The Architecture of HandyMen Server

The HandyMen server is deployed in AWS cloud servers, with the main application deployed in EC2(Elastic Compute Cloud) and database in RDS (Relational Database Service). With AWS’s secure solutions and 99.99% availability, HandyMen server can run stably for 24 hours per day and 7 days per week. Also, with AWS’s load balancer and Elastic beans, the handy men server can be automatically scaled out to multiple instances during heavy load time after correct configuration in the AWS EC2 console.

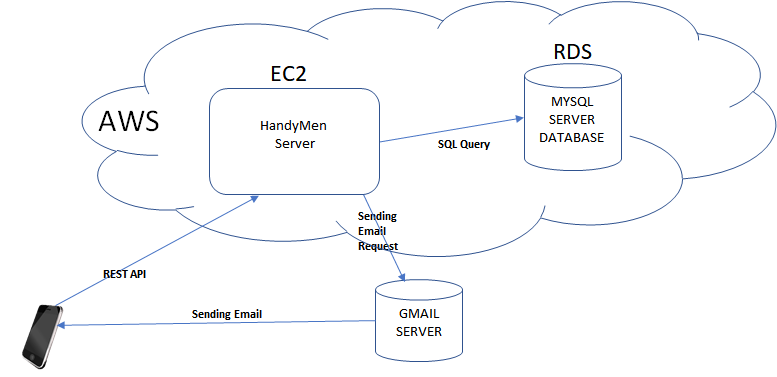


Diagram: HandyMen Server Deploy Diagram

The basic functions of HandyMen server consist of querying from database, providing REST APIs including file uploading and downloading, and sending email notifications. Accordingly, the components of HandyMen server involve the following parts. The internal components of the HandyMen server are HandyMen Service Module, Spring JDBC, Spring MVC Framework, Tomcat and Apache HttpServer. It also has two external components, MySql and Gmail Service. Spring framework is picked as the middle ware to handle the database connections and Java Servlet implementation. Spring framework is the most popular modern Java-based enterprise application framework. It hides the differences of various deployment environments and make application focus on its own business logic. Spring JDBC encapsulates most Java JDBC functions including connection establishment and destroy, SQL querying and result set parsing, which makes it easy and safe for applications to process various database operations without making mistakes of the JDBC layer. Its Spring MVC core is based on Model-View-Controller architecture pattern, providing different controllers and view templates for different types of web applications. The application can mainly concentrate on its model design and realization. In addition, it will be very convenient for the application to evolve based on this MVC architecture pattern. For example, if website is needed for the future product, it just needs to add a webpage view and use different controllers, without modifying any part about the model module. Also, using Tomcat and Apache HttpServer to make the processing more efficient. Apache HttpServer is responsible for static information requests, for example downloading files, while Tomcat handles all the dynamic information requests, such as REST APIs. The choice of MySql database is mainly because it is the most widely-used free relational database for web applications. For our project, the relational data is the most suitable one comparing to other NoSql databases. Regarding to mail notification module, since there is no specific email server for this project, GMAIL is chosen as the email server, which is pretty reliable and secure.

