

CHALMERS



Plattformsoberoende Applikationsutveckling för Smarta Mobiltelefoner

Är en plattformsoberoende utvecklingsmiljö framtiden av mobil applikationsutveckling?

Cross-Platform Application Development for Smartphones

Is cross-platform development environment the future of mobile application development?

EGHOGHO ILESO
JOSEPH HEJDERUP

Examensarbete

Högskoleingenjörsprogrammet för datateknik

CHALMERS TEKNISKA HÖGSKOLA

Institutionen för data- och informationsteknik

Göteborg 2012

Innehållet i detta häfte är skyddat enligt Lagen om upphovsrätt, 1960:729, och får inte reproduceras eller spridas i någon form utan medgivande av författarna. Förbudet gäller hela verket såväl som delar av verket och inkluderar lagring i elektroniska och magnetiska media, visning på bildskärm samt bandupptagning.

“There is no programming language, no matter how structured, that will prevent programmers from making bad programs.”

-Larry Flon

Abstract

This thesis is a study of three established cross-platform development tools for smartphone applications. An evaluation has been conducted to examine whether these development tools can be serious contenders to the most common native development tools such as the Android SDK and XCode.

Firstly, a research of the various cross-platform development tools was conducted to identify the top three contenders on the market. In order to evaluate the selected cross-platform development tools, three identical applications were developed for each cross-platform, and by this approach we gained an understanding of the capabilities of the particular development tool. This strategy makes it possible to verify if the internal resources in the smartphone, such as the camera, GPS, multimedia controllers for music could be utilized in the same way as in the native development environment without any compromise.

The comparison of the chosen cross-platform development tools showed that they had a limited hardware support, poor prospects of debugging and the development process was often time-consuming. There were also some instances regarding some of the development tools, where the developed application worked fine for the Android platform but were not fully supported by iOS and vice versa. The current cross-platform development tools had an overall lack of functionality, but the concept of cross-platforms have a great potential and are likely to be of great importance in application development in the near future.

Key words: Cross platform, Titanium, Phonegap, Marmalade, MoSync, Corona, mobile application development

Sammanfattning

Denna rapport är en sammanställning av en studie av tre etablerade plattformsoberoende utvecklingsmiljöer för mobilapplikationer. En utvärdering har genomförts för att undersöka om dessa utvecklingsmiljöer på allvar kan utmana de vanligaste nativa utvecklingsmiljöerna, Android SDK och XCode.

En inventering av olika plattformsoberoende utvecklingsverktyg gjordes initialt för att finna de tre bästa utmanarna på marknaden. För att kunna utvärdera de valda plattformsoberoende utvecklingsmiljöerna har en identisk applikation skapats med varje utvecklingsverktyg och därigenom bildat en uppfattning av användarvänligheten hos den specifika utvecklingsmiljön och dess potential. Denna strategi gör det möjligt att kontrollera att de inbyggda resurserna i mobiltelefonen t.ex. kamera, GPS, start och stopp funktioner för musik kan fås att fungera. På samma sätt som i den nativa utvecklingsmiljön utan några kompromisser.

Jämförelsen av de undersökta plattformsoberoende utvecklingsverktygen visade att dessa hade brister i hårdvarustödet, begränsade möjligheter till felsökning och utvecklingsprocessen var stundtals tidsödande. Det förekom även i vissa utvecklingsverktyg att applikationen fungerade bra i Android men inte fullt ut i iOS eller tvärtom. De plattformsoberoende utvecklingsverktygen hade genomgående bristande funktionalitet, men de har en stor utvecklingspotential och kommer sannolikt att ha en stor betydelse när det gäller applikationsutveckling i framtiden.

Nyckelord: Cross plattform, Titanium, PhoneGap, Marmalade, MoSync, Corona, mobil applikationsutveckling

Förord

Denna rapport är ett examensarbete inom datateknik motsvarande 15 högskolepoäng. Examensarbetet har utförts vid institutionen för Data- och Informationsteknik, Chalmers Tekniska Högskola.

Grundläggande kunskaper inom programmering och mobil applikationsutveckling underlättar förståelsen av rapporten.

Vi vill tacka Spinit AB, som erbjudit oss detta högaktuella examensarbete samt generöst ställt upp med kunskaper och resurser. Vi vill också uttrycka vår stora tacksamhet till våra handledare Sakib Sistek och Bengt Nilsson som har stöttat oss under arbetets gång.

Slutligen vill vi ge ett stort tack till våra familjer och studiekamrater som varit oss behjälpliga i vårt arbete.

Göteborg juni 2012

Eghogho Ileso & Joseph Hejderup

Innehållsförteckning

ABSTRACT	IV
SAMMANFATTNING	V
FÖRORD	VI
1 INLEDNING	3
1.1 BAKGRUND	3
1.2 PROBLEM	3
1.3 SYFTE	4
1.4 AVGRÄNSNINGAR	4
2 METOD	5
3 TEKNISK BAKGRUND	6
3.1 MOBILA PLATTFORMAR	6
3.1.1 iOS	6
3.1.2 ANDROID	8
3.2 TEKNIKERNA BAKOM CROSS PLATTFORMS UTVECKLINGSVERKTYG	10
3.2.1 INTRODUKTION	10
3.2.2 JAVASCRIPT FRAMEWORKS	10
3.2.3 APP FACTORIES	10
3.2.4 WEB-TO-NATIVE WRAPPERS	10
3.2.5 RUNTIMES	12
3.2.6 SOURCE CODE TRANSLATORS	12
3.2.7 HTML5	12
3.3 URVAL AV CROSS PLATTFORMS VERKTYG	13
3.3.1 PHONEGAP (CORDOVA) MED JQUERY MOBILE	13
3.3.2 TITANIUM	14
3.3.3 MONO FOR ANDROID	15
3.3.4 MARMALADE	16
3.3.5 MOSYNC	17
3.3.6 CORONA (ANSCA MOBILE)	18
3.3.7 JÄMFÖRELSE AV CROSS PLATTFORMAR	19
4 GENOMFÖRANDE	20
4.1 SAMMANSTÄLLNING AV KRAVSPECIFICERING FÖR VAL AV CROSS PLATTFORMAR	20
4.1.1 INTRODUKTION	20
4.1.2 UNDERSÖKNING AV BESTÄLLARFÖRETAGETS VERKSAMHET	20
4.1.3 ANALYS AV ETABLERADE CROSS PLATTFORMAR	20
4.1.4 MOTIVERING TILL VAL AV CROSS PLATTFORMAR BASERAT PÅ KRAV	22
4.1.5 KORRIGERING UNDER UTVÄRDERINGEN	22
4.2 APPLIKATIONEN	23
4.2.1 KRAV	23
4.2.2 KONCEPTET DIGITALDIARY	23
4.2.3 USER STORIES	24
4.3 IMPLEMENTATION AV APPLIKATIONEN I VARJE CROSS PLATTFORM	25
4.3.1 STRUKTUR, LOGIK OCH GRAFISKA GRÄNSSNITTET	25

4.3.2	HÅRDVARA	27
5	RESULTAT	28
5.1	INTRODUKTION	28
5.2	RESULTAT AV IMPLEMENTERING AV APPLIKATIONEN I RESPEKTIVE CROSS PLATTFORM	28
5.2.1	DIARIES	28
5.2.2	NEW ENTRY	28
5.2.3	ENTRIES	29
5.2.4	DATALAGRING	29
5.3	UTVÄRDERING AV RESPEKTIVE CROSS PLATTFORM	29
5.3.1	APPCCELERATOR TITANIUM	29
5.3.2	MONO FOR ANDROID	30
5.3.3	CORDOVA PHONEGAP OCH JQUERY MOBILE	31
6	DISKUSSION OCH SLUTSATS	32
6.1	ANALYS OCH DISKUSSION	32
6.2	SLUTSATS	33
6.3	KRITISK DISKUSSION	34
6.4	FORTSATT FORSKNING	35
7	KÄLLFÖRTECKNING	36
7.1	REFERENSER	36
7.2	BILDKÄLLOR	42
7.3	TABELLKÄLLOR	42

APPENDIX

Appendix A: Ordlista.

Appendix B: Källkoden till applikationen i Titanium.

Appendix C: Källkoden till applikationen i Mono For Android.

Appendix D: Källkoden till applikationen i PhoneGap.

1 INLEDNING

1.1 Bakgrund

Efter lanseringen av Apples mobila operativsystem iOS 2007 har marknaden för smartphones utvecklats kraftigt och har fått en växande betydelse inom handel, marknadsföring, kommunikation och multimedia. Man har med hjälp av smartphones skapat en ekonomiskt, allt viktigare marknadsplats. Apple har vuxit sig stark inom denna bransch genom sitt nytänkande, tilltalande design och bra teknik. På denna marknad har även Googles Android som lanserades 2008 vuxit sig stark och har erövrat marknadsandelar från Apple och börjar bli en jämbördig konkurrent.

Applikationerna som finns att ladda ner har gjort mobiltelefonen till en konkurrent för datorn med sina mångsidiga funktionaliteter som går att bygga ut med ännu fler applikationer. Många företag av betydelse vill gärna ha sin egen applikation för att tillfredsställa marknaden, för att behålla sina kunder. Idag är dessa två mobila operativsystem de största aktörerna och Apples App Store tillhandahåller 540 000 respektive Google Play med 350 000 nedladdningsbara registrerade applikationer [1, 2, 3].

Denna situation har gjort att många applikationsutvecklare tvingats sig till att utveckla mobilapplikationer riktade mot en specifik plattform eller mot flera plattformar beroende på uppdragsgivarens önskemål. Utformning och utveckling av en funktionell mobilapplikation är en både tids- och resurskrävande uppgift som bland annat involverar utformning av grafiskt gränssnitt, skapandet av kravspecifikationer och säkerställning av dess funktionalitet.

