

Iterativ Mjukvaruutveckling 1DV404

Laboration 4 – Projekt

Student: Andréas Anemyr WP -14 (aa223ig@student.lnu.se)
Handledare: Emil Carlsson

Tidslog Iteration 1

p*	Iterationsplanering	Grovplanering
1*	Planering	Detaljnivå
2*	Användningsfall	Analyseras (vid behov även skapa!)
3*	Design	Identifieras, Specificeras & Implementeras
4*	Test	Designas, Specificeras, Implementeras & Exekveras
5*	Reflektion	

Loggning												
Datum	Beskrivning (delmål)	Iteration	Moment id	Börjar	Slutar	Saldo *1	Saldo *2	Saldo *3	Saldo *4	Saldo *5	Saldo iteration	Saldo
150108	Iterationsplanering	-	p	10:00	10:30							00:30
150108	Iterationsplanering	-	p	18:15	19:00							01:15
150108	Planering	1	1	19:00	19:45	00:45	-	-	-	-	00:45	02:00
150108	radera klubb flödesschema #2	1	3	06:30	06:45	-	-	00:15	-	-	01:00	02:15
150109	renderClubList() #2	1	3	06:45	07:15	-	-	00:45	-	-	01:30	02:45
150109	InactivateClub() #2	1	3	07:15	07:45	-	-	01:30	-	-	02:00	03:15
150109	Planering	1	1	08:00	08:15	01:00	-	-	-	-	02:15	03:30
150109	Reflektion	1	5	08:15	08:30	-	-	-	-	00:15	02:30	03:45
150109	Användningsfall	1	2	08:45	09:00	-	00:15	-	-	-	02:45	04:00
150109	Design etc	1	3	09:00	10:00	-	-	02:30	-	-	03:45	05:00
150109	Planering	1	1	11:45	12:45	02:00	-	-	-	-	04:45	06:00
150109	Analyserat användningsfall och aktör. Enkla flödesschema mm.	1	2	13:45	14:30	-	01:00	-	-	-	05:30	06:45
150109	Implementerat setClubAdmin	1	3	14:30	15:00	-	-	03:00	-	-	06:00	07:15
150109	Testimpl testar kod	1	4	15:00	15:30	-	-	-	00:30	-	06:30	07:45
150109	Forts impl. test	1	4	16:00	16:15	-	-	-	00:15	-	06:45	08:00
150109	Imp test. Strular!	1	4	18:15	18:45	-	-	-	00:45	-	07:15	08:30
150109	Reflektion	1	5	18:45	19:00	-	-	-	-	00:30	07:30	08:45

Not. Iteration avbryts vid 7tim och 30min. 30 minuter tidigare än beräknat.

Ansvarig iteration 1: Andréas Anemyr LNU WP -14

Tidslog Iteration 2

p*	Iterationsplanering	Grovplanering
1*	Planering	Detaljnivå
2*	Användningsfall	Analyseras (vid behov även skapa!)
3*	Design	Identifieras, Specificeras & Implementeras
4*	Test	Designas, Specificeras, Implementeras & Exekveras
5*	Reflektion	

Loggning												
Datum	Beskrivning (delmål)	Iteration	Moment id	Börjar	Slutar	Saldo *1	Saldo *2	Saldo *3	Saldo *4	Saldo *5	Saldo iteration	Saldo
150109	Planering	-	p	19:00	19:30						00:00	09:15
150109	Iterationsplanering	2	1	21:55	22:20	00:25	-	-	-	-	00:25	09:40
150109	Användningsfall förberedande	2	2	22:20	22:35	-	00:15	-	-	-	00:40	09:55
150109	Användningsfall delmål #1 registrera lagadmin	2	2	22:35	22:50	-	0:30	-	-	-	00:55	10:10
150109	Iterationsplanering modifierat målsättning (tid)	2	1	22:50	23:10	00:45	-	-	-	-	01:15	10:30
150110	Användningsfall delmål #2 radera lag	2	2	08:15	08:30	-	00:45	-	-	-	01:30	10:45
150110	Reflektion	2	5	08:30	08:45	-	-	-	-	00:15	01:45	11:00
150110	Iterationsplanering Dokumentation	2	1	08:45	08:55	0:55	-	-	-	-	01:55	11:10
150110	Användningsfall delmål #2 radera lag	2	2	08:55	09:15	-	01:05	-	-	-	02:15	11:30
150110	Användningsfall delmål #2 radera lag – fl.diagr	2	2	09:15	09:30	-	01:20	-	-	-	02:30	11:45
150110	Planering div rapport	-	p	09:30	09:35	-	-	-	-	-	-	11:50
150110	Användningsfall granskar krav – försöker förstå....	2	2	09:30	09:50	-	01:40	-	-		02:50	12:10
150110	Användningsfall delmål #3 delegera lagadministratör	2	2	09:50	10:20	-	02:10	-	-	-	03:20	12:40
150110	Reflektion	2	5	10:20	10:25	-	-	-	-	00:05	03:25	12:45
150110	Användningsfall delmål #3 delegera lagadministratör	2	2	13:15	13:30	-	02:25	-	-	-		
150110	Rapporterin tid uppföljning	2	1	13:31	13:40	01:04	-	-	-	-	03:34	12:54
150110	Design delmål #1	2	3	13:40	15:20	-	-	01:40	-	-	05:14	14:34
150110	Planering div rapport	2	1	15:20	15:50	00:30	-	-	-	-	06:44	15:04
150110	Test delmål #1 analys	2	4	15:50	16:10	-		-	00:20	-	07:04	15:24
150110	Test delmål #1 implementation	2	4	16:45	17:35	-	-	-	00:50	-	08:04	16:24