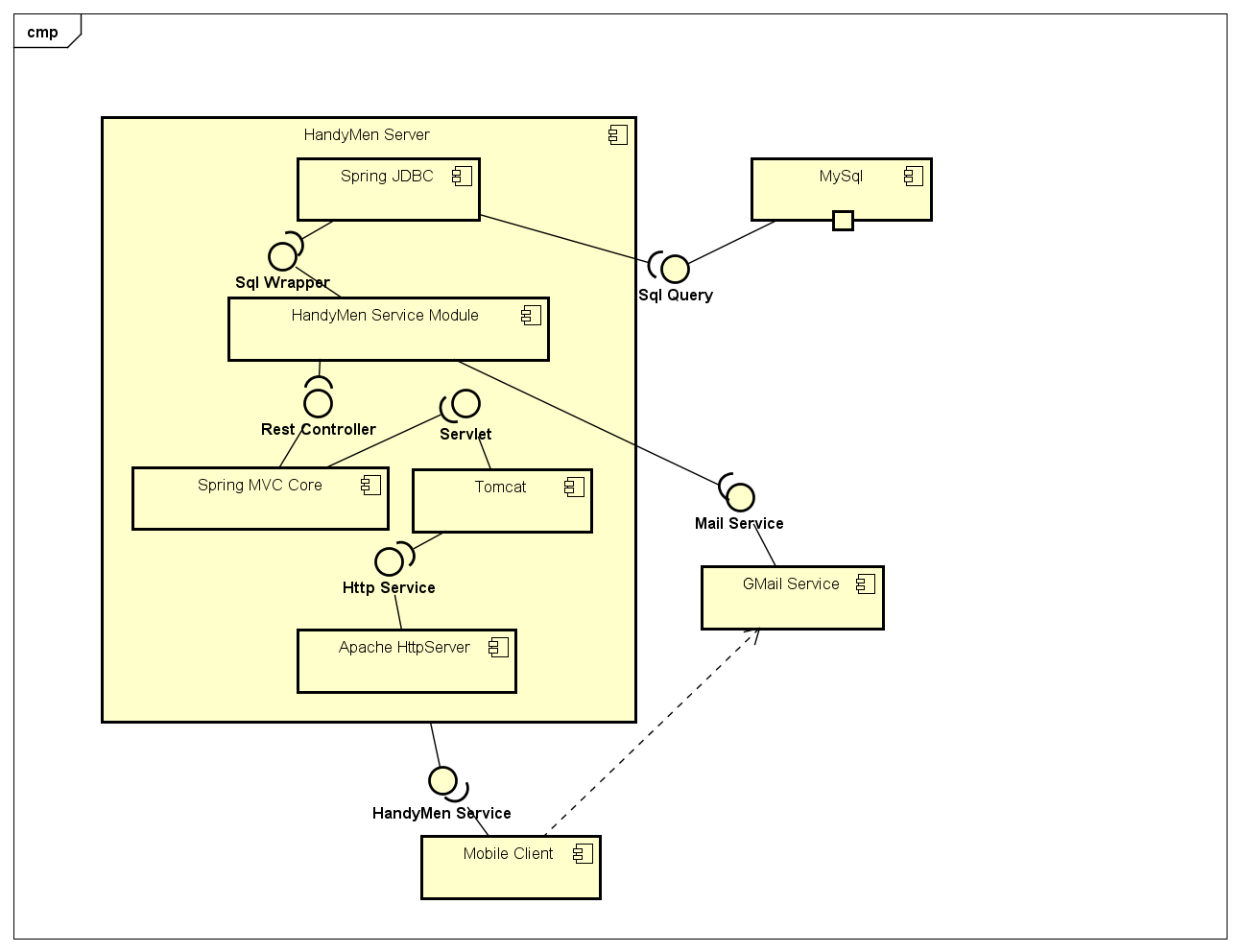


Diagram: HandyMen Server Component Diagram

The architecture types used in the server involves client-server style, pipe and filter style and interpreter style. The client-server style is about that one handy-men server can deal with multiple mobile application clients which may spread anywhere, and they use REST API to communicate. The Handymen Server use pipe and filter style to process different requests. For example, when receiving the user adding review request, the server will first check whether it is the valid user, whether the review already existing, and then put the review into database. The interpreter style is used by Spring MVC core to parse the URL parameters sent by the clients.

The design of Handymen Server

The Handymen server (here it specifically refers to the HandyMen service module in the HandyMen server component diagram which is deployed in the EC2 virtual machine), is based on Java Spring framework, providing services to Handymen mobile application clients, mostly through Rest APIs, which are secured by username and password in this system. Java Spring framework, as a light Java EE platform, is widely used in a large number of enterprises, and even becomes replacement for the Enterprise JavaBeans model in real industry. The Handymen Server integrates some core functions of Spring platform, such as MVC framework, IOC container and data access framework.

According to conventions of Java Spring applications, Handymen server is divided into domain module, service module and controller module. Domain module, basically the model layer in MVC pattern, is about all entity data and data access objects(DAO); service module includes some business code of the project; controller module in this project is Spring RestController, which updates the model data and transforms data from the model layer to JSON format message.

