

ITERATION 1

ETT LAG VILL ANMÄLA SIG TILL EN TRÄFF

SPECIFICERAD PLANERING

KRAV	PLANERAD TID	VERKLIG TID
Analysera användningsfall	30 Minuter	15 Minuter
Specificera	30 Minuter	30 Minuter

DESIGN / IMPLEMENTATION	PLANERAD TID	VERKLIG TID
Identifiera klasser	30 Minuter	30 Minuter
Specificera klasser	45 Minuter	60 Minuter
Implementation	1 Timma 15 Minuter	1 Timma 45 Minuter

TEST	PLANERAD TID	VERKLIG TID
Test design	30 Minuter	45 Minuter
Test specifikation	45 Minuter	45 Minuter
Test implementation	1 Timma 30 Minuter	1 Timma 30 Minuter
Test exekvering	45 Minuter	45 Minuter

REFLEKTION	PLANERAD TID	VERKLIG TID
Reflektera	1 Timma	45 Minuter

TOTAL PLANERAD TID: 8 TIMMAR

TOTAL VERKLIG TID: 8 TIMMAR 30 MINUTER

DESIGN OCH IMPLEMENTATION

FLÖDESSCHEMA

PRE-Villkor

1. Laget skall ha ett registrerat användarkonto i systemet.
2. Laget skall vara inloggad i systemet.

POST-Villkor

1. Verifikation med detaljspecifikationer rörande tid och plats har skickats till lagets kontaktinformation.
2. Laget är anmäld till träffen och inskickat formulär ligger lagrat i systemet.

