

Vad är en bra projektmetod för små IT-projekt?

Ett försök att besvara frågan görs i kursen II1302 ”Projekt och projektmetoder” vid KTH ICT

Adam Kaisoum^{#1}, Joakim Rödin^{#2}, Chosrat Ismail^{#3}, Maher Jabbar^{#4}, Kim Askebris^{#5}

*[#]Kungliga Tekniska Högskolan, ICT Sektionen
Electrum, Kistagången 16, 164 40 Kista*

kaisoum@kth.se

joarod@kth.se

chosrat@kth.se

mahej@kth.se

askebris@kth.se

Abstract— Kurs vid KTH ICT. Syfte och mål med kursen – ”Vad är en bra projektmetod för små IT-projekt?” Kursens metod för att uppnå kursens syfte och mål. Resultat av kursens metod - uppfylls syfte och mål med kursen. Kursen Projekt och Projektmetoder utförde ett projekt om fem studenter. Målet med detta projekt var att projektmedlemmarna skulle få ökad uppfattning och erfarenhet av att bedriva ett mindre IT-projekt samt prova på olika projektmetoder. Inläsning av omfattande litteraturstudier av olika projektmetoder var grunden till att uppnå detta. Varje projektgrupp hade en produkt att utveckla parallellt med litteraturläsning. Vid utveckling av självaste produkten utnyttjades Scrum tillsammans med XP med parprogrammering för att bedriva projektet i sig. Avslutningsvis utvärderar projektgruppen vad som de ansåg vara det mest vitala för en projektmetod för små IT-projekt.

Keywords— Projekt, Scrum, IT, Projektmetoder, Kanban

I. OM DETTA DOKUMENT OCH UNDERSÖKNING

Denna rapport är skriven av fem KTH-studenter som studerar till högskoleingenjör med inriktning dator teknik. Denna rapport är riktad till alla individer som varit aktiva i kursen Projekt och Projektmetoder, det vill säga lärare och studenter, samt personer som är intresserade av små IT-projekt. Innehållet i denna rapport är en översiktlig blick över projektets arbetsgång och gruppens allmänna reflektioner som har uppstått under projektets gång. Innehållets allmänna pålitlighet är relativt hög då innehållet baseras på gruppens erfarenhet av det som skett under projektet, dock sänks pålitligheten en aning på grund av bristande erfarenheter från arbetslivet.

II. INTRODUKTION

Detta kapitel beskriver vilka frågor som gruppen har haft och sökt svar på, hur denna undersökning

har gått till och vilken metod som används. Projektmedlemmarnas roller kommer också förklaras och preciseras.

A. Bakgrund

Syftet med kursen Projekt och Projektmetoder II1302 är att ge studenterna en inblick på hur ett projekt kan utvecklas med tiden, samt få en praktisk erfarenhet av ett projekt inom IT-sektorn. För att studenterna skall uppfylla kursens mål och få ett godkänt betyg måste de delta i en projektgrupp där en teknisk produkt levereras i slutet av kursens gång. Projektgruppen delas in i början av kursen och kommer inte förändras under projektets framfart, detta i likhet med hur det ser ut i mindre IT-projekt ute på marknaden. Projektgruppen använde sig av projektmetodiken SCRUM som kommer utvecklas vartefter iterationerna går framåt och erfarenheter kommer bygga på kunskapen om SCRUM som projektmetodik för hela gruppen. Rapporten ska vara en summering av gruppens erfarenhet av det genomförda projektet och de lärdomar som tagits samt ge svar på frågan ”Vad är en bra projektmetod för små IT-projekt?”. IT-projektet sker i en mindre grupp, varje gruppmedlem bidrar med den kompetens individen besitter från tidigare studier inom dator teknikprogrammet som behövs för att utveckla produkten.[10] Gruppen består av studenter som studerar dator teknik och ekonomi, detta innebär att gruppen inte besitter kunskap inom hårdvara och det har då gett upphov till att produkten endast utvecklats ur ett mjukvaroperspektiv. Vilken projektmetodik som anses vara bäst för små IT-projekt är en subjektiv bedömning men erfarenheterna som samlats kommer ge svaret ur gruppens perspektiv.

B. Problemformulering

Frågeställningen som skall besvaras är, vad menas med en bra projektmetod? Vilka risker kan uppstå och hur kan vi arbeta för att undvika dessa? Denna projektmetod skall ge upphov till ett bra struktur inom projektet, samt underlätta arbetat i form av effektivitet och flexibilitet. Genom det lyckas leverera en produkt i rätt tid och budget enligt kravspecifikation.

C. Undersökningsstrategi/lösningsstrategi

Lösningsstrategi som används för att besvara och undersöka den övergripande frågeställning, vad är en bra projektmetod för små IT-projekt.

Olika ansvarsområden delades upp i gruppen och på så sätt för att fördela arbetet för att effektivt driva projektet framåt. Varje gruppmedlem värderar sitt ansvarsområde och dess innehörd för att försöka besvara den övergripande frågeställningen.

Gruppen använde de agila projektmetoderna SCRUM och XP med parprogrammering för att få en bättre förståelse vad de olika metoderna innehåller. Nedan listas rollerna och deras ansvarsområden.

Projektledare

Projektledarens roll har ett stort ansvar med olika delar som daglig planering, ansvar för projektgruppens arbete, personalhantering, rapportering och riskhantering.

Projektet har drivits enligt metoder för SCRUM och projektledarens roll har anpassats till det. I kompendiet Scrum and XP from the trenches beskriver Kinberg att projektledaren ska agera som en coach, en ledare och inte en chef.[2] Projektledaren ska inspirera, hålla gruppmedlemmarna motiverade, se till så att dynamiken i gruppen är bra då det kan förekomma händelser som påverkar prioriteringen av utvecklingen som på så sätt kan få gruppmedlemmar att fråga sig om det berodde på deras jobb eller något annat, där är dynamiken och förståelsen för känslor oerhört viktig..

Ivar Jacobssons Essence Kernel-Modell delar upp management och ledarskap i två delar. Ledarskap ska ansvara, inspirera, coacha och motivera och management gäller projektplanering, budget och riskhantering.

Projektledaren bör ser till att sakerna sker i rätt ordning och vid rätt tidpunkt för att lyckas med projektet. Kontinuerligt se över planeringen och vara väldigt tydligt om detaljerna i varje iteration och följa upp det dynamiskt. Relaterade dokument för projekt ska framställas där frågeställningar om projektet ska besvaras.

Kund- och Kravansvarig

Med rollen kund- och kravansvarig innehåller det att vara en länk mellan projektgruppen och kunden.

Empatisk, god kommunikatör och analyserande förmåga krävs i den rollen. Att stå för en god kommunikation både med kund och projektgrupp för att samla in och balansera behoven hos alla intressenter. Framföra kundens krav, synpunkter, klagomål och förändringar, bemöta kundens krav och diskutera en kravspecifikation för att vidarebefordra till projektgruppen. Hålla det simpelt med stories då det är ett tydligt och simpelt sätt att kommunicera kraven och får alla att jobba på projektet mot samma mål. Förstå helhetsbilden där det kan förtydligas genom att använda användningsfallsmodellen. Fokus på värdet där det är viktigt att förstå hur systemet ska användas av användarna och där kan användning av use cases vara nyttigt. Det är tre punkter som lyfts fram i kompendiet USE-CASE 2.0 från Ivar Jacobson för kund- och kravansvarig. [4]

Kravansvarig ska diskutera med kund om hur det vill att produkten ska levereras, tidsramen för projektet, ta fram en produktspecifikation och om de kan leverera produkten. Kravansvarig ansvarar för att ta fram modeller som användningsfallsmodell, use-cases för att förtydliga och leverera krav till projektgruppen samt att demonstrera för kund om arbetet med produkten är färdigt eller hur långt det har kommit..

Arkitekt

Arkitekten ansvarar för den tekniska designen dvs skapa en övergripanden systemstruktur. Modellen ska tydligt beskriva hur alla hårdvara- och mjukvarukomponenter kommunicerar med varandra. Modellen byggs på kravspecifikationen som kund- och kravansvarig tagit fram tillsammans med kunden.

Utvecklingsansvarig

Den som har rollen utvecklingsansvarig måste ha god teknisk kompetens då de tillsammans med kund- och kravansvarig tar fram en kravspecifikation som är till grund för arkitekturen. Ska besitta kompetens för att utveckla, forma och driva utveckling av mjukvara efter modellen som arkitekten tagit fram. Utvecklingsansvarig tar beslut som vilka programmeringsspråk som ska användas, design på mjukvaran. De ska också ha förståelse för hur mjukvaran som ska utvecklas ska samverka med användare.

Beroende på hur stort projektet är och projektmetodiken så är ansvaret hos utvecklingsansvarig olika.

Då det här projektet är relativt litet har projektmetodiken XP med parprogrammering varit utgångspunkten för utvecklingen. Parprogrammering användes då det är ett litet system som har utvecklats och med den metoden

har alla varit inblandade i utvecklingen samt det ger en ökad möjlighet att hitta buggar och fel direkt under utvecklingen..

Testare

Testarens roll innebär att inspektera arkitekturen för att sedan ta fram en testplan samt utveckla tester som ska säkerställa funktionaliteten i systemet. Syftet med testerna är att säkerställa att systemet uppfyller kundens krav och en komplett produkt levereras utan buggar. Testerna är också väldigt bra för att demonstrera för kunden vad produkten innehåller och att man kan leverera en fullt fungerande produkt. Det är viktigt att utföra dessa tester kontinuerligt för att tidigt upptäcka eventuella mjukvarufel och korrigera dessa för att inte fortsätta bygga systemet på funktioner som inte fungerar korrekt. Tester skrivs som testfall efter förväntat systembeteende och där man anser att systemet möjligtvis kan krascha.

Till det här projektet har den testansvariges arbete anpassats till storleken på projektet samt systemets storlek. Testansvarig har då byggt användartester från arkitekturen som testar funktionaliteten i systemet för att uppnå kravspecifikationen. Testerna har varit tillgängliga som test-cases för alla gruppmedlemmar så att de kontinuerligt kan utföra dessa tester varefter utvecklingen gått framåt.

D. Relaterade arbeten

Gruppen har ingen kurs under samma period som relaterar till II1302 men tidigare arbeten som relaterar är Ingenjörsmetodik som lästes i början av utbildningen av hela gruppen. Ett par av medlemmarna läser en kurs inom modellering och Enterprise Architecture (EA) som relaterar till hur man som arkitekt behöver tänka för att inte bara tekniskt kunniga personer skall förstå modellerna som visar upp produkterna som tillverkas/utvecklas.

E. Avgränsningar

III. TEORI OCH INGENJÖRSPRAXIS

Detta kapitel listar och i viss mån beskriver teorier och ingenjörspopraxis som används i undersökningen. Det finns två underkapitel, Litteraturstudie och Förstudie.

F. Litteraturstudie

Kursen tillhandahåller en hel del litteratur under kursens gång som alla kursmedlemmar skall läsa för att få en bredare kunskap om hur man kan arbeta i projekt. Gruppen har under planeringen av varje iteration gått igenom vad man skall läsa och sen har det legat på individens egna axlar att läsa

och ta till sig. Genom att diskutera de olika delarna i iterationerna så har medlemmarna kunnat ta tillvara på mer kunskap och förståelse från varandra. En övergripande lista på de böcker, artiklar och andra litterära källor visas upp nedan.

Övergripande källor för hela projektet

- Software engineering, 10th edition. Ian Sommerville [1]
- Scrum and XP from the trenches - How we do Scrum, 2nd edition Henrik Kniberg [2]
- Survey of agile tool usage and needs, Version 5. Gayane Azizyan, Miganoush Katrin Magarian & Mira Kajko-Mattsson [3]
- Use-Case 2.0 - The guide to succeeding with Use Cases. Ivar Jacobsson, Ian Spente & Kurt Bittner 2011 [4]
- Scrum [5]
- Fasindelning [6]
- Arbeta i Projekt: individen, gruppen, ledaren: studentlitteratur (Eklund, 2010) [7]
- Projektgrunder (Eklund, 2010)
- Person- och grupp dynamik (Eklund, 2010)
- Projektets hälsa och status, (OMG, 2013)

G. Förstudie

Vid projektets början var vissa riktlinjer fastställda, tillämpning av projektmetodiken SCRUM, detta var förutbestämt utav läraren, Anders Sjögren. Samt grund utseendet på arbetsstavlan. För att få läsa denna kurs, var studenterna tvungna att ha genomfört kursen Ingenjörsmetodik. Även i den kursen fick en projektgrupp ett teknisk projekt att genomföra med en annan typ av projektmetod. Kunskapen som var till stor nytta var att arbeta i iterativt, vilket var till stor hjälp i kursen Projekt och Projektmetoder. Kursen påbörjas med att deltagarna blir omburda att studera litteratur angående SCRUM, Nedan kommer en beskrivning av teorin på de projektmetoder som gruppen har tagit del av under kursens gång.

Scrum

Scrum är en projektmetodik för systemutveckling skapad av Jeff SutherLand och Ken Schwaber.[5] En världskänd metod som används världen över och även av de allra största företagen i världen, SCRUM som agil metod används väldigt ofta och utbrett inom projekt.[3] Projektmetoden använder sig av en produkt backlog, lista över det funktionalitet som skall existera hos produkten,

därmed finns det ingen anledning att använda sig av en kravspecifikation. Fördelen av att använda sig av en produkt backlog är att funktionaliteterna kan omprioriteras under utvecklingen beroende på vad kunden önskar. Nackdelen som kan uppstå är att förutsättningarna kan förändras under en så kort tid vilket kan leda till risken för förvirring inom gruppen ökar, lösningen blir att sätta upp en riskhanteringsplan över hur man bemöter problemet om detta skulle uppstå.

Tre vitala roller inom SCRUM: [5]

- **Produktägare** har som uppgift att upprätthålla kontakt med kunden, har även hand om ärenden genom att hantera önskemål från kundens sida. Produktägaren kan även välja vad som skall prioriteras vid eventuella ändringar när det kommer till utveckling av produkten. Denna post består av en person.
- **Scrummästare** är gruppens ledare och mellanhanden mellan produktägare och utvecklaren. Har även hand om planeringen inför varje iteration och vad som skall göras.
- **Utvecklare** har hand om utveckling och skapandet av produkten. Vanligtvis brukar utvecklingsgruppen bestå utav tre till fem personer. Dessa besitter olika typer av kompetens.

