

Payroll Management System

A Project-II Report

Submitted in partial fulfillment of requirement of the

Degree of

BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE & ENGINEERING

BY

Mukul Hiranandani

EN18CS301152

Under the Guidance of

Dr. Kailash Bandhu



Department of Computer Science & Engineering

Faculty of Engineering

MEDI-CAPS UNIVERSITY, INDORE- 453331

Jan-June 2022

Payroll Management System

A Project-II Report
Submitted in partial fulfillment of requirement of the
Degree of
**BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE &
ENGINEERING**

BY
Mukul Hiranandani
EN18CS301152

Under the Guidance of
Dr. Kailash Bandhu



Department of Computer Science & Engineering
Faculty of Engineering
MEDI-CAPS UNIVERSITY, INDORE- 453331

Jan- June 2022

Report Approval

The project work “**Payroll Management System**” is hereby approved as a creditable study of an engineering/computer application subject carried out and presented in a manner satisfactory to warrant its acceptance as prerequisite for the Degree for which it has been submitted.

It is to be understood that by this approval the undersigned do not endorse or approved any statement made, opinion expressed, or conclusion drawn there in; but approve the “Project Report” only for the purpose for which it has been submitted.

Internal Examiner

Name:

Designation

Affiliation

External Examiner

Name:

Designation

Affiliation

Declaration

I/We hereby declare that the project entitled “**Payroll Management System**” submitted in partial fulfillment for the award of the degree of Bachelor of Technology in ‘computer science’ completed under the supervision of **Dr. Kailash Bandhu, Professor, Computer Science Department** Faculty of Engineering, Medi-Caps University Indore is an authentic work.

Further, I/we declare that the content of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for the award of any degree or diploma.

Signature and name of the student(s) with date

Certificate

I/We, **Dr. Kailash Bandhu** certify that the project entitled “**Payroll Management System**” submitted in partial fulfillment for the award of the degree of Bachelor of Technology by **Mukul Hiranandani** is the record carried out by him/them under my/our guidance and that the work has not formed the basis of award of any other degree elsewhere.

Dr. Kailash Bandhu

Computer Science Department

Medi-Caps University, Indore

Mr. Purushottam Sadh

<Name of the Department>

Name of the Organization

Dr. Pramod S. Nair

Head of the Department

Computer Science & Engineering

Medi-Caps University, Indore

Offer Letter of the Project work-II/Internship



Internship Offer Letter

Ref: TCSL/AIP 2021-22/Winter/CT20203206476

Date: 03-Jan-2022

Mukul Hiranandani
Medicaps University, Indore
mukulhiranandani25@gmail.com

Dear Mukul Hiranandani,

Sub: Internship Offer

We are pleased to offer you internship in Tata Consultancy Services (TCS) with the following terms and conditions:

1. The tentative start date is 13-Jan-2022 and end date is 22-Apr-2022. These dates can be changed in discussion with the Project Guide
2. You will be assigned a Project Guide under whose supervision you will work on the project assigned to you.
3. You shall complete your project in accordance with the requirements and guidance of the TCS Project Guide, and maintain qualitative standards as required. You will maintain the discipline, dignity, honor and goodwill of TCS.
4. The arrangement is not that of an employer and an employee and as such you shall not be eligible to any allowances or other benefits as may be available to the employees of TCS.
5. You will observe the rules & regulations and discipline of TCS, and also maintain complete confidentiality and secrecy of the matters pertaining to TCS and/or any data that has been provided to you in the course of your project work. The detailed terms of Confidentiality, Data and Intellectual Property Protection are enclosed as Annexure A. You will be permitted to attend any classes in the college / university at the discretion of the Project Guide if so called for during the period of your project assignment.
6. On completion of your internship you will be required to submit a copy of your project report, which will be the sole property of TCS.
7. You shall not undertake any internship in parallel with this internship
8. In the event of any misconduct or breach of terms of this internship on the part of the Intern during the internship period, TCS reserves the right to terminate internship without any notice.
9. This offer of Internship will be governed as per the Laws of India.

TATA CONSULTANCY SERVICES

Tata Consultancy Services Limited

Yantra Park, Opp. Voltas HRD Trg. Center, Subhash Nagar, Pokhran Road No. 2, Thane (West) 400 601 India
Tel +91 22 6778 2000/2222 Fax +91 40 6778 2190 website: www.tcs.com
Registered Office Nirmal Building, 9th Floor, Nariman Point, Mumbai 400 021



You are required to sign and return a copy of this Internship Offer letter and the Annexure towards your acceptance of the terms and conditions stated therein.

For Tata Consultancy Services

Chandra Koduru
Head, Academic Interface Programme

Accepted,

Name of the Intern: Mukul Hiranandani

Date:

TATA CONSULTANCY SERVICES

Tata Consultancy Services Limited
200, Naraina Road No. 2, Thane (West) 400 601 India

Completion certificate/Letter

Building greater futures through innovation and collective knowledge

TCS Commitment

 In it for good.

 Bring everything.

 Know-how.

 Master the journey.



Building on belief

Internship Certificate

Mukul Hiranandani

Course: BTech in Computer Science and Engineering
Institute: Medi-Caps University, Indore

From **13-Jan-2022** to **22-Apr-2022**
Mentor Name: **Purushottam Sadh**
Project: **Payroll Management System**


Chandra Koduru
Head – Academic Interface Programme

Acknowledgements

I would like to express my deepest gratitude to Honorable Chancellor, **Shri R C Mittal**, who has provided me with every facility to successfully carry out this project, and my profound indebtedness to **Prof. (Dr.) Dileep K Patnayak**, Vice Chancellor, Medi-Caps University, whose unfailing support and enthusiasm has always boosted up my morale. I also thank **Prof. (Dr.) D K Panda**, Pro Vice Chancellor, **Dr. Suresh Jain**, Dean Faculty of Engineering, Medi-Caps University, for giving me a chance to work on this project. I would also like to thank my Head of the Department **Dr. Pramod S. Nair** for his continuous encouragement for betterment of the project.

I express my heartfelt gratitude to my **External Guide, Mr. Purushottam Sadh**, Project Lead, Tata Consultancy Services as well as to my Internal Guide, Dr. Kailash Bandhu, Professor, Department of Computer Science Engineering, MU, without whose continuous help and support, this project would ever have reached to the completion.

I would also like to thank to my team at TCS who extended their kind support and help towards the completion of this project.

It is their help and support, due to which we became able to complete the design and technical report. Without their support this report would not have been possible.

Mukul Hiranandani

B.Tech. IV Year

Department of Computer Science & Engineering

Faculty of Engineering

Medi-Caps University, Indore

Abstract

The actual problem is to maintain different database for an organization whose main purpose is to issue pay-slips for their employees every month working in various departments of the organization, and maintain details of all the departments, employees with various grades, their designations and address details.

In the manual system it is difficult to maintain data and generating different reports according to requesting transaction. In the present system it is becoming difficult to issue pay-slip for all the employee every month by manually going through the various record of the organization. i.e., the manger has to go through all the records of the organization of various departments of the and find out the employee working in a particular department and go through his grade, and he have to check the employee leaves of that month, his earnings and his deductions along with his PF and all other deduction including his IT and savings. So, to perform all these activities it is becoming difficult the admin/manager every month.

Hence in order to overcome the difficulties of the organization the present system is automated to perform all the activities of the organization.

Table of Contents

Chapters	Chapter Name	Page No.
	Report Approval	ii
	Declaration	iii
	Certificate	iv
	Offer Letter of the Project work-II/Internship	v
	Completion letter/certificate	vi
	Acknowledgement	vii
	Abstract	viii
	Table of Contents	ix
	List of figures	xi
	List of tables	xii
	Abbreviations	xiii
	Notations & Symbols	xiv
Chapter 1	Introduction	
	1.1 Introduction of Company	1
	1.2 Introduction of Project	2
	1.3 Literature Review	3
	1.4 Objectives	4
	1.5 Significance	5
Chapter 2	Need of new system	
	2.1 Limitations in Manual System	6
	2.2 Circumstances leading to current new system	7
Chapter 3	Hardware and Software Requirements	
	3.1 Programming Language	8
	3.2 Framework	8
	3.3 Software used	9
Chapter 4	Diagrams	
	4.1 Use Case Diagram	12
	4.2 Sequence Diagram	13
	4.3 Activity Diagram	13
	4.4 ER Diagram	14
	4.5 Information Flow Representation	15
	4.6 System Architecture	16
	4.7 Database Design	17