150110	Design delmål #2 Test delmål #2	2	3,4	19:25	20:50	-	-	02:20	01:30	-	09:29	17:49
150110	Planering div rapport	2	3,4	21:45	22:05	-	-	02:30	01:40	-	09:49	18:09
150110	Design delmål #2 Test delmål #2	2	3,4	22:05	22:50	-	-	02:55	02:10	-	10:34	18:54
150110	Försök till komplettering, avryts!	2	3,4	22:50	23:10	-	-	03:05	02:20		10:45	19:04
150110	Reflektion	2	5	23:10	23:40	-	-	-	-		11:15	19:34

Not. Iteration avbryts vid 11tim och 15min. 45 minuter tidigare än beräknat.

Ansvarig iteration 2: Andréas Anemyr LNU WP -14

Tidslog Iteration 3

p*	Iterationsplanering	Grovplanering
1*	Planering	Detaljnivå
2*	Användningsfall	Analyseras (vid behov även skapa!)
3*	Design	Identifieras, Specificeras & Implementeras
4*	Test	Designas, Specificeras, Implementeras & Exekveras
5*	Reflektion	

Loggning												
Datum	Beskrivning (delmål)	Iteration	Moment id	Börjar	Slutar	Saldo *1	Saldo *2	Saldo *3	Saldo *4	Saldo *5	Saldo iteration	Saldo
150111	Planering	3	p	10:00	11:00	01:00	-	-	-	-	01:10	20:34
150111	Reflektion planering	3	5	11:00	11:20	-	-	-	-	00:20	01:30	20:54
150111	Användningsfall delmål #1	3	2	11:20	12:15	-	00:55	-	-	-	02:25	21:49
150111	Användningsfall delmål #1 (failed)	3	2	12:15	13:00	-	01:25	-	-	-	03:10	22:34
150111	Användningsfall delmål #1 (ställtid pga juledighet?)	3	3	13:00	14:55	-	-	01:55	-	-	05:05	24:29
150111	Test delmål #1	3	4	14:55	15:35	-	-	-	00:40	-	05:45	24:59
150111	Reflektion delmål #1	3	5	15:35	16:00	-	-	-	-	00:45	06:05	25:24
150111	Användningsfall delmål #2	3	2	16:00	16:30	-	01:55	-	-	-	06:35	26:04
150111	Design delmål #2	3	3	18:00	19:50	-	-	03:45	-	-	08:25	27:54
150111	Test delmål #2	3	4	19:50	20:05	-	-	-	00:55	-	08:40	28:09
150111	Reflektion	3	5	20:05	20:20	-	-	-	-	01:05	08:55	28:24
150111	Användningsfall	3	2	21:15	21:45	-	02:25	-	-	-	09:25	28:54
150111	Design	3	3	21:45	22:45	-	-	04:45	-	-	10:25	29:54
150111	Designelement Avbryts "formulär"	3	4	22:45	23:30	-	-	05:30	-	-	11:10	30:39
150111	Reflektion #1,2,3 + diverse justeringar	3	5	23:30	00:20	-	-	-	-	01:55	12:00	31:29

Not.

Iteration avbröts vid tolv timmar.

Impl. Av design ej komplett för delmål #3

Inga tester utföra för delmål #3

Ansvarig iteration 3: Andréas Anemyr LNU WP -14

Iterationsplanering (Grovplanering)

Iterationsplaneringen omfattar iteration ett, två och tre.

Iterationsplaneringen avser att med tydlighet, men inte på detaljnivå, framhäva vad som är målsättning för respektive iteration. Varje iteration i sig allokerar en tidsåtgång som ska beaktas.

Projektet har tilldelats en maximal tidsåtgång på 32 timmar totalt.

Vid en eventuell tidsöverskridning > 60m min/iteration kommer den pågående iterationen att avbrytas och rapport kommer att avläggas på det som utvecklats. Fullföljda iterationer med förväntade resultat är ett primärt mål för varje iteration.

