

## Tidslogg

Uppgift	Datum	Tid	Arbetsbeskrivning
2	9/12	6 timmar	Det var svårt att förstå vad som skulle vara med i denna uppgift. Även svårt att hitta information på Open UP.
3	10/12	5 timmar	Förstod inte riktigt hur jag skulle koda uppgiften men efter lite hjälp så fick jag ihop uppgiften.
4	11/12	7 timmar	Färdig med uppgiften.
5	12/12	4 timmar	Blev inte exakt det datumet som jag hade planerat.
6	16/12	4 timmar och 15 minuter	Blev ej datumet jag hade planerat.
7	17/12	1 timmar	Reflektionen blev inte heller på datumet jag hade planerat.

## Uppgift 1 Planering

Testplan Själva testplanen kommer ta ca 2 timmar att genomföra, men jag måste även läsa på om hur testplanen skall utföras så jag tillsätter även 3 timmar till att läsa på om hur den skall genomföras.

Jag börjar med denna uppgift den 9/12 och skall vara färdig samma dag så totalt 4 timmar.

## Design och implementation

Först och främst kommer jag behöva välja ett eller flera användningsfall som jag sedan ska analysera för att kunna identifiera klasser. Sedan implementera och dokumentera klasserna. Jag vet inte riktigt vilka användningsfall jag skall välja så kommer behöva tid till både det och till att analysera och identifiera klasserna, även läsa på mer om hur jag skall göra detta. Efter detta så skall jag implementera och dokumentera klasserna. Behöver även koda ett eller flera av användningsfallen.

Sammanfattningsvis kommer jag behöva 4 timmar och 30 minuter för denna uppgift och planerar att göra denna den 10/12.

Uppgift Datum Tid Arbetsbeskrivning  
 2 9/12 6 timmar Det var svårt att förstå vad som skulle vara med i denna uppgift. Även svårt att hitta information på Open UP.  
 3 10/12 5 timmar Förstod inte riktigt hur jag skulle koda uppgiften men efter lite hjälp så fick jag ihop uppgiften.  
 4 11/12 7 timmar Färdig med uppgiften  
 5 15/12 4 timmar Blev inte exakt det datumet som jag hade planerat.  
 6 16/12 4 timmar och 15 minuter Blev inte heller rätt datum.  
 7 17/12 1 timme Fel datum...

## Enhetstestning

Nu måste jag välja ut 2 eller fler av mina klasser från föregående uppgift för att kunna genomföra en analys på dessa. Jag måste även läsa på mer om xUnit arkitekturen och även kolla igenom föreläsningen en gång till för att friska upp minnet om både xUnit, testsvit, testfixturer och testfall.

Sammanfattningsvis kommer jag behöva 5 timmar och 15 minuter för att genomföra denna uppgift och planerar att göra uppgiften den 11/12.

## Implementera testsviten & kör

När jag kommer till denna uppgift så bör jag veta vad veta vad testsviten är så det här lär ju gå någorlunda snabbt att genomföra. Dock eftersom jag inte vid detta tillfälle är helt säker på vad som skall göras så lägger jag även till 2 timmar för att försäkra mig om att jag kan följa min planering så bra som möjligt.

Denna uppgift bör ta ca 5 timmar att genomföra och skall genomföras den 12/12.

## Integrationstestning

Här skall jag välja ett testfall och skriva en kod som gör ytterligare ett test. Eftersom jag gör en liknande uppgift i uppgift 5 så planerar jag att denna uppgiften tar ca 6 timmar att genomföra.

Denna uppgift bör ta ca 5 timmar att genomföra och planeras att vara färdig den 15/12.

## Reflektion

Här skall man skriva en reflektion på uppgiften.

Denna planeras att vara färdig den 16/12 och bör ta ca 1 timme.

## 2

## Testplan

Introduktion Testplanen ska vara ett dokument som kan ändras vid behov för att på bästa sätt uppfylla projektets mål att leverera ett så bra system som möjligt.

