 05 06 2015. [Online]. Available: <https://scrumexplainer.com/scrum/scrum-methodology/#:~:text=The%20Scrum%20methodology%20is%20an%20agile%20framework%20that,should%20do%20the%20tasks%20but%20not%20in%20detail..>

Kyle Simpson. 2014. *You Don't Know JS: Scope & Closures* (1st. ed.). O'Reilly Media, Inc.

<https://reactjs.org/docs/getting-started.html>

<https://jestjs.io/>

Alex Banks and Eve Porcello. 2017. *Learning React: Functional Web Development with React and Redux* (1st. ed.). O'Reilly Media, Inc.

# Appendices

## Appendices–I Architecture Diagram

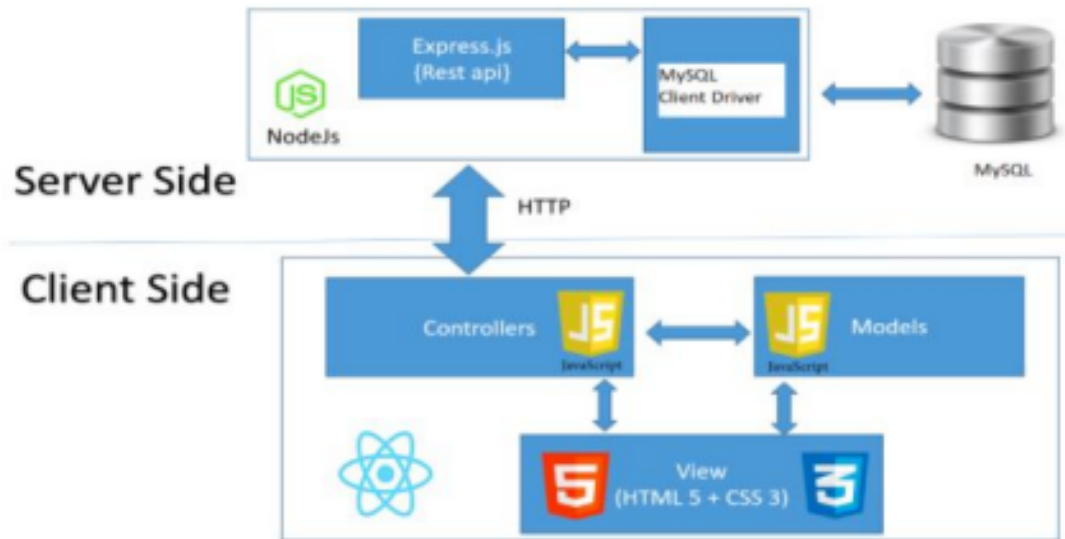


Figure 1: Architecture Diagram

## Appendices–II Flow Diagrams

### Reimbursement Module

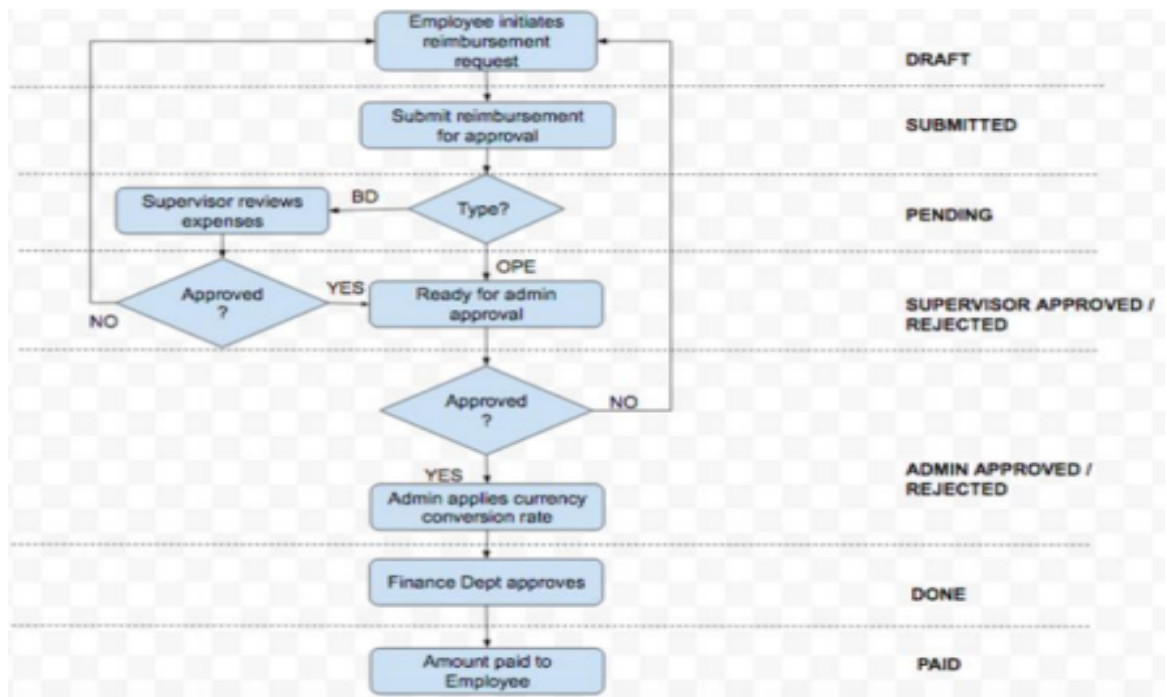