Iteration 1

- Vad
 - Implementera användningfall punkt 1 så långt som möjligt inom en tidsrymd av max 8 timmar.
Vid implementering beaktas visionsdokumentet och applikationens krav.
- Mål
 - Systemadministratören ska kunna
 - Logga in → registrera klubbadministratör #1 :)
 - Logga in → radera klubb (inaktivera) #2 :)
 - Logga in → Delegera klubbadministratör #3 :)
 - Impl. enkelt(minimalt) användargränssnitt om tid finns #4 :(
- Tid
 - 8 tim

Iteration 2

- Vad
 - Implementera användningfall punkt 2 så långt som möjligt inom en tidsrymd av max tolv timmar.
Vid implementering beaktas visionsdokumentet och applikationens krav.
- Mål
 - Klubbadministratören ska kunna
 - Logga in → registrera lagadministratör #1 :)
 - Logga in → radera lag #2 :)
 - Logga in → Lagadministratör ska kunna delegera lagadministratör #3 :)
 - Impl. enkelt(minimalt) användargränssnitt om tid finns #4 :(
- Tid
 - 12 tim

Iteration 3

- Vad
 - Implementera användningfall punkt 5 så långt som möjligt inom en tidsrymd av max tolv timmar.
Vid implementering beaktas visionsdokumentet och applikationens krav.
Samtliga delmål ska kunna genomföras i ett flöde utan att man testat sig fram genom att anropa metoder enligt flödet, såsom gjorts i tidigare iterationer.
Användaren kommer under den här iterationen att förse systemet med variabla data med hjälp av prompt(). Dessa ersätts i efterföljande iterationer efter planering till lämpligt gränssnitt. Dessa används tillfälligt endast för att simulera ett fungerande flöde.
- Mål
 - Lagadministratören ska kunna
 - Logga in → registrera gymnast/lagmedlem #1 :)
 - Logga in → radera gymnast/lagmedlem #2 :)
 - logga in → redigera befintlig data på gymnast/lagmedlem #3 :(
- Tid
 - 12 tim

Iteration 1, på detaljnivå (kravanalys → design → implementation → test → reflektion)

#1

id	Moment	Kalkyltid	Verklig tid	check
1	<ul style="list-style-type: none"> Planering 	01:30	02:00	X
2	<ul style="list-style-type: none"> Användningsfall <ul style="list-style-type: none"> analyseras 	01:00	01:00	X
3	<ul style="list-style-type: none"> Design <ul style="list-style-type: none"> identifieras specificeras implementeras 	02:00	03:00	X
4	<ul style="list-style-type: none"> Test <ul style="list-style-type: none"> designas specificeras implementeras exekveras 	02:00	00:45	
5	<ul style="list-style-type: none"> Reflektion 	00:30	00:30	X
Totalt		7 timmar	7,15 timmar	

Iteration 1 Delmål #1**Logga in → registrera klubbadministratör**

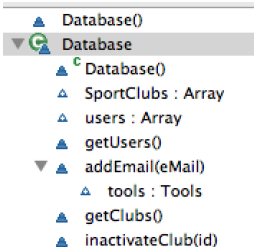
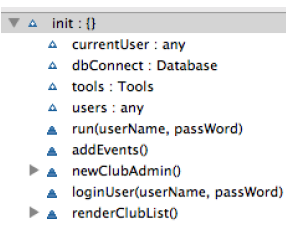
Efter att ha granska föregående labb tog jag beslutet att låta detta delmål vara uppfyllt från den föregående laborationen. Läggs åt sidan tills vidare...

Iteration 1 Delmål #2**Logga in → radera klubb**

Krav: Samtliga lag måste ha ett id och ett namn.
Id ska vara av typen number.

Primärt Flöde	Alternativt flöde
<ol style="list-style-type: none">1. hämta samtliga klubbar i "Database"2. "Output" i webbläsare (GUI)3. Välj lista i en klubb4. Vald klubb raderas (inaktiveras)5. Ny listan renderas	<ol style="list-style-type: none">1. Inget svar från "Database"2. Generera felmeddelande

Designelement och eventuella kommentarer och förtydliganden

Källfil: Database	Källfil: Main	Kommentar
		Database: Klassen database finns sedan tidigare. För att uppnå krav måste metoderna getClubs() och inactivateClub() implementeras. Init{}: Objektet init{} behöver metoden renderClubList för att delmål #2 ska kunna nås

Enhetstestning

Klassen Tools avser att samla upp metoder som systemet behöver nyttja på en mängd olika ställen i programkod.

Testfixtur:

Testsvit:

testDatabase (jsTestDriver)

Version:

2.0

Anrop	Ansvarig	Förväntat	Kommentar	testId	Check
db.setClubAdmin(999, "asdf")	Andréas	fail	Endast typen number accepteras som andra parameter	2.05	Nej

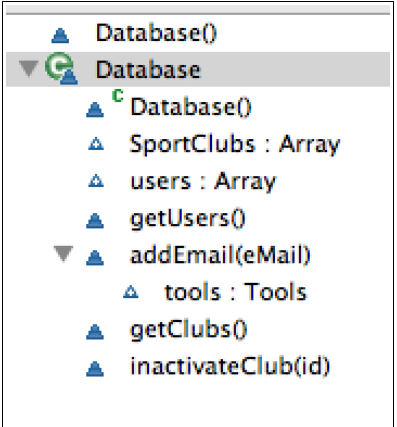
Rapport: test 2dot5 fungerar inte som det ska.

