文档编号：《饭否》网上订单 app –SDS - 2.0

《饭否》网上订单 app
单元测试报告

日期：2020 年 11 月 1 日

文档变更历史记录

序号	变更日期	变更人员	变更内容详情描述	版本
1.0	2020\11\1	司马晨	测试了 UserService 里的 login 和 register 方法	v1.0

1.整体计划：选取合适的测试方法测试注册和登录程序；
2.测试评价标准：覆盖广度，测试结果与预期输出一致。
3.测试核心代码：

```
package com.fanfou.food.service.impl;

import com.fanfou.food.controller.form.LoginForm;
import com.fanfou.food.controller.form.RegisterForm;
import com.fanfou.food.dao.entity.UserEntity;
import com.fanfou.food.dao.repo.UserRepository;
import com.fanfou.food.domain.User;
import com.fanfou.food.exceptions.AuthException;
import com.fanfou.food.service.IUserService;
import com.sun.xml.internal.ws.policy.AssertionSet;
import org.junit.jupiter.api.Assertions;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.function.Executable;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.test.context.ActiveProfiles;

import java.util.Optional;

import static org.junit.jupiter.api.Assertions.*;

/**
 * @author smc
 * @date 2020/10/27 19:07
 */
@SpringBootTest
@ActiveProfiles("test")
class UserServiceImplTest {
    @Autowired
    private IUserService userService;
    @Autowired
    private UserRepository userRepository;


    /**
     * 测试注册成功
     */
    @Test
    void register() {
```

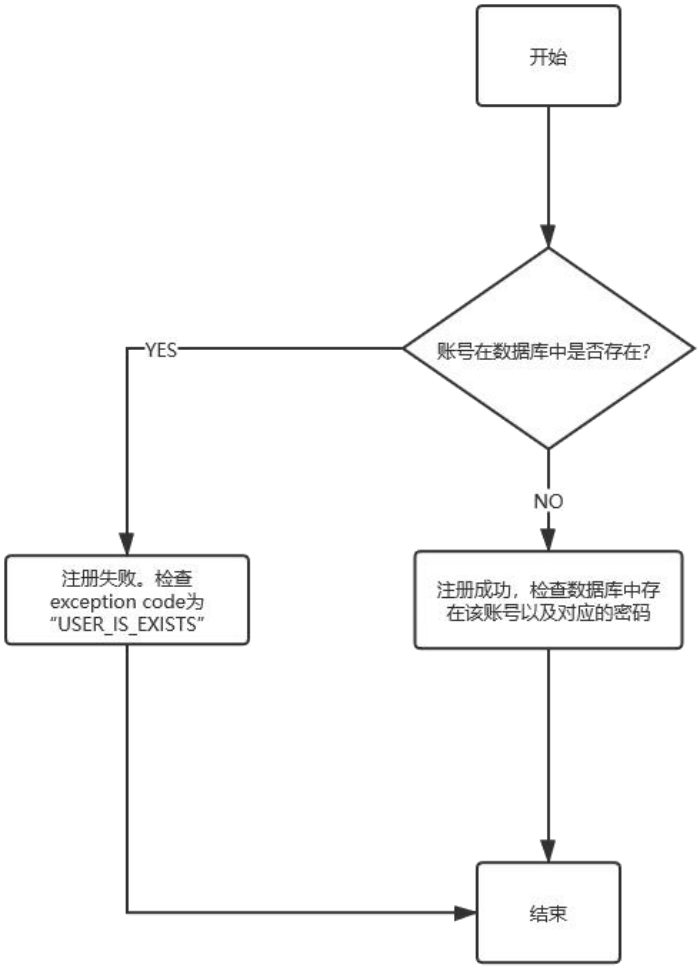
```
        String username = "abc";
        String password = "123";
        RegisterForm registerForm = new RegisterForm();
        registerForm.setUsername(username);
        registerForm.setPassword(password);
        try {
            userService.register(registerForm);
        } catch (AuthException e) {
            e.printStackTrace();
        }
        Optional<UserEntity> optionalUserEntity =
this.userRepository.findUserEntityByUsername("abc");
        assertTrue(optionalUserEntity.isPresent());
        assertEquals(optionalUserEntity.get().getUsername(), "abc");
    }
    /**
     * 测试注册重复用户名
     */


    @Test
    void register2() {
        String username1 = "abc2";
        String password1 = "1232";
        String username2 = "abc2";
        String password2 = "1232";
        RegisterForm registerForm1 = new RegisterForm();
        RegisterForm registerForm2 = new RegisterForm();

        registerForm1.setUsername(username1);
        registerForm1.setPassword(password1);

        Assertions.assertDoesNotThrow(new Executable() {
            @Override
            public void execute() throws Throwable {
                userService.register(registerForm1);
            }
        });

        registerForm2.setUsername(username2);
        registerForm2.setPassword(password2);

        Optional<UserEntity> optionalUserEntity =
this.userRepository.findUserEntityByUsername("abc1");
```

```
        assertTrue(optionalUserEntity.isPresent());
        assertEquals(optionalUserEntity.get().getUsername(), "abc1");


        try {
            userService.register(registerForm1);
        } catch (AuthException e) {
            Assertions.assertEquals(e.getCode(),AuthException.USER_IS_EXISTS);
        }



}

@Test
/*
登录成功
 */
void login1() {
        String username = "abc";
        String password = "123";
        LoginForm loginForm = new LoginForm();
        loginForm.setUsername(username);
        loginForm.setPassword(password);
        try {
            userService.login(loginForm);
        } catch (AuthException e) {
            e.printStackTrace();
        }
}
@Test
/*
用户名不存在
 */
void login2() {
        String username = "a";
        String password = "123";
        LoginForm loginForm = new LoginForm();
        loginForm.setUsername(username);
        loginForm.setPassword(password);

        try {
            userService.login(loginForm);
        } catch (AuthException e) {
```

