東北大學

Northeastern University

PROFESSIONAL PRACTICE REPORT
AT
NEUSOFT CORPORATION

学　号(Student ID): 2128025

姓　名(Name): hasan Ali 拉赫曼

专　业(Major): Software Engineering

学　院(College): Software College

校内导师 Supervisor of NEU:高天寒

校外导师 Tutor of Neusoft: Hongchao Yu

Acknowledgement

Here, we would like to thanks our university (Northeastern University, Shenyang, Liaoning, China) for giving us an excellent opportunity to do our internship program in a very famous, valuable and multinational software company named NeuSoft (Shenyang, Liaoning, China). And also, thanks to NeuSoft for giving us an excellent office environment, co-operative team members, valuable seniors and collogues.

The successful of this internship program could not been possible without the support and assistance of many people. We feel immensely blessed to have this opportunity during our studies. We would like to take this opportunity to offer our earnest admiration to each one of them. We express our sentiment of gratitude to our teachers at NeuSoft Lin CaoYu 曹玉林老师, HongChao Yu 于洪超老师, Xiao MingLiang 肖明亮老师 who have been a continuous source of inspiration. Without their constant guidance, this report would have been nowhere near completion. our gratitude for their trust and generosity goes beyond words.

During the internship, the staff at NeuSoft guiding us were very helpful and extended their valuable guidance and help whenever required for the projects which We worked on, thanks to Mrs Luo Duan 段落老师 and Mr. Xiao Zhang 小张老师 for your great efforts.

Finally, our thanks and appreciation go to our family for their immense encouragement and support. As well as, to our group. Thanks to all our colleagues and friends, you all made this journey worthwhile.

Table 1 below gives a view of the duration of the internship.

Start Date	End Date	Total Duration	Position	Working Days	Supervisor	Class Hours
5th September 2022	**14th March 2023**	6 Months	Intern	5 days a week	Hongchao Yu	6:30 pm to 8:30 pm (2 hours)

Abstract

The NeuSoft Opportunity Project Management System (NPMS) is a sales management software solution designed to help businesses efficiently manage their sales processes and maximize revenue opportunities. The NPMS offers a range of features, including lead management, pipeline tracking, and sales forecasting, that enable businesses to manage their opportunities from initial contact to final sale. The existing system architecture and customized functions can no longer meet the management needs of the company's sustainable development. Therefore, a more complete opportunity management system needs to be re-developed. NPMS was created using spring boot [2][3] on the backend and VUE with a MySQL database on the frontend. Recruiters, clients, and other parties searching for opportunities may all benefit from the platform we have created. The opportunity, customer, and report management systems are the three key components of our platform. Anybody may visit the opportunity platform and see all available opportunities. Customers can change their data and other relevant information from the customers platform, and the report system works well for both systems. To draw more users to our site, we implement a high-quality design and use complementary colors that are pleasing to the eye. Our platform includes a genuine organization opportunities system, thus high security measures are very important for the customers to win the chances. This is why we arrived with high security measures.

【Keywords】NPMS, JAVA, spring boot, MySQL, Vue, Eureka Server

Table of Contents

List of Figures

List of Tables

List of Acronyms

UML	Unified Modeling Language
SQL	Structured Query Language
JVM	Java virtual machine
JDK	Java Development Kit
JDBC	Java Database Connectivity
GUI	Graphical User Interface
IEEE	Institute of Electrical and Electronics Engineering's
JSP	Java Server Page
FR	Functional Requirement
NFR	Non-Functional Requirement
UI	User Interface
Wi-Fi	Wireless Internet Free Internet
XAMPP	Cross-Platform (X), Apache (A), MySQL (M), PHP (P) and Perl (P)
XML	Extensible Markup Language

CHAPTER 1 INTRODUCTION

1.1 Overview

At present, the company's existing information management system consists of SAP (including: finance, personnel, distribution, project and other business functions), NeuSoft project management system, NeuSoft financial management system composition. SAP system has been in operation for many years, with the expansion of the company, the number of users increases, the system risk also increases, and affects the financial. Accounting and other core management business, and the existing system architecture and customer functions have been unable to meet the sustainable management needs of the company. The NPMS only includes opportunity and daily newspaper management, and the project and its related business management functions need to continue to be developed. Based on the above reasons, it is urgent to improve the company's information management system to comprehensively manage the overall business. Following are the areas covered in this **NPMS Project Management System-Opportunity Management.**

1. **Sign up**
   First of all, users have to sign up, then a unique id will be created of users
2. **login**
   User will login to dashboard panel and add menus, users, parameters, when admin will add these details then menus will appear and the user can access the System.
3. **Home Page**
   we designed a very attractive home page. Home page the first impression to our System. As we know that first impression is the last impression. So, we designed home page very cleaned and I have checked very minor details to make it decent and attractive to users.
4. **Customer Management**
   The customer management module includes customer management and handover.
5. **Opportunity management**
   The opportunity management module includes opportunity maintenance, approval and tracking. Opportunities refer to business opportunities brought by customers. For example, Huawei will launch a new product research and development plan recently. This is an opportunity for Huawei, a customer. As mentioned above, one customer can correspond to multiple opportunities.
6. **Report management**
   Daily report management includes employees' daily report filling, leader's daily report query and other related functions.

1.2 Objective and Success Criteria

The main objective of our project is to facilitate the users by providing the best possible Solution. Following areas are under focus to facilitate the users.

1. **Reduce management costs**
   The company will reduce the expenses caused for the advancement of their income.

2. **Developing business**

Relations Opportunity management system is a system used to help businesses improve their sales process. It helps companies identify, organize, and track opportunities in the sales pipeline. It can be used to create detailed reports and dashboards that allow managers to monitor the progress of each opportunity and make data-driven decisions about how best to move forward.

1. The NPMS system covers the whole process management of the project from opportunity to accounting.
2. The SAP system, with the financial, management and human resources of the accounting project as the core
3. Unified the main data of each system and improve the process management of the system
4. Among them, NPMS is one of the key systems of the company's information construction projects, responsible for the comprehensive management of the overall process from the opportunity to the project, and the specific target package

The following contents are included:

1. Sales opportunities can be tracked, customer information maintenance and management.
2. Control the pre-sale work and the bidding process
3. Project profit calculation and contract review
4. Project planning, cost, progress, payment collection and other tracking management
5. Improve the company's business approval process
6. Collect and review documents such as company contracts and achievements

Provide basic information for company operations, financial and other system

1.3 Interfaces

1.3.1 Users Interface

The home page of this system is to control the overall process of the project, covering opportunity management, pre-sales management, contract management and project management at this stage. Opportunity management includes functions from customer creation, opportunity maintenance, opportunity tracking, customer handover; pre-sales management covers pre-sale resource application and allocation, bidding and payment application; contract management includes workload assessment, profit calculation, contract review, etc.; project management includes project resource application, project daily report,

1.4 Terms

Table 1 System Terms and Condition

Numbers	System term description
1.	**Customers: the groups that meet their certain needs through the purchase of enterprise products or services, that is, the individuals or enterprises that have direct economic relations with individuals or enterprises;**
2.	**In the project management system, one customer only belongs to one account manager, and other account managers or sales directors cannot operate on the customer, and it cannot be based on other customers**
3.	**Opportunity: also known as a business opportunity to generate profit; the opportunity is created based on the customer and can only be maintained by the account manager to which the customer belongs**
4.	**Sales department: account manager department, including marketing system, consulting system and technical system**
5.	**Customer transfer: After the change of the account manager personnel, it is necessary to transfer the customer manager's customers and opportunities to other account managers, and this operation becomes the customer transfer. The customer handover shall be performed between the outgoing party and the receiver party**
6.	**Customer owner: customer maintenance personnel, a customer can only be maintained by a customer manager, generally for the customer creator, or may be after the handover of the customer the receiver**
7.	**Opportunity owner: Opportunity maintenance and tracking personnel, an opportunity can only be maintained by one customer, generally for the opportunity creator, or may be the customer handover** **After the receiver**
8.	**Login to: the current user of the system**
9.	**Use case: An abstract description of a set of action sequences executed by the system to produce corresponding results. Functional requirements can be refined by the use case** **description**
10.	**Use case level: Defines the level of functional requirements, mainly guiding development prioritization**
11.	**Occurrence frequency: the frequency of use case scenarios to guide development and testing**
12.	**Use role: various types of people in the use case scenario**
13.	**Participating role: all types of people who view or review the use case scenario data**
14.	**Precondition: necessary factors for use case scenario initiation** **term description**
15.	**Post-condition: the subsequent scene processing triggered after the use case scene is executed**
16.	**Submit: Operation button, click to initiate the approval process**
17.	**Flow chart: operate the button, you can view the process flow legend when connecting the approval process**
18.	**Login to: the current user of the system**
19.	**Retrieval: also known as query, data filtering, according to the query conditions provided**

CHAPTER 2 ANALYSIS OF ACTIVITIES

2.1 Internship timetable

we had different teachers, (in class or via online) and tasks (techniques and Environment) during the internship. the training timetable is shown in **Table 3 Training timetable** that consists of different phases during the internship between the training and programming and the tasks performed with a specific objective for each phase. The schedule also contains information on how one will accomplish those objectives. Each phase builds upon the previous phase to show a progression in the training/internship.

Table 2 Timetable for entire Training

	START	END	TOPIC	WORK
WEEK 1	05-09-2022	10-09-2022	Training	JAVA Web, Servlet
WEEK 2	13-09-2022	18-09-2022	Training	JAVA Web, Servlet
WEEK 3	25-9-2020	2-10-2022	Training	ES6
WEEK 4	21-10-2022	28-10-2022	Training	MVC, MySQL Database
WEEK 5	03-11-2022	10-11-2022	Programing	Applying what we have learned
WEEK 6	17-11-2022	24-11-2022	Training	Building the first VueJs page
WEEK 7	26-1-2022	08-12-2022	Training	Working with Spring Boot
WEEK 8	26-1-2022	08-12-2022	programing	Building the first Web Spring boot
WEEK 9	11-12-2022	16-12-2022	Training	Setup the working environment, design DB
WEEK 10	11-12-2022	16-12-2022	Programing	Prepare the working environment, design DB
WEEK 11	19-12-2022	24-12-2022	Training	building the main structure for the backend, DB entity, Models, Services
WEEK 12	27-12-2022	01-01-2023	Training	Working with Redis and Shiro
WEEK 13	27-12-2022	01-01-2023	Training	Working with MyBatis, DBeaver
WEEK 14	27-12-2022	01-01-2023	Training	Applying MyBatis to the Spring Boot
WEEK 15	27-12-2022	01-01-2023	Review	Back to VueJs
WEEK 16	04-01-2023	08-01-2021	Training	new tools were used in Vue
WEEK 17	11-01-2023	15-01-2023	Training	building the front-end integrated it with back-end
WEEK 18	18-01-2023	22-01-2023	Training	Generate all the back-end files using the database entities and MyBatis Generator.
WEEK 19	25-01-2023	29-01-2023	Training + review	testing the api functions using postman
WEEK 20	01-02-2023	05-02-2023	Training	configuration the Security adapter
WEEK 21	08-02-2023	12-02-2023	Programing	Prepare the Front-end and the Back-end for our final Project
WEEK 22	15-02-2023	19-02-2023	Programing	Working with our final Project
WEEK 23	22-02-	26-02-	Training	Building the Dynamic Router for

	2023	2023		our Menus
WEEK 24	01-03-2023	05-03-2023	Programing	Working with our final Project
WEEK 25	08-03-2023	12-03-2023	Programing	Working with our final Project
WEEK 26	15-03-2023	19-03-2023	Programing	Working with our final Project
WEEK 27	**14-03-2023**		Defense	Final Project defense

2.2 The Technologies

As this internship was taking part, we have the opportunity to work in a project as a team and we used what we learned during the training lessons to build our project.

The project was divided into two section front-end and back-end. We learn during our internship many famous technologies and the list below is showing these technologies we used during our internship classes:

Table 3 List of the techniques

Course	Document
Java Web	A servlet is a Java programming language class used to extend the capabilities of servers that host applications accessed by means of a request-response programming model. Although servlets can respond to any type of request, they are commonly used to extend the applications hosted by web servers. For such applications, Java Servlet technology defines HTTP-specific servlet classes. (i)
	ECMAScript (ES) is a standardized scripting language for JavaScript (JS). The current ES version supported in modern browsers is ES5. However, ES6 tackles a lot of the limitations of the core language, making it easier for developers to code. (ii)
	Vue (pronounced /vju:/, like view) is a progressive framework for building user interfaces. Unlike other monolithic frameworks, Vue is designed from the ground up to be incrementally adoptable. The core library is focused on the view layer only, and is easy to pick up and integrate with other libraries or existing projects. On the other hand, Vue is also perfectly capable of powering sophisticated Single-Page Applications when used in combination with modern tooling and supporting libraries. (iii)
	Spring makes programming Java quicker, easier, and safer for everybody. Spring's focus on speed, simplicity, and productivity has made it the world's most popular Java framework. (iv)
	Spring Web MVC is the original web framework built on the Servlet API and has been included in the Spring Framework from the very beginning. The formal name, "Spring Web MVC," comes from the name of its source module (spring-webmvc), but it is more commonly known as "Spring MVC". (v)

MyBatis	MyBatis is a first-class persistence framework with support for custom SQL, stored procedures and advanced mappings. MyBatis eliminates almost all of the JDBC code and manual setting of parameters and retrieval of results. MyBatis can use simple XML or Annotations for configuration and map primitives, Map interfaces and Java POJOs (Plain Old Java Objects) to database records. (vi)
My Batis plus	MyBatis-Plus is powerful enhanced toolkit of MyBatis for simplify development. This toolkit provides some efficient, useful, out-of-the-box features for MyBatis, use it can effectively save your development time. (vii)
Spring Boot	Spring Boot offers a fast way to build applications. It looks at your classpath and at the beans you have configured, makes reasonable assumptions about what you are missing, and adds those items. With Spring Boot, you can focus more on business features and less on infrastructure. (viii)
redis	Redis is an open source (BSD licensed), in-memory data structure store, used as a database, cache, and message broker. Redis provides data structures such as strings, hashes, lists, sets, sorted sets with range queries, bitmaps, hyperlog logs, geospatial indexes, and streams. Redis has built-in replication, Lua scripting, LRU eviction, transactions, and different levels of on-disk persistence, and provides high availability via Redis Sentinel and automatic partitioning with Redis Cluster. (ix)

1. Back-end: that manage the retrieving and manipulations by using MyBatis plus that has the basic functions of persistent storage of the data while processing the request of the end-user.
2. Front-end: which care about how the data appeared to the end-user after it arrived from the back-end.

For our project the system contains the following modules:

1) Security Module:
    - Manage the login process
    - Manage the sessions
    - Mange the access control

Chapter 3 System Analysis and Design

System analysis is the most challenging phase in the system life cycle. It used for breaking up the system into parts to identify its attribute, functions, and objectives. By studying, the environments of the system will be implemented to give a clear view and good understanding of how to build the system and to choose the right tools and framework for that.

As we had to part in the internship that allows us to work in different techniques and environment. However, all of these techniques serve the same purpose which it designs any system and the same programming paradigms and software design patterns.

3.1 Project developing architecture

This platform used as web application services and for that we use MVC software design pattern that separates an application into three components that have communication between them to complete the request

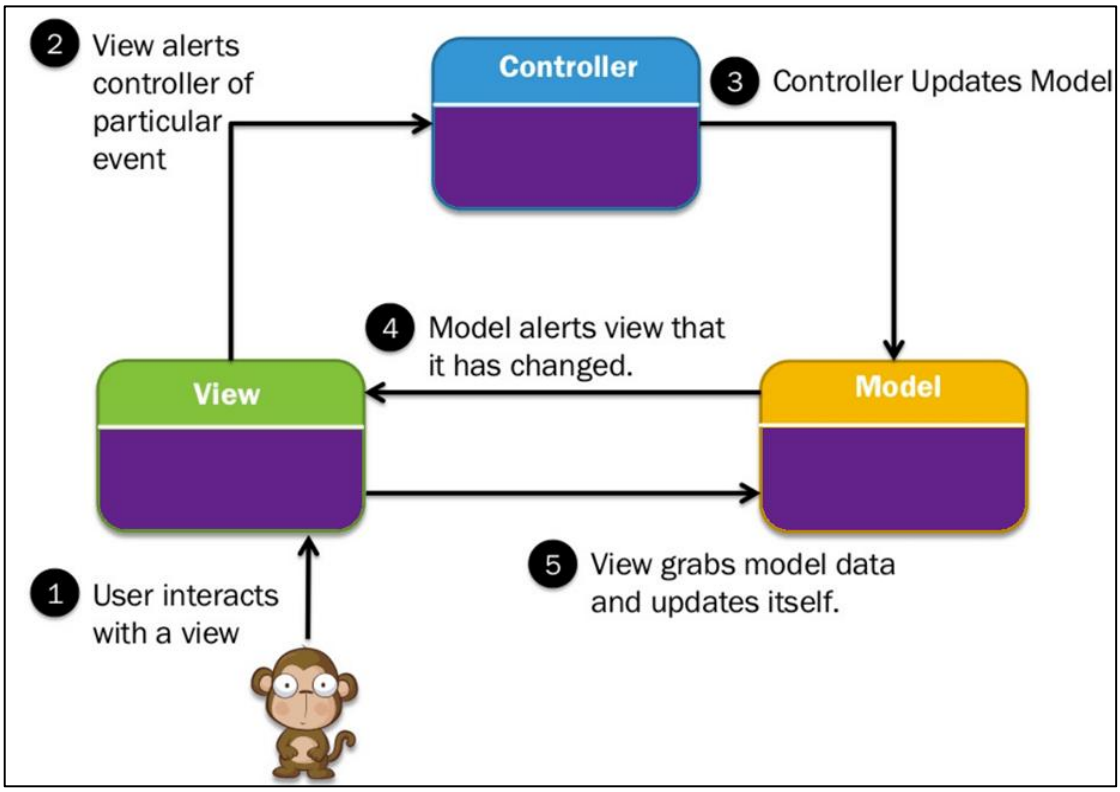


Figure 1 MVC Software design pattern

- **The Model-View-Controller (MVC)** framework is an architectural pattern that separates an application into three main logical components Model, View, and Controller. Hence the abbreviation MVC. Each architecture component is built to handle specific development aspect of an application. MVC separates the business logic and presentation layer from each other. It was traditionally used for desktop graphical user interfaces (GUIs). Nowadays, MVC architecture has become popular for designing web applications as well as mobile apps.