Iteration 1 Delmål #3 Logga in → Delegera klubbadministratör

Pre → Klubbadministratör måste finnas som användare i systemet

Post → Klubbern har tilldelats en admin i form av en databas-användare som är kopplad genom användarens ID

Primärt Flöde <ol style="list-style-type: none">1. hämta samtliga users i "Database"2. "Output" i webbläsare (GUI)3. Välj användare att bli klubbadmin	Alternativt flöde <ol style="list-style-type: none">1. Inget svar från "Database"2. Generera felmeddelande
--	---

Källfil: Database	Database	Kommentar
<pre>{ isactive: true, name: "testLaget", clubId: 1, adminId: 1 }, {</pre>		Database: Klassen database finns sedan tidigare. Objekt av klubbar utökas med ett fält som heter adminId. Metoder som lagts till för delmål är: setTeamAdmin(teamId,)

Enhetstestning delmål #3

Inga tester gjorda eftersom iteration #1 gått över sju timmar.

Eftersom testning inte gick framåt läggs den på is för att återgå till i framtida iteration.

Iteration 1 Reflektion

Jag tog beslutet att fortsätta på laboration tre som utgångspunkt inför laboration fyra. Det som slog mig var att trots att jag själv skapat allt inom detta projektet så var ställtiden för att komma igång betydligt längre än vad jag hade trott. Även att jag kände att laboration nummer tre gav mig en viss förståelse om hur flödet kan se ut inom den iterativa utvecklingeng så var det ändå svårt att komma in i flow från fögåen labb. Att synka ihop mot något som tidigare producerats upplevde jag som långtifrån självklart. Till slut valde jag att börja leka lite med koden och bara test runt i det som tidigare var gjort. Faktum är att det hjälpte till att korta ned ställtiden avsevärt. För mig fungerade/fungerar det helt enkelt inte så bra att bara titta på koden (även om man skrivit den själv) och utifrån detta tro att man kan börja producera diverse flödesdiagram mm, för att sedan jobba efter som mall och komma framåt.

Något jag ska ta med mig framöver är att det är bra att rota runt i koden, förstöra och testa runt lite så kommer man mycket snabbare in i vad jag vill kalla flow.

Efter att ha "lekt" runt lite i koden tog jag papper och penna förstod att jag behövde ett flödesdiagram för att skriva kod till punkt "Logga in → radera klubb" i iteration 1. Därefter tycker jag absolut att de t lossnade och jag blev mycker mer produktiv.

Jag noterar halvvägs, tidsmässigt, in i projektet att jag inte lagt i princip någon tid alls på användningfall. Troligen beror detta på att föregående labb redan hade användningsfall producerade som jag kunde nyttja. Men nästkommande iterationer borde bli annorlunda

Första halva av den första iterationen kändes riktigt bra efter en del strul med att komma igång efter juledigheten. Dock försvann känslan när jag skulle designa tester, implementer och exekvera. Hade en bra känsla av tester från tidigare labb. Men det var som bortblåst. Måste hitta rätt tänk där igen!

Iteration 2, på detaljnivå (kravanalys → design → implementation → test → reflektion)

Mål (Milstolpe) #1

id	Moment	Kalkyltid	Verklig tid	check
1	• Planering	00:30	01:00	X
2	• Användningsfall <ul style="list-style-type: none"> ▪ analyseras 	00:45	00:45	X
3	• Design <ul style="list-style-type: none"> ▪ identifieras ▪ specificeras ▪ implementeras 	01:30	01:00	X
4	• Test <ul style="list-style-type: none"> ▪ designas ▪ specificeras ▪ implementeras ▪ exekveras 	00:45	00:45	66%
5	• Reflektion	00:30	00:10	
6	• Gränssnitt CSS & HTML	-	-	-

Mål (Milstolpe) #2

id	Moment	Kalkyltid	Verklig tid	check
1	• Planering	00:30	01:00	X
2	• Användningsfall <ul style="list-style-type: none"> ▪ analyseras 	00:45	45:00	X
3	• Design <ul style="list-style-type: none"> ▪ identifieras ▪ specificeras ▪ implementeras 	01:30	01:00	X
4	• Test <ul style="list-style-type: none"> ▪ designas ▪ specificeras ▪ implementeras ▪ exekveras 	00:45	00:45	X
5	• Reflektion	00:30	00:10	X
6	• Gränssnitt CSS & HTML	-	-	-

Mål (Milstolpe) #3

id	Moment	Kalkyltid	Verklig tid	check
1	• Planering	00:30	01:00	X
2	• Användningsfall <ul style="list-style-type: none"> ▪ analyseras 	00:45	45:00	X
3	• Design <ul style="list-style-type: none"> ▪ identifieras ▪ specificeras ▪ implementeras 	01:30	01:00	X
4	• Test <ul style="list-style-type: none"> ▪ designas ▪ specificeras ▪ implementeras ▪ exekveras 	00:45	00:45	X
5	• Reflektion	00:30	00:10	X
	• Gränssnitt CSS & HTML	-	-	-