Utveckling av applikationer leder ofta till ett "dubbelarbete" eftersom möjligheten att återanvända befintlig källkod från en applikation för att portera den till andra mobila plattformar är i dagsläget begränsad. De viktigaste orsakerna till dessa begränsningar beror på de fundamentala skillnaderna mellan olika nativa utvecklingsmiljöer. Java används som programmeringsspråk för Android och Objective-C gäller för iOS. En lösning på problemet som utvecklas i snabb takt är olika plattformsberoende utvecklingsverktyg, även kallad cross plattform utvecklingsverktyg, vilket har skapat ett alternativ till de nativa utvecklingsmiljöer och där man kan skapa en applikation som kan exekveras på ett flertal mobilplattformar. Dock finns det för närvarande en stor skepsis till dessa utvecklingsverktyg och en del hävdar att det finns alltför stora brister och att applikationer utvecklade i dessa miljöer inte lever upp till samma användarupplevelse och prestanda i jämförelse med en nativ applikation [4, 5].

1.2 Problem

Finns det någon kraftfull universal utvecklingsmiljö som kan förenkla utvecklingen av applikationer samt portera dessa till flera mobila plattformar samtidigt? Går det att utnyttja mobilens interna resurser på samma sätt som vid användandet av en nativ utvecklingsmiljö?

1.3 Syfte

Syftet är att göra en evaluering utav etablerade plattformsoberende utvecklingsverktyg som finns på marknaden. Som bas för denna evaluering kommer en applikation att skapas för olika cross plattformar. Vi kommer undersöka om de olika plattformarna på allvar kan utmana de nativa utvecklingsmiljöerna.

1.4 Avgränsningar

- Endast tre cross plattforms utvecklingsverktyg kommer att evalueras. Detta för att kunna göra en kvalitativ evaluering under den begränsande tiden för examensarbetet.
- De utvecklade applikationerna kommer inte att jämföras med varandra utan fokus kommer att vara på utvecklingsmiljöerna.
- Moment ur utvecklingsprocessen som exempelvis användandet av versionshanteringssystem, skapandet av testfall kommer inte att ingå i studien.
- Målet med examensarbetet är inte att skapa kommersiella applikationer som kan lanseras.
- Mål plattformar som används i denna evaluering är iOS och Android, andra plattformar som exempelvis Windows phone eller BlackBerry kommer inte att ingå i studien.

2 METOD

- **Metodik av projekt utförandet:** Arbetet har utförts iterativt med användning av den agila Scrum-metoden, där vi har haft täta avstämningar och en sprint varje vecka.
- **Utformandet av kravspecifikationer för cross plattform utvecklingsverktyg:** En intern undersökning har utförts hos beställarföretaget, för att identifiera vilka nuvarande utvecklingsverktyg och programmeringsspråk som utvecklarna i företaget använder sig av samt deras önskemål och krav. Detta är för att minimera risken att vi väljer ett cross-plattform utvecklingsverktyg som använder ett programmeringsspråk eller utvecklingsmiljö som deras anställda inte är bekanta med.
- **Research om cross plattform utvecklingsverktyg:** Efter sammanställning, prioritering av kraven och önskemål har en mindre research utförts för att kartlägga några etablerade cross plattform utvecklingsverktyg. Utifrån detta har tre stycken cross plattform utvecklingsverktyg valts som kan bäst uppfylla de ställda kraven.
- **Efterforskning om fundamentala skillnader mellan Android och iOS:** En efterforskning har gjorts för att se skillnaderna vad gäller utvecklingsprocessen för nativa utvecklingsmiljöer.
- **Applikationer skapade med cross plattform utvecklingsverktyg:** För att kunna evaluera ett utvecklingsverktyg anser vi att det är bäst att utveckla en applikation i denna miljö för att kunna få en förståelse av utvecklingsprocessen samt bristerna i utvecklingsmiljön.

3 TEKNISK BAKGRUND

3.1 Mobila plattformar

3.1.1 iOS

3.1.1.1 Introduktion

iOS är ett mobilt operativsystem som är speciellt utvecklat för Apples mobila produkter som exempelvis iPad, iPod touch och iPhone. Deras vision är att skapa ett operativsystem som håller hög standard med en extraordinär användarupplevelse genom att förändra sättet vi tänker och använder mobila enheter [6].

3.1.1.2 Operativsystem och utvecklingsmiljö

iOS är ett kraftfullt mobilt operativsystem baserat på Apples OS X, som är deras operativsystem för Mac enheter [6].

iOS applikationer utvecklas i den officiella utvecklingsmiljön XCode, som i dagsläget enbart går att installera på Mac OS X. Utvecklingsmiljön erbjuder applikationsutvecklare ett flertal utvecklingsverktyg för design av användargränssnitt, felsökning och simulering av applikationer [14]. Dessutom finns det ett kraftfullt verktyg som kallas för "Instruments" som kan identifiera prestandaförluster och flaskhalsar i applikationen.

Programmeringsspråket Objective-C används för utveckling av iOS applikationer. Objective-C är baserat på standard ANSI C och är utformad för att ge C-språket objekt orienterade egenskaper [7].

3.1.1.3 Distribution och kostnader

iOS applikationer distribueras och marknadsförs via Apples programbutik App Store. Applikationsutvecklare som är registrerade hos Apple har möjligt att använda denna tjänst mot att Apple tar 30 procent av intäkterna för varje såld applikation [8].

Apple har riktlinjer för vilka applikationer som tillåts i deras programbutik. För att inspektera att en applikation inte bryter mot deras överenskommelser görs en granskning på applikationen innan den lanseras på App Store [9].

För att bli en registrerad användare krävs det att man deltar i en så kallad "iOS developer program" där det billigaste alternativet kostar i dagsläget 99 USD/år [10].

3.1.1.4 Funktioner och restriktioner

Apple tillhandahåller tjänster som är unika för deras plattform. Flera av dessa tjänster finns integrerat i officiella API: er och gör det mer attraktivt för utvecklare. Nedan är en beskrivning av ett urval av deras specifika tjänster [11]:

- **iCloud Storage:** Denna "moln"-tjänst kan användas av utvecklare för att lagra dokument och annan data i molnet.

- **Notification Center (Push Notifications):** Detta notifikations system kan enkelt användas tillsammans i egenutvecklade applikationer för att notifiera användaren när applikation körs i bakgrunden eller är avstängd.
- **Game Center:** Är ett “social gaming” nätverk som bland annat samlar statistik på användarens poäng och prestationer för spel som använder denna tjänst.
- **iPod Library:** Åtkomst till musikbiblioteket i applikationen.
- **AirPlay:** Möjliggör strömning av musik och video till en enhet som stödjer AirPlay.

Tabellen nedan ger en kort översikt av funktioner som förövrigt erbjuds i diverse API: er.

Datahantering	SQLite, Åtkomstmöjlighet till kontakter och foton, dock begränsad tillgång till kalender, XML-filer, delning av data mellan applikationer[12].
Nätverk och Internet	BSD Sockets, Bonjour för att hitta enheter, HTTP-strömning, stöd för att spara HTML5 “offline” data i webbläsaren Safari[13].
Grafik	Klarar 2D och 3D rendering med tekniker som OpenGL ES och Quartz 2D [15].

Tabell 1: kort översikt av funktioner som förövrigt erbjuds i diverse API: er

Apple har på vissa av sina API: er begränsningar på tjänster och enhets funktioner. Ett exempel är användningen av SMS i applikationen. Där tillåts det inte att ta emot SMS till applikationen utan enbart skicka SMS [16]. När det gäller Bluetooth funktionerna är dessa begränsade och få tjänster är tillåtna. Detta är främst för att förhindra fildelning mellan olika enheter.

3.1.1.5 Informationskällor (Dokumentation, Communities)

Apple har sin utvecklardokumentation på iOS development center, som är en webbplats med information rörande utveckling med iOS, utvecklarlicenser, API: er, självstudiekurser och kodexempel.

Det existerar ett flertal stora utvecklargrupper som använder olika communities och diverse forum som erbjuder utvecklare att studera små självstudiekurser och få hjälp med problem.

3.1.1.6 Versionshistorik

Nedan är en sammanställning av de större mjukvaruuppdateringarna som Apple har lanserat. Den senaste versionen är iOS 5.1.1 som lanserades den 7 maj 2012.

Version	Datum
iOS 1.0	29 Juni 2007
iOS 2.0	11 Juli 2008

iOS 3.0	17 Juni 2009
iOS 4.0	21 Juni 2010
iOS 5.0	12 Oktober 2011

Tabell 2: En sammanställning av de större mjukvaruuppdateringarna som Apple har lanserat.

3.1.2 Android

3.1.2.1 Introduktion

Android är ett mobilt operativsystem som ägs av Google Inc. Android har med tiden vuxit fram som en stor konkurrent till Apples iOS och erbjuder applikationsutvecklare en miljö som är helt kostnadsfri och byggt på öppen källkod [17, 18, 19]. Deras målsättning är att skapa en plattform som är fri för mobiloperatörer, tillverkare och andra aktörer att själva sätta gränserna [20].

3.1.2.2 Utvecklingsmiljö & Operativsystem

För att skapa en Android applikation måste man använda deras officiella SDK [21]. SDK: n innehåller dokumentation och verktyg såsom DDMS och enhetsemulator som används för felsökning och testning. SDK: n kan installeras på Linux, OS X och Windows.

Google har ingen officiellt integrerat utvecklingsmiljö för Android dock så erbjuder de ett officiellt plugin till Eclipse [22].

Huvudsakligen programmeras applikationer med Java, men det finns möjligheter att delar av applikation kan utvecklas med C/C++ genom användning av deras officiella NDK [23].

3.1.2.3 Distribution och kostnader

Applikationer kan distribueras och marknadsföras på officiella programbutiken Google Play kostnadsfritt. En engångsavgift på 25 USD tas ut för att publicera applikationer [24]. Alternativa programbutiker existerar vid sidan av Google Play som exempelvis Amazon App Store for Android [25].