Beståndsdelar som finns inom Scrum:

- **Product backlog:** En lista med alla krav på produkten som är ordnade efter prioritet, dessa krav kallas för stories. Varje story har ett beskrivande namn på kravet, vad för prioritering kravet har, en kort beskrivning på vad som skall göras samt en uppskattning på hur mycket tid som skall läggas ner på för att lösa uppgiften.
- **Sprint** är en tidsperiod under ett projekt, ett projekt består utav ett antal sprintar. Varje sprint planeras i förväg av Scrummästaren och avslutas med att man utför en granskning och en utvärdering. En sprint är oftast inte längre än 30 dagar men kan vara så kort som en vecka. [6]
- **Sprint backlog** är en lista över det olika stories som ska utföras under en viss sprint.
- **Inkrement**, under varje sprint blir stories klara och funktionerna blir redo att användas i produkten.
- **Sprintplanering.** Planering sker för varje sprint, denna planering presenteras för hela gruppen i något som kallas för sprintmöte som anordnas inför varje sprint. I detta möte diskuteras vad som behöver uppnås under sprinten och vad gruppen tror sig kunna hinna göra, detta görs genom att bryta ner dessa krav till stories.

- **Sprint retrospective**, efter varje utförd sprint så utför gruppen en så kallad sprintgranskning, dessa möten går ut på att gruppen reflekterar över resultatet och vad som kan förbättras inför nästkommande sprint eller projekt. Eventuellt kan även ett scrum demo genomföras.
- **Scrum demo:** Under en sprintgranskning kan ett scrum demo hållas där redovisning av resultatet presenteras för gruppen.
- **Daglig scrummöte** är ett kort möte inom gruppen som hålls i början av dagen för att förmedla om något nytt uppstått eller kunna informera om något annat relevant, det är även till för att ge en översikt om vad alla arbetar på.

Användningsfall (Use- Case)

Användningsfall eller även kallat Use-case är ett sätt att inhämta krav på ett nytt system inom programvaruutveckling. Varje användningsfall är en beskrivning av hur ett system skall interagera med andra komponenter eller användare för att kunna uppfylla en specifik funktionalitet för en användare. Med hjälp av användningsfall kan man få en överblick över produktens funktionalitet. Ivar Jacobson förespråkar att använda sig av sex stycken principer för lyckas generera en god användningsfall:

1. **Keep it simple by telling stories**
2. **Understand the big picture**
3. **Focus on value**
4. **Build the system in slices**
5. **Deliver the system in increments**
6. **Adapt to meet the team's needs**

Inledningsvis skall identifiering av de mest vitala funktionerna göras för att sedan kunna dela upp dessa i "slices" för att förminka ner till mindre delar, detta gör det möjligt för utvecklarna att få en klarare bild över hur systemet är uppbyggt då system ofta kan vara väldigt komplexa.

Kanban

Kanban är ett redskap som skapades av Toyota för att signalera eller synliggöra materialbehov i fabriken, detta blev senare anpassat så att detta kunde användas inom mjukvarubranschen. Vision med detta var att skapa ett antal olika kolumner som representerar olika tillstånd samt ha kort/post-it lappar som representerar olika stories. Alla dessa kort hämtas från product backlog, sedan beroende på vilket tillstånd varje story befinner sig i så placeras de i in dem olika kolumnerna. Nästa kolumn är "checked out", dessa kort har börjats processas. Sedan finns kolumnerna "test" och "complete". Naturligtvis kan det existera mer

kolumner än dem som nämnts. Självklart existerar det regler på hur många kort som får befina sig på respektive kolumn eftersom man vill uppnå ett så stort arbetsflöde som möjligt.

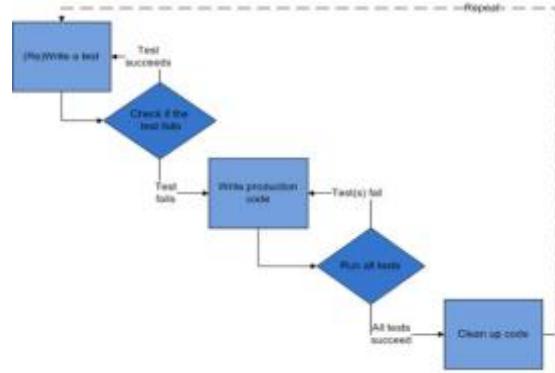


Figur 1, Kanban

TDD

Utvärderingsmetoder Test driven development (TDD) är ett arbetsätt som används flitigt inom mjukvarubranschen och får arbetet att ske inkrementellt. TDD förespråkar att ingen kod får ändras eller infogas förrän ett testprogram har skapats. Koden skall ha testas lokalt av utvecklaren i det nya testprogrammet och först när det har exekverats och fått godkänt är den nya koden godkänd till att införas i programmet. Figuren nedan beskriver hur denna process går till:

1. Framtagning av nya funktioner som produkten kräver.
 2. Ett automatiserat test genereras för själva funktionen, resultat för testet förmedlas vidare vid ett godkänt resultat samt icke godkänt resultat.
 3. Vid godkänt resultat, implementeras testet med andra godkända tester och körs för att kunna säkerställa att testet är kompatibelt med resten av testerna.
 4. När samtliga tester genomgått testning och godkänts kan det nya funktionaliteterna infoga i systemet.



Figur 2, TDD

1. Arbetstavla

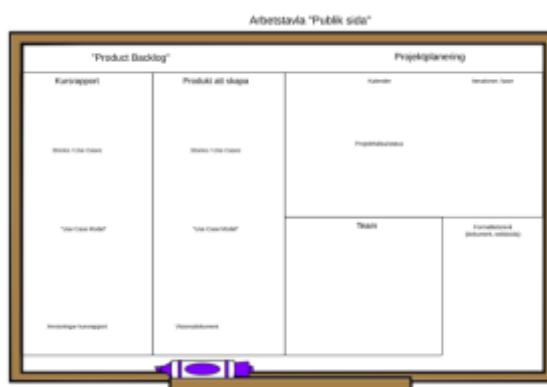
Arbetstavla består av två sidor, en arbetssida och en publik sida. På arbetssidan är en presentation på hur gruppen ligger till i den aktuella sprinten. Den mest aktuella och användbara delen av arbetssidan är överblicken av hur stories för varje sprints flyttas genom varje kolumn(check out, test, Done). Lapparna som är placerade på de tre olika kolumnerna är de aktuella stories för den sprinten. När en stories har bearbetats och gjorts klart så placeras den på kolumnen done. För varje dag som går uppdateras tavlan utefter vad som har gjort i gruppen. Till höger om arbetssidan finns en Burn Down som är en beskrivning över hur gruppen har disponerat tiden under den aktuella sprinten. Man startar högst upp i Burn Down diagrammet, dvs med den maximala tiden och sedan går diagrammet endast för varje timme eller point (om detta används) som används. Projektets arkitektur är även presenterad på arbetssidan av tavlan, detta är för att få en helhetsbild av den tekniska delen av produkten och dess uppbyggnad.



Figur 3, Arbetssidan, bild på hur grunden av arbetssidan har sett ut från början.

Arbetstavlans andra sida är den publika sidan, på denna sida finner man projektets dokumentation

samt en planering över projektet. Detta kan vara i olika former, men vi använde ett Gantt-schema. Vid intresse av projektets status så kan det representeras av dokumentation Alpha som beskriver vilken fas projektet befinner sig i.[8] Publika sidan är främst för utomstående individer utanför projektgruppen som vill ha en inblick på hur projektet utvecklas och även få en uppfattning över hur själva slutprodukten kan se ut. Dokument som projektdefinition, arkitekturbeskrivning, vision, användningsfallsmodell samt krav som bestämts av produktägaren visas här.



Figur 4, publika sidan, bild på hur grunden av den publika sidan har sett ut under kursens gång.

IV. UNDERSÖKNINGSMETODER

Detta kapitel beskriver vilka metoder som används i undersökningen. Metoderna är valda och specificerade så att de skall kunna ge svar på ett antal följdfrågor som identifierats i denna undersökning. Först anges frågorna och sedan följer metodbeskrivning.

H. Frågor att besvara i undersökningen

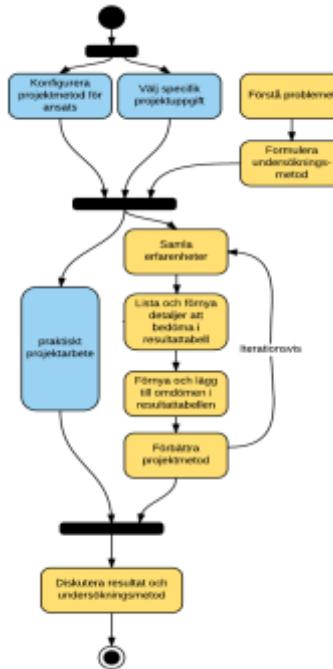
1. Hur kan man bedöma om projektet har varit lyckat?
2. År administrationstyngden befogad i den här typen av projekt?
3. Vilka ansvarsroller skall användas som ansats i projektet?
4. Vad för risker finns det i och med denna typ av projekt?

I. Metodbeskrivning

Den centrala metoden i undersökningen är att avgöra om olika valda projektpraktiker och arbetsätt är ”bra” och om de bidrar till att göra hela projektprocessen bra. Åstadkommer eller skapar projektet rätt saker och konstrueras lösningar på bästa sätt? Metoden för att samla data i denna fråga blir induktiv då erfarenheten i gruppen är ytterst liten. På grund av bristande erfarenhet av

användning av projektmetoder inom gruppen så testades olika typer av projektmetoder. Detta gjorde det självklart betydligt svårare att avgöra vilken projektmetod som var bäst lämpad för just detta och andra små IT-projekt. Arbetsrollerna för projektet har fördelats utifrån individens kompetens och samtidigt tagit hänsyn till önskemål varje individ hade. Varje arbetsroll har ett specifikt område att ansvara för samt inhämtning av all sorts data som behövs, naturligtvis skall alla gruppmedlemmar ha en inblick över projektet.

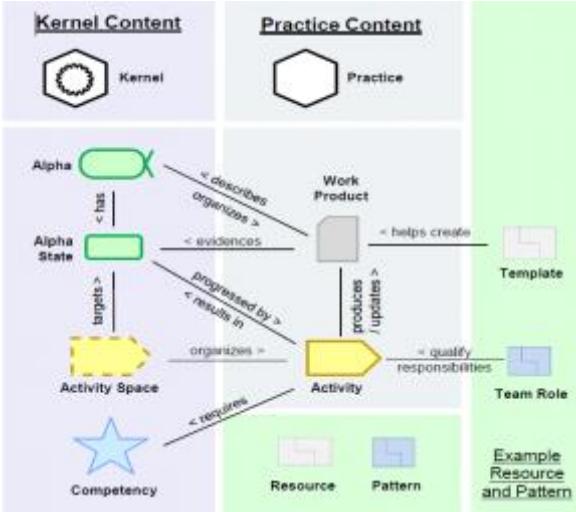
Metod 1: Undersökningsmetod, se figur. De gula fälten är aktiviteter som kopplar till själva undersökningen. Metoden följer principer för vetenskaplighet enligt Andersson och Ekholt (Andersson & Ekholt, 2002 pp 17). Projektmetoden som användes var Scrum, som i sin tur är baserad på iterationer, figur 1 är en animerad bild på hur arbetsflödet sett ut under projektets gång.



Figur 5, Undersökningsmetoden som användes för ”Vad är bra projektmetod för små IT-projekt”.

Metod 2: Begrepp Begrepp som används följer om möjligt OMGS standard *Essence*. [8] Följande bilder listar illustrativt centrala begrepp. I denna artikel kommer de engelska begreppen att fritt översättas till svenska då risken för missförstånd anses liten. Eftersom gruppen har arbetat med projektmetodiken SCRUM och fått lärdom om begrepp i denna projektmetod så har projektmetoden OMG standard används relativt

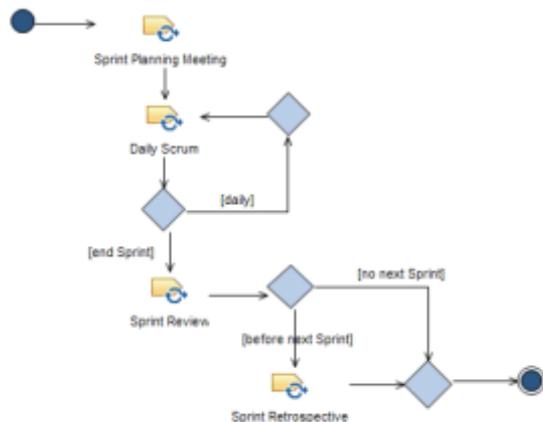
undermedvetet. Scrum är en påbyggnad av OMG kernel, dvs att SCRUM följer standarden.[8]



V. GENOMFÖRANDE

Projektet består av fem iterationer, varje iteration pågår under en period på cirka två veckor. Iterationerna inleds med ett sprint möte där planering sker och vad för stories som ska utföras för den aktuella sprinten samt tidsestimering för stories i sprinten. För varje dag hålls ett scrummöte ”Daily-scrum” där information förmedlas inom gruppen samt vad som behöver göras och om något nytt har uppstått. Vid slutförd iteration genomförs ett scrum demo, där en diskussion genomförs angående vad som uppfyllt under sprinten. Avslutningsvis hålls ett sprint retrospective möte inom gruppen där utvärdering genomförs av arbetet under sprinten.

Genomförandet av iterationer följer helt och hållt mallen för s k ”sprint” i Scrum och beskrivs enklast med medföljande aktivitetsdiagram, se figur 7. [6]



Figur 7, Iteration/Sprint. [11]

För att hålla koll på arbetet användes en arbetsstavlå som ständigt uppdateras för att administrera projektet, detta gör det lättare att följa arbetet.

Gruppen arbetade stora delar av projektet tillsammans i Electrum, Kista. Det planerades in i början av projektet genom en gemensam kalender som gruppmedlemmarna hade tillgång till. Arbetade timmar rapporterades även in i ett delat excel-dokument för att få en översikt över hur många timmar respektive gruppmedlem arbetat, det användes även för att kontrollera effektiviteten i gruppen.