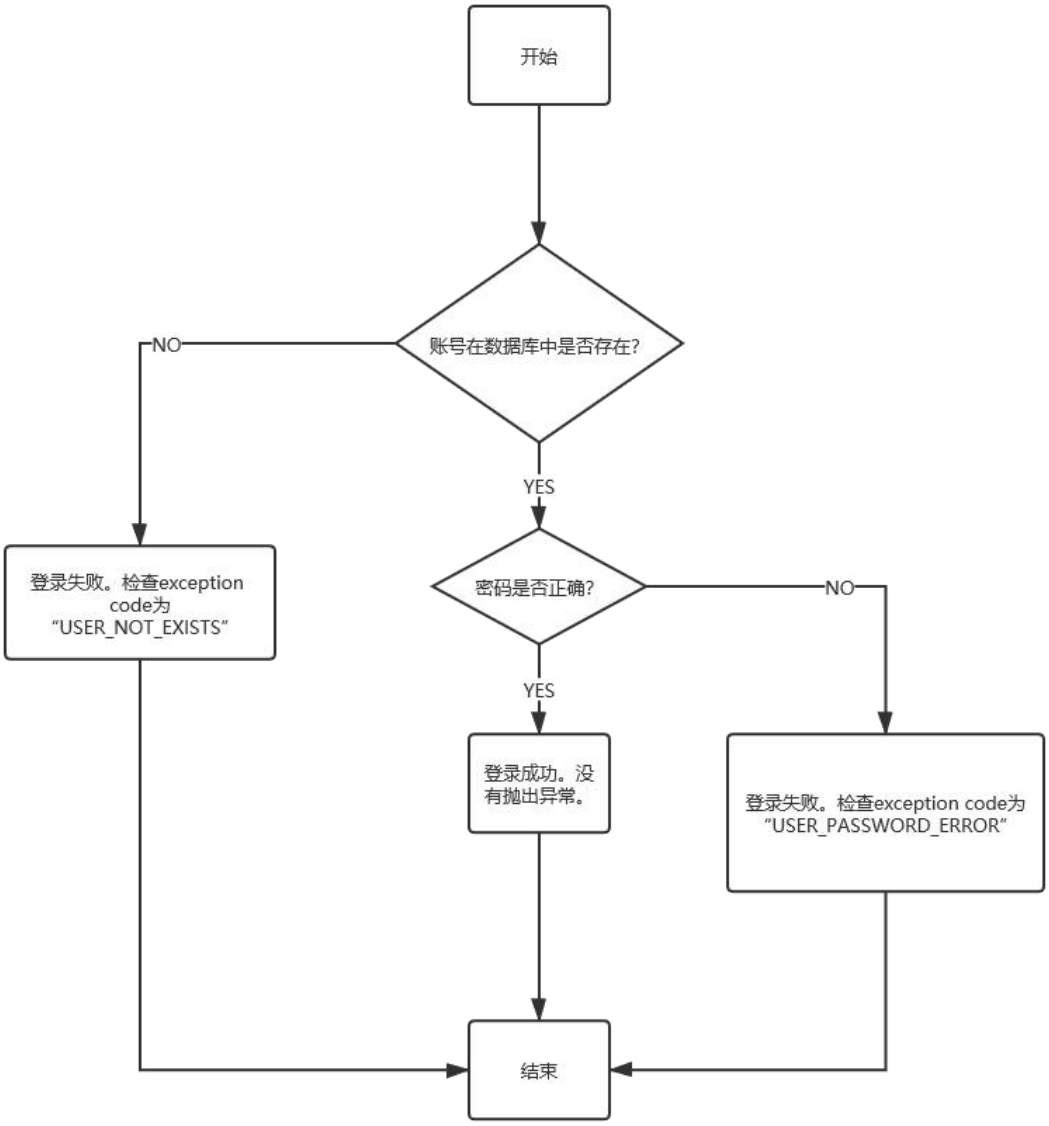
```
            Assertions.assertEquals(e.getCode(),AuthException.USER_NOT_EXISTS);
        }
    }

    @Test
    /*
    密码错误
     */
    void login3() {
        String username = "abc";
        String password = "111";
        LoginForm loginForm = new LoginForm();
        loginForm.setUsername(username);
        loginForm.setPassword(password);

        try {
            User user=userService.login(loginForm);
            Assertions.assertEquals(user.getName(),"abc");
        } catch (AuthException e) {
            Assertions.assertEquals(e.getCode(),AuthException.USER_PASSWORD_ERROR);
        }
    }

}
```