3.1.2.4 Funktioner och restriktioner

Google har ålagt sig att Android skall vara ett fritt OS där utvecklare och andra parter har fria händer[20]. Denna inriktning skiljer sig från Apple som hindrar inflytande av tredje parter i deras plattform. Tredje parter kan med Android skapa och definiera sina egna tjänster obehindrat.

Den senaste versionen är Android 4.0 som även kallas för "Ice Cream Sandwich". Nyheterna i denna version innehåller förbättringar som gjorts i vissa API: er, nedan är en kort beskrivning några utav dessa [26].

Text-to-speech API: Utvecklare kan utnyttja denna funktion för att läsa upp text i applikationen.

WiFi Direct: Idén med WiFi direct är att kunna ansluta till en annan enhet direkt utan en internet förbindelse eller access punkt. Detta möjliggör för utvecklare att använda denna teknik för att överföra filer eller för exempelvis strömning av musik.

Restriktioner i API: er

För närvarande finns det inga allmänna begränsningar av funktionalitet i de officiella API: erna. Dock finns det möjlighet för utvecklare att begränsa tjänster eller användning av interna resurser som exempelvis användning av kamera i applikationen. Detta fastställs i en fil som kallas för Android Manifest. Manifestet innehåller information om applikationen som Android systemet behöver veta innan applikationen kan exekveras [27].

Datahantering	Shared preferences: Användas för att spara primitiv datatyper. Internal Storage: Enhetens interna minne, applikationsdata är privat och kan inte användas av andra applikationer eller tjänster. External Storage: Detta är det externa minnet som vanligtvis är SD-kortet. Applikationsdata som lagras här är inte skyddad och kan läsas och skrivas av andra applikationer eller tjänster. SQLite database: En enklare version av SQL och databasen finns internt i mobilen[28].
Nätverk och Internet	Wi-Fi direct[27].
Grafik	Stödjer 2D och 3D rendering och använder tekniker som exempelvis OpenGL ES[27].

Tabell 3: kort översikt av funktioner som förövrigt erbjuds i diverse API:er

3.1.2.5 Informationskällor (Dokumentation, Communities)

Officiell dokumentation med API: er, självstudiekurser och kodexempel finns både tillgängligt på Android SDK och deras officiella webbplats på internet.

Precis som för iOS finns det väletablerade communities för att få rådgivning vid problem eller läsa mindre självstudiekurser.

3.1.2.6 Versionshistorik

Nedan är översiktlig sammanställning av de släppta versionerna. Den nuvarande versionen är 4.0.4 som lanserades torsdagen den 29 Mars 2012.

Version	Namn	Datum
Android 1.0	-	23 September 2008
Android 1.6	Donut	15 September 2009
Android 2.0	Éclair	26 Oktober 2009
Android 2.3	Gingerbread	6 December 2010
Android 3.0	Honeycomb	22 Februari 2011
Android 4.0.1	Ice Cream Sandwich	19 oktober 2011

Tabell 4: En sammanställning av de större mjukvaruuppdateringarna som Android har lanserat.

3.2 Teknikerna bakom cross plattforms utvecklingsverktyg

3.2.1 Introduktion

Plattformsoberende utvecklingsverktyg bygger på många olika underliggande tekniker för att nå ut till andra målgrupper än den traditionella målgruppen som består av mjukvaruutvecklare. I boken *Vision Mobile Cross-Platform Developer Tools* beskrivs fem distinkta tekniker som används i etablerade verktyg.

3.2.2 JavaScript frameworks

Ett Javascript framework är ett ramverk framtaget för att underlätta webbutvecklare att utnyttja egenutvecklade moduler som vanligtvis inte ingår i standard biblioteken [29]. I cross plattformar tillämpas detta för att skapa färdiga moduler som hanterar interaktionen i pekskrmar eller grafiska komponenter för att bygga ett gränssnitt som kan användas i flera mobila plattformar [30]. Exempel på cross plattformar som använder detta är JQuery Mobile och Sencha Touch.

3.2.3 App factories

“App factories” är en teknik som är inriktat för personer som saknar programmeringskunskaper. Man skall med denna teknik enkelt kunna skapa enkla applikationer utan att behöva skriva en rad kod [30]. Exempel på ett utvecklingsverktyg är AppMakr.

3.2.4 Web-to-native wrappers

Idén med denna teknik är att man skall kunna utnyttja webbt teknologier såsom HTML5, CSS och JavaScript för att skapa applikationer och vid kompilering av källkoden skapas en körbarativ applikation. Den nativa applikationen är ingen direkt översättning av webbkod till nativ kod utan man använder ett bibliotek som är en brygga mellan webbkoden och den underliggande nativa funktionalitet man vill ha åtkomst till [30].

Ett cross plattformsverktyg som bygger på denna teknik är PhoneGap. I Android implementeras en WebView i PhoneGap, som i princip är en webbläsare utan synliga navigationsknappar. Nedan är ett exempel av hur överbryggning ser ut för att få information om enhetens ID. Detta är hämtat ur källkoden för PhoneGap [31, 32].

```

51 public PhoneGap(Context ctx, WebView appView) {
52     this.mContext = ctx;
53     this.mAppView = appView;
54     audio = new AudioHandler("/sdcard/tmprecording.mp3", ctx);
55     uuid = getUuid();
56 }
57
58 public void beep(long pattern)
59 {
60     Uri ringtone = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
61     Ringtone notification = RingtoneManager.getRingtone(mContext, ringtone);
62     for (long i = 0; i < pattern; ++i)
63     {
64         notification.play();
65     }
66 }
67
68 public void vibrate(long pattern){
69     // Start the vibration, 0 defaults to half a second.
70     if (pattern == 0)
71         pattern = 500;
72     Vibrator vibrator = (Vibrator) mContext.getSystemService(Context.VIBRATOR_SERVICE);
73     vibrator.vibrate(pattern);
74 }
75
76 public String getPlatform()
77 {
78     return this.platform;
79 }
80
81 public String getUuid()
82 {
83     TelephonyManager operator = (TelephonyManager) mContext.getSystemService(Context.TELEPHONY_SERVICE);
84     String uuid = operator.getDeviceId();
85     return uuid;
86 }

```

Figur 1: Java källkoden för PhoneGap.

```

1 /**
2  * this represents the mobile device, and provides properties for inspecting the model, version, UUID of the
3  * phone, etc.
4  * @constructor
5  */
6 function Device() {
7     this.available = PhoneGap.available;
8     this.platform = null;
9     this.version = null;
10    this.name = null;
11    this.gap = null;
12    this.uuid = null;
13    try {
14        if (window.DroidGap) {
15            this.available = true;
16            this.uuid = window.DroidGap.getUuid();
17            this.version = window.DroidGap.getOSVersion();
18            this.gapVersion = window.DroidGap.getVersion();
19            this.platform = window.DroidGap.getPlatform();
20            this.name = window.DroidGap.getProductName();
21        }
22    } catch(e) {
23        this.available = false;
24    }
25 }
26
27 PhoneGap.addConstructor(function() {
28     navigator.device = window.device = new Device();
29 });

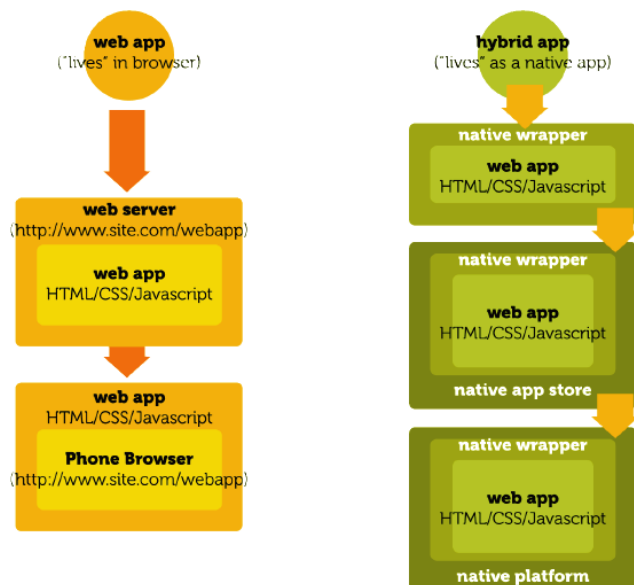
```

Figur 2: Javascript källkoden för device.js

Som man ser i figur 2 anropas *this.uuid = window.DroidGap.getUUID()* i JavaScript koden, som sedan anropar metoden *getUUID* i Java koden i figur 1.

Applikationer skapade på detta sätt kallas för *Hybrida applikationer*. Fördelarna med hybrida applikationer är att den kombinerar det bästa av två världar, nämligen att med webbapplikationer har man en större distributions bas och med nativa applikationer kan man utnyttja den underliggande funktionaliteten som inte är optimerat för mobila webbapplikationer [33].

Figuren nedan illustrerar skillnader mellan en webbapplikation och en hybridapplikation i en mobiltelefon.



Figur 3: Sammansättning av webb- och hybridapplikation.

3.2.5 Runtimes

En “runtime enviroment” är en körbar miljö som behövs för applikationer som kräver särskilt mjukvara för att kunna exekvera. Applikationer som kräver en runtime kan inte exekveras direkt i en nativ miljö[34]. Tekniken används i olika cross plattformar och har konstruerats så att delar av den skrivna programkoden skall översättas till byte kod vid “design-time” och vid “run-time” exekveras bytekoden[35]. Plattformar som bygger på denna teknik är Appcelerator Titanium och Xamarin [36].

3.2.6 Source code translators

Dessa lösningar bygger på att källkoden kompileras till byte kod, nativt programmeringsspråk (C++, Objective-C) eller till assembler direkt[35]. Denna teknik används i huvudsak för applikationer som kräver prestanda optimeringar och komplex logik[36]. Exempel på utvecklingsverktyg som använder källkods översättning är MoSync och Marmalade.

