

# 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.

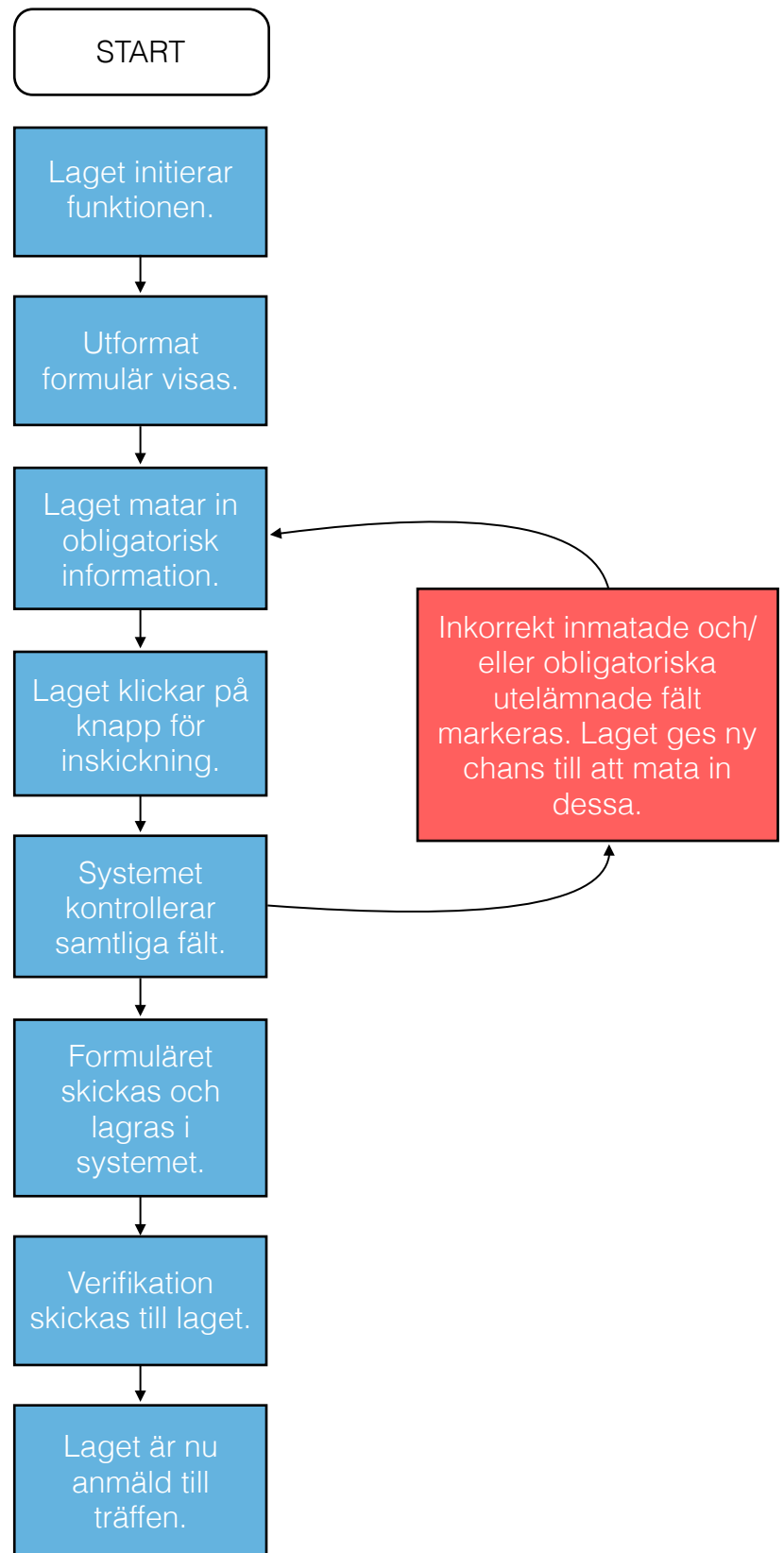
### PRIMÄRT FLÖDE

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. Övanliga 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.

## 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.