

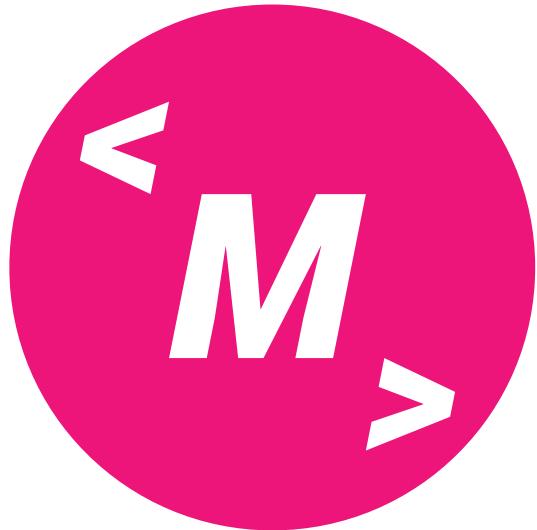
PPD-1

The Meta Family

Version 0.0

den 17 februari 2014

André Nyström
Axel Riese
Daniel Bergström
David Masko
Kerry Zhang
Marcus Hertz
Marcus Nordberg
Paulina Hensman
Simon Karlsson
William Perkola



*The
Meta
Family*

Sammanfattning

Innehåll

Versionshistorik

1 Problembeskrivning

1.1 Detaljerad problembeskrivning

Projektet framför oss består av att vi ska producera ett Software Development Kit (SDK) till vår projektgivare The Beta Family. Detta blir ett utvecklingsverktyg för applikationsutvecklare att direkt kunna få information från användaren om hur deras mobila applikation används och på så sätt få nödvändig information om hur de ska utveckla applikationer som är mer användarvänliga. Denna tjänst har redan fler än 9000 testare och har färdigställt tester för 2000 applikationer.

Tjänsten vi ska utveckla fungerar så att utvecklaren lägger upp sin applikation till The Beta Family och väljer den målgrupp som applikationen riktar sig mot, vilket operativsystem som används samt hur mycket personen som testar applikationen ska få i ersättning för att utföra detta test. Efter att testet är genomfört och testaren har rapporterat eventuella buggar och fel, betygsätts testaren av utvecklaren som betalar för testet.

Det erbjuds en SDK till iOS men inte till Android så projektet riktar sig till att utveckla en SDK för Android. Eftersom Androidmarknaden är mycket mer differentierad än vad den är på iOS-plattformen så uppstår många fler utmaningar att lösa för att slutföra projektet. The Beta Family har även förbestämda krav och önskemål på hur projektet ska utvecklas och därför ger detta projektet en del ramar att hålla sig inom.

Krav:

- Android enheten ska ej behöva vara “rooted” vilket betyder att man inte ska behöva ha tillgång till alla processer i enheten utan det ska fungera med begränsad tillgänglighet som kan förekomma för normala Android användare.
- Projektets SDK ska vara byggd i Native Java.
- iOS versionen är väldigt enkel och tar endast 2 minuter att applicera och därför bör Android versionen också vara enkel att applicera.
- Telefonens modell, skärmmens storlek samt position skall skickas tillsammans till servern med information om knapptryck och fingerrörelser.
- Användarens ansikte och röst skall spelas in och kan skickas in till servern som en separat video om det skulle vara nödvändigt.
- På applikationens Front-end så skall vyerna användaren besöker sparas och presenteras på en tidslinje.

Önskemål:

- Applikationens grafiska användargränssnitt borde vara så likt iOS-versionen som möjligt.
- Sensorer likt TestFairy som kan läggas till senare i framtiden.
- Visuell bevakning och jämförelse är framstående funktioner (Se TestFairy).
- Hur spelar vi in vyer på tredjeparts SDK:er t.ex. för Google Maps?

Då detta är de grundläggande krav och önskemål till SDK:n så ska detta utvecklas till Android och inte iOS vilket ger en del begränsningar som vi i projektgruppen inte kan komma förbi på grund av skillnaderna mellan dessa plattformar. Problem som bra bildfrekvens från kameran är problem som inte går att komma undan. Detta är sådant som vi i projektgruppen har diskuterat och kommit fram till att begränsningarna är något vi inte kan göra något åt och är helt enkelt något vi får arbeta runt och göra en så bra lösning som möjligt.

