



HÖGSKOLAN
DALARNA

EXAMENSARBETE

En utvärdering av Android och Iphone

*För utveckling av webbtjänstunderstödda applikationer
ämnade för intern distribution.*

An evaluation of Android and Iphone

*For development of web service supported
applications intended for internal distribution.*

*Mikaela Hedberg
Anders Neva-Juoni*

2010-06-10

Akademin industri
och samhälle

Nr: 07/2010



EXAMENSARBETE, Grundnivå 2 i Informatik

Ämne Informatik, Grundnivå 2	Reg. nr 07/2010	Omfattning 15 hp
Namn Mikaela Hedberg Anders Neva-Juoni	Month/Year Juni 2010	
	Handledare: Examinator:	Pär Eriksson Mark Dougherty
Företag/Institution Sogeti	Handledare vid företaget/institutionen Martin Olsén	
Title En utvärdering av Android och Iphone - För webbtjänstunderstödda applikationer ämnade för intern distribution inom organisationer.		
Keywords Utvärdering, Android, Iphone, webbtjänstunderstödd, intern distribution, smartphone, Xcode, Eclipse, applikation		

SAMMANFATTNING

Tack vare bättre och bättre förutsättningar för utveckling av mobila applikationer, samt utbredning av internetbaserade tjänster, presenteras här ett underlag inför val av mobil utvecklingsplattform. De undersökta plattformarna är Android och Iphone på grund av deras växande utbredning på smartphonemarknaden. Studien presenterar förutsättningar för att utveckla webbtjänstunderstödda applikationer för intern distribuering. Ett visuellt tilltalande användargränssnitt är också i fokus för den typen av applikation studien riktar sig mot.

Vi har kommit fram till att Android är lättare att lära sig om man kommer ifrån en Java- eller .Netmiljö samt lättare att distribuera. Iphone har däremot bättre stöd för att utveckla grafiskt tilltalande applikationer. Båda plattformarna har dock bristfälligt stöd för kommunikation via webbtjänster. Detta resultat uppnåddes genom litteraturstudier, samt en fallstudie där vi utvecklade applikationer med fokus på just webbtjänstkommunikation, intern distribuering samt ett tilltalande användargränssnitt.



DEGREE PROJECT, Undergraduate level 2 in Informatics

Subject Informatics, Undergraduate level 2	Reg. number 07/2010	Extent 15 hp
Name Mikaela Hedberg Anders Neva-Juoni	Month/Year June 2010	Supervisor: Pär Eriksson Examiner: Mark Dougherty
Company/Department Sogeti	Supervisor at the company/department Martin Olsén	
Title An evaluation of Android and Iphone - <i>For web service supported applications intended for internal distribution within organizations.</i>		
Keywords Evaluation, Android, Iphone, web service supported, distribution, smartphone, Xcode, Eclipse, application		

SUMMARY

Thanks to better and better conditions for development of mobile applications, and the prevalence of Internet-based services, this paper presents a foundation for a choice of mobile development platform. The examined platforms are Android and Iphone because of their growing prevalence of smartphone market. The study presents prerequisites to develop web service-supported applications for internal distribution. A visually appealing user interface is also the focus of the type of application covered in this study.

We have come to the conclusion that Android is easier to learn if you derive from a Java or .Net-environment and that it is easier to distribute. Iphone, on the other hand, has better support for the development of graphically appealing applications. Both platforms have insufficient support for communication via web services. This result was achieved through literature studies and a case study where we developed applications with a focus on web service communications, internal distribution and an appealing user interface.

INNEHÅLLSFÖRTECKNING

1. INLEDNING	1
1.1. Bakgrund.....	1
1.1.1. Uppdragsgivare.....	2
1.1.2. Befintlig applikation	2
1.2. Problematisering	2
1.2.1. Vad krävs för att kunna utveckla en applikation såsom golfapplikationen för Android respektive Iphone?.....	3
1.3. Syfte.....	3
1.4. Mål	4
1.5. Avgränsningar	4
2. METOD	5
2.1. Metodik.....	5
2.1.1. Kunskapssyn och kunskapsbidrag	5
2.1.2. Induktiv forskningsansats	5
2.1.3. Litteraturstudier	5
2.1.4. Fallstudie	6
2.2. Tillvägagångssätt	6
2.2.1. Fas 1 - Analys av existerande applikation.....	7
2.2.2. Fas 2 – Fastställa förutsättningar för Android och Iphone	8
2.2.3. Fas 3 - Utformning av applikation	9
2.2.4. Fas 4 - Utvärdering av Android och Iphone.....	10
3. CENTRALA BEGREPP	11
3.1. Smartphone	11
3.2. Java.....	11
3.3. Android.....	12
3.4. Objective-C.....	13
3.5. Iphone OS.....	13
3.6. MVC.....	13
3.7. XML.....	14
3.8. Webbtjänster.....	14
3.8.1. SOAP	15
4. MOBILUTVECKLING OCH OPERATIVSYSTEM	17
4.1. Bakgrund för utveckling av mobilapplikationer.....	17

4.2. Utvecklingsplattformar och mobila operativsystem.....	18
4.2.1. <i>Fördelning av operativsystem hos smartphones</i>	19
4.3. Användning av webbtjänster hos mobilapplikationer	21
5. RESULTAT	22
5.1. Analys av existerande mobilapplikation	22
5.1.1. <i>Flöde och funktioner</i>	22
5.1.2. <i>Webbtjänsten</i>	23
5.1.3. <i>Datakommunikation</i>	23
5.1.4. <i>Identifierade krav baserad på tidigare funktionalitet</i>	23
5.2. Förutsättningar för Android.....	24
5.2.1. <i>Utvecklingsmiljö</i>	24
5.2.2. <i>Användargränssnitt och språkstöd</i>	25
5.2.3. <i>Distribution och test</i>	26
5.2.4. <i>Webbtjänstkommunikation</i>	27
5.3. Förutsättningar för Iphone	27
5.3.1. <i>Utvecklingsmiljö</i>	27
5.3.2. <i>Användargränssnitt och språkstöd</i>	29
5.3.3. <i>Distribution och test</i>	30
5.3.4. <i>Webbtjänstkommunikation</i>	31
5.4. Golfapplikationen	32
5.4.1. <i>Design av applikationerna</i>	32
5.4.2. <i>Problem med den existerande webbtjänsten</i>	32
5.4.3. <i>Realisering av applikationerna</i>	33
5.4.4. <i>Distribution av golfapplikationen</i>	34
6. UTVÄRDERING AV ANDROID OCH IPHONE	35
6.1. Inläring av utvecklingsplattform.....	35
6.1.1. <i>Utvecklingsmiljö</i>	35
6.2. Användargränssnitt och språkstöd	36
6.3. Distribution och test	36
6.4. Webbtjänstkommunikation	37
7. UNDERLAG FÖR VAL AV PLATTFORM	38
7.1. Android är lättare att lära sig och distribuera	38
7.2. Iphone, bättre förutsättningar för tilltalande GUI.....	39
7.3. Otillräckligt stöd för webbtjänster	39
8. DISKUSSION	41

8.1. Syftets och frågeställningarnas relevans.....	41
8.2. Reflektion över resultatets kvalitet.....	41
8.3. Vidareutveckling av golfapplikationen.....	42
8.4. Uppslag till vidare studier.....	42

FIGURFÖRTECKNING

Figur 2-1: Tillvägagångssätt	7
Figur 3-1: Webbtjänst - Applikationscentrerad webb (Cerami, 2002).....	15
Figur 4-1: Operativsystem (försäljning), Gartner, 2010.....	19
Figur 4-2: Operativsystem (försäljning), Gartner, 2010	20
Figur 4-3: Operativsystem (användning), Admob, 2010	20
Figur 5-1: Ad-hoc-distribution av Iphoneapplikation	31

1. INLEDNING

I detta kapitel beskrivs bakgrunden på vilket arbetet är baserat. Här beskrivs uppdragsgivaren Sogeti och deras mål med de applikationer som kommer produceras under arbetets gång. Baserat på bakgrunden har en problematik trätt fram, vilket presenteras i avsnitt 1.2. Vidare omnämns arbetets syfte och dess avgränsning.

1.1. Bakgrund

Utveckling på mobila plattformar har de senaste åren tagit väldigt stora steg framåt. Där det tidigare fanns begränsningar hos plattformarna i form av minnes- och processorresurser och det faktum att uppkopplingshastigheten var väldigt långsam (se avsnitt 4.1), finns idag istället plattformar med hög prestanda och goda tekniska förutsättningar. Detta gör det möjligt att utveckla applikationer som är väldigt resurskrävande, t.ex. video och 3D-animerade spel. Detta är möjligt främst p.g.a. att smartphones har tagit större och större delar av marknaden (se avsnitt 4.2). Smartphones, i denna kontext, identifieras som ett mellanting mellan handdator och mobiltelefon, med fokus på applikationer som t.ex. webbläsare, e-postklient och applikationer för sociala nätverk (Fling, 2009). Denna progression har lett till att tillverkarna fokuserar mer på att utveckla applikationer för dessa enheter istället för traditionella mobiltelefoner.

En annan utveckling för mobila enheter är dess tillgänglighet. I och med trådlös internetuppkoppling och 3G-teknik kan telefoner kommunicera med webbaserade tjänster för att ge tillgång till aktuell information hämtad ifrån nätet. Detta öppnar nya sätt att se på hur mobiltelefonapplikationer kommunicerar med andra resurser, såsom webbtjänster (Krejcar, Janckulik, & Motalova, 2010).

Enligt en rapport från Admob, har Iphone OS, Android och Symbian OS tillsammans 92 % av mobiltelefonmarknaden. Denna statistik uppnås genom att mobiltelefonanvändaren navigerar in på någon av Admobs kunders webbsidor, således representerar dessa siffror bara de telefoner som används till att faktiskt surfa på internet (se avsnitt 4.2).

Vi känner att vi saknar förståelse för många av dessa nya mobila operativsystem och plattformar och är intresserade på att ta reda på mer om vad dessa plattformar har att erbjuda framtidens mobiltelefoniutveckling, vad gäller kommunika-

tionsmöjligheter via webbtjänster samt utveckling av intuitiva användargränssnitt. Vi känner även ett behov av att ta reda på vad de olika plattformarna erbjuder oss som utvecklare samt att skapa en utgångspunkt för andra utvecklare att ta ställning till vilken plattform som bör väljas vid utveckling av sådana mobila applikationer.

1.1.1. Uppdragsgivare

Sogeti är en ledande leverantör av professionella tekniska tjänster, specialiserade på IT-styrningstjänster, IT-specialisttjänster, utvecklings- och integrationsprojekt, testning, systemförvaltning samt Rightshore-tjänster. Sogeti har över 20 000 anställda i 15 länder och finns på över 200 platser i Europa, USA och Indien. Sogeti är ett helägt dotterbolag till Cap Gemini SA, noterat på Parisbörsen. (Sogeti.com, 2010)

Sogeti AB - Borlänge, är ett konsultföretag som främst jobbar mot större företag i Dalaregionen. De erbjuder "lokala IT-lösningar på global grund" (Sogeti.se, 2010), med detta menas att de med hjälp av hela deras globala nätverk av experter kan erbjuda sina kunder förstklassig hjälp men att lokala anknytningar till kunden bibehålls, tack vare att de har kontor så nära.

1.1.2. Befintlig applikation

Uppdragsgivaren har idag en applikation för inrapportering och översikt av golfresultat, med en webbapplikation skriven i ASP.NET och en tillhörande mobil applikation för Windows Mobile 5. Denna applikation behöver nu utvecklas för att stödja fler, och mer aktuella, mobila plattformar.

På den webbaserade delen av denna applikation finns mer omfattande administration i form av hantering av banor, förberedning av uppkommande golfrunda, hantering av spelare, översikt över tidigare resultat o.s.v. I mobilapplikationen är det sedan möjligt att hämta information om tävlingar och resultat, samt rapportera in nya resultat. Denna information kommuniceras via en webbtjänst.

1.2. Problematisering

Med grund i den ständigt växande smartphonemarknaden, användandet av webbtjänster samt uppdragsgivarens behov att förnya sin applikation har vissa frågeställningar börjat formas.

Vår kunskap om utveckling av den typen av applikation som uppdragsgivaren efterfrågar är i nuläget väldigt liten. Därmed börjar funderingar över kriterier och

förutsättningar för sådan typ av utveckling formars och ett behov av en större förståelse för detta finns. Vi saknar underlag för vilka för och nackdelar som finns hos Android- och Iphoneplattformarna för att utveckla en webbtjänstunderstödd, visuellt tilltalad och lätt distribuerad mobil applikation.

