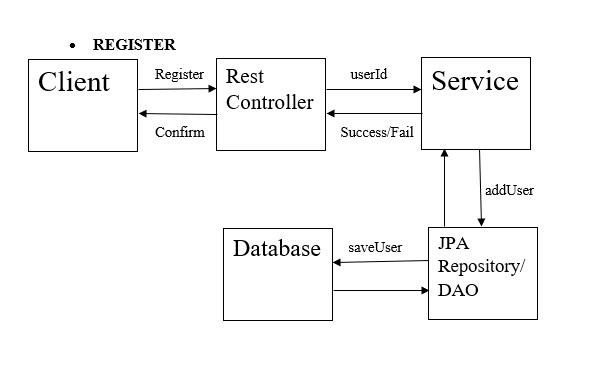
**IN-HOUSE MARKETPLACE APPLICATION**

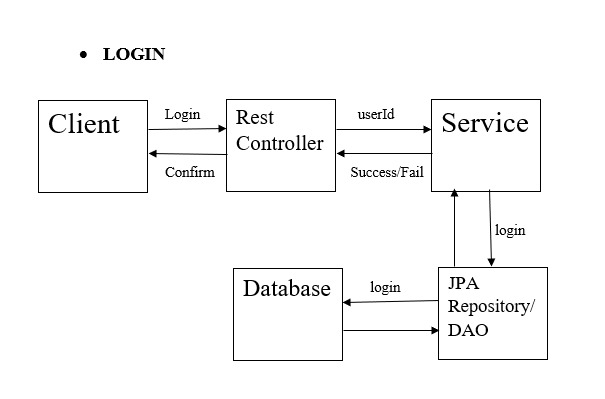
**INTRODUCTION-**

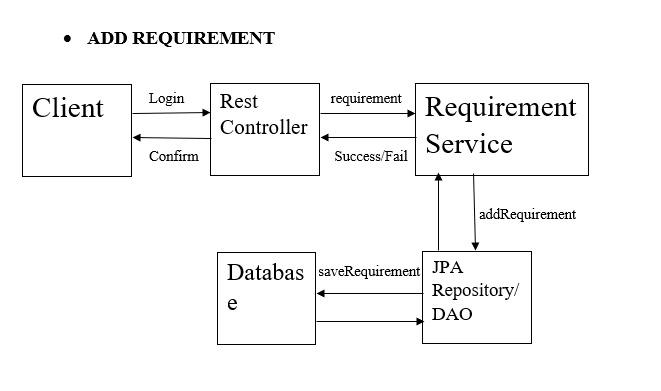
Inhouse marketplace has buyers and sellers find each other, in our case the employees can interact and efficiently cooperate for services and products. Sharing economy model employees coordinate via an online platform and are able to share access to goods and services. Employee can put up request for any product to buy and sell, for example if anyone would like to seek or offer something, like finding PG or Selling phone/bike etc. Employees may offer/require help, product or service to/from others. This application is only for the employees of a particular organization.

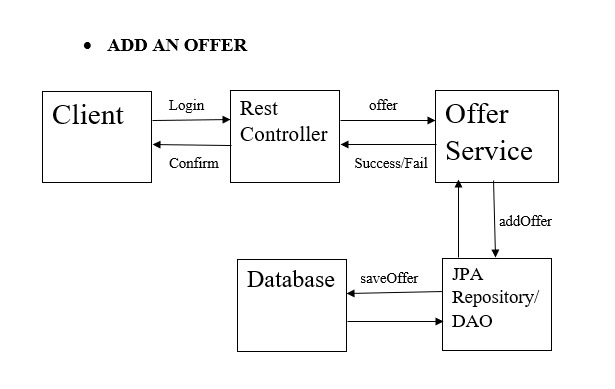
In other word’s it can be classified as the business-to-business model, which enables businesses to use a marketplace for bringing more customers to sellers, expand the area of reach and perform the function of product distribution at best.