När produkten är färdigställd så kommer den att bestå utav en “Front-end” som är det som kommer visas för användaren samt att ta in information från testaren samt utav “Back-end” som är den del som kommer att ta hand om och lagra all information som samlats in efter

användning. "Front-end" skall byggas upp av HTML, CSS och JavaScript medan "Back-end" skall byggas upp i PHP med en databas i mySQL. (**catalog**)

1.2 Motivering

Alla gruppmedlemmar har ett stort intresse för teknisk utveckling och det är något vi alla vill arbeta med i framtiden. Det projekt vi valde reflekterar våra framtidiga ambitioner fast på en mildare nivå; att arbeta i grupp med att ta fram en teknisk lösning som skall levereras till ett företag (arbetsgivaren, The Beta Family i vårat fall) inom en given tidsram. Genom att prova på hur det är att jobba i grupp mot ett företag med ett projekt som faktiskt är tänkt att nå marknaden kommer vi att få erfarenheter som vi kommer kunna utnyttja i framtiden. Det är helt klart en fördel att känna till hur utvecklingsprocessen av en app går till när man som nyexaminerad civilingenjör når arbetslivet.

1.2.1 Tekniska områden

Projektet innehåller olika tekniska områden som tillsammans med kursens mål kommer att ge oss en inblick i hur projektarbete går till i arbetslivet.

- Androidprogrammering
 - Back-end i form av inspelning av skärm, kamera, ljud och touch samt insamling av telefondata så som modell, skärmstorlek m.m.
 - Front-end i form av ett intuitivt grafiskt användargränssnitt.
 - Hur denna programvara kan integreras i appar som ska testas.
- Datahantering; hur den insamlade datan från telefonen ska sammanställas, skickas och tas emot av en webbserver för att sedan presenteras i ett webbgränssnitt.

1.2.2 Framtiden inom teknisk utveckling

Den tekniska utvecklingen och i synnerhet mobiltelefoni har under de senaste åren gått framåt i rasande tempo. Fler och fler företag använder sig utav mobilappar för att nå ut till sina kunder med information, tjänster och produkter. Innan dessa mobilappar når marknaden måste de testas. Det som måste testas är tekniken bakom appen och användarvänligheten; hur den tänkta målgruppen reagerar när de använder appen. The Beta Family jobbar med att förmedla testare till apputvecklarna vilket vi tror starkt på eftersom appen kan nå en bredare grupp av testare från olika målgrupper och i mycket större skala än om utvecklarna testa appen själva.

1.3 Målsättning

Med en växande marknad för mobila enheter och med mjukvara för dessa ökar relevansen för att kunna programmera denna mjukvara för sådana enheter. Detta är inte bara nödvändigt för de som är intresserade att etablera sig som mjukvarukonstruktörer utan gynnar även dem som arbetar inom projekt som innehåller mobilapplikationer till någon del. En grafisk designer måste exempelvis vara väl medveten om begränsningarna för alla de operativsystem för smarta telefoner som produkten skall stödja när denne utformar de illustrationer som sedan skall programmeras. Detta betyder att alla medlemmarna kommer att kunna dra karriärsmässig nytta genom detta projekt oavsett vilken väg inom branschen som väljs eftersom risken är stor att nästintill alla någon gång kommer att arbeta med något relaterat till applikationer för smarta telefoner.

Eftersom projektets omfattning tillsammans med medlemmarnas antal gör det omöjligt att tillämpa en strategi där alla tar ansvar projektets helhet görs detta till ett stort samarbete. Det finns väldigt spridd erfarenhet av programmeringsprojekt i större grupper samt erfarenhet

av storleken på projektet. Detta gör att projektet kommer att vara en god möjlighet för alla medlemmarna att få mer eller första erfarenhet av större projekt inom mjukvarukonstruktion. Detta leder dessutom till att design av systemarkitektur blir centralt vilket ytterligare är en färdighet som kommer att slipas genom detta projekt.

Genom utformningen på kursen kommer även lärdom om en professionell arbetsprocess inom digital produktframtagning att läras. Detta innefattar många delar såsom rapportskrivningen, men även att från en specifikation hitta användarkrav för produkten och sedan specificera produkten i detalj och sedan implementera denna.

Slutligen fås det inte glömmas att en gemensam uppfattning inom gruppen är att projektet ska vara en rolig upplevelse tillsammans som en driven och ambitiös grupp. Målet är även att avsluta projektet tillsammans som en bättre grupp och känna att vi har utvecklats genom något vi fann roligt att göra tillsammans.