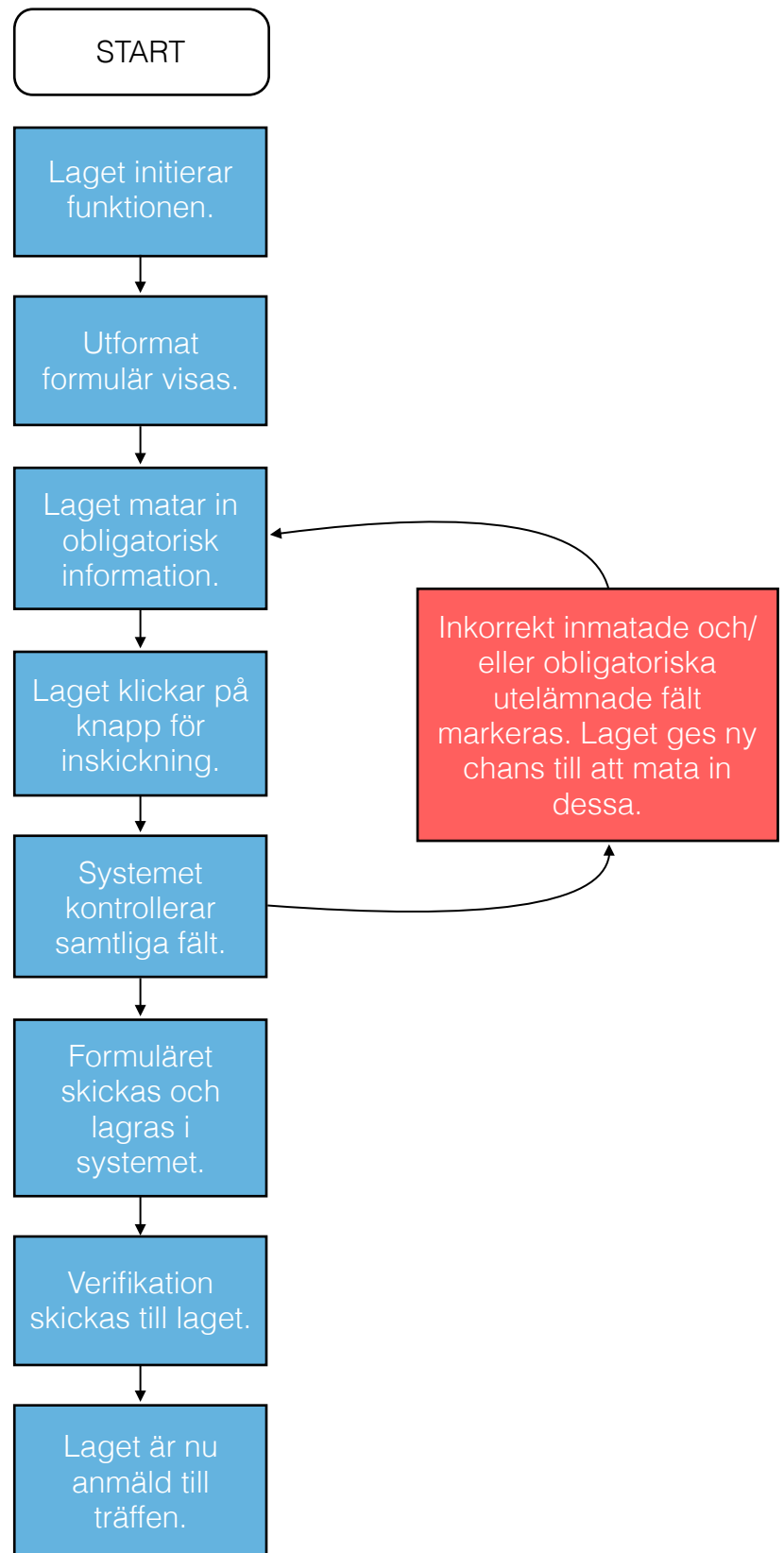
PRIMÄRT FLÖDE

1. Laget navigerar sig till och initierar funktionen för anmälning till träff.
2. Utformat formulär för inmatning av lagnamn och gymnaster med tillhörande namn, ålder samt kön visas.
3. Laget matar in obligatorisk information.
4. När samtlig information är inmatad klickar laget på knapp för inskickning.
5. Systemet kontrollerar samtliga fält i formuläret.
6. Formuläret skickas och lagras i systemet.
7. Verifikation med detaljspecifikationer rörande tid och plats skickas till lagets kontaktinformation.
8. Laget är nu anmäld till träffen.

SEKUNDÄRT FLÖDE

- 5A. Inkorrekt inmatade och/eller obligatoriska utelämnade fält markeras. Laget ges ny chans till att mata in dessa. Korrekt inmatad information raderas inte.

FLÖDESDIAGRAM



KLASSER MED BESKRIVNINGAR

UserAccount
Class

Fields

authority

logged

Properties

Authority

Logged

Methods

UserAccount (+ 1 overload)

UserAccount

Klassens uppgift är att validera huruvida användaren är inloggad i systemet och således tillåten att initiera funktionen. Beroende på användarens status skrivs lämpliga meddelanden ut. I denna iteration nyttjas enbart fältet *logged* med dess tillhörande egenskap *Logged*.

Fältet *authority* med dess tillhörande egenskap *Authority* beskrivs i den tredje och avslutande iterationen.

ApplyTeam
Class

Fields

gymnastAge

gymnastGender

gymnastName

numberOfGymnasts

teamName

Properties

GymnastAge

GymnastGender

GymnastName

NumberOfGymnasts

TeamName

Methods

ApplyTeam (+ 1 overload)

CheckGymnastGender

CheckGymnastName

CheckTeamNameInput

ApplyTeam

Klassens uppgift är att validera den tillhörande information ett lag matar in vid anmälan till träff. Namnen på klassens tillhörande fält, egenskaper samt metoder är relativt självförklarande. De övre tre fälten/egenskaperna representerar varje enskild gymnast i laget. De validerar och fastställer de inmatade värdena gällande gymnastens ålder, kön och namn. De nedre två fälten/egenskaperna representerar lagets övriga information. De validerar och fastställer lagets antal tävlande samt lagnamnet.

Klassens metoder kontrollerar de inmatade strängarna gällande lagnamn, gymnastnamn och kön på deltagande gymnast.

CheckTeamNameInput tillåter endast 3-25 tecken. *CheckGymnastName* tillåter endast bokstäver (stora/små) med eventuella mellanrum. Övriga tecken eller siffror tillåts således ej. *CheckGymnastGender* tillåter endast någon av följande strängar: "M", "Male", "F", "Female".

ENHETSTESTNING

TESTSVIT

Enhetstest för metoderna i klassen UserAccount

Testfall	Vad skall testas	Testdata	Förväntat resultat	Resultat
Lyckad initiering av funktion.	Kontrollera om användaren är inloggad.	logged = true	Initiering lyckas.	Initiering lyckas.
Misslyckad initiering av funktion.	Kontrollera om användaren är inloggad.	logged = false	Initiering misslyckas.	Initiering misslyckas.