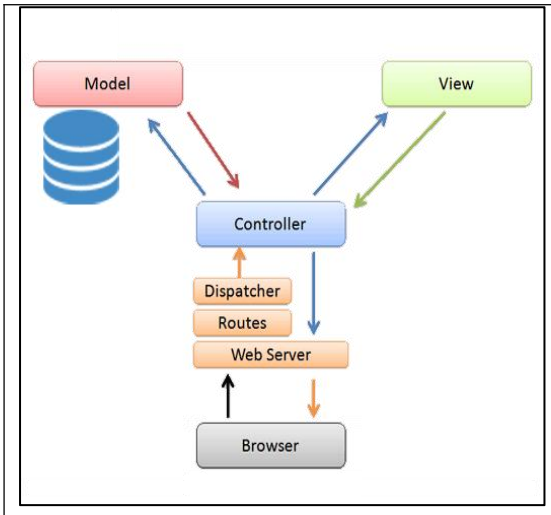


Figure 2 MVC connected to the Internet

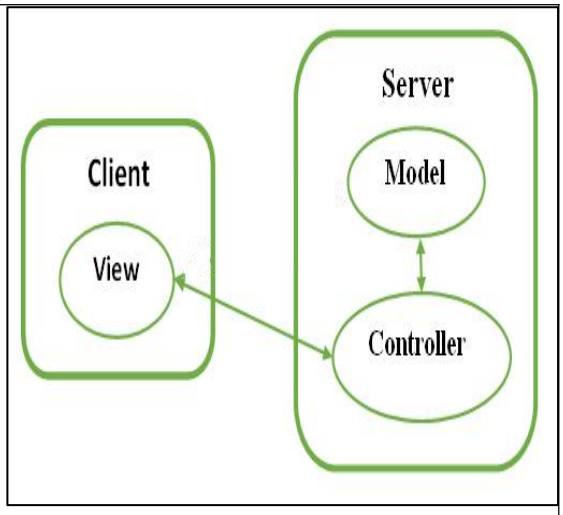


Figure 3 MVC as Client/Server Diagram

- **Spring Boot**: Spring Boot makes it easy to quickly bootstrap and start developing a Spring-based application. It avoids a lot of boilerplate code. It hides a lot of complexity behind the scene so that the developer can quickly get started and develop Spring-based applications easily.
- **Spring MVC**: Spring MVC is a Web MVC Framework for building web applications. It contains a lot of configuration files for various capabilities. It is an HTTP oriented web application development framework.

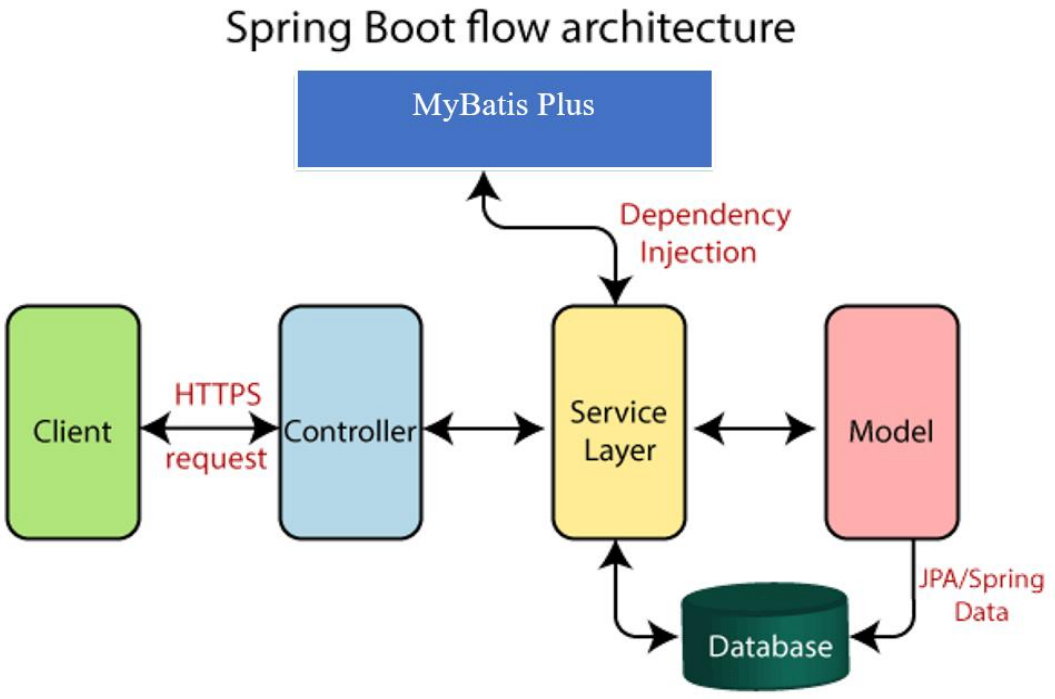


Figure 4 Spring Boot Flow Architecture

- In the development Process, several models can be used which includes the classical Waterfall, the Agile and Reuse-oriented software engineering etc.

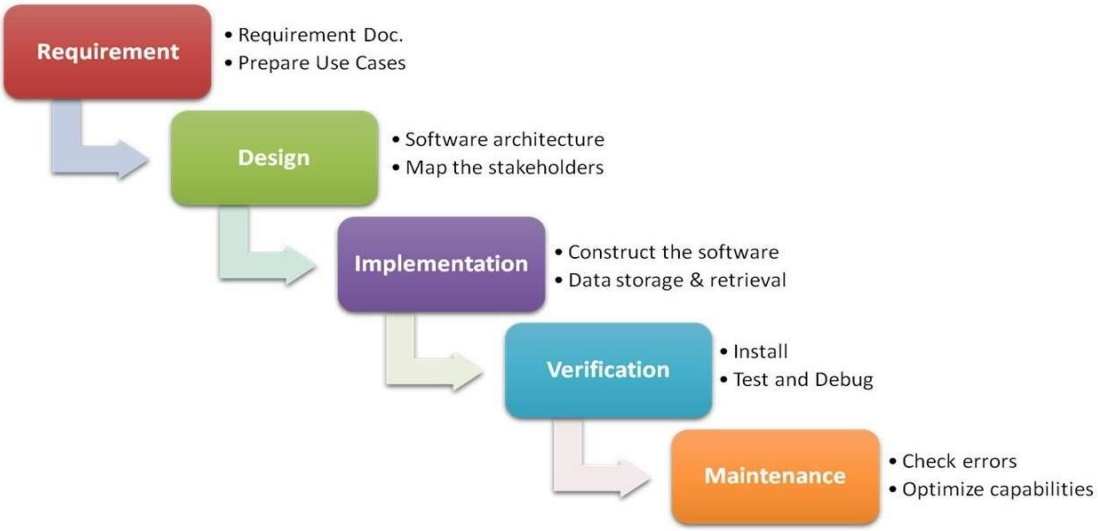


Figure 5 Software Development process (waterFull Model)

3.2 System implementation

3.2.1 Customer Management

1. When adding a new customer, I will realize an auto increment by myself, which is select the max customer id from the table and add one.
2. The customer association cannot be associated with multi-level parent nodes. If A->B, B->C then A can't -> C.
   If the current customer is D and A and B are added as associated customers, then see if you can find A C in the table where A->C and B->C (add, modify)
   If the current customer is D and A and B are added as associated customers, then check the table for A->B (add, modify)
   If the current customer is D, and if A is added as an associated customer, then see if you can find an E in the table where A->E and D->E (modify)
3. Using transactions for unified commit or rollback by adding @Transactional (rollback For=Exception. Class)

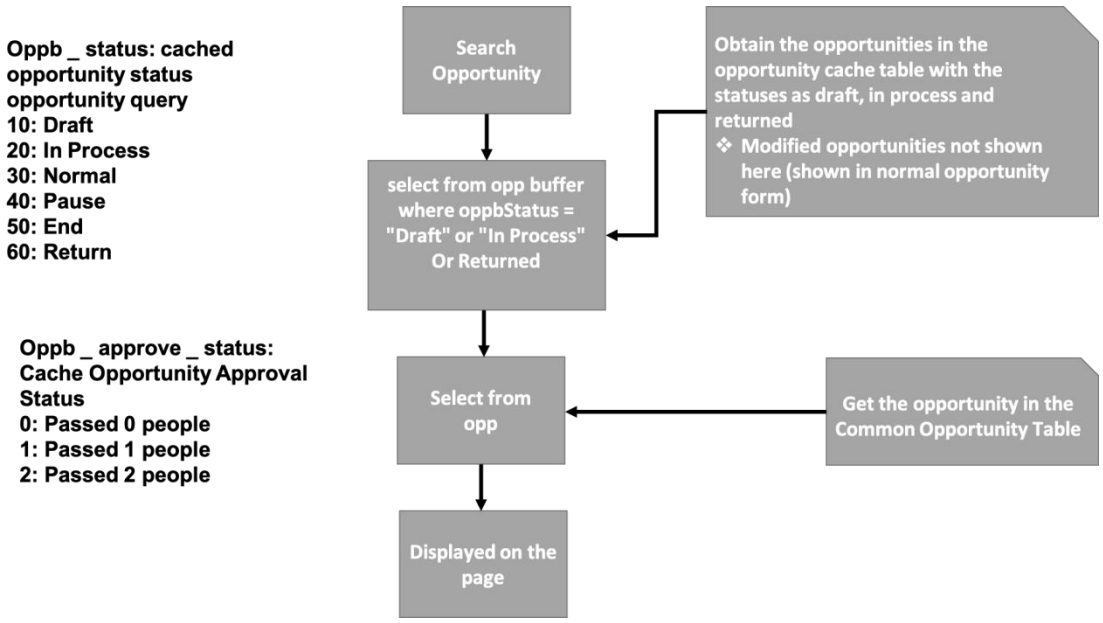


Figure 6 querying opportunity Process

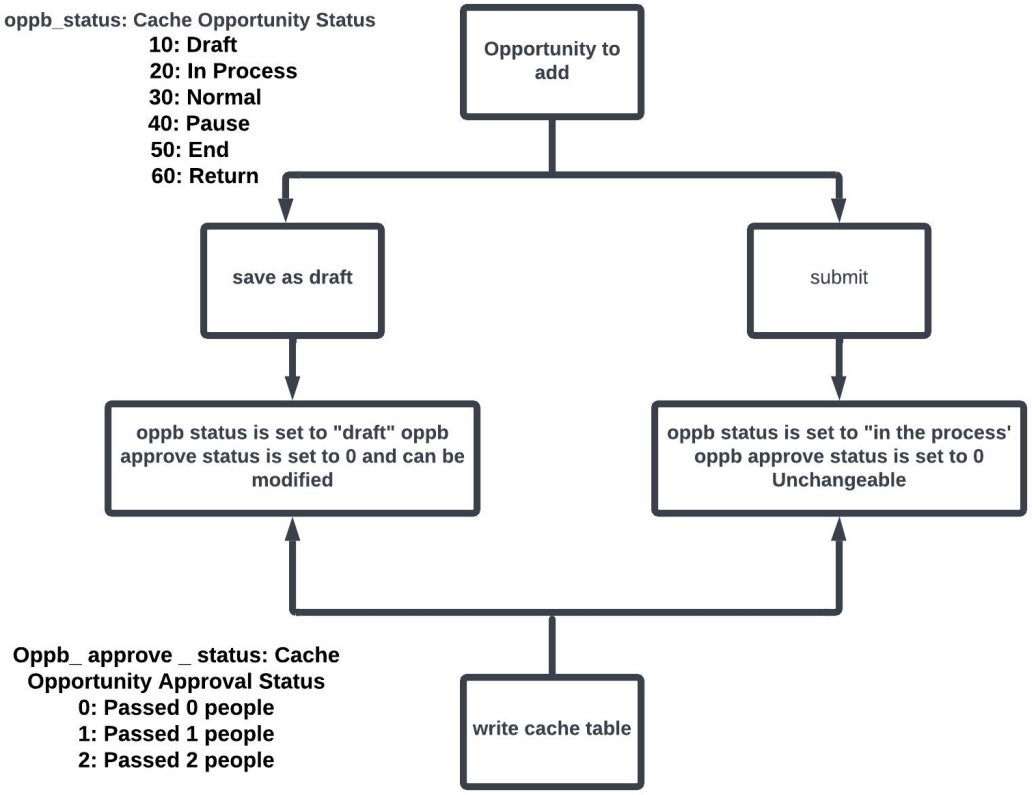
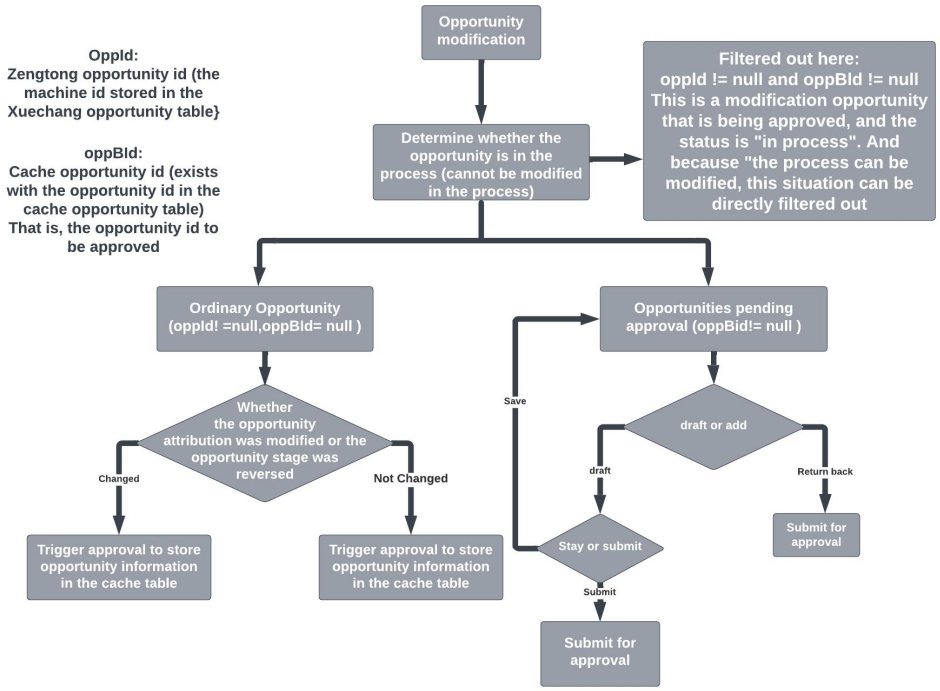


Figure 7 Adding opportunity



3.3

Figure 8 Updating opportunity

Approval

The four tables are created in order to store the inserted or updated opportunity information. They are opportunity buffer, sub_opportunity_buffer, competitor buffer and payer_buffer. These four tables are corresponding to the original four tables, which are opportunity, sub opportunity, competitor, and payer. In this way, I can store the new data in the buffer. When the approval is pass, we will transfer the data from the buffer to the normal table.

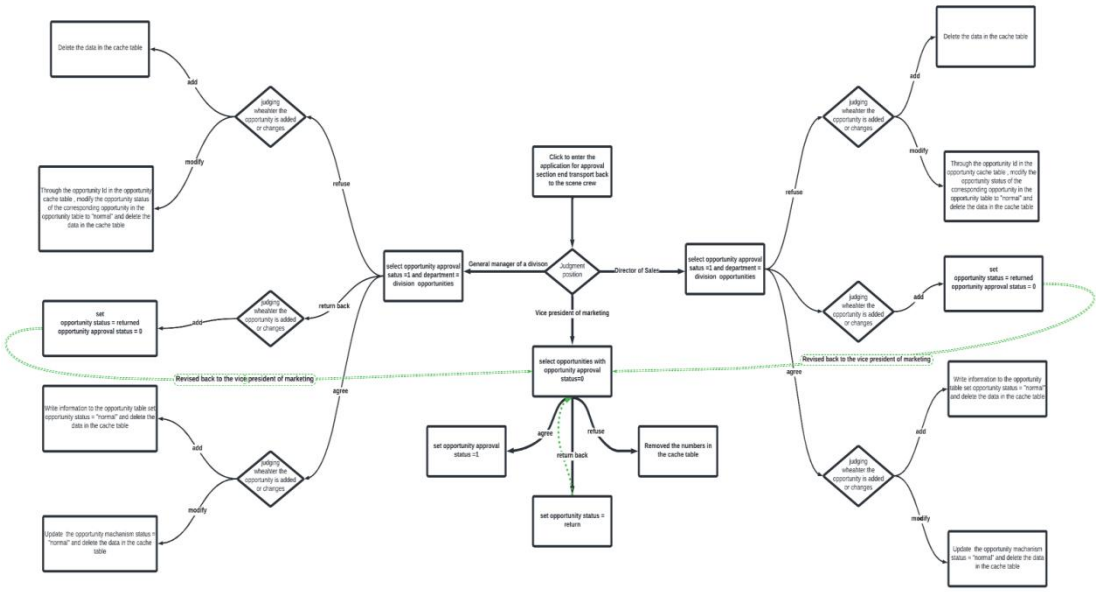


Figure 9 Approval flowchart

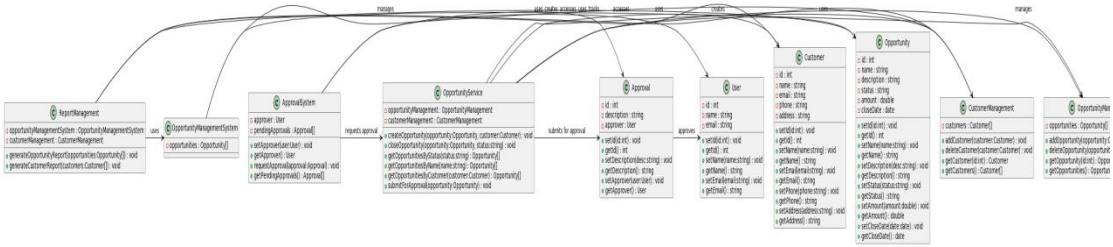


Figure 10 Class diagram

In this diagram, a Approval System class has been added to handle the approval process for opportunity creation. Approval System uses User and Approval objects to track and approve the creation of new opportunities. Opportunity Service now also includes the ability to submit opportunities for approval using submitForApproval. Note that Opportunity Service continues to use Opportunity Management and Customer Management for creating and managing opportunities.

