Office Management and HR Portal

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

ROHIT RANJAN(171381)

Under the supervision of
(MR. Aman Chadha)

To

Department of Computer Science & Engineering
Jaypee University of Information Technology Waknaghat, Solan-173234
Himachal Pradesh

DECLARATION BY CANDIDATE

I hereby declare that the work which presented in this report entitled **“Office Management and HR Portal”** , in partial fulfillment of the requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the Department Computer Science & Engineering and Information Technology, Jaypee University of Information Technology waknaghat, is an authentic record of our own work at Vectoscalat Technology Pvt. Ltd. carried out over a period from Feb. 2021 to May 2021.

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

Rohit Ranjan

Name: Rohit Ranjan
Roll No:171381
Branch: CSE

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

Aman Chadha

(MR. Aman Chadha)
Technical Lead
Vectoscalar Technology Pvt. Ltd.
Dated:- 25th May 2021

Acknowledgement

No volume of words is enough to express our gratitude towards my guide, Mr Aman Chadha, Technical Lead, Vectoscalar Technology Pvt. Ltd., Noida, who has been very concerned and has aided for all the materials essential for the preparation of this major project report. She has helped us to explore this vast field in an organized manner and provided us with all the ideas on how to work towards a research oriented venture.

I am also thankful to prof. (Dr.) Samir Dev Gupta, Head of Computer Science and Engineering Department and Mr. Ravi Raina, for the motivation and inspiration that triggered us for the major project work.

Most importantly, we would like to thank my parents, friends and the almighty for showing us the right direction out of the blue, to help us stay calm in the oddest of the times and keep moving even at times when there was no hope.

Rohit Ranjan
9354507962

Table of Contents

CHAPTER 4: TESTING

CHAPTER 5: CONCLUSION AND REFERENCES

CHAPTER 6: ANNEXURES

LIST OF FIGURES

LIST OF TABLES

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

Introduction about the company

Problem Statement

Proposed Solution

My Role

Deliverables

Introduction about the Company

VectoScalar Technologies Pvt. Ltd. started it's eventful operation in software/ and application development in 2010. Located in Noida, India we have considered the increasing enterprise Software-business-needs, in a time and cost effective manner. We covers every stage of the application-development-lifecycle: from the business analysis, the design, and the prototyping to the real development, the quality assurance, and the post-project maintenance and the support. By going through the continuously evolving process, we have ensured good quality work result and long lasting relationships with our Client. We create native, hybrid, and cross platform applications which run on all the major operating system such as iOS and the Android as well as the Web and the Desktop application.

Industry Expertise

AWS Service:

- Expertise in web hosting and App Deployment. Use depth skill to turn ideas into a Creative User Experience.

iOS application:

- A great experience in creating high class and beautiful iOS application for both iPhones and the iPad.

Web development:

- Innovative and great web solution to develop unparalleled good performance. Create some good transformative Web experience.

Android application:

- Creating engaging mobile experience, while keeping in mind of everything from the design to the development for your idea.

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 application:

- Depth skill, robust processes and flexible to create paramount hybrid solution for IONIC and METEOR mobile application.

Desktop Application:

- We give cutting edge solution for Desktop and Native application for all of the operating system.

Beacon:

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

Problem Statement

No Company wants to waste its time and resources on daily menial tasks of organization that has nothing to do with organizational 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 on a daily basis, lock employee’s work hours, attendance, reimbursements in order to determine the employees performance on a particular project and to get realistic billable amount from the client.

Proposed Solution

We need to develop a tool which helps to manage company’s employees work details along with effort spend per day at one place (in-line) for multiple projects and also helps supervisors to monitor supervisees task status and time spent on task.

This application has mainly following 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:

- Login to the system using their "User Name" and "Password".
- 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. Their additional right in the application along with employee rights are:

- Login to the system using their admin "Username" and the "Password".
- Add new employees or edit existing employees.
- Admin should be able to approve or reject the employee's timesheet and reimbursements.
- Admin generate invoice based on 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 particular modules that are accessible to him.
- Admin is able to 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 apply for their leaves.
- 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.

My Role

My role is to develop and unit 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:

States	Description
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 1.1: Reimbursement States

Nature of Expenses:

Types	Description
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 1.2: Nature of Expenses

Project Management:

- This module manages the information about the client-projects.
- Admin will have the privilege 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:

States	Description
Open	Admin has privileges to change the status of projects
Closed	Admin has privileges to change the status of projects

Table 1.3: Project States

Project Types:

- Project must be in one of the following types:

Types	Description
Retainer ship	A fixed monthly amount paid irrespective of man days
Fixed	A fixed monthly amount paid based on no. of man days
Timesheet	Depending on no. of hours
Retainer ship + Timesheet	Combination of Retainer ship & Timesheet

Table 1.4: Project Types

Access Control via Roles:

Types	Privileges
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 1.5: Access Control via Roles

Deliverables

S No.	Phase	Deliverables	Page No.
1	Requirements Analysis	Use Case, Flow Diagram	
2	Design	ERD	
3	Implementation	DFD	
4	Unit Testing	Test Results	

Table 1.6: Deliverables

CHAPTER 2

PROJECT DESCRIPTION

System Interfaces

The overall System has the following Interfaces:

Login Page

- This is the interface which permits the users to enter into the system.
- It provides authentication based on form.
- The authentication as well as the authorization processes are performed by it.