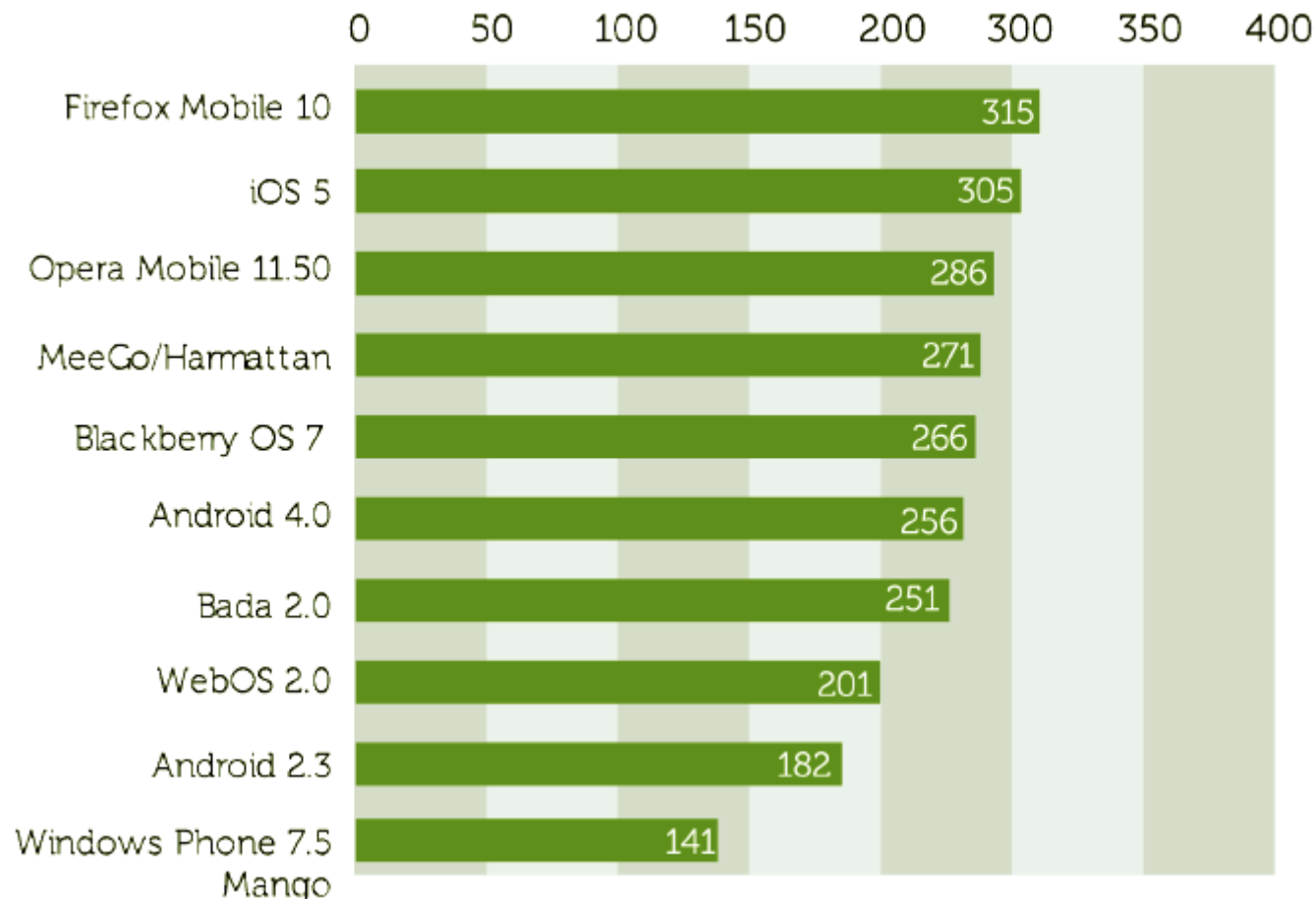
3.2.7 HTML5

HTML5 är den nya standarden för märkspråken HTML och XHTML. I dagsläget är den ingen officiell standard och ingen webbläsare har fullt stöd för HTML5. Man vill med HTML5 bland annat minska användning av plugins som exempelvis flash, förbättra fel hanteringen och göra den enhetsoberoende [39]. Förutom detta har man bland annat lagt till nya attributet och element som ger stöd för intern lagring, innehålls specifika element så som <article>, <footer>, <header>, <nav>, <section> [39].

HTML5 har hög popularitet bland webbutvecklare för att man inte behöver använda Adobe Flash som inte stöds i vissa mobila webbläsare som exempelvis Safari för iOS [40]. Stödet för HTML5 i mobila webbläsare är varierande, där Firefox Mobile 10 och iOS har bäst stöd medan Windows Phone 7.5 och Android 2.3 har sämst stöd.

En webbtjänst som poängsätter webbläsarens stöd för HTML5 är “The HTML5 test” och är ett testsystem som analyserar vilka HTML5 funktioner som finns implementerade i webbläsaren. Den maximala poängsättningen är 500 och poängen graderas beroende funktionens svårighetsgrad [41]. Nedan är en sammanställning av deras testresultat:

Figur 4: Jämförelse av HTML5 stöd i olika mobila webbläsare.



3.3 Urval av cross plattforms verktyg

3.3.1 PhoneGap (Cordova) med JQuery Mobile

3.3.1.1 Introduktion

PhoneGap är ett ramverk baserad på öppna webbt teknologier. Idén är att man skall utveckla en applikation med nativ funktionalitet med hjälp av webbt tekniker, som skall kunna köras på flera plattformar [42].

PhoneGap startades av företaget Nitobi och började som ett projekt på utvecklarevenemanget iPhoneDevCamp2008. Under detta evenemang har webbutvecklare samlats tillsammans för skapa nativa iOS applikationer genom att använda olika webbt teknologier. PhoneGap blev populärare och mer välkänt efter att ha vunnit Web 2.0 expo LaunchPad år 2009. Idag går PhoneGap under namnet Cordova efter att Adobe köpte Nitobi år 2011[43].

För närvarande stödjer PhoneGap iOS, Android, Samsung Bada, BlackBerry, Symbian, Windows Phone 7 och HP webOS.

3.3.1.2 Utvecklingsmiljö och verktyg

PhoneGap fungerar mer som ett plug-in till de nativa utvecklingsmiljöerna. Detta innebär att man utan problem kan emulera applikationen med en nativ emulator. Dessvärre finns det begränsade möjligheter för felsökning och enhetstestning av applikationer, de nativa debuggnings verktygen går inte att använda fullt ut, utan man måste använda andra tekniker för detta [44].

För att skapa ett PhoneGap projekt går man tillväga på samma sätt som i ett nativt projekt [45]. Applikationer utvecklas med webbt tekniker såsom HTML, CSS och JavaScript.

Det tillkommer inga särskilda verktyg för PhoneGap utan på deras hemsida har de en lista på inofficiella verktyg som kan användas tillsammans med PhoneGap [46].

3.3.1.3 Informationskällor (Dokumentation, communities)

Officiella API: er, kodexempel och lathundar finns att tillgå från PhoneGaps hemsida. Det finns även officiella Google grupper, IRC, bloggar för att få hjälp och läsa små självstudiekurser.

3.3.1.4 JQuery Mobile

JQuery Mobile är en pekskärm-optimerad webb ramverk för smartphones som är byggt på JQuery och HTML5. Detta ramverk ger utvecklare tillgång till moduler som hanterar olika pek-händelser och grafiska komponenter som är anpassat för mobila enheter [47].

3.3.2 Titanium

3.3.2.1 Introduktion

Titanium lanserades år 2008 av företaget Appcelerator. Denna cross plattformswerktyg är speciellt inriktat mot webbutvecklare. Titanium erbjuder en egen utvecklingsmiljö och använder JavaScript för utveckling av applikationer.

I dagsläget stödjer Titanium iOS, Android och snart BlackBerry.

3.3.2.2 Utvecklingsmiljö och verktyg

Utvecklingsmiljön Titanium Studio med tillhörande Titanium SDK har en egen utvecklingsmiljö med de nativa emulatorerna integrerat. Utvecklingsmiljön är baserad på Eclipse IDE och skrivs med HTML, JavaScript och CSS och stöd finns även för PHP, Ruby och Python. SDK: n innehåller mer än 5000 API: er som bland annat ger åtkomst till nativa funktioner och komponenter till det grafiska gränssnittet [48, 49]. För enhetstestning finns det inget officiellt tillägg, dock existerar det ett flertal tredje parts tillägg som exempelvis verktyget titanium-jasmine [50].

Titanium tillhandahåller en online butik där utvecklare kan sälja och marknadsföra speciella moduler för att lägga till extra funktionalitet som inte ingår i standard API:erna. Exempel på moduler som finns att ladda ner är PayPal som erbjuder betalningsmöjligheter[51].

3.3.2.3 Utvecklaravgifter

Titanium är fri att ladda ner och installera. Det finns även licenser för att få tillgång till extra funktioner och support via chat och telefon [52]. Exempel på extra funktioner är "Analytics" som samlar statistik på vilka funktioner i applikation används mest och hur väl applikation presenterar.

I API:erna ingår inte iOS tjänster som exempelvis Game Center eller Twitter integrering utan dessa finns att köpa som moduler i deras online butik.

3.3.2.4 Informationskällor (Dokumentation, communities)

Officiell dokumentation med API:er, självstudiekurser och kodexempel finns på deras hemsida. Appcelerator erbjuder många videobaserade självstudiekurser för att komma igång med utveckling med JavaScript. Det finns även en officiell "Community Questions and Answers" för att ställa frågor eller läsa tidigare frågeställningar med svar.

3.3.3 Mono for Android

3.3.3.1 Introduktion

Mono for Android eller *MonoDroid* lanserades år 2011 av Xamarin för utveckling av Android applikationer med C# och .NET. År 2009 kom en iOS-version vid namnet MonoTouch ut på marknaden. Båda verktygen är baserade på en öppen källkod version av Microsoft .NET runtime[53].

Utöver stöd för iOS och Android finns även stöd Windows Phone 7.

3.3.3.2 Utvecklingsmiljö och verktyg

Xamarin har både ett egenutvecklat utvecklingsmiljö som kallas för MonoDevelop IDE och en plug-in för Visual Studio. För både utvecklingsmiljön och plug-in:et används nativa emulatorer, stöd finns för felsökning med debugger och enhetstestning.