1.4 Färdigheter och förkunskaper

Vårt projekts största och viktigaste delar är i stort sett androidprogrammering. Eftersom det inte finns någon erfaren expertis inom området i gruppen är androidprogrammering, till den nivå som krävs, kunskap som saknas. En stor del av projektet är därmed att många medlemmar i gruppen måste lära sig detta. Fördelaktigt är att språket för android är i java och xml, och gällande java har alla i gruppen stor erfarenhet av språket. För android erbjuds väldigt mycket officiell gratis webbaserad information om allt från steg för steg-pedagogik till specificerad detaljnivå för enskilda implementationer och funktionaliteter. Dessutom finns det mångfallet av forum och dylikt där lösningar för specifika problem diskuteras och presenteras av privatpersoner. Allt detta är en fantastisk stor mängd av material att studera för att nå den kunskap som behövs för att genomföra projektet.

Utöver de tekniska färdigheterna finns även sociala erfarenheter i form av ledarskap, större programmeringsprojekt, grafiska gränssnitt genom Swing, och journalistik. Ledarskapsfarenheterna är mycket relevanta i möjliggörandet av effektivitet och organisation i gruppen. Erfarenheterna av stora programmeringsprojekt lämpar sig väl när systemarkitektur och API ska designas samt att dela upp programeringen och samordna denna. Erfarenhet med grafiska gränssnitt som skrivits i java är en tillgång för att kunna implementera eventuella gränssnitt för projekten. Journalistiken kommer väl till pass i dokumenteringen av projektet samt i rapportskrivningarna.

Eftersom det är en möjlighet att projektet även kommer att omfatta serverkommunikation är det relevant att det finns erfarenhet inom PHP och SQL i gruppen. Övriga kunskaper som inte är direkt förknippade med projektet är HTML, CSS, Ruby och javascript.

2 Bakgrund

2.1 Kommersiell bakgrund

På grund av begränsningar i operativsystemet är det relativt få som försökt sig på att lansera en skärminspelare till Android. Det är först i den senaste versionen av Android som möjligheten att spela in vad som händer på skärmen finns och då endast i utvecklingssyfte (**kitkat**). Det krävs att mobilen är kopplad till en dator via USB och detta är främst tänkt att användas för att göra instruktionsfilmer och dylikt.

2.1.1 TestFairy

TestFairy är ett företag med en affärsidé som påminner om den The Beta Family har. Målet är att underlätta betatestning av Androidapplikationer. Genom att använda sig av deras lösning kan

man få en video som visar hur användaren navigerar i appen, grafer över diverse intressanta data (Exempelvis minnes- och CPU-prestanda), samt en mängd metadata.

Problemet med att Android inte har stöd för skärminspelning löser TestFairy genom att ta en skärmdump en gång i sekunden. Detta resulterar dock i en video som snarare påminner om ett bildspel än en rullande video. Dessutom förlorar man information om hur användaren reagerar på ljud från appen. Den visar dock det väsentliga användaren håller på med och ger i det stora hela en god överblick över hur appen används.

Jämfört med The Beta Family verkar TestFairy vilja fokusera mer på den tekniska aspekten av en app. Fokus ligger på att finna prestandaproblem snarare än att hitta problem i människa-dator-interaktionen.

TestFairys betatestning går till som så att man laddar upp sin .apk-fil¹, väljer vilka testare man vill skicka ut till och sedan väntar man på resultatet. Detta underlättar för utvecklaren som inte behöver oroa sig över hur man implementerar TestFairys SDK² men en nackdel är att TestFairy inte har en krets med testare. The Beta Family har i dagsläget cirka 9 000 testare medan man i TestFairys fall får förlita sig på att man känner tillräckligt många vänner som vill testa appen.

2.1.2 Diverse appar som kräver root-access

Det finns en mängd appar som möjliggör inspelning av mobilskärmen men dessa kräver root-access. Att skaffa root-access är något som kräver en viss kunskap och innebär en viss säkerhetsrisk då man låser upp hela mobilen, ungefär som ett adminkonto i Windows. Detta innebär att dessa appar är svåra att använda i betatestningssyfte då man som företag inte kan kräva att alla betatestare rootar sina mobiler. Dessutom är det oftast mer avancerade användare som rootar sina mobiler och i betatestning är sannolikheten stor att man inte bara vill ha återkoppling från avancerade användare.

Eftersom dessa appar inte primärt är tänkta för betatestning går det inte uttala sig om deras lämplighet i sådana situationer. Appar har i regel inte tillåtelse att ta bilder av andra appars vyer så en app för betatestning skulle inte fungera utan det krävs någon form utav SDK. Apparna utför en delmängd (skärminspelning) utav det som detta projekt syftar genomföra men vissa delar (röstinspelning och kamerainspelning) saknas.