Home Page

- This is the next page after the user is logged 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.

System Specification

H/W Requirements

- Architecture X86 or X86-64 bit hardware architecture
- Processing Power Core 2 Duo 2.4-gigahertz (GHz) processor or faster
- Memory 1 gigabyte (GB) of RAM (4 GB is recommended)

S/W Requirements

- Library
  - React JS
- Run-Time Environment
  - Node.js
- Database
  - MySQL

Methodology and Tools Used

Methodology Used

The concerned Project uses the Agile-methodology of iterative development in which the requirement and the solution evolves through collaborating with the client. Agile Development is a development model for the development of the desktop application. It is more powerful and efficient within a short period of time than the other model and it incorporate front to front communication, & it contains customer and technical personnel both as a component of the team. The agile desktop development uses the business analyst, the project manager, forces on the planning, iterative delivery and the clear goal. The agile development ensure the sucessful fulfillment of the products at the end of each delivery.

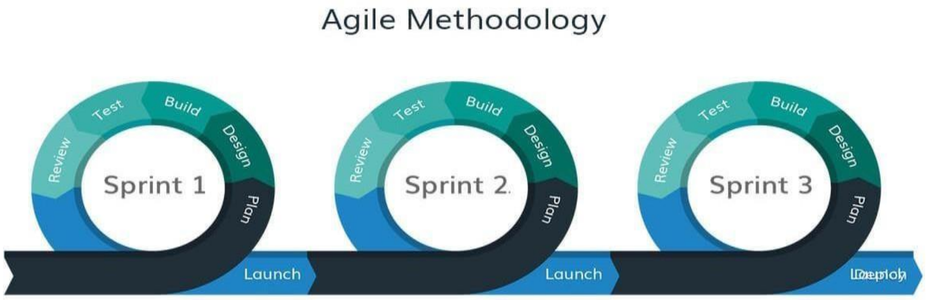

Figure 2.1: AGILE Methodology

Architecture Used

3 Tier Architecture used with defined Layers

The layered architecture focus on the grouping of similar functionality in an application in the distinct layer those are stacked vertically above of each other. Communication among the layer are explicitly & loosely coupled.

Layering the application appropriately help to support a strong separation of concern which, in return, support the flexibility and maintainability. At the most abstract & highest level, the logical architecture view of the System could be considered as a set of co-operating component which is grouped in the layer.

- **Presentation layer**

Presentation Layer (UI) is the top most layer of application where user performs their activity. Basically user's input validation and rule processing performs in this layer. UI layer interface with the controller class in the layered architecture of the application.

- **Business-Logical-Layer**

Business Logic layer implements the main functionality of system, & encapsulates the relevant business logic and the validations that have to be applied on the input data. It comprises of component, some of which might expose the service interface which other caller can use. It controls an application's functionality by performing detailed processing.

- **Data-access-layer**

Data Access layer gives permission to information hosted in the boundary of the Systems, & the information exposed by the other networked system; perhaps accessed through the service. The data layer exposes general interface which are parts in the business layer could consume.

The Requirement-phase

1. Use-case-model: The use-case-model enables us in exploring that in which way the users will work with our systems. The use-cases has been explained in various type of scenarios of work into these models.

2. Initial-domain-model: The model which identify the basic business-entity-type & the relationship among them. A general E.R diagram has also been defined which shows relation of an Entity with others.

3. **Website-structure-model:** The website-structure-diagram had been explained as the diagram that show the complete structure of application & list all screen which could be access by the users.

The requirement has been defined for the first deliverable that includes a System that can get users response on various policies. The requirement has been categorized under below functional, usability, and reliability requirement.

The discovery phase included these steps:

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

- Set the project scope and schedule

Functional-Requirements

- All the users should have must login with their username and password.
- Role based Security needed.
- User can view and manage his 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

- System should save the state of the application persistently.

Design-phase

The Design-model is followed by a testing based designing philosophy. Under the model, the unit-testing will be happened along with the designing & the progress of functional unit in parallel. The very stating step is to quickly add a testing, mainly little more for code to not succeed. Next new tests are run to make sure their failure. The functional code is then updated to make it pass the new test. The 4th phase is to run tests again. If the test fail the functional code is updated and rechecked. Once the test pass the next step is to start over with the new testing.

The Design-phase commenced when the requirement will be completed & the fixed. The Design-phase tries to uncover the different entity which are involve into the system & the associating behavior & the interfaces which will be allotted by the system. Data-Flow-Diagrams for the System will be developed.

Development Phase

Model-**V**iew-**C**ontroller or MVC as it is called, is a software-design-pattern for creating web application. MVC patterns are created by the following 3 components:

- The model – It is the most lower level of this pattern that are dependable for the maintainance of data.
- The View − It is dependable to display every or part of the data to the user.
- Controller – It is the software code which is used to control the interaction among the model & the view.