Tidssummering för samtliga delmål i iteration 2

12 TIM

12

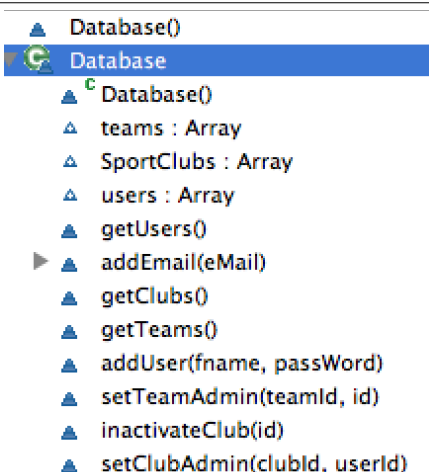
Iteration 2 Delmål #1 Logga in → Registrera lagadministratör

Aktörer	→	Klubbadministratör
	→	Lagadministratör
Pre	→	Klubbadministratören systemanvändare (redan registrerad)
Post	→	Lagadministratören har tilldelats adminbehörighet för ett lag

Primärt Flöde	Alternativt flöde
<ol style="list-style-type: none"> 1. Klubbadministratör loggar in 2. Väljer att lägga till användare 3. Användaren tilldelas behörighet till att vara ett lags lagadministratör 	<ol style="list-style-type: none"> 1. Klubbadministratören loggar in 2. Väljer att lägga till användare 3. Användaren tilldelas behörighet till en lag i en inaktiv klubb.

Designelement och eventuella kommentarer och förtydliganden

Kommentarer:

Database		Kommentar
<pre> this.teams = [{ clubId: 1, isActive: true, name: "P09", gender: "male", adminId: 1, secondAdminId: 2, }]; </pre>		<p>Database:</p> <p>Databas måste kompletteras med en array innehållandes lagobjekt</p> <p>Metoder som lagts till för detta delmål är:</p> <p>addUser(fname, passWord) getTeams() setTeamAdmin(teamId, id)</p>

Enhetstestning Iteration 2 delmål #1

Testning avser att testa nya metoder för gällande delmål.

Testfixtur:	Database
Testsvit:	testDatabase (jsTestDriver)
Version:	2.0

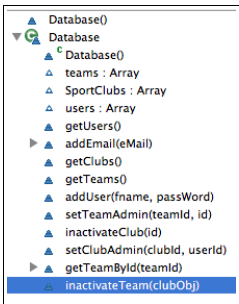
Anrop	Ansvarig	Förväntat	Kommentar	testId	Check
addUser("test", 999)	Andréas	pass	Kontrollerar användare lagts till	2.06	X
getTeams()	Andréas	pass	Kontroller om en array returnerats	2.07	Nej!
setTeamAdmin	Andréas	pass	Inte implementerad alls!		

Iteration 2 Delmål #2 Logga in → Radera lag

Aktörer	→	Lagadministratör
Pre	→	Lagadministratören är tidigare registrerad i systemet
Post	→	Ett lag har raderats ur den klubb som lagadministratören tillhör

Primärt Flöde	Alternativt flöde
<ol style="list-style-type: none"> 1. Lagadministratör loggar in 2. Lista för klubbens samtliga lag genereras 3. Välj ett lag att radera (inaktivera) med hjälp av ett lags unika id. Verifier att id finns i aktuel klubba 4. Lagadministratör bekräftat inaktivering av lag. 5. Laget inaktiveras 6. Lagadministratör får ”feedback” av system 	<ol style="list-style-type: none"> 1. Lagadministratör loggar in 2. Lista för klubbens samtliga lag genereras. 3. Välj ett lag att radera (inaktivera) med hjälp av lagets unika id. 4. Lagadministratören nekar inaktivering 5. Ny chans för inaktivering 6. Lagadministratör nekar inaktivering 7. Flödet avslutas utan ytterligar meddelanden

Designelement och eventuella kommentarer och förtydliganden

Database	Kommentar
	<p>Database:</p> <p>Databas måste kompletteras med en array innehållandes lagobjekt</p> <p>Metoder som lags till för detta delmål är:</p> <p>getTeamById(clubId) : array inactivateTeam(teamObj) : void</p> <p>Ej ännu tillagd: messageInactivatedTeam(clubObj)</p>

Enhetstestning Iteration 2 delmål #2

Testning avser att testa nya metoder för gällande delmål.

Testfixtur:	Database
Testsvit:	testDatabase (jsTestDriver)
Version:	2.0

Anrop	Ansvarig	Förväntat	Metodens funktion	testId	Check
getTeamById(clubId)	Andréas	pass	Hämtar ett lag med des id	2.08	X
inactivateTeam(teamObj)	Andréas	pass	Skicka med ett clubObj så inaktiveras teamet	2.09	X
messageInactivatedTeam(clubObj)	Andréas	pass	Returnerar ett meddelande om ett inaktiverat lag	2.10	Nej

Obs!