Chapter 5	Tasks Performed	
	5.1 Revision of Java Concepts	20
	5.2 Learning about micro-service and Spring Boot	21
	5.3 Learning about Postman	22
	5.4 Learning about Git and GitHub	23
	5.5 Project Development	23
	5.6 Module Creation	23
	5.7 Addition of new Library	24
	5.8 Creations of request and response classes	24
	5.9 Shifting of database from Apache derby to PostgreSQL	24
	5.10 Inter service communication	26
	5.11 Implementation of Eureka server	26
	5.12 Learning about Zipkin and Sleuth	28
	5.13 Learning and implementing Postman Collection	28
	5.14 Connecting all micro-services through Feign Client	28
	5.15 Handling Exceptions	28
	5.16 Elimination of repeated code	28
	5.17 Returning Message	28
	5.18 Addition of Salary slip module	28
	5.19 Implementing Logger	28
	5.20 Implementing Swagger	29
	5.21 GitHub Deployment	31
Chapter 6	Annotations	
	6.1 Spring Boot Annotations	33
Chapter 7	Future Scope and Conclusion	
	7.1 Future Scope	34
	7.2 Conclusion	35
	7.3 Bibliography	36

List of Figures

Fig No.	Figure Name	Page No.
Fig 3.1	Spring Boot Logo	9
Fig 3.2	STS Logo	9
Fig 3.3	Postman Logo	10
Fig 3.4	PostgreSQL Logo	10
Fig 3.5	Swagger Logo	11
Fig 4.1	Use Case Diagram	12
Fig 4.2	Sequence Diagram	13
Fig 4.3	Activity Diagram	14
Fig 4.4	ER Diagram	14
Fig 4.5	Information Flow Representation	16
Fig 4.6	System Architecture	17
Fig 4.7	Database design of Company	17
Fig 4.8	Database design of Department	18
Fig 4.9	Database design of Employee	18
Fig 4.10	Database design of Pay Slips	19
Fig 5.1	Postman Snapshot-1	21
Fig 5.2	Postman Snapshot-2	22
Fig 5.3	Postman Snapshot-3	22
Fig 5.4	PostgreSQL Snapshot-1	25
Fig 5.5	PostgreSQL Snapshot-2	25
Fig 5.6	Company's API documentation in Swagger	29
Fig 5.7	Department's API documentation in Swagger	30
Fig 5.8	Employee's API documentation in Swagger	30
Fig 5.9	Pay slips API documentation in Swagger	31
Fig 5.10	GitHub Deployment	32

List of Tables

Table No.	Table Name	Page No.
Table 3.1	Variables in Database	19

Abbreviations

TCS – Tata Consultancy services

IT- Information technology

IoT- Internet of things

MD & CEO – Managing Director & Chief Executive officer

ERD- Entity Relationship Diagram

RAM- Random Access Memory

MHz-Mega Hertz


IFR-Information Flow Representation

STS- Spring Suite Tool

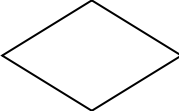
Notations and Symbols/Nomenclatures

Process 

External Entity 

Data Flow 

Multiplicity *

Association 

Chapter 1: Introduction

1.1 Introduction of the organization/client

Tata Consultancy Services is an Indian multinational information technology services and consulting company headquartered in Mumbai, Maharashtra.

It was established on April 1, 1968 by Tata Sons.

The company is a global leader in IT services, digital, and business solution and partner with clients to simplify, strengthen and transform their businesses.

The company provides services in various fields like

- ✓ Analytics and insights
- ✓ Block chain
- ✓ Cognitive Business Operations
- ✓ Cyber Security
- ✓ Enterprise Applications
- ✓ Quality Engineering
- ✓ Automation and Artificial Intelligence
- ✓ Cloud Technology
- ✓ Consulting
- ✓ IOT and digital Engineering
- ✓ Sustainability Services

Key People

- Natarajan Chandrasekaran (Chairman)
- Rajesh Gopinathan (MD & CEO)

1.2 Introduction to Project

A payroll management system is a software that is used to manage all employee's financial records in a simple and automated fashion. This payroll management system manages employee's salaries, deductions, other conveyance, net pay, bonuses and generation of pay-slips, etc.

We don't have to worry about handling, managing, and creating payslips, salaries, and deductions of the employees. The tax deductions are also automated or handled by the outsourcing team. You only have to focus on major tasks such as the efficiency of sales, revenue, strategy, etc.

1.3 Literature review

Payroll processing is crucial in an organization because it involves the payment of the organization's work forces and protection of its reputation by ensuring that the organization complies with the government authority's employment legislations. It calculates salary, allowance, overtime, contributions and deductions of employees that varies across designations. The payroll process must responsive to changes in employment status, latest rules or acts by the government authorities such as re-allocation of employee's contribution in Employees Provident Fund and taxes legislation. However, an organization may face several challenges in payroll processing such as to pay employees accurately on time, meet obligations between employees and employers and uphold other legislative responsibilities. Tedious, time consuming and increased effort to process the payroll, particularly in large organizations with huge number of employees, are common issues in manual payroll management system. As the demand to produce timely, accurate and efficient payroll intensify, it leads to increasing needs for computerized or automated payroll management system.

1.4 Objective of the project

The payroll management system is a set of processes that helps you streamline salaries, bonuses, deductions, taxes, and other necessary aspects of the net pay of all the employees in your organization.

There are two primary objectives of the payroll management system in India. One is the macro-objective, which is related to sales, strategy, revenue, etc. Another is micro, which is associated with the daily tasks of the business.

The aim behind having a payroll management system is to automate and streamline micro tasks such that the HR team has time to focus on the macro tasks.

The main objective to start this project was learning new technologies and skills to implement ahead in the company projects.

What is to be achieved and method of measuring the extent of that achievement

This application can be further enhanced to cover each aspect of the institutes. Employee management system will be extended towards a complete management package for the institute. It will really help to manage and organize the employees' detail very efficiently and effectively. The application will prove itself as a powerful tool which enhance the working power of the administration department and can help in improving overall management of the institute.

The following are some of the important enhancements that can take place into the current working software application:

This includes management of employee's salary and each kind of financial features which helps the institute as a financial tool.

In Our System an Administrator easily Modify and Update the any Employee's information.

1.5 Significance of the Project

The majority of the time, data security rules are stringent. Employee information is kept as private as possible, which is reflected in several degrees of administrative and access privileges. It also gives employees access to information about their salary impairments.

This also gives a comprehensive picture of the company's salary, which is crucial for a financial evaluation. In the event of pay hikes, a simple computation may provide the corporation with the full financial impact of such information.

Many modern payroll systems are as cost-effective as remote cloud alternatives in most circumstances. To keep the database up to date, the company merely requires active registration. Employee salary management systems use a small number of expensive resources, which saves a significant amount of money.

Chapter 2: Need of new system

2.1 Limitations in Manual System

In the manual system it is difficult to maintain data and generating different reports according to requesting transaction. In the present system it is becoming difficult to issue pay-slip for all the employee every month by manually going through the various record of the organization. i.e., the manger has to go through all the records of the organization of various departments of the and find out the employee working in a particular department and go through his grade, and he have to check the employee leaves of that month, his earnings and his deductions along with his provident fund and all other deduction including his IT and savings. So, to perform all these activities it is becoming difficult the admin/manager every month.

2.2 Circumstances leading to the current new system

Employees are the backbone of any company; their management plays a major role in deciding the success of the organization. Employee information management helps in deciding the future management needs and any changes that have to be made for greater productivity. It keeps the records of the functions performed by the individual employee playing a vital role at the time of performance appraisal. Employee management software can carry out many functions like employee data analysis, employee monitoring, centralized employee database, management of the time sheet, etc.

The usage of payroll management system would ultimately reduce the overall management costs thus ensuring greater profits and reduced burdens. The system is very easy to use and can adjust into any business frame.

Chapter-3: Hardware and Software Requirements

3.1 Programming Language

Java-Java is a general-purpose, class-based, object-oriented programming language designed for having lesser implementation dependencies. It is a computing platform for application development. Java is fast, secure, and reliable, therefore. It is widely used for developing Java applications in laptops, data centers, game consoles, scientific supercomputers, cell phones, etc.

What is Java used for?

Here are some important Java applications:

- It is used for developing Android Apps
- Helps you to create Enterprise Software
- Wide range of Mobile java Applications
- Scientific Computing Applications
- Use for Big Data Analytics
- Java Programming of Hardware devices
- Used for Server-Side Technologies like Apache, JBoss, GlassFish, etc.

3.2 Framework

Spring Boot

Spring Boot is an open-source Java-based framework used to create a micro- Service. It is developed by Pivotal Team and is used to build stand-alone and production ready spring applications.

Spring Boot provides a good platform for Java developers to develop a stand-alone and production-grade spring application that we can just run with minimum configurations and without the need for an entire Spring configuration setup.

Advantages

Spring Boot offers the following advantages to its developers –

- Easy to understand and develop spring applications
- Increases productivity
- Reduces the development time

Goals

Spring Boot is designed with the following goals –

- To avoid complex XML configuration in Spring
- To develop a production ready Spring applications in an easier way
- To reduce the development time and run the application independently
- Offer an easier way of getting started with the application



Fig 3.1 Spring Boot Logo

Why Spring Boot?