Model View Controller is as famous as it isolate the applications logics from the users interface layer & support the concerns of separatism. The controller will receive each of the request for application and then work with the model to develop the information which is required by the view. The view uses the information which is developed by the controller to create the last presentable response. The Model View Controller abstraction could be graphically represented as following:

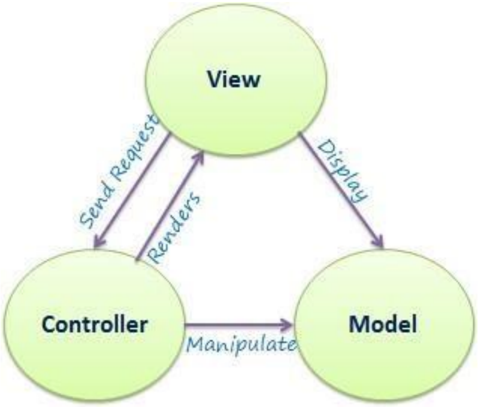

Figure 2.2 : MVC

The Model

The model is dependable for the management of data of the application. This response to request from the view & also response to the instruction from the controller to update themself.

The view

This mean presenting the data in a special design, which is triggered by a controller decision to show the information. These are based on screen template system and easy to be integrated with the AJAX technology.

The controller

It is responsible for the response to the user input and it performs the interaction on the data model object. The controller gets the input, validate the input and then perform the business operations which modifies the state of the data model.

Implementation Phase

The System is implemented on a local server with tomcat component installed. User interacts through a web browser that send the HTTP request on the server which will reply back just when the requested page will be processed. Hence, the system will be implemented as the client-side server model. To get the information, server get connected to the server that will reply with requested data sets.

Testing Phase

Bottom up testing was performed on deliverable of project starting from Business Access Layer.

Unit-testing

The unit-testing at data access layer were happened on the procedure before getting added into the context to test the direct working on database. At business logic layer, the unit-testing were tested on the various businesses & edited the objects using drivers at the Interface layer to check retrieving & accuracy of the information at the layer of data access.

Integration-testing

The integration testing within the deliverable involves integration of all of the layer & to test the secure transmission of data under the explained role. The integrity check was checked on all the information transferred to the layer data access from the user interface layer.

System-testing

The system testing were happened under the environment of deployment to check for acceptance under environment.

User Characteristics

Under Role Based Security four types of Users have been defined: The Admin user, supervisor user, Employee user, Finance user. Under Role Based Security four types of Users have been defined:

Types	**Privileges**
Employee	Employee can fill timesheet, raise reimbursement, etc.
Supervisor	First level approver. Supervisor can approve timesheet & reimbursement
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

