# Iteration A

# Planering

|  |  |  |
| --- | --- | --- |
| Uppgift | Planerad tid | Utförd tid |
| Planering | 1 t | 50 m |
| Analys | 1 t | 45 m |
| Implementering av klasser | 2 t | 1 t 25 min |
| Testning | 3 t | 3 t |
| Reflektion | 1 t | 50 m |

Kommer försöka jobba enligt denna planeringen med arbetet, det vill säga med tidsgränser som ej skall bör överskridas.

# Användningsfall: Inloggningsfunktion

Detta användningsfall kommer beskriva hur inloggning i systemet kommer fungera.

## Aktörer

### Användare

### Lag

### Gymnaster

### Domare/juryn

### Klubbar

## Pre-villkor

Användarna är registrerade i systemet och har tillgång till sina personliga inloggningsuppgifter.

Användaren får verifikations mail vid registrering av sin profil i systemet.

Användaren som skall registrerar sig behöver ha tillgång till sina personuppgifter eller om det är en klubb som skall registrera sig så behöver de ha tillgång till sitt organisationsnummer.

## Post-villkor

Användaren är inloggad i systemet.

Användaren får mail angående om någon har skrivit fel lösenord eller användarnamn. Vid detta spärras även profilen och användaren måste då kontakta supporten.

## Primära flöden

### Användaren matar in sina inloggningsuppgifter.

### Systemet validerar uppgifterna.

### Systemet hämtar användarens profil.

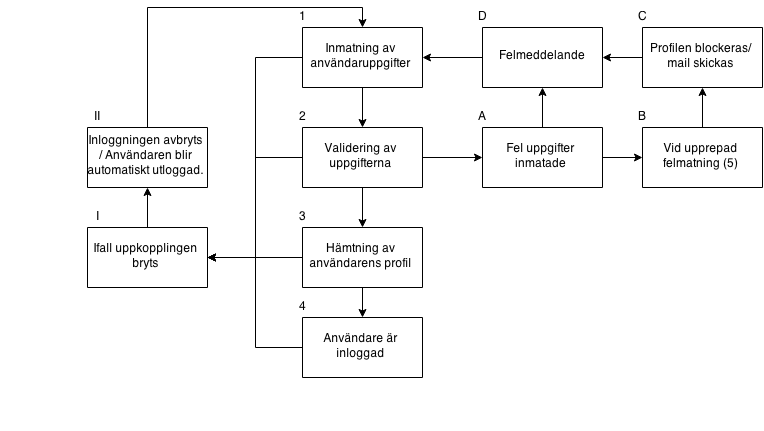
### Användaren är nu inloggad i systemet.

## Alternativa flöden

1. Fel uppgifter inmatade.
2. Vid upprepad felinmatning (5gånger).
3. Profilen blockeras/Mail skickas.
4. Felmeddelande.

## Exceptionella flöden

1. Ifall uppkopplingen skulle brytas av okänd anledning.
2. Inloggningen avbryts / användaren blir automatisk utloggad.



# Analys

## Klasser

Efter analys av flödena som jag har planerat så planerar jag att ha:

### Klassen User:

I klassen User kommer jag sätta en bestämd längd på användarens lösenord och användarnamn. Kommer även införa en regex för användarmailen.

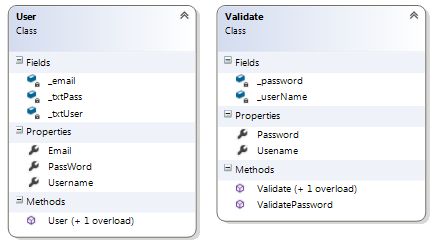
* Bestämd längd på lösenord och användarnamn
* Regex
* Email

### Klassen Validate:

I klassen Validate kommer jag validera lösenordet och användarnamnet som användaren matar in vid inloggning.

* Användarnamns validering
* Felmeddelande

Namnen på klasserna säger ganska mycket om vad de gör så känner att jag behöver ej gå in djupare på detta, har även kommenterat allt.



# Testsvit

Under testerna så matar jag in antingen godkänd data eller icke godkänd data för att få ut ett väntat resultat. Alltså om jag skall testa så att man ej kan skriva in ett lagnamn som endast innehåller fem bokstäver, så skickar jag in ett lagnamn på fem bokstäver så förväntar jag mig att ett felmeddelande kastas.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Metod | Testfall | Data | Förväntat resultat | Resultat |
| TestPasswordCorrect | Godkänd data matas in | ”aaaaa” | metoden returnerar true | Testet lyckades |
| TestPasswordIncorrect | Icke godkänd data matas in | ”aaaaaaaaaaaaa” | Kastar felmeddelande | Testet lyckades |
| TestUsernameCorrect | Godkänd data matas in | ”aaaaaaaaaaaaaaaaaa” | Metoden returnerar true | Testet lyckades |
| testUsernameIncorrect | Icke godkänd data matas in | ”aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa” | Kastar felmeddelande | Testet lyckades |
| TestEmailPass | Godkänd data matas in | "hej@student.com" | Metoden returnerar true | Testet lyckades |
| TestEmailFail | Icke godkänd data matas in | "hej.com" | Kastar felmeddelande | Testet lyckades |
| TestValidate | Godkänd data matas in | "hej", "123" | Metoden returnerar true | Testet lyckades |
| TestValidatefail | Icke godkänd data matas in | "hej", "1233" | Kastar felmeddelande | Testet lyckades |

# Reflektion

Jag valde att fortsätta bygga på min föregående iteration och lägga till fler test. Detta gjorde jag för att jag kände att jag kunde förbättra iterationen och göra den lite större och ändå få med det jag ville ha för att få mina iterationer att kunna var i samma projekt.

Tidsmässigt så gick denna iteration för bra för att jag ska vara nöjd. Skulle nog ha kunnat lägga till mer tester och mer funktioner men eftersom jag vill hinna med allt så känner jag att jag vill gå vidare på nästa iteration så fort som möjligt ifall jag skulle få problem där. Om detta skulle hända så skulle jag inte kunna få problem med nästa iteration och då skulle det bildas en kedjereaktion igenom alla iterationer.