Xcodes Interface Builder går även att använda för att skapa det grafiska gränssnittet i iOS. Xamarin planerar att släppa en liknade verktyg under namnet Xamarin Designer framöver så att utvecklare kan på samma sätt som i Eclipse skapa grafiska gränssnitt till Android applikationer.

MonoDevelop IDE fungerar att installera för både Windows och Mac.

3.3.3.3 Utvecklaravgifter

Både MonoTouch och Mono for Android krävs en licens för att installera applikationen på en fysisk enhet under utveckling samt för att distribuera i de olika

programbutikerna. Den billigaste licensen kostar 399 USD för ett år och det finns licenser för företag och privatpersoner. Det finns en utvärderingsversion som enbart tillåter att provköra applikationen på nativa emulatorer [54, 55].

3.3.3.4 Informationskällor (Dokumentation, Communities)

Xamarin tillhandhåller dokumentation i form av API: er, självstudiekurser och kodexempel. Utöver detta finns det information om hur man använder iOS specifika tjänster som exempelvis iCloud och den integrerade Twitter-tjänsten för iOS 5 [56, 57]. I Android kan man enkelt tillgå information från Android communities utan att behöva vända sig till Xamarin. Strukturen och namngivning av metoder i både Mono for Android och den nativa är den samma.

3.3.4 Marmalade

3.3.4.1 Introduktion

Marmalade är en cross plattform med fokus på ett brett plattformsstöd och stöd för utveckling av prestanda krävande applikationer [58]. Ursprungligen hette verktyget AirPlay SDK som släpptes år 2009. Företag som bland annat använder detta verktyg är Electronic Arts, Square Enix, Samsung, LG and RIM.

Marmalade lockar främst till erfarna applikationsutvecklare som jobbar med komplexa applikationer[59]. Ideaworks har historiskt sätt alltid inriktats mot spelutvecklare och i Marmalade finns det bibliotek för hantering av 3D grafik, 2D/3D geometri och animationer [59].

Plattformen stödjer iOS, Android, Bada, BlackBerry Playbook OS och LG Smart TV. Det finns begränsat stöd till Symbian i dagsläget.

3.3.4.2 Utvecklingsmiljö och verktyg

Marmalade applikationer utvecklas med hjälp av utvecklingsmiljöerna Microsoft Visual Studio eller XCode via ett nedladdningsbart plugin [60]. Den innehåller bland annat emulatorer, ARM-debugger och UI designer [61].

Det finns även stöd för iOS specifika tjänster så som Game Center, iAD och notifieringar [62].

3.3.4.3 Utvecklaravgifter

Marmalade är inte kostnadsfritt utan kräver licens för användning. Det finns tre typer av licenser:

- Community, för hobby och semi professionella utvecklare [63].
- Indie, för minde företag.
- Professional, för professionella utvecklare.

3.3.4.4 Informationskällor (Dokumentation, Communities)

Marmalade har ett utvecklarnätverk där information rörande installationer, API: er, och vilken funktionalitet det finns för de olika plattformarna. Det finns även att få support och hjälp via ett officiellt forum [64].

3.3.5 MoSync

3.3.5.1 Introduktion

MoSync AB är ett svensk mjukvaruföretag som låter både webbutvecklare och mjukvaruutvecklare att skapa en applikation, med en kodbas som fungerar på flera plattformar [65]. Detta verktyg kan användas för att bygga HTML5 applikationer, hybrid applikationer, C/C++ applikationer och nativa applikationer.

MoSync stödjer för närvarande Android, BlackBerry, iOS, Java ME, Symbian, Windows Mobile och Windows phone.

3.3.5.2 Utvecklingsmiljö och verktyg

MoSync har två utvecklingsverktyg: MoSync SDK och MoSync Reload. Det senare verktyget inriktar sig för webbutvecklare och bygger på samma koncept som PhoneGap fast med mer funktionalitet och tillgång till nativt gränssnitt [66]. Applikationerna utvecklas med HTML5 eller JavaScript och verktyget kan installeras i Linux, Mac eller Windows [67, 68].

I MoSync SDK ingår utvecklingsverktyget MoSync IDE som är baserad på Eclipse IDE. Utöver det som erbjuds i den ordinarie versionen av Eclipse, har MoSync integrerat funktioner som bland annat kan överföra applikationen via Bluetooth till en mobiltelefon[69]. MoSync erbjuder förutom användning av de nativa emulatorerna även en egen emulator vid namnet "MoRE" [70]. Det finns även möjlighet att integrera SDK: n i Visual Studio[70].

MoSync erbjuder även en unik teknik som möjliggör utvecklare att skriva funktioner med C/C++ kod och via ett bibliotek sedan anropa dessa funktioner i JavaScript. Denna teknik kallas "Wormhole technology" [71].

MoSync stödjer förvävarande iOS 3.0 - 4.3 och Android 1.5 - 2.3.3. De har inte stöd för populära iOS specifika tjänster som iCloud eller Game Center [72].

3.3.5.3 Utvecklaravgifter

MoSync är en öppen källkods företag och erbjuder samtliga verktyg kostnadsfritt [73].

3.3.5.4 Informationskällor (Dokumentation, Communities)

Dokumentation med lathundar och kodexempel finns tillgänglig på deras hemsida [74]. Det finns ett officiellt forum för att ställa frågor och läsa andras frågeställningar.

3.3.6 Corona (Anscas Mobile)

3.3.6.1 Introduktion

Corona är en cross plattform som är speciellt inriktad på spelutveckling. Corona SDK lanserades år 2009 av Anscas Mobile, ett företag som har en mångårig erfarenhet av både Adobe och Apple applikationer. Corona SDK är tänkt som ett alternativ till Adobe Flash [75].

3.3.6.2 Utvecklingsmiljöer och verktyg

Corona SDK erbjuder god felsöknings- och emuleringsmöjligheter. SDK:n innehåller egenutvecklade verktyg för bland annat simulering av både iOS och Android applikationer och även felsökning [76].

Corona SDK är enbart en utvecklings kit och innehåller inte en fullständig utvecklingsmiljö. Det existerar tredje parts verktyg som exempelvis Corona Project Manager som kan användas som ett IDE tillsammans med Corona SDK [77]. Utvecklingsspråket är Lua och används för att skriva skript i spel som exempelvis spelet World of Warcraft [78].

I API:erna finns det stöd för exempelvis animationer och fysikmotor som är användbart i spelutveckling. Dessutom finns det stöd i API:erna för Apple specifika tjänster som bland annat in-app purchase, iCloud backup och Game center [79].

3.3.6.3 Utvecklaravgifter

Corona är inte kostnadsfri och det finns tre former av licenser att välja ifrån [80].

- Indie iOS only 199 USD/år
- Indie Android only 199 USD/år
- PRO, stödjer utveckling för båda plattformarna 349 USD/år

Gemensamt för alla licenser så ingår det bland annat tillgång till speciella forum och distribution av applikationer i respektive nativ programbutik.

3.3.6.4 Informationskällor (Dokumentation, Communities)

Anscas Mobile erbjuder API:er, allmän information, design mallar och videobaserade kurser för Corona SDK. Det finns även en officiell community där man kan ställa

frågor i forumet och även dela kod med andra användare [81]. Det existerar även ett flertal etablerade i communities.

3.3.7 Jämförelse av cross plattformar

Nedan är en sammanfattning av alla cross-plattformar som ingår i detta kapitel:

	PhoneGap (Cordova)	Titanium	MonoTouch MonoDroid	Marmalade	MoSync	Corona
Företag	Nitobi (uppköpt av Adobe år 2011)	Appcelerator	Novell (Xamarin uppköpt år 2011)	Ideaworks3 D Ltd.	MoSync	Anasca mobile
Grundat (år)	2008	2007	1979	1998	2004	2008
Lanseringsår	2008	2009	2009 (Mono- Touch)	2009	2011	2009
Bakomliggan de tekniker	Web-to-native wrappers	Runtime	Runtime	Source code translator	Source code translator	Runtime
Utvecklingssp råk	HTML5, CSS, JS	HTML5, CSS, JS	C#, .NET	C/C++	C++, HTML5, CSS, JS	Lua
Utvecklings- miljö	Webb utvecklings- verktyg/Ordb ehandlare	Titanium Studio	Mono- Develop IDE/Visual Studio/ XCode	Visual Studio/XCod e	MoSync Reload/MoSync IDE/Visual Studio	Ordbehandlare
Applikations format	hybrid applikation	nativ, webb- och hybrid applikationer	nativ	nativ	nativ/hybrid applikationer	nativ
Mål applikationer	Fulländade web applikationer	Fulländade information och media applikationer	Applikatione r för företags- verksamhet	Spel inspirerade applikationer	Komplexa applikationer	Spel inspirerade applikationer
Distributions kostnad (min)	Kostnadsfri	Kostnadsfri (exkl. moduler)	399 USD per plattform per år	140 USD/år	Kostnadsfri	199 USD per plattform per år
Android	JA	JA	JA	JA	JA	JA
iOS	JA	JA	JA	JA	JA	JA
Bada (Samsung)	JA	NEJ	NEJ	JA	NEJ	NEJ
BlackBerry	JA	JA	NEJ	JA (QNX)	NEJ (QNX)	NEJ
Symbian (Nokia)	JA	NEJ	NEJ	JA	JA	NEJ
Windows Phone 7	JA	NEJ	NEJ	NEJ	JA	NEJ
WebOS (HP)	NEJ	NEJ	NEJ	NEJ	NEJ	NEJ

Tabell 5: Fakta sammansättning av cross plattform verktygen.

