

# Writing the Unit Test Scripts

SE 323 Software Construction Testing and Maintenance

# What is Unit Test

- Test in the smallest unit of source code
- Normally the method

# There are two types of method in unit test

- Method with no dependency
- Method with dependency

# Method with no dependency

- No other methods is called inside the code

# Code for test

- Every method should return output
- The Unit test can be done automatically
- No void methods

# Writing the test for method without dependency

- Simple
- Just put the input, and expected output
- Test environment may be required

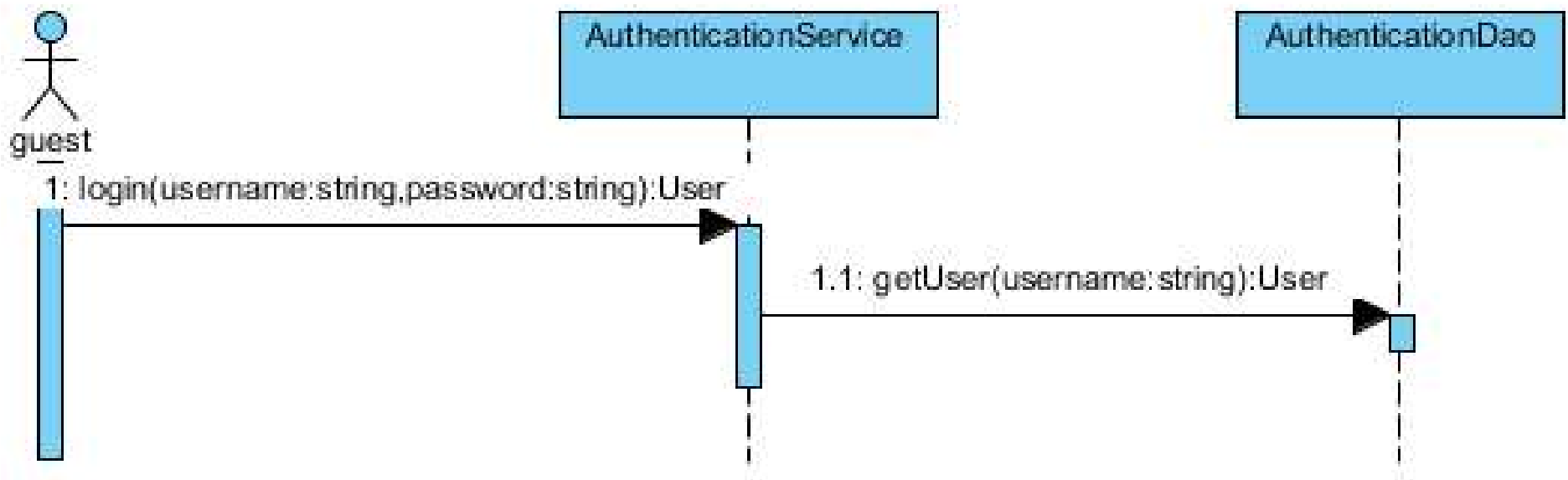
# Method description

- `getDiscount(dob:Date):double`
- The `getDiscount` method will receive the birthdate of the customer as the date object in the `dob` object, and then calculate the age of the customer from today. The method will return the discount percent from the age. The maximum discount rate is 100 percent
- If the age is less than 7 years old, they can get the maximum discount
- If the age is less than 14 years old, they can get 50 percent discount
- And If the age is 60 years old, they will get initial 10 percent discount, and plus 2 percent discount for each year which greater than 60 years old.

Write the test case of the given method



# The method with dependency



## In this case

- `getUser(username:string):user`
  - The method receives the username and then search for the database
  - If the user table contains the input username, the method return the user object which contains the given username
  - Otherwise return null

| User               |
|--------------------|
| -username : string |
| -password : string |
| -name : string     |
| -surname : string  |
| -id : Long         |

# Next method

- `login(username: string, password: string): User`
  - The login method is used for searching for authorization. The method call `getUser` using the given username to find the user object.
  - If the method `getUser` return the User object, the method will check that the given method contains the same password attribute as input password or not. If the password is the same, the login method return the User object, otherwise return null

# Unit test

- Must be independent
  - Not relied on any other implementation
- But our code requires dependency

# MockObject

- Mock object
  - The object that represent the implementation
  - Use the same signature
  - Return what we want
  - The inheritance /polymorphism can be used

# To write the unit test

- Provide the mock object first
- Provide the data to be return
  - Regarding to input

# Mock object for getUser(username:String):User

| Username | Return  |
|----------|---|
| Admin    | {“username”:”Admin”,”password”:”1234”,<br>”name”:”admin”,”surname”:”yesSir”,”id”:<br>500} |
| User1    | null  |

# Then create test case

- Using the provided mock data



# Assignment

- ~~2 works~~
- ~~Submit Friday Feb 14~~
- ~~I will give you two requirements specification, and two design document specification~~
- ~~For UAT test script, select two use cases and create the UAT test~~
  - ~~The login is not accepted~~
- ~~For unit test (discuss next class), select two methods to create the Unit test~~
- ~~Printed and submit with your student name, and ID~~
- ~~Individual work~~

QA