

**1DV404 – HT2014**  
**Iterativ mjukvaruutveckling**  
**Laboration 3 - Testning**

**av Da-Sing Trinh**  
**@dt222cc**

Planering
-----------

Uppgift	Beskrivning	Planerad tid HH:MM
1	Planering	00:30
2	Testplan	03:00
3	Design och implementation	06:00
4	Enhetstestning	04:00
5	Implementera testsviten och kör	04:00
6	Integrationstestning	03:00
8	Reflektion	01:00
	Sammanställa dokument	00:30

Summering	22:00
-----------	-------

Tidslog
---------

Datum	Uppgift	Beskrivning	Förbrukad tid HH:MM
11.12.2014	1	Planering	00:20
12.12.2014	2	Mycket förstudier, förvirrad över hur mycket jag ska ha i dokumentet. Saknar kanske material. Kommer att komplementtera dokumentet mera längre fram.	03:30
13.12.2014	3	För mycket tid lade ner här. Det var väldigt svårt och detta hade jag planerat in. Fick göra om en hel del, göra ändringar på användningsfallen. Det var svårt att implementera ett helt funktionellt kod. Jag har förstått tanken med detta men som jag nämde var det mycket svårt att arbeta med detta. Det känns som jag har missat en del.	07:00
13.12.14	3	Jag gjorde om uppgiften. Jag bytte användningsfall till ett "enklare". Kodade i JavaScript (det strulade med visualstudio).	03:00
Summering			

Version	Datum	Testplan	Författare
1.1	2014-11-12		dt222cc

## 1. Introduktion

Dokumentet ska specificera vilka tester som ska utföras och hur de ska genomföras. Dokumentet kommer att fungera som en referens för att visa att de ställda kraven har uppfyllts.

### **Bakgrund**

Testerna ska se till att kunden får en produkt som har den kvalitet och prestanda som förväntades att produkten ska ha. Testerna som ska genomföras är till för att säkerställa att de krav som har ställts på systemet ska hålla.

## 2. Översikt

Dokumentet kommer att definiera är:

- Omfattning, fokusområden och mål.
- Vilka tester som ska utföras och hur de ska genomföras.
- Ansvarsområden.
- Den teststrategi och typer av tester för den här versionen.
- Eventuella risker, frågor, antaganden och test beroenden

## 3. Testmål

Testplanen har som mål att definiera de alla områden på ett tydligt sätt. Alla krav ska verifieras och riskerna minimerade.

Det som testerna kommer att innefattas av är:

- Funktionella krav
- Prestanda krav
- Säkerhets krav
- Use-Case krav

## 4. Teststrategi

De aktuella Use-Casen samt de krav som visionen och den kompletterande/supplementerande specifikationen tar upp kommer att användas för att driva testerna.

Systemet ska testas genom att utsätta den för olika fall som den kan stöta på och måste kunna hanteras med. Testplanen är strukturerad så att alla test som ska utföras på en enhet ligger tillsammans.

Detta ska förenkla jobbet med att se vilka tester som ska utföras på varje enhet. Meningen med strukturen är att den lätt ska kunna matchas mot kravspecifikationen.

#### 4.1 Testens presentation

I detta dokument kommer alla tester att beskrivas med en tabellrad enligt nedan.

Test nr x	Testansvarig	Beskrivning av test	Deadline
-----------	--------------	---------------------	----------

#### 4.2 Misslyckade tester

Vid ett misslyckat test ska resultaten analyseras för att ta reda på vad som gick fel vid testet, åtgärder ska vidtas och sen ska ett nytt test utföras. Felrapportering ska också utföras.

Visar det sig att ett test för att verifiera ett krav är omöjligt att utföra, ska ett gemensamt beslut mellan projektgrupp, beställare och kund tas, så att det kan ändras eller till och med strykas.

### 5. Testomgivning

[...]

### 6. Modultester

[...]

### 7. Integrationstester

[...]

### 8. Systemtester

[...]

### 9. Resurser

Roll	Kontakt uppgifter	Ansvar/uppgifter
System testare	[namn] [e-mail adress] [telefon/mobilnummer]	Utför tester, rapporterar/dokumenterar resultat.

### 10. Planering/schema

[...]

