

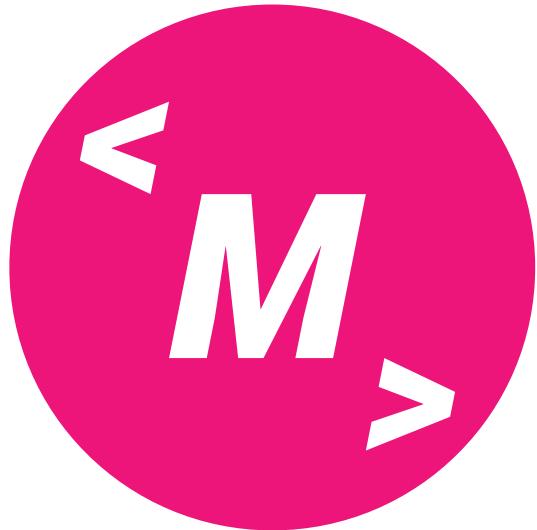
PPD-1

The Meta Family

Version 0.0

den 15 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
1 Problembeskrivning	2
1.1 Detaljerad problembeskrivning	2
1.2 Motivering	2
1.2.1 Tekniska områden	2
1.2.2 Framtiden inom teknisk utveckling	2
1.3 Målsättning	2
1.4 Färdigheter och förkunskaper	2
2 Bakgrund	2
2.1 Kommersiell bakgrund	2
2.1.1 TestFairy	3
2.1.2 Diverse appar som kräver root-access	3
2.2 Teknisk bakgrund	3
2.2.1 Fragmentation	3
2.2.2 Application Sandboxing	4
2.2.3 Back- och front-end	4
3 Riskbedömning	4
3.1 Ljudinspelning	4
4 Projektplan	5
4.1 Veckoplanering	6
5 Genomförbarhetsstudie	7
5.1 Kravlista	7
5.1.1 Mobilén får inte vara "rootad"	7
5.1.2 Mjukvaran måste vara skriven i Java	8
5.1.3 SDK:n måste vara enkel att integrera	8
5.1.4 Information som måste skickas med	8
5.1.5 Användarens ansikte och röst ska kunna spelas in	8
5.1.6 Informationen ska presenteras som ett tidsflöde	8
5.1.7 En serverapplikation som behandlar och presenterar datan	8
5.2 Sidouppgifter	8
5.2.1 SDK:n ska vara så lik iOS-versionen som möjligt	8
5.2.2 Tredjeparts-program ska kunna spelas in	8
5.3 Summering	8
Appendix	8
A Gruppmedlemmar	8
B Ursprungligt projektförslag	8

Versionshistorik

1 Problembeskrivning

1.1 Detaljerad problembeskrivning

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.

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. Det är helt klart en fördel att känna till hur utvecklingsprocessen av en app går till när man som nyexaminerad civilingenjör når arbetslivet.

1.3 Målsättning

1.4 Färdigheter och förkunskaper

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 nавигerar 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 tillstånd 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 Fragmentation

“Android fragmentation” 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

¹ Installationsfilen för applikationen

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

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% 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 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, JS

Fron-end:

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

3 Riskbedömning

3.1 Ljudinspelning

En potentiell risk för projektet är inspelningen från mikrofonen då ingen i gruppen har tidigare erharenhet av just detta. Vid närmare undersökning ser det dock inte ut som ett större hinder då ramverket för multimedia stöder ljudinspelning med en rad olika ljudformat med klassen MediaRecorder(Google Inc, 2013) och det finns dessutom en steg-för-steg-guide på Androids utvecklarhemsida. Med gruppens samlade erharenhet av programmering anses därför ljudinspelningen vara en större risk. Av denna anledning har tidsåtgången för ljuinspelningen uppskattats till 5 timmar (se avsnitt 4).

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 möjliger 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 kodgrundens	Arbete med kodgrundens
		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 kodgrundens	Arbete med skärminspelning
	Presentationsförberedelse		
	Inlämning av URD-1		
	Arbete med kodgrundens		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” (föklarad i 2.1.2) 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 2.1.1. 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

5.1.3 SDK:n måste vara enkel att integrera

5.1.4 Information som måste skickas med

5.1.5 Användarens ansikte och röst ska kunna spelas in

5.1.6 Informationen ska presenteras som ett tidsflöde

5.1.7 En serverapplikation som behandlar och presenterar datan

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

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	crazybanana3255@gmail.com	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

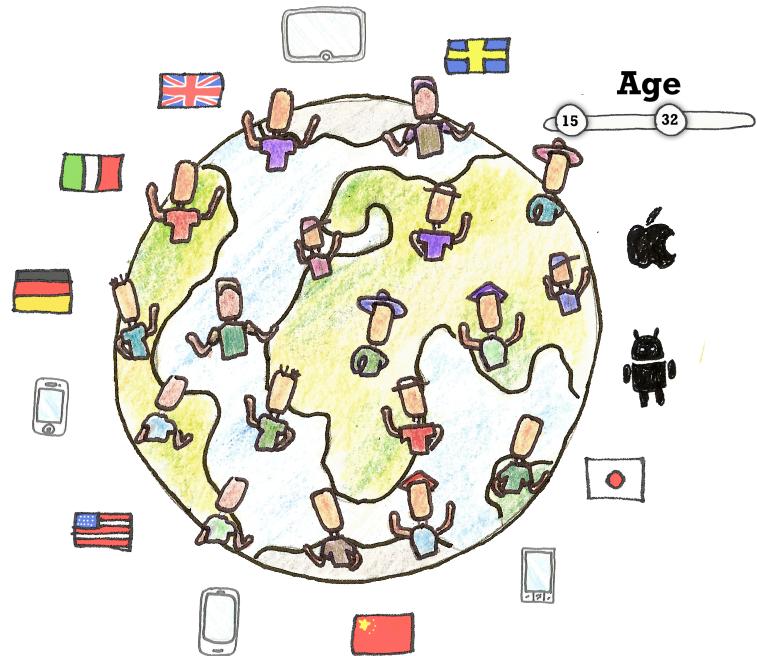
SuperRecorder - Record Android applications

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.

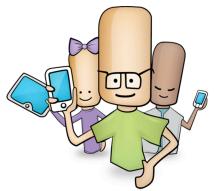


Recording applications

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

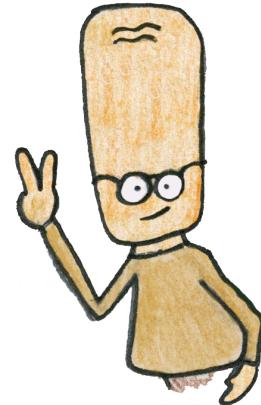
- The Android device should not need to be rooted
- Native, the SDK will be built in Java
- The SDK should be easy to integrate. (To give you an idea, the iOS version takes 2 min)
- The recorded screen should give an ok framerate (this should be one of the first things to test in a Proof of concept)
- 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