4 GENOMFÖRANDE

4.1 Sammanställning av kravspecifisering för val av cross plattformar

4.1.1 Introduktion

Examensarbetet togs fram av It-företaget Spinit AB som vill expandera sitt företag genom att skapa en ny avdelning för mobil applikationsutveckling. De har vid ett tidigare skede utvecklat en iPhone applikation för en kund som har visat sig vara populär och företaget har fått en förfrågan av kunden att skapa en identisk applikation till Android. De inser att framtida kunder kommer att vilja ha applikationer för både iOS och Android. Därav vill de att vi ska finna en cross plattforms verktyg som följer ordspråket “write once, run everywhere”. Detta verktyg skall även kunna skapa kommersiellt konkurrens kraftiga applikationer.

4.1.2 Undersökning av beställarföretagets verksamhet

En undersökning har utförts av företagets verksamhet för att bilda en uppfattning om vilka programmeringsspråk och integrerade utvecklingsmiljöer som används. Detta skall vara en riktlinje för att välja ut cross plattformar som bäst överensstämmer med utvecklarnas kompetens och förväntningar.

Undersökningen har genomförts genom att tala med utvecklarna i företaget. Nedan är en beskrivning av vilka programmeringsspråk och utvecklingsmiljöer företaget använder:

- **Utvecklingsmiljöer:** Microsoft Visual Studio(.NET) och Apple XCode
- **Programmeringsspråk:** Objective-C, Java, C#, HTML, CSS och HTML5

Det visade sig att företaget är Microsoft-inriktad och att majoriteten av utvecklare använder Visual Studio och särskilt med C#. Det finns ett starkt önskemål att kunna utveckla iOS i Windows-miljö.

4.1.3 Analys av etablerade cross plattformar

En analys av etablerade cross plattformar har gjorts för att skapa en uppfattning av vilka funktioner och övriga tjänster som finns. Analysen visade på att det finns ett större utbud av cross plattformar än förväntad och många riktade sig till fler målgrupper än den traditionella gruppen av mjukvaruutvecklare. För att minska urvalet behövdes det framställas krav för vilka cross plattformar som är lämpligast i denna rapport.

För att skapa en uppfattning av vilka verktyg som skall finnas i en cross plattform har en undersökning gjorts i de nativa utvecklingsmiljöerna för att identifiera gemensamma verktyg och funktioner som finns i både iOS och Android. Tabellen

nedan är sammanställning av de vanligaste tjänsterna som erbjuds i båda utvecklingsmiljöerna.

Tjänster	Android	iOS
Emulering av applikationen	Android Emulator	iOS Simulator
Utvecklingsspråk	Java	Objective-C
Verktyg för design av GUI	Android UI Designer	XCode Interface Builder
Felsökningssystem	Android DDMS	GDB Debugger
Test ramverk	Flertal, både manuella och automatiska, exempel JUnit for Android och Robotium	XCode Unit testning, testning för exceptions och även automatiskt UI testning i Instruments finns

Tabell 6: sammanställning av de vanligaste tjänsterna som erbjuds i båda utvecklingsmiljöerna.

Det visade sig att många utav cross plattformarna inte stödjer någon form av emulering av applikationen eller har något felsökningssystem. De etablerade cross plattformarna som stödjer dessa tjänster valdes att analysera vidare var följande: Marmalade, Corona SDK, Titanium, MoSync, PhoneGap, MonoTouch och Mono for Android.

Nästa steg är att analysera dokumentationen för de utvalda plattformarna och vi har inriktat oss på stödet för underliggande funktionaliteten. Nedan är en sammanställning av de vanligaste förekommande enhetsspecifika funktioner i smarta mobiler som stöds eller inte stöds av de valde plattformarna.

	Marmalade	Corona SDK	Titanium	MoSync	Mono for Android	PhoneGap
Kamera	JA	JA	JA	JA	JA	JA
Ljudinspelning	JA	JA	JA	JA	JA	JA
Ljud uppspelning	JA	JA	JA	JA	JA	JA
PIM Kontakter	JA	NEJ	JA	JA	JA	JA
Bluetooth	-	NEJ	NEJ	JA	JA	NEJ
GPS	JA	JA	JA	JA	JA	JA
Accelerometer	JA	JA	JA	JA	JA	JA
SMS	JA	JA	JA	JA	JA	NEJ

Vibrator	JA	JA	JA	JA	JA	JA
-----------------	----	----	----	----	----	----

Tabell 7: Översikt av alla cross plattformars stöd till de enhetsspecifika funktionerna.

Tabellen ovan visar att alla cross plattformarna har ett brett stöd till de enhetsspecifika funktionerna.

4.1.4 Motivering till val av cross plattformar baserat på krav

Utifrån våra undersökningar valdes plattformarna Titanium, MoSync och Mono for Android. För att välja ut tre plattformar för utvärderingen valde vi att gå efter två tillvägagångssätt nämligen “write once, run everywhere” samt det andra sätt som innebär att man använder samma programmeringsspråk och utvecklingsmiljö. Både Titanium och MoSync bygger på “write once, run everywhere” medan Mono for Android bygger på det senare. Corona SDK var även en tänkbar kandidat istället för Titanium men Corona SDK är mer spelinriktad. Nedan är mer ingående motivering av plattformarna.

Titanium: Beställarföretagets verksamhet är mycket inriktad mot webblösningar och majoriteten av utvecklarna är webbutvecklare. Titanium använder webbt teknologier för kodning och har en stark dokumentation med många API: er och självstudiekurser. Vi anser att detta val skulle led till en naturlig övergång för deras webbutvecklare.

MoSync: MoSync erbjuder en intressant lösning som möjliggör att mjukvaruutvecklare och webbutvecklare kan använda samma utvecklingsmiljö. Man har möjlighet att utveckla med både C/C++ eller webbt tekniker. De har även utvecklat en speciell teknik som kallas för “wormhole teknologi” där kan man använda funktioner i MoSync C/C++ biblioteken när det blir hinder vid användning av webbt teknikerna.

Mono for Android: Mono for Android skiljer sig från andra cross plattformar som inriktar sig på att skapa lösningar som bygger “write once, run everywhere”. De har istället valt att skapa en plattform som använder C# för utveckling av Android, iOS och Windows Phone 7 dock utvecklas applikationen precis på samma sätt som i en nativmiljö. Xamarin erbjuder samma funktionalitet och tjänster som finns i de nativa API: erna. Gemensamt för iOS, Android och Windows Phone 7 är att det finns standard bibliotek som kan användas för att skapa plattformsberoende kod för logik och databas.

4.1.5 Korrigering under utvärderingen

Vi valde efter kortare användning av MoSync att övergå till PhoneGap istället. Vi ansåg att utvecklingstiden av applikationen skulle ta längre tid än planerat eftersom inlärningskurvan är betydligt högre med MoSync. MoSync använder C/C++ vilket kräver en god förståelse om hantering av minnesläckor. Ett försök gjordes även med att använda den webbaserade MoSync Reload som använder HTML5, dock så upptäcktes det snabbt att vissa funktioner som exempelvis att byta färg på bakgrunden inte fungerade.

Vi valde att fortsätta med PhoneGap som ger fler valmöjligheter till design av användargränssnittet. PhoneGap är ett ramverk som endast tillhandahåller åtkomst till underliggande funktionalitet och har inga moduler för uppbyggnad av det grafiska gränssnittet. Det grafiska gränssnittet kan enkelt skapas själv. Ett flertal grafiska ramverk existerar med möjlighet till användning av speciella teman. Vi valde att använda JQuery Mobile för att man enkelt kan bygga robusta och användarvänliga mobilanpassade webbsidor.

4.2 Applikationen

4.2.1 Krav

Att framställa rimliga krav kräver en god balans mellan vår disponerade tid och de aspekter vi vill undersöka. Målet med applikationen är att försöka hålla en kommersiell standard som följer den vanliga arbetsprocessen innefattande av design och user stories. Applikationen skall vara konstruerad så att den har en enkel och robust struktur som kan vara ett bra underlag till utvärderingen. Följande krav har fastställts:

- Det grafiska gränssnittet skall vara enkel och användarvänlig.
- Applikationen skall utnyttja kameran för bild tagning, ljudinspelning och GPS.
- Applikationen skall även kunna läsa och skriva till det interna eller externa lagringsmediet.

4.2.1.1 Grafiska gränssnittet

Det grafiska gränssnittet skall ha en enkel utformning så att mer tid kan ägnas på den logiska delen av applikationen. Det grafiska gränssnittet kan vara ett nativt gränssnitt.

4.2.1.2 Underliggande funktionalitet

Beställarföretaget vill att vi provar om kameran, ljudinspelning och GPS kan utnyttjas i applikationen. Detta är tänkbare funktioner som framtida kunder kan tänka sig använda i deras applikationer.

4.2.1.3 Utvecklingsprocessen

Som nämnts i avgränsningen så kommer delmoment som exempelvis användning av versionshanteringsystem eller enhetstestning inte att ingå. Dessa moment är inte relevanta i undersökning och är utanför utvärderingens syfte.

Applikationen skall utvecklas i både Windows- och Macmiljö för att se om det finns några skillnader eller begränsningar. Exempelvis kommer vi titta på om verktyg eller någon annan form av underlättande utvecklingstjänster saknas.

4.2.2 Konceptet DigitalDiary

Idén är att skapa en enkel digital dagbok där användaren kan skriva ett inlägg i dagboken och lägga till ett foto, sin position och ett inspelat ljudklipp. Dagboken skall vara konfidentiell så att obehöriga inte kan få tillträde till inläggen i mobilen.

4.2.3 User stories

Vattenfallsmetoden kräver att man skapar en detaljerad dokumentation med krav och testfall. Scrum metoden inriktar sig istället på att skapa enkla så kallade “user stories”, detta är olika händelser ur användarens synvinkel. Dessa är prioriterade efter relevans i fallande ordning.

As a user I want to create new entry into one of my diaries.

As a user I want to delete an entry so that I remove it from my diary

As a user I want to access previous entries so that I can read old entries.

As a user I want to play the sound recordings in every entry so that I can listen to previous recordings.

As a user I want to use the built-in camera to take pictures so that I can added it to the current entry and also be able to view it.