Exempel på dessa appar är *SCR Screen Recorder (scr)* och *Screen Recorder (sr)*

2.2 Teknisk bakgrund

Målet med projektet är att utveckla en SDK åt Androidapplikationer och ett krav från arbetsgivaren är att koden ska vara “native”, vilket innebär att utvecklingen kommer ske främst i Java, i Androids SDK.

2.2.1 Androidfragmentering

Androidfragmentering syftar till det faktum att Androidplattformen körs i olika versioner på hundratals olika hårdvaruspecifikationer, och då måste skillnader i exempelvis skärmstorlek och tillgängliga bibliotek tas i hänsyn under utvecklingen. Ett av målen med projektet är just att ge applikationsutvecklarna möjligheten att fastställa att deras applikationer faktiskt fungerar som de är tänkta på de olika versionerna av plattformen. Under utvecklingsfasen för detta projekt kommer testning framförallt att utföras på emulatorer som bl.a. ingår i Androids SDK, eller som det finns separat programvara för, exempelvis Genymotion som har en gratis licens. I skrivandets stund (2014-02-15) är KitKat (version 4.4) den senaste Androidversionen, men 20% av enheterna i cirkulation använder fortfarande Gingerbread (version 2.3.7).(**androidversions**) Endast 1,3%

¹ Installationsfilen för applikationen

² The Beta Familys iOS-testning förlitar sig på att utvecklaren själv implementerar deras SDK

använder en lägre version och Gingerbread är också den längsta versionen som stöds av Genymotion. Allt detta har lett till beslutet att Gingerbread är den längsta versionen som detta projekt kommer att stödja.

2.2.2 Application Sandboxing

Androidarkitekturen implementerar "application sandboxing" vilket innebär att olika applikationer inte kan interagera med varandra direkt. (**sandbox**) Av denna anledning är det en SDK som behöver utvecklas som sedan kan integreras med applikationen som behöver testas. En fristående applikation skulle inte ha tillräckligt mycket behörighet för att spela in skärmen på den applikationen som ska testas. Denna sandboxing medför även att tredjepartens applikationer som tangentbord och Google Maps inte kan spelas in. Android har denna arkitektur på grund av säkerhetsskäl. En applikation som exempelvis har tillgång till tangentbordet skulle kunna registrera allt användaren skriver, vilket potentiellt skulle kunna inkludera lösenordet till dennes internetbank.

2.2.3 Metadata

I projektet ingår att samla in metadata om den mobiltelefon som testaren använder. Specifikt ingår i kravlistan att ta reda på testarens telefonmodell och skärmstorlek, samt kontinuerligt skicka med testarens skärmorientering. Det finns goda möjligheter att enkelt begära fram dessa data ur det officiella Android-API:t genom modulerna *Build*, *view.WindowManager* och *app.Activity*. Åtkomst till dessa kräver endast grundläggande utvecklarbehörigheter (**adoc**).

2.2.4 Datapresentation

Insamlade data ska presenteras som ett flöde av information som gestaltar testarens användarmönster. Upprepade skärmdumpar ska tillsammans med testarens fingeravtryck bli till en video. Tillsammans med kamera- och ljudupptagning på testaren ska beställaren av testningen enkelt kunna bilda sig en uppfattning om hur produkten används.

Androids officiella API inkluderar en klocka med en upplösning på nanosekunder, *System.nanoTime()* (**adoc**). Klockan borde räcka gått och väl för att synkronisera kamera- och ljudupptagning med fingerläsning och skärminspelning. Med samma klocka kan man trivsamt spara tidsdata för varje skärmdump som tas, samt för varje fingerrörelse testaren gör. Man kan då representera testarens handlingar genom att "måla över" skärmdumparna med linjer som representerar fingerrörelser. Denna typ av enklare bildbehandlingsverktyg finns inbyggd i Android, i modulen *Canvas* (**adoc**). Dessa ihopslagna bilder kan sedan konverteras till en video med hjälp av den inbyggda modulnen *media.MediaCodec* (**adoc**). Skulle det visa sig att *media.MediaCodec* inte är kraftfull nog för våra ändamål finns även det fria alternativet *ffmpeg* tillgängligt på Android (**roman10**).