1.2.1. Vad krävs för att kunna utveckla en applikation såsom golfapplikationen för Android respektive Iphone?

Problematiseringen har lett oss till att ställa frågan: Vad krävs för att utveckla en applikation såsom golfapplikationen för Android respektive Iphone. Vi är undrande över vilka förutsättningar, i form av kunskapsresurser och distributionsmöjligheter, som finns för att bygga en applikation av den typen som efterfrågas. Detaljer om applikationens önskade utformning beskrivs i större detalj i avsnitt 1.4. Ett centralt krav för applikationen är kommunikation med en redan existerande webbtjänst och vi kommer att undersöka hur applikationer med stöd för webbtjänstkommunikation utvecklas för Android- och Iphoneplattformarna.

De förutsättningar hos plattformarna som vi specifikt kommer att koncentrera oss på är följande:

- Lätthet att lära sig grundläggande plattformsstruktur, programmeringsspråk och utvecklingsmiljö.
- Förutsättningar för utveckling av ett tilltalande användargränssnitt, samt stöd för flera språk i applikationen.
- Förutsättningar för test och distribution
- Möjlighet till kommunikation med webbtjänst.

1.3. Syfte

Studien kommer att erbjuda underlag för beslut rörande huruvida Android eller Iphone är den bäst passande plattformen, vad gäller hur enkelt det är att lära sig plattformen, distributionsmöjligheter och förutsättningar för utveckling av användargränssnitt. Underlaget är riktat till företag eller organisationer som önskar utveckla grafiskt tilltalade, webbtjänstunderstödda applikationer för intern distribution och står inför ett beslut rörande val av plattform.

1.4. Mål

Uppdragsgivaren efterfrågar en applikation som ska användas för inrapportering av golfresultat. Vidare ska användaren av golfapplikationen kunna se en översikt över sina egna, och sina medtävlades resultat.

Telefonen ska agera som en presentationsklient, då all data lagras på en server och hämtas till telefonen då användaren önskar se den. Data som telefonen behöver ska införskaffas via kommunikation med en webbtjänst. Det finns i dagsläget redan en webbtjänst för datakommunikation. Vidare ska de resultat som rapporteras in via telefonen skickas till en server via samma webbtjänst.

Applikationen ska fungera till dagens och framtidens mobiltelefoner. Detta medför ett behov av en applikation som ska ha ett intuitivt grafiskt användargränssnitt, anpassat för pekskärmar och utomhusmiljö. Det grafiska användargränssnittet bör därför bestå av stora och tydliga komponenter med hög kontrast, så att det är lätt att navigera runt i programmet via mobiltelefonen. Det ska även finnas stöd för flera språk i applikationen.

Applikationen ska distribueras internt och till några av uppdragsgivarens kunder. Uppdragsgivaren vill således kontrollera distributionen av applikationerna, snarare än att göra dem tillgängliga för en offentlig marknad.

1.5. Avgränsningar

Arbetet kommer att avgränsa sig till att enbart jämföra Android- och Iphoneplattformarna baserat på kriterier för att utveckla applikationer, snarare än att se till hur applikationer för de respektive plattformar används och uppfattas av de användare som ska använda applikationen.

Webbtjänstkommunikationen kommer att omfatta SOAP-baserade webbtjänster, snarare än t.ex. webbtjänster baserade på RPC-XML eller REST. Detta för att den existerande webbtjänsten använder sig av SOAP, så en naturlig infallsvinkel blir att undersöka förutsättningar för SOAP-baserade webbtjänster snarare än de andra alternativen.

Vi kommer inte att jämföra alla versioner av Android- och Iphoneplattformarna, utan vi kommer att koncentrera oss på Iphone OS 3.1.3 samt Android 2.1, som i skrivande stund är de nyaste versionerna för respektive operativsystem.

2. METOD

Nedan beskrivs dels den mer generella metodiken runt hur detta arbete genomförts, samt mer detaljerad beskrivning av tillvägagångssätt.

2.1. Metodik

För att ge inblick i våra arbetsmetoder och det synsätt vi har på detta arbete presenteras här en översiktlig beskrivning för att klargöra vad som menas med litteratur- respektive fallstudie som metoder för detta arbete. Vidare beskrivs vår kunskapssyn, vilken typ av kunskap studien kommer att bidra med och vilken ansats vi har inför detta arbete.

2.1.1. Kunskapssyn och kunskapsbidrag

I detta arbete har vi en icke positivistisk kunskapssyn då vi vill bygga upp en förståelsebas för de båda plattformarna. Vi kommer därmed att bidra med karaktäriserande kunskap då syftet är att förmedla ett beskrivande underlag för vilka förutsättningar som finns hos Android och Iphone i avseendet att utveckla en webbtjänstunderstödd, visuellt tilltalande applikation för intern distribution.

2.1.2. Induktiv forskningsansats

En induktiv forskningsansats finns i detta arbete, vilket innebär att data samlas in utan grund i tidigare teorier (Björklund & Paulsson, 2003).

Detta stämmer väl in på vårt arbete, då vi har ett kreativt synsätt på arbetet och vi, genom själva utvecklingen av applikationerna, kommer att införskaffa kunskapen som krävs för att ta ställning till den problematik som vi formulerat. Vi baserar inte våra problem på existerande teori, utan utgår ifrån våra egna slutsatser vid utveckling av applikationerna för att utifrån detta bidra med den teori som är syftet för detta arbete. Vi har även sett att det är ont om tidigare teorier rörande ämnet, då utveckling för dessa plattformar är en relativt ny företeelse, vilket också leder oss mot den induktiva forskningsansatsen.

2.1.3. Litteraturstudier

Litteraturstudier, i betydelsen "allt skrivet material" (Björklund & Paulsson, 2003), kommer att bedrivas. Vi kommer att utforska tillgängliga utvecklingsmiljöer, krav och omfattning av de båda plattformarna för att få en grundläggande förståelse inför utvecklingsarbetet.

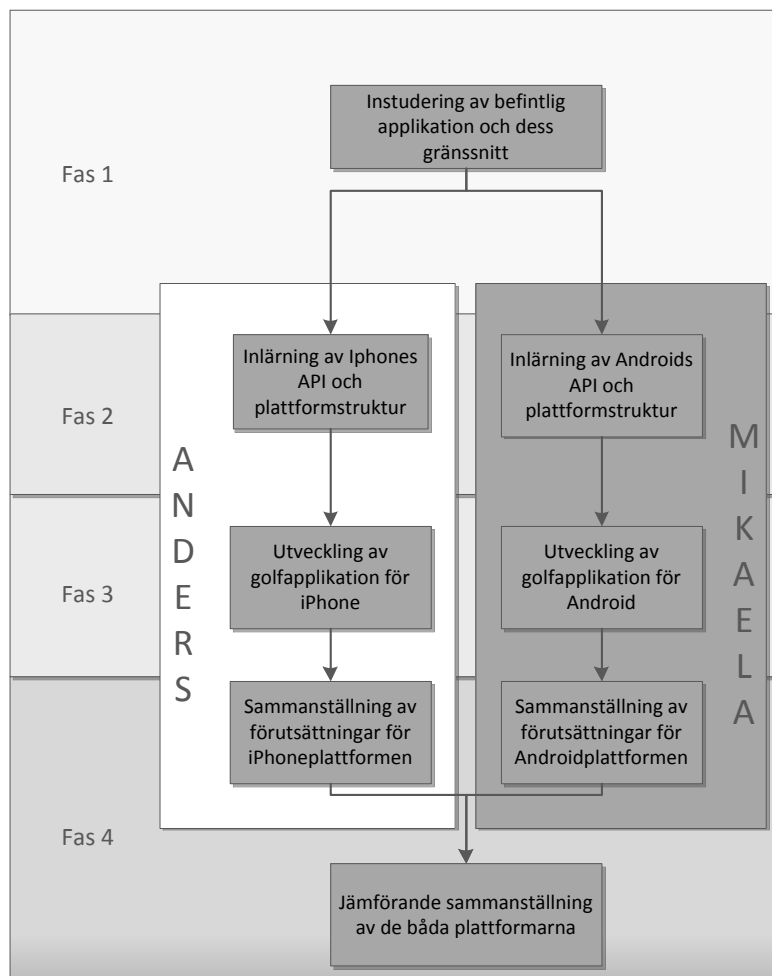
2.1.4. Fallstudie

En undersökande fallstudie, vilket är en metod för att undersöka något specifikt i sin realistiska miljö (Backman, 1998), används som metod i detta arbete. Fallen som studeras i detta arbete är de plattformar som applikationerna bygger på. Fallstudien är ett sätt att få mer djupgående praktisk kunskap om plattformarna som inte skulle kunna uppnås med enbart litteraturstudier.

2.2. Tillvägagångssätt

Arbetet kommer att delas upp i fyra faser, varav två av dem uppdelas för att undersöka Android respektive Iphone var för sig. I den första fasen analyseras den existerande mobilapplikationen för att få insikt i vilka funktionella krav som finns för den applikationen som ska byggas. I de följande två faserna kommer författarna separat införskaffa kunskap och utveckla golfapplikationen för respektive plattform; Anders för Iphone och Mikaela för Android.

Slutligen kommer den sammanställande fasen att resultera i en utvärdering av de båda plattformarna. Nedan beskrivs en övergripande graf över faserna, följt av en mer detaljerad beskrivning av vårt tillvägagångssätt.



FIGUR 2-1: TILLVÄGAGÅNGSSÄTT

2.2.1. Fas 1 - Analys av existerande applikation

Första steget innebär att analysera den redan existerande golfapplikationen för Windows Mobile 5, för att ge underlag för en funktionsbeskrivning. För att uppnå detta kommer flödet av de skärmbilder som presenteras i Windows Mobile 5-applikationen att inspekteras. Detta ger en översiktlig bild över vilka funktioner som kommer behövas för att golfapplikationen ska fungera så funktionellt som möjligt.

Genom att sedan undersöka källkoden hos den applikationen ser vi flödet av data och mer konkret realisering av funktioner för att ge uppslag till idéer på lösningar för våra versioner av applikationen.

Även den existerande webbtjänsten kommer att undersökas. Detta för att få en insikt i dess funktionalitet, vilka metoder som finns att tillgå för data och för att se

om den kommer att kunna användas för de applikationer som ska byggas för Android och Iphone.

Nackdelen med denna metod är att det inte utformas någon konkret kravspecifikation genom dialog med uppdragsgivaren och tänkta användare, utan dessa krav analyseras ur utvecklarnas egna perspektiv och kanske kan missuppfattas. Intervjuer, av personer som tidigare har använt applikationen, är en alternativ metod för denna fas, men vi anser att analys av den existera applikationen är adekvat för detta relativt snäva arbete.

Denna analys kommer att presenteras i avsnitt 5.1. Detta avsnitt kommer även att innehålla identifierade krav på Android- och Iphoneapplikationerna baserade på denna analys samt uppdragsgivarens mål.

2.2.2. Fas 2 – Fastställa förutsättningar för Android och Iphone

Litteratur kommer i denna fas att undersökas, för att ge en grund att stå på inför utveckling av applikationen. Litteraturstudierna innebär för Androidplattformen inläsning av dokumentation från Android Developers (Android, 2010), samt utvecklarforum och bloggar med exempel på Androidprogram. Exempel på sådana sidor är Voxisland¹ och Anddev².

För inläring av Iphone har videor från Stanford University via Itunes använts tillsammans med källkodsexempel från Iphone Developer Center³.

För att få inledande kunskap om hur man utvecklar för respektive plattform upplevs litteraturstudier som det mest passande att studera den litteratur som finns tillgänglig från Google respektive Apple. En nackdel med denna metod kan vara att kvalitén på materialet kan skilja sig, samt att den lösning man hittar sällan är konkret anpassad för det aktuella problemet.

Även i denna fas kan intervjuer användas. Sådana intervjuer hade kunnat göras med utvecklare som har tidigare erfarenhet av Android- respektive Iphone-utveckling. Vi undersökte denna möjlighet, men då det är få eller inga utvecklare

¹ <http://android.voxisland.com/>

² <http://www.anddev.org/>

³ <http://developer.apple.com/iphone/index.action>

med kunskaper om de plattformarna hos vår uppdragsgivare fann vi det naturligare att söka ut denna information själva.