Table 2.1: User Characteristics

CHAPTER 3

FUNCTIONALITY

Logical Database Design

3.1.1ERD

3.1.2 Table Structures

Use Case Description

User Login

Initiate Reimbursement

Approve or Reject Reimbursement Request

View Projects

Project Management

Logical Database Design

ERD

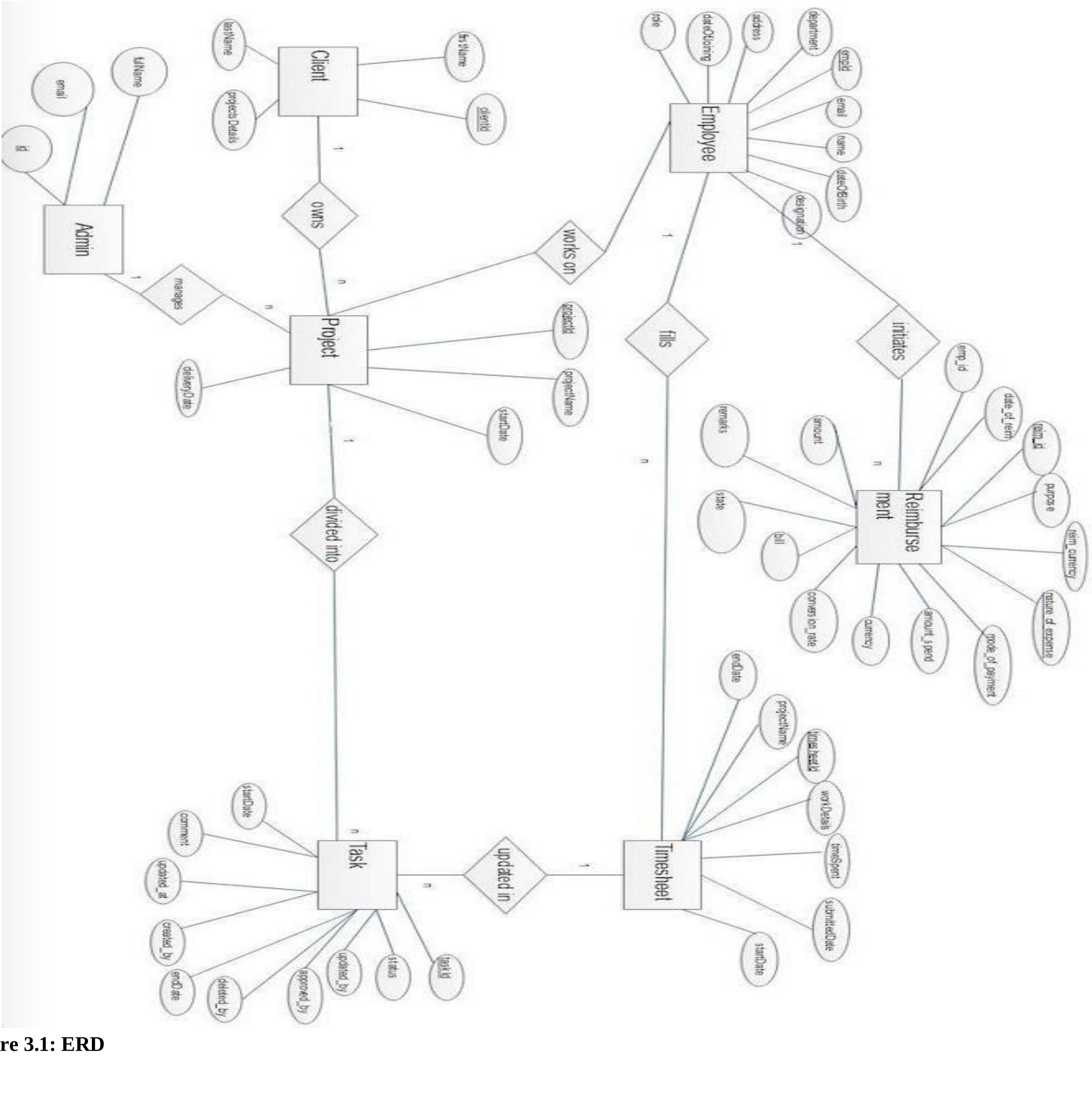

Figure 3.1: ERD

Table Structures

- **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-id
Contact Mobile#	Textbox	Mobile No.	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**

Field	Control	Description	Constraint
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	No. of Employee	Mandatory
No. of Employees	Textbox	Total No. 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 agreement	Optional

Table 3.2: Project Table

- **Timesheet Table:**

Field	Control	Description	Constraint
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:**

Field	Control	Description	Constraint
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

Field	Control	Description	Constraint
Full Name	Textbox	Full Name	Mandatory
Email	Textbox	Email	Mandatory

Table 3.5: Admin Table

- Tasks Table

Field	Control	Description	Constraint
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

Field	Control	Description	Constraint
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

User Login

use case name	login
purpose	to login into the system.
actors	admin, employee
preconditions	User must have user name and password to login into the system.
post conditions	After login, user redirects to the dashboard.
Basic Flow	▪ user enter a user name and password to login into the system. ▪ Password validate, if username and password is valid, user is successfully logged- in the system. ▪ If username and password is not valid, user can’t be able to logged-in to the system. ▪ User redirects to the dashboard page.
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 3.5: User Login Use Case

Initiate Reimbursement

Use Case Name	Initiate Reimbursement
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	▪ User clicks on initiate reimbursement button. ▪ Selects project, enters reimbursement amount. ▪ Save and submit reimbursement request
Alternate Flows	N/A

Table 3.6: Initiate Reimbursement Use Case

Approve or Reject Reimbursement Request

Use Case Name	Approve or Reject Reimbursement
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	▪ User submits a reimbursement request for approval. ▪ Supervisor views pending reimbursement. ▪ Supervisor can either approve or reject reimbursement. ▪ Reimbursement status will bechanged. ▪ If reimbursement rejects, employee need to edit reimbursement and again send for approval.
Alternate Flows	Admin can also reject reimbursement.

Table 3.7: Approve/Reject Reimbursement Use Case

View Projects

Use Case Name	View Projects
Purpose	To View Ongoing Projects assigned to an Employee
Actors	Employee
Preconditions	User must be logged in.
Post Conditions	N/A
Basic Flow	▪ User clicks on view projects ▪ User can then view projects assigned to him/her.
Alternate Flows	N/A

Table 3.8: View Projects Use Case

Project Management

Use Case Name	Project Management
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	▪ Admin clicks on create project ▪ Admin can then assign projects to employees.
Alternate Flows	▪ Admin clicks on manage project ▪ Admin can then assign projects to employees.

Table 3.9: Project Management Use Case

CHAPTER 4

TESTING

- Testing Activities
- Test Environment
- Scope of Testing
    - Modules to Be Tested
    - Types of Testing
    - Test Cases

Implementation

Testing Activity

- **Project Initiation**

Testing targets will be set during this phase. From the testing point of view, the major task was to organize the method towards testing & start setting up the testing environment if it doesn't exist already.

- **Development team testing**

The entire team strategy was followed where people with testing skill were mainly embedded in the team of development & the team was responsible for the large part of the testing. This strategy will work well.

- **End-of-lifecycle testing**

The important part in the release effort were end of lifecycle testing in which an independent testing team validates whether the system is ready to start production.

Test Environment

Hardware	Software	Supporting Tools
Windows, Mac	Chrome, Safari	Postman, Jest

Table 4.1: Test Environment

Scope of testing

Modules to be tested

S.No.	Modules
1	Reimbursement Module
2	Project Management Module

Table 4.2: Modules to be tested

Types of Testing

Testing performed at module/project level

Unit Testing

Tests individual units / components of a software.

- The unit testing is the procedure during which individual parts of a program are tested to determine whether they work as expected.
- 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.

Used Methodology

The unit testing was performed by the development engineer in the development environment only. The developer will test the code that their respective unit under the test is behaving as expected or not.

Tool used: Jest (Third Party Libraries)

Test Case

Test id	Test case name	Test case description
1.	Login with the right credential only.	With the correct username & password, the user will be easily logged in.
2.	Error on wrong 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 4.3: Unit Test Cases

Integration Testing

- Individual units are combined and tested as a group.