2.2.5 Back- och front-end

En back-end behövs för att ta emot datan som generas efter att en användare har betatstat en applikation och en front-end behövs för att呈现出 denna data för utvecklarna. Arbetsgivarna The Beta Family har redan en iOS-version av denna SDK med tillhörande back-end och front-end. Deras rekommendation är att det här projektet ska ha en egen miljö så att det inte uppstår ett beroende av deras kod, då data som generas i Android är annorlunda mot den som genereras i iOS. Dock bör samma ramverk som deras system är byggda på användas för att förenkla en senare integrering. Dessa ramverk är:

Back-end:

- Ramverk: Laravel

- Språk: PHP, MySQL, JavaScript

Front-end:

- Ramverk: Zurb
- Språk: HTML, CSS, JavaScript

2.2.6 Ljudinspelning

Ramverket för multimedia stöder ljudinspelning med en rad olika ljudformat med klassen MediaRecorder. Ljudformat och codecs som stöds är AAC LC, HE-AACv1, HE-AACv2, AAC ELD, AMR-NB, AMR-WB, FLAC, MP3, MIDI, Vorbis och PCM/WAVE (**sound**).

3 Riskbedömning

3.1 Inspelning av rörelser

I den version av Screen Recorder som används till iOS syns testarens rörelser i videon. Eftersom iOS-versionen har andra förutsättningar som möjliggör en högre bilduppdateringsfrekvens blir rörelserna naturliga. Om det inte är möjligt att genomföra på Android finns det en risk i hur man vill lösa presentationen av rörelser. Antag att lösningen blir liknande den TestFairy använder, detta innebär i så fall att en bild visas per sekund. Visas bara en punkt där fingret befinner sig på varje bild blir det omöjligt att avgöra hur användaren rör sig runt i applikationen. Det behövs någon form av "historia" för rörelserna för att man ska kunna avgöra om testaren drar fingret, trycker eller sveper. Risken ligger inte lika mycket i den tekniska biten som i att gruppen måste komma överens om hur det ska presenteras.

3.2 Skärminspelning

I och med att skärminspelning inte är lika enkelt att utföra på Android som på IOS om man inte har tillgång till root finns risken att vi inte får så många bilder per sekund som vi önskar. TBF har meddelat att om det inte finns någon möjlighet att få en bra bildhastighet så räcker det med att få det lika bra som TestFairy. Bristen på erfarenhet av att skapa en screen-capture gör dock att det finns en risk att prestandan inte blir tillräckligt bra. I och med att detta anses vara en risk har det tagits i åtanke vid skapandet av planeringen.

4 Projektplan

Till och med vecka 12 ligger två relativt tunga kurser parallellt med projektet. Därför kommer mindre tid kunna läggas på projektet då. Fokus läggs på rapporterna PPD1 och URD1 men även de lättare delarna av koden kommer färdigställas under den här tiden. Innan vecka 12 kan 10 timmar per person och vecka läggas på projektet, och efter vecka 12 beräknas 20 timmar per person och vecka kunna läggas, de sista veckorna möjligen mer.

Vecka 13 och framåt kommer mycket mer tid läggas på koden. Till att börja med ska skärminspelningen vara färdig till den 6/4. Sedan måste det jobbas fram ett sätt att komprimera media, och ett mindre backend-system att överföra inspelat data till. Till sist behövs en transfer-lösning. De sista punkterna kommer nog gå in i varandra en del, så de har alla samma deadline, 4/5, då all kod ska vara färdigställd.

Därefter är det dags att呈现出 och lämna in rapporterna ADD, PPD2 och URD2. Rapporterna kommer ha påbörjats parallellt med kodandet i April, därför kommer det inte

krävas så mycket jobb för att färdigställa dem. Dessutom får de med arbetsprocessen om de skrivs kontinuerligt under projektets gång. Eftersom de ska spegla slutresultatet kommer de inte kunna bli klara innan projektet är klart.

Deadline för all kod läggs den 4/5. Deadline för rapporterna och presentationsförberedelser läggs den 9/5.

Deliverable	Deadline	Uppskattad tidsåtgång
PPD-1	v.8 (19/2)	100h
URD-1	v.10 (5/3)	100h
ADD-1	Kursens slut	150h
PPD-2	Kursens slut	50h
URD-2	Kursens slut	50h

Deliverables

Systemmodul	Datum	Uppskattad tidsåtgång
Mikrofon	16/3	5h
Kamera	16/3	10h
Touch	16/3	10h
GUI	16/3	50h
Skärminspelning	6/4	300h
Media	4/5	300h
Transfer	4/5	200h
Backend	4/5	200h

Utveckling

Aktivitet	Deadline	Uppskattad tidsåtgång
Möten	-	260h
Mötesprotokoll	-	50h
Demo	Början av Maj	50h