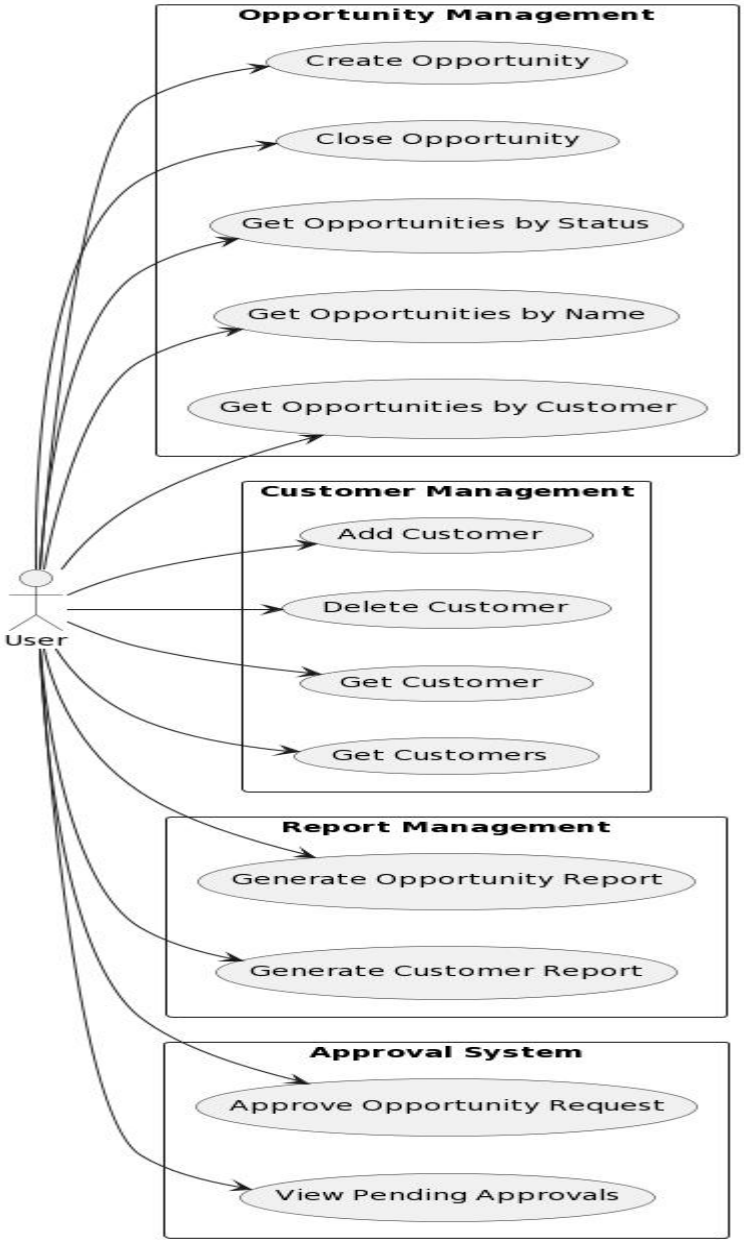


Figure 11 Use Case Diagram

In this diagram, is the actor that interacts with the system. The rectangle "Opportunity Management" includes use cases for creating opportunities, closing opportunities, and retrieving opportunities by various criteria. The rectangle "Customer Management" includes use cases for adding, deleting, and retrieving customers. The rectangle "Report Management" includes use cases for generating opportunity and customer reports. The rectangle "Approval System" includes use cases for approving opportunity requests and viewing pending approvals.

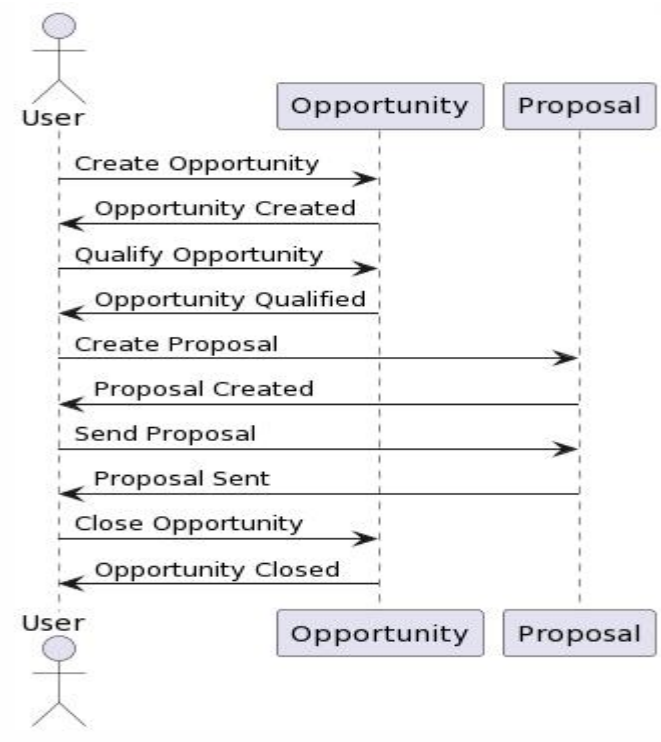


Figure 12 Sequence diagram

Opportunity Management System Sequence Diagram. This sequence diagram shows the interaction between the user, opportunity, and proposal objects. It starts with the user creating a new opportunity, which is then qualified. The user then creates and sends a proposal, and finally, the opportunity is closed.

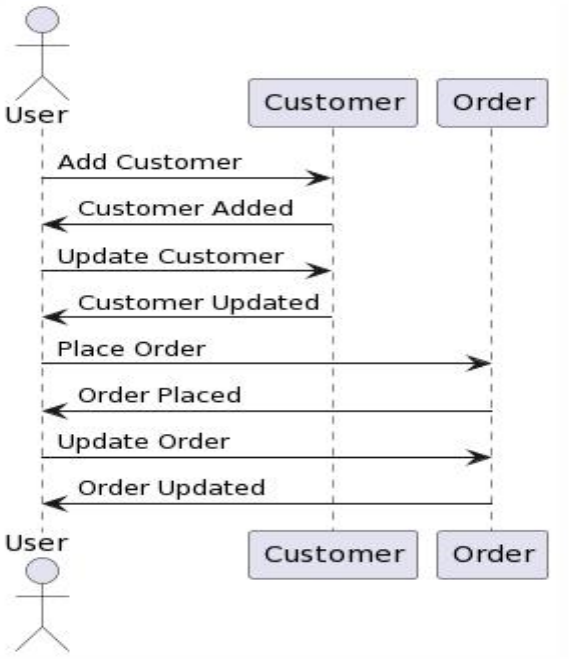


Figure 13 Flow Chart Diagram

This sequence diagram shows the interaction between the user, customer, and order objects. It starts with the user adding or updating a customer profile and then placing or updating an order.

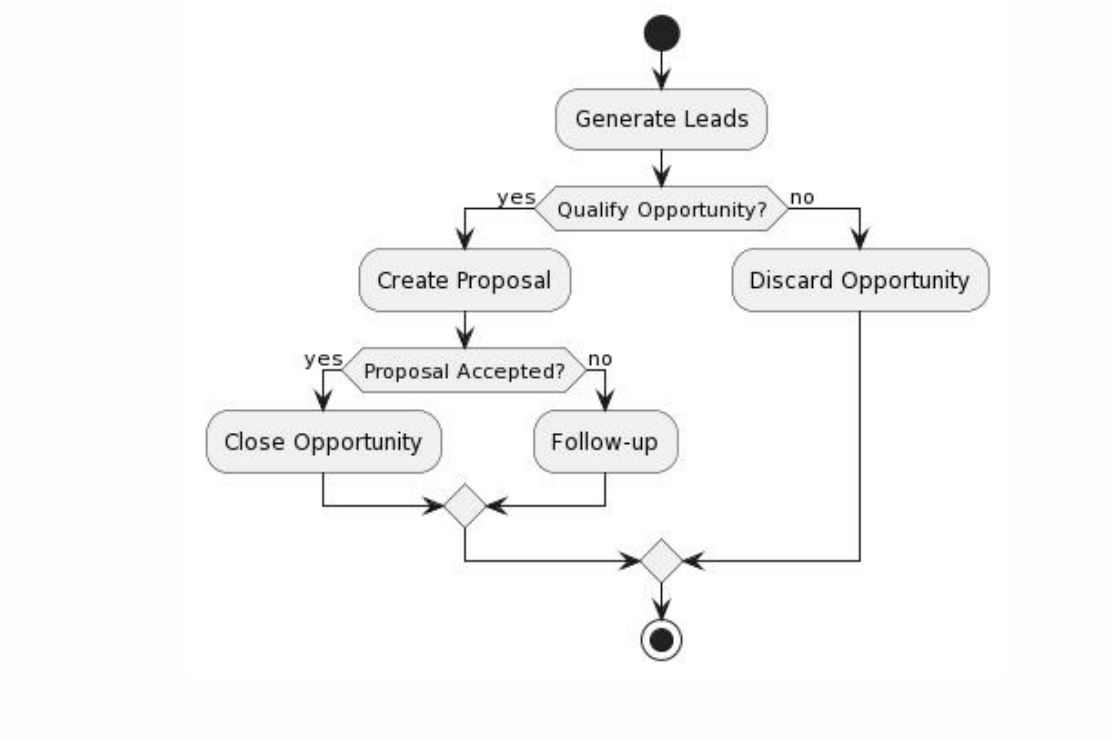


Figure 14 Flow chart diagram of opportunity management

The above flowchart that depicts the opportunity management system's different steps, including lead generation, opportunity qualification, proposal creation, and opportunity closure.

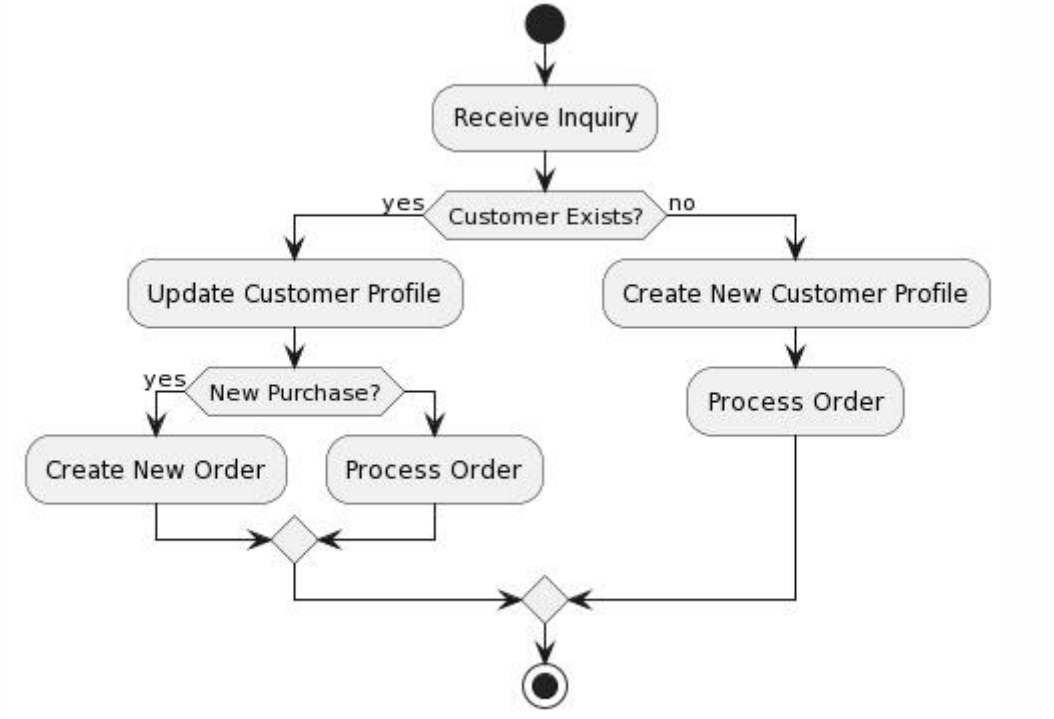


Figure 15 Flowchart diagram customer management

Customer management process, including receiving inquiries, updating customer profile, new purchases or processing of orders, and creating new customer profiles.

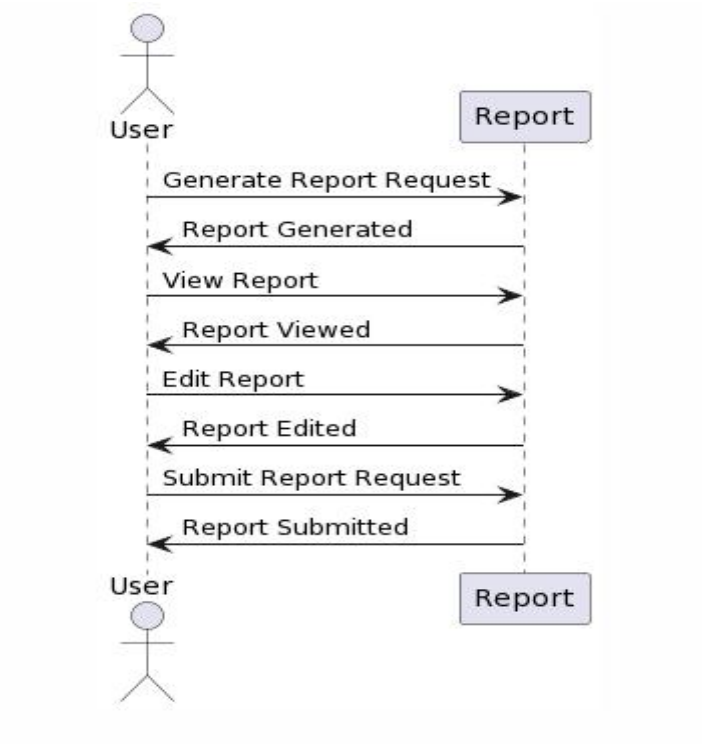


Figure 16 Sequence diagram

This sequence diagram shows the interaction between the user and the report object. It starts with the user generating a report request, which is then generated and viewed. The user can then edit or submit the report request, and the report will be edited or submitted accordingly.

3.1.1 ERD Diagram

Figure 17 Shows ERD Diagram of our Platform:
An entity relationship model depicts interrelated things of interest in a particular space of information. An essential ER model is made out of element types and indicates connections that can exist between substances. Entity Relationship Diagram, also known as ERD, ER Diagram or ER model, is a type of structural diagram for use in database design. An ERD contains different symbols and connectors that visualize two important information: The major entities within the system scope, and the inter-relationships among these entities. And that's why it's called "Entity" "Relationship" diagram (ERD)!
When we talk about entities in ERD, very often we are referring to business objects such as people/roles (e.g., Student), tangible business objects (e.g., Product), intangible business objects (`e.g., Log), etc. "Relationship" is about how these entities relate to each other within the system.

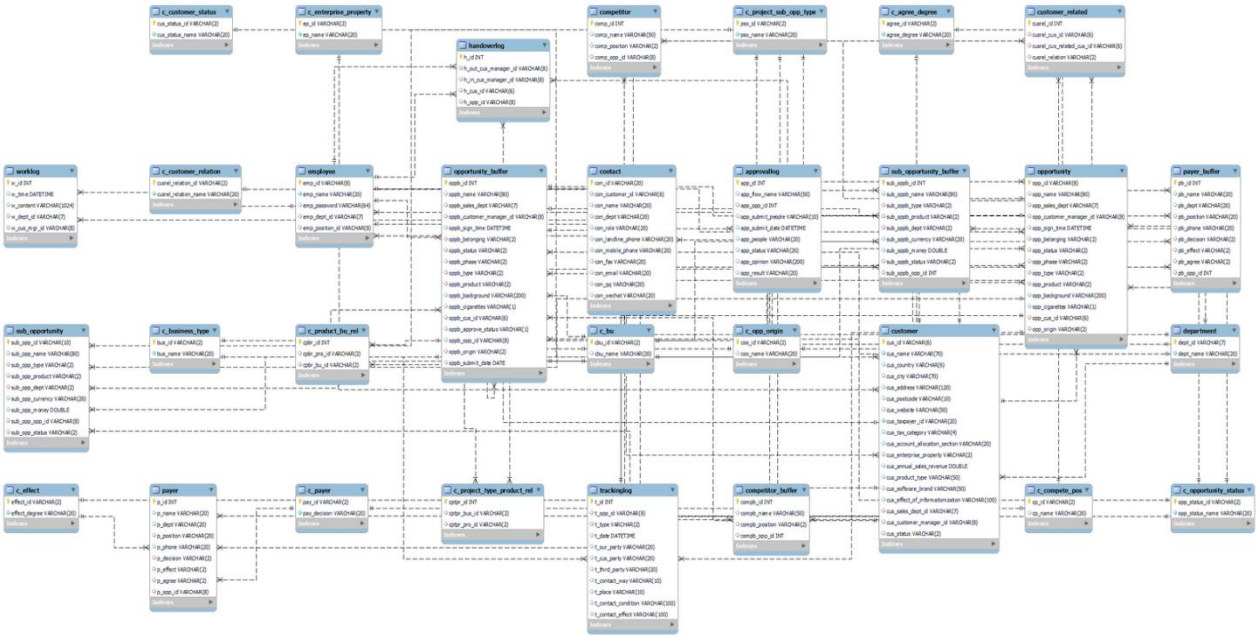


Figure 17 ER Diagram

.

The ER diagram for the Opportunity Management System, Customer Management, and Report Management with an Approval System can be explained in the following way:

Opportunity Management System: The system manages business opportunities for a company. It consists of entities such as Opportunity, Account, Contact, and Activities. The Opportunity entity represents a potential sale or deal, while Account represents a company or organization that the opportunity belongs to, and Contact represents a person associated with the account. Activities include tasks and events related to the opportunity.

Customer Management: This system manages the customer data and interactions. It consists of entities such as Customer, Orders, Products, and Sales Representatives. The Customer entity represents a company or individual who has purchased a product, while Orders represent the transactions made by the customer. Products entity represents the products sold by the company, and Sales Representative entity represents the company's employees responsible for sales.

Report Management: The system manages reports generated by different systems. It consists of entities such as Report, User, and Report Type. The Report entity represents the generated report, while User represents the user who requested the report. The Report Type entity represents the different types of reports generated by the system.

Approval System: This system handles the approval process for various tasks and reports. It consists of entities such as Request, Approver, and Status. The Request entity represents the task or report that requires approval, while Approver represents the person responsible for approving the request. The Status entity represents the status of the request, whether it is pending, approved, or rejected.

In conclusion, the ER diagram for the Opportunity Management System, Customer Management, and Report Management with an Approval System captures the entities

and relationships between the various systems. It provides a comprehensive view of the data flow and relationships between different entities, enabling effective management of the operations of the company.

Chapter 4 NeuSoft Opportunity Project Management System

4.1 Software Operating Environments

- The Web NPMS:
  - Will be to operate in Window, Vista, Window7,Window8/8.1 and Window 10.
  - Is compatible with these Web browsers:
    - ⇒ Microsoft Internet Explorer Version 5.0 and onwards
    - ⇒ Google chrome version 5 and onwards
    - ⇒ Mozilla Firefox version 5 and onwards

  Will be accessible on different platforms:
  - Database Technology Stack
    - ⇒ Xampp Control panel Version 2.5.8
    - ⇒ Xampp for Windows Version 1.7.2
    - ⇒ MySQL Workbench Version 8.0.22
  - Backend Technology Stack
    - ⇒ IdeaIU
    - ⇒ MyBatis
    - ⇒ Spring Boot
    - ⇒ Spring Security
    - ⇒ Web Stock
    - ⇒ JDK Version 8.0.1
    - ⇒ Redis Version 7.0.8
  - Frontend Technology Stack
    - ⇒ VUE
    - ⇒ Element-UI
    - ⇒ Visual Studio Code
    - ⇒ Axios
    - ⇒ Vue-Router
    - ⇒ VUEx
    - ⇒ VUE-cli4

Eureka Server is a component of the Spring Cloud framework that provides a service registry for microservices applications. It is responsible for maintaining a registry of all the running instances of microservices and their availability to clients.