J. Projektledning (Chosrat)

K. Management & Arkitektur (Joakim)

L. Testansvarig(Maher)

M. Utvecklingsansvarig(Kim)

N. Kravansvarig(Adam)

VI. RESULTAT

Projektet genomfördes i huvudsaken med projektformen Scrum på ett lyckat sätt, produkten har utvecklats och uppfyller kundens krav. Under projektets gång har gruppmedlemmarna fått uppleva hur det är att arbeta i en projektgrupp där arbetet är uppdelat utifrån arbetsroller. Då gruppdynamik sattes på prov vid ett par tillfällen användes riskhantering enligt Scrum och det resulterade i positiva resultat. Gruppen har kommit fram till att de absolut viktigaste ansvarsrollerna i små IT-projekt är projektledare, kravansvarig & utvecklingsansvarig. Detta då projektledaren håller övergripig kontroll, kravansvarige håller reda på vad som måste finnas och utvecklingsansvarig ser till att det utvecklas. De andra titlarna är viktiga men inte absolut nödvändiga i små projekt. Administrationen är som tidigare beskrivit alldeles för stor för denna storlek av projekt och tar för mycket tid, så det är inte värt den mängden resurser som det kostat. Att bedöma om projektet är lyckat är en väldigt subjektiv bedömning. Bedömningen beror mycket på vad man var ute efter, gruppen anser detta projekt som lyckat om vi bedömer vad vi har lärt oss. Vi har lärt oss oerhört mycket som kommer kunna tas med framöver i utbildningen och arbetslivet. Själva produkten blev lyckad då alla grundkrav uppfylls, men det finns vissa saker som skulle kunna utvecklas också som vi inte gjorde.

VII. DISKUSSION

Ett av syftena med denna rapport var att besvara frågan "Vad är en bra projektmetod för små IT-projekt?", något som besvarats genom att göra en distinkt undersökningsmetod. Med hjälp av inläsning av teoretisk litteratur samt det praktiska arbetet med produkten har projektgruppen analyserat och utvärderat de nya projektmetoderna som används för detta arbete. Metoddiskussion samt resultatdiskussion kommer sammanfatta samt sammanställa gruppens värderingar kring undersökningsmetoderna och resultatet.

K. Metoddiskussion

Utformningen av frågeställningen som ställdes för undersökningsmetoden ansåg gruppen vara välformulerad samt okomplicerad att besvara fastän frågeställningen kan besvaras på en rad olika aspekter. En väsentlig fråga som uppstod kring projektgruppen var hur stor vikt som skulle läggas på dokumentering för ett sådant litet projekt. Slutsatsen som drogs var för ett projekt av denna kalibern så skall mindre vikt läggas på dokumentation men dock inget man bör avskaffa helt. Givetvis vid alltmer omfattande projekt bör mer vikt läggas på dokumentering.

En väsentlig samt vital gren för gruppens framgång är hur lätt implementering av en projektmetod för gruppen. Denna aspekt kanske inte har lika stor påverkan då det existerar en projektmedlem med erfarenhet av scrum som kan driva gruppen till framgång då komplikationer uppstår.

Projektgruppen anser att frågeställning är legitim då risken av riktningsförändringar av arbetet kan ske över tid under ett projekt samt gäller det även riskhanteringen eftersom en välplanerad riskhanteringsplan är nyckel till att få ett framgångsrikt projekt.

En aspekt som kom att ha en stor påverkan på vad projektgruppen hade för uppfattning kring de olika projektmetoderna var tiden. Projektgruppen ansåg att tiden inte räckte till för att kunna ge en rättvis, korrekt samt pålitlig bedömning över projektmetoderna. Lösningen hade varit att få arbeta med undersökningsmetoden under en längre tid för att kunna analysera allt djupare och kunna ge en alltmer omfattande bedömning.

L. Resultatdiskussion

Projektgruppen ansåg att projektmetoden scrum var alltför omfattande då inläsningsperioden av teori blev betydligt längre än beräknad för att en fullständig förståelse för hantering av projektmetoden. Detta ledde till ineffektivitet och mindre tid till att arbeta med produkten samt att denna projektmetoden lade betydligt mer fokus på dokumentation som fick produktionen av varan att försenas en aning. Fördelen med scrum var att

strukturen blev lätt att följa samt att arbetet skedde inkrementellt samt att dynamiken förbättrades efter varje iteration eftersom avvikeler skedde under föregående iteration. Ännu en fördel som gruppen tog del av var att prioritera om funktioner under projektets gång beroende på funktionens betydelse samt att hantering av risk enligt scrum medförde till anpassning beroende på den aktuella situationen. Resultatet av att använda scrum för relativt små IT-projekt ansåg projektgruppen vara tillfredsställande med anledning av att inlärningsperioden var så pass lång samt alltför omfattande dokumentering.

Kanban påminner givetvis om scrum men ansågs inom projektgruppen vara aning bättre anpassad för mindre IT-projekt eftersom dokumentationen inte var lika omfattande. Sorgligt nog tillämpade projektgruppen inte denna projektmetodik som de hade önskad eftersom tillämpningen skedde endast under en iteration.

Systemutvecklingsmetoden TDD var något som projektgruppen kunde få bättre uppfattning över samt att inlärningsprocess var betydligt kortare än scrum. Det lades inte alltför mycket tid på dokumentation vilket vissa individer i gruppen uppskattade eftersom större fokus kunde läggas på själva arbetet samt att samtliga individer fick en ökad inblick av utveckling av koden på grund av genereringen av testfallen som var ett krav för att implementera nya funktioner till produkten. Riskhantering för TDD fungera utmärkt!

Programutvecklingstekniken som uppskattades betydligt mer än de andra var parprogrammering. Det är enkelt att implementera denna metod i utvecklingssyfte. Denna metod får arbetet att fortskrida betydligt snabbare på grund av att ingen dokumentation behöver utföras samt att fel upptäcks i tidigt stadie. Denna projektmetod ansåg gruppen vara bäst lämpad för gruppens framgång.

M. Bidrag till vetenskaplighet, ingenjörserfarenhet (studenterfarenhet?)

Genom detta projekt så lär vi oss hur en modern ingenjör utför sitt arbete på ett ingenjörsmässigt sätt främst med den agila metoden Scrum, även viss vetenskaplighet kan finnas i utförandet, en bättre projektmetod kan tas fram för mindre IT-projek, vilken kan komma att användas av andra som är ute efter sådana projekt metoder.

SLUTORD

Gruppens medlemmar hade begränsad kunskap inom området men drev projektet från start till mål. Stötte på en del hinder på under projektets gång men med hårt arbete och engagemang från gruppen var det inget som vi inte kunde ta oss förbi. De

utmaningarna som gruppen stötte på under projektet har varit väldigt lärorikt och det är något gruppmedlemmarna kommer att ta stor nytta av i framtida projekt.

REFERENSER

- [1] Ian Sommerville. (2015). Software Engineering, 10th edition.
- [2] Henrik Kniberg. (2015). Scrum and Xp from the trenches - how we do scrum, 2nd edition.
- [3] Gayane Azizyan, Miganoush Katri Magarian & Mira Kajko-Mattsson. Survey of agile tool usage and needs. Version 5.
- [4] I.Jacobson, I. Spence, and C. Bittner, "Use case 2.0 - The guide to succeeding with use-cases."
- Kruchten, P. B. (1995). The 4+ 1 view model of architecture. *Software, IEEE*, 12(6), 42-50.
- [5] Scrum Basics Available Online. Retrieved 23 May, 2017, from [Https://www.youtube.com/watch?v=aOrsVfjbQZ4](https://www.youtube.com/watch?v=aOrsVfjbQZ4)
- [6] Scrum Process. Available Online. Retrieved 23 May, 2017 from <https://www.youtube.com/watch?v=29dnS7XGgqs>
- [7] Eklund, S. (2010). *Arbete i projekt: individen, gruppen, ledaren:* Studentlitteratur.
- [8] OMG. (2013). Kernel and Language for Software Engineering Methods (Essence). 1.0. Retrieved 23 May, 2017, from <http://www.omg.org/spec/Essence/1.0/>
- [9] Elvesæter, B., Benguria, G., & Ilieva, S. (2013). *A comparison of the Essence 1.0 and SPEM 2.0 specifications for software engineering methods*. Paper presented at the Proceedings of the Third Workshop on Process-Based Approaches for Model-Driven Engineering.
- [10] Högskoleingenjör inom data teknik. Retrieved 30 may, 2017 from <https://www.kth.se/social/program/tidab/>
- [11] Elvesæter, B., Striewe, M., McNeile, A., & Berre, A.-J. (2012). Towards an Agile Foundation for the Creation and Enactment of Software Engineering Methods: The SEMAT Approach. Second Workshop on Process-based approaches for Model-Driven Engineering (PMDE 2012).

Grupp 10

Rollbeskrivning

2017-05-30

KTH Electrum, 16440 Kista, Sverige
Joakim Rödin

joarod@kth.se

Innehållsförteckning

1	Introduktion.....	2
1.1	Inledning.....	2
1.2	Dokument omfattning	2
1.3	Dokument Översikt	2
2	Gruppens uppgift	3
3	Riktlinjer	3
4	Ansvar	3
4.1	Kompetens.....	4
4.2	Administrations och dokumentationsupplägg.....	4
5	Guide.....	4
5.1	Tips.....	4
5.2	Vanliga misstag	4

1 Introduktion

1.1 Inledning

Detta dokument syftar till att vägleda en projektmanager igenom framförallt ett litet IT-projekt. Det kommer inte ge några exakta svar men förhoppningsvis leda en annan i samma roll framåt.

1.2 Dokument omfattning

Dokumentet behandlar följander punkter:

- Kommunikation i team
- Administration under IT-projekt
- Organisering av dokumentation

1.3 Dokument Översikt

- **Gruppens uppgift**
- **Riktslinjer**
- **Ansvar**
- **Kompetens**
- **Administrations och dokumentationsupplägg**
- **Tips**
- **Vanliga misstag**

2 Gruppens uppgift

Projektgruppen har fått i uppgift av kunden att utveckla ett system för att hantera en elektronisk display, kunden vill kunna uppdatera skärmen varv än kunden är i världen. Displayen ska enligt kundens beställning kunna skriva ut meddelanden för att informera om var kunden är för tillfället. Denna information skall kunna ändras varv kunden än är så länge den har tillgång till internet.

3 Riktlinjer

Saker som är bra att tänka på för att hantera managementrollen i gruppen.

- Gruppkoordination är en otroligt viktig del för en manager, dessa punkter är grunden för det.
 - Kommunikation
 - Administration
 - Organisering
 - Resursplanering
 - Kostnadsrapportering
- Planering av projektet, resurs och tidsmässigt.
- Dokumentationshanteringen måste vara tydligt och tillräckligt enkelt.

4 Ansvar

En projektmanager ska administrera och hantera gruppens sätt att arbeta. Genom att ge gruppen verktygen för hur dokumentation skall göras samt hur mycket resurser(tid) som ska läggas på de olika delarna i projektet. Projektmanagern är den som har ansvar över att dokumentationen görs och att planeringen resursmässigt görs och att kostnaden rapporteras vartefter.

- Planering tillsammans med projektledaren
- Se till att dokumentation görs under tiden och själv göra mycket av det.
- Resursåtgångsdokumentation
- Hantera platsen där all dokumentation lagras.
- Kontrollera och se till att deltagarna i projektet inte har dokument och liknande på andra platser än där dessa skall vara.

En projektmanager skall utöver detta ovan även jobba tillsammans med projektledaren, tätt tillsammans för att minska riskerna för att grupp dynamik som projektledaren hanterar och dokumentation och administration inte skapar problem mellan varandra eller i gruppen.

4.1 Kompetens

En duktig projektmanager måste vara strukturerad, göra sakerna som skall göras i tid och se till att gruppmedlemmarna gör detsamma, ansvaret att hantera resursåtgången för projektet ligger som tidigare sagt på projektmanagern och detta är en kompetens som måste finnas hos denna. Allt detta tillsammans med motiveringsförmåga är de viktigaste kompetenserna.

4.2 Administrations och dokumentationsupplägg

Hur man lägger upp dokumentationen finns det lika många sätt som grupper, men vi valde att i detta projekt använda oss av en gemensam disk i molnet för att alla skulle kunna arbeta samtidigt och se hur det går framåt för de olika personerna och hela gruppen.

5 Guide

5.1 Tips

Först och främst så måste projektmanagern vara organiserad och ha ett sinne för att hålla koll på saker och ting. Du behöver ha egenskapen att få en överblick över projektet. Se vad som behöver göras nu för att kunna ta nästa steg. Att kunna arbeta med en projektledare vid sidan som håller ihop gruppen och informera denne om hur resurserna går åt och var det kanske går för mycket eller vad som behöver mer är oerhört viktigt. Kommunicera ut vad du vill och hur du vill att det skall göras, var lyhörd för vad gruppen tycker, det kan finnas mycket kunskap och tankar som är värda att använda sig av.

5.2 Vanliga misstag

Ett stort misstag är att man inte tror att dokumentationen tar någon tid alls, dokumentationen behöver göras på ett kontrollerat sätt, allting behöver ha sin plats i mappstrukturen. Desto bättre man gör detta desto lättare blir det i slutändan. Problemet som kan uppstå under tiden är att dokumentation som andra i gruppen gör försvinner då de inte lagrar det på rätt plats eller liknande. Detta är någonting som lätt blir ett jätteproblem och måste hanteras direkt genom att man är tydlig med hur allting görs och att det inte är okej att lagra dokumentation på fel platser eller inte fylla i timrapporteringar för att visa hur mycket resurser man lagt.

Grupp 10 Rollbeskrivning

2017-05-30

KTH Electrum, 16440 Kista, Sverige
Joakim Rödin
joarod@kth.se

Innehållsförteckning

1	Introduktion.....	2
1.1	Dokumentets Syfte	2
1.2	Dokumentets Omfattning	2
2	Kravfunktioner.....	3
3	Implementationsbegränsningar	3
4	Tillgängliga Interfaces	4
5	Interfacebehov.....	4
6	Intern Struktur.....	5

1 Introduktion

1.1 Dokumentets Syfte

Detta dokument finns för att visa hur Raspberry Pi 2 (RP2) konfigureras för att kunna hantera vår webbsida och en databas. Detta dokumenteras nedan och leder läsaren igenom installation för att få en förståelse för hur det är konfigurerat.