Metoden messageInactivatedTeam(clubObj) kommer att implementeras i slutet av iteration två eller tre om det finns tid att tillgå.

Iteration 2 Delmål #3 Logga in → Delegera lagadministratör

Aktörer	→	Lagadministratör (första)
	→	Lagadministratör (andra)
Pre	→	Lagadministratören är tidigare registrerad i systemet
Post	→	En användare är tillagd i systemet med behörighet att vara administratör för ett lag

Primärt Flöde	Alternativt flöde
<ol style="list-style-type: none">1. Lagadministratören lollar in som lagadministratör2. Möjlighet att delegera lagadministratör erhålls genom att klicka på en länk3. Uppgifter fylls i för blivand lagadministratör (fname och password)4. Användare skapas i database.users5. Laget tilldelas en ”andra” lagadministratör6. Lagadministratör få meddelande om att processen genomförts med lyckat resultat	<ol style="list-style-type: none">1. Lagadministratören lollar in som lagadministratör2. Möjlighet att delegera lagadministratör erhålls genom att klicka på en länk3. Uppgifter fylls i för blivand lagadministratör (fname och password)4. Användare skapas i database.users5. Laget tilldelas en ”andra” lagadministratör Lagadministratör få meddelande om att processen inte gick att genomföra genom ett felmeddelande6. Nytt registrering erbjuds på nytt

Database
*setTeamSecondAdmin(teamId,userID)

* Metod som kompletterar befintlig ”Database” sedan tidigare

Enhetstestning Iteration 2 delmål #3

Testning avser att testa nya metoder för gällande delmål.

Testfixtur:	Database
Testsvit:	testDatabase (jsTestDriver)
Version:	2.0

Anrop	Ansvarig	Förväntat	Metodens funktion	testId	Check
setTeamSecondAdmin(teamId,userID)	Andréas	pass	Sätter id för second admin av ett team	2.11	X

Känslan inför iteration två var att jag inte kände mig speciellt nöjd med den första iterationen i sin helhet. Delar av den kändes ok. Det var framförallt vid testning som jag sladdade ut i mina tankar och tappade fokus. Inför denna iterationen tänkte jag att försöka bli mer disciplinerad och strukturerad. Iterera enligt de planerade punkterna utan att hoppa fram och tillbaka.

En sak som jag upptäckte var när jag skulle skapa funktionalitet för att radera lag. För att kunna radera ett lag så måste laget vara skapat. Detta har inte framgått av tidigare iterationer. Däremot finns ju "sportsClub" i den simulerade databasen. Jag valde att förbise detta under själva iterationen. Men en sån grej kan vara bra att notera och eventuellt föra över till en nästa iteration. Alternativt implementera denna funktionalitet om det visar sig att det finns lite tid kvar inom den aktuella iterationen.

Angående användningsfall så tyckte jag det var lite knepigt att hålla isär och faktiskt direkt begripa mina tidigare skapade användningsfall i tidigare iterationer. I framtiden gäller känns det som att jag helst håller de korta för att eventuellt underlätta något.

Efter två planerade iterationer så känns det som att jag nu börjar förstå vad en iterativ utveckling handlar om. Men det innebär också att jag börjar ifrågasätta hur jag tidigare kodat. Nu känns det väldigt uppstyplat på något sätt. Min förhoppning är dock att jag ska få in att jobba iterativt utan att egentligen fundera på att jag gör det. Ett problem som jag haft är att jag på de planerade timmarna inte hunnit implementera något grafiskt gränssnitt. Detta löste jag genom att i js-konsollen istället testa att skicka in de data som behövde för att simulera flödet från ett usecase. I den sista iteration ska jag ge mig på att istället tro att jag hinner implementera css och html så ska jag istället simulera detta med js funktionerna prompt och confirm. Jag vet att det här inte är nåt att föredra. Men i testsyfte och att slippa rodda runt med css och html tycker jag att det kan vara värt det. Så jag hittar nån känsla för att få ut genererad data ut till någon form av gränssnitt.

Iteration 3, på detaljnivå (kravanalys → design → implementation → test → reflektion)

Mål (Milstolpe) #1

id	Moment	Kalkyltid	Verklig tid	check
1	• Planering	00:30	00:20	
2	• Användningsfall <ul style="list-style-type: none"> ▪ analyseras 	00:45	00:50	
3	• Design <ul style="list-style-type: none"> ▪ identifieras ▪ specificeras ▪ implementeras 	01:30	01:40	
4	• Test <ul style="list-style-type: none"> ▪ designas ▪ specificeras ▪ implementeras ▪ exekveras 	00:45	00:30	
5	• Reflektion	00:30	00:40	
6	• Gränssnitt CSS & HTML	-		

Mål (Milstolpe) #2

id	Moment	Kalkyltid	Verklig tid	check
1	• Planering	00:30	00:20	
2	• Användningsfall <ul style="list-style-type: none"> ▪ analyseras 	00:45	00:50	
3	• Design <ul style="list-style-type: none"> ▪ identifieras ▪ specificeras ▪ implementeras 	01:30	01:40	
4	• Test <ul style="list-style-type: none"> ▪ designas ▪ specificeras ▪ implementeras ▪ exekveras 	00:45	00:30	
5	• Reflektion	00:30	00:40	
6	• Gränssnitt CSS & HTML	-	-	-