We can choose Spring Boot because of the features and benefits it offers

- It provides a flexible way to configure Java Beans, XML configurations, and Database Transactions.
- It provides a powerful batch processing and manages REST endpoints.
- In Spring Boot, everything is auto configured; no manual configurations are needed.
- It offers annotation-based spring application
- Eases dependency management
- It includes Embedded Servlet Container

3.3 Software Used

Spring Tool Suite (STS)

Spring Tool Suite is an IDE to develop Spring applications. It is an Eclipse-based development environment. It provides a ready-to-use environment to implement, run, deploy, and debug the application. It validates our application and provides quick fixes for the applications.



Fig 3.2 STS Logo

Postman

Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs—faster.



Fig 3.3 Postman Logo

PostgreSQL

PostgreSQL is a powerful, open-source object-relational database system that uses and extends the SQL language combined with many features that safely store and scale the most complicated data workloads.

PostgreSQL comes with many features aimed to help developers build applications, administrators to protect data integrity and build fault-tolerant environments, and help you manage your data no matter how big or small the dataset. In addition to being free and open source, PostgreSQL is highly extensible. For example, you can define your own data types, build out custom functions, even write code from different programming languages without recompiling your database.



Fig 3.4 PostgreSQL Logo

Swagger

Swagger is the standard way of documenting the Standard APIs. Swagger is helpful when deploying APIs in azure. Swagger is primarily used for documenting API; now the question arises that why document APIs. The building APIs that are internal in the enterprise or for the public consumption, the theme is the same that the developers usually use in the apps that they are building. For the other developers to be able to use our API, the API must be properly documented; otherwise, how would they know that what are the endpoints exposed by the API and what are the operations supported on those endpoints? What parameters should they pass, and what will they get back? What authentication methods to use. To answer these questions, it is very important to document the APIs; if you want APIs to be consumed and properly used. Swagger and Open API specification are the ways to document an API specifying that what exactly APIs can do.



Fig 3.5 Swagger Logo

3.4 System Requirements

Hardware Requirements:

- 233 MHz Pentium processor or other compatible
- 512 MB SD-RAM
- GB hard disk

Software Requirements:

- Operating System -Windows 98 onwards
- Application Software –Spring Tool Suite
- Database Tool – PostgreSQL
- Postman

Chapter-4: Diagrams

4.1 Use Case Diagram

Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the user use it, but not how the system operates

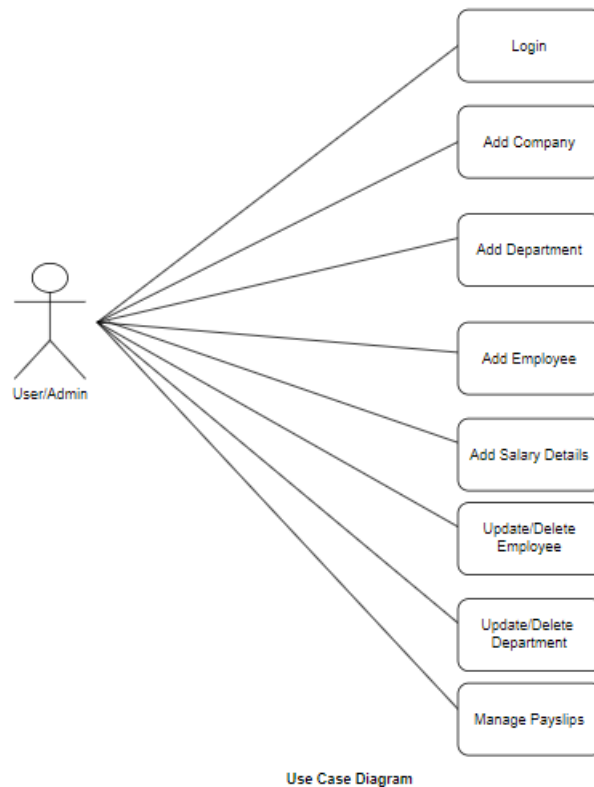
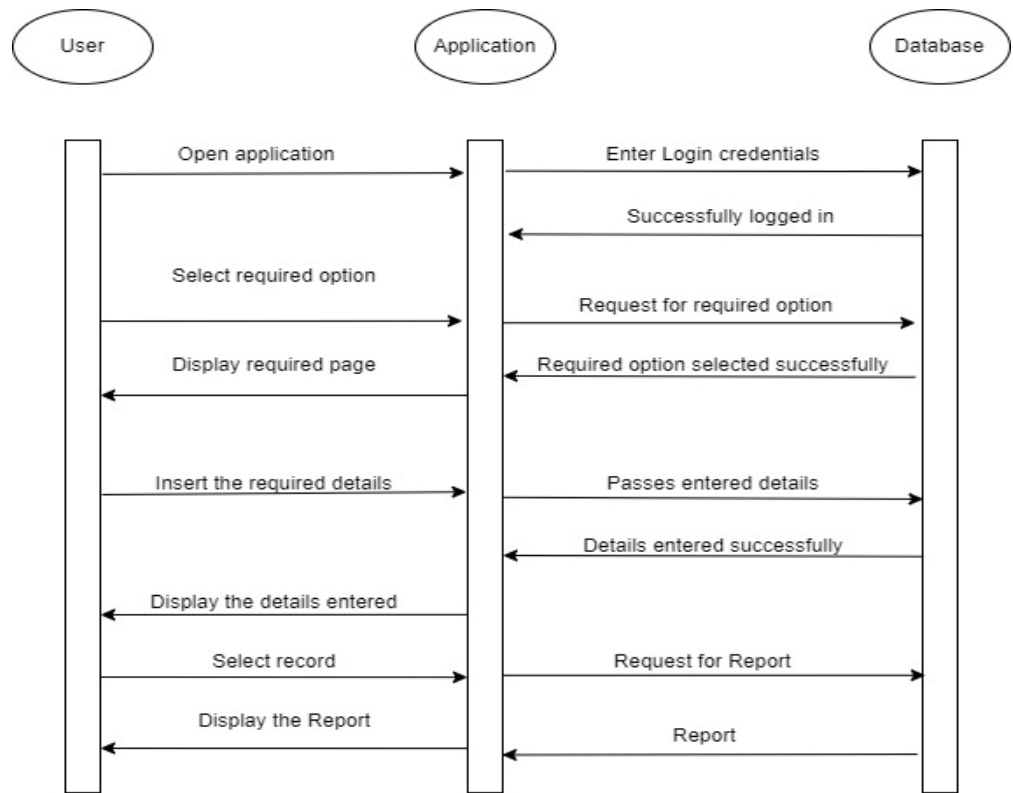


Fig 4.1 Use Case Diagram

4.2 Sequence Diagram

A sequence diagram or system sequence diagram (SSD) shows object interactions arranged in time sequence in the field of software engineering. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.



Sequence Diagram

Fig 4.2 Sequence Diagram

4.3 Activity Diagram

An activity diagram is a behavioral diagram i.e., it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.

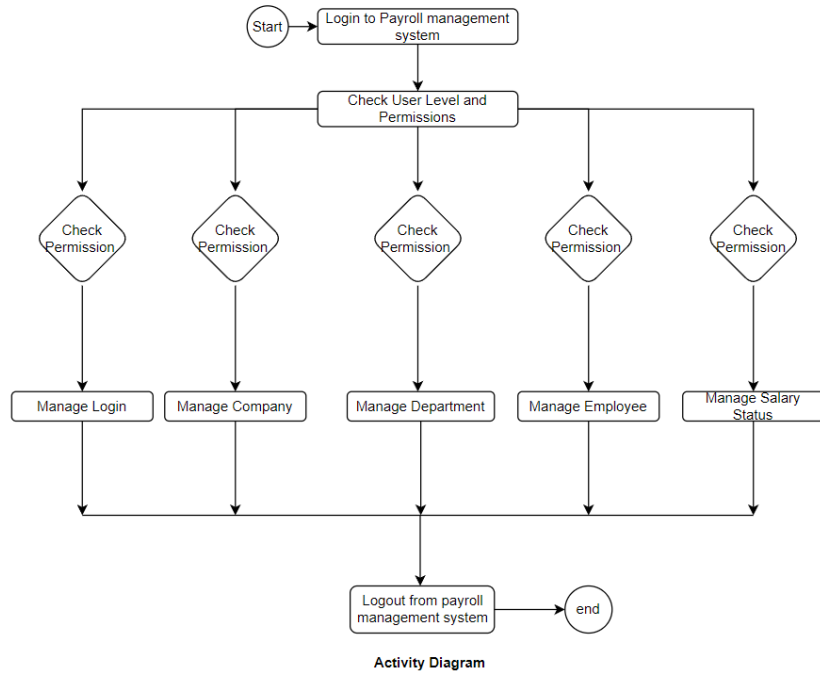


Fig 4.3 Activity Diagram

4.4 ER Diagram

ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.