Det omfattar även hur Login funktionen fungerar och implementeras.

Delete message-funktionen.

1.2 Dokumentets Omfattning

Detta dokument omfattar inte hur en RP2 fungerar utan enbart vilken programvara som installerats på den. Hur dessa har installerats och hur uppkopplingen av RP2 har gjorts både fysiskt och nätverksbaserat.

Loginfunktionen och Delete messages omfattas genom hur den är kopplad till de olika interfacen och vad som krävs för att den skall vara fullständig och hur den fungerar. Koden kommer visas i bild men funktionaliteten i koden kommer inte förklaras närmare.

2 Kravfunktioner

Denna sektion kommer visa upp vad för funktioner som är nödvändiga för de tre olika delarna, vad som krävs för att funktionerna skall anses klara och tillföra systemet det som behövs.

Funktioner som kommer finnas på hemsidan

- Skriva meddelande
- Logga in (Utveckling ej klar)
- Radera meddelandehistorik
- Visa meddelandehistorik
- Hämta meddelande till skärmen
- Uppkoppling till databas

RP2 krav:

- Uppdaterat operativsystem
- Installationer:
 - MySQL
 - PHPMyAdmin
 - För att kunna hantera databasen och tabellerna där.
 - Apache server
 - PHP
- Uppkoppling mot LAN och möjlighet att nås från plats utanför LAN.

Login & Delete message krav:

- Möjlighet att logga in
- Sessionskontroll (login)
- Databashantering (Uppdatering)
- Användarinformationskontroll

3 Implementation

3.1 Begränsningar

RP2 måste sättas upp inom nätverket där den är konfigurerad och den kan inte använda sig av trådlös uppkoppling.

Kontroll av inloggad användare sker genom sessionskontroll.

Användare kan inte registreras genom vyn för login utan måste detta göras av ägaren av databasen och systemet.

Språkkonvention, PHP, SQL & Javascript är de språk som används.

3.2 Backend

RP2 installeras genom att de olika delarna installeras via kommandotolken. Administratörsrättigheter krävs för detta.

DeleteMessages (Figur 3) är en funktion som tömmer databasens ena tabell på meddelanden. Detta görs genom att funktionen kopplar upp sig mot databasen och sen skapar ett SQL statement som sedan tömmer tabellen. Skulle någonting gå fel så finns det ett exception som hanterar detta.

Loginfunktionen är skapad men inte helt klar. Kraven på funktionen syns i detta dokument men allt är som sagt inte utvecklat än. Detta beror på att det inte var ett grundkrav från kunden. Loginfunktionen är annars tänkt att använda sig av sessions dessa sessions ska kunna styra så att enbart den inloggade användaren kan ta bort och lägg till nya meddelanden. Det som redan finns på plats är uppkopplingen till databasen och fältet för att ta emot användarnamn och lösenord i loginvyn.

4 Tillgängliga Interfaces

Databasinterface I form av phpmyadmin för vidare konfiguration av databas.

Operativsystemsny mot skärm från RP2.

FTP möjlighet via annan host mot RP2.

En loginvy för användare som vill skriva meddelande till informationsdisplayen (IDP).

Ett jämförelseinterface för att kontrollera om användardatan stämmer överens med det som databasen har registrerat.

En "Clear History" knapp som tömmer historiken av meddelanden från hemsidan, denna finns under rubriken "Message history".

5 Interfacebehov

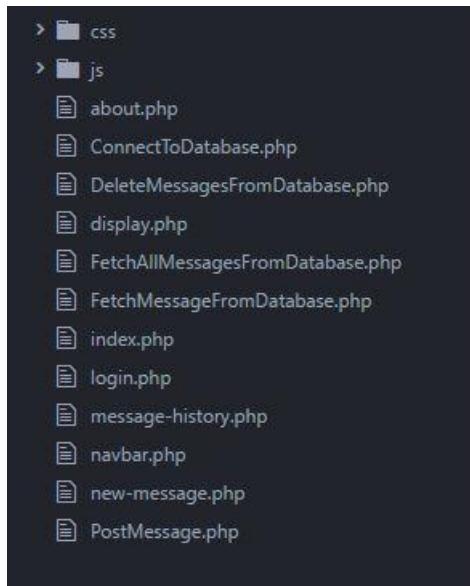
- Operativsystem installerat på RP2.
- En Databashanterare.
- Uppkoppling mot databasen för att hämta data som senare kan jämföras med användarens eller för att kunna ta bort meddelandehistorik.
- Interface som hanterar sessionvaliditet.

6 Intern Struktur

Översiktsbilder över de olika strukturerna och koden för DeleteMessages.



Figure 1 Databastabeller



Figur 2 Översiktig struktur

```
>DeleteMessagesFromDatabase.php
?>
    include 'ConnectToDatabase.php';
    try {
        $sql = "DELETE FROM messages";
        $q = $conn->query($sql);
    } catch(PDOException $e) {
        echo $e->getMessage();
    }
?>
```

Figur 3 Delete message

Grupp 10

Grupp 10

Rollbeskrivning

2017-05-29

KTH Electrum, 16440 Kista, Sverige

Chosrat Ismail

chosrat@kth.se



Innehållsförteckning

1	Introduktion.....	3
1.1	Inledning	3
1.2	Dokument omfattning	3
1.3	Översikt.....	3
2	Gruppens uppgift.....	4
3	Riktlinjer	4
4	Ansvar	4
4.1	Kompetens	4
4.2	Metod	5
5	Vägledning	5
5.1	Tips.....	5
5.2	Vanliga misstag.....	5
	Referenser	6

1 Introduktion

1.1 Inledning

Syftet med detta dokument är att förmedla gruppens uppgift, organisering och vägledande principer för en projektledare i ett mindre IT-projekt.

1.2 Dokument omfattning

Dokumentet behandlar följande punkter:

- Principer i att vägleda ett team
- Definiera projektstruktur
- Identifiera projektroller

1.3 Översikt

Dokumentets innehåll:

- **Gruppens uppgift**
- **Riktlinjer** – riktlinjer för en projektledare
- **Ansvar** – vad gör en projektledare
- **Kompetens** – vilken kompetens en projektledare bör besitta
- **Metod** – val av projektmetod
- **Tips** – vad en projektledare bör tänka på
- **Vanliga misstag** – misstag som projektledare lätt kan göra

2 Gruppens uppgift

Projektgruppen har fått i uppgift av kunden att utveckla ett system för att hantera en elektronisk display, systemet ska ha en funktion för att uppdatera information på displayen och det ska fjärrstyrts dvs beroende av var kunden befinner sig. Gruppen ska även leverera en produkt i form av en kursrapport där de behandlar arbetets gång och lärdomar.

3 Riktlinjer

Saker som är bra att tänka på för att leda projektgruppen.

- Gruppdynamik som är en väldigt viktig punkt
 - Inspirera
 - Motivera
 - Konfliktlösning
 - Driva
 - Omtanke
- Väldefinierad projektmetodik, i detta projekt användes metoden SCRUM[2].
- Tydliga arbetsroller och fördelning av arbete

4 Ansvar

Vad ska en projektledare göra?

Projektledaren ska agera som en coach, en ledare som inspirerar, motiverar och håller grupp dynamiken bra. Fördela arbetsroller med tydliga rollbeskrivningar och fördela arbetet bland projektgruppen. Projektledaren ska även ansvara för andra viktiga punkter som är listade här nedanför[1].

- Planering
- Budget
- Riskhantering
- Ta fram relaterade dokument ex, projektdefinition, iterationsplanering, osv.
- Uppdatera stakeholders för projektet

Förutom punkterna som nämns ovan ska en projektledare alltid följa upp arbetet i gruppen. Dagliga möten för att informera om något nytt har uppstått eller bara en lägeskontroll för att motivera gruppen.

4.1 Kompetens

För att vara en kompetent projektledare bör personen i rollen ha erfarenhet inom ämnet, vara strukturerad, god kommunikationsförmåga, administrativa erfarenheter och ett driv och engagemang för att motivera och leda en grupp.

4.2 Metod

Projektmetodik ska även väljas och det finns en del att välja mellan. Men beroende på projekt och i fallet för det här projektet handlade det om att utveckla ett mindre IT-system och då passade projektmetodiken SCRUM[2] väldigt bra.

5 Vägledning

5.1 Tips

Som projektledare bör du redan från början vara bestämd och tydligt med vad målet med projektet är och ta fram en välplanerad projektplan. Du ska ha en analyserande förmåga genom att ständigt se över grupp dynamiken. Om du som projektledare misstänker dålig dynamik ska du i tidigt skede hantera detta på bästa möjliga sätt utan att skapa oro i gruppen. Ständigt se över riskerna för att det inte ska uppkomma, vara väldigt tydlig gällande vad som ska göras och vem ska göra det och alltid följa upp arbetet, helst dagligen för att inte bli sena i planeringen.

5.2 Vanliga misstag

Ett väldigt vanligt misstag som kan ske är att vara en passiv projektledare och det kan smitta av sig till hela gruppen. Grupp dynamiken blir låg då och saker blir inte gjorde, de tar inte dig som projektledare på allvar och det slutar med att gruppen hamnar i en fas som är väldigt svår att tag sig ur.

Referenser

[1]<http://www.omg.org/spec/Essence/1.0/PDF/>

[2]Henrik Kniberg. (2015). Scrum and Xp from the trenches - how we do scrum, 2nd edition

**Grupp 10
Elektronisk display
Teknisk dokumentation**

2017-05-29

KTH Electrum, 16440 Kista, Sverige

Chosrat Ismail

chosrat@kth.se

Innehållsförteckning

1	Inledning	3
1.1	Syfte	3
1.2	Omfattning	3
2	Funktioner	3
3	Implementation	4
3.1	Begränsningar	4
3.2	Backend	4
4	Tillgängliga Interfaces	4
5	Interface Behov	5
6	Internal Structure	5

1 Inledning

Målet med projektet var att utveckla ett system med en Raspberry Pi som trådlöst kan uppdatera information på en elektronisk display.

1.1 Syfte

Syftet med det här dokumentet är att beskriva en del av den tekniska utvecklingen av systemet. Den behandlar funktionerna *login*, *postMessage* & *deleteMessageFromDatabase*.

1.2 Omfattning

Dokumentet behandlar funktionerna *login*, *postMessage* & *deleteMessageFromDatabase*. Den kommer visa koden i bild men inte beskriva koden för funktionerna men mera funktionalitet.

Kravspecifikationer, begränsningar, implementation är punkter som kommer att tas upp i dokumentet.

För att se systemets andra delar i detalj se övriga Tekniska dokumentationer.

2 Funktioner

Webbapplikationen ska uppfylla kraven från kund, där kunden vill uppdatera informationen på en elektronisk display genom webben, dvs oberoende av vart kunden befinner sig men så länge tillgång till internet finns.

Användaren ska genom att logga in på hemsidan få tillgång till en funktion som uppdaterar informationen på displayen.

Funktioner som kommer att finnas på hemsidan är.

- Login (utveckling ej klar)
- Etablera uppkoppling till databasen
- Radera meddelandehistorik
- Visa meddelandehistorik
- Hämta meddelande till displayen
- Skriva nytt meddelande

Kraven som ställs på funktionerna *login*, *postMessage* & *deleteMessageFromDatabase*

- Etablera en uppkoppling till databasen
- Uppdatera databasen
- Logga in genom verifiering med användare info från databasen
- Sessionskontroll (login)

3 Implementation

3.1 Begränsningar

Begränsningar för funktionerna *login*, *postMessage* & *deleteMessageFromDatabase*:

- Använda PHP, SQL & Javascript
- Följa kodkonvention
- God struktur på koden
- Sessionskontroll (login)

Det är punkter som projektgruppen bestämde att följa för att skapa koden för funktionerna som behandlar databasen, det underlättar för att ge vidare funktionen till gruppmedlemmen som hanterade vyerna och överföringen till Raspberry Pi för att implementera det.

3.2 Backend

Funktionen *postMessage* se figur 2 är implementerad med språket PHP och SQL.

Det funktionen gör är att den börjar med att etablera en uppkoppling till databasen och tar sedan emot meddelandet som användaren matar in på hemsidan. Sedan går den vidare för att lagra det nya meddelandet i databasen och om ett fel uppstår så tar funktionen upp det som en ”exception”.

Funktionen *deleteMessageFromDatabase* se figur 3 är implementerad på ett liknande sätt som *postMessage*. Den kopplar upp till databasen för att sedan skicka in ett SQL statement som raderar meddelanden i tabellen *messages*.

Login är en påbörjad funktion som inte slutfördes pga. att det inte var ett krav från kunden så prioriteringen var inte hög på den funktionen. Men utvecklingen påbörjades och behöver lite komplettering för att fungera korrekt.

Så den ser ut idag är att vyn för login har skapats i html och tar emot information från användaren, uppkoppling till databasen är etablerad. Det som behövs kompletteras är att verifiera inloggningen genom att jämföra användarens input med användarinfo från databasen och sedan starta en session för att användaren ska få tillgång till funktionen att skriva nya meddelanden. Saknas även en *logga ut* funktion där den avslutar sessionen så att användaren loggas ut.

4 Tillgängliga Interfaces

Vyerna som är tillgängliga för funktionerna *login*, *postMessage* & *deleteMessageFromDatabase* är vyn *login.php* för att hantera funktionen för login.

Vyn *new-message.php* där användaren ska skriva meddelandet som ska visas på displayen.

Vyn *message-history.php* där en ”Clear history” knapp finns tillgänglig som ska rensa historiken på hemsidan.