### 11. Referenser

<http://epf.eclipse.org/wikis/openup/>

<http://www.softwaretestinghelp.com/test-plan-sample-softwaretesting-and-quality-assurance-templates/>

<http://www.computing.dcu.ie/~davids/courses/CA267/ieee829mtp.pdf>

<http://www.isy.liu.se/edu/projekt/tsrt71/2007/ikp/Testplan%20version%201.0.pdf>

<http://www.adelaide.edu.au>

Version	Datum	Klassbeskrivningar	Författare
1.1	2014-12-12	För UC7– Registrera gren (avregistrera gren)	dt222cc

## Flöden – Registrera gren

UC

Initiering: Användaren väljer att registrera en ny gren.

Pre: Deltävlingen registrerade. Användaren (administratör) är inloggad i systemet.

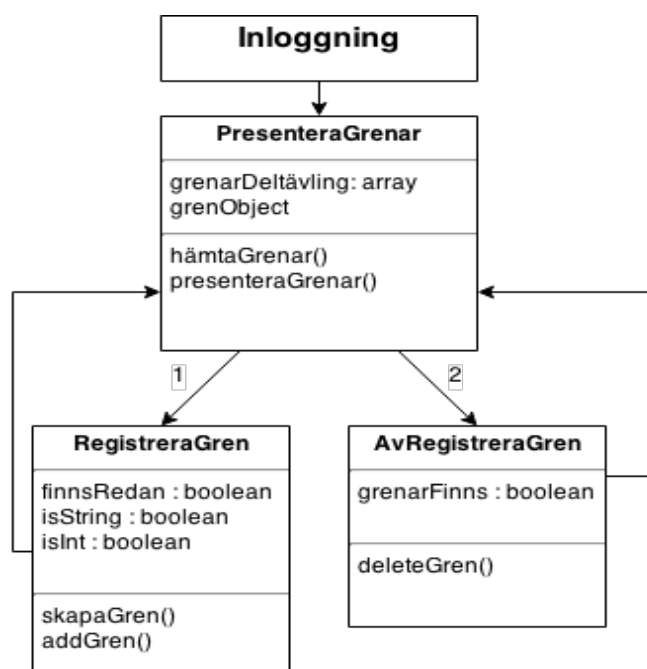
Primärt flöde

1. Aktören väljer i vilken deltävling som det ska läggas till en ny gren.
2. Systemet presenterar ett formulär med namn för grenen, och antal deltagare som det ska vara för grenen.
3. Den nya grenen läggs till i en lista med alla grenar för deltävlingen.

Alternativ flöde

1. Grenen finns redan
2. Användaren måste mata in ett heltal för antal deltagare.
3. Användaren väljer att avbryta

## Designelement



## Kod

Jag är väldigt osäker på hur detta skulle ske. Det skulle ta för mycket tid för mig att försöka implementera kod pga brist på erfarenheter (tycker jag). Jag har en tanke på hur proceduren ska ske men jag har det svårt att implementera dem som jag nämnde. Jag vet inte om detta är tillräckligt eller inte. Körde på javascript.

```
"use strict";

window.onload = function() {
    new Grenar();
};

function Grenar() {
    this.grenObject = {};
    this.init();
}

Grenar.prototype.init = function() {
    console.log("Initierar programmet");

    this.skapaGren();
};

Grenar.prototype.skapaGren = function() {
    var number = document.getElementById("number");
    var name = document.getElementById("name");
    var size = document.getElementById("size");
    var button = document.getElementById("register");

    var that = this;

    button.onclick = function() {
        that.addGren(number.value, name.value, size.value);
    };
};

Grenar.prototype.addGren = function(number, name, size) {
    var content = document.getElementById("grenID");
    var li = document.createElement("li");
    var dt = document.createElement("dt");
    var dd = document.createElement("dd");

    li.id = "gren" + number;
    dt.innerHTML = number + " " + name;
    dd.innerHTML = "Antal deltagare: " + size;

    li.appendChild(dt);
    li.appendChild(dd);
    content.appendChild(li);

    alert("Grenen " + name + " är nu skapad.");
};
```

```
// funktionalitet som saknas:  
// 1. verifiering av de värden som skickas:  
//     istället för att ange nummer så görs det automatiskt genom att alla grenObjekt sparas i en array.  
//     att namn är en sträng (och det första bokstaven är versal)  
//     att antaldeltagare är en nummer  
// 2. avregistrering av grenar (lik labb 2 labbymezzage från idv403 med en knapp)  
// 3. tömma fälterna efter varje registrering  
// 4. samla grenObjekter i en array, rendera alla objekt från array till dokumentet
```