Denna fas kommer att presenteras i två avsnitt för Android och Iphone respektive. Resultatet presenteras som en genomgång av punkter motsvarande de som formulerats i vår frågeställning i avsnitt 4.2 och 5.3 respektive.

2.2.3. Fas 3 - Utformning av applikation

Genom att i denna fas utveckla applikationerna kommer vi att få möjligheten att undersöka plattformarna i sin realistiska miljö. Detta gör att vi kommer att kunna placera vår problematik i en praktisk kontext och därmed lättare kunna få klarhet i begrepp och struktur för plattformarna. Fallstudien är ett sätt att få mer djupgående praktisk kunskap om plattformarna som inte skulle kunna uppnås med enbart litteraturstudier.

Vårt arbete innefattar komplexa frågeställningar, då vi kommer att utforska skillnader mellan plattformarna på flera olika punkter. Detta gör det passande att undersöka plattformarna i sin realistiska miljö, vilket gör att en fallstudie är en passande metod enligt definitionen av fallstudie enligt Backman (1998).

För vårt forskningsbidrag hade det eventuellt varit tillräckligt med en omfattande litteraturstudie. En fallstudie kan bli subjektiv, då företeelsen inspekteras ur endast ett perspektiv. Med enbart litteraturstudier hade ett bredare perspektiv och mer objektivt synsätt antagligen kunnat uppnås.

Men då uppdragsgivaren efterfrågar applikationerna är detta inte applicerbart i denna kontext.

Denna fas, samt fas 2, kommer att utföras separat av författarna; Mikaela utför undersökning av Android och utveckling av Androidapplikationen, Anders utför undersökning av Iphone och utveckling av Iphoneapplikationen. Att separera dessa två faser mellan utvecklarna beror till stor del på tidsbrist men även på författarnas personliga intresse. För att uppnå ett mer objektivt resultat skulle båda applikationerna kunna utvecklas av båda författarna, för att sedan jämföras. Detta är dock inte applicerbart i detta arbete då godtyckligt med tid inte finns för att båda författarna ska kunna införskaffa kunskaper om och utveckla båda plattformarna.

Resultatet av denna fas presenteras som en mer tekniskt vinklad beskrivning av den utvecklade golfapplikationen, tillsammans med större problem som påträffats under arbetet i avsnitt 5.4.

2.2.4. Fas 4 - Utvärdering av Android och Iphone

Med underlag av punkterna i fas 3 kommer en sammanställande utvärdering av de båda plattformarna presenteras i avsnitt 6. Här kommer våra samlade erfarenheter bidra till en utvärdering av plattformarna. Här ställs punkterna som vi tidigare formulerat i frågeställningarna mot varandra och vi försöker så objektivt som möjligt finna fördelar respektive nackdelar med de olika plattformarna.

På grund av vår bakgrund i Java-programmering kan vårt perspektiv i utvärderingen bidra till en subjektiv bedömning av ämnet. Detta då bredare programmeringskunskap hade kunnat ge en mer objektiv infallsvinkel på utvärderingen. Det perspektiv vi har som utvecklare kan leda till ett gott underlag för andra utvecklare som står inför ett liknande val.

3. CENTRALA BEGREPP

I detta avsnitt kommer begrepp klargöras för att läsaren ska förstå de begrepp som står mest centralt för detta arbete. De mest centrala begreppen i denna uppsats är förstås Android och Iphone och företeelser rörande dessa plattformar, såsom hur smartphones, Java, och Objective-C. Även en övergripande förklaring av webbtjänster kommer att presenteras.

3.1. Smartphone

Exakt vad en smartphone är, är svårdefinierat. Dock är det en mobiltelefon som är något mer likt en PC (Fling, 2009). De använder operativsystem såsom Android, Iphone OS, Symbian eller Windows Phone (tidigare Windows Mobile).

Förutom de vanliga möjligheterna att ringa, skicka textmeddelanden, och surfa finns vissa egenskaper som anses vara de facto standard idag. Några av dessa egenskaper är tryckkänslig skärm, QWERTY-tangentbord¹ (Fling, 2009) och GPS. De nyare telefonerna har processorer på mellan 600 MHz – 1 GHz (HTC, 2010), (Apple Inc., 2010), vilket tillsammans med det allt större arbetsminnet ger stora möjligheter att skapa avancerade program med lättanvända och intuitiva användargränssnitt.

De flesta telefonerna idag har även stöd för uppkoppling mot trådlöst nätverk, WLAN, vilket ger tillgång hög bandbredd och detta skapar helt nya möjligheter i form av strömmande video och ljud till telefonerna.

3.2. Java

Java är ett objektorienterat programmeringsspråk och plattform släppt av Sun Microsystems. Program skrivna i Java körs genom Javas virtuella maskin, JVM. Java används i kommersiella sammanhang ledfrasen *Write once, run everywhere*, med faktumet att det enda som krävs är JVM för att köra programmen, oavsett vilken plattform som ligger i botten. (Oracle Corporation, 2010)

¹ Tangentbord med den vanligaste placeringen av bokstäver, dvs. att de första fem tangenterna på den översta raden av bokstäver är QWERTY. Detta inte nödvändigtvis ett fysiskt tangentbord, utan kan även existera som mjukvara i telefonen.

3.3. Android

Android är mjukvara för mobiltelefoner, i form av operativsystem, middleware (en mjukvarukomponent som ligger mellan applikationerna som tillåter dessa att kommunicera och dela data), samt en bas av grundläggande applikationer som finns tillgängliga vid installation. (Open Handset Alliance, 2009). Android är baserad på öppen källkod, släppt under Apachelicensen. Det är baserat på operativsystemet Linux och skrivet i en variant av Java. Android underhålls av Open Handset Alliance, en allians för utveckling av öppna standarder, som uppkom med Google i ledningen (Open Handset Alliance, 2007).

Telefoner med Androidoperativsystemet har tillgång till verktyg som virtuellt tangentbord och en rad applikationer. Däribland hemskaerm för s.k. widgets¹ och genvägar, vanliga telefonfunktioner, webbläsare, mediaspelare, e-post klient, Android Market för nedladdning och uppdatering av program m.m. Bibliotek som Webkit används för webbläsaren och OpenGL ES 1.0 för 2D och 3D grafik. Stöd för Wide VGA (skärmar med 800x480 eller 854x480 pixlars upplösning) finns (Android Open Source Project, 2008). Trådlöst lokalt nätverk och andra trådlösa tekniker såsom HSPA² och 3G gör det enkelt att använda olika internetbaserade tjänster i telefonen. Synkronisering av e-post, kalender och telefonkontakter kan göras via Google eller Microsoft Exchange Activesync. Sociala medier såsom Facebook är integrerat i operativsystemet.

I skrivande stund är den senaste Androidversionen version 2.1, en mindre uppdatering efter version 2.0. Versionerna 2.0 och 2.1 går även under namnet Eclair. Android 2.1 bygger på linuxkärnan 2.6.29 och en rad nya funktioner introducerades. Bland annat fick det grafiska användargränssnittet sig ett lyft, hårdvaran optimerades och en ny webbläsare med stöd för HTML 5 lades till. Vidare finns i denna version stöd för digital zoom och blixtn för fotografering. Det virtuella tangentbordet har också fått lite uppdateringar. SDK:n för version 2.0 släpptes i oktober 2009, följt av version 2.0.1 i december och slutligen 2.1 den 12 januari 2010. (Android, 2010).

¹ Små interaktiva applikationer för lätt åtkomst på t.ex. en telefons hemskaerm eller en dators skrivbord.

² Förstärkt 3G som stödjer hastigheter på upp till 14Mbit/s nedströms.

3.4. Objective-C

Objective-C är en objektorienterad utvidgning av programmeringsspråket C. Objective-C har lite annorlunda syntax gentemot de traditionella objektorienterade språken som baseras på C++ syntax (t.ex. Java). I Objective-C måste utvecklaren själv sköta minneshantering, dvs. allokera och avallokera minne för skapade objekt och variabler, precis som i C. I språk som t.ex. Java sköts det av en s.k. garbage collector, vilket är inbyggd funktionalitet för att frigöra minne.

3.5. Iphone OS

Iphone är en smartphone tillverkad av Apple. Operativsystemet som används på en Iphone är Iphone OS, just nu version 3.1.3. Hittills har det kommit tre versioner av iPhone, dessa är iPhone Original, iPhone 3G och iPhone 3GS. Alla tre har liknande hårdvara i form av multitouch-skärm¹, GPS, accelerometer och trådlöst lokalt nätverk. Iphone 3G och 3GS använder sig av 3G teknik för att ringa och överföra data när inget trådlöst nätverk finns i närheten. Iphone 3G har bättre prestanda än Iphone original, och Iphone 3GS har i sin tur bättre prestanda än Iphone 3G (Apple Inc., 2010).

Ipod Touch och Ipad är två andra enheter som också använder Iphone OS. Dessa har liknande egenskaper som en Iphone har. Ipod touch kan köra alla applikationer som Iphone kan, förutom de som kräver en mikrofon och GPS, då detta saknas på Ipod Touch (Apple Inc., 2010). Ipad kan även den köra de flesta applikationer som är utvecklade för till Iphone, men den har en fyra gånger så stor skärm vilket gör att de grafiska delarna i programmen inte ser lika tilltalande ut.

Program som utvecklas för Iphone OS blir, efter godkännande från Apple, tillgängliga för nedladdning via Apples ”App Store”.

3.6. MVC

MVC, som står för Model-View-Controller är ett designmönster för att dela upp en applikations uppbyggnad i olika segment, nämligen modell, vy och controller. Modellen är basen för logik och för att hantera data. Den ser till att data är aktuellt

¹ Tryckkänslig skärm som stödjer att användaren använder flera fingrar på skärmen för att få tillgång till olika funktioner som t.ex. inzoomning i en webbläsare.

och uppdaterat samt notifierar övriga komponenter om förändringar sker. Vyn presenterar det grafiska användargränssnittet så att användaren kan använda och interagera med applikationen. Controllern tar emot och hanterar input från användaren. Baserat på användarens input hämtas rätt data via modellen för att kunna presentera den informationen i vyn (Plakalović & Simić, 2010).

3.7. XML

eXtensible Markup Language är ett vitt använt metaspråk på dagens webb. Med metaspråk menas att XML är ett språk som ligger till grund för att ge struktur och regler för andra språk, t.ex. XHTML (Alesso & Smith, 2004). Det är ett simpelt, flexibelt textformat baserat på SGML (ISO 8879) (W3C, 2010). Dessa standarder har satts upp för att tillhandahålla grammatiska riktlinjer för hur XML-dokument måste vara uppbyggda.

Ett XML-dokument inleds av en Processing Instruction (PI), vilken talar om för en tolkare, även kallat parser, att detta är ett XML-dokument och vilken version av XML som används.

3.8. Webbtjänster

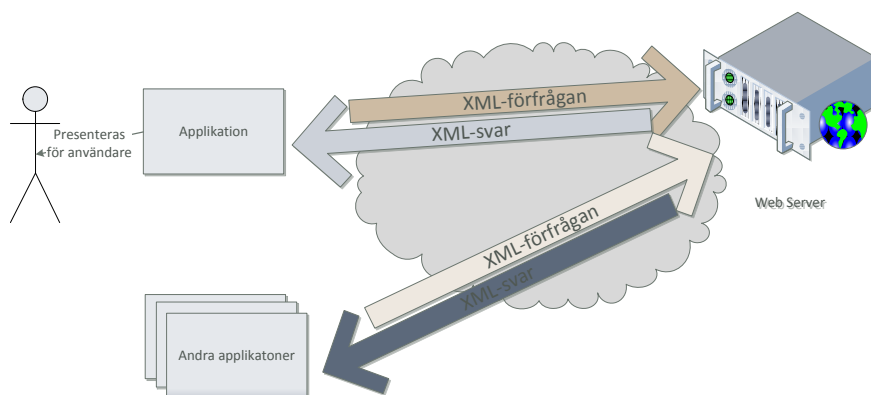
En webbtjänst identifieras genom att de är tillgängliga över internet, använder en strukturerad standard såsom t.ex. XML för meddelanden, inte är bundet till operativsystem och är publikt tillgänglig genom WSDL-specifikationer (Cerami, 2002). Alternativ till XML för att strukturera webbtjänstmeddelanden är t.ex. att använda JSON¹. Denna studie behandlar dock bara webbtjänster som använder XML och SOAP.