Figure 2: Reimbursement Module Flow Diagram



## Project Module

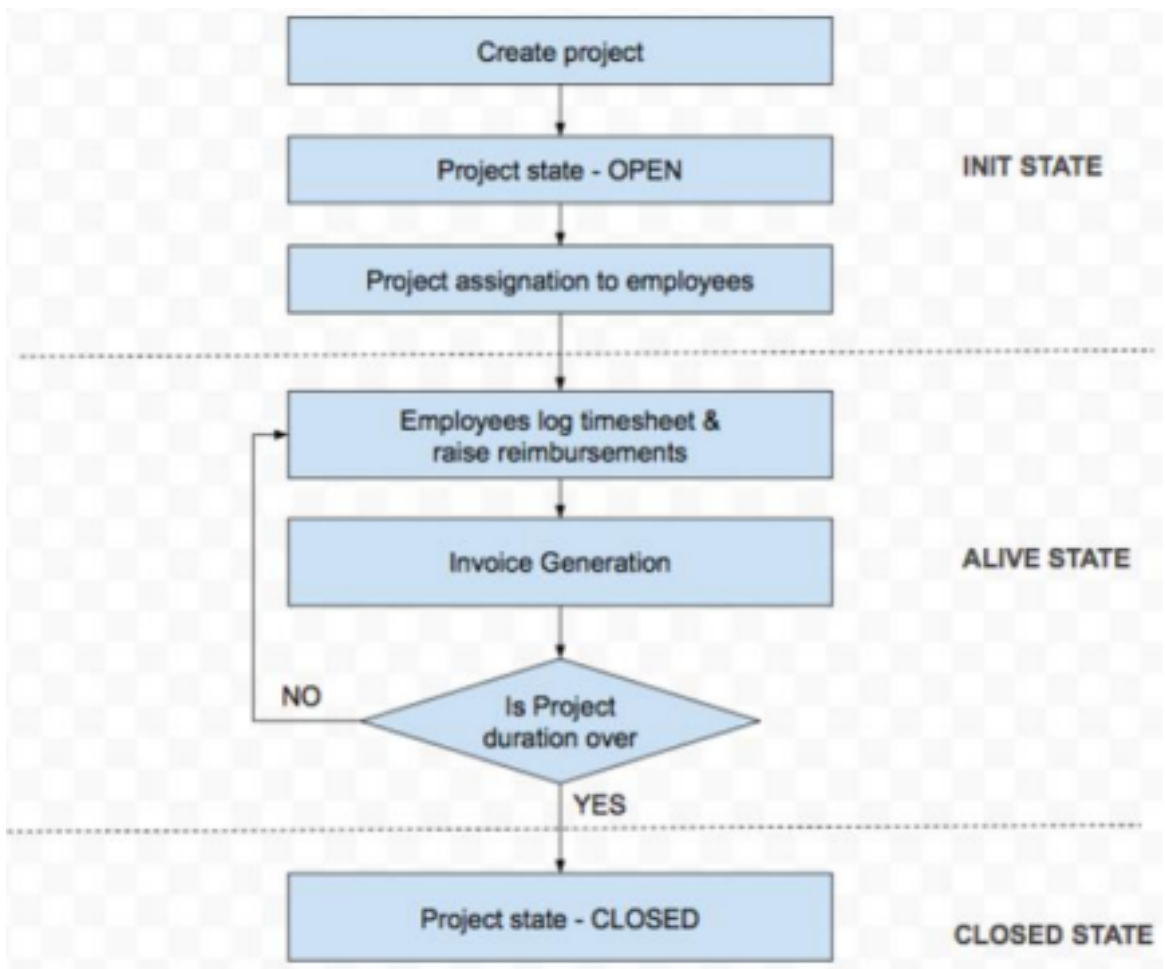


Figure 3: Project Module Flow

## Appendices–III Context Diagram



Figure 4: Level 0 DFD

## Appendices–III DFD Level 1

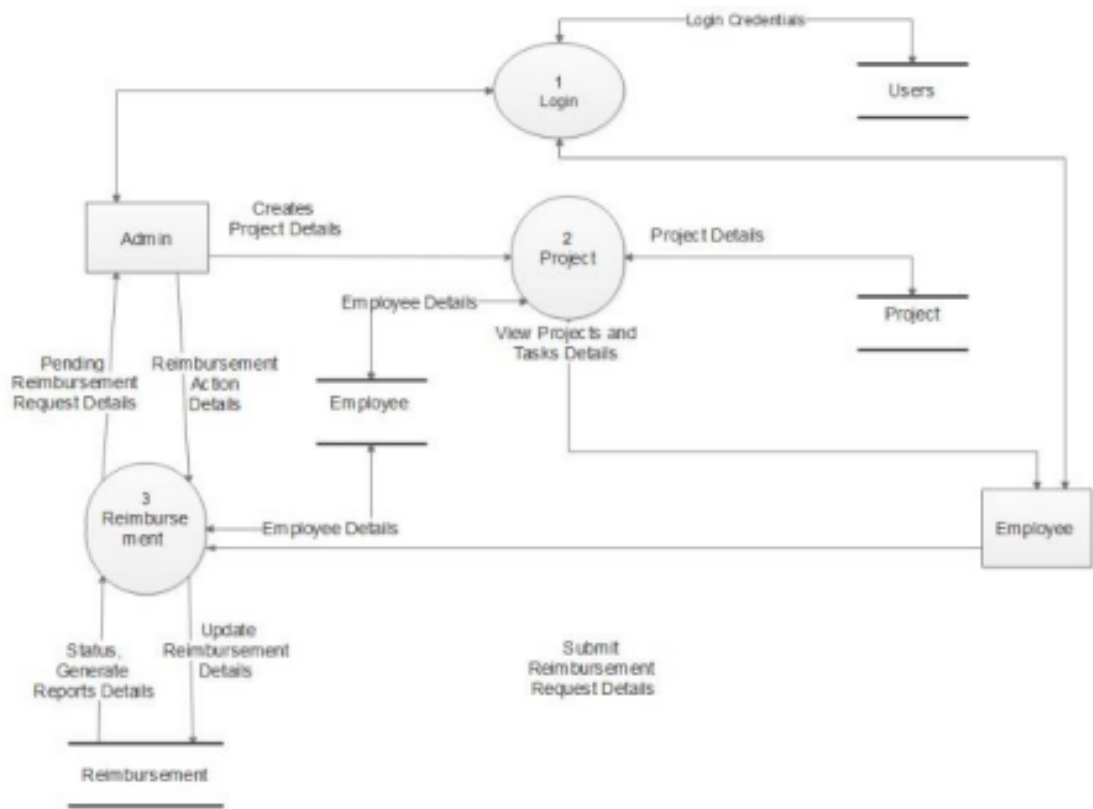


Figure 5: Level 1 DFD