Methodology Used

For use integration test bottom-up method was used in which the most lower level components were tested first, and then it was used to facilitate the testing of higher level component. The processes were repeated regularly until the components at the most top of the hierarchy was tested. Individual unit was combined and tested as a group

Tools Used

Not Applicable.

Functional Testing

- System is tested against the functional requirements/specifications.

Methodology used

The entire system is checked in the presence of the developer team under this. Mainly, entire of the functionality were tested here according to the requirement.

Used Tools

Testing were carried out manually.

Non-Functional testing

- To test the non-functional points (usability, performance, reliability, etc.) of a software application.

Methodology Used

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

Tools Used

Not applicable.

Manual-testing were carried out.

Implementations

All the test was performed on personal unit on first time under the unit testing. All of the Functionality input was separated as the true or false input and then the mixture of all of these were checked on the related functional module. The Result of the input were explained in the testcase result. Under the system testing, the whole functionalities of the systems according to the requirement were checked. All of the requirements were analyzed & it’s implementations with the related functionalities were allotted.

CHAPTER : 05

Conclusion & References

Conclusion:

References/ Bibliography

Conclusion

- 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 status as well.

References:

Books

- Learning React: Functional Web Development with React and Redux
- JavaScript: The Definitive Guide
- You Don't Know JS: ES6 & Beyond

Online Resources

- Mozilla JS Documentation: - https://developer.mozilla.org/bm/docs/Web/JavaScript
- React JS Official Documentation: - https://reactjs.org/docs/getting-started.html
- Jest Documentation: - https://jestjs.io/docs/en/mock-functions
- Using enzyme with Jest: - https://enzymejs.github.io/enzyme/docs/guides/jest.html

CHAPTER 6

ANNEXURES

A-1 Architecture Diagram

A-2 Use case Diagram

Flow Diagrams

Context Diagram (DFD Level 0) A-5

DFD Level 1

Architecture Diagram:

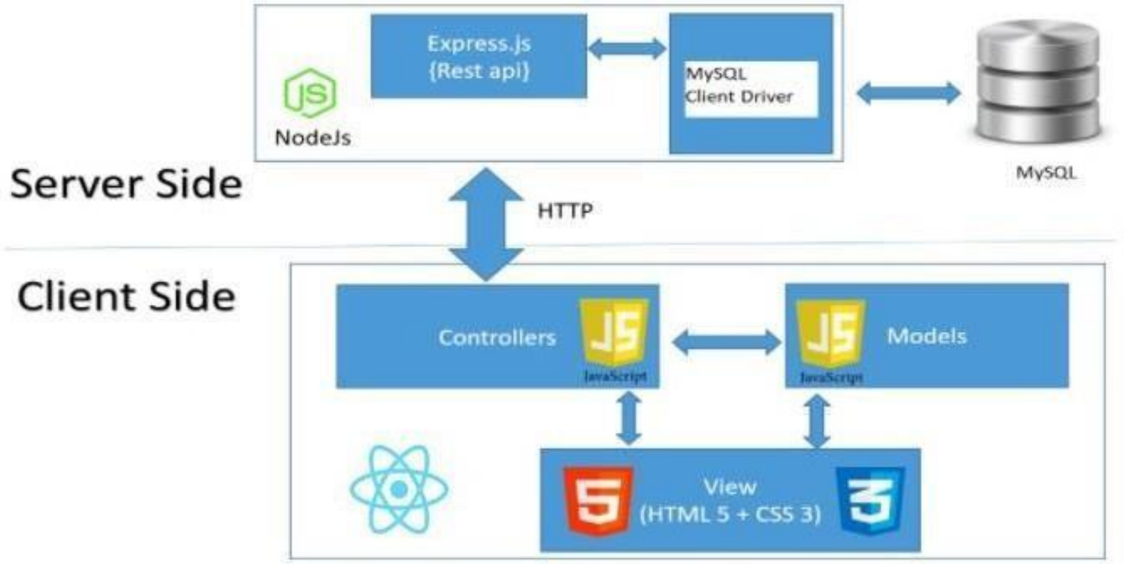

Figure 6.1 Architecture Diagram

USE CASE Diagram:

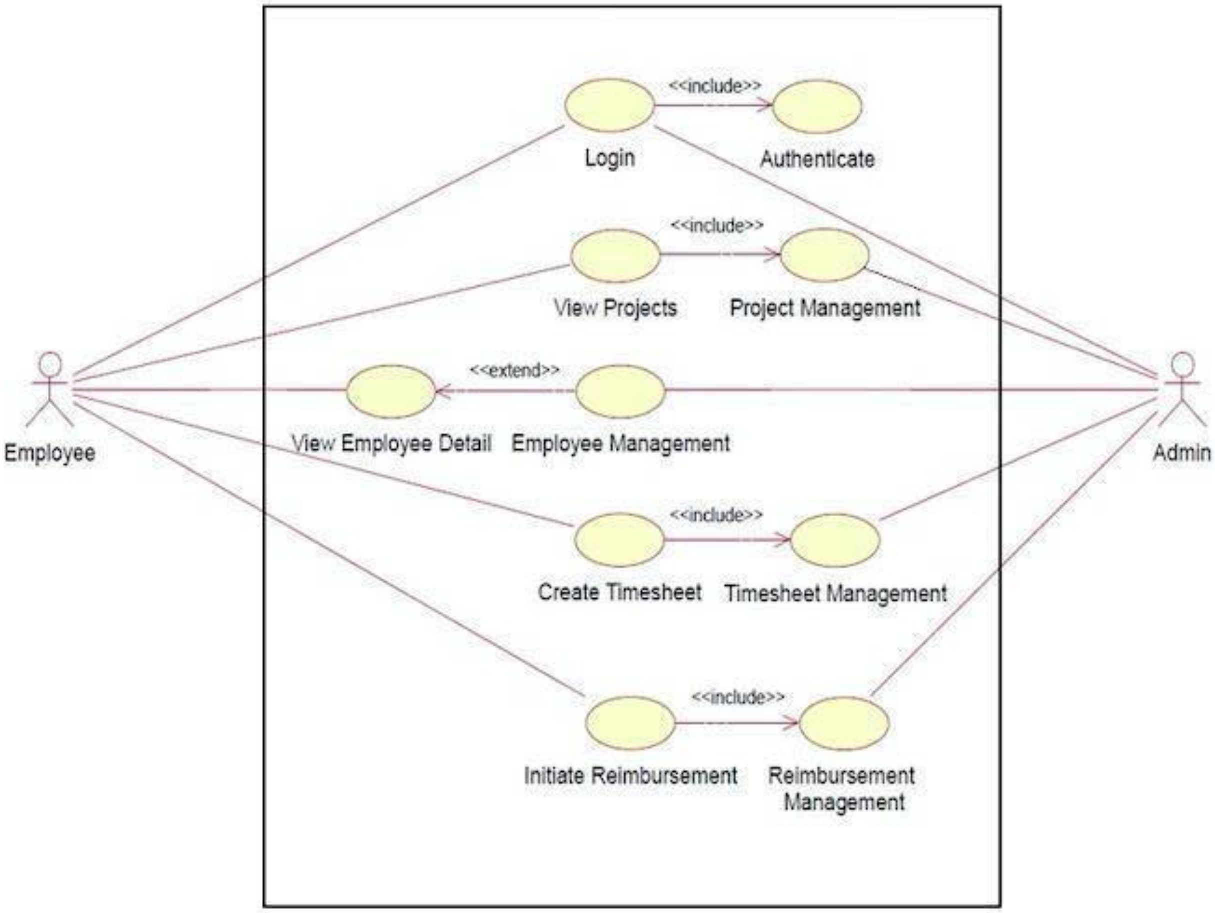

Figure 6.2 Use Case Diagram

Flow Diagrams

- **Reimbursement Module**

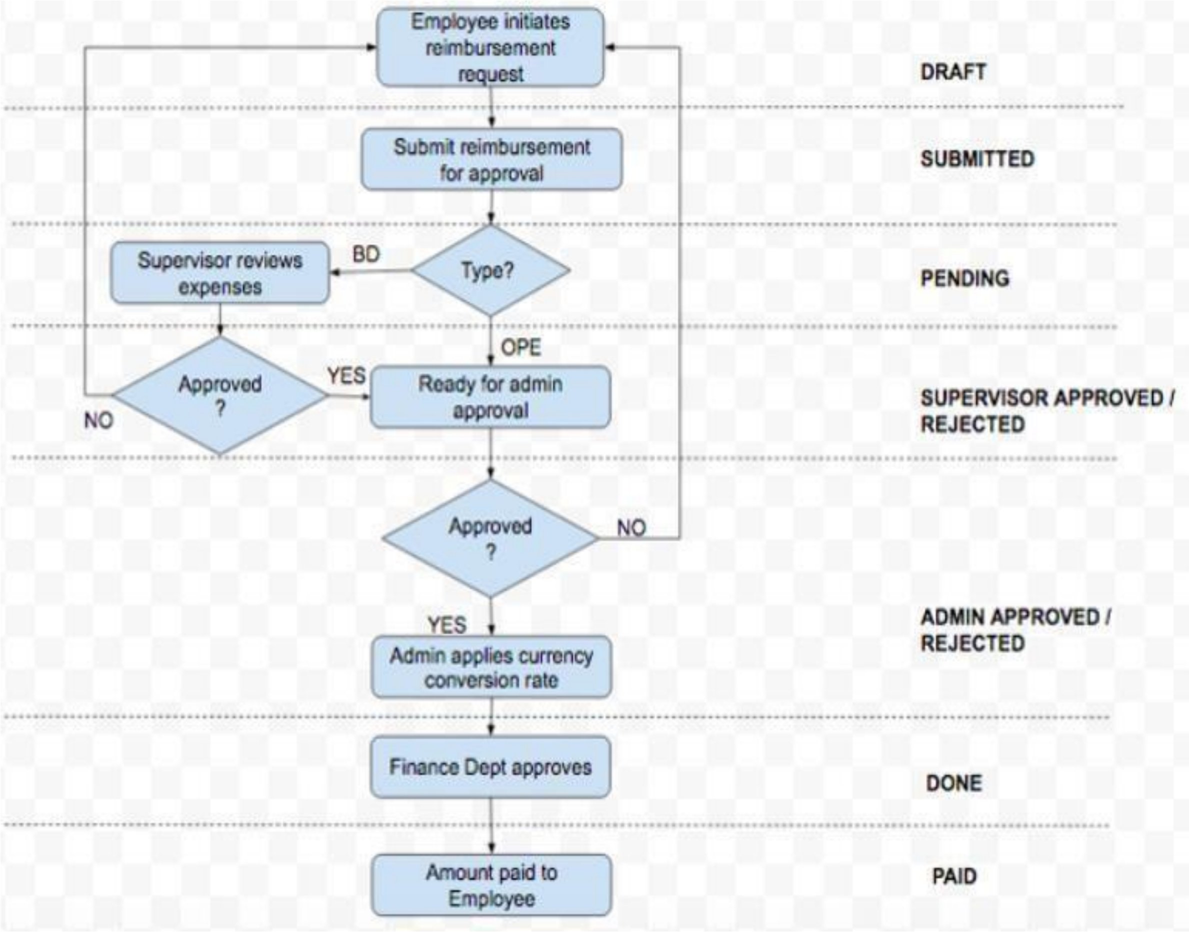
Employee initiates reimbursement request
DRAFT
Submit reimbursement for approval
SUBMITTED
Supervisor reviews expenses
BD
Type?
PENDING
OPE
Approved ?
YES
NO
Ready for admin approval
SUPERVISOR APPROVED / REJECTED
Approved ?
NO
YES
Admin applies currency conversion rate
ADMIN APPROVED / REJECTED
Finance Dept approves
DONE
Amount paid to Employee
PAID