Webbtjänster bidrar till en applikationscentrerad webb, snarare än en mänskocentrerad webb, där användaren gör förfrågan till en server om vilken webbsida denne önskar se, och servern skickar tillbaka svaret som visas i användarens webbläsare. Med webbtjänster skickar applikationen en fråga, t.ex. med hjälp av ett XML-meddelande, till servern via HTTP-protokollet. Servern hanterar sedan frågan för att skapa upp och returnera ett svar till applikationen.

¹ JavaScript Object Notation, en lättviktsstandard för att representera enkla datastrukturer.

3. Centrala begrepp - Webbtjänster

Det är sedan applikationens uppgift att tyda meddelandet för presentation till användaren. (Cerami, 2002).



FIGUR 3-1: WEBBTJÄNST - APPLIKATIONS-CENTRERAD WEBB (CERAMI, 2002)

Det finns olika alternativ för strukturering av XML-meddelanden, t.ex. XML-RPC, REST och SOAP (Cerami, 2002), varav den sistnämnda kommer att beskrivas mer i detalj nedan.

Utvecklingen av webbtjänsten kan t.ex. ske i Java Enterprise Edition (Java EE), COBRA eller Microsofts .Net-plattform.

De flesta databaser såsom Oracle, SQL Server och DB2 stödjer XML parsning, vilket gör det enkelt att låta databaser interagera direkt med webbtjänster (Newcommer, 2002).

3.8.1. SOAP

SOAP är ett protokoll för att strukturera XML-dokument som kommuniceras via webbtjänster. Standarden utvecklas och övervakas idag av W3C (Jorgensen, 2002).

Soap används för att utbyta strukturerad information i en decentraliserad, distribuerad miljö (W3C, 2007). Även om SOAP kan användas för att utbyta information i flera olika meddelandesystem är huvudsyftet att transportera Remote Procedure Call (RPC) över HTTP-protokollet (Cerami, 2002). Förfrågan gör alltså via HTTP-request via HTTP-POST eller -GET och svaret kommer med ett HTTP-response (Jorgensen, 2002).

SOAP identifierar ett antal XML element som omsluter den data som kommuniceras via webbtjänsten.

4. MOBILUTVECKLING OCH OPERATIVSYSTEM

Nedan beskrivs den allmänna synen på utveckling för mobila applikationer. Vidare visas också en översikt på vilka operativsystem och plattformar som finns för mobiltelefoner just nu.

4.1. Bakgrund för utveckling av mobilapplikationer

Att utveckla för mobila enheter skiljer sig på många punkter från traditionell systemutveckling, då det alltid finns begränsningar att ta hänsyn till hos mobila enheter (Rahimian & Habibi, 2008).

Trådlös kommunikation har genom tiderna varit en utmaning för utvecklare att hantera. Jämfört med trådbunden kommunikation, är den lättpåverkad av omgivningen och mottaglig för störningar, vilket gör att hänsyn måste tas till lägre bandbredd, avbrott i uppkopplingen och stora variationer i bandbredden. (Forman & Zahorjan, 1994).

Detta gör att applikationer måste anpassas så att datakommunikation enbart sker när det är nödvändigt. Data måste alltid finnas tillgängligt, även när uppkopplingen är begränsad. Att använda mobila databaser för sin applikation är en lösning på detta då det möjliggör åtkomst till data utan nätverksuppkoppling, minimering av nätverkstraffik, vilket också bidrar till mindre processor- och batterianvändning. (Yu, Amjad, Goel, & Talawat, 2008).

Hårdvaruresurser är också en viktig del att ta hänsyn till vid utveckling. På grund av en mobiltelefons begränsade storlek och energitillgångar, måste liten display och låg processorhastighet tas i beaktning, jämfört med vid utveckling för stationära system. Även RAM-minne och lagringsutrymme är begränsat (Forman & Zahorjan, 1994), så i tillägg till att data måste var åtkomligt trots nätverksstörningar, kan inte heller för mycket data lagras i mobiltelefonen.

Generellt, vid utveckling av applikationer för mobiltelefoner, har portabilitet varit ett problem. De applikationer som utvecklas måste kunna fungera på olika typer av telefoner, med olika förutsättningar vad gäller minne, lagringsutrymme, operativsystem och skärmstorlek. Design av gemensamma metodologier för dessa portabilitetsproblem, liksom med de tidigare nämnda problemen, har länge varit i fokus. Rahiman och Ramsin (2008) presenterade en agil metodologi för utveckling

av mobila applikationer i deras artikel *Designing an Agile Methodology for Mobile Software Development: A Hybrid Method Engineering Approach*.

4.2. Utvecklingsplattformar och mobila operativsystem

Utöver Apples Iphone OS och Android, finns ett flertal plattformar för mobiltelefoner idag. Nedan följer en kort beskrivning av några av dem, samt en översikt över hur fördelningen ser ut mellan dem.

- **Java ME**

Javas mobila plattform, Java ME, har varit brett använd för utveckling av mobila applikationer. P.g.a. applikationernas förmåga att existera på olika mobiltelefoner av olika tillverkare och med olika operativsystem (Beji & Kadhi, 2008). I verkligheten har kanske inte Java ME-applikationer varit så plattformsoberoende som det påståtts, då det alltid funnits hårdvaruskillnader att ta ställning till (Braun & Eckhaus, 2008). Java ME-applikationer distribueras enkelt genom att tillhandahålla den jar-fil som skapas vid kompilering av applikationen.

- **Windows Mobile/Windows Phone**

Microsoft Windows Phone, tidigare kallat Windows CE och Windows Mobile, har varit länge varit ett operativsystem för smartphones. Windows Phone bygger på Microsoft .NET Compact Framework (Microsoft Corporation, 2010). Utveckling för detta operativsystem sker således Microsoft Visual Studio, med ett tillägg för Windows Phones SDK (Fertalj & Horvat, 2007). Tack vare .Net-plattformen och utvecklingsmiljön Visual Studio kommunikationen mellan webbtjänster, speciellt de utvecklade i .Net, och den mobila applikationen välimplementerad i ramverket.

- **Symbian**

Symbian OS är ett operativsystem för smartphones, skrivet i C++. Symbian OS finns, i skrivande stund ute i version 1. Denna version kallas Symbian^1 och motsvarar Nokias "gamla" S60-serie. Operativsystemet är övervakat och drivet av Symbian Foundation som är en icke vinstdrivande organisation, som uppkom efter Nokias köp av Symbian Ltd. 2008 (Symbian, 2010). Applikationerna för Symbian OS skrivs dock i Java ME och kan distribueras via jad-filer.

- **RIM Blackberry**

Kanadensiska företaget Research in Motion (RIM) är mest kända för att ha tillverkat Blackberry-telefonerna och dess operativsystem sedan 1999 (Research In Motion, 2010). Tredjepartsutvecklare kan utveckla applikationer för BlackBerry OS, genom att använda sig av dess API. Applikationer för Blackberry skrivs i Java (Research In Motion, 2010).

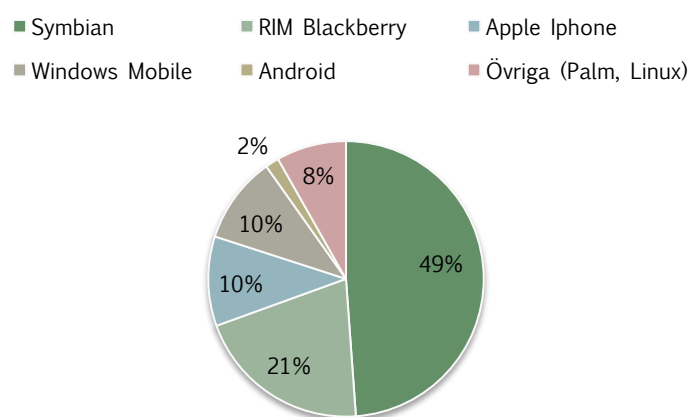
4.2.1. Fördelning av operativsystem hos smartphones

Att försäljningen av smartphones ständigt ökar visar statistik ifrån Gartner. Där presenteras att det såldes 314,653.50 smartphones under första kvartalet 2010 jämfört med 269,120.10 enheter första kvartalet 2009 (Gartner, 2010).

Fördelningen av försäljning och användning av olika mobila operativsystem förändras ständigt. Nedan visas två grafer för att ge en inblick i hur läget ser ut kring tiden då arbetet utfördes. Graferna är till för att visa både utveckling av försäljning av enheter med de olika operativsystemen, samt vilka operativsystem som används av telefoner som besökt olika webbplatser.

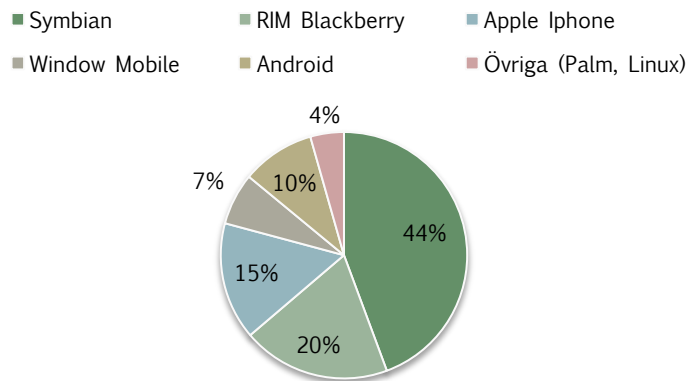
Från Gartner presenteras statistik över sålda enheter för första kvartalet 2009 respektive 2010.

Operativsystem för smartphones, sålda första kvartalet 2009



FIGUR 4-1: OPERATIVSYSTEM (FÖRSÄLJNING), GARTNER, 2010

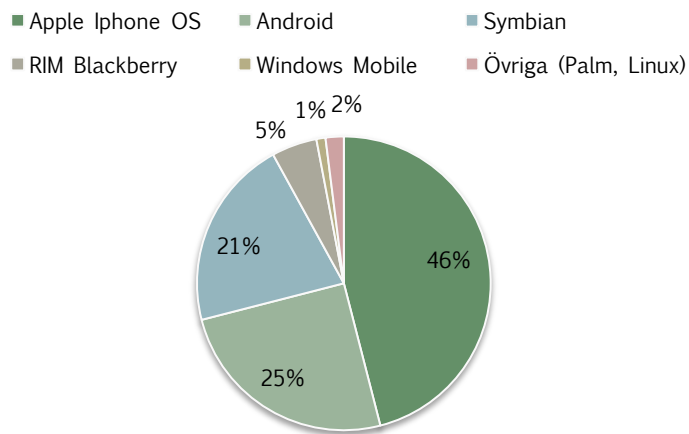
Operativsystem för smartphones, sålda första kvartalet 2010



FIGUR 4-2: OPERATIVSYSTEM (FÖRSÄLJNING), GARTNER, 2010

Admob, å andra sidan, som samlar in sin information genom webbplatser som besöks via användares smartphones visar följande statistik (Admob.com, 2010).

Användning av operativsystem för smartphones, mars 2010



FIGUR 4-3: OPERATIVSYSTEM (ANVÄNDNING), ADMOB, 2010

4.3. Användning av webbtjänster hos mobilapplikationer

Hos dagens mobiltelefoner, speciellt smartphones, finns ett behov att ha aktuell och dynamisk data tillgängligt för de applikationer som används. (Krejcar, Janckulik, & Motalova, 2010). Detta har lett till utbredning av webbtjänstekommunikation för att tillhandahålla data som applikationen just för tillfället behöver (Sheshagiri, Sadeh, & Gandon, 2004).

Genom utbredning av 3G-teknik och uppkopplingsmöjligheter via trådlöst Internet för mobiltelefoner (Krejcar, Janckulik, & Motalova, 2010), kan webbtjänster användas för att förse användaren med aktuell information. Att använda webbtjänster för datahantering hos mobila enheter är även ett sätt att komma runt problemen med begränsat lagringsutrymme hos mobiltelefoner (Sheshagiri, Sadeh, & Gandon, 2004), genom att data efterfrågas hos webbtjänsten när användaren behöver det, och i fallet med sammanhangsorienterade applikationer, beroende på det sammanhang som användaren befinner sig i.