ER Diagram

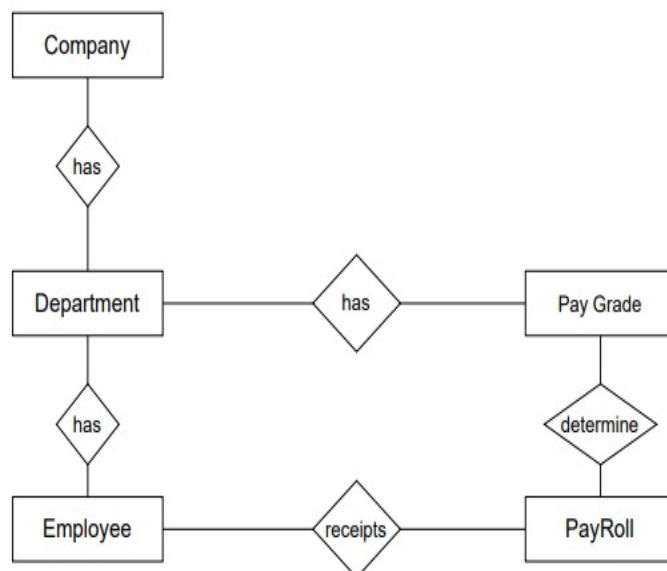


Fig 4.4 ER Diagram

We have analyzed the feasibility of the system in terms of following spans:

- Technical Feasibility
- Operational Feasibility
- Economic Feasibility

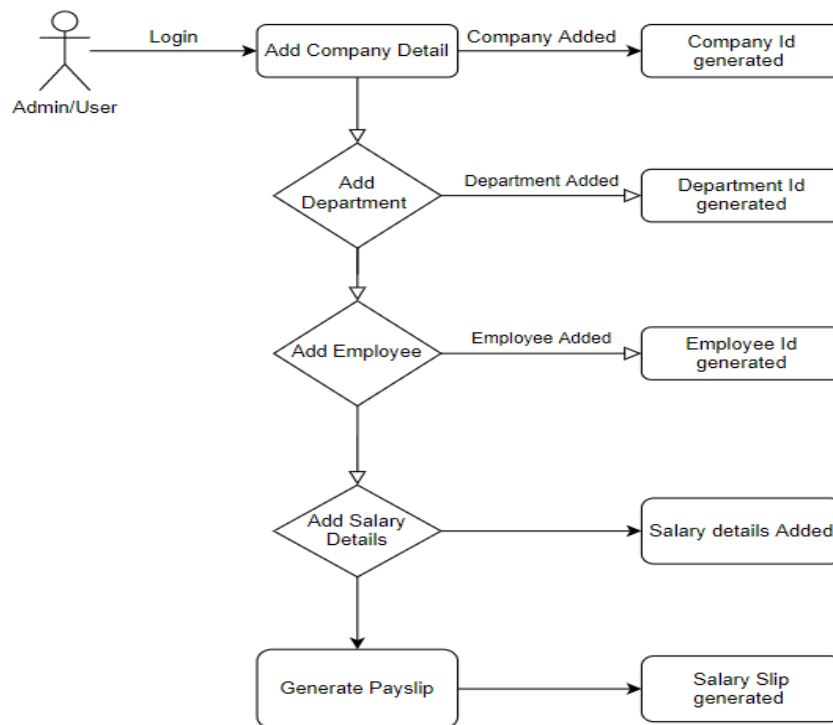
Technical Feasibility: The development process of Payroll Management System would be advantageous to the organization because we would make use of only the currently available resources of the organization. All the tools needed for the development are already available with the organization and the organization does not have to acquire any new resources. The technical feasibility is also attributed to the fact that the system does not make use of any additional or external third-party components which can lead to increased load on the system.

Operational Feasibility: The Payroll Management System is intended to provide a very user-friendly and easy to use interface which is beneficial for both the visitors as well as the operators who help in providing support for the system. This system would also be easily acceptable among the visitor and administrator, as there is no need of any special skill set for using the application. This system also benefits the users as they do not have to download anything on their terminals increasing their efficiency and ease of use.

Economic Feasibility: The Payroll Management System has a very low development cost. The low cost is attributed to the usage of the existing resources of the organization. As the software is very user friendly and easy to use, there is no need to provide special training to the users of the software, thus saving valuable time and money.

4.5 Information Flow Representation

An information flow diagram (IFD) is a diagram that shows how information is communicated (or "flows") from a source to a receiver or target (e.g., $A \rightarrow C$), through some medium. The medium acts as a bridge, a means of transmitting the information.



Information Flow Representation

Fig 4.5 Information Flow Representation

4.6 System Architecture

A system architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

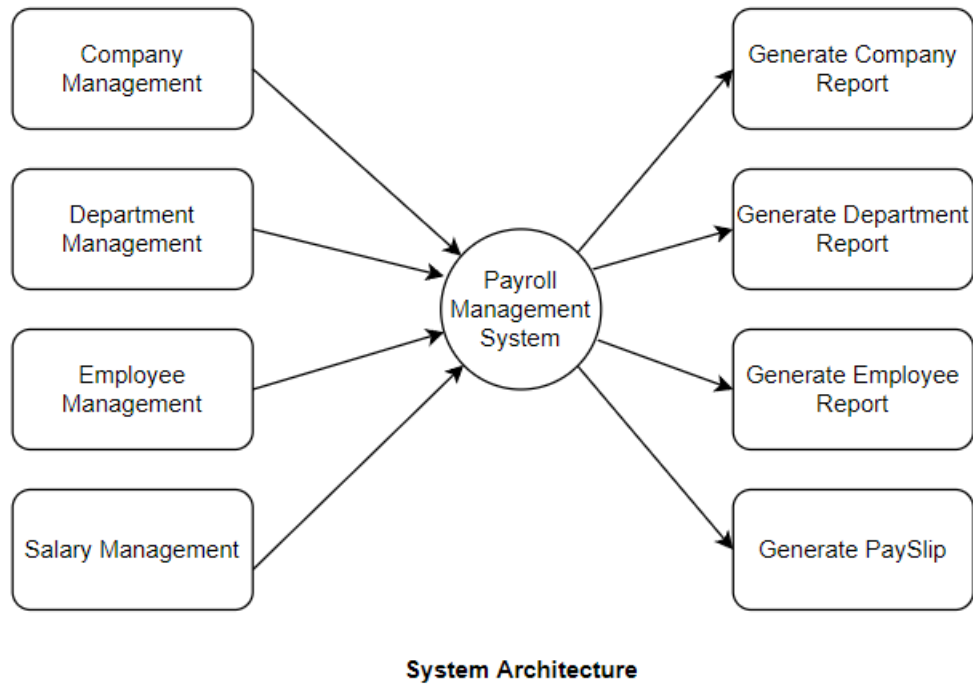


Fig 4.6 System Architecture

4.7 Database Design

Database design is the organization of data according to a database model. The designer determines what data must be stored and how the data elements interrelate. With this information, they can begin to fit the data to the database model. Database management system manages the data accordingly.