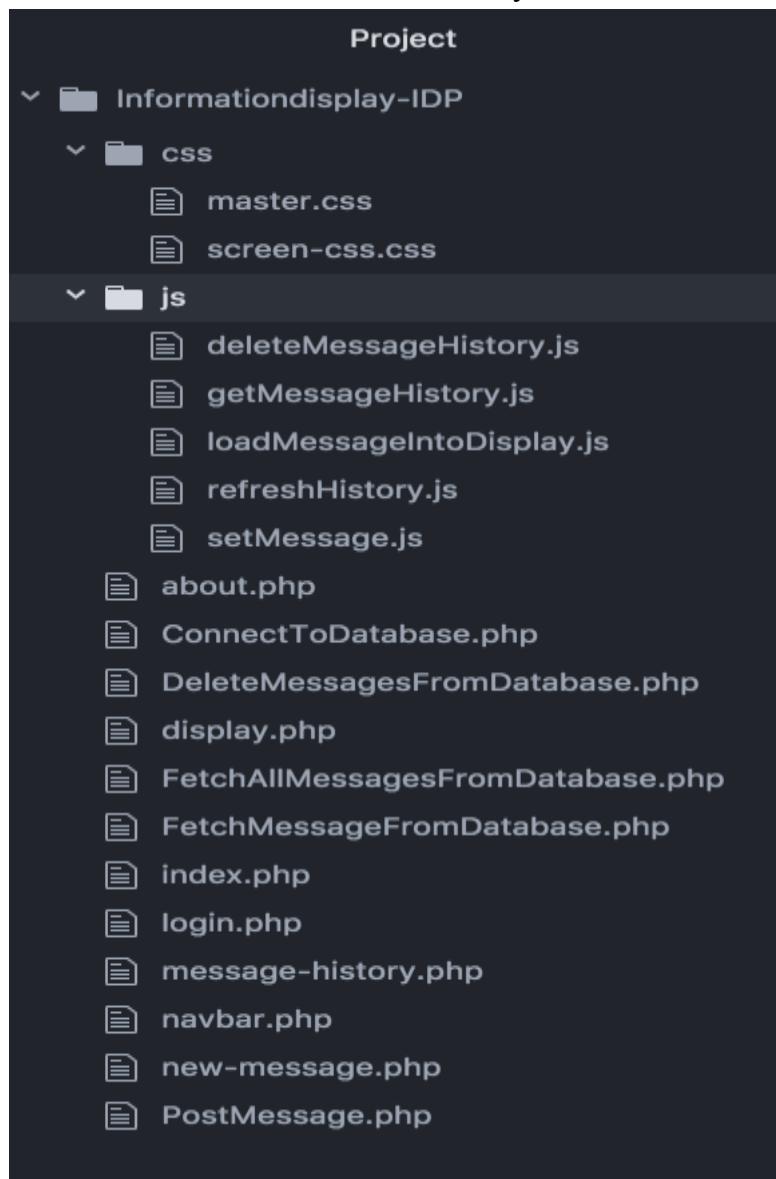
5 Interface Behov

Interfaces som behövs för att möjliggöra funktionerna.

- Databashanterare
- Uppkoppling till databasen
- Interface som hanterar session validitet

6 Internal Structure

Bilden visar strukturen för klasserna i systemet.



Figur 1, Översikt över klasserna

```
<?php
    include 'ConnectToDatabase.php';
    $message = $_POST['textMessage'];
    echo $message;
    try {
        $sql = "INSERT INTO message (message) VALUES ('$message')";
        $sql2 = "INSERT INTO messages (message) VALUES ('$message')";
        $sth = $conn->query($sql);
        $sth2 = $conn->query($sql2);
    } catch(PDOException $e) {
        echo $e->getMessage();
    }
?>
```

Figur 2, postMessage.php

```
<?php
    include 'ConnectToDatabase.php';
    try {
        $sql = "DELETE FROM messages";
        $q = $conn->query($sql);
    } catch(PDOException $e) {
        echo $e->getMessage();
    }

?>
```

Figur 3, deleteMessageFromDatabase.php

Projekt

<Uppdatering av meddelande> Use- Case Specifikation

Abstract

Use-caset <Uppdatering av meddelande> specificerar alla funktioner som påverkar produktens system. Dokumentet presenterar de nödvändiga funktionella samt de icke funktionella kraven för denna produkt.

Version History

Date	Version	Author	Description
24/04/2017	<1.0>	Adam Kaisoum	Skapad
05/05/2017	<1.2>	Adam Kaisoum	Utkast
17/05/2017	<1.3>	Adam Kaisoum	Uppdatering
24/05/2017	<2.0 >	Adam Kaisoum	Färdig

Innehållsförteckning

1	Introduktion	3
1.1	Dokumentets syfte	3
1.2	Dokumentets omfattning	3
1.3	Dokumentets översikt	3
2	Kort beskrivning	4
3	Förvillkor	4
Fel! Bokmärket är inte definierat.		
4	Grundflöde	4
5	Alternativt flöde	5
	5.1.1 Alternativt flöde 1: Komplikationer med att skriva/radera meddelande	
	5.1.2 Alternativt flöde 2: Applikationen svarar inte.	
	5.1.3 Alternativt flöde 3: Meddelandet inte existerar i databasen	
6	Förbättringsmöjligheter - Vidare Utveckling	6

Introduktion

1.1 Dokumentets syfte

Målet med denna tekniska dokumentation är att ge en tydligt inblick om det användningsfall som används för detta system. Detta kommer presenteras genom att ge en tydlig förklaring om själva systemet samt att hur navigeringen sker för en användare för att kunna skriva samt radera ett meddelande.

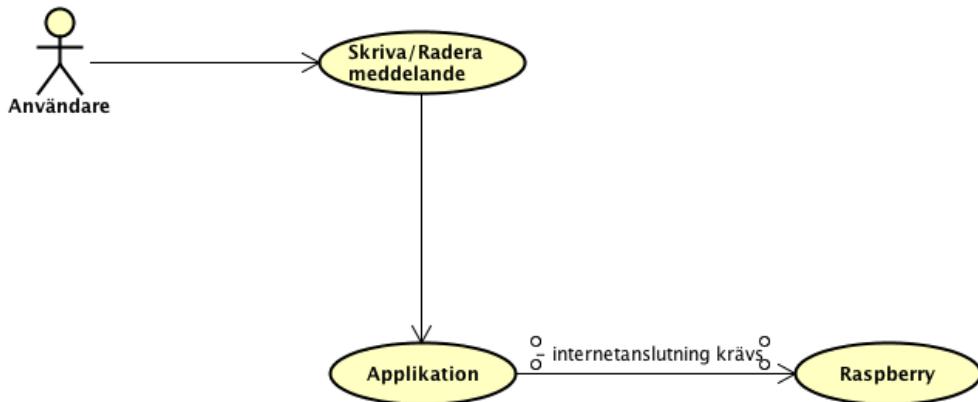
1.2 Dokumentets omfattning

Störst fokus kommer ligga på ett av systemets användningsfall. Detta kommer primärt förklaras med en bild som presenterar hur en användare kommunicerar med systemet. Avslutningsvis presenteras en stegvis beskrivning över hur navigationen för applikationen går till för ny användare.

1.3 Dokumentets översikt

- **Kort beskrivning-** kort förklaring av användningsfallet
- **Förvillkor-** För att ett utförande skall kunna göras i applikationen behöver användaren uppfylla ett par villkor.
- **Grundflöde-** En stegvis beskrivning hur en process kan ske.
- **Alternativt flöde-** Förförklaring kring hur ett flöde kan forma sig på olika sätt.
- **Eftervillkor-** Beskrivning av resultat
- **Speciella krav-** Funktioner som kan implementeras för förbättringar av produkt
- **Ytterligare information-** Extra information som kan komma till nytta.

2 Kort beskrivning



Figur 1: Användningsfall över hur en individ kan skriva/radera ett meddelande.

3 Förvillkor

Villkor som skall uppfyllas för att systemet skall kunna fungera korrekt.

- Användaren måste ha tillgång till en extern apparat exempelvis en dator för att kunna installera applikationen.
- Den externa apparaten skall ha internetanslutning tillgängligt.
- Resterande enheter måste vara synkroniserade med varandra samt applikationen.

4 Grundflöde

- Installation av applikationen för användaren.
- Synkronisering mellan applikationen och externa system.
- Navigering av applikationen för skriva samt radera meddelande.
- Information skickas från applikationen om vilken enhet som skall skriva samt radera meddelande till server.
- Säkerställer att rätt id från enheten mottags för att synkroniseringen sker rätt till.

5 Alternativt flöde

I denna del kommer en beskrivning ske utav alternativa flödet som kan uppstå utöver de grundflödet som beskrev i föregående avsnitt. För varje alternativt flöde som kan uppstå kommer beskrivningen enligt denna struktur:

- Vid vilket tillfälle det alternativ flöde kan äga rum.
- Anledning till varför den alternativa flödet äger rum.
- Händelslista över de alternativa flöden.
- Skedet då flödet återgår till grundflödet.

5.1.1 Alternativt flöde 1: Komplikationer med att skriva/radera meddelande

1. Individen som nyttjar applikationen lyckas inte skriva/radera ett meddelande eftersom ett nätverksanslutningsfel mellan applikationen och hårdvaran har uppstått.
2. Systemet kan bekräfta att det inte finns en nätverksanslutning och användaren bör utföra samma förförande igen.
3. Sker samma komplikation igen skall applikationen uppdateras och påbörja processen för grundflötet återigen.
4. Skulle de alternativen ovan inte fungera ombeds användaren att kontakta systemhanterare för vidare assistans.

5.1.2 Alternativt flöde 2: Applikationen svarar inte.

Vid en eventuell krasch av applikationen skall användaren påbörja omstartning av applikationen. Användaren kommer behöva utföra processen för grundflödet.

5.1.3 Alternativt flöde 3: Meddelandet inte existerar i databasen

1. Systemet kontrollerar om meddelandet existerar i databasen.
2. Systemet indikerar på att meddelandet inte existera i databasen.
3. Det aktuella flödet återvänder till grundflödetillstånd.

6 Eftervillkor

Visar det sig att användaren utför användningsfallet korrekt och enligt beskrivningen ovan så skall enheten visa upp det meddelande som skrivits av användare eller tagit bort meddelandet från skärmen.

7 Förbättringsmöjligheter

I detta avsnitt presenteras utvecklingar som kan tillkomma i systemet i framtiden, något som inte lyckats utvecklats sedan innan på grund av kostnaderna.

- Applikationen skall bli kompatibel för mjukvarusystem som Android samt IOS.
- Förbättring av prestandan genom att förbättra responstiden från att skriver/raderar meddelandet i enheten till att det lokalt i databasen.
- Förbättring av applikationens gränssnittsutformning för att skapa goda förutsättningar för ett gott samspel mellan applikationen och användaren.
- Databassäkerhet för att skydda mot databasintrång.
- Applikationen skall kunna hantera så att flera användare kan ha tillgång till att skriva/radera meddelande på skärmen.

Grupp 10

Teknisk dokumentation

Beskrivning av applikationens mjukvara

2017-05-23
KTH Electrum, 16440 Kista, Sverige
Kaisoum@kth.se

Abstract

I denna dokumentation sker en beskrivning av programkod för generera samt designa användargränssnittet för denna applikation.

Version History

Date	Version	Author	Description
19/05/2017	1.0	Adam Kaisoum	Skapad
20/05/2017	1.2	Adam Kaisoum	Utkast
22/05/2017	1.3	Adam Kaisoum	Uppdatering
23/05/2017	2.0	Adam Kaisoum	Färdig

innehållsförteckning

1	Introduktion.....	3
1.1	Dokumentets syfte	3
1.2	Bakgrundsinformation	3
1.3	Ansvarsområde	3
1.4	Dokumentöversikt	
2	Funktionalitetskrav	4
2.1	Funktionella krav	
2.2	Kunskapskrav	
2.3	Icke-funktionella	
3	Implementationsvillkor	4
4	Programkodens funktioner.....	5
4.1	HTML	
4.2	PHP	
4.3	Bootstramp	
4.4	Datorsäkerhet	
5	Förbättringsmöjligheter - Vidare Utveckling.....	7

1 Introduktion

1.1 Dokumentets syfte

Syftet med detta dokument att ge en tydlig förklaring samt få en överblick över den tekniska delen av produkten, det vill säga mjukvaran. Samt att idéer kommer framföras, beskrivningar av produkten samt teknologiska lösningar, eventuella beteende som kan uppstå samt processer med kunder som skall nyttja produkten i sin egen verksamhet.

Målet med projekt har varit att utveckla ett system för en informationsdisplay. För detta projekt har systemet endast delats in i en del eftersom projektmedlemmarna hade endast kompetens inom mjukvara, det vill säga att utveckling av hårdvara är ingen som åstadkommits. Mjukvarudelen innehåller all utvecklad mjukvara, det vill säga utveckling av användargränssnitt för applikationen, samt rad andra funktionaliteter tillhörande systemet samt kommunikationen mellan applikationen och databasen.

1.2 Bakgrundsinformation

Utvecklandet av programkod samt skapandet av hemsidan har skett på Atom samt IntelliJ. Dessa text-redigeringsprogrammen är lättanterliga samt moderna som tillämpas för utveckling av funktion för system samt för webbdesign. Dessa program har varit ett viktigt redskap för behandling av programmeringsspråken HTML, PHP, JavaScript samt Bootstrap. Dessa programmeringsspråken har använts för att skapa funktionalitet för systemet samt webbdesign.

1.3 Ansvarsområde

Dokumentets innehåll består utav förklaringar samt beskrivningar om den aktuella mjukvaran som konstruerats och utvecklats för detta projekt. Här presenteras den detaljerade informationen för användargränssnittet som användaren kommer i kontakt med vid användning av applikationen samt andra funktionaliteter, till exempel:

- Hemsidans funktionaliteter som erbjuds.
- Konstruktionen av hemsidan.
- Programmeringsspråk som kommit till användning.
- Kommunikationen mellan applikationen och databasen.
- Presentation av rad olika tester för olika komponenter för att säkerställa att produktens funktioner fungerar korrekt.

Denna dokumentation är riktad till produktägaren och relevanta individer inom detta område. Eftersom en teknisk produkt har levererats kan en del tekniska begrepp uppkomma och för en individ som inte besitter kompetens inom dator teknik kan detta leda till oklarheter samt förvirring.

Projektgruppen har valt att använda sig av tre olika implementationsområden för designen utav användarvy som kan ses som fundamentet av själva hemsidan. Dessa tre olika implementationsområdena har skapat av tre programmeringsspråk. Detta kommer senare avsnitt presenteras mer ingående.