5. RESULTAT

Detta kapitel beskriver resultat av de faser som beskrivs i avsnitt 2.2. Analys av den existerande applikationen presenteras tillsammans med de identifierade krav för den applikation som ska byggas. Vidare beskrivs underlag för Android och Iphone som används till stöd för själva realiserandet av de båda applikationerna, vilken är beskriven i avsnitt 5.4. Detta resultat har tagits fram för att vi ska få en förståelse för hur applikationer utvecklas hos plattformarna så att ett underlag för val av plattform kan presenteras.

5.1. Analys av existerande mobilapplikation

För att identifiera krav på den applikation som ska byggas har en analys gjorts av den existerande mobilapplikationen. Detta avsnitt presenterar den grafiska utformningen och hur data kommuniceras hos den redan existerande versionen av golfapplikationen. Denna utformning, tillsammans med uppdragsgivarens mål (se avsnitt 1.4) ligger till grund för att kunna presentera generella krav för de applikationerna som ska utvecklas för Android och Iphone.

5.1.1. Flöde och funktioner

Den existerande mobilapplikationen är uppbyggd med hjälp av tabbar som presenterar olika delar av applikationen när en boll väl är vald för inrapportering av resultat. För att välja en boll används ett menyval där användaren sedan får uppge önskad tävling och boll som denne vill hämta. De olika tabbarna visar information om golfbanan, inrapportering av resultat, möjlighet för analys av slag, individuellt poängkort, leaderboard samt fakta om tävlingen.

Detta flöde har resulterat i att följande funktioner har identifierats:

- Val av tävling
- Val av boll
- Exportering av boll till enhet
- Översikt över boll och tävlingsinformation
- Inrapportering av resultat
- Visning av poängkort och leaderboard.

5.1.2. Webbtjänsten

För att tillhandahålla data till den mobila enheten finns en webbtjänst som kommunikationsgränssnitt. Denna webbtjänst får sin information från en databas knuten till en webbapplikation. I denna webbapplikation registrerar användaren deltagare, tävlingar, bollar och kan se resultat m.m. Webbtjänsten tillhandahåller metoder för att t.ex. leverera de registrerade tävlingar som ligger framåt i tiden, att exportera en specifik boll för en tävling till den mobila enheten, rapportera in resultat och hämta leaderboard.

Tävlingar måste således registreras i webbapplikationen av en användare innan den kan användas och spelas i den mobila enheten.

5.1.3. Datakommunikation

Data kommuniceras till och från webbtjänsten med hjälp av s.k. datasets, vilket gör det möjligt för webbapplikationen att använda webbtjänstens resultat direkt för presentation av data i applikationen.

Dessa dataset är objekt som kan kommuniceras via webbtjänsten. Genom att ange en referens till webbtjänsten som ska användas, kan webbtjänsten initieras som ett objekt vars metoder kan tillgås med vanlig punktnotation. De objekt som webbtjänstens metoder returnerar är dessa dataset som kan användas för att hämta ut data som webbtjänsten förmedlar, med bara några få rader kod.

I grund och botten är det trots allt ren XML som förmedlas via webbtjänsten, men dess struktur och vissa specifika taggar och attribut bidrar till att .Net-ramverket kan känna igen svaren som dataset.

Vidare syns att applikationen inte lagrar någon data på telefonen utan att all data inhämtas och skickas via webbtjänsten.

5.1.4. Identifierade krav baserad på tidigare funktionalitet

Baserat på de funktioner som identifierats hos den tidigare applikationen samt de mål som angetts av arbetsgivaren har följande slutsatser dragits rörande krav på applikationerna.

- **Applikationen ska ha tydliga komponenter och ett linjärt flöde**
Användargränssnittet kan inte ha en tabbstruktur då Windows Mobile 5-applikationens gränssnitt är allt för litet och oanvändbart för en applikation

som ska vara anpassad för utomhusmiljö, detta gör att vi måste bygga ett användargränssnitt som inte använder tabbar.

- **Data ska inte sparas lokalt på telefonen**

Samtlig data ska kommuniceras via den webbtjänst som finns och inte sparas ner i t.ex. en lokal databas.

5.2. Förutsättningar för Android

Detta avsnitt beskriver vilka förutsättningar Androidplattformen har för att utveckla uppdragsgivarens önskade applikation och erbjuda ett underlag för framtida val av plattform för liknande applikation. Dessa förutsättningar, tillsammans med förutsättningarna för Iphone som presenteras i nästa avsnitt, samt erfarenhet av utveckling av själva golfapplikationen ligger till grund för det underlag för val av plattform som erbjuds av denna studie.

5.2.1. Utvecklingsmiljö

För att utveckla applikationer för Android krävs kunskaper i Java och XML. Java ligger till grunden för logiken i applikationen medan XML används för att bygga upp det grafiska användargränssnittet.

Det finns ett stort stöd för att lära sig grunderna i Androidutveckling. Mycket hjälp finns att få från sidan Android Developers och den exempelkod som följer med SDK:n. Vidare finns en rad diskussionsforum där frågor rörande utvecklingen kan diskuteras eller bloggar med olika guider som kan vara till hjälp.

Grundläggande applikationsstruktur

Designmönstret Model-View-Controller används för att konstruera applikationer. XML-filer anger hur det grafiska användargränssnittet och dessa XML-filer kopplas till var sin aktivitet. Aktiviteten är ”controller”-delen i MVC och är en applikationskomponent som ansvarar för presentation av vyn och för att hantera användarens input. För varje skärmbild i applikationen används en ny controller.

Aktiviteten är den bakomliggande Javaklass som ansvarar för att hantera användarens input, samt för att hämta och förse XML-vyerna med rätt data. Varje skärmbild i en applikation motsvarar en egen aktivitet. För att starta nya aktiviteter, samt för att kommunicera data mellan aktiviteter används ett meddelandeobjekt som kallas intent.

Utvecklingsverktyg

Android SDK krävs för att börja utveckla. Det är en samling av Androidramverket samt verktyg för att kunna utveckla för plattformen. Med SDK:n följer en Androidtelefonemulator, s.k. AVD med, så att applikationerna kan testas. Vidare följer även Dalvik Debug Monitor (DDMS) med. Denna används för att kontrollera emulatorens. Där är det möjligt att se vilka processer som körs i emulatorens, utföra debug, se output från applikationen, samt interagera med den t.ex. skicka sms, utföra ett samtal till den eller uppdatera GPS-koordinater för att simulera hur någon rör sig med telefonen.

Det är inte ett krav att använda en IDE för att utveckla applikationerna. Det går att köra emulatorens via en kommandotolk och skriva koden i önskad textredigerare. En IDE som rekommenderas av Android Developers är Eclipse. För att använda Eclipse behövs tillägget Android Developer Tools (ADT). Med tillägget får utvecklaren tillgång till emulatorer och DDMS genom Eclipses utvecklingsmiljö.

För att utveckla för Androidplattformen krävs Windows (XP 32-bit eller Vista/7 32- eller 64-bit), Mac OS X (10.5.8 eller senare, med x86-arkitektur) eller Linux. Vidare krävs att Java Developer Kit (JDK) version 5 eller senare är installerat på datorn (Android, 2010).

5.2.2. Användargränssnitt och språkstöd

XML-filen, som används tack vare att Androidramverket använder sig av MVC-mönstret vid uppbyggnad av applikationer, innehåller struktur för de grafiska komponenter som användaren kan se och interagera med.

I XML-dokumentet, struktureras hur grafiska komponenter, dvs. knappar, text, bilder etc., ska placeras ut på skärmen. Placering och utseende på dessa komponenter styrs av dess omslutande layoutelement, samt olika attribut som finns att tillgå.

Det går att använda stilar och teman för att bestämma utseende på grafiska komponenter. Stilar kan liknas vid Cascading Style Sheets för XHTML, för ett enhetligt utseende för en samling komponenter. Teman är mer övergripande för hela applikationen, t.ex. som att bakgrunden i samtliga fönster ska visa en viss bild.

En vanligt förekommande grafisk komponent är listvyn, som används för att presentera en lista av objekt på skärmen. Denna lista kan populeras i Javakoden

med hjälp av s.k. Adapterklasser som mappar olika grafiska element mot data från t.ex. en databas eller någon annan samling.

Effekter

Olika grafiska effekter, som kan bidra till ett mer tilltalande gränssnitt, finns till viss del implementerat i ramverket. Vissa effekter, såsom en liten animation när en ny aktivitet startas, finns inbyggt i gränssnittet utan att utvecklaren behöver ange att så ska hända. Dessa inbyggda effekter är dock begränsade och de flesta animationerna får utvecklaren själv bygga ihop och implementera i koden för sina aktiviteter.

Resursfiler för olika hårdvaruförutsättningar och språk

Då alla Androidtelefoner inte delar identiska hårdvaruförutsättningar och språkställningar, finns resursfiler och mappstruktur för att ange olika alternativ för hur applikationen ska renderas vid kompilering. Resursfilerna är också uppbyggda i XML och kan innehålla t.ex. strängar som representerar den text som syns i applikationen.

Om telefonen har en högupplöst skärm används bilder i mappen för högupplöst grafik. Vidare, om telefonens språk är inställt på svenska används en resursfil med strängar placerad i en mapp som representerar visning av svenskt språk i applikationen.

5.2.3. Distribution och test

För att göra sin Androidapplikation tillgänglig för nedladdning via Android Market, måste Androidutvecklare registrera sig hos Android Market. En engångskostnad på \$25 betalas då och därefter kan utvecklaren göra ett obegränsat antal applikationer tillgängliga via Android Market. (Android, 2010).

Om distribuering via Android Market inte är något som önskas, skapas alltid en installationsfil vid exekvering av applikationen. Denna fil kan laddas ner eller överföras till Androidtelefoner så att programmet kan installeras. För att kunna installera applikationen krävs att telefonen har den version av operativsystemet, som krävs för att använda applikationen.

Emulatorn, som ingår i Android SDK, kan konfigureras med olika förutsättningar, såsom skärmstorlek, uppkopplingshastighet, språk, olika versioner av Androidoperativsystemet, etc. Detta är nödvändigt för att Androidtelefoner kan ha olika

förutsättningar på flera punkter, då det finns flera olika tillverkare som producerar telefonerna. Dessa förutsättningar är knutna till vilken resursfil eller resursmapp som kommer att användas för att visa grafik eller presentera text.

För att testa en applikation på en fysisk enhet, går det bra att koppla in kompatibel telefon och ange att applikationen ska testas i den inkopplade telefonen, istället för i emulatorn.

5.2.4. Webbtjänstkommunikation

Att använda SOAP-baserade webbtjänster stöds idag inte av Androids grund-API. Tredjepartsbiblioteket Ksoap2-android, finns dock att tillgå.

Detta bibliotek ger tillgång till grundläggande kommunikation med en webbtjänst via SOAP-protokollet. Frågan ställs till webbtjänsten genom ett SOAP-objekt och svaret returneras på samma sätt. Det är sedan upp till utvecklaren att ta hand om det SOAP-objekt som kommer tillbaka från webbtjänsten för att tolka och hantera det så att det kan användas i applikationen.

5.3. Förutsättningar för Iphone

De ovanstående punkterna, som beskrevs för Androidramverket och som formulerades i frågeställningen, presenteras här för Iphone för att belysa förutsättningarna som plattformen har för utveckling av golfapplikationen.

5.3.1. Utvecklingsmiljö

För att överhuvudtaget kunna börja utveckla till Iphone behövs en Intelbaserad Macintosh med *Snow Leopard*-versionen av operativsystemet Mac OS X (Apple Inc., 2010)

Objective-C används som utvecklingsspråk för Iphoneapplikationer. Eftersom Objective-C är en utvidgning av C går det även bra att skriva applikationer för Iphone i C.

Grundläggande applikationsstruktur

Alla komponenter i Iphones ramverk använder Model-View-Controller (MVC) designmönstret. Detta för att ge en enhetlig uppbyggnad för samtliga komponenter. Det finns några grundläggande begrepp inom Objective-C som krävs för att förstå utveckling för Iphone.

- **Delegates**

Delegates eller lyssnare används genomgående i Iphoneutvecklingen. Dessa används för att en komponent ska veta vilken annan komponent som skall ha data som skapats. Ett tydligt exempel är *UITableViewDelegate* som används som datakälla till de individuella cellerna som skall visas i en tabellvy.

- **Selectors**

Selektorer används som pekare till metoder. Eftersom Objective-C är ett löst typat språk kan ett användningsområde vara att kolla om ett objekt är av en specifik typ och i så fall köra metoden.