## Testuppdrag

Mål Målet går ut på att skapa en testplan för hur testarbetet skall genomföras med fokus på strategi och att skapa arbetsmetoder för att kunna identifiera och eliminera risker med systemet effektivt.

## Risker

- Säkerhet Då systemet innehåller känslig information om privatpersoner (personuppgifter) så krävs en viss grad av säkerhet för att kunna följa kraven från konsumentverket och för att kunna följa de lagar och regler som existerar. T.ex. Personuppgiftslagen (PUL).
- Beräkningssystem Eftersom att det skall genomföras uträkningar på poängen som utdelas av de olika jurymedlemmarna så kommer man behöva fokusera en hel del på att få detta rätt. Utan poängberäkningsfunktionen faller en stor del av systemets funktionalitet.

- Inloggningssystemet Ifall att anmälningsfunktionen icke fungerar så faller hela systemet. Detta skulle påverka all inverkan gjort av företaget och skulle kunna vara en stor vikt att hålla uppe.

Förutom dessa huvudområden så finns det självklart andra risker som kan påverka systemet, men jag bedömer dessa till de tre huvuduppgifterna som systemet lutar mot. Andra risker skulle kunna vara t.ex. underhåll och drift, Användarfunktionalitet och krav från kunden.

## Teststrategi

Teststrategin för detta projekt är att genomföra test som med hjälp av krav från kund. Lagar och regler från konsumentverket för att kunna identifiera riskområden i systemet. Man vill även minimera risken för användarproblem uppstår vid "release". Genom att skapa en bred teststrategi innan man börjar med systemet kommer hjälpa alla inblandade i projektet.

### Testprocess

- För att minimera

risker Hade man ej gjort tester undertiden av programmeringen så hade man aldrig kunnat släppa ett system utan problem/buggar. Vilket skulle kunna påverka förtroendet mellan kunden, användarna och utvecklarna.

- Ekonomisk aspekt

Det hade även påverkat den ekonomiska aspekten på grund av att det är betydligt dyrare att släppa ett halvfärdigt system och sedan behöva, efter "releasen" av systemet färdigställa systemet. Detta kan även påverka utvecklarnas rykte vilket kan som kan få följden av mindre kunder.

- Alpha test

Första testningsfasen skall göras av utvecklarna. Som gör i samband med programmeringen.

- Beta test

Systemet skall sedan beta testas med hjälp av kund, användare mm.

- Utvecklarna

Först och främst så skall utvecklarna testa systemet under testfasen. För att kunna få bort de största uppenbara felen. Framförallt undersöka om huvuduppgifterna fungerar som de ska.

- Beta test

Man kan även efter att utvecklarna själva har testat systemet och ej hittat några problem låta några speciellt utvalda "beta" testare prova systemet för att de skall kunna hitta de mindre felen. Detta skulle t.ex. vara kunden och/eller framtida användare.

### Hur skall vi dokumentera?

Under testfasen så skall allt dokumenteras av både utvecklarna och beta testarna. Och detta skall dokumenteras som statusrapporter, buggrapporter och diagram som skickas in till utvecklarna och även till intressenter hos konsumentverket som vill ta del av informationen, viktigast är att

utvecklarna får informationen så att de får in feedbacken så fort som möjligt för att de skall kunna göra en felsökning så fort som möjligt.