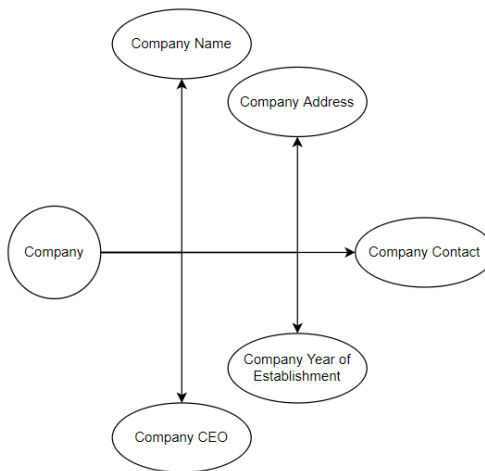


Fig 4.7 Database Design of Company

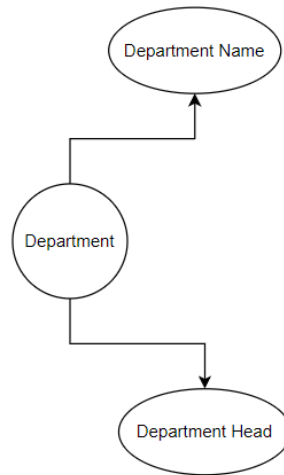


Fig 4.8 Database Design of Department

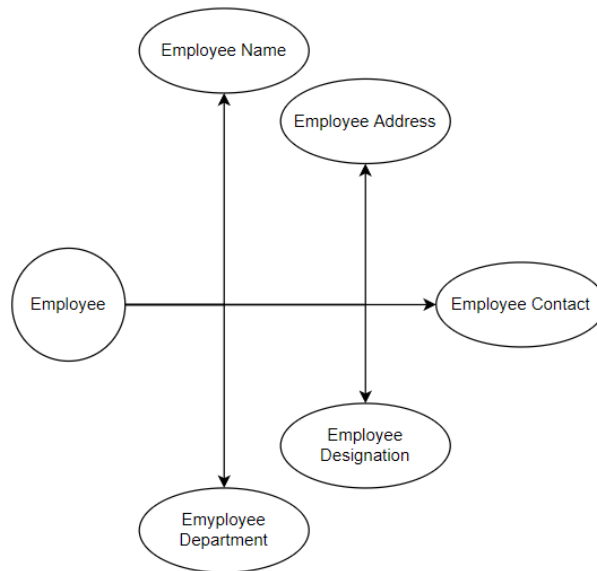


Fig 4.9 Database Design of Employee

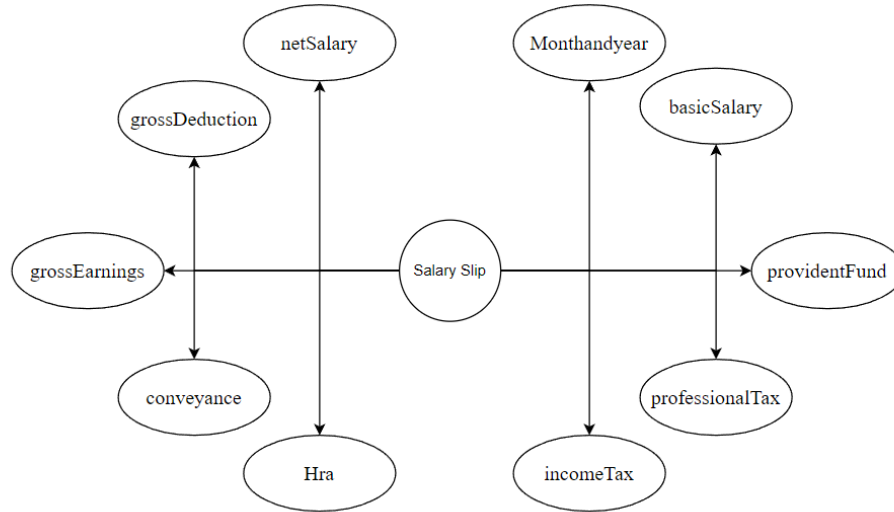


Fig 4.10 Database Design of Pay slip

We created four entities, Company, Department, Employee and Salary slip. In each of these we created variables as information of respective entities.

Company	Department	Employee	Salary Slip
companyID	departmentID	employeeID	Monthandyear
companyName	departmentName	employeeName	basicSalary
companyAddress	departmentHead	employeeContact	providentFund
companyContact	timestamp	employeeAddress	professionalTax
companyYOE	deleted	employeeDept	incomeTax
CompanyCEO	company	employeeDesign	Hra
Timestamp	empList	Timestamp	conveyance
Deleted		Company	grossEarnings
empList		Deleted	grossDeduction
depList		Dept	netSalary

Table 4.1 Variables in Database

Chapter-5: Tasks Performed

5.1 Revision of Java Concepts

The internship began with me brushing up on my skills of Advance Java and Collection Framework.

What is Collection Framework?

A Java collection framework provides an architecture to store and manipulate a group of objects. A Java collection framework includes the following:

- Interfaces
- Classes
- Algorithm

The Java collection framework provides the developers to access pre-packaged data structures as well as algorithms to manipulate data. Next, let us move to the Java collections framework hierarchy and see where these interfaces and classes resides.

5.2 Learning about Micro Service and Spring Boot

Micro Service is an architecture that allows the developers to develop and deploy services independently. Each service running has its own process and this achieves the lightweight model to support business applications.

Advantages

Micro services offer the following advantages to its developers –

- Easy deployment
- Simple scalability
- Compatible with Containers
- Minimum configuration
- Lesser production time

Spring Boot

Spring Boot provides a good platform for Java developers to develop a stand-alone and production-grade spring application that you can just run. You can get started with minimum configurations without the need for an entire Spring configuration setup.

Advantages

- Easy to understand and develop spring applications
- Increases productivity
- Reduces the development time

5.3 Learning about Postman

Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs—faster.

All the HTTP Requests were tested here.