Stöd för utvecklare

För att underlätta utvecklarnas arbete finns till Iphone väldokumenterade API:er som i detalj beskriver vilka metoder som finns att tillgå. Även kodexempel som visar hur i princip alla de olika funktionerna som telefonen har kan användas. Kodexemplen används också för att visa olika animationer och annat som kan användas i applikationen. Det finns ett officiellt utvecklarforum med väldigt högt deltagande.

Utvecklingsmiljö

För att kunna utveckla till Iphone behövs Iphone SDK, denna består av ett antal verktyg som är till för att hjälpa utvecklaren, dessa består av:

- **Xcode**

Det program, som används för att skriva själva koden, innehåller de vanliga komponenter som brukar finnas i moderna IDE:er. T.ex. intellisense¹, lättåtkomlig dokumentation och hjälp att hitta syntaxfel.

- **Interface Builder**

För att bygga upp grafiska komponenter och vyer som används för att visa data för användaren används Interface Builder (IB). IB används även för att knyta samman grafiska komponenter med den kod som skrivits i Xcode. Det går även att bygga grafiska komponenter programmatiskt med Xcode, detta täcks emellertid inte av denna studie. IB är ett WYSIWYG²-verktyg där det

¹ Automatisk komplettering av kod.

² What You See Is What You Get, redigerare där GUI-uppbyggnad kan ske genom att dra och släppa komponenter och se resultatet i realtid

går att bygga upp användargränssnittet med hjälp av att dra och släppa komponenter.

- **Hjälpprogramvaror**

Det följer även med ett antal hjälpprogramvaror som underlättar att hitta minnesläckor och andra buggar. Dessa kommer inte diskuteras i detalj.

- **Iphone emulator**

En emulator som möjliggör att testköra programmet utan att behöva installera det på en fysisk enhet.

5.3.2. Användargränssnitt och språkstöd

Det grafiska användargränssnittet byggs upp med hjälp av Interface Builder, som nämns i ovanstående stycke. För att bygga upp användargränssnittet i Iphone används så kallade nib-filer. I dessa definieras vilka komponenter som skall synas på skärmen och vilka "outlets" eller "actions" de har. En action är en definierad metod som skall köras i en specifik klass när användaren t.ex. trycker på en knapp eller annan GUI komponent. En outlet är en grafisk komponent som reagerar på ett specifikt sätt från programmets kod; exempelvis kan programmet genom en outlet ändra den text som står i ett textfält i användargränssnittet.

För att bygga en applikation som stödjer flera språk användes dels en resursfil där den översatta texten med en förklaring ligger. Nib-filerna delas även upp i kataloger med namnet xx.lproj där xx ersätts med en standardiserad landskod, t.ex. sv eller en. Denna uppdelning görs p.g.a. att utseendet kan skilja i applikationen beroende på vilket språk som är valt.

De grafiska komponenterna använder, precis som andra delar av ramverket, MVC som designmönster. Detta gör att strukturen på programmeringen ser lika ut genom hela utvecklingsfasen. Ett exempel är listkontrollen, denna används för visa ett antal celler på skärmen, dessa celler är helt anpassningsbara. För att visa celler definieras en delegat och en datakälla som håller reda på hur många celler som skall kunna synas på skärmen. Dessa celler skapas dynamiskt upp av datakällan precis när den behövs och på detta sätt sparas systemresurser.

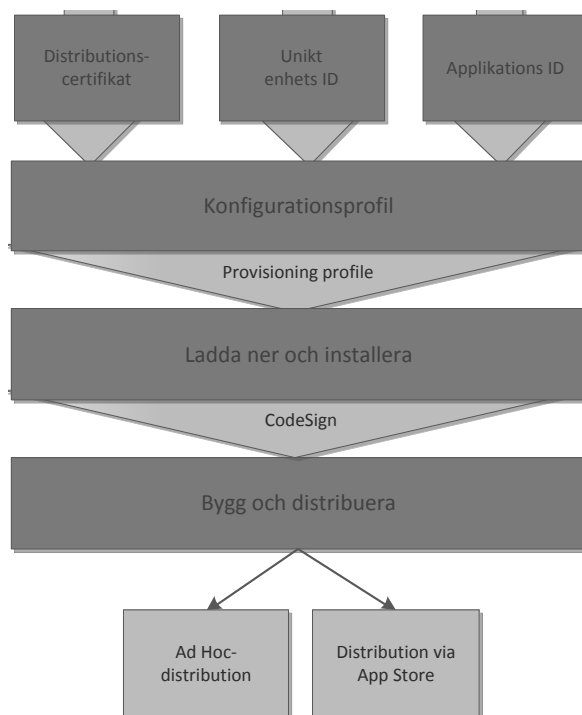
Eftersom iPhone OS till Iphone endast finns till en sorts telefon så vet utvecklaren exakt vilken skärmstorlek och vilka fysiska knappar som finns på enheten. Detta gör det väldigt lätt att göra det grafiska användargränssnittet då de grafiska komponenterna aldrig behöver skalas om på telefonen.

Det finns metoder för att animera övergångar mellan t.ex. vyer eller för att byta bild. Dessa effekter är hårdvaruaccelererade med hjälp av telefonens grafikkrets och bidrar ett bättre flyt i animationer än när enbart den vanliga processorn används.

5.3.3. Distribution och test

För att distribuera programmet till slutkund finns tre vägar att gå. För att distribuera till konsumenter går utvecklaren med i ett Apple Developer Program, detta kostar \$99/år och möjliggör distribution av ett oändligt antal applikationer via Apples App Store. För större företag finns möjlighet till att distribuera in-house applikationer, detta kostar \$299/år och möjliggör endast lokal spridning av programvaran.

Vidare kan en så kallad ad-hoc distribution användas. I denna läggs varje enskild enhets unika ID och ett applikations ID som sedan byggs in i ett distributionspaket som kan skickas till respektive användare. Användaren lägger sedan in distributionspaketet tillsammans med en "provisioning profile" in i telefonen genom iTunes. Det är inte möjligt att bygga en generell installationsfil som kan installeras på valfri telefon. Detta fungerar endast för 100 enheter. Bilden nedan, inspirerad av pdf-dokumentet Iphone Developer Program (Apple Inc., 2010), visar hur ad-hocdistribution fungerar. Även det kostar \$299/år. Detta kräver däremot att enheten, som är det är tänkt att applikationen ska installeras på, registreras via Apples system.



FIGUR 5-1: AD-HOC-DISTRIBUTION AV IPHONEAPPLIKATION

Även vid testning av applikationer, måste testtelefonen i fråga vara registrerad hos Apple och inneha ett certifikat som tillåter installation av applikationen som utvecklas.

Om utvecklaren inte önskar att gå den vägen för att förbereda sin telefon för testning, kan denne förlita sig på emulatorn för testning istället. Den saknar dock inbyggt stöd för att simulera olika förutsättningar, såsom GPS-position och olika uppkopplingshastigheter.

5.3.4. Webbtjänstkommunikation

Som nämnts tidigare behöver golfapplikationen kommunicera med en webbtjänst för att fungera. Stöd för detta finns inte i Iphone OS grundläggande bibliotek, den funktionaliteten måste byggas manuellt. Det finns ett simpelt bibliotek som kan användas för att underlätta kommunikationen med SOAP tjänsten, men detta bibliotek kräver en hel del manuell handpåläggning när informationen från tjänsten ska läsas ut.

5.4. Golfapplikationen

Den applikation som har utvecklats för uppdragsgivaren i detta arbete resulterade i en golfapplikation. Den generella funktionaliteten för hur applikationen fungerar är gemensamt för både Android- och Iphoneapplikationen och redogörs nedan. Vidare beskrivs kort hur implementeringen utförts och hur integreringen med webbtjänsten hanterats.

5.4.1. Design av applikationerna

För att tillhandahålla de kärnfunktioner som identifierades genom analys av den tidigare mobilapplikationen har ett linjärt applikationsflöde utvecklats där varje funktion presenteras på en vy som vid val leder till nästa vy. För att hämta information kring en boll till enheten väljs i första vyn en tävling, som leder till val av boll för den tävlingen, o.s.v.

Applikationerna är i huvudsak uppbyggd av listor, t.ex. för att visa tillgängliga tävlingar, leaderboard och individuella poängresultat. Dessa listor fylls på med hjälp av anrop till passande metoder hos webbtjänsten. Varje funktion i applikationen har en motsvarande metod hos webbtjänsten för att få rätt data.

Kärnan i applikationen består av att se en listöversikt av hål som kan väljas för inrapportering av resultat. För att undvika många menyval och göra navigeringen i applikationen så enkel som möjligt, utnyttjar vi funktioner som att visa olika vyer beroende på om användaren håller telefonen i landskaps- eller porträttläge. Detta innebär att inrapportering av resultat och översikt över hål presenteras i porträttläge, medan leaderboard och individuella poängkort presenteras i landskapsläge.

Gemensamt GUI

För att underlätta för användare, oavsett vilken plattform de använder, har vi kommit överens om ett gemensamt GUI för applikationen. Detta innebär att vyerna på de olika sidorna har liknande utformning och innehåller samma information.

5.4.2. Problem med den existerande webbtjänsten

Den webbtjänst som fanns från början för att mobila enheter skulle kunna kommunicera med den webbaserade applikationen visade sig inte riktigt fungera bra i Android- och Iphoneplattformarna. Strukturen på svaren och förfrågningarna

krävde ibland att s.k. datasetobjekt kommunicerades. Dataset är något som används i .Net-ramverket, och var svårt att implementera.

Detta resulterade i att en omstrukturering av den existerande webbtjänsten var nödvändig innan kommunikation kunde förekomma fritt mellan Android-applikationen och webbtjänsten. Efter ombyggnad gick det bra att för Android-applikationen att kommunicera med webbapplikationen.

Vi själva utvecklade inte de förändringarna hos webbtjänsten som krävdes, men vi efterfrågade att det skulle vara rena XML-strängar som kommunicerades, för att underlätta parsning hos de mobila enheterna.

I och med att webbtjänsten byggdes om, kunde metoder skraddarsys för att passa de mobila applikationernas vyer och visning av data.

5.4.3. Realisering av applikationerna

Följande stycke ger en lite mer ingående beskrivning av hur applikationen har utvecklats och hur funktionerna för webbtjänstkommunikation har utvecklats.

Android

Applikationen har utvecklats i lager, där kommunikation med webbtjänsten sker i bottenlagret. Däremellan ligger en klass som hanterar förfrågningar mot webbtjänsten från aktiviteterna och sedan tyder webbtjänstens svar för att tillhandahålla objekt med data från webbtjänsten till Aktiviteterna. Lagret ovanför webbtjänsthanteraren är därmed aktiviteterna, som innehar affärslogik och ser till att data från webbtjänsten är tillgänglig för XML-filernas grafiska element.

All kommunikation med webbtjänsten sker i trådar separerade från GUI-trådarna, för att gränssnittet inte ska frysa vid belastningen.

För kommunikationen med webbtjänsten används biblioteket Ksoap2-android. Detta ger hjälpklasser och metoder för uppbyggnad av SOAP-förfrågningar och för att skicka iväg dem till önskad webbtjänst. Biblioteket tillhandahåller en grundklass, kallad SoapObject, som är en direkt avbildning av webbtjänstens XML-svar. Ur detta det är möjligt att hämta ut önskad data genom att använda XML-taggens namn.

När data hämtats upp från webbtjänsten till aktiviteterna, är det aktiviteternas ansvar att koppla data till motsvarande GUI-komponent i listor. Detta sköts till

största del av så kallade ListAdapters, som finns i Androidramverket. Dessa erbjuder sätt för utvecklaren att knyta värden i ett samlingsobjekt direkt till textelement hos listor i den XML-fil som representerar GUI:t. När varje listelement skulle ha mer komplexa komponenter, t.ex. knappar, som är fallet för inrapportering av resultat, måste egna ListAdapters byggas för att kunna knyta data och knapplyssnare till varje listelement.

Iphone

För att kommunicera med webbtjänsten används en hjälpklass som sköter all kommunikation, det finns i Iphone OS inget stöd för SOAP vilket medför att utvecklaren själv måste bygga upp en SOAP förfrågan med SOAP-formaterad XML-kod och skicka den med HTTP-POST. För att tolka det data som kommer från webbtjänsten används Iphones XML-API libxml2. Detta API hanterar inte kommunikationen med webbtjänsten, den funktionaliteten får utvecklaren själv implementera. I vårt fall användes ett tredjepartsbibliotek som hjälpte till med strukturering av SOAP-förfrågningarna.