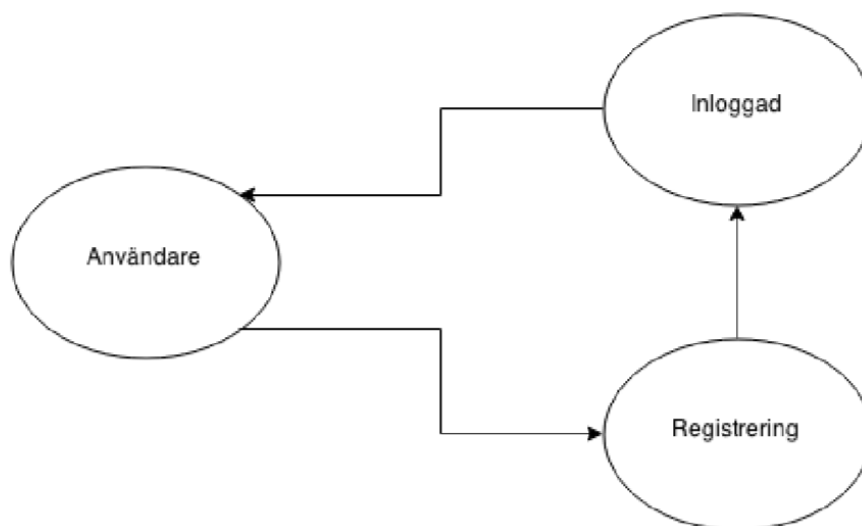
### 3

## Design & implementering

### Allmänna kommentarer och noter som beskriver användningsfall

Inloggningsfunktionen är skapad för att användare skall kunna logga in i systemet och kunna ha en profil som är kopplad till olika funktioner för de olika användarna.

### Användningsfallsmodell



### Krav

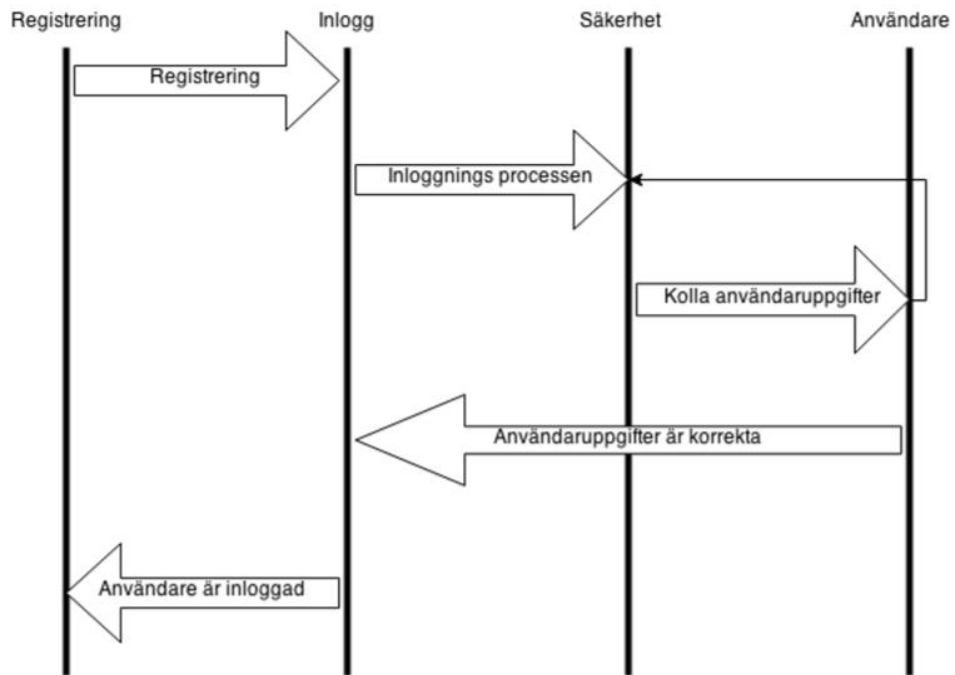
- Användarna skall kunna logga in på sin profil och kunna nå de funktionerna som de vill ha tillgång till. T.ex. om användaren var ett lag, så skall de ha möjlighet att kunna logga in och anmäla sig till. De skall även ha tillgång till att se rankinglistan och när Ligan . Användarens lösenord och användarnamn får endast vara en viss satt längd.