Övrigt

4.1 Veckoplanering

Vecka	4	5	6
Antal timmar	100	60	70
Per person	10	6	7
Händelser	Föreläsningar	Föreläsning	Föreläsning
	Möte	Möte	Möte med The Beta Family
	Val av projekt	Fick projekt	Lunchmöte
	Skapa infrastruktur	Research om The Beta Family	Genomförbarhetsresearch
	Google Drive		

Vecka	7	8	9
Antal timmar	110	100	100
Per person	11	10	10
Händelser	Föreläsning	Möte	Möte
	Möte	Sammanfogning av PPD-1	
	Uppdelning av PPD-1	Presentationsförberedelse	
	Arbete med PPD-1	Inlämning av PPD-1	Arbete med URD-1
	Sätta igång med Trello	Arbete med kodgrunden	Arbete med kodgrunden
		Uppdelning av URD-1	
Vecka	10	11	12
Antal timmar	100	100	100
Per person	10	10	10
Händelser	Möte	Möte	Möte
	Sammanfogning av URD-1	Färdigställa kodgrunden	Arbete med skärminspelning
	Presentationsförberedelse		
	Inlämning av URD-1		
	Arbete med kodgrunden		Arbete med URD-2
Vecka	13	14	15
Antal timmar	200	200	150
Per person	20	20	15
Händelser	Arbete med skärminspelning	Färdigställa skärminspelning	Mediaprogrammering
			Backendprogrammering
			Transferprogrammering
			Arbete med ADD
	Arbete med URD-2	Arbete med URD-2	Arbete med URD-2
Vecka	16	17	18
Antal timmar	200	200	300
Per person	20	20	30
Händelser	Möte	Möte	Möte
	Mediaprogrammering	Mediaprogrammering	Deadline för all kod
	Backendprogrammering	Backendprogrammering	
	Transferprogrammering	Transferprogrammering	
	Arbete med ADD	Arbete med ADD	Arbete med ADD
	Arbete med URD-2	Arbete med URD-2	Arbete med URD-2

Vecka	19	20
Antal timmar	250	30
Per person	25	3
Händelser	Möte	Presentation
	Förbereda DEMO-presentation	
	Färdigställa ADD	
	Färdigställa PPD-2	
	Färdigställa URD-2	

I schemat står alla aktiviteter som ska utföras under veckan. Vissa aktiviteter som utförs varje vecka står inte med, dessa är mötesprotokollsskrivning och rapportstyrning som utförs av William, projektstyrning som utförs av Paulina och styrning av programmeringsarbetet som utförs av Daniel. Det kommer med största sannolikhet även tillkomma mindre uppgifter. På grund av detta innebär inte de uppskattade timmarna i veckoschemat att alla i gruppen kommer att arbeta med just de uppgifter som står i schemat hela den tiden. Troligt är dock att de utan specifika ansvarsområden kommer ägna nästan all sin planerade tid åt de schemalagda aktiviteterna.

Det finns inte så mycket utrymme för projektet att ta längre tid än planerat. Men tiden som krävs för varje delmoment är uppskattad med god marginal, så det är sannolikt att projektet kommer hamna före schemat. Det finns även utrymme att spendera mer tid per vecka än planerat om det skulle behövas. Projektledaren kommer kontinuerligt att följa upp och justera tidsplaneringen så ingenting blir försenat. Det blir Chief Programmer's uppgift att sätta upp delmål i programmeringsdelen och följa upp dessa. Projektledaren och Chief Programmer håller nära kontakt.

5 Genomförbarhetsstudie

5.1 Kravlista

5.1.1 Mobilen får inte vara “rootad”

Kravet att mobilen inte får vara “rootad” (föklarat i ??) sänker dessvärre möjligheterna att kunna implementera en snabb och täckande skärminspelare. Utan “root”-behörighet kan tredjepartsutvecklare inte komma åt den centrala grafiska modulen i Android (**adoc**), och kan därför varken utnyttja systemets snabbhet eller kunna fånga en heltäckande bild av skärmen. Även om det tekniskt skulle vara mycket enkelt att släppa denna behörighet fri även från tredjepartsutvecklare, har Google gjort det klart att det inte ligger i deras intresse för närvarande (**uhno**).

Detta gör att gruppen endast är förmögen att skriva en SDK som bara kan spara grafiken som hör till den testade applikationen. Exempelvis tangentbord, webbläsare och andra inbyggda systemmoduler verkar inte gå att fånga i nuläget. Även hastigheten kan komma att bli begränsad, i förhållande till vad som skulle vara möjligt med “root”-behörighet. Troligen kommer prestanda och funktion att ha svårt att överskrida *TestFairy*, som behandlas i ?. Nöjar arbetsgivaren sig med den kvalitetsnivån är kravet dock genomförbart, då *TestFairy* inte kräver någon “root”-behörighet.