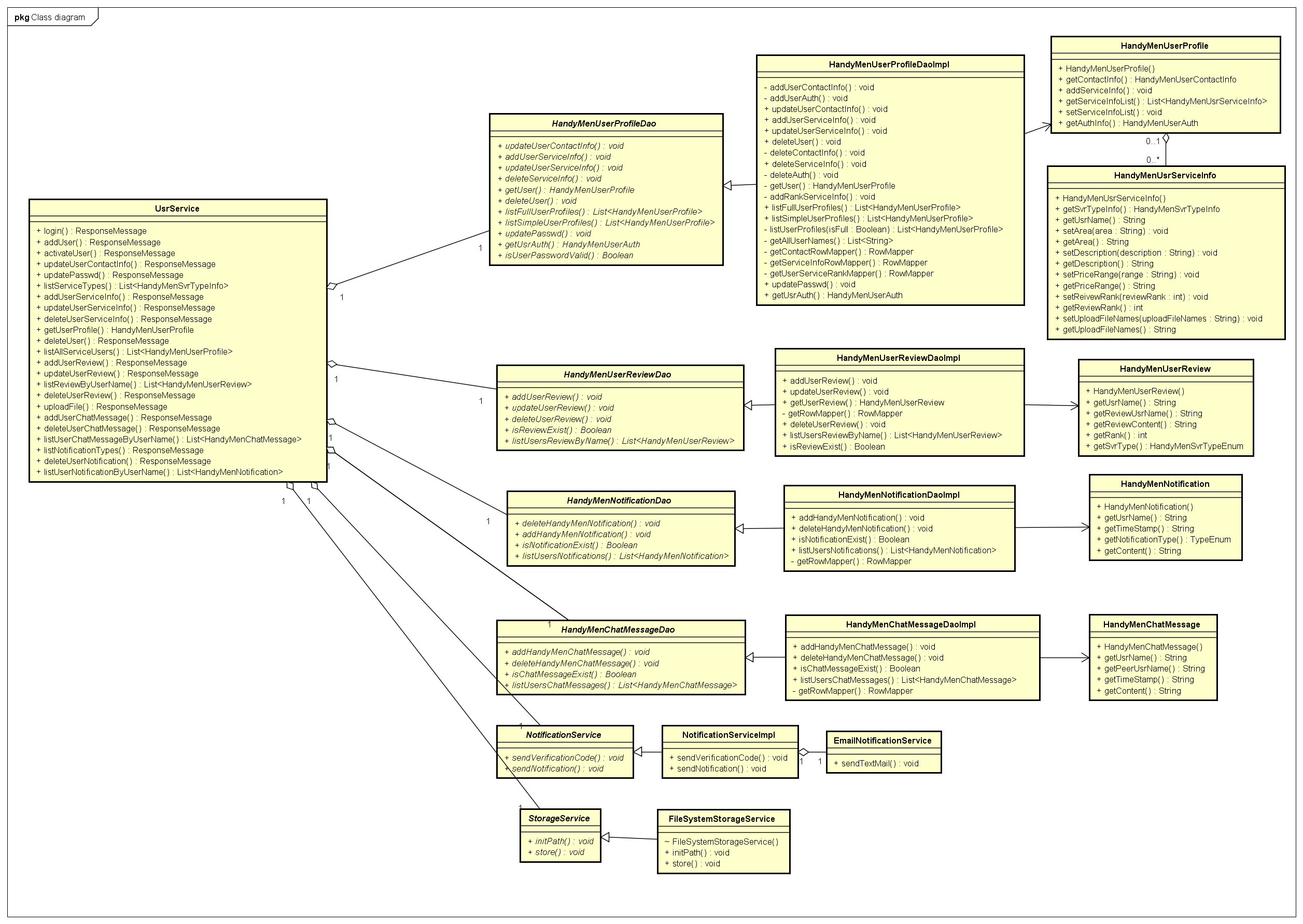


Diagram: HandyMen Server Class Diagram

The design of the classes in Handymen server is based on the domain, service and controller modules, and every class belongs to one of the modules. The relationship and connection of all the classes are displayed clearly in HandyMen Server Class Diagram. For the domain module, it has two types of classes. One type is the entity of HandyMen server, and the other is DAO, which is responsible for all operations of the database. The entity classes in this application mostly include *HandyMenUserProfile, HandyMenUsrServiceInfo, HandyMenUserReview, HandyMenNotification, HandyMenChatMessage,* etc. These classes represent different parts of the HandyMen application in the real world. Accordingly, for every entity class, a DAO class is needed to process the relevant operation for this entity. For example, class HandyMenUserProfile has its DAO class *HandyMenUserProfileDao*, which encapsulates the create, update, read and delete operations for the object handy user profile to the database. Based on SOLID principle, every class should depend on interface instead of concrete implementation. So, all the DAO classes are only interfaces, and implementation classes are generated for every DAO interface, such as class *HandyMenUserProfileDaoImpl for class HandyMenUserProfileDao.* For the service module, there are two service classes, NotificationService and StorageService. NotificationService sends email notifications by connecting to Gmail server; StorageService stores the uploaded files into the file system. The controller module only has one class in this system, UsrService, which provides all the Rest APIs to the clients by using Spring RestController, and updates the domain data by calling DAO functions.

The HandyMen server sequence diagram shows the interactions among different roles of HandyMen server when receiving client querying events. It includes CRUD operations for different objects, such as adding user, updating service information, getting user information and deleting user. Every sequence has almost the same process, which can be divided into several steps: client calling Rest APIs, user service calling functions of DAO or service module, DAO executing operations to the database, and then user service return JSON response messages to clients.

Based on Spring framework core technology, IoC (Inversion of Control, also known as dependency injection) container, all the dependencies in the HandyMen Server system are injected by Spring framework without direct construction of classes by using annotation @Autowired before dependencies. Spring frames provides several stereotype annotations: @Component, @Service, and @Controller, with which class will be treated as managed components by this platform. By default, the scope these autodetected components is singleton, which can be changed through @Scope annotation. In HandyMen server, all the components use the default singleton scope, which just meets the system’s requirement. By adding annotation @SpringBootApplication to the main application class, all components can be automatically scanned.