### Begränsningar

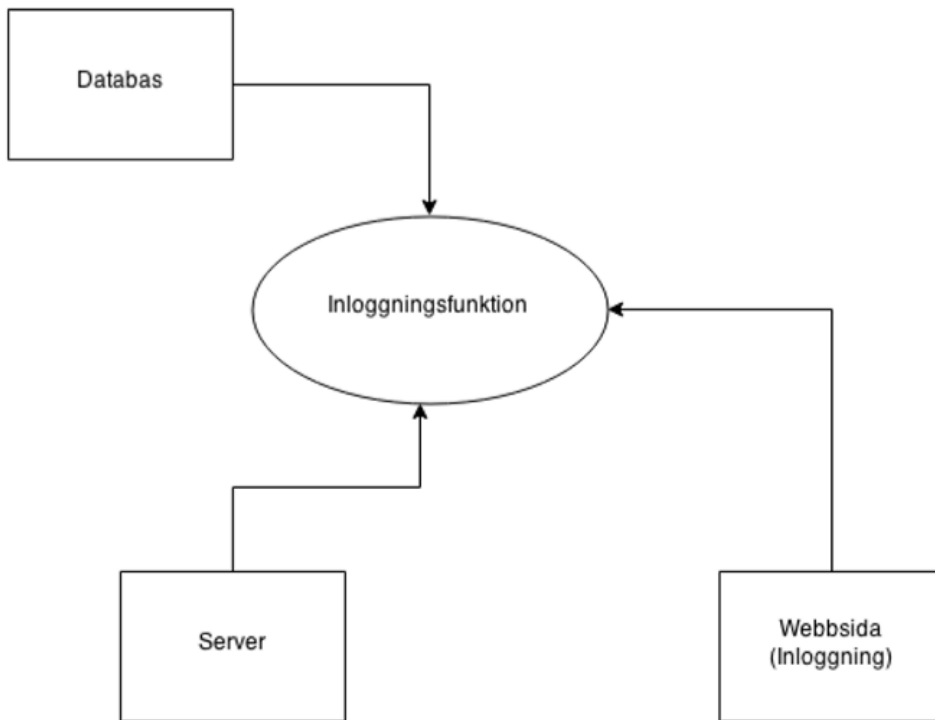
- Begränsningar i funktionen är man först och främst (pre) måste ha registrerat sig för ett konto innan man har möjlighet att logga in i systemet. Man måste även ha skapat ett användarnamn och lösenord.
- Sen när man har skapat ett konto (post) så måste man ha tillgång till användarnamn och lösenord för att kunna logga in i systemet.

## Scenarios

### Sekvensdiagram



### Implementeringsdiagram



## Användningsfallsbeskrivning

### Inloggningsfunktion

En användare skall kunna logga in i systemet.

- Aktörer
- Klubbar
- Lag
- Gymnaster
- Domare/Jury

### Pre villkor

Användarna måste ha inloggningsuppgifter och vara registrerade i systemet. Användaren får verifikations mail vid registrering. Användaren måste antingen ha tillgång till sina personuppgifter eller organisationsnummer för att kunna registrera sig i systemet.

### Post villkor

Användaren får mail angående att någon har försökt logga in på deras profil, när man har skrivit fel lösenord eller användarnamn 4 gånger. Måste ha en funktion som kan hjälpa användaren vid förlorat användarnamn och lösenord.

## Primära flöden

1. Användaren registrerar sig i systemet. 2. Användaren matar in sitt användarnamn och lösenord. 3. Om användaren matar in fel användaruppgifter skickas de till steg 4, om användaren matar in korrekt uppgifter skickas de vidare till steg 6. 4. Systemet upptäcker att det är fel användaruppgifter och skickar dem till steg 5. 5. Systemet visar felmeddelande, tillbaka till steg 2 (fel lösenord eller användarnamn). 6. Systemet hämtar användarens privata uppgifter och profil. 7. Användaren är nu inloggad i systemet. 8. Ifall användaren skriver fel lösenord 4 gånger i rad, så skickas det ett mail till användarens registrerade mail angående detta.

## Alternativa flöden

Om användaren matar in fel användarnamn eller fel lösenord så skickar systemet felmeddelande och skickar tillbaka användare till inloggningssidan. Användarnamnet och lösenordet får ej gå över en satt längd.

## Klasser med beskrivningar

### User

I klassen User sätter man lösenord och användarnamn för användare.

### Parametrar

- txtPass(string)

### Håller användarens lösenord.

- txtUser(string)

## Egenskaper

- PassWord

I denna egenskap så görs txtPass om till value så att jag senare kan ta och hämta detta från min validerings klass.