Eureka Server is used in the following ways:

Service registration: Microservices register themselves with the Eureka Server during their initialization process so that other microservices and clients can discover and interact with them.

Service discovery: Clients of microservices in the system can use the Eureka Server to locate the running instances of the microservices.

Eureka Server uses the REST APIs to communicate with microservices and clients in the system. It provides a web-based UI for visualizing the state of the registry, which can be useful in debugging and monitoring applications.

Eureka Server has the following features:

High availability: Eureka Server can be run in a cluster mode so that the registry can still be accessed even if one or more servers fail.

Load balancing: Eureka Server provides load balancing for clients by returning a list of available instances of a service.

Self-registration: Eureka Server can also register itself as a service with other Eureka servers in the same cluster.

Customizable: Eureka Server can be customized to work in different environments, including private networks and public clouds.

4.2 Database

We have used MySQL database in our project. MySQL is an open-source relational database management system. Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.

- Workbench

  MySQL Workbench is a visual database design tool that integrates SQL development, administration, database design, creation and maintenance into a single integrated development environment for the MySQL database system

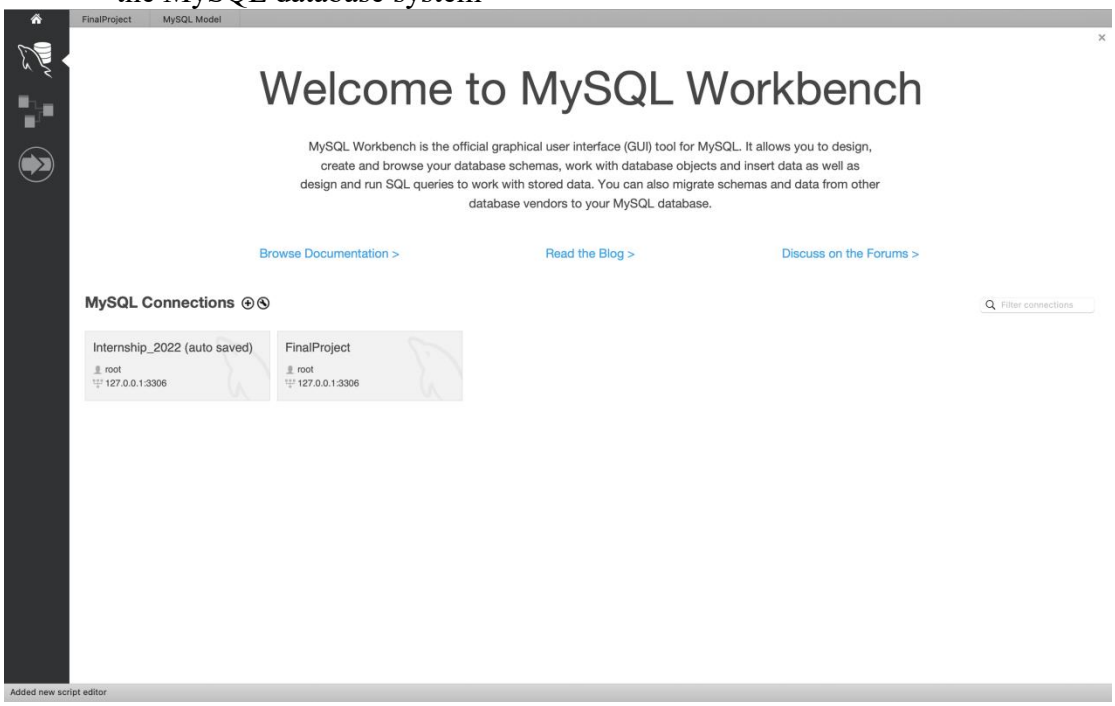


Figure 18 Workbench

Approvallog Database Table

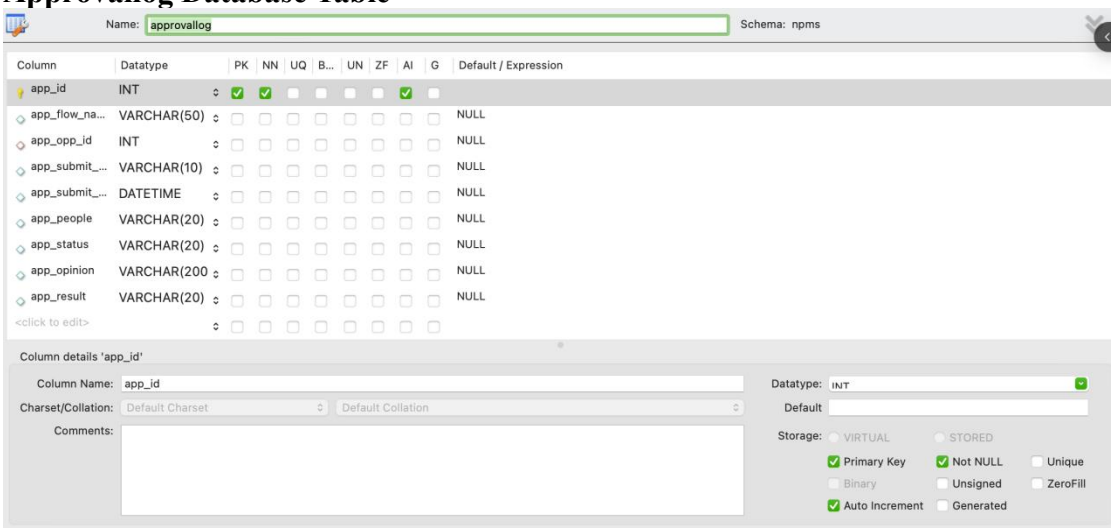


Table 4 Approvallog Database

Opportunity Database Table

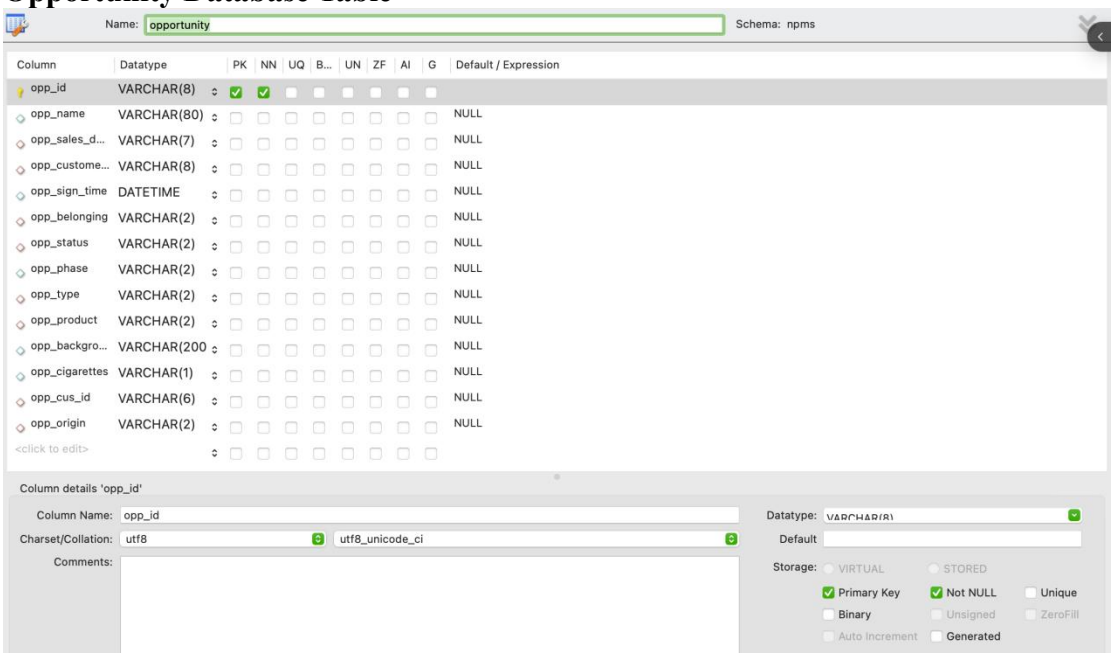


Table 5 Opportunity Database

Customer Database Table

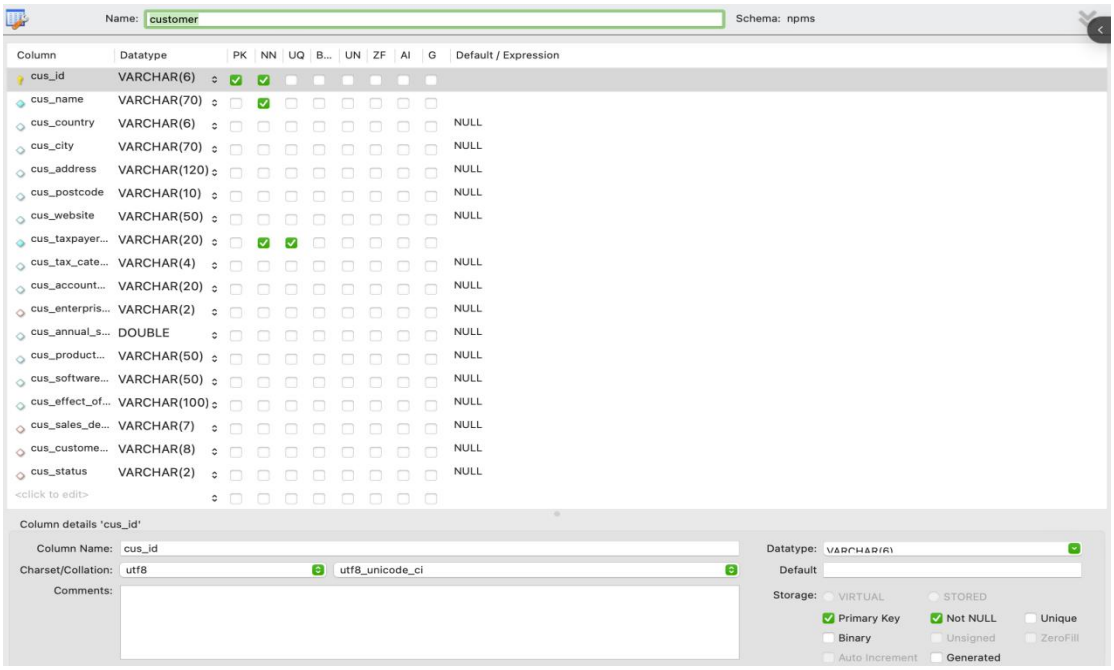


Table 6 Customer Database

Department Database Table

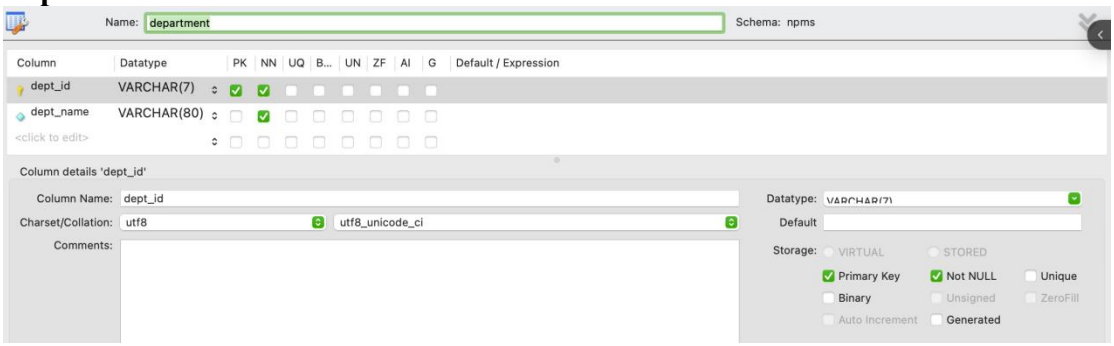


Table 7 Department Database

Employee Database Table

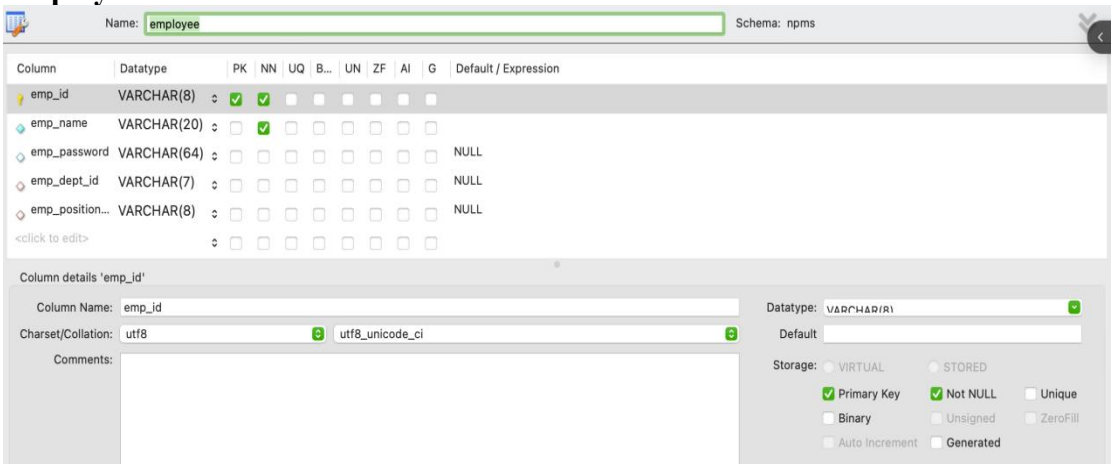


Table 8 Employee Database

Customer_Position Database Table

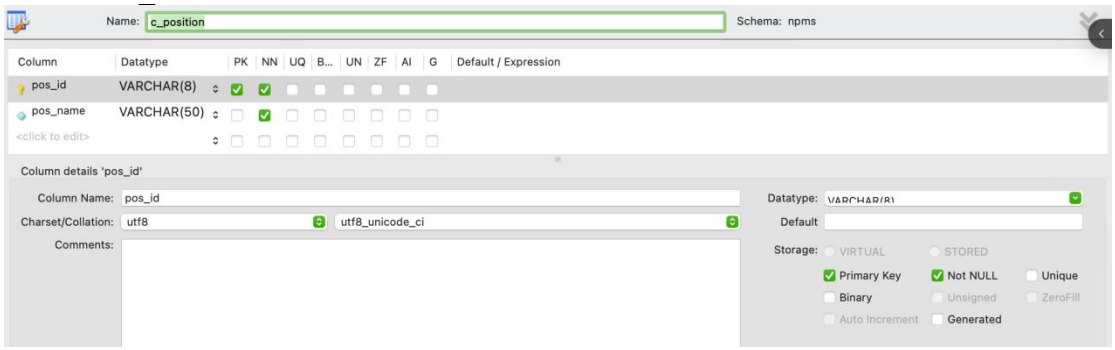


Table 9 Customer Position Database

Competitor Database Table

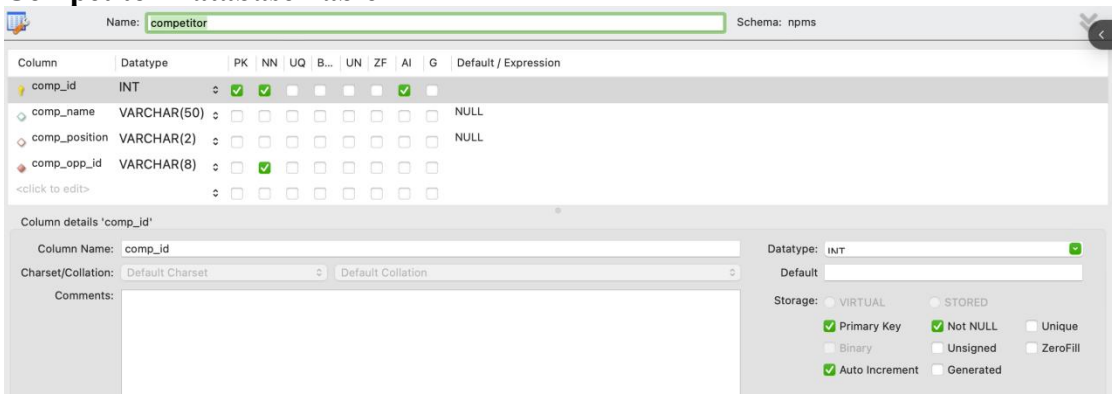


Table 10 Competitor Database

Sub_Opportunity Database Table

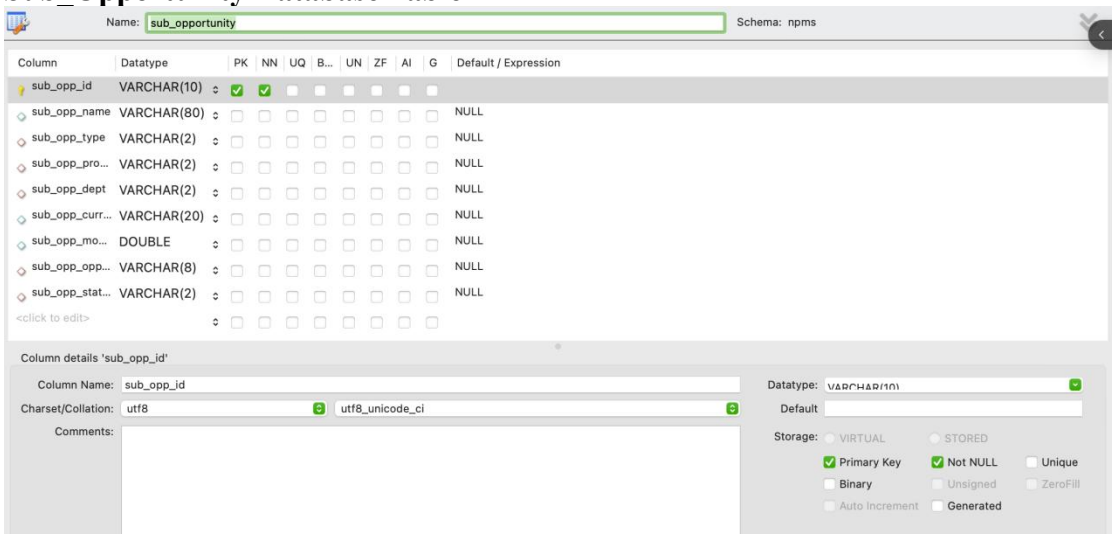


Table 11 Sub_Opportunity

TrackingLog Database Table

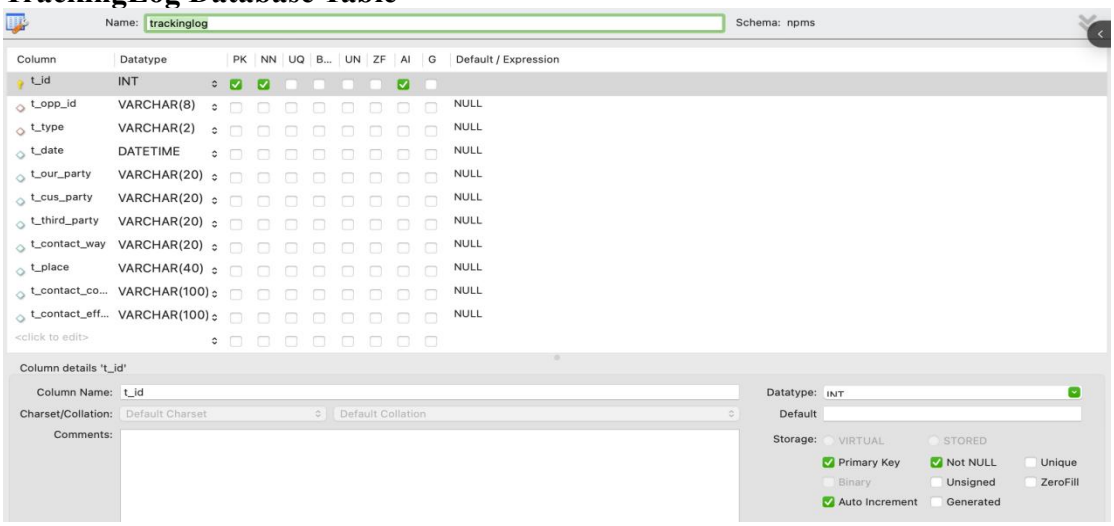


Table 12 TrackingLog Database

Payer Database Table

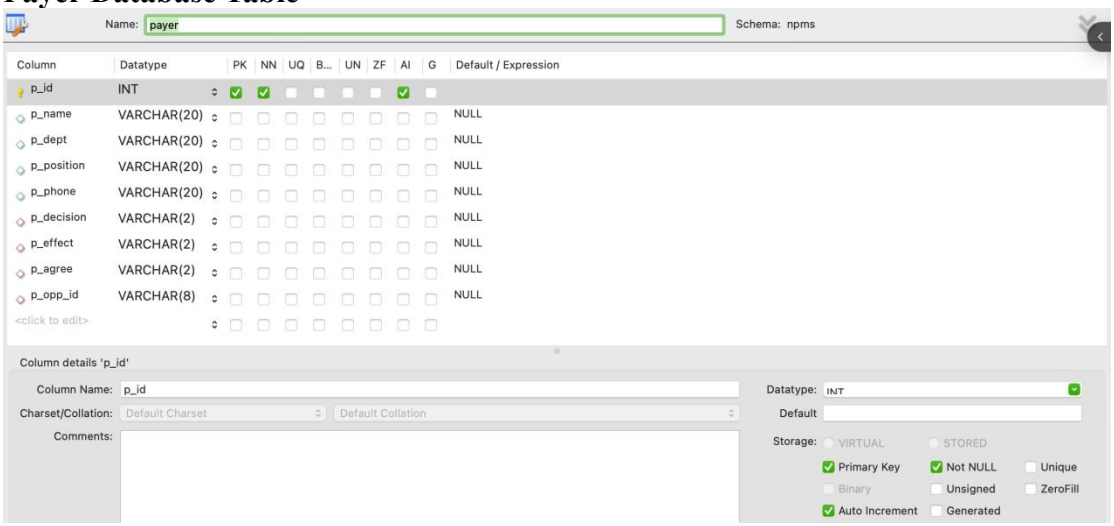


Table 13 Payer Database

Contact Database Table

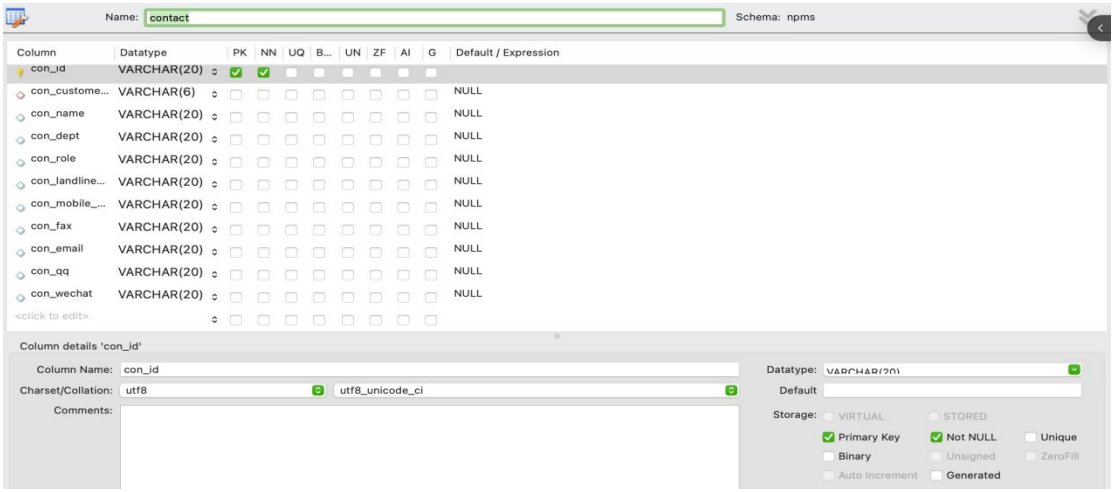


Table 14 Contact Database

4.3 Backend

My work (2128025)

I am responsible for the back-end development of the system. First clarify the requirements, design ER diagrams, class diagrams, sequence diagrams and other related models. Then, referring to these diagrams, 33 tables of customers, employees, departments, opportunities, handover logs, etc. were designed. Use MySQL to complete the creation of the database. Use mybatis as the persistence layer framework, and use mybatis-plus to automatically generate controller, service, and mapper.

After that, I participated in the preparation of all the back-end API documents. Among them, the customer management module has 18 APIs, including the addition, deletion and modification of customers, customer handover, customer freezing, and many secondary linkages and other page functions. The opportunity management module has 19 APIs, including opportunity addition, deletion, modification, inspection, opportunity tracking, opportunity approval, and so on. The daily report management module has 5 APIs, including daily report filling, daily report query and so on.

I use spring cloud when developing, and treat the above three modules, customer management, opportunity management, and daily report management as three independent services, in other words, consumers. The gateway is used for access, and the authorization authentication uses oath2 and spring-security. The entire service includes three eureka servers, one gateway, one oath2-auth and three consumers. I also used Redis as a cache database and published the project on the Alibaba Cloud server.

My innovation point (2128025)

In opportunity approval, multiple approvals by different leaders are involved, and the order of approval is determined. Only after the first approver passes the process will the process proceed to the next person for approval. In addition, the approval results have three results: “pass, reject, and reject the modification”. In order to handle this business, I designed a number of temporary storage tables (buffer) to store the “opportunity” information pending approval. In this way, the original “opportunities” table cannot be modified, making the entire approval process safer for database operations.
We utilized the Spring Boot framework, MyBatis, JDK, IntelliJ IDEA, and some functionality for our project. This is a brief introduction to each of them.
You may create graphical user interfaces (GUI) for your applications by using Swing library components with the GUI Designer in IntelliJ IDEA[10].

An open-source Java-based framework called SpringBoot[1][4] is utilized to create a little Service. It was developed by the Pivotal Team and is used to put together standalone and ready-made spring apps. This section introduces you to Spring Boot and familiarizes you with its key concepts.

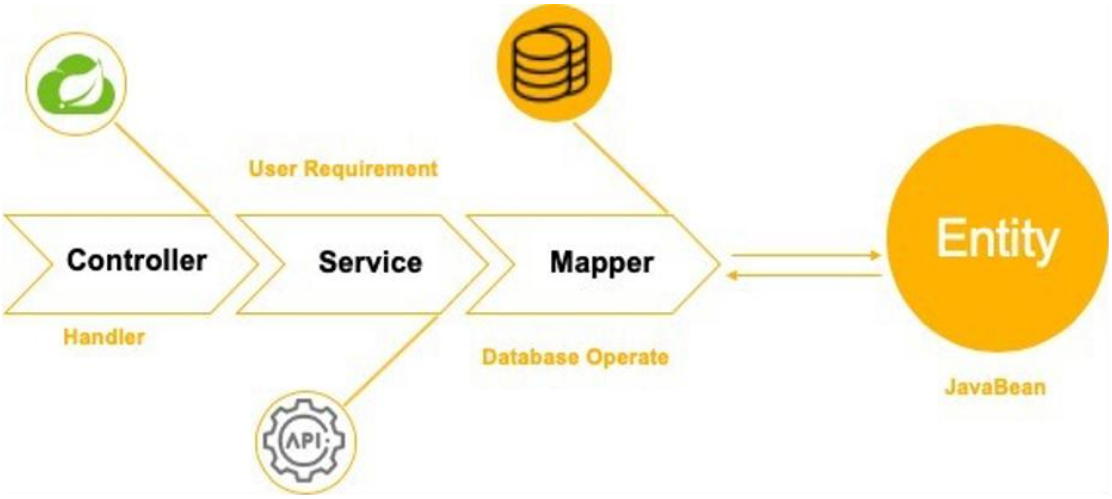


Figure 19 Working Model of a Spring Boot Application

As more developers use Redis[8] as a key-value data store to process information in real-time, the open source project continues to gain popularity. Redis use is increasing, especially at infrastructure-as-a-service businesses (like Atlantic.Net)[9].
MyBatis is an excellent tirelessness framework that supports custom SQL, stored methods, and advanced mappings. MyBatis eliminates almost all of the JDBC code as well as manual boundary setup and result recovery.