Data som hämtas presenteras i listvyer med anpassade celler. Dessa celler kopplas ihop med data i Interface Builder. Detta görs genom att datakopplingar dras till specifik etikett med hjälp av musen, vilket gör det möjligt att under utvecklingen se hur användargränssnittet kommer att se ut.

5.4.4. Distribution av golfapplikationen

Vid kompilering av Androidapplikationen skapas en installationsfil som kommer att distribueras via den webbaserade golfapplikationen för nedladdning och installation hos en kompatibel Androidtelefon. Denna installationsfil kommer att användas vid distribution av Androidversionen av golfapplikationen.

För att kunna distribuera applikationen till Iphoneanvändare har vi valt att använda oss av en så kallad ad-hoc-lösning (se avsnitt 5.3.3). Att distribuera publikt via Apples Appstore är inte ett alternativ i denna kontext, då golfapplikationen endast är ämnad för intern distribution och inte är av allmänintresse. Därmed kommer den inte att godkännas av Apple för offentlig distribuering via Apple Appstore.

6. UTVÄRDERING AV ANDROID OCH IPHONE

Nedan beskrivs författarnas analys och åsikter kring hur plattformarna lämpar sig för att utveckla en applikation av den typ som golfapplikationen är.

6.1. Inläring av utvecklingsplattform

I och med att vi kommer ifrån Java- och .Net-miljö, upplevdes steget till Androids utvecklingsplattform inte så stort. Det enda som kändes obekant var användning av XML-filer för att presentera GUI och bekanta sig med hur de olika funktionerna, såsom växling mellan vyer hanterades.

Den initiala tröskeln för att lära sig Objective-C, var betydligt större och mer omfattande. Dels för att syntaxen i Objective-C är så annorlunda jämfört med t.ex. Java, men även för att minneshanteringen måste skötas av utvecklaren. När syntaxen väl är inlärd, upplevs dock språket som behagligt att arbeta med.

Både Android- och Iphoneramverkern och dess API är välstrukturerade och väldokumenterade. Det finns gott stöd för de funktionerna som efterfrågades, med undantag för webbtjänstkommunikationsstandarden SOAP, som vi kommer beröra mer i detalj några stycken ned.

6.1.1. Utvecklingsmiljö

Även om det är möjligt att utveckla Androidapplikationer utan att använda någon speciell utvecklingsmiljö (se avsnitt 5.2.1), användes Eclipse i detta arbete. IDE:n i sig känns lite ofärdig t.ex. skulle intellisense kunna stödjas bättre. Däremot är integreringen av ADT (Android Developer Toolkit) enkel. Det är väldigt enkelt att använda sig av emulatorens vid test av applikationerna och goda möjligheter för debug och felsökning finns. Då Eclipse är en generell utvecklingsmiljö som används för javaprogrammering i allmänhet är det en bekant utvecklingsmiljö för många javaprogrammerare.

Initialt känns lösningen med att bygga grafiska komponenter (Interface Builder) och skriva kodimplementeringen i ett annat (Xcode) överdrivet krångligt. Eftersom det inte liknar någon annan utvecklingsmiljö vi har stött på (se avsnitt 5.3.1). Men när utvecklingsmiljön blivit mer bekant, går det väldigt smidigt att utveckla med hjälp av Xcode och Interface Builder. Intellisense funktionen i Xcode kräver god förståelse för hur utvecklingsstrukturen är uppbyggd.

Interface Builder är väldigt lätt att förstå och har ett väldigt intuitivt användargränssnitt.

Androidapplikationer går att utveckla på i stort sett valfri plattform, medan Iphoneutveckling är bundet till Macintoshdatorer. Detta gör Androidplattformen mer flexibel på denna punkt.

6.2. Användargränssnitt och språkstöd

Vi upplever att båda plattformarna har välutvecklade komponenter för att bygga GUI. Tack vare Iphones användning av hårdvaruacceleration vid generering av GUI, flyter animationer och effekter på bättre i Iphone än hos Androidapplikationer. Det upplevs även enklare att skapa dessa animationer för Iphone än för Android.

Hanteringen av GUI-utformningen hos Android hade kunnat vara bättre. Då komponenter placeras ut via XML hade det varit önskvärt att ha en lite bättre WYSIWYG-redigerare, så att det inte var ett måste att köra applikationen bara för att se en GUI-förändring.

Tack vare användning av resursfiler hos Android och nib-filer för Iphone, blev det enkelt att låta applikationen stödja flera olika språk.

6.3. Distribution och test

Emulatore för både Android och Iphone var perfekt i alla anseenden förutom bristen på simulering av olika uppkopplingshastigheter.

Att distribuera applikationen till en fysisk enhet är dock betydligt mindre komplicerat för Android än för Iphone. Att bara kunna koppla in en telefon och testa direkt via Eclipse är en väldigt bra funktion för test. Att det inte är ett måste att lägga upp en applikation på Android Market är också uppskattat. Möjligheten att bara kunna distribuera en installationsfil för Androidapplikationer (se avsnitt 5.2.3) kan dock vara en säkerhetsrisk, då det är svårt att ha kontroll över vem som har åtkomst till applikationen.

Vad gäller Iphoneapplikationen hade vi inte möjlighet att distribuera applikationen via Appstore, då applikationen inte skulle bli godkänd av Apple för distribuering. Detta lämnade oss med alternativet ad-hoc-lösning (se avsnitt 5.3.3), som vi

uppfattar är en väldigt omständlig och onödigt komplicerad process för att tillgängliggöra denna applikation för Sogeti och deras kunder.

Det hade varit bättre om det bara var möjligt att tillhandahålla en fil för nedladdning, vilket är möjligt med Androidapplikationen. Den ad-hoc lösning, som till sist användes för distribuering, upplevs för just denna enkla applikation som för komplicerad. Om det istället hade rört sig om en applikation med högre krav på säkerhet för intern distribution och under 100 användare hade ad-hoc-lösningen varit ett mycket bra och säkert alternativ för applikationsdistribution.

6.4. Webbtjänstkommunikation

Ksoap2-android föreföll sig ganska smidigt, men saknade på många punkter god dokumentation. Detta gjorde det svårt att få den grundläggande förståelsen för hur kommunikationen mellan Androidtelefonen och webbtjänsten skulle gå till väga.

När väl kommunikationen mellan Androidapplikation och webbtjänst hade uppnåtts gick det bra att hämta ut SOAP-informationen och "översätta" det till de objekt som används i Aktiviteterna.

Till Iphone finns endast hjälp med kommunikationen till webbtjänsten. All tolkning (parsning) behövdes göras manuellt, vilket var väldigt tidskrävande. Däremot gav den stor hjälp med att bygga upp SOAP förfrågan.

7. UNDERLAG FÖR VAL AV PLATTFORM

Detta kapitel presenterar ett underlag för val av plattform vid utveckling av webbtjänstunderstödda Android- och Iphoneapplikationer. Underlaget är riktat till företag som önskar utveckla en Android- eller Iphoneapplikation som kommer att distribueras internt, kommunicera med en webbtjänst och ha ett grafiskt tilltalande användargränssnitt.

Baserat på empirin och analysen presenterat i avsnitt 5.2, 5.3 och 6 har vissa punkter trätt fram. Dessa presenteras nedan.

7.1. Android är lättare att lära sig och distribuera

Utvärderingen i föregående kapitel belyser några centrala punkter rörande utveckling av golfapplikationen, som togs fram under problematiseringen. Där syns bl.a. att Android upplevs lättare att lära sig om man kommer från en Java eller .Net bakgrund, vilket vi gör. Detta för att Objective-C skiljer sig vad gäller syntax och minneshantering.

Utvecklingsmiljön i form av Interface Builder och Xcode är väldigt kraftfull och har vissa fördelar jämfört med Eclipse, speciellt vad gäller utformning av grafiskt användargränssnitt.

Det finns en stor gemenskap för att lära sig både Android och Iphone utveckling, vilket är till stor nytta för den som önskar lära sig att utveckla applikationer för plattformarna.

För den här typen av applikation, som ska distribueras internt och inte är av allmänintresse, har Android enklare möjligheter för distribuering. Då det inte rör sig om något högriskprogram förefaller ad-hoc lösningen från Apple onödigt krånglig, och det är mer passande att applikationsfilen kan laddas ner via webbplatsen.

Slutligen bör faktumet att en Macintosh behövs för att utveckla applikationer för Iphone nämnas, då företag eller organisationer som använder Windows- eller Linuxbaserade miljöer för utveckling då måste ta ställning till eventuella inköp av hårdvara för utveckling av Iphoneapplikationer.

7.2. Iphone, bättre förutsättningar för tilltalande GUI

Tack vare Interface Builder är utvecklingen av Iphones användargränssnitt betydligt smidigare. Hanteringen av Androids XML-filer tillsammans med bristfälligt stöd i Eclipse för att i realtid visa hur användargränssnittet kommer att se ut, gör det onödigt krångligt att utveckla GUI för Android. Det är självklart en vanesak att lära sig bygga upp GUI genom XML-struktur, men för oss är Interface Builder klart överlägset Eclipses verktyg för att bygga grafiska användargränssnitt. Även att vänja sig vid användandet av ListAdapters för sammankoppling av data till listelement, var ganska svårt i början.

Att använda sig av egenanpassade celler i Iphoneapplikationen för att presentera information i Iphoneapplikationen kändes krångligt till en början, men när tanken bakom outlets och delegates hade klarnat gick utvecklingen väldigt smidigt.

Det faktum att Iphone använder sig av hårdvaruacceleration för användargränssnittsrendering ger också ett stort plus för Iphone då det bidrar till ett snabbt och tilltalande användargränssnitt.

7.3. Otillräckligt stöd för webbtjänster

Vi upplever att det krävs ett bättre stöd för webbtjänster för både Android och Iphone OS. Speciellt rörande webbtjänster byggda i .NET och som använder dataset för att förmedla data. Det finns inga inbyggda funktioner för detta idag utan man måste använda egenskriven kod eller använda externa mjukvarubibliotek.

Båda plattformarna bör integrera bättre stöd för hantering av webbtjänster. Huruvida det är bättre implementerat för andra webbtjänstestrukturer såsom REST eller RPC-XML är inte grundligt undersökt, men den allmänna uppfattningen är att webbtjänster behöver ett bredare stöd hos båda plattformarna.

Det går förvisso att använda sig av rena HTTP-anrop, få tillbaka ett rent XML-svar till servern som sedan tolkas av en XML-parser som finns inbyggd i både Android- och Iphone ramverket. Vid den korta inblick av den mobila applikation utvecklad för Windows Mobile 5, och den grundläggande insikten i hur dataset används för att hantera webbtjänstkommunikation upplever vi dock att betydligt bättre implementering för webbtjänster borde utvecklas för Android- och Iphoneplattformarna inom en överskådlig framtid.

Med det ökade kravet på aktuell och även situationsanpassad data, speciellt för smartphones (se avsnitt 4.3), är det konstigt att det inte finns bättre implementeringar för webbtjänsteunderstödda applikationer för dessa plattformar.

8. DISKUSSION

Denna utvärdering har gett oss goda grundförutsättningar för att förstå Android- och Iphoneplattformarna. Det har gett oss insikt i skillnader, fördelar och nackdelar med de olika plattformarna, som säkerligen kan vara till nytta för andra som vill veta förutsättningar för att utveckla liknande applikationer till respektive plattform.

8.1. Syftets och frågeställningarnas relevans

Bakgrundsbeskrivningen tillsammans med presenterad teori över den växande trenden av smartphones samt populariteten hos operativsystem för smartphones (se avsnitt 4.2.1) har fått oss att inse hur aktuell och välbehövad denna undersökning är. Det finns idag bristfälligt underlag rörande två av de större aktörerna på smartphone marknaden, nämligen Android och Iphone.

Med webbtjänster som en mer och mer naturlig kommunikationsväg för mobilapplikationer, känns just den utvecklade golfapplikationen bra representant av applikationer för smartphones.

I vår problematisering väcks frågan vad som krävs för att utveckla en webbtjänstunderstödd applikation för Android och Iphone. Genom de förutsättningar som presenteras i avsnitt 5.2, 5.3 och 5.4 återspeglas och besvaras frågorna för att sedan analyseras i avsnitt 6 och 7. Våra litteraturstudier tillsammans med fallstudien där vi konkret utvecklade applikationerna har gett oss tillräckligt med kunskap för att kunna presentera den data och baserat på dem dra våra slutsatser.

