Contract Management System

Data Access Layer Development

Contents

- Function
- Design
- Implementation

Copyright Declaration

Function

Iterative development on the basis of Register (Business Logic Layer Development).

In this iteration, we'll complete Data Access Layer Development. Implement the storage of registered user data and fulfill the function of register by combining both View Layer Development and Business Logic Layer Development.

1. Develop the data access layer, to realize the storage of user data in t_user.

(1) Create isExist(), to check whether there is any record of registered user name in t_user.

(2) Create add(), to save user data in t_user.

2. Modify register() of the business logic layer in UserService class.

To process business in register(),

(1) Call isExist() of data access layer to check whether there are any identical user name registered.

(2) If not, call add() of data access layer to save the user data into database.

Design

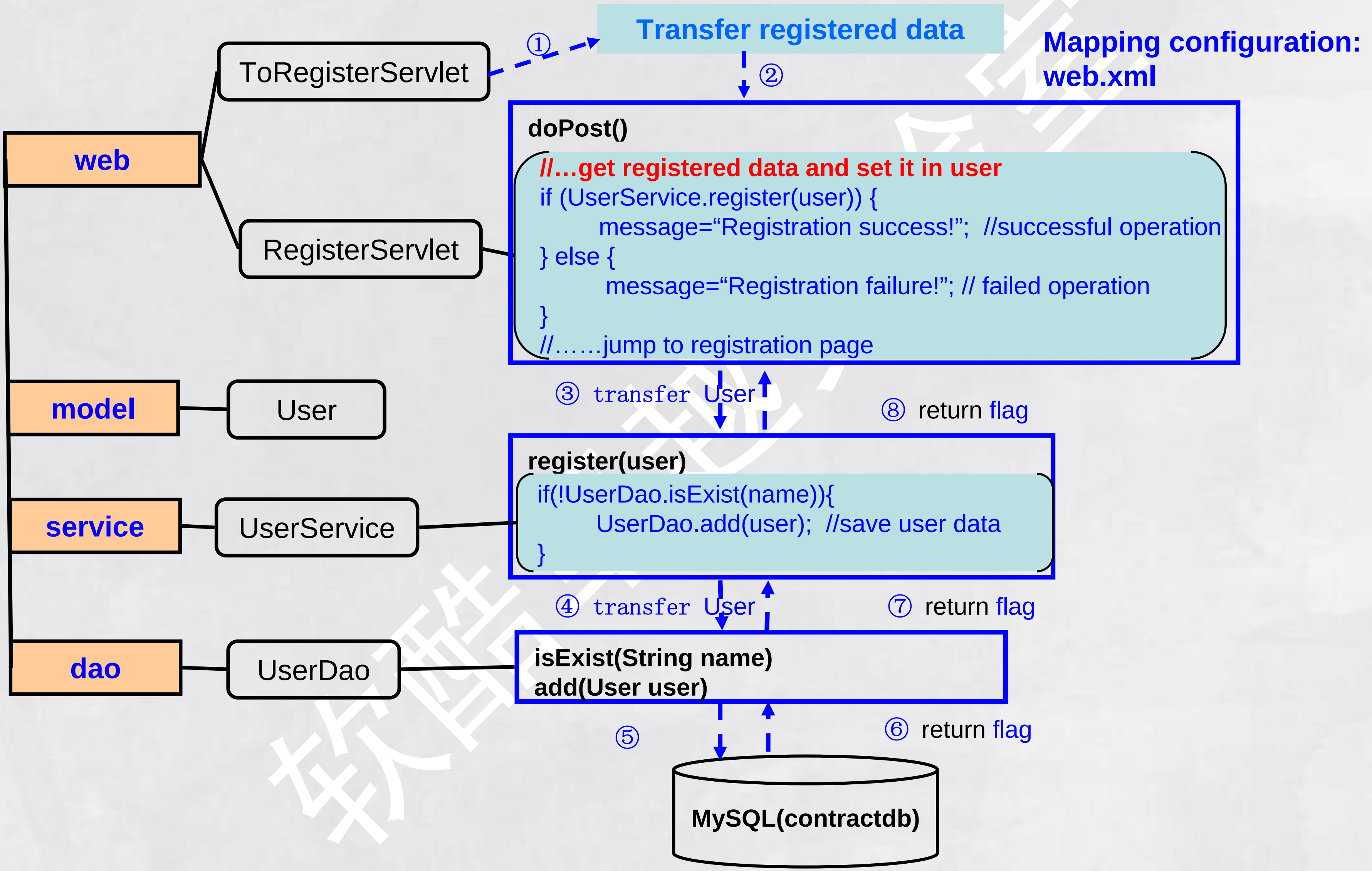
In this iteration, we’ll carry out data access layer development on the basis of business logic layer development.

(1) Code UserDao interface and UserDaoImpl to realize the storage of registered information.

(2) Call UserDao in UserService class to implement the function of register in combination with 3L development.

Design

1. The relationship of classes in different layers



```
doPost()
  //…get registered data and set it in user
  if (UserService.register(user)) {
          message="Registration success!";  //successful operation
  } else {
           message="Registration failure!"; // failed operation
  }
  //……jump to registration page
```

```
register(user)
 if(!UserDao.isExist(name)){
         UserDao.add(user);  //save user data
 }
```

```
isExist(String name)
add(User user)
```

Design

2. Data access layer: UserDao interface design, UserDaoImpl class design

(1) UserDao: user data access layer interface

Create UserDao interface in com.ruanko.dao, and define isExist() and add().

Method 1: public boolean isExist(String name) throws AppException;

Function: check whether the registered user name has been used.

Parameter: user name.

Return value: true for yes and false for no.

Method 2: public boolean add(User user) throws AppException;

Function: save registered user data.

Parameter: user object.

Return value: true for successful saving operation and false for failed.

(2) UserDaoImpl: implementation class in data access layer.

Create UserDaoImpl in com.ruanko.dao.impl to implement UserDao interface.

Implement isExist() and add() by adopting JDBC technology.

Design

3. Modify business logic layer class UserService

According the business logic, UserService class calls UserDao for database accessing to realize the operation of user name verification before saves it in database table for the implementation of register function.

Modify register() of UserService class for the logic processing of register.

(1) The instance of UserDaoImpl

It's needed to call data access layer method, so you need to declare data member userDao in class. userDao is the instance of UserDaoImpl, which is instantiated in constructor.

(2) Check the repetition of user name

Call isExist() of UserDao in data access layer to check if the name has been registered.

(3) Save registered information

If the name is not found registered in record, call add() of UserDao in data access layer to save it in database.

Implementation

Iterative development on the basis of Register (Business Logic Layer Development).

Firstly, carry out data access layer development. Implement the storage of registered user data.

Secondly, modify business logic layer. Process the logic of register and return processing result back to the view layer. According to the requirement of business logic, no two or more identical user names are allowed. That’s a full completion of register function.

Step 1, save user data.

Step 2, implement register.

Step 3, solve the problem of user name repetition.

www.ruanko.com

Thanks

Register