Together with the JVM (Java Virtual Machine) and the JRE, the Java Development Kit (JDK)[11] is one of the three primary technology packages used in Java development (Java Runtime Environment).
I captured a couple back-end functionality screenshots.
Application Properties
I used spring-cloud and registered all servers to eureka server, the figure are as follow.

A Spring Boot application is a self-contained Java-based application that can be easily deployed and run. It follows the principle of convention over configuration, which means that the framework provides sensible defaults for configuration to reduce the amount of boilerplate code required to get up and running.

The working model of a Spring Boot application involves the following steps:

Application Startup: When a Spring Boot application is started, it first reads the configuration files such as application. Properties or application, which contain the configuration options for the application.

Object Creation: The next step is to create objects for the various components of the application. These include controllers, services, repositories, and other components required by the application.

Dependency Injection: The Spring framework uses dependency injection to inject the required dependencies into the objects. This allows the application to remain flexible and easy to maintain, as dependencies can be added or removed without altering the application code.

Component Scan: Spring Boot automatically discovers components by scanning the classpath for classes annotated with @Component, @Service, @Repository, and other annotations. This allows developers to easily create new components and add them to the application.

Web Container: Spring Boot includes an embedded web container, such as Tomcat or Jetty, that is automatically configured based on the application's dependencies.

Request Handling: When a request is received by the web container, it is forwarded to the appropriate controller for handling. Controllers are responsible for processing the request, interacting with the application's data, and returning a response.

Data Access: Spring Boot provides an abstraction layer for data access, which allows developers to work with data from multiple sources such as databases, files, or web services. The framework can automatically generate CRUD operations for entities and provide standard implementations of data repositories.

Configuration: Spring Boot provides a number of ways to configure the application. This includes properties files, YAML files, environment variables, and various other options.

Testing: Spring Boot includes a number of testing tools and annotations to facilitate writing unit and integration tests. It also provides tools for mocking dependencies and for testing web applications.

Overall, the working model of a Spring Boot application involves automatically configuring, discovering, and wiring components, which allows developers to focus on writing business logic rather than worrying about infrastructure or plumbing.

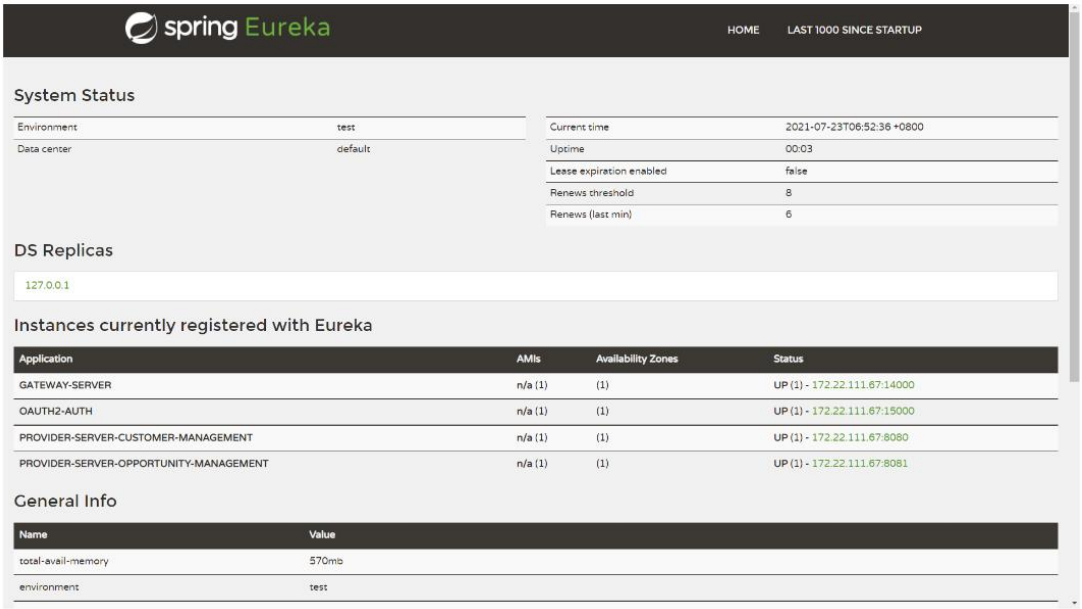


Figure 20 Oauth2& Gateway

In realizing oauth2 authentication using gateway, I met cross-domain problems which had bothered me for a long time. In the end, I got the answer online that I need to add

a configuration file in gateway to allow cross-domain. The sample codes are shown as following.

Then I considered all the roles of those employees and set the rules of requests url that some rules can or can't access.

```
Override
public void configure(HttpSecurity http) throws Exception {
http.authorizeRequests()
/.antMatchers(HttpethodposT, .antPatterns:
"/oppManagement/opportunity/add0pportunity").hasAnyRole...oles: "30000010" "30000030")//
antMatchers(HttpMethod.PosT,
antatters:"/opManagement/oportunity/showUpdatePage").hasAnyRole(...oles: "30000010"
"30000030")
anthatchers(HtpMethodposT,..antatterms:"/oppManagement/opportunity/update0pportunity").has
AnyRole( ..oles: "30000010" "3000030"
antMatchers(HttpHethod.GE, .-antatterns:
"/oppManagement/opportunity/getptrackWainPage").hasnyRole( .-.roles: "30000010" "30000030"

antMatchers(HttpMethodosT, .antPatterns:
"/oppManagement/opportunity/curdTrackinglog").hasAnyRole( ..roles:"30000010" "30000030")
antMatchers(HttpMethod.GE, .antatters:"/oppManagement/opportunity/getApprovalPage").hasAn
vRole . roles: "3000010" "1000030" "20000010"//
anthatchers(HttMethod.GE,antattems: "oolanagement /portuit/ShwDoAproveDeta ")asrvolel .roles
"3000010""000030" "2000010"//
.anthatchers(HttpMethod.PosT,.-.antatterns:
"/oppManagement/opportunity/approval").hasAnyRole( .roles: "3000010" "1000030" "20000010")
.anyRequest().authenticated( );
```

the code above provided is a configuration method that overrides the default configuration of the Spring Security module in a Java-based Spring Boot application. Specifically, the method configures the security rules for various endpoints/endpoints patterns of the application.

The rules specified in the code block specify the required permissions (roles) for accessing different endpoints. For example, the rule ".antMatchers(HttpMethod.POST, .antPatterns: "/oppManagement/opportunity/add0pportunity").hasAnyRole...oles: "30000010" "30000030")" specifies that any user with the roles "30000010" or "30000030" can access the "/oppManagement/opportunity/add0pportunity" endpoint using the HTTP POST method.

Similarly, the other antMatchers() methods specify the required roles for accessing other endpoints using various HTTP methods.

The last rule ".anyRequest().authenticated()" specifies that all other requests that are not matched by the previous antMatchers() methods should be authenticated before accessing them.

Overall, this method provides a powerful way to configure the security of the Spring Boot application by defining required permissions for accessing each endpoint.
Here we can see the database name, server port number of opportunity management.