4. 测试评价标准：覆盖广度，测试结果与预期输出一致。

5. 根据代码绘制流程图与有向图：



5.1 register 方法测试流程图



5.2 login 方法测试流程图

6. 测试用例设计

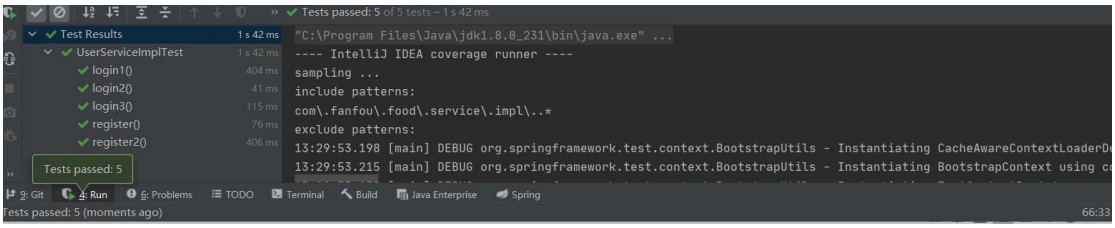
注册功能测试：

用例编号	用户名	密码	预期结果
1	abc	123	注册成功
2	abc1、abc1	123、123	注册失败。失败code：USER_IS_EXSITS

登录功能测试：

用例编号	用户名	密码	预期结果
1	abc	123	登录成功
2	a	123	登录失败。失败 code：USER_NOT_EXSITS
3	abc	111	登录失败。失败 code：USER_PASSWORD_ERROR

7. 测试结果



可以发现，所有测试结果和预期相符。测试基本完成。