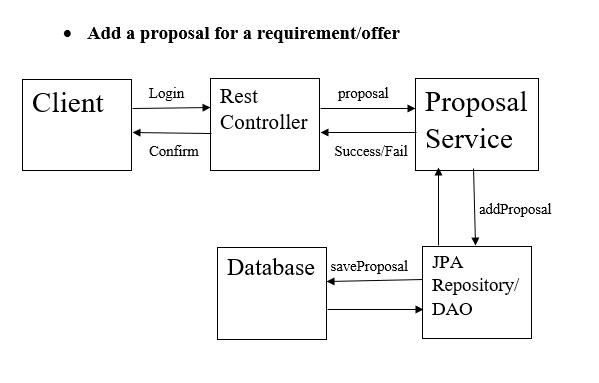
In our project Employee would be able to: -

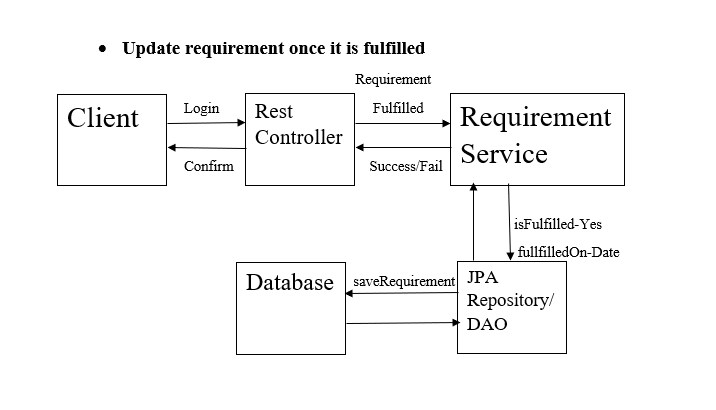


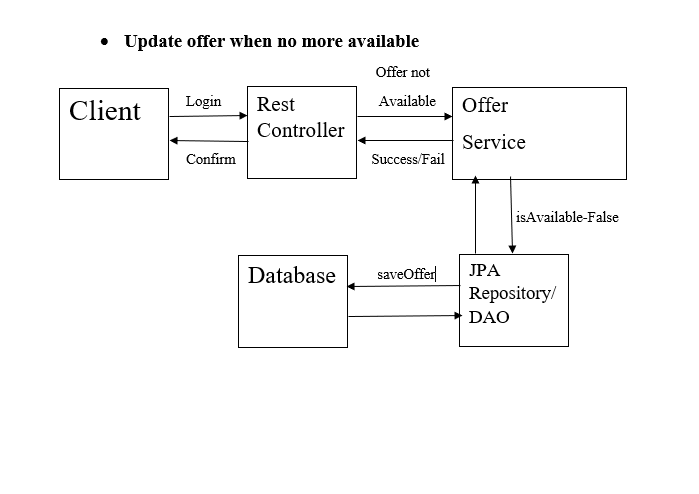


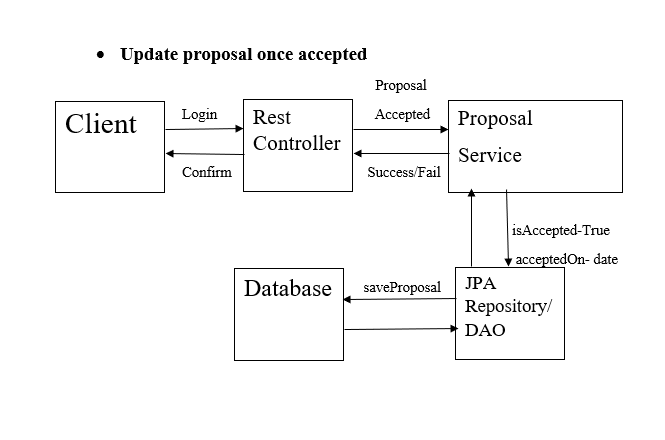


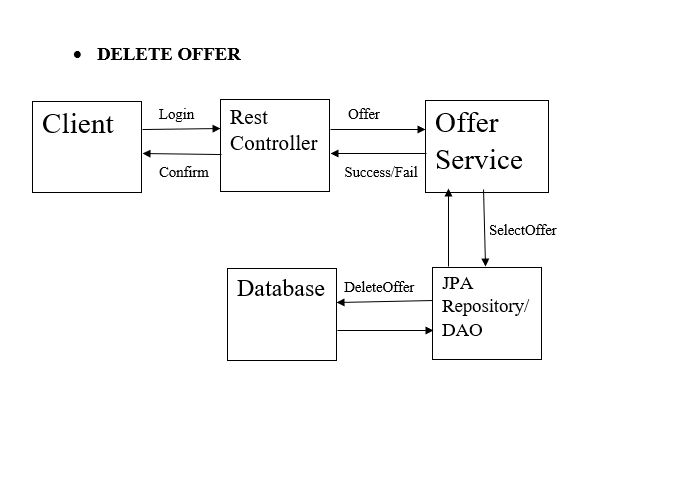


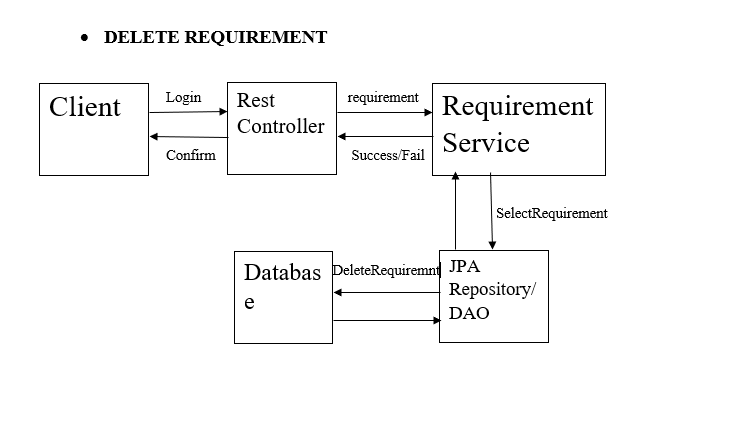


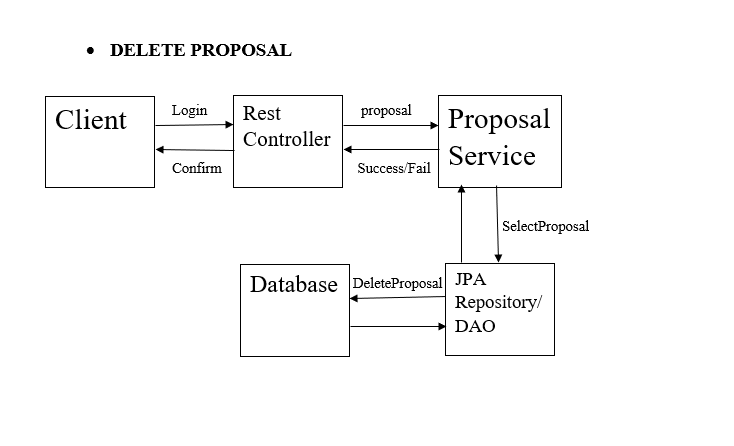








****

****

**METHODOLOGY-**

First step for our project is studying the class diagram, it shows the outline layer of our project, which includes modules for testing and the classes for backend. After finalizing the layout, we can start developing the components required for the application. We will be using Spring Boot, JPA with Hibernate and Rest Controller to build our project.

**Spring Framework-**

Spring Boot is an open-source Java-based framework used to create a micro-service. Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run". We take an opinionated view of the Spring platform and third-party libraries so you can get started with minimum fuss. Most Spring Boot applications need minimal Spring configuration.

**JPA with Hibernate-**

Object-Relational Mapping (ORM) is the process of converting Java objects to database tables. In other words, this allows us to interact with a relational database without any SQL. The Java Persistence API (JPA) is a specification that defines how to persist data in Java applications. The primary focus of JPA is the ORM layer.

Hibernate is one of the most popular Java ORM frameworks in use today. Its first release was almost twenty years ago, and still has excellent community support and regular releases. Additionally, Hibernate is a standard implementation of the JPA specification, with a few additional features that are specific to Hibernate. Let's take a look at some core features of JPA and Hibernate.

**Spring Rest Controller-**

Spring Rest Controller annotation is used to create RESTful web services using Spring MVC. Spring Rest Controller takes care of mapping request data to the defined request handler method. Once response body is generated from the handler method, it converts it to JSON or XML response. Spring Rest Controller annotation is a convenience annotation that is itself annotated with @Controller and @ResponseBody. This annotation is applied to a class to mark it as a request handler.

**MODULE DESCRIPTION-**

**Login Module-**

The login module will have user id and password for the employee to login. In service layer we will have login, logout, we can add a new employee, we can edit an employee, and we can also remove an employee.