Enhetstest för metoderna i klassen ApplyTeam

Testfall	Vad skall testas	Testdata	Förväntat resultat	Resultat
Lyckad tilldelning av lagnamn.	Kontrollera längden av lagnamn.	teamName = "DettaÄrLagomLångt"	Strängen accepteras.	Strängen accepteras.
Misslyckad tilldelning av lagnamn.	Kontrollera längden av lagnamn.	teamName = "DettaÄrKanskeAlldelesFörLångt"	Strängen accepteras inte.	Strängen accepteras inte.
Lyckad inmatning av antal gymnaster.	Kontrollera värdet på antalet gymnaster.	numberOfgymnasts = 10	Värdet accepteras.	Värdet accepteras.
Misslyckad inmatning av antal gymnaster.	Kontrollera värdet på antalet gymnaster.	numberOfgymnasts = 13	Värdet accepteras inte.	Värdet accepteras inte.
Lyckad tilldelning av gymnastnamn.	Kontrollerar strängens tecken.	gymnastName = "Chuck Norris"	Strängen accepteras.	Strängen accepteras.
Misslyckad tilldelning av gymnastnamn.	Kontrollerar strängens tecken.	gymnastName = "Chuck Norris 1337"	Strängen accepteras inte.	Strängen accepteras inte.
Lyckad inmatning av gymnasts ålder.	Kontrollerar värdet på ålder.	gymnastAge = 24	Värdet accepteras.	Värdet accepteras.
Misslyckad inmatning av gymnasts ålder #1.	Kontrollerar värdet på ålder.	gymnastAge = 10	Värdet accepteras inte.	Värdet accepteras inte.
Misslyckad inmatning av gymnasts ålder #2.	Kontrollerar värdet på ålder.	gymnastAge = 85	Värdet accepteras inte.	Värdet accepteras inte.
Lyckad inmatning av genus.	Kontrollerar strängens tecken.	gymnastGender = "Male"	Strängen accepteras.	Strängen accepteras.
Misslyckad inmatning av genus.	Kontrollerar strängens tecken.	gymnastGender = "Females"	Strängen accepteras inte.	Strängen accepteras inte.

Lyckad validering av formulärets fält.	Kontrollerar att samtliga fält är ifyllda och har korrekta värden.	teamName = "DettaÄrLagomLångt" numberOfGymnasts = 10 gymnastName = "Chuck Norris" gymnastAge = 24 gymnastGender = "Male"	Värdena och strängarna accepteras. Formulärets fält godkänns.	Värdena och strängarna accepteras. Formulärets fält godkänns.
Misslyckad validering av formulärets fält.	Kontrollerar att samtliga fält är ifyllda och har korrekta värden.	teamName = "DettaÄrLagomLångt" numberOfGymnasts = 10 gymnastName = "Chuck Norris 1337" gymnastAge = 24 gymnastGender = "Males"	Värdena och strängarna accepteras inte. Formuläret skriver ut ett felmeddelande till användaren som uppmanas fylla i det/de fält som inte godkänts.	Värdena och strängarna accepteras inte. Formuläret skriver ut ett felmeddelande till användaren som uppmanas fylla i det/de fält som inte godkänts.

MOTIVERING

Testsviten är utformad på ett tydligt och överskådligt vis, som vid eventuell utökning av formuläret dessutom möjliggör för vidare påfyllning av ytterligare enhetstester och testfall. Testsviten innehåller två enhetstester med tillhörande testfall vilka verifierar och dokumenterar samtliga fält, egenskaper och metoder av mitt tillämpade användningsfall. Testsvitens samtliga testfall består av fem kolumner som innehåller beskrivande och förklarande information kring vad som skall utföras i varje enskilt test. Planerad testdata presenteras samt förväntade och slutgiltiga resultat. De dokumenterade testfallen anser jag, utefter den grad formuläret är utformat, kontrollerar flertalet tänkbara utfall av de metoder som finns med i denna iterations användningsfall.

IMPLEMENTERA TESTSVITEN OCH KÖR

DOKUMENTATION AV DE UTFÖRDA TESTERNA

Bilden nedan presenterar mina kompillerade enhetstester med tillhörande testfall. Denna inledande iteration testar inmatade värden på samtliga fält vilka ingår i den blankett ett lag förväntas fylla i för en träffanmälan. Varje specifikt områdestest utgörs av två till tre tester där det ena testar godkänt värde, och det andra testar ett icke godtagbart inmatat värde.