1.4 Dokumentöversikt

Följande avsnitt innehålls i dokumentet:

- **Funktionalitetskrav** – Beskrivning av de funktionalitetskrav som existerar i systemet.
- **Kunskapskrav** – Kompetens som läsaren behöver ha.
- **Implementationsvillkor** – Beskrivning av miljön som applikationen är avsedd för.
- **Programkodens funktionalitet** – Beskrivning över applikationens programkod.
- **Förbättringsmöjligheter** – Beskrivning över implementationer för förbättra och öka prestandan på produkten.

2 Funktionalitetskrav

2.1 Funktionella krav

Denna del av systemet är avsedd för att få individen som användaren att kunna navigera, nyttja samt få en förståelse för funktionaliteterna som existerar i applikationen. De funktionella kraven som existerar är följande:

- Användaren skall kunna skriva samt radera meddelande som sedan skall kunna visas på en informationsdisplay för förbipasserande (För detta används en databas där användarens meddelande lagras eller raderas).
- Individet som använder sig av applikationen skall kunna navigera sig igenom applikationen utan att det uppstår några komplikationer på applikationen. Samt att användaren skall ha ett fullt medvetande över vad han/hon utför.
- Programkoden för webbdesignen skall vara strukturerad samt att hemsidan skall innehålla tydliga navigationsfunktioner som gör det lätt för navigering.

2.2 Kunskapskrav

De förkunskap som måste erhållas är grundläggande kompetens inom applikation för internet, där involveras programmeringsspråken PHP, HTML samt JavaScript.

2.3 Icke-funktionella krav

Användaren skall alltid ha tillgång till applikation och dess funktionaliteter för att kunna komma åt de icke-funktionella kraven som existerar för att kunna nyttja dem. Programkoden skall naturligtvis vara objektorienterad som leder till att strukturen för systemet blir lätt korrigerbart vid eventuella förändringar eller om nyttkomna funktioner skall implementeras. Ett stort säkerhetstänk har gett upphov till att applikationen skall ha en del säkerhetsåtgärder mot inkräktare. Samt att få tillgång av databasinnehåll från databasen skall vara så effektiv som möjligt när det kommer till tidsaspekten.

3 Implementationsvillkor

Den programkod som genererats under projektets gång kommer endast agera under mjukvarumiljö. Programkoden har genererats och kompilerats i två olika textredigeringsprogram Atom samt IntelliJ samt att programmeringsspråken som används är HTML, PHP, Bootstramp samt JavaScript. Under själva projekts gång har stor uppmärksamhet lagts på strukturen samt optimering av framtida utvecklare tagits. Samt har åtgärder tagit gällande programkoden för att öka lättläsigheten för eventuella justeringar som kan göras av andra utvecklare i framtiden. Avslutningsvis har utvecklarna strävat efter att öka prestandan för mjukvaran så pass mycket som möjligt genom att utföra en så effektiv kod som möjligt.

4 Programkodens funktioner

Programkoden har delats in i tre delområden för att skapa användargränssnittet för webapplikationen. För varje delområde utgörs av ett utvecklingsspråk, varje specifikt utvecklingsspråk har speciella funktioner för skapa väsentliga funktionalitet för systemet. En kort beskrivning om varje utvecklingsspråk samt till vilka användningsområden de nyttjats till.

4.1 HTML

HTML står för *Hyper text markup language*. HTML är ett standard markeringsspråk för att skapa webbhemsidor samt webbapplikationer. Med HTML-kod kan man skapa olika typer av rader och paragrafer med text som tillsammans sätts ihop till divisioner. Samt med CSS och JavaScript skapar man själva designen med för att större välkomnande för användaren. Bilden till vänster ett exempel på hur HTML kod kan se ut och vad resultatet blir av det. HTML är

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
```

This is a Heading

This is a paragraph.

i sig inte ett programmeringsspråk utan faller in i kategorin markeringsspråk.

4.2 Bootstrap

Bootstrap är ett webbramverk som används för redigering av design för en webbapplikation. Ändring av positionering av olika bilder, ändring av typsnitt, textstorlek samt färg. Det finns en rad olika funktioner som kan nyttjas med bootstrap. Ramverket är väl anpassad för utveckla avancerade hemsidor för alla typer av färgnivåer samt skärmupplösningar.

4.3 PHP, MySQL, Databas

PHP står för hypertext preprocessor och är ett server inriktad språk som används för att kommunicera till exempel mellan en webbapplikation med självverkande innehåll och en databas.

Denna programkod är utdrag från systemets funktioner som möjliggör hämtning av ett specifikt meddelande. För möjliggöra hämtningen av ett meddelande måste en förbindelse skapas med en specifik databas, detta för att göra det möjligt för förbipasserande att kunna se meddelandet på displayen. Detta görs enligt figuren nedan.

```
1  <?php
2      include 'ConnectToDatabase.php';
3      try {
4          $sql = "SELECT * FROM messages";
5          $q = $conn->query($sql);
6          $emparray = array();
7          while($row = $q->fetch(PDO::FETCH_ASSOC)){
8              $emparray[] = $row;
9          }
10         echo json_encode($emparray);
11         header("content-type:application/json");
12         exit();
13     } catch(PDOException $e) {
14         echo $e->getMessage();
15     }
16     ?>
17
18
19 <?php
20     include 'connectToDB.php';
21     try {
22         $sql = "SELECT * FROM messages ORDER BY messageId DESC LIMIT 1";
23         $q = $conn->query($sql);
24         $message = $q->fetchColumn(1);
25         echo $message;
26     } catch(PDOException $e) {
27         echo $e->getMessage();
28     }
29     ?>
```

MySQL-databasen är en plats där alla meddelande hämtas ifrån samt att hämtade mätvärden kan lokaliseras i databasen. Vid hämtning av specifika meddelande används PHP för att kunna knyta åt sig rätt meddelande.

4.4 Datasäkerhet

All typ av datasäkerhet är väsentlig för en webapplikation för att skydda själva applikationen samt användare då känslig information kan vara lagrat i databasen. Detta gynnar även resultatet av ett projekt i sig på grund av att pålitlighet skapas mellan kund och företag då man lägger stort fokus på kundens säkerhet gällande känslig information, detta leder oftast till en kedjeffekt som bidrar till ökning av kunder.

Attacker som SQL-injektioner eller när inkräktare lyckas komma åt känslig information som till exempel en användares lösenord eller bankuppgifter leder garanterat till ett missnöje bland kunder vilket kan resultera till kundkretsen minskar markant.

För att undvika sådana förseelser bör man implementerar säkerhetsfunktioner i systemet, detta kan ske på ett antal olika sätt men i detta system hade arbetet börjat med att implementera dessa funktioner samt att ökning av säkerhet är under utveckling.

Att förhindra SQL-injektion kan ske på flera olika sätt, men ett lätt sätt att undvika en sådan attack är att använda sig av ”stored procedures” vilket är lagrade procedurer i systemet. Man skapar SQL-satser som lagras direkt in i databasen samt användning av parametrar för att lagra och uppdatera data.

För att förhindra inkräktare att komma åt känslig information bör lösenordkryptering implementeras in i systemet. För att kryptera lösenord så används funktionen MD5- Message digest algorith 5 som är en krypteringsfunktion i programmeringsspråket PHP. Vid användning av denna funktion omvandlas lösenordet i databasen till ett 128-bitars hash värde. Syftet med detta är att lösenordet blir krypterad och gör det betydligt mycket svårare för ovälkommna användare att stjäla eller komma åt det rätta lösenordet.

5 Förbättringsmöjligheter - Vidare utveckling