**Employee Module-**

The employee module will have employee id, employee name, department name, employee's location. In service layer we have the method to add an employee, method to edit an employee, method to get the employee details with the employee id, update is available method to check the offers, update fulfilled in which it will check whether the requirement is completed or not, update accepted method through which we can check if the employee's proposal is accepted or not, get all offer method with which we can get all the offers available with the employee id, get all requirement method with which we can get all the requirements with the help of the employee id.

**Requirement Module-**

This module is based upon the requirement of the employee whether the resources required i.e. product help or services are available or not. It also tells if the requirement is fulfilled and if yes which date it is fulfilled on. In this we will have requirement id, variable to check if the requirement is fulfilled or not, variable for checking the date of when it is fulfilled and a list of proposals available. In service layer, we will have methods to add requirements, edit requirements, get requirements with the use of id, remove a requirement, get list of all requirements and get all the requirements based upon the category and type.

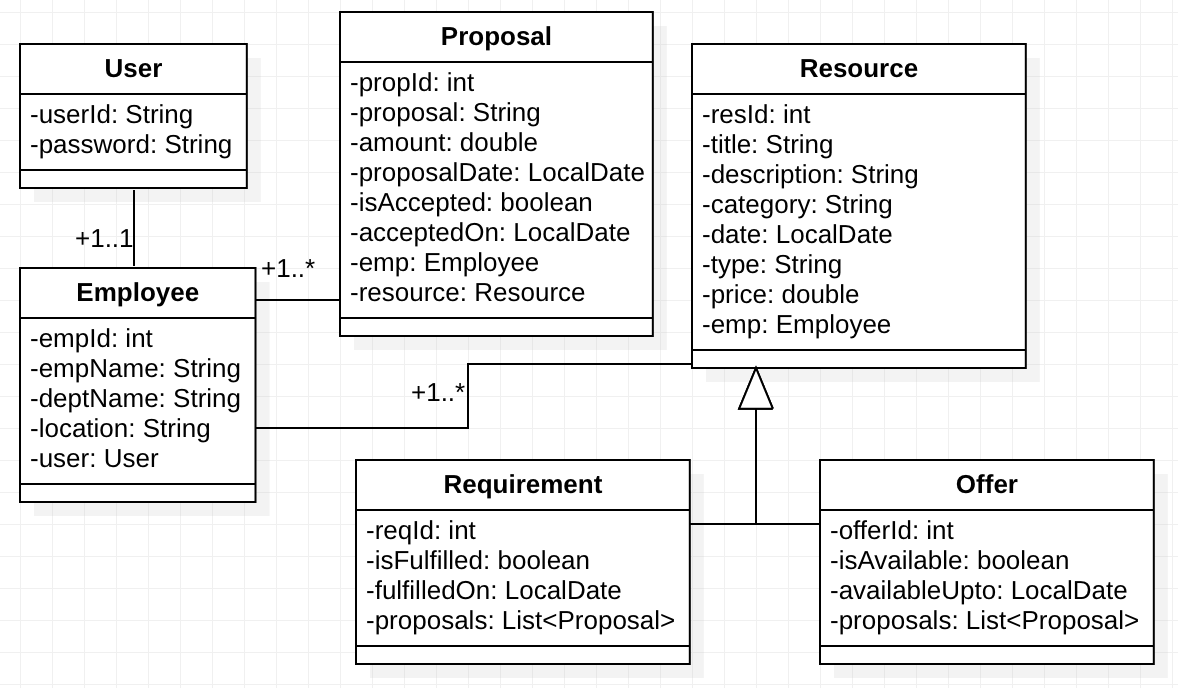
**Offer Module-**

This module is to check for the offers and has variable for offer id, variable to check the availablity of the offer, variable for the date of availablity and proposals for the list of proposals. In service layer we will have methods for add offers, edit offers, get offers, remove offer, get list of all offers and get all the offers based upon the category and type.

**Proposal Module-**

This module will have the proposals offered by different employees. It will have proposal id, proposal, amount of the proposal, proposal date, variable to check if the proposal is accepted, proposal acceptance date, employee class and resource. In service layer we will have methods for add proposal, edit proposal, get proposal, remove a proposal and get all the proposals.

**Class Design: POJO/Entity Classes**



**SOFTWARE REQUIREMENTS-**

* PostgreSQL
* Eclipse(J2EE)
* Swagger
* DevOps
* GitHub
* Jenkins
* Sonarlint