

Contract Management System

Business Logic Layer Development

Contents

- Function
- Design
- Implementation



软酷网
www.RuanKo.com

Copyright Declaration

Contents included in this document are protected by copyright laws. The copyright owner belongs to solely Ruankosoft Technologies (Shenzhen) Co., Ltd, except those recited from third party by remark.

Without prior written notice from Ruankosoft Technologies (Shenzhen) Co., Ltd, no one shall be allowed to copy, amend, sale or reproduce any contents from this book, or to produce e-copies, store it in search engines or use for any other commercial purpose.

All copyrights belong to Ruankosoft Technologies (Shenzhen) Co., Ltd. and Ruanko shall reserve the right to any infringement of it.

Iterative development on the basis of [Register \(View Layer Development\)](#).

In this iteration, we'll fulfill Business Logic Layer Development. To complete the logic processing of register, and to implement the transfer and display of register result.

1. Implement the processing of register logic in Business Logic Layer.

Business logic layer provides services for view layer. Register information is passed from view layer to business logic layer for further processing.

To implement register function, you need to fulfill steps described below.

(1) Business logic layer calls `isExist()` of data access layer to inquire if the user name has been used.

(2) If not, call `add()` of data access layer to save register data in database.

2. Call business logic layer in view layer to process the request.

(1) The view layer encapsulates register data in the object of User entity class.

(2) Call business logic layer to process the logic of register. And pass prompting message to the register page for display according to the result of processing.

To implement register function, call UserService of business logic layer in RegisterServlet of view layer, and call UserDao in business logic layer. Entity class User is used for data transfer between layers.

The function of register is implemented in an iterative way.

(1) The last iteration fulfills View Layer Development by creating the register page and processing register request.

(2) In this iteration, we realize Business Logic Layer Development.

Create UserService class in business logic layer. Implement the business processing of register by hard-coding in the class.

And then modify view layer. Call business logic layer to process register logic. And pass prompting message to the register page for display according to the result of processing.

(3) Next iteration, we'll fulfill Data Access Layer Development.

In data access layer, code UserDao interface and UserDaoImpl class to implement database accessing.

And then modify UserService class, call method of data access layer to realize the function of register.

1. Business procedure analysis

(1) Business procedure

Register information is last encapsulated in User entity and passed to the business logic layer for processing.

The business logic layer calls UserDao of data access layer to complete the operation of saving register information.

To register, you need to query user table t_user and insert data in it.

(2) The processing of business logic layer

The business logic layer processes register logic in register(). Call isExist() of UserDao in data access layer to query if the user name has been used. If not, call add() of UserDao in data access layer to save registered data in database.

2. UserService class design

Create the business logic class UserService in com.ruanko.service.

Prototype: `public boolean register(User user) throws AppException;`

Functional description: To process the business logic of register, call `isExist()` of data access layer to verify the repetition of registered data, and then call `add()` to save it.

Parameter description: User entity information.

Return value. True for success and false for failure.

```
if(not used){  
    save registered data  
}
```

3. Modify view layer classes

(1) Modify RegisterServlet

Modify doPost() of [RegisterServlet](#) class to process the request of register.

- 1) Encapsulate registered information in the object of User class.
- 2) Call [UserService.register\(user\)](#) of the business logic layer to process.
- 3) According to the processing result, set prompting message to the object of request and display it on register page by transmission.

(2) Modify ToRegisterServlet

Modify doGet() of [ToRegisterServlet](#) class for result display.

Get prompting message out of the object of request and display it over submission button of the form.

Iterative development on the basis of [Register \(View Layer Development\)](#).

Firstly, carry out business logic development to fulfill the processing of register. Next, modify view layer. Encapsulate registered information in the object of User and call business logic layer to process register logic. According to the result of processing, pass prompting message to register page for display.

Step 1, Create UserService and process register logic.

Step 2, View layer process register request.

Step 3, display register result.

www.ruankoweb.com

Thanks

Register