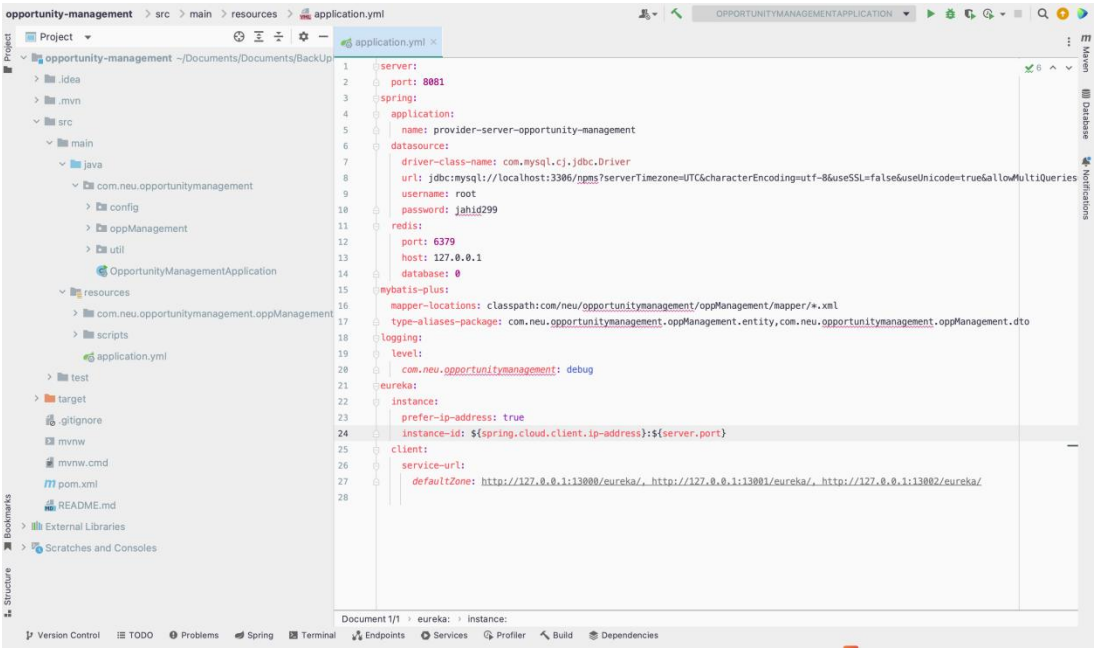


Figure 21 Opportunity Management Application Properties

Here we can see the database name, server port of number customers management

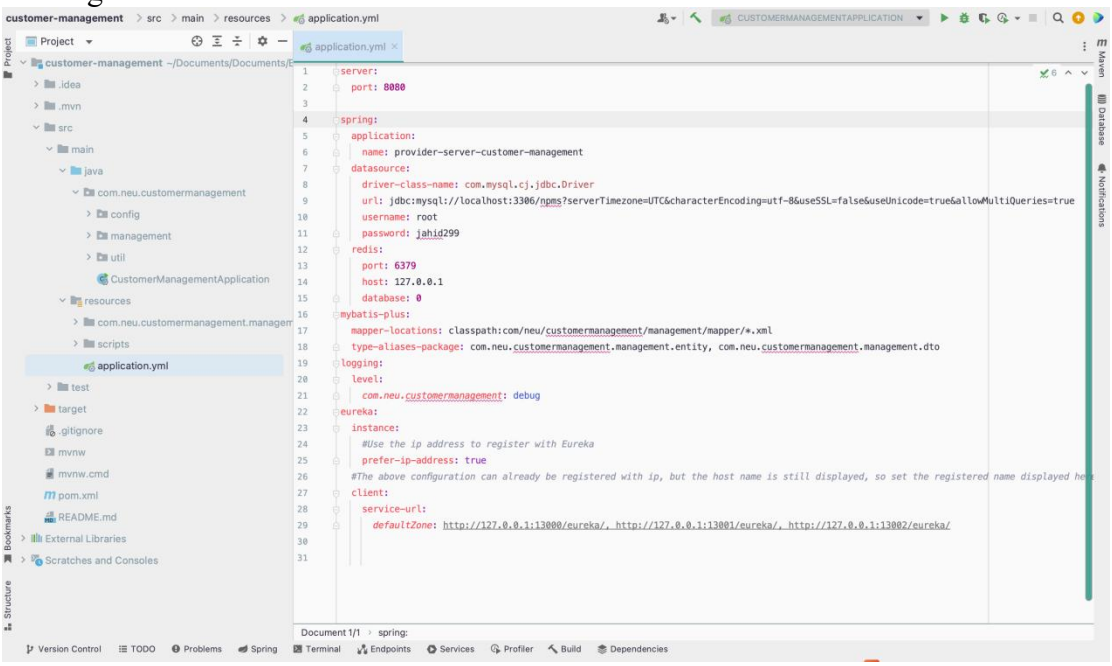


Figure 22 Customers Management Application Properties

Here we can see the database name, server port number of Eureka Server

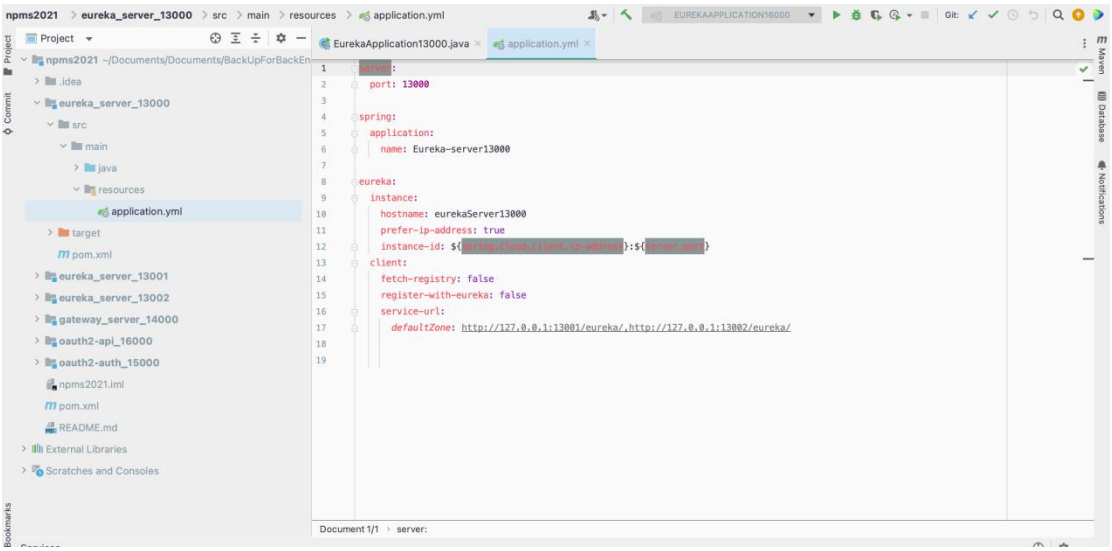


Figure 23 Eureka Server Application Properties

Opportunity Management Controller

Here we can see the backend of **Opportunity**. Where we can see the get all Opportunity, edit Opportunity and delete Opportunity.

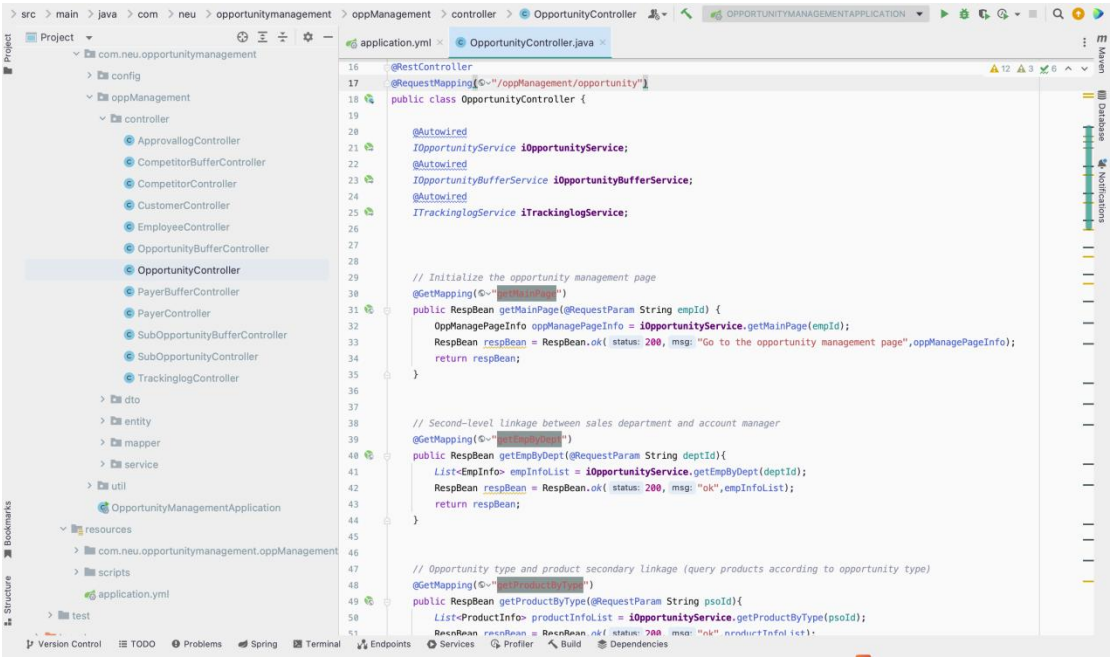


Figure 24 Opportunity Management Controller

The above class It contains various methods that handle different types of operations such as retrieving, adding, updating, and deleting opportunities, along with other operations such as displaying details and performing approvals.

These methods use objects and services from the IOpportunityService, IOpportunityBufferService, and ITrackinglogService interfaces to provide the required functionality.
Some of the notable methods and their purpose include:

getMainPage: Initializes the opportunity management page and returns main page info based on the employee ID.
getEmpByDept, getProductByType, getTypeByProduct: These methods provide two-level cascading selection of employees, products, and opportunity types based on the selected department or product.
showOppDetail and showOppApproveDetail: These methods retrieve detailed information about a specific opportunity based on the opportunity ID.
getOpportunity: This method searches for opportunities based on the set conditions and returns the searched results.
testAddRepetition and testUpdateRepetition: These methods check for duplicate opportunity names and IDs before adding or updating opportunities.
addOpportunity and updateOpportunity: These methods add and update opportunities based on the provided information.
getOppTrackMainPage: Initializes the opportunity tracking page and returns the main page info based on the opportunity ID.
curdTrackinglog: Performs create, update, or delete operations on the tracking records of an opportunity.
getApprovalPage: Initializes the opportunity approval page and returns the main page info based on the employee ID.
approval: Approves or rejects opportunities based on the provided information.
Overall, this controller provides the necessary endpoints for managing opportunities in the application, using various service interfaces to provide the required functionality.

Customer Management Controller

Here we can see the backend of customer. Where we can see the get all customer, edit customer and delete customer.

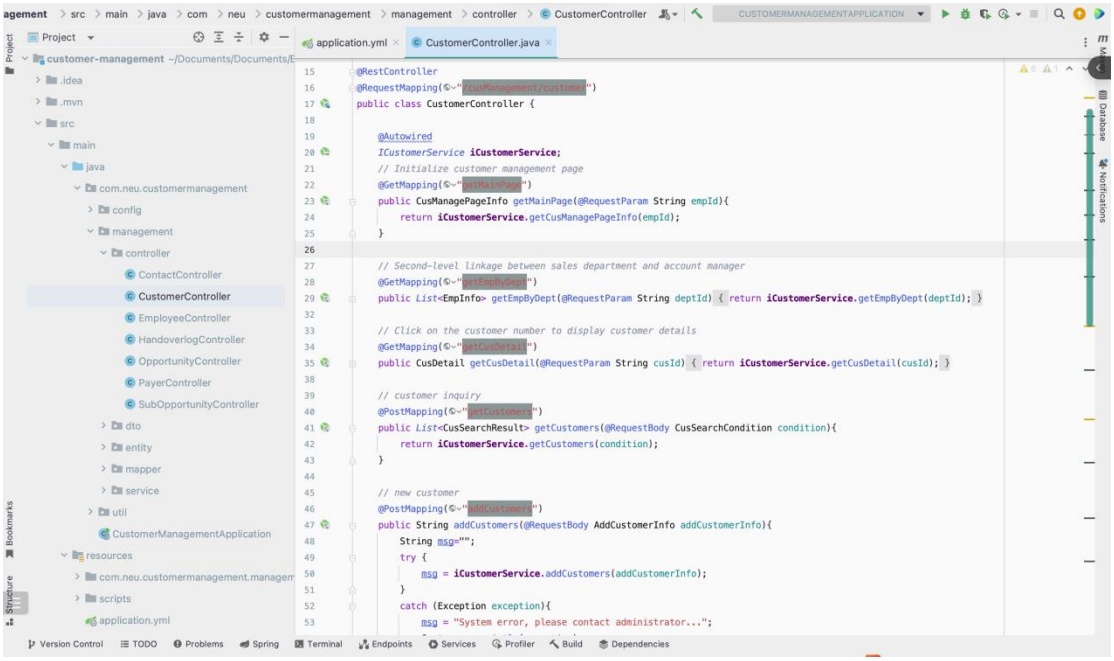


Figure 25 Customer Management Controller

Handoverlog Controller

Here we can see the backend of Handover log. Where we can see the get all Handoverlog, edit Handoverlog and delete Handoverlog.

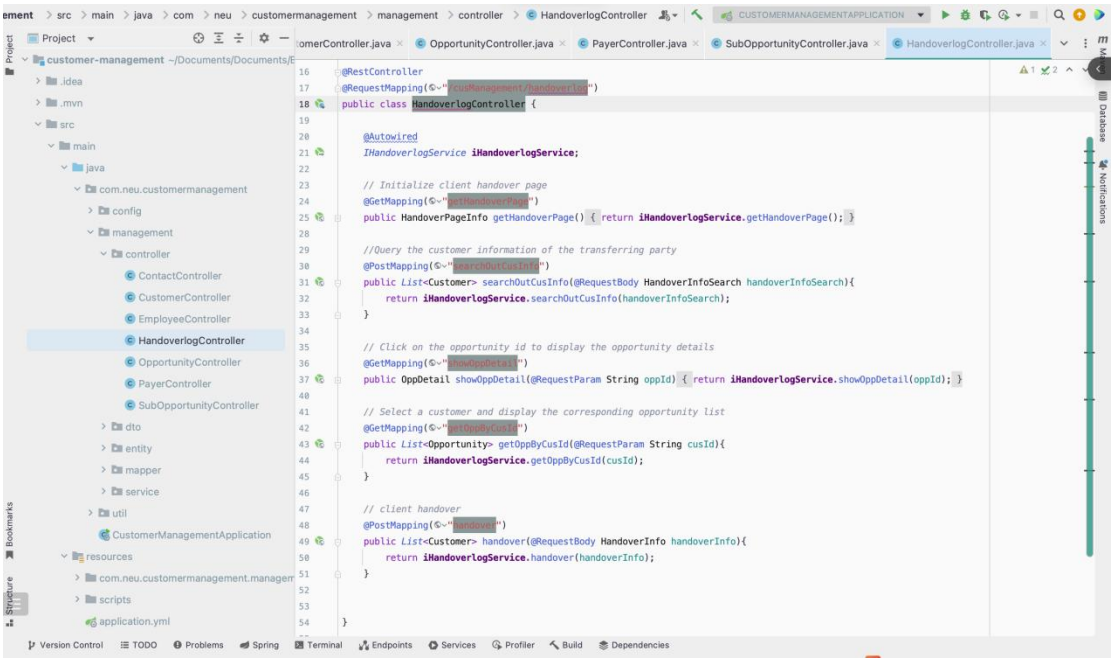


Figure 26 Handoverlog Controller

POM.XML

Here we can see the backend of dependencies which we have added into our project.

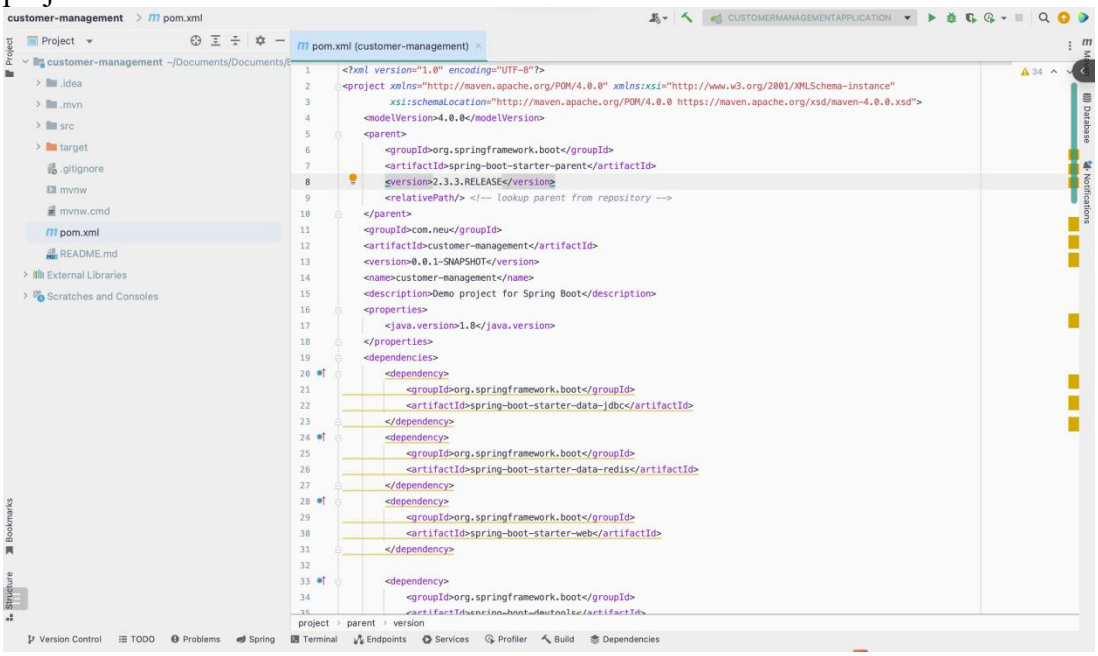


Figure 27 Pom Xml

My Contribution for Frontend (2128030)

I am responsible for the back-end development of the system. Vue.js is a progressive JavaScript framework used for building user interfaces and single-page applications (SPAs). It was created by Evan You in 2014, and since then it has gained a lot of popularity among web developers. Vue.js is known for its simplicity and flexibility, making it easy to learn and use. It is also known for its reactive and composable view components, which allow developers to create complex user interfaces using reusable code. Some of the key features of Vue.js include:
Reactive data binding: Vue.js uses a reactive data binding system, which means that changes in the data are automatically reflected in the UI, without the need for explicit DOM manipulation.
Component-based architecture: Vue.js allows developers to create reusable and composable components, which can be combined to create complex user interfaces.
Virtual DOM: Vue.js uses a virtual DOM, which allows it to update the UI efficiently by only updating the parts of the DOM that have changed.
Templates: Vue.js uses templates to define the structure and appearance of the UI, making it easy to create dynamic and responsive UIs.
Directives: Vue.js provides built-in directives that allow developers to add special functionality to their UI components, such as conditional rendering, looping, and event handling.

As a frontend developer using Vue.js, my role could involve developing reusable UI components, implementing navigation and user input handling, integrating with backend APIs utilizing Axios, developing custom directives and filters, and setting up client-side routing with the Vue Router. One important feature of Vue is the ability to use components. Components I use for reusable Vue instances with custom elements. Components can be reused as many times as I want or used in another component, making it a child component. Data, computed, watch, and methods can be used in a Vue component. Each Vue component instance goes through a series of initialization steps when it's created - for example, it needs to set up data observation, compile the template, mount the instance to the DOM, and update the DOM when data changes. I also Vuex serves as a centralized store for all the components in an application, with rules ensuring that the state can only be mutated in a predictable fashion as well as used Composition API is a set of APIs that allows us to author Vue components using imported functions instead of declaring options. It is an umbrella term that covers the following APIs: Reactivity API, e.g. ref() and reactive() , that allows us to directly create reactive state, computed state, and watchers. Finally, Axios executes on both client and a server, which makes an API request, does the task to produce the result and specifies easier concepts to read and debug.

In addition, as a Vue.js developer, I need collaborate with the backend developers to ensure that the frontend and backend are properly integrated, and ensure that the web application or website is optimized for performance and is responsive across multiple devices.