Implementationer samt förbättringar av funktioner för applikationer kan alltid vidare utvecklats men kostnadsaspekten är något som måste räknas med i kalkylen. Till detta system har utvecklarna inte haft ett så stort säkerhetstänk utan fokus har lagt på att leverera en produkt med det funktionerna som efterfrågats. Förbättringspunkt som hade prioriteras med all säkerhet är att få struktur på programkoden enligt MVC-modellen vilket bidrar till vidare utveckling av andra utvecklare i framtiden samt att systemet blir så objektorienterad som möjligt. Några förbättringspunkter som hade utvecklats:

- Applikationen skall bli kompatibel för mjukvarusystem som Android samt IOS.
- Förbättring av prestandan genom att förbättra responstiden från att skriver/raderar meddelandet i enheten till att det lokalt i databasen.
- Förbättring av applikationens gränssnittsutformning för att skapa goda förutsättningar för ett gott samspel mellan applikationen och användaren.
- Databassäkerhet för att skydda mot databasintrång.
- Applikationen skall kunna hantera så att flera användare kan ha tillgång till att skriva/radera meddelande på skärmen.

En förbättringspunkt som verkligen var något projektgruppen ville utveckla var att göra produkten så miljövänlig som möjligt. Genom att göra produkten så energisnål så möjligt, detta hade kunna lösas genom att lägga mer fokus på att utveckla en display som har som funktion att vilolägesfunktion eller en sensor för uppfatta om det finns någon individ som är behov av informationen som finns på skärmen. Detta skulle ge upphov till försörjningen av batterier för att driva displayen minskas markant vid användning av trådlös display.

Grupp10

Grupp 10

Rollbeskrivning

Kim Askebris

2017-05-29

Innehållsförteckning

1	Introduktion.....	4
1.1	Inledning.....	4
1.2	Dokument Omfattning.....	4
1.3	Dokument Översikt	3
2	Resultatmål.....	4
3	Riktlinjer.....	5
4	Rollen	4
5	Vägledning	5
5.1	Lärdom	5
5.2	Tips.....	5

1 Introduktion

1.1 Inledning

Syftet med detta dokument är att förmedla gruppens uppgift, organisering och vägledande principer för en projektledare i ett mindre IT-projekt.

1.2 Dokument omfattning

Dokumentet behandlar följande:

- Hur arbetar man som utvecklingsansvarig
- Vad bör man tänka på som utvecklingsansvarig
- Vilka misstag kan man begå som utvecklingsansvarig

1.3 Översikt

Dokumentets innehåll

- Resultatmål
- Riktslinjer
- Rollen
- Lärdomar
- Tips

2 Resultatmål

Projektgruppen har fått i uppgift av kunden att utveckla ett system för att hantera en elektronisk display, systemet ska ha en funktion för att uppdatera information på displayen och det ska fjärrstyrt dvs oberoende av var kunden befinner sig. Gruppen ska även leverera en produkt i form av en kursrapport där de behandlar arbetets gång och lärdomar.

3 Riktlinjer

Vad man bör tänka på som utvecklingansvarig

- Filstruktur
- Versionshantering
- Programmeringsprinciper
- Arbeta mot deadlines
- God kontakt med kravansvarig

4 Rollen

Som utvecklingansvarig arbetar du med att utveckla mjukvarusystemet. Du har det övergripande ansvaret för att systemet uppfyller de krav som ställs från kravansvarig. En god kontakt med kravansvarig är viktig så det inte blir några missförstånd kring kraven. Det är också viktigt att förstå systemarkitekturen för att kunna transformera det till en välfungerande filstruktur. Att förstå goda programmeringsprinciper är ytters viktigt för att få ett optimerat mjukvarusystem.

Som utvecklingsansvarig har du ett ansvar för att sprint- och projektmålen nås och att arbeta mot deadlines är en central del för rollen. Det är därför viktigt med en noggrann planering och plan för arbetet.

5 Vägledning

5.1 Lärdom

Rollen som utvecklingsansvarig har varit väldigt givande och lärrik. Man har fått en bred kompetens inom hur man bygger effektiva mjukvarusystem, arbeta mot deadlines samt arbetar tillsammans i en projektgrupp. Som utvecklingsansvarig är det betydelsefullt att ha god kommunikation med andra gruppmedlemmar och att man är tydlig med vad som förväntas. Arbeta redan från start med versionshantering och använd det konsistent. Det ger en bra överblick över projektets utveckling.

5.2 *Tips*

Tips till dig som är blivande utvecklingsansvarig är att få med alla i gruppen i arbetet kring koden. Det är viktigt att alla har en grundlig förståelse om vad koden gör. Ha heller inte för bråttom att sätta igång att koda, det är viktigt att ni först har en bra systemarkitektur och filarkitektur, detta kommer underlätta kodandet när man sätter igång.

Ta inte heller på dig för stort ansvar, oftast kan det vara så att den som kodat mest tar på sig utvecklingsrollen vilket leder till ett aldeles för stort ansvar. Låt dina gruppmedlemmar arbeta på uppgifter utan att peta i allting och sist men inte minst, lita på dina gruppmedlemmar, människor växer med ansvar.

Grupp10

Grupp 10

Systemdokumentation

2017-05-29

Innehållsförteckning

1	Introduktion.....	3
1.1	Dokumentets Syfte	3
1.2	Dokument Omfattning.....	3
1.3	Dokument Översikt	3
2	Kravfunktioner	3
3	Implementationsbegränsningar	3
4	Tillhandahållna interfaces	5
5	Interfacekrav	4
6	Intern Struktur	4

1 Introduktion

1.1 Dokumentets Syfte

Detta dokument visar hur projektet är genomfört med frameworkt Java Spring. En grundläggande förklaring och översikt tillhandahålls för läsaren för att ge en bra bild av hur Java Spring används i webbapplikationer.

Dokumentet visar även hur man enkelt kan använda JavaScript för att koda en modern webbsida.

1.2 Dokumentets Omfattning

Dokumentet omfattar hur man kan använda Java Spring i webbapplikationer. En analys av Spring MVC och hur filstrukturen är uppdelad. Hur olika komponenter som server, client, databas hänger ihop.

Dokumentet visar även hur JavaScript används för att få en användarvänlig och responsiv sida.

2 Kravfunktioner

Denna sektion kommer visa upp vad för funktioner som är nödvändiga för de olika delarna, vad som krävs för att funktionerna skall anses klara och tillföra systemet det som behövs.

Användaren ska kunna skriva in ett meddelande i ett webbformulär. Meddelande ska sedan tas emot på servern och sparas i en databas. Meddelandet ska sedan pollas var tredje sekund med ett AJAX-script som visar det på en skärm.

För att genomföra dessa krav krävs

- Ett användargränssnitt där användaren kan skriva in ett meddelanden
- Backendkod som kan ta emot och hantera meddelanden
- Ett API för databasanrop
- Javascripts för AJAX-anrop

3 Implementationsbegränsningar

För att köra Java Spring med Intelij krävs Intelij Ultimate Edition. Den går att ladda ner gratis med ett KTH-konto. För att använda Java Spring krävs Java 7 eller senare.

4 Tillhandahållna Interfaces

I Java Spring använder man MVC ramverket. Du har ett publikt användargränssnitt med vyer samt ett API i dina Controllers.

5 Interfacebehov

Java Spring:

- IDE(Intelij, Netbeans)
- Apache Tomcat

När man ska deploya(mergea ihop alla filer och lägga det på en server) en spring applikation sammanslagas alla filer till en .WAR fil. För att en servern ska kunna hantera request krävs det en Apache Tomcat. Det är en Apache Server som innehåller en Tomcat.

Tomcat är en behållare som tar emot request från apachen och gör om requesten så att Java Spring kan förstå dem. För en översiktlig bild se figure 3.

6 Intern Struktur

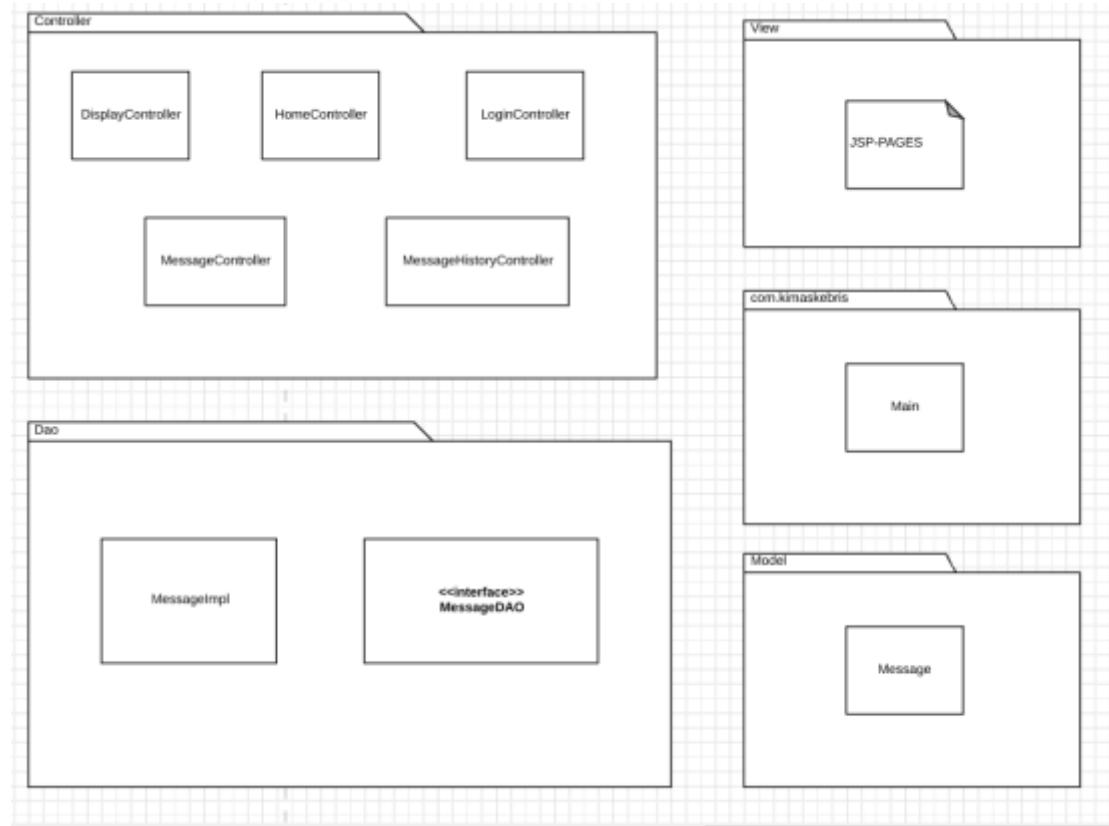


Figure 1, Classdiagram

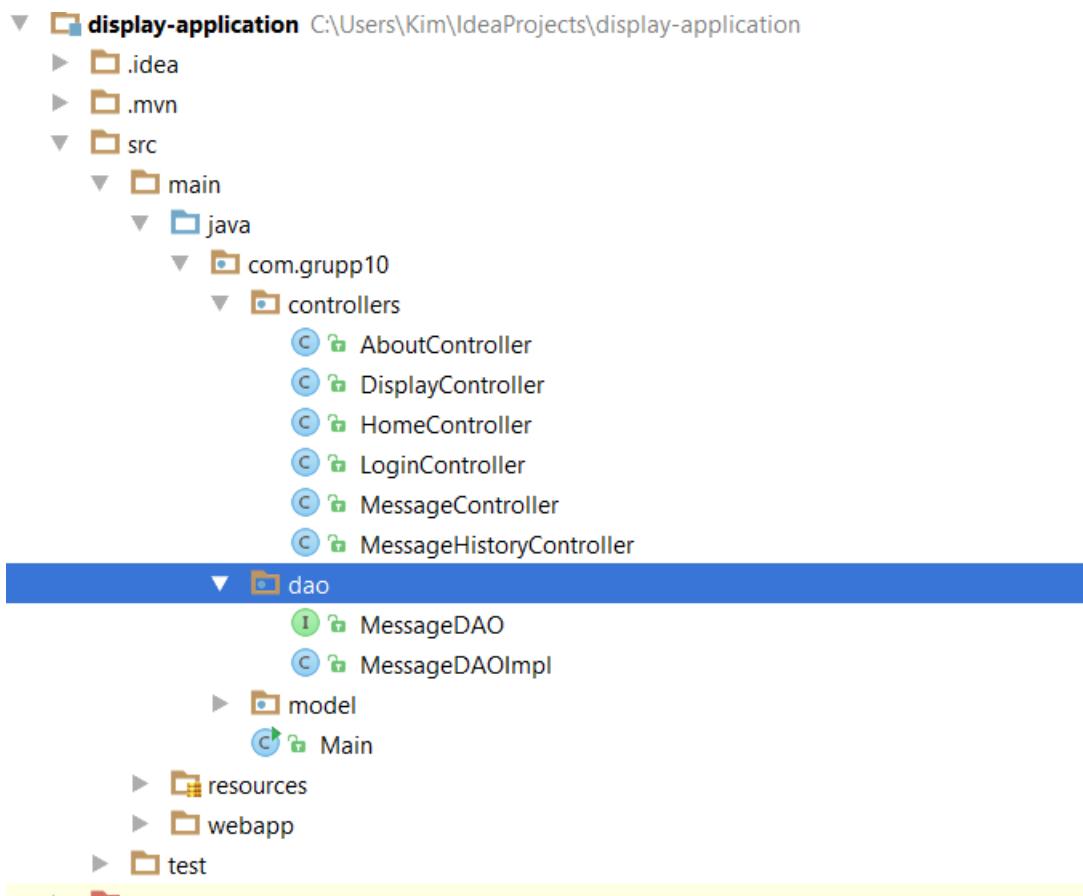


Figure 2, filstruktur

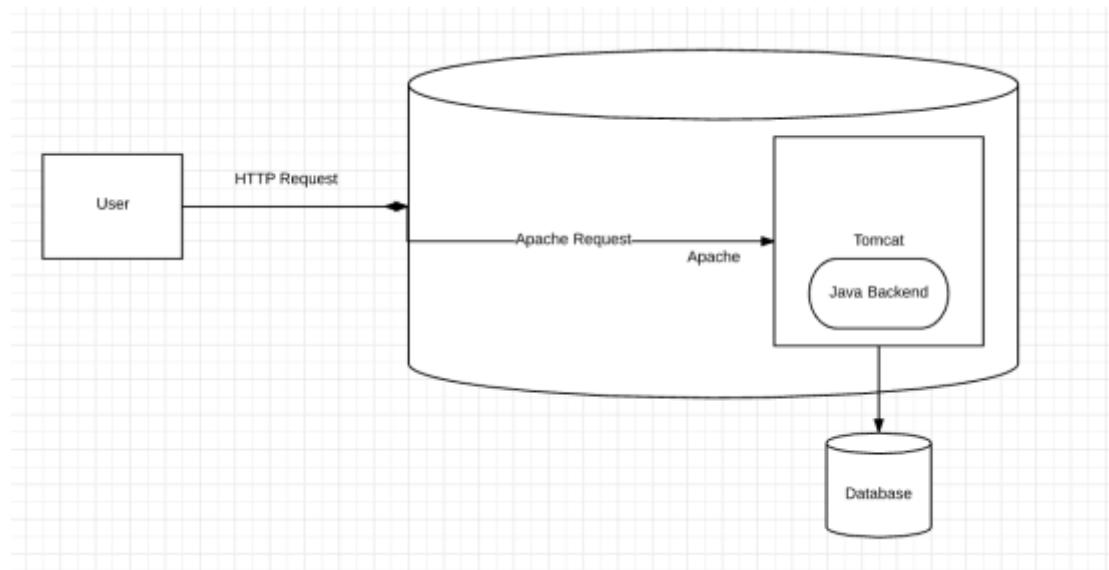


Figure 3, Apache tomcat översikt

I systemet finns sex stycken scripts för att fylla systemets olika krav. Alla script använder sig utav AJAX vilket ger en responsiv webbsida. Scriptet i figure 4 tar emot ett meddelande från clienten och gör ett AJAX-anrop till servern. Meddelandet valideras så att det inte är ett tomt meddelande som skickas. Efter meddelandet är skickat töms formuläret.