Because of the Spring MVC framework which decouples the modules, HandyMen server can evolve easily without too many modifications. For example, to support websites, it just needs to add a webpage view and use another controller. Also, using Spring JDBC makes the DAO decoupled from one specific database. If another type of database is imported into the system, according to Spring JDBC, only configuration is needed to support the new database.

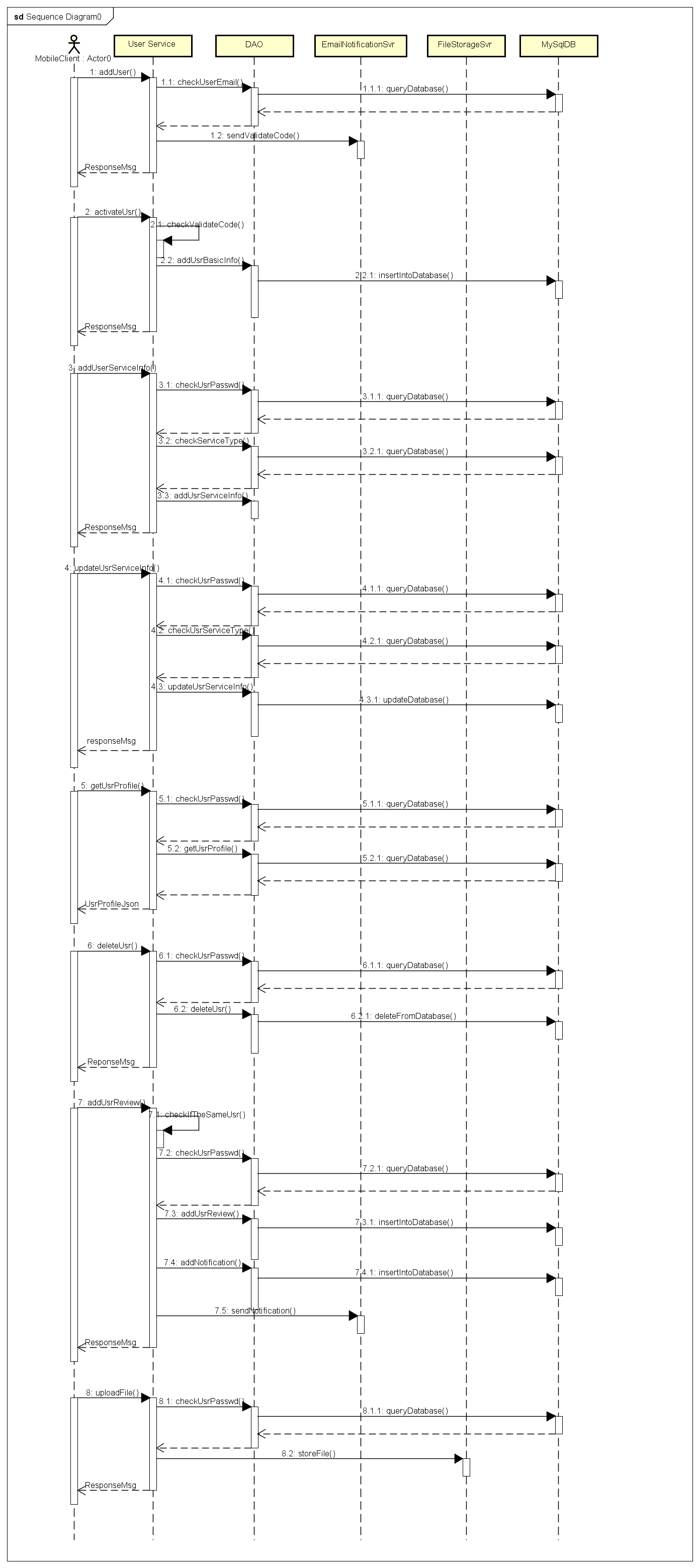


Diagram: HandyMen Server Sequence Diagram

About the future optimization of HandyMen server architecture, there are two points. The first one is to use Java Persistence API(JPA) provided by the Spring data access framework instead of Spring JDBC. JPA is a higher level of abstraction of data processing than JDBC, which transforms all SQL operations into objects. Another point is to use some memory cache system to improve the operating speed, when the query load increases. Memchache or Redis can be chosen in this scenario, and a new service class can be provided for this function.

The artifact of the HandeMen server can be obtained in this repository: <https://github.com/lnnn1982/handymenServer>.

There is also a sample HandyMen server RestAPI client repository: <https://github.com/lnnn1982/RestAPIQuerySample>.

Participation journal:

Responsible for all the server-end codes and documents. The githup commit records are as follows: https://github.com/lnnn1982/handymenServer/commits/master