- Username

I denna egenskap så görs txtUser om till value så att jag senare kan ta och hämta detta från min validerings klass.

## Validate

I klassen Validate kollar programmet, så att både lösenordet och användarnamnet är korrekt.

### Parametrar

- passwords(string)
- users(string)
- returnline(string)

## 4

## Enhetstestning

Till mina tester har jag valt klasserna User & Validate.

### Testsvit 1

- Testfall

Kör test som kollar så att det ej går att sätta ett "förbjudet" värde.

- Testar Password
- Förväntat resultat

Förväntas att det ej går att mata in ett "förbjudet" värde.

- Resultat av testningen

Resultatet av testningen visar att det ej gick att skriva in ett förbjudet värde.

### Testsvit 2

- Testfall

Kör test som kollar så att det ej går att sätta ett "förbjudet" värde.

- Testar Username
- Förväntat resultat



Förväntas att det ej går att mata in ett "förbjudet" värde.

- Resultat av testningen

Resultatet av testningen visar att det ej gick att skriva in ett förbjudet värde.

- Motivering till testsviten och testfallen

Gör testet i en testsvit, den testar att sätta ett förbjudet värde som användaren ej skall kunna mata in.

Motiveringen till detta är att det känns logiskt att testa all funktionalitet som finns i klasserna.

## 5

### Dokumentation av de utförda testerna / testdata

#### TestPasswordCorrect()

Denna egenskap testar så att om användaren matar in ett lösenord som är kortare än 12 tecken så är det godkänt, för att testa så det fungerar.

Efter testningen så blev resultatet att det visades ett meddelande att lösenordet var i godkänd längd.

#### TestPasswordIncorrect()

Denna egenskap testar så att om användaren skulle mata in ett lösenord som är för långt så skickar det ett `ArgumentOutOfRangeException`. För att hindra användare från att mata in ett för långt lösenord.

Efter testningen så blev resultatet att det visades ett meddelande att lösenordet var för långt.

#### TestUsernameCorrect()

Denna egenskap testar så att om användaren matar in ett för långt lösenord så får användaren ett meddelande om att användarnamnet är för långt. För att hindra användare från att mata in ett för långt lösenord.

Efter testningen så blev resultatet att det visades ett meddelande att användarnamnet var inom godkänd längd.

#### TestUsernameIncorrect()

Denna egenskap testar om användaren matar in ett lösenord som är för långt så skickas det ett `ArgumentOutOfRangeException` med ett meddelande. För att hindra användare från att skriva in för långa användarnamn.

Efter testet så blev resultatet att det visades ett meddelande att användarnamnet var för långt.

## Reflektion

Det jag anser är det svåraste med att planera en uppgift som är så stor som denna är att man aldrig vet riktigt vad som man kan fastna på. Det kan vara så att man planerar att kodningen skall ta t.ex. ca 3 timmar men sedan när man sitter med koden så kan man fastna på någon del och inte hitta en lösning och så slutar det med att man har spenderat 6 timmar på kodningen och sen har man kommit efter i hela planeringen.

Det jag anser att man alltid bör göra vid en uppgift som är så här stor, är att alltid ha åtminstone 2 helt oplanerade dagar i slutet av uppgiften så att man kan ändra, rätta till eller om man ej har hunnit med i planeringen, ha extra tid att genomföra uppgifterna.

Jag tycker dock att denna uppgift var förvirrande i hur man skulle lägga upp uppgifterna och att vi knapp har gått igenom hälften av de metoderna som vi skall använda oss av. Även t.ex. när man skriver "Använd metoden från föreläsningen ...". Kunde man inte då ha skrivit med vilken föreläsning som man menar. Eftersom att man ej hade en föreläsning samtidigt som man var på denna uppgiften så är det väldigt svårt att förstå vilken man menar.

Dock känner jag att jag har lärt mig väldigt mycket om hur man skall planera en så här stor uppgift. Som t.ex. att man ej kommer göra detta ensam utan man kommer arbeta med detta som en grupp.