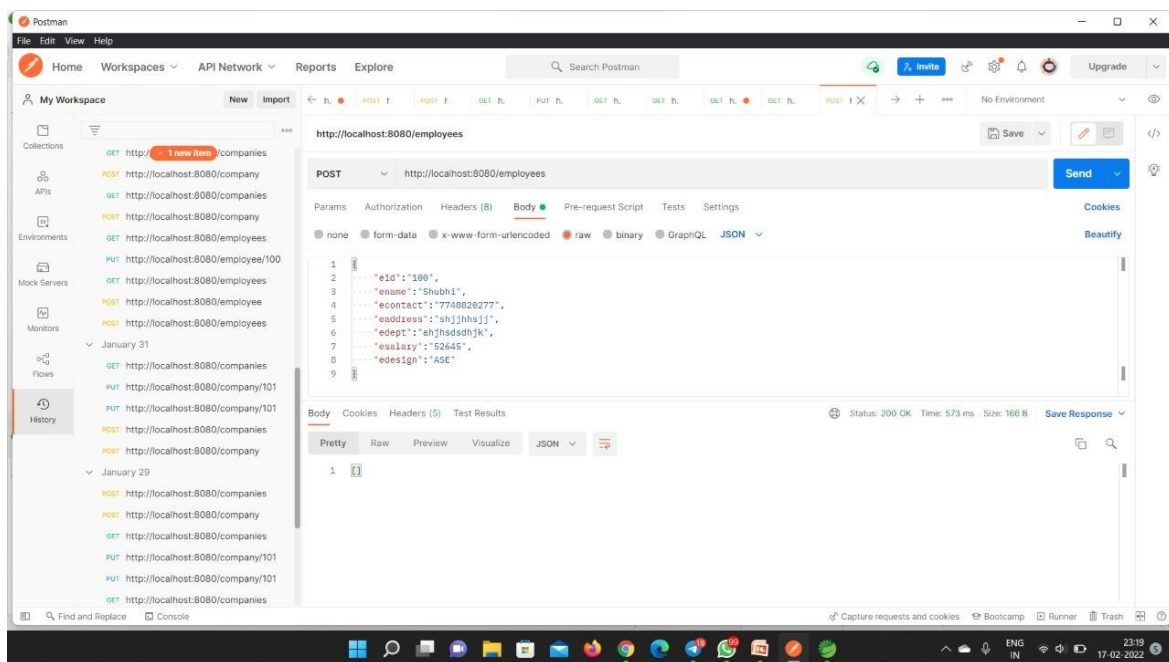


Fig 5.1 Postman Snapshot-1

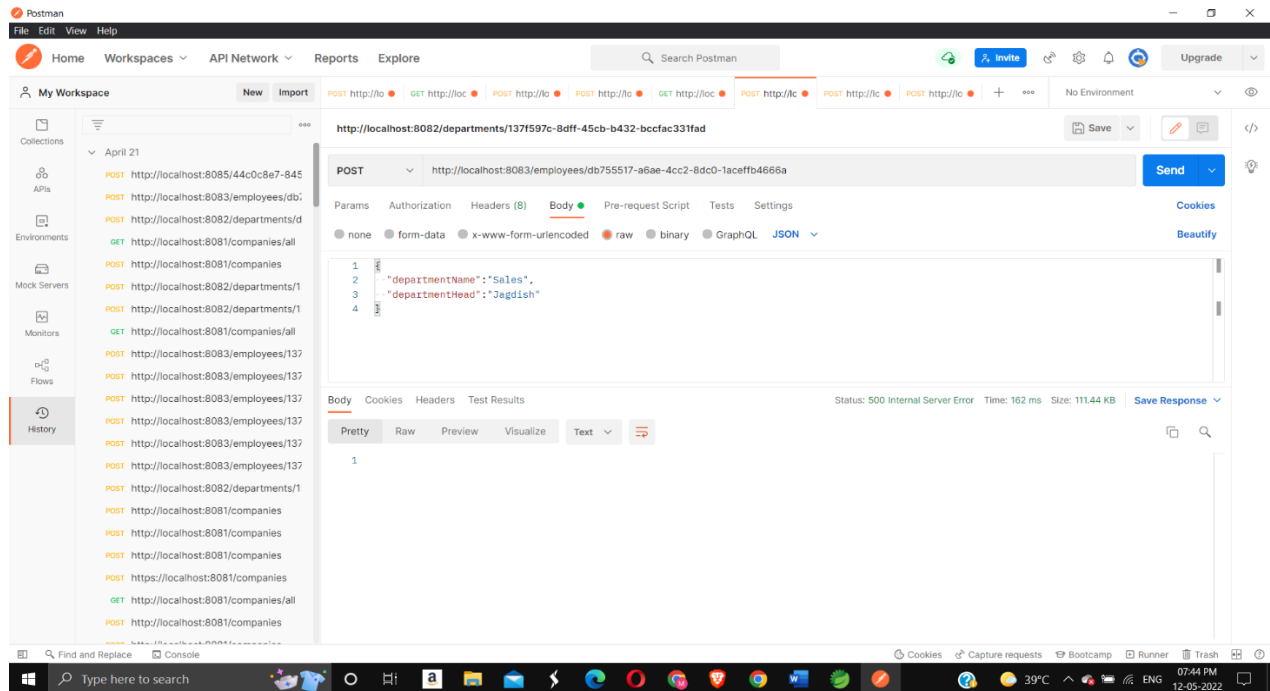


Fig 5.2 Postman Snapshot-2

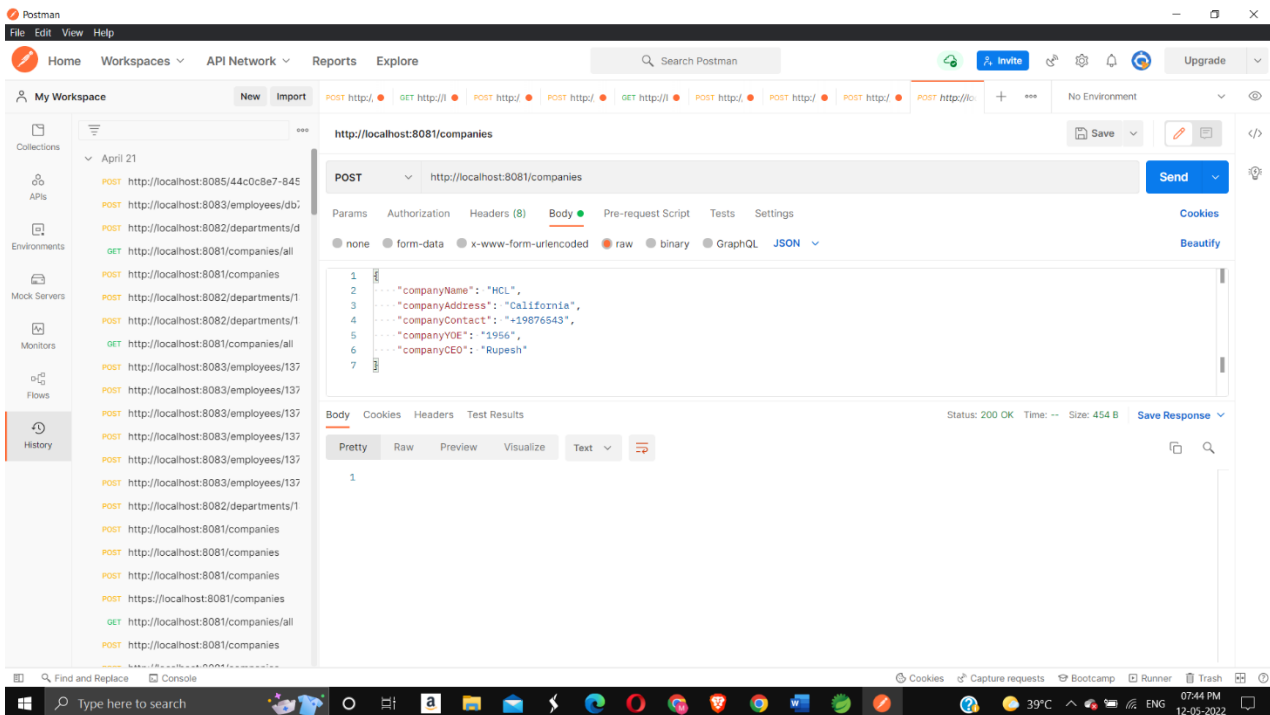


Fig 5.3 Postman Snapshot-3

5.4 Learning about Git and GitHub

What is Git?

Git is software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows.

What is GitHub?

GitHub is an online software development platform used for storing, tracking, and collaborating on software projects. It enables developers to upload their own code files and to collaborate with fellow developers on open-source projects. GitHub also serves as a social networking site in which developers can openly network, collaborate, and pitch their work

5.5 Project Development

Project Development began by creating a rough sketch of how our payroll management system will look like and how it will work. Then according to that rough idea, all the necessary diagrams like use case diagram, activity diagram, sequence diagram etc. were made.

5.6 Module Creation

There were three modules namely company module, department module and employee module. These modules were created in spring boot and various dependencies like spring starter web, JDBC, Apache Derby were included in all three.

What is Apache Derby?

Apache Derby is a local relational database provided by Spring Boot to store data. The data is stored temporarily in it and gets erased every time when the Spring Boot is shut down.

Basic operations like Create, Read, Update and Delete i.e., CRUD operations were performed in all the three modules.

5.7 Addition of new library

Lombok- Lombok is a java library which was implemented into the project, by use of which, we don't need to use getters and setters which makes the code short and efficient.

5.8 Creation of Request and Response Classes

Instead of directly accessing the object of one class by another class a bridge class was built so if one class wants to call an object of another class it is done by bridge class

5.9 Shifting of database from Apache Derby to PostgreSQL

The database was changed from Apache Derby to Postgres SQL as because Apache Derby was a temporary database. Records get permanently deleted when the Spring Boot was shutdown.

PostgreSQL is a powerful, open-source object-relational database system with over 30 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance.

