Laboration 4 – 1DV404

Lucas Wik

**Iteration 1 – Planering:**

(Jag bestämde mig att fortsätta på min laboration 3)

*Introduktion:* Så nu när jag har edit möjligheter och kan registrera lag så måste jag se till att bara tillåtna kan komma åt meny valen som att byta poäng eller ändra lag namn osv. Så mitt mål med min första iteration är att implementera ett logg-in system med X mängd av konton som ges ut till personer som ska ha möjlighet att komma åt dessa funktioner i programmet. Just nu så gäller detta för meny val 2 och 3.

Måste också ta i tanke om hur jag ska hantera detta med de inkomande iterationerna eftersom en admin ska kunna edita val som ska implementeras i senare iterationer.

*Iteration 1 – Mål:*

* Admin konton ska skapas då de ska ha möjlighet att editera dessa lag. Allt ska kunna ändras av en admin, lagnamn, medlem, träff plats. Allt. Ta detta i tanke vid uppsättningen av systemet.
* Systemet behöver inte se till talande ut men de olika valen användaren har ska framgå tydligt vad de gör.

Tidsberäkning: 8 tim. (Motivation för beslut: eftersom jag redan har det mesta i grunden och denna iteration är mest om en funktion som jag inte implementerade i labb 3 så tror jag inte att det kommer ta så långt tid.)

**Iteration 2 – Planering:**

*Introduktion:* Nu när basen är helt färdig så måste vi slipa kanterna. Det jag vill fokusera på i denna iteration är att lägga till mera system krav och stila upp det lite. Till och börja med så ska poängen räkningen implementeras så att ett lags poäng räknas ut genom att addera ihop alla medlemmarnas poäng i laget och att en medlems poäng i en gren kan ges av en sekreterare. En medlems poäng är summan av alla poängen den personen har fått i alla grenar hon/ han har deltagit i.

Träffar behöver också bestämmas så att när en registrering sker så måste de också ange vilken stad de vill tävla i samt vilken träff.

*Iteration 2 – Mål:*

* Poängberäkningen, en admin(sekreterare) ska kunna ge en medlem poäng i varja liga han/ hon deltar i. Därefter så ska systemet räkna ut varje medlems poäng genom att addera alla poängen personen har fått i alla grenar, ett lags poäng är summan av alla dessa slut poäng adderat ihop.
* Vid registrering så sker också ett val som bestämmer vilken träff laget ska delta i, de får välja angiven stad och tid/datum som programmet presenterar.
* Vid detta stadiet så behöver inte systemet se snyggt ut men meny uppsättningen och valen ska vara bra arrangerat.

Tidsberäkning:11 tim. (Motivation för beslut: eftersom dessa mål är saker som ska läggas till mitt i applikationen så förväntar jag mig att behöva omstrukturera någon del och därför tror jag att den kommer ta mer tid än iteration 1.)

**Iteration 3 – Planering:**

*Introduktion:* Så nu när de mesta kraven är ifyllda så är det dags göra de sista tilläggen. Det jag hade tänkt mig för iteration 3 är att användaren har valet att se en lista av alla lag, med alla medlemmars poäng och lagens poäng och ordna dem från laget med mest poäng överst i listan och laget med minst poäng längst ner i listan. Detta blir ett val användarna kan vilja se vid slutet av en säsong eller bara kolla hur de ligger till i deras träff. Samt så ska systemet se grafiskt tilltalande ut.

*Iteration 3 – Mål:*

* Systemet ska se grafiskt till talande ut och meny valen ska vara bra arrangerat.
* En användare ska kunna se en lista över alla lag och medlemmar då de arrangeras i lednings order, dvs. att de som leder är överst på listan.

Tidsberäkning: 10 tim. (Motivation för beslut: även om det inte känns så mycket att göra så är jag inte 100% på hur jag ska göra en fungerande uppsättningen för det ännu.)

**Slutkrav för projektet (sammanfatting):**

* Man ska kunna registrera och admins ska kunna editera lag.
* Vid registrering så ska man välja vilken träff man kommer att delta i.
* Grafiskt tilltalande.
* Ett poängberäknings system som underlättar för sekreteraren.

**Iteration 1**

*Detaljerad planering:*

Krav – analysera hur den totala uppsättningen kommer att se ut. (Skriv tiden här)

Implementation – Identifiera klasser och deras uppgifter. (Skriv tiden här)

Test – Identifiera olika test fall för en aktör och deras mening. (Skriv tiden här)

Reflektion – Reflektera på iterationen. (Skriv tiden här)

**Steg 1 – Krav:**

Klasser:

* LoggIn, denna klass ska ta användar namn och lösen som argument och retunera true eller false beroende om uppgifterna överstämmer.

Aktörer:

* Sekreterare, behöver denna funktion så hon/ han kan vara ensam om att editera.

Földen:

* Sekreterare:
  + Startar appliaktionen.
    - Krashar. - >Misslyckat försök.
  + Anger loggin valet och skriver in användar namn och lösen.
    - Anger fel uppgifter -> blir meddelad om misstag.
  + Blir lyckat inloggad, ska nu visas för att göra användaren medveten om att hon/ han är inloggad.

**Steg 2 – Implementation:**

Klasspecifikationer:

* LoggIn:
  + Metod: ValidateInfo
    - Kollar så att den inskriva infon över stämmer med inloggnings info, retunerar true om sant, false om falskt.
  + Egenskaper och fält:
    - AccountName: ett fält som tar värdet av det med sickadekonto namnet.
    - AccountPasswrod: ett fält som tar värdet av det medsickade konto lösenordet.
    - AccountNames: Array som innehåller konto namn, konto namnets lösen har samma array index som AccountPasswords.
    - AccountPassword: Array som innehåller konto lösenord, lösenorderts index i arrayn är samma som dess konto namn i AccountNames.
  + Konstuktorer:
    - 1 konstruktor som tar emot 2 parametrar. Den första är konto namn den andra är lösen.

**Steg 3 – Test:**

*Testfall:*

* Vad händer om användaren skriver in fel logg-in info? Hur meddelas dem? Var händer?
  + Förväntat resultat: Användare meddelas om att inloggnings infon är fel, samt vilken av dem som är fel, dvs. kontonamn eller lösen.
* Kan man se om man är inloggad/ inte inloggad? Hur visas det och när?
  + Förväntat resultat: En text ska visas på menyn som visar ens status, om man är inloggad eller inte, den förväntade är att den uppdateras vid inloggning/ utloggning.
* Kan man skriva in ett konto namn och sen lyckas använda ett annat kontos lösen och fortfarande loggas in?
  + Förväntat resultat: Nej, detta ska inte lyckas. Endast 1 lösen ska funka för varje konto namn.

**Reflektion:**