# Iteration B

## Planering

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

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

# Användningsfall: Registrering av lag

Detta användningsfall kommer beskriva vad som måste vara med vid registrering av lag i systemet.

## Aktörer

* Lag

## Pre-villkor

Måste vara mellan 5-12 medlemmar.

## Post-villkor

Laget är registrerat.

## Primära flöden

1. Användaren loggar in
2. Användaren matar in lagnamn
3. Systemet bekräftar om poäng är korrekt inmatade
4. Laget är skapat

## Alternativa flöden

### Felaktiga Inmatning av lagnamn

### Felmeddelande visas

### Ej korrekt antal medlemmar

### D. Felmeddelande



# Analys

## Identifera klasser

I analysen har jag kommit fram till att det behövs en klass.

Klassen för användare som registrerar lag.

### Klassen: TeamReg

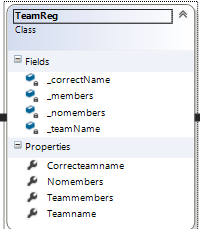
Denna klass kommer kontrollera användaren från att skriva in ett för långt eller för kort lagnamn.

Göra så att lagnamnet måste börja med stor bokstav.

Antalet medlemmar måste vara mellan 5-12 per lag.

# Klassen TeamReg

I koden så är metodnamnen väldigt förklarande och jag har även kommenterat allt, så anser att man inte behöver klassbeskrivning utan att det blir ganska självklart.



# 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 | Förväntat resultat | Data som matas in | Resultat |
| Teamnamecorrect | Godkänd data matas in | metoden returnerar true | ”Testning” | Testet lyckades |
| Teamnameincorrect | Icke godkänd data matas in | Kastar felmeddelande | ”test” | Testet lyckades |
| Teambiglettercorrect | Godkänd data matas in | Metoden returnerar true | ”Thebigletters” | Testet lyckades |
| Teambigletterincorrect | Icke godkänd data matas in | Kastar felmeddelande | ”thereisnobigletter” | Testet lyckades |
| Correctamountofmembers | Godkänd data matas in | Metoden returnerar true | 10 | Testet lyckades |
| Incorrectamountofmembers | Icke godkänd data matas in | Kastar felmeddelande | 3 | Testet lyckades |
| Nomembers | Icke godkänd data matas in | Kastar felmeddelande | 0 | Testet lyckades |

# Reflektion

Jag är nu färdig med min andra iteration. Denna uppgift har jag inte haft så stora problem med allt har flutit på ganska bra testningen gick lite snabbare än planerat men känner att jag kommer lägga ner lite extra på nästa uppgift eftersom jag inte riktigt vet vilka tester som jag bör implementera. Använder mig även i denna uppgift av tester baserade på try catch.