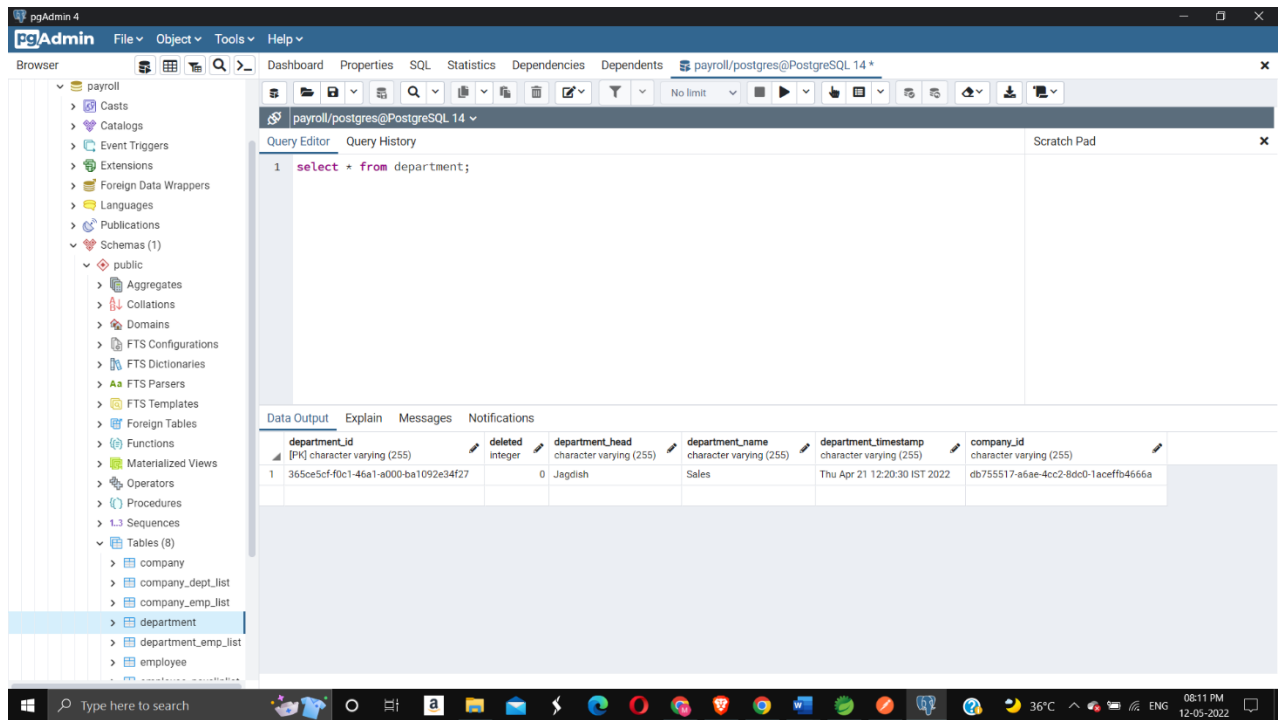


Fig 5.4 PostgreSQL Snapshot-1

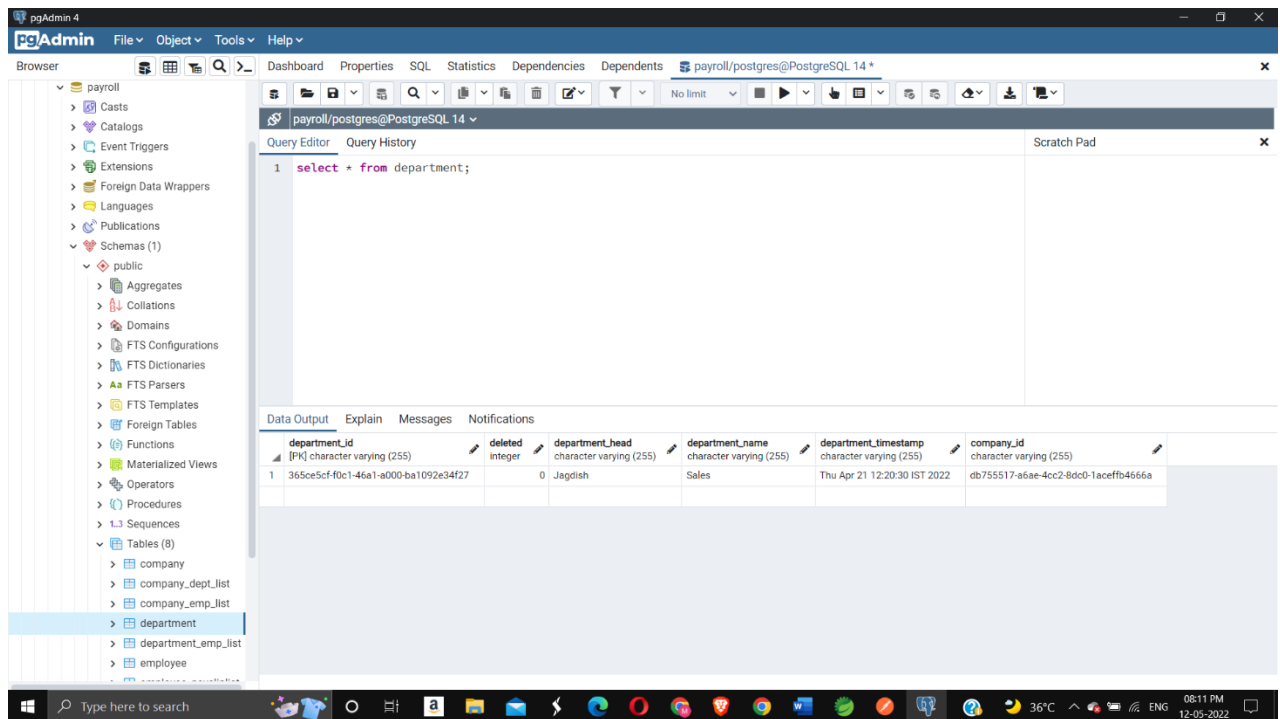


Fig 5.5 PostgreSQL Snapshot-2

5.10 Inter Service Communication

Inter-service communication is basically two Microservices communicating using HTTP protocol

Inter Service communication are of two types

- a) **Synchronous Communication Style-** In this communication style, the client service expects a response within time and wait for a response by blocking a while. This style can be used by simple using HTTP protocol usually via REST. It is the simplest possible solution for microservices inter-service communication to interact with services. The client can make a REST call to interact with other services. The client sends a request to the server and waits for a response from the service (Mostly JSON over HTTP). For example, Spring Cloud Netflix provides the most common pattern for synchronous REST communication such as Feign.
- b) **Asynchronous Communication Style-** In this communication style, the client service doesn't wait for the response coming from another service. So, the client doesn't block a thread while it is waiting for a response from the server. Such type of communications is possible by using lightweight messaging brokers. The message producer service doesn't wait for a response. It just generates a message and sends message to the broker; it waits for the only acknowledgement from the message broker to know the message has been received by a message broker or not.

5.11 Implementation of Eureka Server

Eureka Server is an application that holds the information about all client-service applications. Every Micro service will register into the Eureka server and Eureka server knows all the client applications running on each port and IP address. Eureka Server is also known as Discovery Server.

Eureka Server is used to keep track of each microservice so if in case any error occurs, it can be tracked.

5.12: Learning about Zipkin and Sleuth

Zipkin is a Java-based distributed tracing system to collect and look up data from distributed systems. Too many things could happen when a request to an HTTP application is made. A request could include a call to a database engine, to a cache server, or any other dependency like another microservice. That's where a service like Zipkin can come in handy. An application could be sending timing data in the background so that when it's time to troubleshoot, you can have an integrated view with Zipkin.

Sleuth is another tool from the Spring cloud family. It is used to generate the trace id, span id and add this information to the service calls in the headers and MDC, so that It can be used by tools like Zipkin and ELK etc. to store, index and process log files.

5.13 Learning and Implementing Postman Collection

Collection in Postman means a group of API requests that are already saved in the Postman and can be arranged into folders. Any number of folders can be created inside a collection.

Putting similar requests into folders and collections helps the client in better organization and documentation of their requests.

All the APIs requests can be stored and saved within a collection, and these collections can be shared amongst the team in the Postman workspace.

5.14 Connecting all microservices through Feign Client

All the micro services were connected through Feign Client method. The Feign is a declarative web service (HTTP client) developed by Netflix. Its aim is to simplify the HTTP API clients. It is a Java to HTTP client binder. If you want to use Feign, create an interface, and annotate it. It provides pluggable annotation support, including Feign annotations and JAX-RS annotations.

It is a library for creating REST API clients. It makes web service clients easier. The developers can use declarative annotations to call the REST services instead of writing representative boilerplate code.

Spring Cloud OpenFeign provides OpenFeign integrations for Spring Boot apps through auto-configuration and binding to the Spring Environment. Without Feign, in Spring Boot application, we use RestTemplate to call the User service. To use the Feign, we need to add spring-cloud-starter-openfeign dependency in the pom.xml file.

5.15 Handling Exceptions

The Exception Handling in Java is one of the powerful mechanisms to handle the runtime errors so that the normal flow of the application can be maintained. All the exceptions were handled in every part of the code.

5.16 Elimination of Repeated Code

We made our code a lot more efficient by implementing normal function methods and removed all the repeated code.

5.17 Returning Message

Message was returned after every successful CRUD operation performed in postman. Our mentor asked us to return the whole body which we were giving as input as well as a message “added successfully” as the output. In case of an error, message with the error number was returned.

5.18 Addition of Salary Slip Module

At first, we were giving the salary details through employee module but later we created a separate module for generating pay slips. More variables were added into this new module and overall, it was

5.19 Implementing Logger

Spring boot logging is defined as a framework that enables developers to trace out errors that might occur in the running of the application. Logging in spring boot is basically an API that provides tracing out of information along with a recording of any critical failure that might occur in the application during its run. Spring boot uses a common logging framework to implement all internal logging and leaves the log implementation open.

5.20 Implementing Swagger

Swagger is the standard way of documenting the Standard APIs. Swagger is helpful when deploying APIs in azure. Swagger is primarily used for documenting API; now the question arises that why document APIs. The building APIs that are internal in the enterprise or for the public consumption, the theme is the same that the developers usually use in the apps that they are building. For the other developers to be able to use our API, the API must be properly documented; otherwise, how would they know that what are the endpoints exposed by the API and what are the operations supported on those endpoints? What parameters should they pass, and what will they get back? What authentication methods to use. To answer these questions, it is very important to document the APIs; if you want APIs to be consumed and properly used. Swagger and Open API specification are the ways to document an API specifying that what exactly APIs can do.

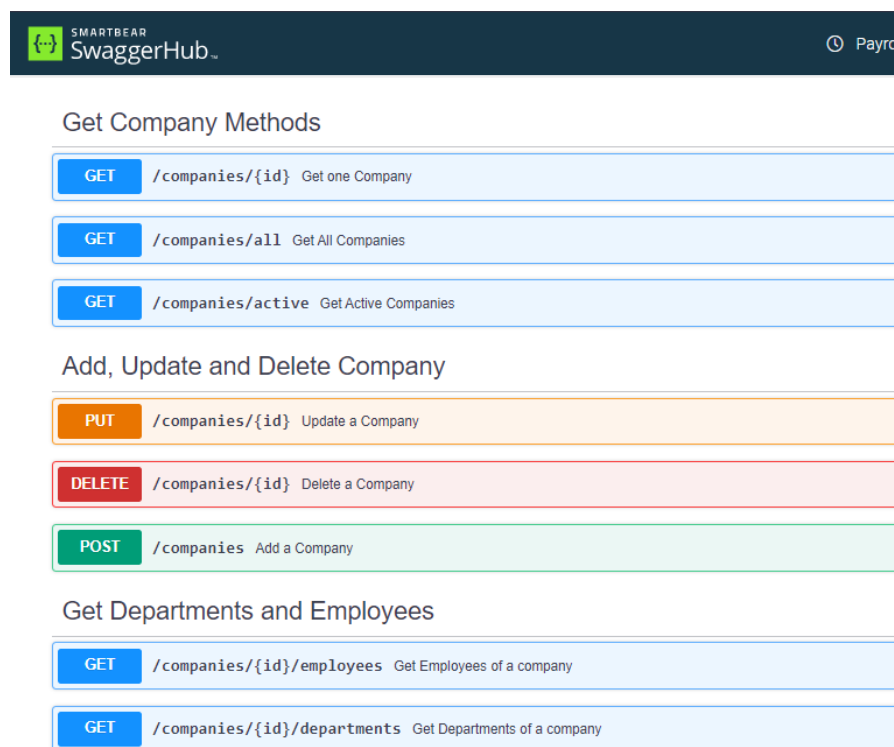


Fig 5.6 Company's API documentation in Swagger

GET methods for departments

GET	/departments/{id}	Get a department
GET	/departments/all	Get all departments

Add, Update and Delete Department

PUT	/departments/{id}	Update a department
POST	/departments/{id}	Add a department
DELETE	/departments/{id}	Delete a department

Fig 5.7 Department's API documentation in Swagger


		
GET methods for employees		
GET	/employees/{id}	Get an employee
GET	/employees/all	Get all employees
Add, Update and Delete Employee		
PUT	/employees/{id}	Update an Employee
POST	/employees/{id}	Add an Employee
DELETE	/employees/{id}	Delete an Employee
Salary Slips Operations		
GET	/employees/{id}/salaryslips	Get all Salary Slips
GET	/employees/{id}/salaryslips/{month}/{year}	Get particular Salary Slip