- Det inledande testet validerar huruvida användaren är inloggad eller ej, och därmed bestämmer om användaren är tillåten att initiera funktionen.
- Det andra testet kontrollerar längden på det inmatade lagnamnet vilket enbart får bestå av 3-25 tecken. Ett namn som är kortare alternativt längre än detta godkänns ej.
- Det tredje testet kontrollerar mängden gymnaster som matas in för laget. Ett anmält lag får enbart bestå av 5-12 stycken gymnaster. Ett färre eller större antal tillåts ej, och ett felmeddelande visas således för användaren om så är fallet.
- Det fjärde testet validerar formatet på inmatade gymnastnamn. Detta för att inte vilka tecken som helst, så som exempelvis siffror, skall kunna matas in.
- Det femte testet kontrollerar värdet på den inmatade åldern för var och en av lagets gymnaster. Enbart personer från tolv år upp till och med åttio år tillåts delta.
- Det sjätte testfallet kontrollerar den textsträng som skall matas in för att informera om en gymnast är man eller kvinna. Endast godtagbara strängar för detta är följande: "M", "Male", "F", och "Female". Allt över detta visar ett felmeddelande.
- Det sjunde och avslutande testfallet för denna iteration kontrollerar huruvida samtliga fält i blanketten matats in (korrekt) eller inte. Detta för att kunna fortgå med den verifikation som skall skickas till användaren.

ANALYSERA OCH FÖRESLÅ FÖRBÄTTRINGAR

Överlag är jag mycket nöjd över de olika testfallen för denna inledande iteration. Jag har valt att testa samtliga fält då de alla är obligatoriska av användaren att mata in. Inget får utelämnas. Rörande informering av deltagarnas genus skulle man dock kunna låta dem bocka i en av två knappar istället för att uppge en eventuellt krånglig textsträng som lätt kan bli fel och därmed leda till felmeddelande vilket kan uppbringa irritation. En annan förbättring som kan lyftas fram är det reguljära uttryck vilket hanterar formatet på gymnasternas inmatade namn. Även om detta fångar många fel och fungerar väl generellt är det inte alla internationella tecken och namnuppbyggnader som tillåts, och därför är detta ett uttryck att arbeta vidare på.

Utöver ovan nämnda kritik är jag ändå nöjd med testfallen i sin helhet.

BILD TAGEN PÅ KOMPILATOR FÖR TESTFALLEN AV INLEDANDE ITERATION

```
ITERATION 1: ETT LAG VILL ANMÄLA SIG TILL EN TRÄFF
Validerar huruvida användaren är inloggad,
och således tillåten att initiera funktion.
Test 1: Användaren är tillåten att initiera funktion.
Test 2: Användaren är ej tillåten att initiera funktion.

Validerar längden på tillgivet lagnamn.
Test 1: Tilldelat lagnamn OK.
Test 2: Lagnamnet får endast bestå av 3-25 tecken.

Validerar mängden gymnaster.
Test 1: Antal lagmedlemmar OK.
Test 2: Laget får endast bestå av 5-12 gymnaster.

Validerar format på namn.
Test 1: Inmatat namn OK.
Test 2: Inmatat namn tillåts ej.

Validerar värde på ålder.
Test 1: Inmatad ålder OK.
Test 2: Gymnast tillåts ej delta pga låg ålder.
Test 3: Gymnast tillåts ej delta pga hög ålder.

Validerar gymnastens angivna genus.
Test 1: Inmatat genus OK.
Test 2: Svar tillåts ej. Var god ange kön igen ('M', 'Male', 'F', 'Female').

Validerar formulärets fält.
Test 1: Formulärets fält OK
        Formulär skickas in och lagras i systemet.
        Verifikation skickas till användaren.
Test 2: Ett eller flera fält är inkorrekt inmatade. Var god försök igen.
```


REFLEKTION

Jag valde att implementera användningsfallet som möjliggör för ett lag att anmäla sig till en träff i min inledande iteration. Eftersom jag beslutade att fortgå med mitt tidigare arbete från Laboration 3, och själva inloggningsfunktionen därav redan var programmerad, valde jag att fokusera på en av de viktigaste funktionella kraven i systemet. Överlag tycker jag att iterationen har gått bra, och jag har lyckats väl med att hålla min planering. Mitt flödesschema och flödesdiagram har jag kunnat återanvända utan större redigeringar, och därav dokumenterade jag min kravanalys samt kravspecifikation relativt tidigt i arbetsprocessen. Det rörde sig enbart om en vidareutveckling av de steg i flödet jag redan skrivit. Jag valde bland annat att tillägga fält för inmatning av lagnamn, deltagarnamn samt ålder och kön på denne.

Med hjälp av dessa mer specifika tillägg har jag kunnat utföra betydligt fler tester där jag även kontrollerar de inmatade fältens värden och strängar. Detta var en välbehövlig utveckling av min applikation. Det som emellertid krävde extra tid var att fundera ut hur mina tester skulle byggas upp samt komma igång med själva programmeringen. När jag dock påbörjade kodningen fortlöpte arbetet relativt smärtfritt, och iterationen kunde slutföras nästintill planenligt.