As a user I want to record my voice or sounds in my surrendering so that I can add it to the current entry and be able to listen to it later.

As a user I want to use GeoLocation so that I can see locations I have visited or my current position.

As a user I want to have several diaries so that I can have different categories of diaries or a new diary each year.

As a user I want to create a diary so that I can add entries.

As a user I want to delete a diary containing all my entries.

As a user I want to have a great overview of my entries of each diary so that I easily can select an entry and read the content of it.

As a user I want to edit previous entry so that I can add or remove information.

As a user I want to have a calendar view so that I can select a date and view a summary of the entries made that day from all the diaries.

As a user I want to login to my DigitalDiary so that I can access my entries with confidentiality.

4.3 Implementation av applikationen i varje cross plattform

4.3.1 Struktur, logik och grafiska gränssnittet

Applikationen skall implementera ett grafiskt gränssnitt med tabbar. Nedan är en kort introduktion om varje tabb: s uppgift:

- **Diaries:** Denna tabb skall innehålla en enkel lista med alla dagböcker som finns i systemet. Användaren skall kunna lägga till, redigera och radera dagböcker. Användaren skall även här välja vilken dagbok som skall skrivas eller läsas ur.
- **New Entry:** Här skall användaren kunna skapa inlägg för den valda dagboken i Diaries. Användaren skall kunna ta en bild med kameran, göra ljudinspelning och spara sin nuvarande position.
- **Entries:** Användaren skall kunna läsa och redigera tidigare inlägg från den valda dagboken i Diaries. Varje inlägg innehåller eventuellt bild, ljudklipp och position.

Databas

Dagböcker och dess inlägg kommer att lagras i en intern databas. De valda cross plattformarna har stöd för interna databaser. Både Titanium och Mono for Android använder SQLite databas, och PhoneGap använder W3C Web SQL Database.

Databasen kommer att bestå av två tabeller. En tabell kommer att vara avsedd för att lagra inlägg i dagboken. Då användaren har möjlighet att ha flera dagböcker kommer namngivningen av tabellen att vara dagbokens namn. Den andra tabellen kommer att hålla reda på antalet dagböcker. Ur en teknisk synvinkel så behövs inte denna tabell eftersom denna information redan finns tillgänglig i databasen. Dock så kan man inte hämta information direkt, utan man måste filtrerar bort annan irrelevant information. Därför skall denna tabell användas.

Nedan är den planerade relationen av databasen:

NameOfDiary (date, title, story, image, sound, longitude, latitude);

Diaries (nameofdiary);

Attributen "date" är en primär nyckel eftersom att användaren enbart kan göra ett inlägg per dag. NameOfDiary är titeln på den skapade dagboken.

Datalagring

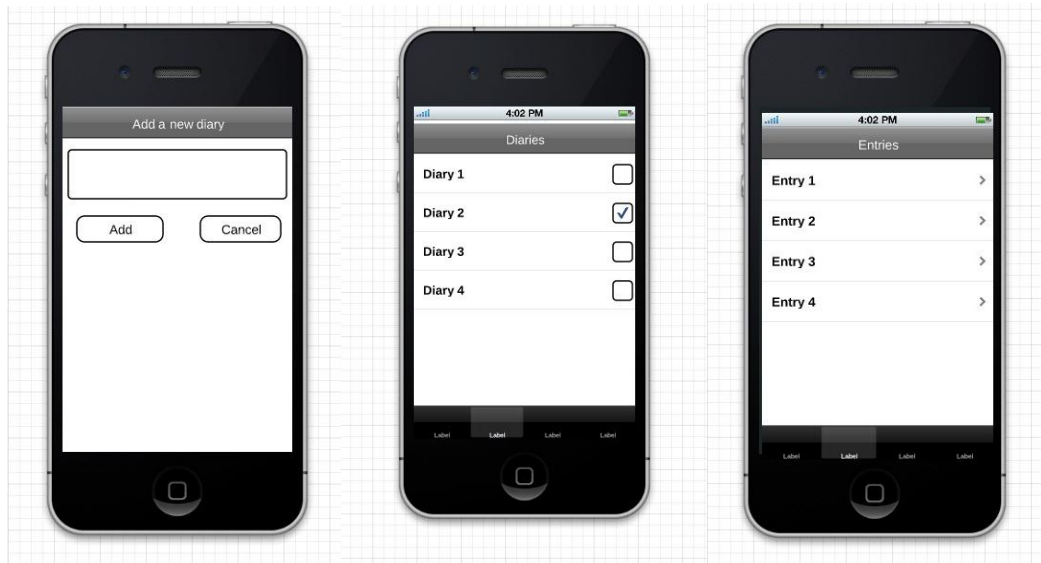
Filer i applikationen såsom kamerabilder, ljudklipp kommer att sparas på det externa minnet.

I databasen kommer filens sökväg att lagras. De valda plattformarna har stöd för detta och även stöd för lagring av primitiva datatyper.

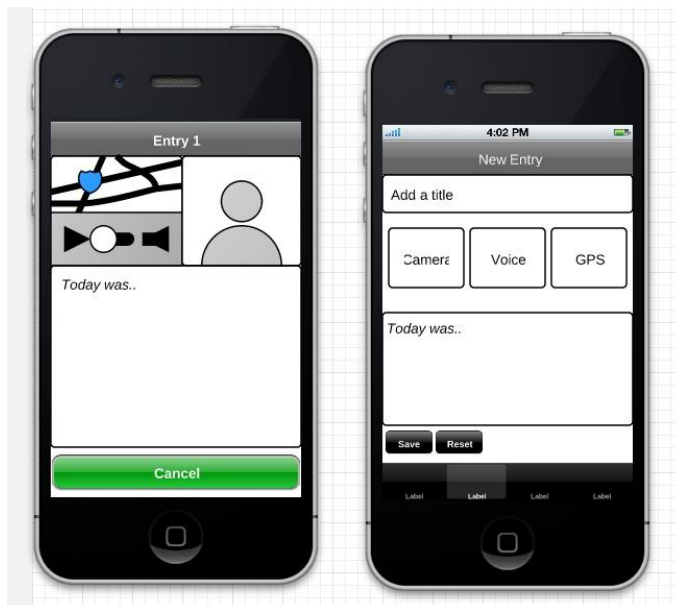
Möjligheter för säker inloggning varierar mellan de valda plattformarna. I både PhoneGap och Titanium finns det inget stöd för kryptering av data i API: erna. Det finns dock en plugin att köpa till i Titanium för AES-kryptering. I Mono for Android finns det stöd för symmetrisk och asymmetrisk kryptografi i API: et. I denna applikation kommer säkerhetsaspekten att inte ha hög prioritering.

Grafiska gränssnittet

Här är idé skisser på utseendet av applikationen. Denna utformning kommer att användas som riktlinje i utvecklingen av applikationerna i respektive plattform.



Figur 6: (Fr. väster) Fönster för att lägga till att en dagbok, nästa är Diaries och sist Entries.



Figur 7: (Fr. vänster) Fönster för att presentera innehållet i ett inlägg, nästa är New Entry.

4.3.2 Hårdvara

De datorer och smarta mobiler som kommer att användas i studien beskrivs nedan:

Windows dator:

Processor: Intel Core 2 Duo Solo CPU 1,4GHz

RAM: 8GB DDR3

OS: Windows 7 Professional 64-bit

Mac dator:

Processor: Intel Core 2 Duo 2.16GHz

RAM: 4GB DDR2 667Mhz

OS: Mac OS X 10.7.4 (Lion)

Android telefon:

Modell: Samsung Galaxy Spica i5700

Processor: ARM11 800MHz

RAM: 128 MB

OS: Android 2.2.1

5 RESULTAT

5.1 Introduktion

Detta kapitel ämnar ta upp de tester som har utförts under utvärderingen. De funktioner som hade låg prioritet i User Story sektion 4.2.3 implanterades inte på grund av tidsbrist.

Gemensamt för alla cross plattformar, implementerades ingen möjlighet för redigering av titeln till en skapt dagbok eller säker inloggning. Underliggande funktionalitet som exempelvis kamera kunde inte provas på en fysisk iOS enhet utan vissa funktioner som ljudinspelning och positionering kan göras i en iOS simulator.

Utveckling med iOS applikationer i Windows miljö med cross plattformar var inte möjligt. Det går att skapa kod som kan exekveras i en iOS-enhet men det går inte att simulera eller felsöka applikationen.

En motsvarig applikation till iOS med användning av MonoTouch utfördes inte på grund av tidsbrist. Ingen gemensam kodbas skapades i Mono for Android, som skulle kunna användas i MonoTouch.

I Appendix B, C och D innehåller kod till samtliga applikationer som har skapats.

5.2 Resultat av implementering av applikationen i respektive cross plattform

5.2.1 Diaries

PhoneGap

Möjligheten att lagra eller ta bort dagböcker samt välja vilken dagbok man ville använda fungerar i iOS och Android miljö. Det förekommer i enstaka fall att fönstret för att lägga till en ny dagbok blir blank i både Android emulator och fysiska enheten.

Titanium

Denna del av applikationen har implementerat möjligheter att lägga till eller ta bort dagbok samt att välja den dagbok man vill använda. Detta fungerar i både iOS och Android utan problem.

Mono for Android

Denna del utav applikationen har implementerats och fungerar som tänkt.

5.2.2 New Entry

PhoneGap

Användning utav enhetens interna funktioner så som kamera och positionering gick att implementera samt spara inlägget i databasen fungerar. Dock så implementerades inte ljudinspelning och detta beror på att applikationen blev mer instabil när fler funktioner implementerades under arbetsgången. I Android så fungerar det ovan nämnda bra förutom att ikonerna i Diaries ändrades till samma som för New Entry i vissa fall, detta problem uppstår inte i iOS.

Titanium

Kamera, ljudinspelning och positionering implementerades och fungerar. Ljudinspelningen fungerar dock inte på Android.

Mono for Android

Alla delar tillhörande New Entry är implementerad och fungerar som tänkt.