Fig 5.8 Employee's API documentation in Swagger

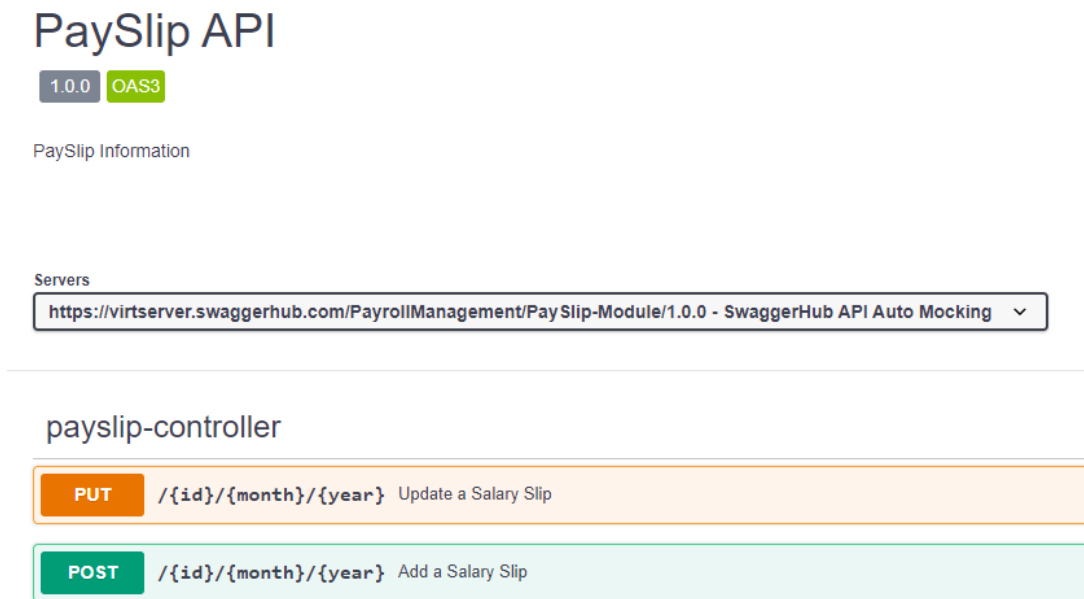


Fig 5.9 Pay slip API documentation in Swagger

5.21 Deploying project on GitHub

At last, the project was deployed on GitHub in five separate branches namely Company branch, Department branch, Employee branch, Eureka Server branch and Pay slip branch. Separate branches were created so as the whole code doesn't get mixed up. Along with the code all the other documents were also uploaded.

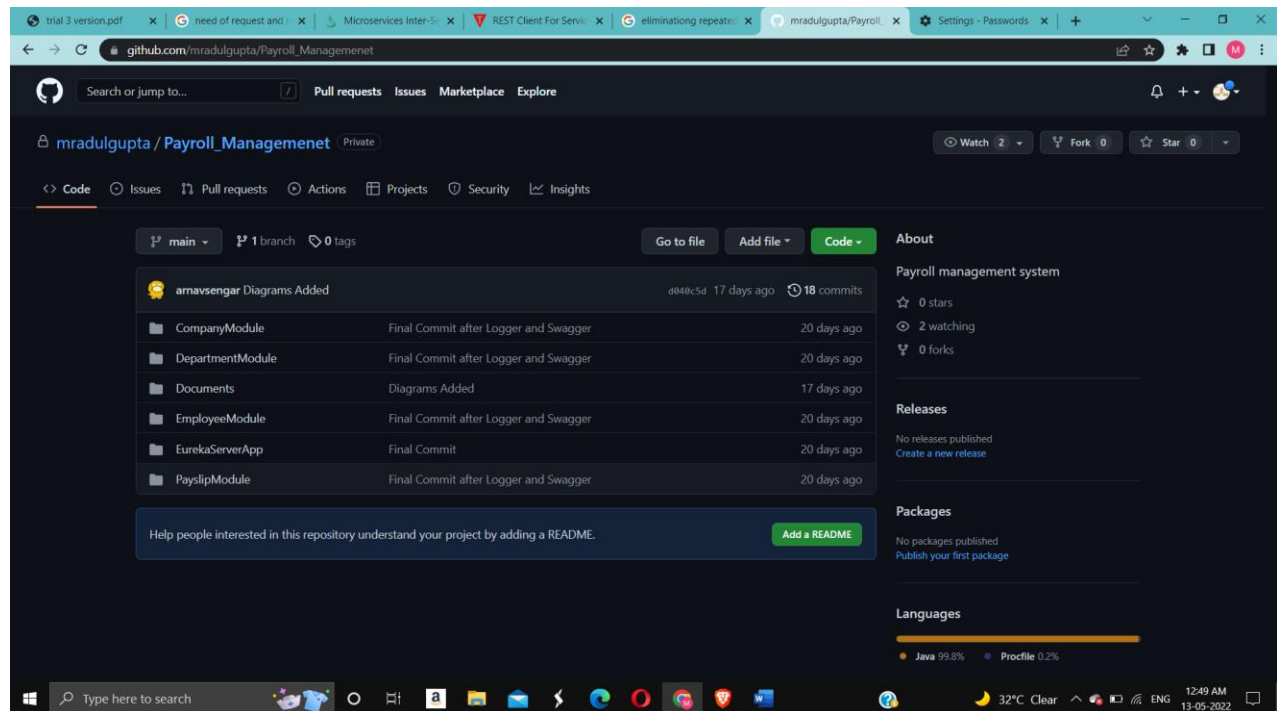


Fig 5.10 GitHub Deployment

Chapter 6: Annotations

6.1 Spring Boot Annotations

@Entity: annotation specifies that the class is an entity and is mapped to a database table.

@id: annotation specifies the name of the database table to be used for mapping.

@RestController: Spring RestController annotation is used to create RESTful web services using Spring MVC. Spring RestController takes care of mapping request data to the defined request handler method.

@AutoWired: Spring provides annotation-based auto-wiring by providing @Autowired annotation. It is used to autowire spring bean on setter methods, instance variable, and constructor.

@Request Mapping: It is used to map the web requests. It has many optional elements like consumes, header, method, name, params, path, produces, and value. We use it with the class as well as the method.

@RequestBody: It is used to bind HTTP request with an object in a method parameter. Internally it uses HTTP MessageConverters to convert the body of the request.

@PathVariable: The @PathVariable annotation can be used to handle template variables in the request URI mapping

@service: Spring @Service annotation is used with classes that provide some business functionalities. Spring context will autodetect these classes when annotation-based configuration and class-path scanning is used.

@SpringBootApplication: Spring Boot @SpringBootApplication annotation is used to mark a configuration class that declares one or more @Bean methods and also triggers auto-configuration and component scanning. It's same as declaring a class with @Configuration, @EnableAutoConfiguration and @ComponentScan annotations.

Chapter 7: Future Scope

Automated payroll management system

Organizations are moving past geographical boundaries to hire employees. With the advancement of technology, a company's ability to hire employees from different parts of the world has come into play. Remote workers and freelancers rely on the highly advanced automated payroll management process of organizations.

The implementation of automation will help the company safeguard valuable employee information falling into wrong hands.

Pay transparency

A lot of organizations are giving up discrimination on the basis of gender especially in pay. Pay transparency is a tool that is greatly helpful in this. It gives the team complete information about the salaries being paid for different roles and how the pay structure is devised.

HR technology making payroll more efficient

A major misconception about payroll outsourcing and automated payroll system is that it's very expensive. A balance between HR technology and payroll management will garner efficient results for your business. When investing in such technology, make sure you target at programs that make time for tracking and scheduling a lot easier.

Payment frequency changes

Different companies have different payment models like bi-weekly or monthly. This approach is changing, especially now that more and more gig workers are becoming a part of the organization. The trend of instant payments has become popular and is easy for the employee and the organization as well. This is a major change we're seeing in new HR payroll software.

Conclusion

“Payroll Management System” software developed for a company has been designed to achieve maximum efficiency and reduce the time taken to handle the payroll activity. It is designed to replace an existing manual record system thereby reducing time taken for calculations and for storing data. The system is strong enough to withstand regressive daily operations under conditions where the database is maintained and cleared over a certain time of span. The implementation of the system in the organization will considerably reduce data entry time and also provide readily calculated reports.

Bibliography

Documentation of Company:

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

Project Documentation through:

<https://app.swaggerhub.com/apis/PayrollManagement/Company-Module/1.0.0>

<https://app.swaggerhub.com/apis/PayrollManagement/Department-Module/1.0.0>

<https://app.swaggerhub.com/apis/PayrollManagement/Employee-Module/1.0.0>

<https://app.swaggerhub.com/apis/PayrollManagement/PaySlip-Module/1.0.0>

Spring Boot

<https://www.youtube.com/c/JavaBrainsChannel>