Overall, I would be responsible for ensuring that the web application or website is visually appealing, user-friendly, and meets the requirements provided by the project team

My Innovation Point (2128030)

User interface (UI) elements are the parts I use to build sites for add interactivity to a user interface, providing touchpoints for the user as they navigate their way around; think buttons, scrollbars, menu items and checkboxes, Search bar, deleting, editing as well as adding new information functionality. In order to make more standard out from the crowd, I focused on creating a unique and engaging user experience. This can involve implementing creative user interface (UI) designs, adding innovative animations and transitions, or incorporating intuitive and user-friendly features. Designed approval, chances, customer list, customer handover more intuitively using UI elements. Home page is more innovative to look and all sort of detail shown be in this page which is unique.

For our project, I used the Spring Boot framework, My Batis, Java, IntelliJ IDEA, and a little functionality. This serves as a succinct overview of each of them.

I utilize the GUI Designer in IntelliJ IDEA [10] to design graphical user interfaces (GUI) for your projects utilizing Swing library components. The creation of a small Service makes use of the SpringBoot [1][4] open-source Java-based platform. The Pivotal Team created it, and it's used to assemble standalone and pre-made spring applications. You learn about Spring Boot in this part and get acquainted with its main ideas.

4.4 Frontend

We utilized VUE[5], Element-Ui, Axios, and Tomcat in our project. This is a brief introduction to each of them.Vue is a reformist system for creating user interfaces (articulated/vju/, like view). Vue was designed from the outset to be progressively adopted, unlike other rigid constructions. The core library is simple to get, is easy to integrate with other libraries or ongoing projects, and is only focused on the view layer. It is a magical vue[6] administrator that depends on the most recent upgrade stack of vue, implicit i18n arrangement, typical formats for large-scale corporate applications, and a ton of great features.
Axios is a well-known JavaScript library that can be used to send HTTP requests and is compatible with both browsers and nodes.

jQuery 7 platform. Microsoft created the freeware source-code editor Visual Studio Code[12] for Windows, Linux, and macOS. Java Servlet, JavaServer Pages (JSP), and the WebSockets Application Programming Interface are all implemented by Apache Tomcat (API). It basically functions as a pure Java HTTP web server that supports Java code, giving your website more cross-platform flexibility than some of its counterparts.

Login Page

Login page have following functionality

- Login if you are already registered
- Check the correct username and password
- Login if you have corrected login details
- Show error if you don’t have correct login details

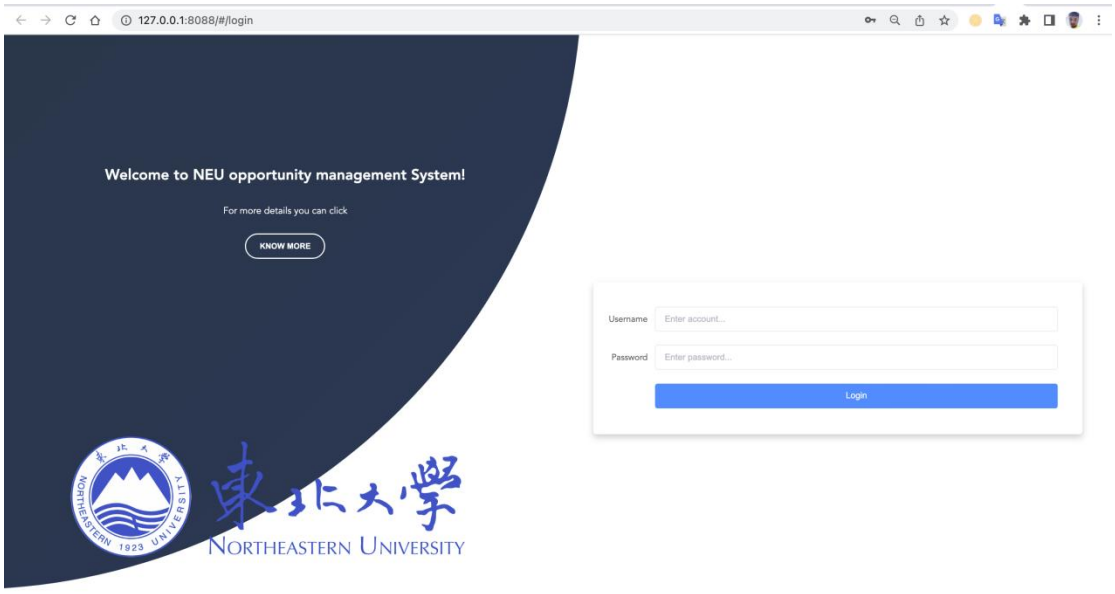


Figure 28 Login page

Home Page

Home page have following functionality

- According to your authentication you can access the website
- You can see the opportunity and customer management details and all other details
- Different menu will be show for data summary, opportunity statistics, opportunity distribution, and customer satisfactions
- I designed home page with very detail

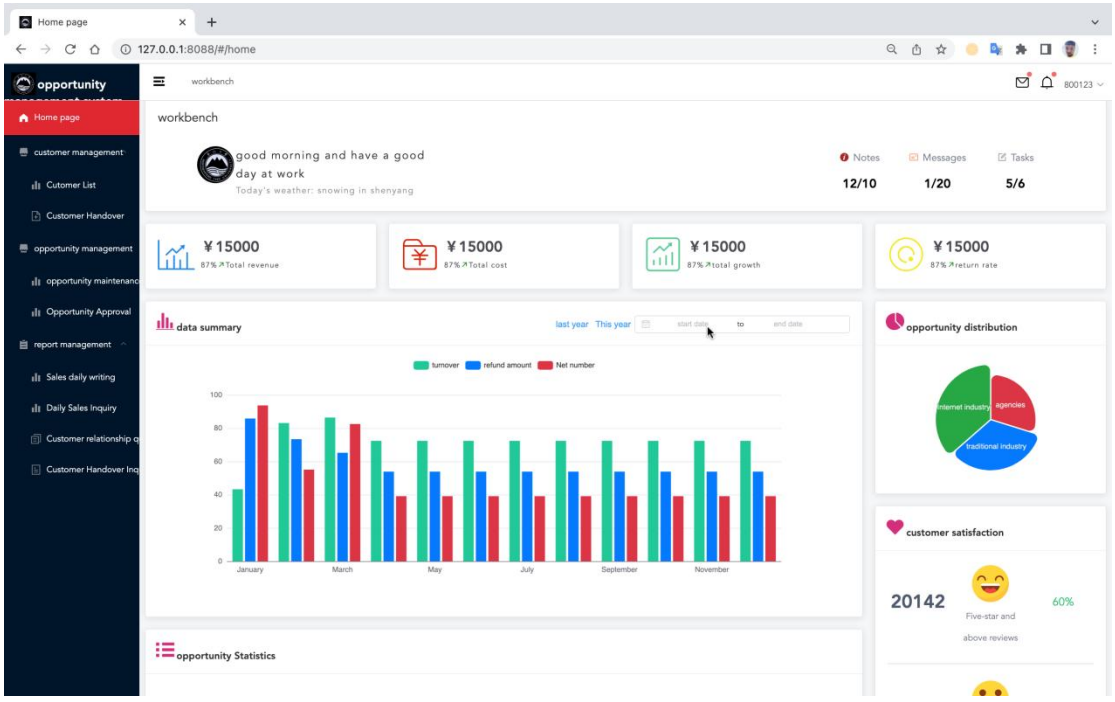


Figure 29 Home page

Customer Management Page

Customer Management page have following functionality

- Show the information to about the customer list
- Search information of Customers
- Admin can reset the customer from platform

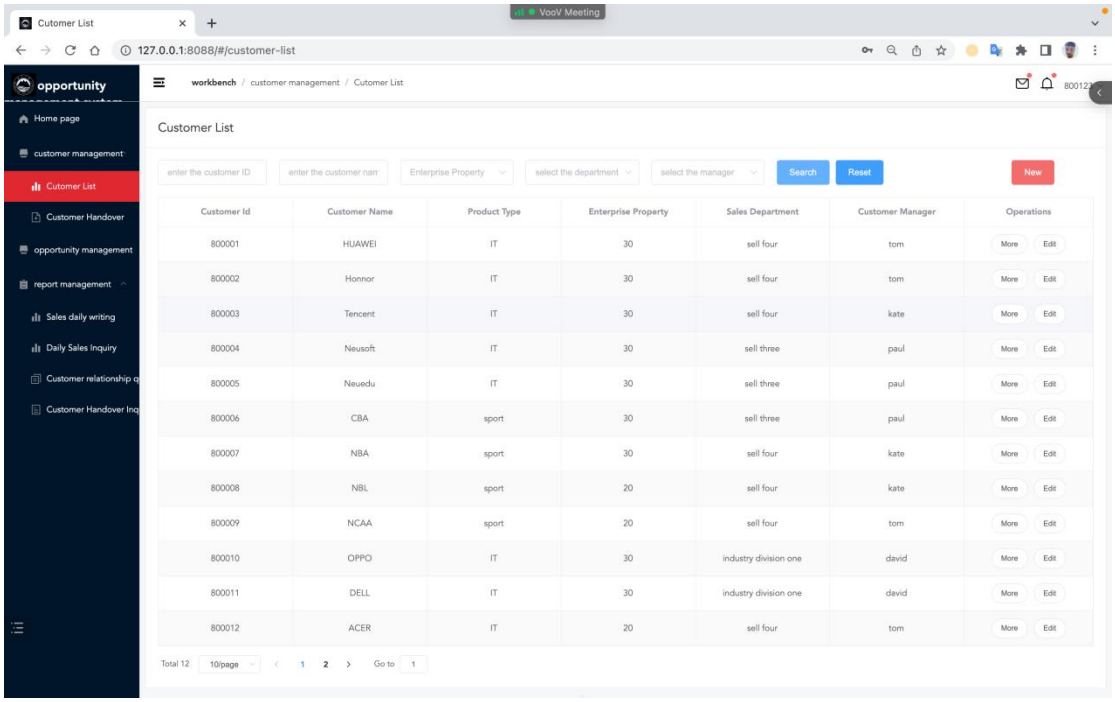


Figure 30 Customer Management Page

Customer Handover Management Page

Customer Handover Management page have following functionality

- Show the information to about the customer Handover Aside bar
- Search information of Customer Handover
- Admin can reset the customerHandover from the platform.
- There is an Receiver option as well Admin handover the customer and Receiver can submit to the department.

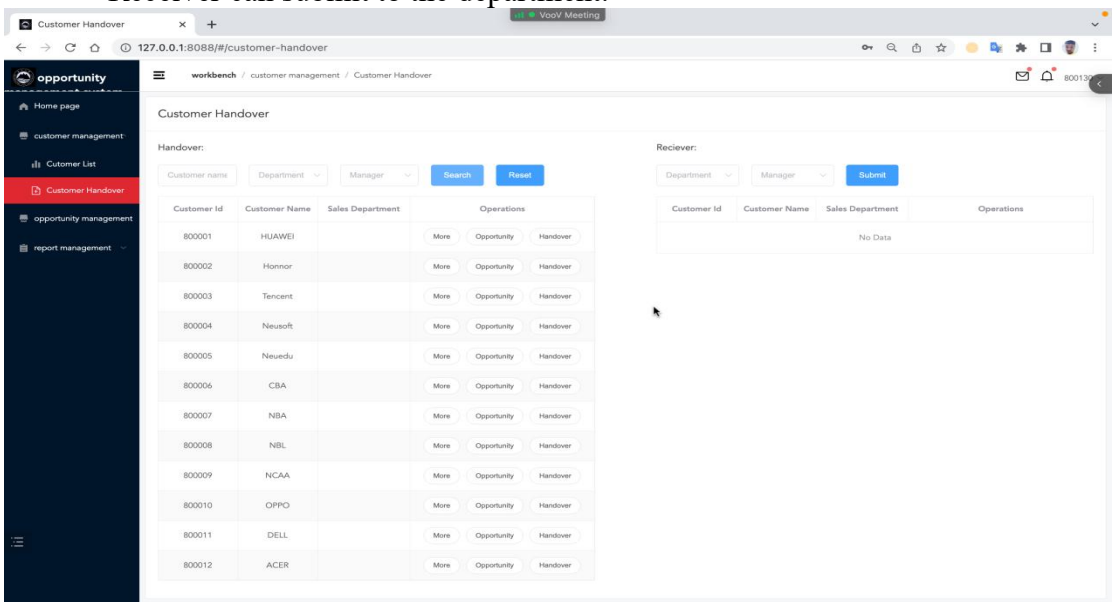


Figure 31 Customer Handover Management Page

Opportunity Approval Management Page

Opportunity Approval Management page have following functionality

- Show the information to about the Opportunity Approval
- Search information of Opportunity Approval
- User can operate the Opportunity Approval from the platform and can see other details.
- Opportunity Approval Management page is an essential tool for managing and accessing business opportunities before committing resources to them. With a streamlined approval process and effective management tools, companies can minimize risks and maximize their chances of success.

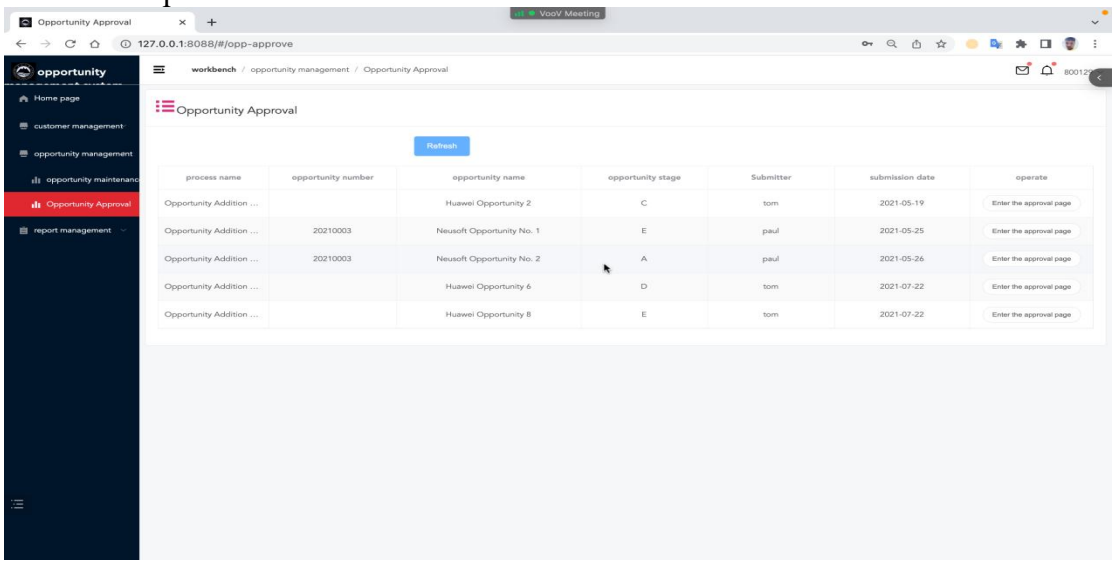


Figure 32 Opportunity Approval Management Page

Report Management Page

Report Management page have following functionality

- Show the information to about the Customers Relations Query
- Search information of Customers Relations
- Admin can reset the information's

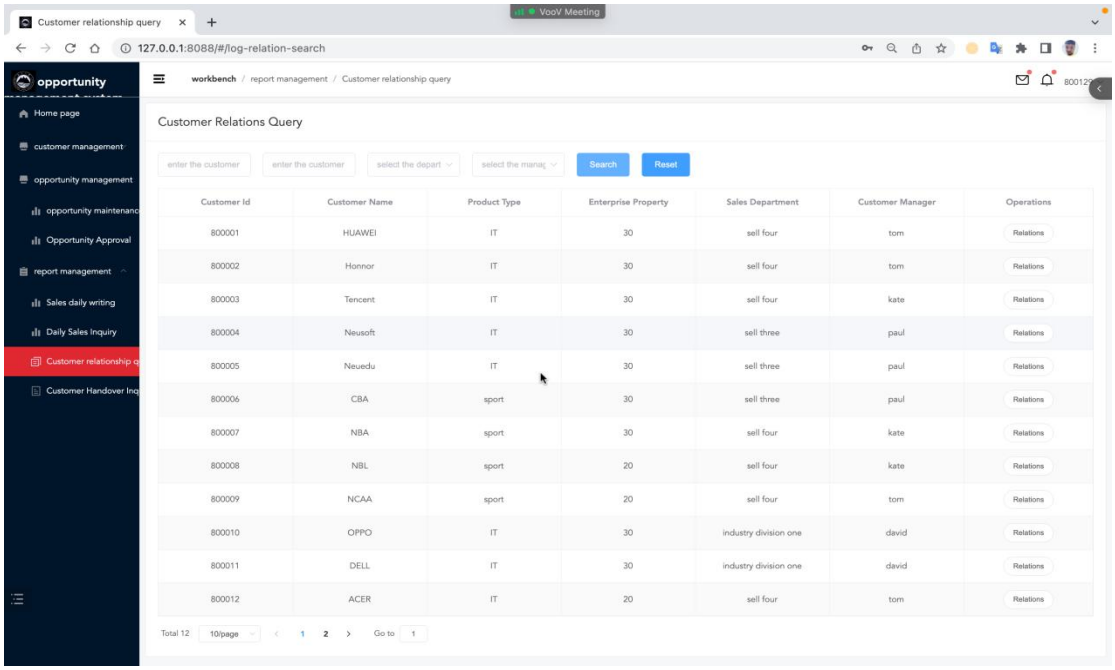


Figure 33 Report Management Page

Chapter 5 SOFTWARE QUALITY ASSURANCE

Testing, auditing, formal affirmation, deformation avoidance, adjusting to internal disappointment, and other frameworks may be used to attempt the nature of a product item. The approaches we utilized to test the functionality of our framework system will be examined in this section.

5.1 Software Testing

5.1.1 Black Box Testing

Table 15 Admin Login

Test Case Id	**Test-01**
Functional Requirement	Admin Log in.
Test Description	To test this function, will it Log in Admin account or not?
Steps	Fill in the details and click "Sign In" to access the account.
Actual Result	Admin logged in successfully.
Expected Result	Admin will be logged in into his account.

Table 16 Checking for incomplete Fields

Test Case Id	**Test-02**
Functional Requirement	Checking for incomplete field
Test Description	To test this function, will it allow user to complete action or not?
Steps	Missing input fields and click submit.
Expected Result	It must show errors or ask users for entering text.
Actual Result	Same as above.

Table 17 Admin updating his My Info's details in fund check to database

Test Case Id	**Test-03**
Functional Requirement	Updating my info's details in database.
Test Description	To test this function, will it update my info's details in database or Not?
Steps	Fill in the details and click "Update" button to add data to database.
Expected Result	Data will be successfully update in database.
Actual Result	Data successfully the updated in database.