```
$ (document).ready(function () {
    $('#text-message-post-button').click(function () {
        setMessage();
    });
});

function setMessage() {
    if(validateForm()){
        makeAjaxRequest();
    } else{
        return false;
    }
}

function validateForm() {
    var x = document.getElementById("message-input").value;
    if (x == "") {
        return false;
    }
    return true;
}

function makeAjaxRequest() {
    $.post('setMessage',
    {
        textMessage : document.getElementById('message-input').value
    },
    function(data){
        resetFormInput();
        $('.posted-message').empty().append("<br><br>Message: " + data.valueOf());
    });
}

function resetFormInput(){
    document.getElementById('message-form').reset();
}
```

Figure 4, Sendmeddelande script

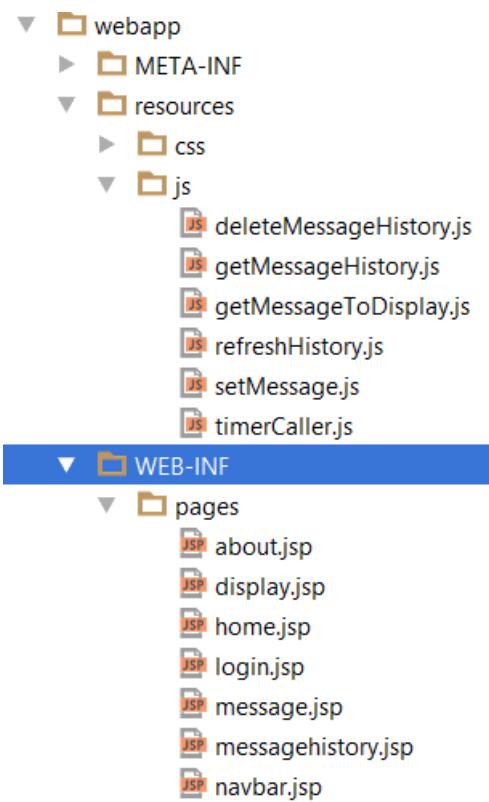


Figure 5, Översikt över resources för applikationen

Testplan

Hur testas produkten

Versions historik

Datum	Version	Författare	Beskrivning
24/04/2017	1.1	Maher Jabbar	Första utkast
02/05/2017	1.2	Maher Jabbar	Byte till svenska
10/05/2017	1.3	Maher Jabbar	Skriva och placera nya rubriker
20/05/2017	1.4	Maher Jabbar	Fortsatt uppdatering av mallen

Table of Contents

1.	Bakgrund	3
2.	Syfte	3
3.	Mål	3
4.	Testning	3
4.1	Testmetoder	3
4.2	Testplan	3
4.3	Enhetstest	3
4.4	Komponent test	3
4.5	System test	4
4.6	Test Driven utveckling	4
4.7	Tester och testcases	4
4.7.1	Testning på tavlan	4
4.7.2	Installation.....	4
4.7.3	Ansluta till hemsidan (Remote)	4
4.7.4	Skriva meddelande på hemsidan.....	5
4.7.5	Testa databasen med Selenium WebDrive	5
4.7.6	Kontrollera att meddelandet syns både på hemsidan och displayen.....	5
5.	Resultat	6
6.	Risker	6
	Appendix A - References	6

1. Bakgrund

Som del av kursen projekt och projektmetoder så ska vi lära oss vad gäller testning när man arbetet i ett projekt och speciellt ett IT projekt. Testning är en viktig del av utvecklingen för produkten då det är steget som ser till att produkten är redo för användarmarknaden

2. Syfte

Syftet med dokumentet att beskriva vilka tester som ska utföras och en enkel beskrivning för hur de ska testas. Detta ska hjälpa projektmedlemarna att kunna planera sina tester samt få en överblick på alla tester i projekt

3. Mål

Detta dokument ska verifiera att den utvecklade hård och mjukvaran faktiskt fungerar och uppfylla kundens krav och en komplett produkt levereras utan buggar. Därutöver ska dokumentet förklara hur detta har genomförts

4. Testning

4.1 Testmetoder

Testet ska ge en bild av hur testning har organiserat och genomförts under projektets gång. Alla enheter kommer och testas och utvecklas så att produkten kan leveras färdigt.

4.2 Testplan

Våra tester kommer att bestå av enhetstester för att se till koden som skrivs garanterar funktionalitet. Samt enklare tester där vi testar att raspen och webbsidan och ser om allt fungerar.

4.3 Enhetstest

Vi testar olika enheter i systemet, det kan exempelvis vara en metod eller en knapp eller det kan vara nätverk uttaget till raspen eller det kan vara själva displayen som visar meddelande. Enhetstesten ska kunna visa att dessa fungerar och gör som de skall.

4.4 Komponent test

Syftet med komponent test är att fastställa att funktionaliteten av systemet bibehålls efter att enheterna samverkar med varandra. När alla enhetstest för en komponent är godkända räknas även komponenten som godkänd.

4.5 System test

Systemtest täcker hela systemet, i detta fall gäller det alltså att testa alla små system båda med rätt och fel input. Detta är det slutgiltiga test av systemet under testningars som ska fungera och uppfylla kundens krav.

4.6 Test Driven utveckling

TDD är ett testredskap. Innan man implementerar funktioner så ska man skapa test för dem . Detta är en fördel då blir det lättare att upptäcka fel i designen eller arkitekturen innan man har börjat utveckla funktioner för enheterna.

4.7 Tester och testcases

4.7.1 Testning på tavlan

Det är en viktig del av testning för att underlätta strukturen av tester. På tavlan finns det den hel del avsedd för systemarkitekturen där alla olika delar av systemet bör ingå. Man använder sig av test case kort som man skriver på lämpligt sätt. man skriver kortfattat vad testet gäller och sen kan man kryssa på delarna som har blivit testat och godkända. Själva projektet har utförts med projektmodellen Scrum.

Testningen kan integreras med SCRUM. Båda design och arkitektur skissas upp på tavlan då är det lättare och ta fram enhetstest för varje delkomponent.

4.7.2 Installation

Installation av Server och Databas för raspberry. Målet med detta är starta åtkomst av hemsidan.

Testidentiferare	1
Testnamn	Installation
Testets avsikt	Se till att det finns en server och databas på raspberry
Status	Gödkänt - 2015-04-20

4.7.3 Ansluta till hemsidan (Remote)

Att testa och ansluta till hemsidan via en enhet som är "ute" på internet är ett grundkrav för att kunna lösa kundens kravspecifikation. För att kunna ansluta till hemsidan så krävs det att en enhet är uppkopplad mot internet och kan surfa in på grp10.it.kth.se

Testidentiferare	2
Testnamn	Ansluta till hemsidan(Remote)
Testets avsikt	Se till att användaren kan komma åt hemsidan och se innehållet
Status	Gödkänt - 2015-04-25

4.7.4 Skriva meddelande på hemsidan

Att skriva till hemsidan innebär i första steget att någonting kan ta emot datan som skrivs in och sen skriva ut den rakt på hemsidan. Nästa del kräver att vi har en databas uppkopplad för att datan skall kunna skickas till denna. Det första testet är godkänt när datan som skrivs in i fältet "skriva meddelande" skrivs ut av applikationen på hemsidan. det andra testet är godkänt när datan hamnar i databasens "Messages" tabell och sparar där.

Testidentiferare	3
Testnamn	Skriva/Uppdatera meddelande
Testets avsikt	Se till att användaren kan skriva meddelande och kunna uppdatera meddelandet
Status	Gödkänt - 2015-05-02

4.7.5 Testa databasen med Selenium WebDrive

Databas testning är verifieringen av hämtade värden från databasen med en webb eller datorprogram. Data som visas på användargränssnittet bör matcha enligt de uppgifter som lagras i databasen.

Testidentiferare	4
Testnamn	Databas testing
Testets avsikt	Se till att information som vi lägger finns också i databasen
Status	Gödkänt - 2015-05-03

4.7.6 Kontrollera att meddelandet syns både på hemsidan och displayen

Testet utförs av att man skriver en text på webbsidan från serveren i raspen och tittar ifall texten syns i displayen. skickar sedan ny text för att skriva över den gamla och tittar ifall det fungerar även att skriva över. Detta görs via att gå till länken <http://grp10.it.kth.se/display.php> Här ser vi vad som visas på displayen För att se vad som står på hemsidan så går vi till <http://grp10.it.kth.se/message-history.php> NOTE: Dessa två skall vara samma sak så länge man inte har tömt historiken i message-history.

Testidentiferare	5
Testnamn	Displayen
Testets avsikt	Se till att användare kan se texten på displayen
Status	Gödkänt - 2015-05-11

5. Resultat

Alla enhet tester som testades i verkligheten och de fungerar som det önskas. Det har lett till en produkt som möter kravspecifikationerna.

6. Risker

Det mest sannolika riskerna som kan förekomma för testerna är buggar och fel som leder till många timmar bortkastade på felsökning på produkten samt brist kunskap inom området då detta är första gången vi utför en test. Utvecklaren av testen ska vara extra noggrann över hur testet utförs för att motverka dessa riskerna.

Appendix A - References

- [1] Boken “Software Engineering- Sommerville” Chapter 8 “Software testing”

Tekniskt Dokument

Versions historik

Datum	Version	Författare	Beskrivning
24/04/2017	1.1	Maher Jabbar	Första utkast
10/05/2017	1.2	Maher Jabbar	Lagt till beskrivning av Database
20/05/2017	1.3	Maher Jabbar	Uppdatering av dokumentet

Table of Contents

1. Introduktion.....	3
2. Ansvarsområde.....	3
3. Förkunskaper	3
4. operativsystem	3
5. Programvara.....	3
6. Databas.....	4

1. *Introduktion*

Uppdragsgivaren vill hitta en lösning att på ett smidigt sätt kunna informera kollegor samt besökare om var personen befinner sig under arbetsdagen. Detta genom att visa ett meddelande på en elektronisk display som är uppsatt på kontoret. Uppdragsgivaren vill kunna ändra denna information även om den inte är på plats. Genom att använda sig av en Raspberry Pi, en elektronisk display och en webbapplikation skapa ett system som behandlar dessa frågor samt att utöka våra kunskaper inom "systemutveckling"

2. *Ansvarsområde*

Jag har ansvar för installera Serveren och implementera Databasen på raspen samt testa att allt fungerar utan bugar.

3. *Förkunskaper*

Kunskap för database har jag haft från kursen Datalagring och applikationer i internet

4. *operativsystem*

Raspian är linuxbaserat och det operativsystem som körs på den raspberry pi enhet som används i projektet är Raspbian. Raspian är utvecklat specifikt för Raspberry pi

5. *Programvara*

För programmering av database så användes PHPStorm som programvara och för att testa databasen så har jag använt Eclipse med Selenium webDriver som är ett testverktyg för att skapa automationstester för webbsidor och webb applikationer.

6. Databas

Vi hade behovet av en databas för att lagra meddelandet och kunna uppdatera meddelandet på displayen. För en databas skulle man kunna tänka sig att använda MySQL, Microsoft Access. Vi valde att använda MySQL som lagringsmedia för våra meddelanden som ska visa på display. Först behöver jag kommunikation mellan terminalen och raspen så använder jag SSH sen installerade Mysql på raspen. Databasen lagrar all meddelande som användarna lägger upp. Jag har använt PHP för att koppla vår applikation med databasen. På koden heter databasen displayapplication och Message är den delen där alla meddelanden sparas.

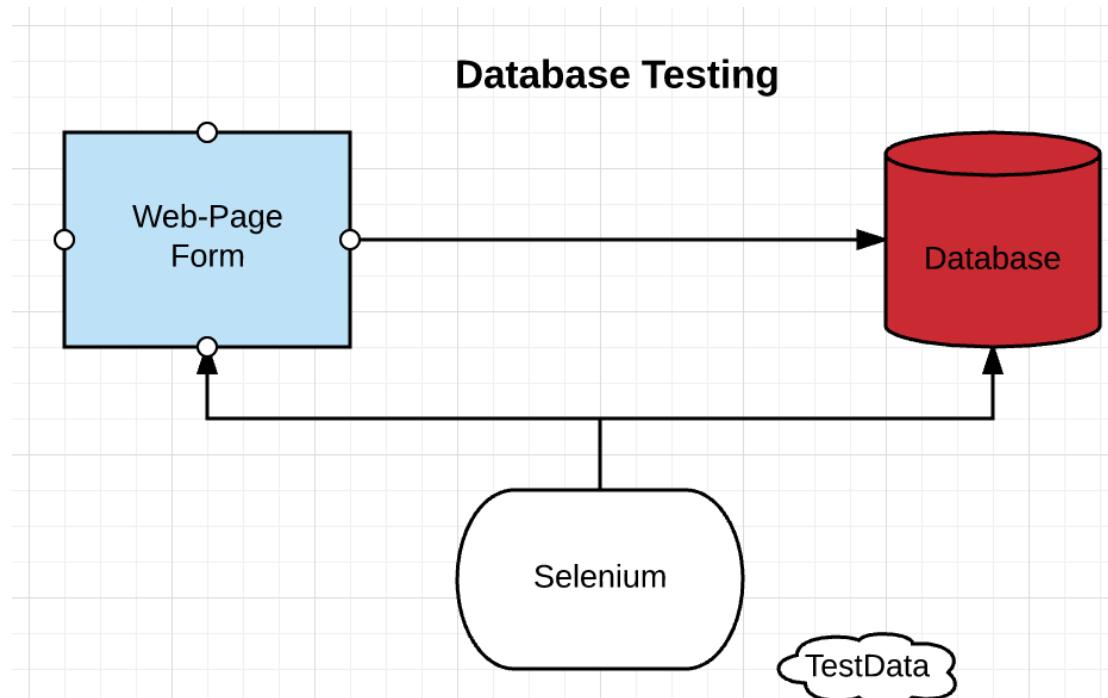
The screenshot shows two windows side-by-side. On the left is a code editor window titled 'connectToDB.php' containing PHP code for connecting to a MySQL database named 'displayapplication'. The code uses PDO to handle the connection and includes error handling for PDO exceptions. On the right is a screenshot of the phpMyAdmin interface, showing the 'Struktur' (Structure) tab for the 'displayapplication' database. It lists two tables: 'message' and 'messages'. The 'message' table has 7 rows and the 'messages' table has 3 rows. Both are InnoDB tables using the latin1_swedish_ci character set. Below the table list, there is a 'Skapa tabell' (Create Table) form with a 'Namn:' field containing 'message' and an 'Antal kolumner:' field set to 4.

```

1 <?php
2
3 /**
4  * Created by PhpStorm.
5  * User: maherjabbar
6  * Date: 2017-05-02
7  * Time: 18:44
8 */
9
10 $servername = "localhost";
11 $username = "root";
12 $password = "grupp10";
13 $charset = "utf8";
14 try {
15     $conn = new PDO( dsn: "mysql:host=$servername;dbname=displayapplication;charset=utf8", $username, $password);
16     // set the PDO error mode to exception
17     $conn->setAttribute( attribute: PDO::ATTR_ERRMODE, value: PDO::ERRMODE_EXCEPTION );
18 }
19
20 catch(PDOException $e)
21 {
22     echo "Connection failed: " . $e->getMessage();
23 }
24
25 ?>

```

Med hjälp av selenium webb driver, kommer jag att verifiera att detaljer som jag lägger den i formuläret återspeglar i databasen. Jag har använt Java för att göra det här testet.



Screenshot of an IDE showing Java code for database testing using TestNG and Selenium.

Project Structure:

- Grupp10
 - src
 - DatabaseTest
 - test2.java
 - HtmlTest
 - TestNG
 - testng.xml
 - testng.jar
 - jcommander.jar
 - bsh-2.0b4.jar
 - snakemake.jar
- Referenced Libraries
 - selenium-server-standalone-3.4.0.jar
 - client-combined-3.4.0-nodeps.jar
 - mysql-connector.jar

Code in test2.java:

```

1 package DatabaseTest;
2 import java.sql.Statement;import java.sql.SQLException;
3 import java.sql.Connection;import java.sql.DriverManager;import java.sql.ResultSet;
4 import org.testng.annotations.Test;
5
6 public class test2 {
7     @Test
8     public void Database() throws ClassNotFoundException, SQLException{
9
10         Class.forName("com.mysql.jdbc.Driver");
11         System.out.println("Läser in Driven");
12
13         Connection con = DriverManager.getConnection("jdbc:mysql://localhost/displayapplication", "root", "");
14         System.out.println("Ansluten till MySql DB");
15         System.out.println();
16
17         Statement smt = con.createStatement();
18         ResultSet rs = smt.executeQuery("select * from message");
19         while(rs.next()){
20             String messages = rs.getString("message");
21             System.out.println("Databasrekord är "+messages);
22         } System.out.println();
23     }
24 }
  
```

Console Output:

```

<terminated> test2 [TestNG] /Library/Java/JavaVirtualMachines/jdk1.8.0_131.jdk/Contents/Home/bin/java (28 May 2017, 20:18:17)
Läser in Driven
Ansluten till MySql DB

Databasrekord är grupp10
Databasrekord är kalle
Databasrekord är hej
Databasrekord är Hejsan
Databasrekord är Kalle
Databasrekord är semster
Databasrekord är På semster , återkommer o

PASSED: Database

=====
Default test
Tests run: 1, Failures: 0, Skips: 0
=====

=====
Default suite
Total tests run: 1, Failures: 0, Skips: 0
  
```

En test gjordes för att verifiera vår webbsida titel. Testet gjorde i Eclipse genom att använda TestNG som metod för testa HTML title.

The screenshot shows the Eclipse IDE interface with the following components visible:

- Package Explorer:** Shows the project structure with a package named "Grupp10" containing a source folder "src" with a file "Test1.java". Other entries include "JRE System Library [JavaSE-1.8]", "Referenced Libraries" (including "selenium-server-standalone-3.4.0.jar" and "client-combined-3.4.0-nodeps.jar"), and "TestNG".
- Test1.java:** The code for the test class:

```
1 package HtmlTest;
2
3 import org.junit.Assert;
4 import org.openqa.selenium.WebDriver;
5 import org.openqa.selenium.htmlunit.HtmlUnitDriver;
6 import org.testng.annotations.Test;
7
8 public class Test1 {
9
10    @Test
11    public void VerifyGrupp10Title(){
12
13        WebDriver driver = new HtmlUnitDriver();
14        driver.get("http://grp10.it.kth.se");
15        String title = driver.getTitle();
16        Assert.assertTrue(title.contains("Grupp10"));
17    }
18
19
20 }
21
```

- Console:** Displays the output of the test execution:

```
<terminated> Test1 [TestNG] /Library/Java/JavaVirtualMachines/jdk1.8.0_131.jdk/Contents/Home/bin/java (25 May 2017, 17:10:52)
PASSED: VerifyGrupp10Title
=====
Default test
Tests run: 1, Failures: 0, Skips: 0
=====

=====
Default suite
Total tests run: 1, Failures: 0, Skips: 0
=====
```

- Task List:** An empty list.
- Outline:** Shows the structure of the "Test1" class, including the "VerifyGrupp10Title" method.