Mål (Milstolpe) #3

id	Moment	Kalkyltid	Verklig tid	check
1	• Planering	00:30	00:20	X
2	• Användningsfall <ul style="list-style-type: none"> ▪ analyseras 	00:45	00:50	X
3	• Design <ul style="list-style-type: none"> ▪ identifieras ▪ specificeras ▪ implementeras 	01:30	01:40	X
4	• Test <ul style="list-style-type: none"> ▪ designas ▪ specificeras ▪ implementeras ▪ exekveras 	00:45	00:00 Ej påbörjad!	X
5	• Reflektion	00:30	00:40	
	• Gränssnitt CSS & HTML	-	-	-

Tidssummering för samtliga delmål i iteration 2

12 TIM

12

Iteration 3 Delmål #1 Logga in → Registrera gymnast/lagmedlem

Aktörer	→	Lagadministratör
	→	Gymnast
Pre	→	Ett lag har en tilldelad lagadministratör, som kan logga in.
Post	→	En gymnast eller lagmedlem har lagt in i systemet och tillhör "lagadministratörens" lag.

Primärt Flöde	Alternativt flöde
<ol style="list-style-type: none">1. Lagadministratör loggar in2. Startar metod som lägger till en ny gymnast3. Fyller i formulär för gymnast4. Medlem adderas databa (simulerad endast...)5. Respons ges på samtliga lagmedlemmar med ny medlem "highlighted"	<ol style="list-style-type: none">1. Lagadministratör loggar in2. Startar metod som lägger till en ny gymnast3. Fyller i formulär för gymnast4. Proceduren avbryts

Designelement och eventuella kommentarer och förtydliganden

TeamMember	Database
id fname lname clubId teamId TeamMember()	TeamMembers: array popTeamMembers(teamMember) : void getTeamMembers() : array

Enhetstestning Iteration 3 delmål #1

Testning avser att testa nya metoder för gällande delmål.

Testfixtur:	Database
Testsvit:	testDatabase (jsTestDriver)
Version:	2.0

Anrop	Ansvarig	Förväntat	Kommentar	testId	Check
popTeamMembers()	Andréas	pass	Test implementerad eller inte?	2.12	X
getTeamMembers()	Andréas	pass	Test implementerad eller inte?	2:13	X

Iteration 3 Delmål #2 Logga in → Radera gymnast/lagmedlem

Aktörer	→	Lagadministratör
	→	Gymnast
Pre	→	Ett lag har en tilldelad lagadministratör, som kan logga in.
Post	→	En gymnast eller lagmedlem har raderats i systemet.

Primärt Flöde	Alternativt flöde
<ol style="list-style-type: none">1. Lagadministratör loggar in2. Startar metod som som raderar (inaktiverar) lagmedlem med medlemmens id3. Dialogruta frågar efter medlems Id4. Lagadministratör bekräfta inaktivering5. Systemet rendera meddelande på skärm	<ol style="list-style-type: none">1. Lagadministratör loggar in2. Startar metod som raderar (inaktiverar) lagmedlem med medlemmens id3. Dialogruta frågar efter medlems Id4. Lagadministratör förser systemet med Id5. Lagadministratör bekräftar INTE att fullfölja6. Proceduren avbryts

Designelement och eventuella kommentarer och förtydliganden

Database	TeamMember	Källfil: main.js
inactivateTeamMember()	id fname lname clubId teamId isActive	init.inactivateTeamMember() En länk med id="inactivateTeamMember" triggat metoden i init.inactivateTeamMember()
	TeamMember()	

Enhetstestning Iteration 3 delmål #2

Testning avser att testa nya metoder för gällande delmål.

Testfixtur:	Database
Testsvit:	testDatabase (jsTestDriver)
Version:	2.0

Anrop	Ansvarig	Förväntat	Metodens funktion	testId	Check
inactivateTeamMember()	Andréas	pass	Test. Metod implementerad?	2.14	X

Iteration 3 Delmål #3 Logga in → Redigera gymnast/lagmedlem

Aktörer	→	Lagadministratör
	→	Gymnast
Pre	→	Ett lag har en tilldelad lagadministratör, som kan logga in.
Post	→	En gymnast eller lagmedlem data har förändrats i databasen

Primärt Flöde	Alternativt flöde
<ol style="list-style-type: none">1. Lagadministratör loggar in2. Länk aktiverar metod för att redigera en lagmedlem.3. Selektion utförs med medlems Id4. Formulär genereras förfyllda med medlems befintliga datan5. Skicka/Uppdatera ändrade data	<ol style="list-style-type: none">1. Lagadministratör loggar in2. Länk aktiverar metod för att redigera en lagmedlem.3. Selektion utförs med medlems Id4. Formulär genereras förfyllda med medlems befintliga datan5. Proceduren ska avbrytas