5.1.2 Mjukvaran måste vara skriven i Java

Alla gruppens medlemmar har gått samma introduktionskurs i datalogi, vilken till stor del behandlar språket Java (**inda**). Språket Java används som en grundsten i Android-plattformen, och Androids officiella API³ är skrivet just i Java (**adoc**).

Detta officiella API innehåller tillräcklig programkod för att spela in ljud och film, spara fingerrörelser över tid samt ta delvis täckande skärmdumpar. Med bakgrund av gruppmedlemmarnas erfarenheter i Java, samt de möjligheter som Androids Java-API erbjuder, kan kravet klassas som genomförbart.

5.1.3 SDK:n måste vara enkel att integrera

Helst av allt ska SDK:n leva upp till den nuvarande versionen för iOS, som ska gå att integrera på ca 2 minuter (**superrec**). Enligt nuvarande implementationsplaner behöver bara SDK:n länkas till ett objekt från den applikation som ska testas, nämligen rot-vyn⁴. Att göra detta är i stort sett en trivialitet, så länge applikationsutvecklaren som vill testa sitt program kan ändra fritt i sin egen källkod. Kravet borde därför vara i högsta grad genomförbart.

5.1.4 Information som måste skickas med

Specifikt måste telefonmodell, skärmstorlek och skärmorientering skickas till arbetsgivarens server från SDK:n. Detta har behandlats i **??**. Även information om fingertryck och -gester ska skickas med.

5.1.5 Användarens ansikte och röst ska kunna spelas in

5.1.6 Informationen ska presenteras som ett tidsflöde

Som behandlats i **??** finns många bra verktyg för en sekventiell, högupplöst datarepresenation. Det finns dock en risk att skärminspelningens låga frekvens kan ha en negativ inverkan på informationsförmedlingen. Att skapa ett flöde av information från den insamlade datan verkar dock möjligt med befintliga kunskaper och verktyg.

5.1.7 En serverapplikation som behandlar och presenterar data

Att skriva en serverapplikation i de språk och ramverk som nämns i **??** är något som gruppens medlemmar överlag har låg erfarenhet av. Serverapplikationen kommer att innehålla en databas av typen relationsdatabas, något som alla gruppens medlemmar lär sig eller har lärt sig i kursern *Databasteknik för D* (**dbas**). Den här delen är heller ingen prioritet för projektet, utan är mest till för att ta emot och呈现出 de data som ansamlas i SDK:n. På grund av serverapplikationens låga vikt för projektet är det gruppmedlemmarnas åsikt att modulen i fråga är fullt möjlig att producera, medlemmarnas erfarenheter till trots.

³Protokoll för ett program som ska kommunicera med andra program

⁴Den mest täckande grafiska representationen av applikationen

5.2 Sidouppgifter

5.2.1 SDK:n ska vara så lik iOS-versionen som möjligt

5.2.2 Tredjeparts-program ska kunna spelas in

Som behandlas i ?? verkar det vara en teknisk omöjlighet att spela in tredjepartsprogram med befintliga verktyg och krav från arbetsgivaren. Då detta är ett i dagsläget olöst problem, även i industrin bland erfarnare systemutvecklare, är det inte rimligt att vänta sig att gruppen ska kunna lösa detta.

5.3 Summering

Appendix

A Gruppmedlemmar

Daniel Bergström	Role	danbe187@gmail.com	0704134946
Paulina Hensman	Role	paulina.hensman@gmail.com	0763058320
Marcus Hertz	Role	hertz.marcus@gmail.com	0706570078
Simon Karlsson	Role	krlsn743@gmail.com	0735401871
David Masko	Role	david.maskoo@gmail.com	0707987913
Marcus Nordberg	Role	nordbergaren93@gmail.com	0732544789
André Nyström	Role	andreny@kth.se	0730882106
William Perkola	Role	perkola.william@gmail.com	0703024342
Axel Riese	Role	axel.riese@gmail.com	0732455919
Kerry Zhang	Role	kerryzhang92@gmail.com	0737633721