Table 18 Admin showing opportunity info's details from database

Test Case Id	Test-04
Functional	Showing opportunity info's details from

Requirement	database.
Test Description	To test this function, will showing record's details from database or Not?
Steps	Enter the user's id in id field and click "My Info" button to show it From database.
Expected Result	Details should be successfully shown from database.
Actual Result	Details successfully shown from database.

Table 19 Admin adding opportunity details to database

Test Case Id	Test-05
Functional Requirement	Adding opportunity details in database.
Test Description	To test this function, will it add opportunity details in database Or not?
Steps	Fill in the details and click "Save" button to add data to database.
Expected Result	Data will be successfully added to database.
Actual Result	Data successfully added to the database.

Table 20 Admin updating opportunity details to database

Test Case Id	Test-06
Functional Requirement	Updating opportunity details in database.
Test Description	To test this function, will it update opportunity details in database Or not?
Steps	Fill in the details and click "Update" button to add data to database.
Expected Result	Data will be successfully update in database.
Actual Result	Data successfully the updated in database.

Table 21 Admin deleting opportunity details to database

Test Case Id	Test-07

Functional Requirement	Deleting opportunity details from database.
Test Description	To test this function, will delete opportunity details from Database or not?
Steps	Enter the paramter id in id field and click “Delete” button to Delete it from database.
Expected Result	Details should be successfully deleted from database.
Actual Result	Details successfully deleted from database.

Table 22 show opportunity details to the admin in list

Test Case Id	Test-08
Functional Requirement	Show all opportunity details in list to the admin.
Test Description	To test this function, will it fetch opportunity details from Database and show it to admin or not?
Steps	Go to opportunity section in admin panel if data is present it will Be shown.
Expected Result	All opportunity details should be successfully listed to the admin.
Actual Result	All opportunity details is listed to the admin.

Table 23 10 Customer adding user’s details to database

Test Case Id	Test-09
Functional Requirement	Adding user’s details in database.
Test Description	To test this function, will it add user’s details in database or not?
Steps	Fill in the details and click “Save” button to add data to database.
Expected Result	Data will be successfully added to database.
Actual Result	Data successfully added to the database.

Table 24 Customer updating user's details to database

Test Case Id	Test-10
Functional Requirement	Updating user's details in database.
Test Description	To test this function, will it update user's details in database or Not?
Steps	Fill in the details and click "Update" button to add data to database.
Expected Result	Data will be successfully update in database.
Actual Result	Data successfully the updated in database.

Table 25 Customer deleting user's details to database

Test Case Id	Test-11
Functional Requirement	Customer user's details from database.
Test Description	To test this function, will delete user's details from database or Not?
Steps	Enter the user's id in id field and click "Delete" button to delete it From database.
Expected Result	Details should be successfully deleted from database.
Actual Result	Details successfully deleted from database.

Table 26 show user's details to the customer in list

Test Case Id	Test-12
Functional Requirement	Show all user's details in list to the Customer.
Test Description	To test this function, will it fetch user's details from database and Show it to customer or not?
Steps	Go to user section in admin panel if data is present, it will be shown.
Expected Result	All the user's details should be successfully listed to the customer.
Actual Result	All the user's details is listed to the customer.

Table 27 customer showing his record's details from database

Test Case Id	Test-16
Functional Requirement	Showing his record's details from database.
Test Description	To test this function, will showing record's details from database or Not?
Steps	Enter the user's id in id field and click "Record" button to show it From database.
Expected Result	Details should be successfully shown from database.
Actual Result	Details successfully shown from database.

Table 28 show data dictionary's details to the Customer in list

Test Case Id	Test-17
Functional Requirement	Show all data dictionary's details in list to the customer.
Test Description	To test this function, will it fetch data dictionary's details from database and Show it to admin or not?
Steps	Go to data dictionary's section in admin panel if data is present, it will be shown.
Expected Result	All the data dictionary's details should be successfully listed to the customer.
Actual Result	All the data dictionary's is listed to the customer.

Table 29 Customer updating his menu's detail to database

Test Case Id	Test-18
Functional Requirement	Updating menu's details in database.
Test Description	To test this function, will it update menu's details in database or not?
Steps	Fill in the details and click "Update" button to add data to database.
Expected Result	Data will be successfully update in database.
Actual Result	Data successfully the updated in database.

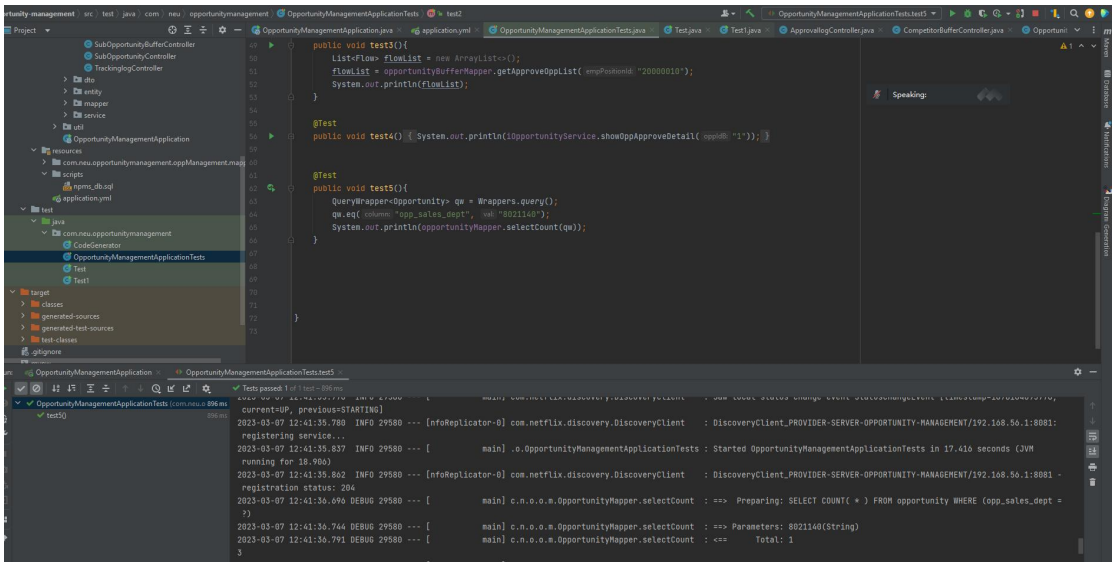


Figure 34 Opportunity Management Application test

This is a JUnit test class for the customer management application. It contains several test methods that are used to test different functionalities of the application.

The @SpringBootTest annotation is used to load the application context and create an instance of the application for testing. The @Autowired annotation is used to inject dependencies into the test class.

The test methods perform various operations such as retrieving customer, employee, and contact data, adding new customers, updating customer data, and mapping data to DTOs for display purposes. The methods also test error handling functionality for invalid input.

Overall, this class helps to ensure that the application functions correctly and satisfies its requirements.

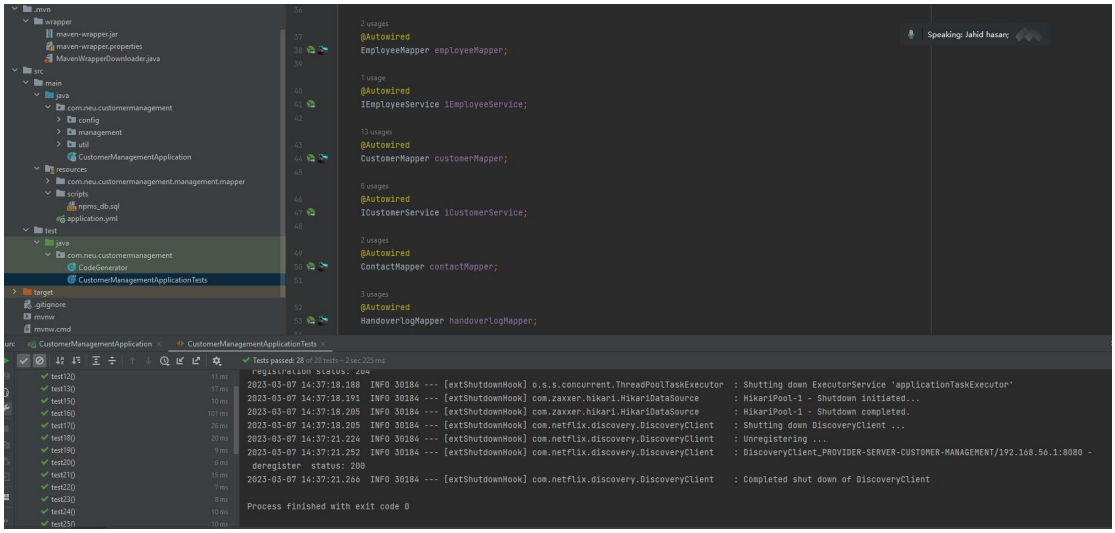


Figure 35 Customer Management application test

This is a test class for the Opportunity Management application. It contains several test methods that perform various operations on the Opportunity entity and its related entities. The class imports various packages, including MyBatisPlus, the Opportunity entity, and its related DTOs and mappers. The @Autowired annotation is used to inject the OpportunityMapper, OpportunityBufferMapper, and IOpportunityService objects.

The test methods perform the following operations:

contextLoads() - This is an empty method used to test if the Spring context loads successfully.
test1() - This method tests the getOpportunity() method of the OpportunityMapper by passing an instance of OppSearchCondition. It prints the output of the method to the console.
test2() - This method tests the getOpportunityB() method of the OpportunityMapper by passing an instance of OppSearchCondition. It prints the output of the method to the console.
test3() - This method tests the getApproveOppList() method of the OpportunityBufferMapper by passing an approver ID. It prints the output of the method to the console.
test4() - This method tests the showOppApproveDetail() method of the IOpportunityService by passing an opportunity ID. It prints the output of the method to the console.
test5() - This method tests the selectCount() method of the OpportunityMapper by using a QueryWrapper to filter the opportunities by sales department. It prints the output of the method to the console.

5.1.2 White Box Testing

By doing verbalization coverage, decision coverage, condition coverage, and other tests, we have carried out white box testing. The test cases that follow reflect all of the coverages.

Table 30 Statement Coverage

Test Case Id	**Test-01**

Test Description	final TabHost tabHost = (TabHost) findViewById(R.id.tabhost); tabHost.setup(); TabHost.TabSpec tabSpec = tabHost.newTabSpec("Accomodation"); tabSpec.setContent(R.id.accomodation); tabSpec.setIndicator("Accomodation"); tabHost.addTab(tabSpec);
	
	tabSpec = tabHost.newTabSpec("Users Detail"); tabSpec.setContent(R.id.Users_Detail); tabSpec.setIndicator("Users Detail"); tabHost.addTab(tabSpec);
	
	tabSpec = tabHost.newTabSpec("Users Schedule"); tabSpec.setContent(R.id.User_schedule); tabSpec.setIndicator("Users Schedule"); tabHost.addTab(tabSpec);

Type of White Box Testing	Statement coverage
Result	Tabhost get the id assigned in XML and used for creating new tabs.

Table 31 Decision Condition Coverage

Test Case Id	**Test-02**
Test Description	signup.setOnClickListener(new View.OnClickListener() { @Override public void onClick(View view) { if(usrusr.length() ==0){ usrusr.setError("Enter User Name"); } if(passwordd.length()==0) { passwordd.setError("Ent er Password");

	}
	
	if(mobphone.length()==0){ mobphon
	e.setError("Enter Phone Number");
	}
	
	if(cnic.length()==0){ cnic.setE
	rror("Enter User Name");
	}
	
	Auth.createUserWithEmailAndPassword(email,password).addOnCompleteListene
	r(new OnCompleteListener<AuthResult>() {
	@Override
	public void onComplete(@NonNull
	Task<AuthResult> task) { if(task.isSuccessful()){
	Toast.makeText(signup.this, "Success",
	Toast.LENGTH_SHORT).show(); Intent it = new
	Intent(signup.this,MainActivity.class);
	
	startActi
	vity(it);
	finish();
	}
	if(!task.isSuccessful()){
	Exception e = task.getException();
	Toast.makeText(signup.this, "Error
	"+e ,Toast.LENGTH_LONG).show();
	}
	}
	});
	}
	});

Type of White Box Testing	Decision coverage plus condition coverage are satisfied. Decision-Condition coverage.
Result	If-else conditions are executed line by line.

5.2 Inspection

It is essentially a review or a small group of software quality engineers accepting a piece of software made by a group to examine the source code in order to identify flaws and improve viability. For this, many participants are chosen to find defects throughout the inspection process. To carry out the inspection procedure in table 5...., a checklist is established.

Table 32 Inspection

S. No	Conditions	Yes	No	N/A
Requirements				

1	Are the requirements specification gathered properly?	Y		
2	Are the software requirements specification complete?	Y		
3	Does the developer fully understand the requirements completely before implementation?	Y		
4	Are the requirements changing dynamically with time?	Y		

Design				
1	Does the software design implement all the specification requirements?	Y		
2	Does the software design's physical characteristics i.e. color, layout etc. meet the requirements?	Y		
3	Are the user requirements (i.e. easy to use, easy to learn, easy to understand etc.) met?	Y		
5	Are the structure of the software product meets end user requirements?	Y		
6	Can the design modules and specification be tested to satisfy user requirements?	Y		
7	Is error detection and recovery is provided or not (e.g., input checking performed or not, recovery of database)?	Y		
9	Is there a clear representation of data flow and interfaces (icons, images, text etc.)?	Y		
10	Does all parts of design reflect user requirements or not?	Y		
Code				
1	Are all the program elements (i.e. data elements, functions, variable names etc.) Consistent?	Y		
2	Is the code consistent with specified requirements?	Y		
3	Does the detailed logical design implement properly through code?	Y		
4	Is every variable name utilized for just a retiring reason?	Y		
5	Is the logic of program correct and complete?	Y		
6	Is the code clear (i.e. comments, layout, interface, design)?	Y		
7	Are information components, procedures, and functions named and exploited reliably all through the system set and with outside interfaces?	Y		

6.3 Defect Prevention

Defect prevention is a processing the software quality engineering that is used to identify maximum critical defects and prevent them from occurring repeatedly. It involves three steps identify risk, estimated expected impact and minimize expected impact that are as following:

Table 33 Defect Prevention

Identify Critical Risks	**Estimated Expected Impact**	**Minimize Expected Impact**
Platform, admin panel if network connection is limited, carrying out any	Greater the unavailability of internet, greater is the unexpected impact	Try to make Internet connection available, appropriately.
action may cause unanticipated behavior		
Platform, if transaction is off locating nearby hospitals will never be found	Unavailability of transaction will cause greater is the unexpected impact.	Try to turn on the transaction before searching for the hospital.

Chapter 6 CONCLUSION AND FUTURE WORK

6.1 Conclusion

In conclusion, the Neusoft Opportunity Project Management System (NPMS) is a powerful tool that helps businesses of all sizes to effectively manage their opportunities and improve their sales performance. With its robust features such as lead management, pipeline tracking, and sales forecasting, the NPMS offers businesses an efficient and streamlined approach to sales management.

Furthermore, the NPMS offers customizable dashboards, real-time reporting, and analysis, which enables businesses to make data-driven decisions and stay ahead of their competition. In our NPMS platform customers can login and can use the system. Customers will have control over the menu, making it incredibly simple for administrators to alter the menu's title or icon from the front end without accessing the back end. The opportunity and other information are readily manageable by the customer. We made sure that our platform's user interface was incredibly simple so that anybody seeking for possibilities could go through it and utilize it without any trouble. High security measures, which are crucial for any kind of NPMS platform, were put in place by us. To draw consumers' attention to our site, I manually built a respectable and eye-catching design. Overall, the Neusoft Opportunity Management System is an excellent choice for businesses seeking to optimize their sales processes, increase their sales revenue, and gain a competitive edge in their industry.

6.2 Future Work

There are several potential areas for future work on the Neusoft Opportunity Management System project. Some of these areas could include:

Integration with additional business systems: While the Neusoft OMS is already designed to integrate with a range of business systems, there may be opportunities to expand this integration to other platforms and tools that businesses use. This could help to further streamline sales processes and improve the overall efficiency of the system.

Improved mobile functionality: While the Neusoft OMS already offers mobile access, there may be opportunities to enhance the system's mobile capabilities to further support salespeople on the go. This could include features such as push notifications for important sales activities or additional mobile-specific tools to help salespeople manage their opportunities from any location.

Advanced analytics and reporting: While the Neusoft OMS already offer robust reporting and analytics, there may be opportunities to further enhance these features to provide even more detailed insights into sales performance. This could include predictive analytics or more advanced data visualization tools to help businesses make data-driven decisions.

Machine learning and AI: There may be opportunities to incorporate machine learning and artificial intelligence into the Neusoft NPMS to further optimize sales processes and improve sales performance. For example, AI-powered lead scoring could help salespeople focus their efforts on the most promising opportunities, while machine learning algorithms could help to identify patterns and trends in sales data.

Overall, there are many potential areas for future work on the Neusoft Opportunity Management System project, and continued development and improvement could help to keep the system at the forefront of sales management technology.

References

[1] SpringBoot, Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications https://spring.io/projects/spring-boot

[2] SpringBoot, "Spring Boot Tutorial provides basic and advanced concepts of Spring Framework: https://www.javatpoint.com/spring-boot-tutorial

[3] Jordan Baker, "All About Spring Boot [Tutorials and Articles]": https://dzone.com/articles/spring-boot-framework-tutorials

[4] SpringBoot, "Documentation"-using database in Web available: https://docs.spring.io/spring-boot/docs/

[5] Vue.js, The Progressive JavaScript Framewor: The Vue.js story begins in 2013 when Evan You was working at Google creating lots of prototypes right within a browser. https://vuejs.org/

[6] Vue.js, Advantages and Disadvantages: The Good and the Bad of Vue.js Programming Framwork: https://www.altexsoft.com/blog/engineering/pros-and-cons-of-vue-js/

[7] Node.js, available: Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine. https://nodejs.org/en/

[8] Redis, A high performance Node.js Redis client: Redis is an open source (BSD licensed), https://redis.io/clients

[9] Redis, keeps becoming more prominent grade. Redis Advantages. Exceptionally fast https://aws.amazon.com/redis/

[10] IntelliJ IDEA. "IntelliJ IDEA is an integrated development environment written in Java for developing computer software. https://www.jetbrains.com/idea/

[11] Java Development Kit, "The Java Development Kit is an implementation of either one of the Java Platform:
https://www.oracle.com/java/technologies/javase-downloads.html

[12] Visual Studio Code, "Visual Studio Code is a freeware source-code editor made by Microsoft for Windows, Linux and macOS: https://code.visualstudio.com/