Designelement och eventuella kommentarer och förtydliganden

TeamMember	Källfil: main.js
renderForm() updateTeamMember()	init.inactivateTeamMember() En länk med id="updateTeamMember" triggat metoden i init.updateTeamMember()

Planering

Min avsikt var/blev att fundera på vad som skulle förbättras från den föregående iterationens planering. Sättet jag tänkte på var att tänka efter på saker som jag inte var nöjd med, eller saker man skulle kunna ha gjort bättre, om just planeringen hade utförts och varit annorlunda på något sätt. Jag upplever generellt att jag har mer på fötterna nu angående planering. I alla fall projektet av denna art. På det hela taget känner jag mig relativt nöjd med planeringen vid min summering. Min känsla är att jag behöver lägga mitt fokus på att gå från användningsfall (med hänsyn till krav) till design och implementationen av kod.

Användningsfall

Eftersom jag inte hade något flödesdiagram eller hade analyserat användningfallet för iteration tre och dess delmål analyserade jag visionsdokument för att försöka att komma till klarhet i vad det var som skulle utöfras. Jag har noterat att jag tänker i kod när jag skriver användningsfall. Jag vet inte om detta begränsar en slutprodukt. Kanske det är negativt?

Design

Delmål/Milstolpe #1

Användningfallen tyckte jag gick ganska smidigt att utföra. Däremot upplever jag fortfarande inga klara gränser mellan mina olika arbetsmoment. Jag hoppar fram och tillbaka nästan lite okontrollerat. Vid design kan jag helt plötsligt upptäcka att jag petar i ett test eller med justeringar av användningsfall. Detta trots att jag försökt att arbeta mig från punkt till punkt för vardera delmål.

Delmål/Milstolpe #2

Äntligen så börjar det lossna lite och jag försår flödet bättre och bättre utan att behöva ”återblicka” ständigt.

Test

Valde att utföra minimalt med tester för delmål ett och två. Tyckte att arbete inte flöt på vidare genom första iterationen. Andra iterationen kunde jag troligen effektivt adderat ytterligare tester. Men jag valde att om tid finns efter delmål tre så läggs tid på fler tester.

4 Slutreflektion

Det har varit en krävande uppgift på flera sätt. För varje iteration som genomförts har det inneburit att en del frågor väcks. Vad är det som tar tid när man ska programutveckla? Hur ska jag planera kommande ”skarpa” projekt? Behöver jag lära mig mer om att planera kod innan man faktiskt kodar?

En intressant sak som jag framförallt kände av i sista iterationen var att planeringen på detaljnivån drog mig framåt på flera sätt. Framförallt värt att notera var att analyserna av användningsfallen tillsammans med ett enkelt diagram tvingade mig att försöka tänka fram kodlösningar på ett annat sätt som jag tidigare fått insikt av. På något sätt upplever jag det som att jag i fortsättningen kommer att bli bättre på att lägga upp mina jobb. Troligen kommer jag lägga mer analys innan jag kodar mestadels för att jag tror att det kan spara tid när jag utvecklar. Dessutom är det också så att om man lägger ned lite tid på planeringen så får man också samtidigt bra och värdefull dokumentation som kan vara bra att ha i framtiden.

Det här med tester är något som jag tycker är både kul men också tidskrävande. Det fina med tester är att det ger möjlighet att känna att man får mer ”feedback” på att koden verkligen fungerar som det är tänkt. Råkar man göra en justering på ett sätt som skadar logiken i koden kan det eventuellt innebära att man faktiskt upptäcker detta på ett tidigare stadie än om man inte testat sin kod. Sådär i efterhand är jag väldigt nöjd att jag tillbringade en dag med att försöka hitta ett testverktyg för javascript. Det fanns en del att välja på men jag valde ett som heter jsTestDriver (xUnit) som är utvecklat av google. Eftersom jag använde mig av IDE:en Eclipse för detta projekt valde jag att installera den som ett plugin direkt under Eclipse. Innan hade jag aldrig jobbat med asserts och ”felmeddelande” i testning. Men efter laboration fyra förstår jag vade det handlar om och det är absolut något som jag kommer att använda mig av i framtiden.

Ytterligare en sak som jag lagt märke till är att det är svårt att lära om sig hur man lägger upp sin planering när man ska koda. Innan detta miniprojekt har jag i princip planerat all min kodning i själva kodandet utan att dokumentera. Men dokumenterandet gör att man lär känna sin kod snabbare och på ett effektivt sätt. Intressant!

Värt att nämna angående det jag utvecklat i laboration fyra, är att det inte på långa vägar är fulländat och komplett. Det återstår mängder av arbete och planering om man skulle få det på det sättet som min bild av applikationen ser ut i inne i mitt huvud och tankarna som jag har hur saker borde fungera. Men det viktiga hoppas jag ändå trots allt är att jag faktiskt till slut fått en insikt om varför och hur man kan jobba med iterativ mjukvaruutveckling.