B Ursprungligt projektförslag



The Beta Family

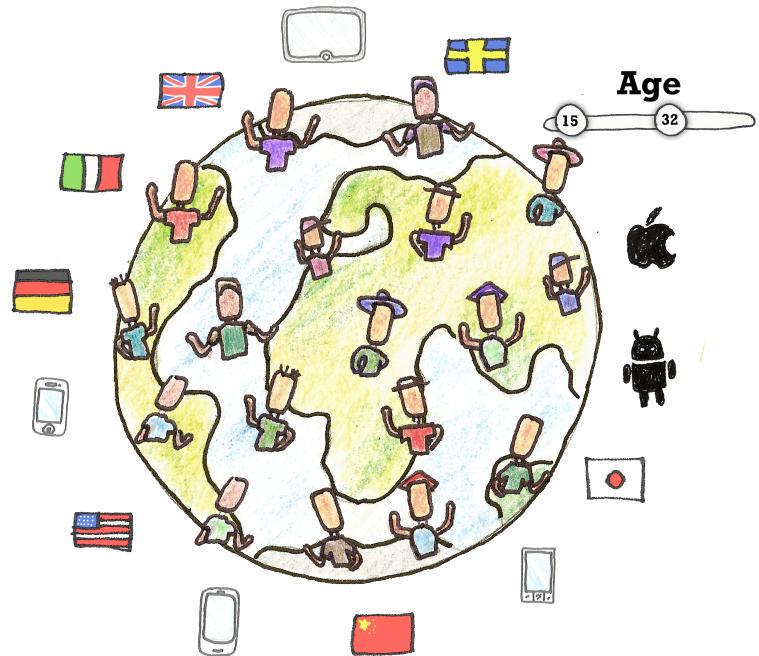
SuperRecorder - Record Android applications

Background

The Beta Family is a web-based crowdsourcing platform for beta testing mobile applications. Developers can test their apps on relevant target groups, operating systems and devices. The testers give feedback on the app so the developer can fix bugs and improve the app before release or get suggestions on new features for upcoming updates. This results in better and more user-friendly applications and increasing sales. The service has more than 8000 testers and has completed 1500 application tests.

How the service works

The developer uploads an app to The Beta Family and selects the target group, nationality, OS and device models that the app should be tested on and how much the tester will make by participating in the test. The members that fit the target group can apply to test the app. Accepted testers will answer questions regarding usability, graphics, sound, and bugs and errors. Finally, the developer receives a test report, rate the tester and pay the reward.

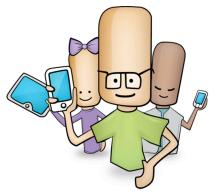


Recording applications

To get the full picture of how the tester is really using the app and how it looks and works on different screen sizes and OS, The Beta Family released the SuperRecorder SDK for iOS apps in October 2013. The SDK makes it possible to record a video of the full mobile/app-screen, and shows touches, swipes, pinching, and the testers voice and face expressions (recording the front camera). Go here for a demo video: <http://thebetafamily.com/superrecorder>

Project description – Super Recorder Android SDK

We want your help to build an SDK for Android! This will probably be the worlds first SDK doing this, and will be very useful since the Android market is even more differentiated (screen sizes, OS versions etc) than on the iOS-platform.



The Beta Family

Requirements and challenges

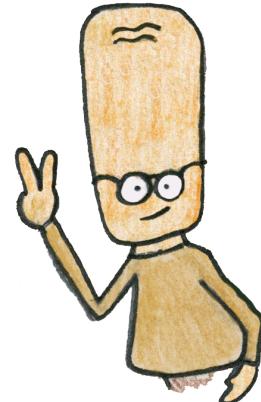
- The Android device should not need to be rooted
- Native, the SDK will be built in Java
- The SDK should be easy to integrate. (To give you an idea, the iOS version takes 2 min)
- The recorded screen should give an ok framerate (this should be one of the first things to test in a Proof of concept)
- Touches and swipes should be marked in green, see iOS version for reference.
- The phone screen size and orientation should be sent to the server
- The users face and voice should be recorded; this can be uploaded to the server as a separate video if needed.
- Can we make hooks to OpenGL to record eg. games with a good frame rate?
- How do we record views with 3rd party SDKs like eg. Google maps?

Resources

We have done some research on this and know that it's possible, so we can have a good dialog on how to make this in to reality!

Other cool stuff

- The Beta Family is founded by app developers from KTH.
- The company is looking for talents for extra and full time work and sees this project as a way to get in contact with great people.
- The SDK will be used by thousands of people and is a great working reference.
- The company is working in a rapidly growing market, with more than 2000 apps added to the app stores each day.
- The Beta Family has a fun entrepreneurial team working in a spectacular office, next to the water at Långholmen, Hornstull.



Contact information and supervisor

Axel Nordenström
axel@thebetafamily.com
+46 70 404 92 72

The B Family AB
Mälarvarvsbacken 8
117 33 Stockholm

Looking forward to work with you

Axel Nordenström, CEO