Uppdragsgivarens mål ledde oss mot undersökningen av applikationer för intern distribuering. Detta kanske inte är den bredaste gruppen av applikationer som utvecklas idag, men det är en relevant aspekt för applikationers distribution som inte belyses tillräckligt. Även om offentlig distribution är det vanligaste alternativet för majoriteten av utvecklare, anser vi att just företag ofta söker efter ett mer kontrollerat sätt att distribuera applikationer på.

8.2. Reflektion över resultatets kvalitet

Resultatet av undersökningen (se avsnitt **Error! Reference source not found.**) är starkt baserat på författarnas tidigare erfarenheter samt tidsramen av arbetet.

Som vi nämnde i metodkapitlet hade slutsatsen att Android är lättare att sätta sig in i troligen blivit annorlunda om vi hade en bakgrund i C-programmering. Vidare hade stödet för webbtjänster säkerligen kunnat utforskas i större utsträckning, snarare än att bygga om den existerande webbtjänsten för att få den att fungera med våra applikationer. Ett exempel hade varit att anpassa de XML-strängar som skickas från enheterna så att de motsvarade de dataset som webbtjänsten krävde, på samma sätt som att tolkningen av datasetet skulle ha kunnat utläsas då även de i grund och botten är vanlig XML.

På grund av tidsbrist valde vi att utveckla applikationerna till de två plattformarna separat, detta leder naturligtvis till att vi inte har möjlighet att individuellt få insikt i de båda plattformarna. Detta i sin tur leder till att analysen egentligen inte blir av plattformarna, utan av våra subjektiva tolkningar av dessa.

8.3. Vidareutveckling av golfapplikationen

Utöver de funktioner som utvecklades för golfapplikationen i detta arbete, finns flera andra tekniker som skulle ha kunnat utnyttjas för en starkare användarupplevelse, t.ex. genom att göra applikationen mer medveten om sammanhanget som användaren befinner sig i. En funktion skulle kunna vara att använda GPS-positionering för att bara presentera de uppkommande tävlingar i det specifika området som användaren befinner sig i. Vidare skulle GPS-positionering kunna användas för att registrera längd på slag för ytterligare statistik över hur deltagarna presterade i golftävlingen.

8.4. Uppslag till vidare studier

Området runt utveckling för mobila applikationer är allt för dåligt täckt i akademiska kretsar. Något som skulle kunna undersökas är utveckling av applikationer med andra förutsättningar och för andra centrala plattformar än de som inringats i detta arbete. Vidare skulle metodologier för att porta applikationer mellan olika plattformar kunna utformas, då spridningen av operativsystem för smartphones troligen aldrig kommer att koncentreras till endast en leverantör.

LITTERATURFÖRTECKNING

- Admob.com. (mars 2010). *metrics.amob.com*. Hämtat från Admob Mobile Metrics Report: <http://metrics.admob.com/wp-content/uploads/2010/04/AdMob-Mobile-Metrics-March-2010.pdf> den 26 mars 2010
- Alesso, P. H., & Smith, C. F. (2004). *Developing Semantic Web Services*. Wellesey, MA: A K Peters, Limited.
- Android. (2010). *Android 2.1 Platform*. Hämtat från Android Developers: <http://developer.android.com/sdk/android-2.1.html> den 26 april 2010
- Android. (2010). *Android 2.1 Platform*. Hämtat från Android Developers: <http://developer.android.com/sdk/android-2.1.html> den 26 April 2010
- Android. (2010). *Download the Android SDK*. Hämtat från Android Developers: <http://developer.android.com/sdk/index.html> den 17 maj 2010
- Android. (2010). *Download the Android SDK*. Hämtat från Android Developers: <http://developer.android.com/sdk/index.html> den 17 Maj 2010
- Android. (2010). *Getting Started*. Hämtat från Android Market: <http://market.android.com/publish/signup> den 17 maj 2010
- Android Open Source Project. (2008). *Release Features - Android 1.0*. Hämtat från Android Open Source Project: <http://source.android.com/release-features---android-1-0> den 26 april 2010
- Android Open Source Project. (2008). *Release Features - Android 1.0*. Hämtat från Android Open Source Project: <http://source.android.com/release-features---android-1-0> den 26 April 2010
- Android. (2010). *What is Android?* Hämtat från Android Developers: <http://developer.android.com/guide/basics/what-is-android.html> den 17 maj 2010
- Apple Inc. (n.d.). *iPhone OS Reference Library*. Retrieved 05 14, 2010, from iPhone OS Technology Overview: <http://developer.apple.com/iphone/library/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/iPhoneOSTechnologies/iPhoneOSTechnologies.html>

Apple Inc. (2010). *iPhone 3GS*. Hämtat från Apple: <http://www.apple.com/iphone> den 27 april 2010

Apple Inc. (2010). *iPhone 3GS*. Hämtat från Apple: <http://www.apple.com/iphone> den 27 April 2010

Apple Inc. (2010). *iPhone Developer Program - Standard Program User Guide for iPhone OS 3.1 v. 2.6*.

Apple Inc. (2010). *iPod Touch*. Hämtat från Apple: <http://www.apple.com/ipodtouch/> den 27 april 2010

Apple Inc. (2010). *iPod Touch*. Hämtat från Apple: <http://www.apple.com/ipodtouch/> den 27 April 2010

Apple Inc. (Maj 2010). *Press Release Library*. Hämtat från Apple: <http://www.apple.com/pr/library/> den 27 april 2010

AppleInsider. (den 21 augusti 2009). *Canalys: iPhone outsold all Windows Mobile phones in Q2 2009*. Hämtat från Apple Insider: http://www.appleinsider.com/articles/09/08/21/canalys_iphone_outsold_all_windows_mobile_phones_in_q2_2009.html den 27 april 2010

Backman, J. (1998). *Rapporter och uppsatser*. Lund: Studentlitteratur.

Beji, S., & Kadhi, N. E. (2008). An overview of mobile applications architecture and the associated. *The Fourth International Conference on Wireless and Mobile Communications* , 77-83.

Björklund, M., & Paulsson, U. (2003). *Seminarieboken - att skriva, presentera och opponera*. Lund: Studentlitteratur.

Braun, P., & Eckhaus, R. (2008). Experiences on Model-driven Software Development. *15th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems* , 490-493.

Cerami, E. (2002). *Web Services Essentials*. Sebastopol, CA: O'Reilly & Associates Inc. .

Fertalj, K., & Horvat, M. (2007). Comparing architectures of mobile applications. *arXiv* , 8.

Fling, B. (2009). *Mobile Design and Development*. Sebastopol, CA: O'Reilly Media, Inc. .

- Forman, G. H., & Zahorjan, J. (1994). The Challenges of Mobile Computing. *Computer* , 38-47.
- Free Software Foundation, Inc. (2007). *GNU General Public License*. Hämtat från <http://www.gnu.org/licenses/gpl.html> den 26 april 2010
- Free Software Foundation, Inc. (2007). *GNU General Public License*. Hämtat från <http://www.gnu.org/licenses/gpl.html> den 26 April 2010
- Gartner. (den 17 maj 2010). *Pressreleases - Gartner Says Worldwide Mobile Phone Sales Grew 17 Per Cent in First Quarter 2010*. Hämtat från Gartner: <http://www.gartner.com/it/page.jsp?id=1372013> den 24 maj 2010
- Hochgurtel, B. (2003). *Cross-platform Web Services Using C# and Java*. Hingham, MA: Charles River Media.
- HTC. (2010). *HTC Products*. Hämtat från HTC: <http://www.htc.com/europe/product.aspx> den 27 04 2010
- Jorgensen, D. (2002). *Developing .NET Web Services with XML*. Rockland, MA: Syngress Publishing.
- Krejcar, O., Janckulik, D., & Motalova, L. (2010). Dataflow Optimization Using of WiFi, GSM, UMTS, BT and GPS. *2010 Second International Conference on Computer Engineering and Applications* , 127-131.
- Microsoft Corporation. (2010). *.NET Compact Framework*. Hämtat från Microsoft MSDN: <http://msdn.microsoft.com/en-us/netframework/aa497273.aspx> den 27 april 2010
- Newcommer, E. (2002). *Understanding Web Services - XML, WSDL, SOAP and UDDI*. Boston, MA: Addison-Wesley Professional.
- Open Handset Alliance. (september 2009). *Android Overview*. Hämtat från Open Handset Alliance: http://www.openhandsetalliance.com/android_overview.html den 26 april 2010
- Open Handset Alliance. (September 2009). *Android Overview*. Hämtat från Open Handset Alliance: http://www.openhandsetalliance.com/android_overview.html den 26 April 2010

- Open Handset Alliance. (den 5 November 2007). *Industry Leaders Announce Open Platform for Mobile Devices*. Hämtat från Open Handset Alliance: http://www.openhandsetalliance.com/press_110507.html den 26 April 2010
- Open Source Initiative. (2010). *Open Source Initiative*. Hämtat från <http://www.opensource.org/> den 26 april 2010
- Open Source Initiative. (2010). *Open Source Initiative*. Hämtat från <http://www.opensource.org/> den 26 April 2010
- Oracle Corporation. (den 12 januari 2010). *About the Java Technology*. Hämtat från Java: <http://java.sun.com/docs/books/tutorial/getStarted/intro/definition.html> den 30 maj 2010
- Plakalović, D., & Simić, D. (2010). Applying MVC and PAC patterns in mobile. *Journal of Computing, Volume 2, Issue 1*, 65-72.
- Rahimian, V., & Habibi, J. (2008). Performance Evaluation of Mobile Software Systems: Challenges for a Software Engineer. *2008 5th International Conference on Electrical Engineering, Computing Science and Automatic Control*, 346-351.
- Research In Motion. (2010). *BlackBerry Java Application Development*. Hämtat från BlackBerry: <http://na.blackberry.com/eng/developers/javaappdev/> den 27 april 2010
- Research In Motion. (2010). *Company*. Hämtat från RIM: <http://www.rim.com/company/> den 27 april 2010
- Richardson, L., & Ruby, S. (2007). *RESTful Web Services*. Sebastopol, CA: O'Reilly Media.
- Sheshagiri, M., Sadeh, N. M., & Gandon, F. (2004). *Using Semantic Web Services for Context-Aware Mobile Applications*. Pittsburgh, PA: Carnegie Mellon University.
- Sogeti.com. (2010). *About Us*. Hämtat från Sogeti.com: <http://www.sogeti.com/Curious-about-Sogeti/About-us/> den 03 april 2010
- Sogeti.se. (2010). *Om Sogeti - Affärsidé/Vision*. Hämtat från Sogeti.se: <http://www.sogeti.se/Om-Sogeti/AffarsideVision/> den 03 April 2010
- Symbian. (den 27 04 2010). *About Symbian*. Hämtat från Symbian: <http://www.symbian.org/about-us> den 27 04 2010

- The Apache Software Foundation. (2010). Hämtat från The Apache Software Foundation:
<http://www.apache.org/> den 26 april 2010
- W3C. (den 26 november 2008). *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. Hämtat från W3C: <http://www.w3.org/TR/2008/REC-xml-20081126/#sec-origin-goals> den 27 april 2010
- W3C. (den 26 November 2008). *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. Hämtat från W3C: <http://www.w3.org/TR/2008/REC-xml-20081126/#sec-origin-goals> den 27 April 2010
- W3C. (den 14 mars 2010). *Extensible Markup Language (XML)*. Hämtat från W3C: <http://www.w3.org/XML/> den 27 april 2010
- W3C. (den 14 Mars 2010). *Extensible Markup Language (XML)*. Hämtat från W3C: <http://www.w3.org/XML/> den 27 April 2010
- W3C. (den 27 april 2007). *SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)*. Hämtat från W3C: <http://www.w3.org/TR/soap12-part1/#intro> den 27 april 2010
- W3C. (den 27 April 2007). *SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)*. Hämtat från W3C: <http://www.w3.org/TR/soap12-part1/#intro> den 27 April 2010
- Yin, R. (1994). *Case Study Research - Design and Methods 2nd edition*. Thousand Oaks, California: Sage Publications.
- Yu, W. D., Amjad, T., Goel, H., & Talawat, T. (2008). An Approach of Mobile Database Design Methodology for Mobile Software Solutions. *The 3rd International Conference on Grid and Pervasive Computing - Workshops* , 136-144.