Figure 6.3 Flow Diagram for Reimbursement Module

- **Project Module**

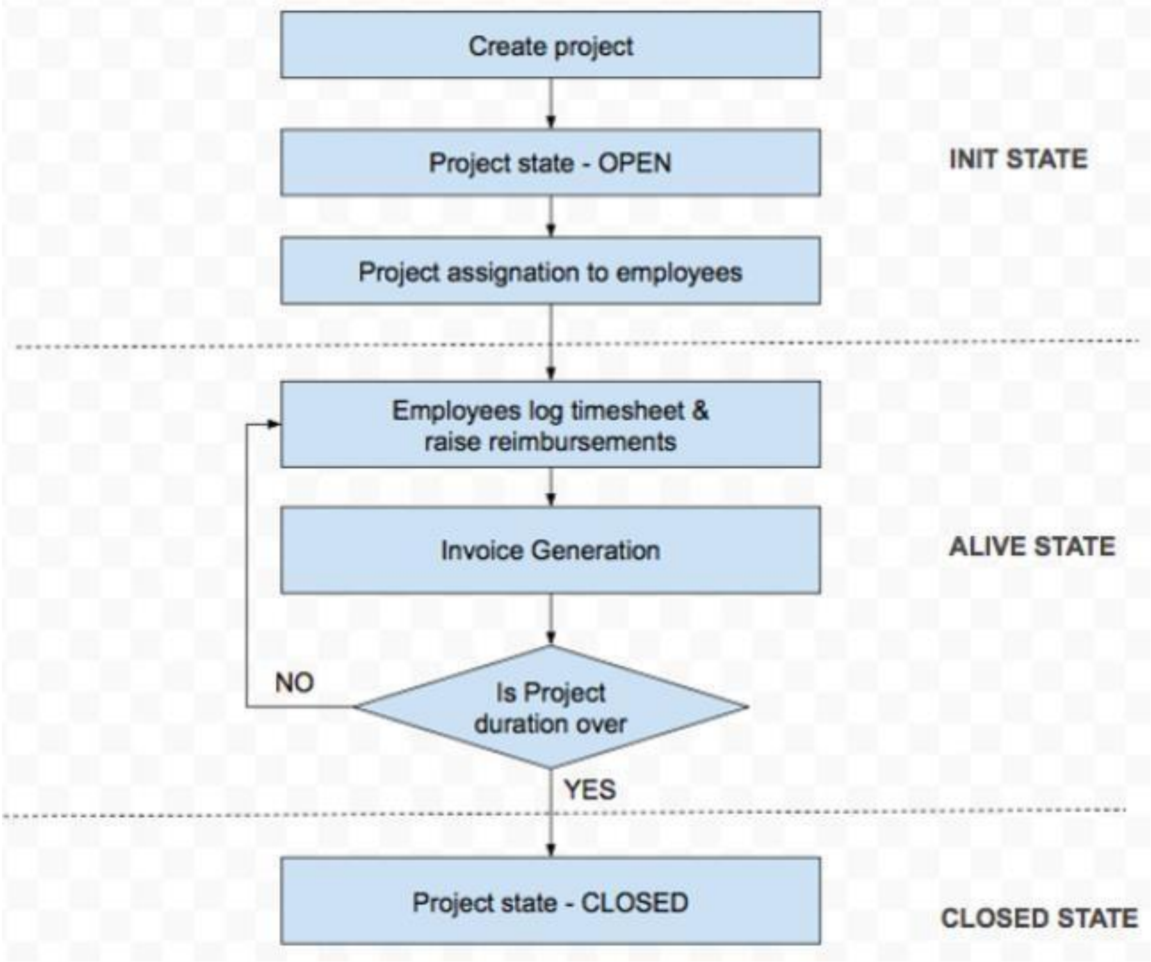

Figure 6.4 Flow Diagram for Project Module

Context Diagram (DFD Level 0)

