

## Testsvit

### Testmål

Målet är att kontrollera att enskild funktion i systemet fungerar utifrån de krav som presenterats i användningsfallen. Målet är också att hitta och säkerställa nya buggar som kan hittas i systemet jag skrivit.

Klasserna jag testa ska vara implementerade i ett mindre system jag skapat utifrån användningsfallen.

Följande 2 klasser här ner har jag valt att testa,

1. Klassen control
2. Klassen register

## Klassen control

### Testfixtur

logincontrol()

### Scenario 1

1. Konstruktorn i klassen Control kallas med två värden "control(name, password); "
2. När name och password har fått värden så kallas logincontrol()
3. logincontrol()tittar igenom medlems registret och godkänner till gång till systemet om användare finns, Logincontrol funkar.

### Alternativ scenario 1

1. Konstruktorn i klassen Control kallas med två värden "control(name, password); "
2. När name och password har fått värden så kallas logincontrol()
3. logincontrol()tittar igenom medlems registret och nekar till gång till systemet om användare inte finns, Logincontrol hantering av fel funkar.

| Testfall 1                          | Testfall 2   |
|-------------------------------------|--|
| <b>Fins i systemet</b>              | <b>Fins inte i systemet</b>  |
| användarnamn: Marco<br>password:123 | användarnamn: Zom<br>password:133  |
| logincontrol()                      | logincontrol()   |
| Du är inloggad                      | du fins inte i systemet,<br>kontrollera att<br>inloggnings namn och<br>pwassword är rätt |

## Testfixtur

add()

### Scenario 1

1. lägger till ny användare i medlems register
2. add(name, password);
3. medlemsregister skrivs ut med nya användaren registrerad

| Testfall 1   |
|--|
| <b>Lägga till ny användare i medlems register</b>              |
| användarnamn: Zom<br>password:133                              |
| add(Zom, 133);   |
| medlemsregister skrivs<br>ut med nya användaren<br>registrerad |

## Testfixtur

Load()

### Scenario 1

1. kalla på load()
2. skriv ut medlems register, load funkar

| Testfall 1                       |
|----------------------------------|
| <b>Ladda in medlems register</b> |
| Load()                           |

Skriv ut hela medlems  
register, load funkar

## Testfixtur

Save()

### Scenario 1

1. åter skapar funktionen add(); i test klassen för att isolerar bara Save() funktionen
2. kalla på Save()
3. kalla på Load()
4. skriv ut medlems register
5. ny användare finns i medlems register, save funkar

| Testfall 1                                       | Testfall 1   |
|--|--|
| <b>spara ny användare i<br/>medlems register</b> | <b>Ladda in medlems<br/>register</b>                     |
| användarnamn: R2D2<br>password:1905              | Load()   |
| b1.namen.Add(n);<br>b1.password.Add(p);          | Skriv ut hela medlems<br>register                        |
| Save()   | R2D2<br>1905<br>finns i medlems<br>register, Save funkar |

### Klassen Regist

#### Testfixtur

namnkontrol()

#### scenario 1

1. kalla på funktionen namnkontrol() som finns i klassen register
2. mata in namne
3. om namne har mellanslag, får tillbaks värdet "true"

#### Alternativ scenario 2

1. kalla på funktionen namnkontrol() som finns i klassen register
2. mata in namne
3. om namne inte har något mellanslag, får tillbaks värdet "false"

### Resultat

| Testfall 1   | Testfall 2   |
|--|--|
| <b>Kontrollerar<br/>mellanslag i nytt<br/>namn</b> | <b>Kontrollerar<br/>mellanslag i nytt<br/>namn</b> |
| användarnamn: R 2 D 2                              | användarnamn: R2D2                                 |
| namne har mellanslag,<br>fel meddelande            | namne har inte<br>mellanslag                       |
| får tillbaks värdet<br>"true"                      | får tillbaks värdet<br>"false"                     |

### Testfixtur

passwordkontrol()

### Scenario 1

1. kalla på funktionen passwordkontrol() som finns i klassen register
2. mata in lösenord
3. om lösenord har mellanslag, får vi tillbaks värdet "true"

### Alternativ scenario 2

1. kalla på funktionen passwordkontrol() som finns i klassen register
2. mata in lösenord
3. om lösenord har mellanslag, får vi tillbaks värdet "false"

| Testfall 1   | Testfall 2   |
|--|--|
| <b>Kontrollerar<br/>mellanslag i nytt<br/>lösenord</b> | <b>Kontrollerar<br/>mellanslag i nytt<br/>lösenord</b> |
| password:1 9 0 5                                       | password:1905  |
| lösenord har<br>mellanslag, fel<br>meddelande          | lösenord har inte<br>mellanslag                        |
| får tillbaks värdet<br>"true"                          | får tillbaks värdet<br>"false"                         |

### Testfixtur

bekrefta(string nyn, string nyp)

### Scenario 1

1. kalla på bekräfta
2. medlande med alternativ visas, Registrera användare, svar: Ja
3. medlande: "du är nu registrerad" visas och du tas till start menyn

### Alternativ scenario 1

1. kalla på bekräfta
2. medlande med alternativ visas, Registrera användare, svar: Nej
3. medlande: "registrering stoppad" visas och du tas till start menyn

### Resultat

| Testfall 1           | Testfall 2                |
|----------------------|---------------------------|
| Godkänn registrering | Inte Godkänn registrering |
|                      |                           |

|                                     |                                     |
|-------------------------------------|-------------------------------------|
| användarnamn: R2D2<br>password:1905 | användarnamn: R2D2<br>password:1905 |
| Registrera användare,<br>svar: Ja   | Registrera användare,<br>svar: Nej  |
| Du är nu registrerad                | Registrering stoppad                |

### **Motivering till testsviterna och testfallen**

Varje funktion i klasserna ska testas för att försäkras oss om att klassen control och Klassen Regist funkar i sin helhet och att användningsfallens flöden är testade. När testfallen har testats och de förväntade resultaten uppnåtts. Kan klasserna fastställa som fungerande och klarar för att implementeras i gymnastiktävlingssystemet.