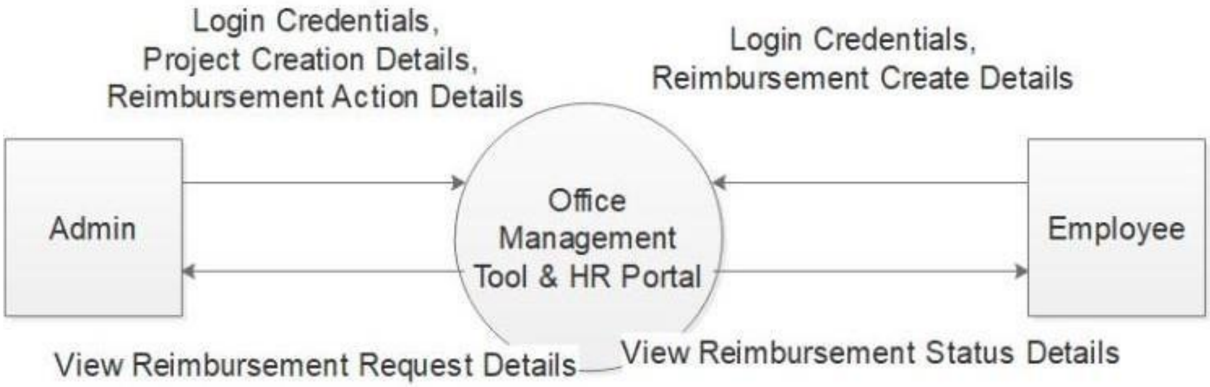
Login Credentials,
Project Creation Details,
Reimbursement Action Details
Login Credentials,
Reimbursement Create Details
Admin
Office Management Tool & HR Portal
Employee
View Reimbursement Request Details
View Reimbursement Status Details

Figure 6.5 Context level Diagram

DFD Level 1

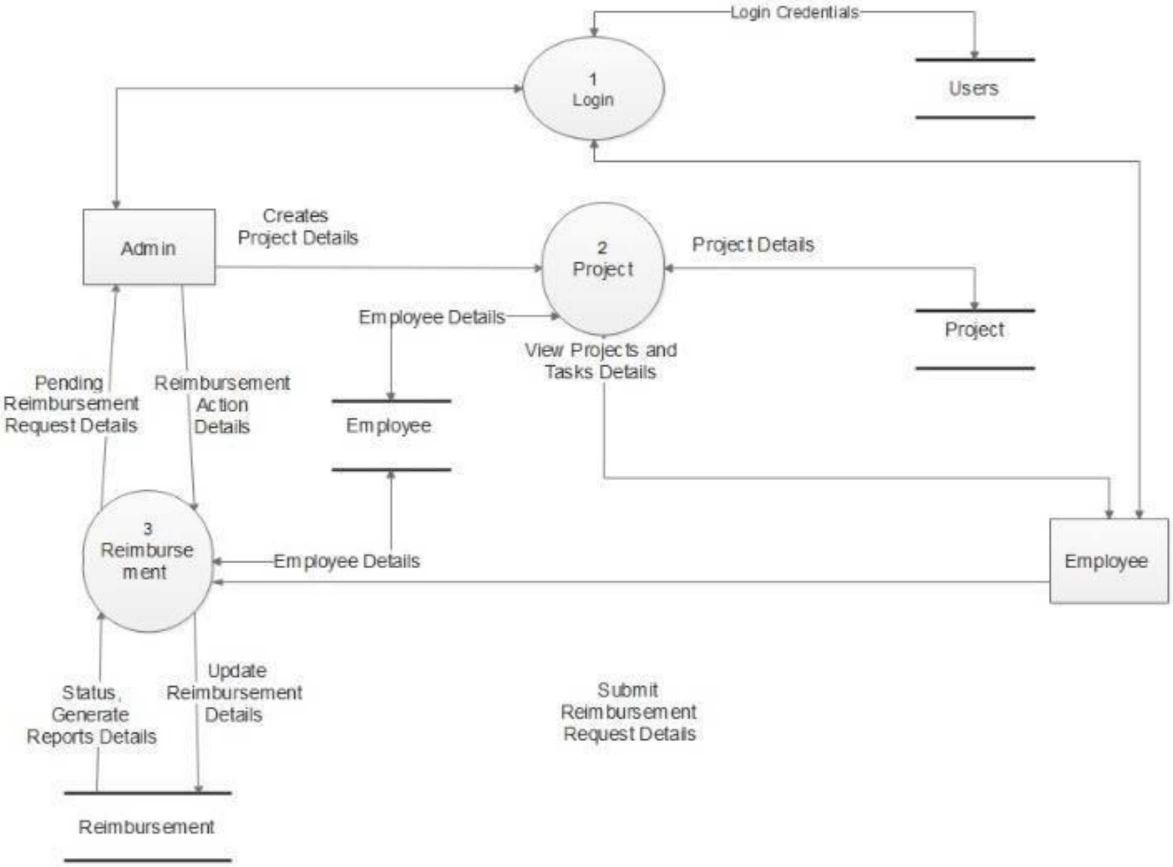

Figure 6.6 DFD Level 1