5.2.3 Entries

PhoneGap

I iOS fungerade listning utav alla inlägg i fönstret, dessvärre uppstod det allt för ofta en blank skärm, både i emulaton och på den fysiska enheten i Android. Att presentera information för ett valt inlägg har implementerats men är dock inte komplett eller fungerade. Det uppkommer även här att ikoner blir felaktiga i tabb panelen i Android.

Titanium

Listning av alla inlägg för en vald dagbok har implementerats och fungerar. Presentation av innehållet i inlägget fungerar, dock återstår implementering av kartfunktion för att se den sparade position och uppspelning av det inspelade ljudet.

Mono for Android

Till skillnad från Titanium och PhoneGap så har samtliga funktioner som nämnts tidigare implementerats och fungerar väl.

5.2.4 Datalagring

PhoneGap

Den interna databasen implementerades och fungerar enligt de utsatta kravspecifikationerna. Bilder går att spara och texten tillhörande ett specifikt inlägg sparades i databasen utan att först omvandla till en text fil. Att spara primitiva datatyper fungerar även.

Titanium och Mono for Android

Interna databasen, lagring av bilder, ljudinspelning, text till fil samt att spara primitiv data har implementerats och fungerar.

5.3 Utvärdering av respektive cross plattform

5.3.1 Appcelerator Titanium

Intrycket vi fick av Titanium var att det var ett kompetent och kraftfullt verktyg avsett för webbutvecklare. Den är kraftfull i den meningen att man kan skapa enkla och robusta nativa applikationer utan problem, dock är den inte lämplig om man är ute efter att göra mer resurskrävande uppgifter som exempelvis bildprocessering.

Specifika tjänster som erbjuds inom nativa plattformar som exempelvis Twitter integrering och GameKit för Game center finns att köpa som plug-in i deras online butik. Vi tycker att detta är negativt eftersom detta erbjuds kostnadsfritt för nativa plattformar. Deras online butik erbjuder förutom detta andra användbara plugin som DropBox eller Paypal.

Inlärningskurvan är låg eftersom Appcelerator tillhandhåller en väldokumenterad dokumentation, videobaserade självstudiekurser och programexempel för snabbare

inläring och förståelse. Det vi ansåg vara speciellt bra är deras Kitchen Sink exempel som är en applikation där all funktionalitet som erbjuds i API: erna finns implementerat för demonstration och man kan få förståelse för hur man skall göra det på egen hand. Det som var negativt med dokumentationen är att informationen är otydlig om en specifik plattform stöds eller inte vid implementering av en funktion. Detta blev ett problem när vi skulle implementera ljudinspelning i applikationen, det fungerade för iOS men dock inte för Android. Det fram gick senare efter att ha läst på ett flertal forum att detta inte stöds. En annan aspekt är strukturen av applikationen, man får en känsla av att det inte finns ett standard sätt att utveckla med JavaScript. I vissa videoexempel eller kodexempel så tillämpas ett äldre sätt där man använder namespaces, för strukturering av koden vilket skapar förvirring. För en nybörjare är det svårt att konvertera om det till det nya sättet commonJS. Utveckling med JavaScript var lite problematisk i början eftersom man vanligtvis inte gör applikationer med detta språk, men det visade sig vara ganska kraftfull och man slipper mycket av komplexiteten som förekommer i en nativ miljö. Det går förhållandevis snabbt att skapa applikationer i Titanium.

Det som förlängde utvecklingstiden avsevärt var kompilering- och fel hanteringstiden. Titanium visar felmeddelande efter att applikationen har laddats upp i enheten och påbörjat sin exekvering. Detta är besvärande när det kommer till att utveckla för Android plattformen där det är känt sedan tidigare att emulatorn tar lång tid att starta. Många av felmeddelandena man får är i vissa fall svåra att tyda och ger väldigt lite information om vad som faktiskt är fel. Det blir speciellt problematiskt när man exempelvis ska undersöka uppkomsten av felet i dokumentation där det är oerhört svårt att finna information. Många utav metoderna eller modulerna i API: erna fungerar som 'black box', det vill säga man vet inte riktigt vilka underliggande processer som görs eller hur koden ser ut.

Ett annat stort problem som var besvärande under utvecklingsprocessen var placeringen av de grafiska komponenterna. Detta ställde till problem när man använder samma konfiguration i en annan plattform. I många av fallen så behövdes det specificeras flera egenskaper för att de grafiska komponenterna ska kunna placeras rätt i applikationen för den andra plattformen. Detta kan bli problematiskt om man enbart utvecklar på Windows och gör en Android applikation som skall fungera på en iOS enhet.

5.3.2 Mono for Android

Den här cross plattformen anser vi erbjuder samma funktionalitet som finns i Android API: erna. Denna plattform tillsammans med MonoTouch använder ett gemensamt programmeringsspråk så att kodbasen kan återanvändas utan större ansträngningar. Den stora fördelen är att man kan utnyttja sina tidigare kunskaper i Android eller iOS och bygga applikationer på samma sätt. Vi har tidigare utvecklat med Android och arbetsprocessen var väldigt lik nativ Android utveckling och inlärningskurvan var därav mycket låg.

Dokumentationen är komplett och man förlorar inga funktioner eller finesser. Det framgår tydligt från deras hemsida, vad som stöds eller inte stöds. Detta underlättar väldigt mycket vid start av ett nytt projekt. De erbjuder inte samma mängd kodexempel eller självstudiekurser som exempelvis Titanium, utan något som var väldigt bra är att man kan utgå från nativa kodexempel eller självstudiekurser och

göra på samma sätt med C#. Denna möjlighet gör att man kan ta del av information från två communities och man behöver inte förlita sig enbart på Xamarin.

Det vi tycker är negativt med den här cross plattformen är att den är väldigt kostsam för en enskild utvecklare, då man behöver köpa en utvecklingslicens per plattform per år. Det finns få exempel och lite dokumentation för hur man skall skapa kodbasen som kan användas på flera plattformar eller strategier för detta. Det var svårt att identifiera områden i vår kod som man kan göra till plattformsoberoende.

5.3.3 Cordova PhoneGap och JQuery Mobile

Inlärningskurvan för den här cross plattformen var väldigt låg då man i själva verket skapar en mobil hemsida som har åtkomst till interna resurser. JQuery Mobile ger ett användarvänligt och trevligt grafiskt utseende som liknar det nativa. Det är väldigt lätt att skapa hemsidorna och placera de grafiska komponenterna.

Applikationen i detta fall kunde inte slutföras, detta berodde på att för varje funktion som lades till i applikationen försämrades prestandan avsevärt. Detta nådde till en punkt där sidorna blev blanka. Orsaken till detta är fortfarande oklart, vi kunde inte hitta orsaken till detta eller prova alternativa lösningar då tiden inte räckte till. Vi har provat ett flertal applikationer som vi hittat i Google Play på vår fysiska enhet. Dessa har innehållit flera funktioner och haft bra prestanda. Det vi vill prova är att se om det är PhoneGap tillsammans med JQuery Mobile som försämrar prestandan eller om det i själva verket är den fysiska enheten som inte klarar av det. Det märktes tydligt att applikationen fungerade betydligt smidigare och bättre på iOS till skillnad från Android. Detta tror vi har med att iOS har bättre stöd än Android för HTML5.

Vi anser att PhoneGap är betydligt mer lämpat att användas i en klient- och serverarkitektur och inte för att lagra information internt i mobiltelefonen. I många kodexempel är tänket inriktat på att information såsom kamerabilder och kontakter skall överföras till en server.

6 DISKUSSION OCH SLUTSATS

6.1 Analys och Diskussion

När vi påbörjade detta examensarbete trodde vi att cross plattformar levde upp till ordspråket "*write once, run everywhere*". Detta är ett sant påstående för vissa utvecklingsverktyg, men de flesta andra verktygen kräver att man gör anpassningar för varje mobil plattform. Ett exempel som stödjer ordspråket är PhoneGap där tekniken bygger på att man använder webbsidor i sin applikation, däremot i Titanium är man tvungen att ta hänsyn till olikheter som finns hos varje plattform.

Marknadsföringen av cross plattformar ger ett intryck av att det mesta är möjligt med dessa verktyg och man skapar en förtroende ingivande fasad, vilket är klart vilseledande. Exempelvis på detta är Mono for Android där man inte kan använda bockar för att markera i en listview, utan man är själv tvungen att skapa en egen lösning. När det gäller begräsningar i API: er är det väldigt svårt att hitta tillräcklig information, utan dessa upptäcks under applikationsutvecklingens gång. I Titanium förekommer ingen information om att ljudinspelning inte stöds i Android i deras officiella API: er.

De flesta cross plattformar använder sig av en "minsta gemensamma nämnare", vilket innebär att applikationsutvecklaren inte kan utnyttja den underliggande funktionaliteten maximalt eller de specifika tjänster som erbjuds av nativa plattformar. I PhoneGap finns det enbart tre metoder avsedda för positionering, medan det i jämförelse med Androids egen API, erbjuds denna fler funktioner och kontroller för positionering. Detta beror främst på att hanteringen av positioneringen skiljer sig mellan iOS och Android. Så även om en cross plattform påstår sig stödja en funktion innebär det inte att man kan utnyttja dess potential till fullo.

Under vår research för att hitta lämpliga cross plattformar visade det sig att ett stort antal av plattformarna var så bristfälliga att de saknade viktiga verktyg och emulatorer. Många kändes inte redo att användas för fullskalig utveckling. Detta påstående får medhåll från boken Vision Mobile Cross-Platform Developer Tools som hävdar att knappt 10 procent av utvecklarna försätter förbi punkten att skapa en "hello world"-applikation [82].

Det är även viktigt att påpeka att många utav cross plattformarna är tänkta att fungera mer som ett komplement tillativ utveckling och vill ge möjlighet till andra målgrupper att skapa enklare applikationer. Detta har medfört till en ökad demokratisering av applikationsutvecklingen för smarta mobiler. Många cross plattformar är ute efter andra målgrupper än mjukvaruutvecklare och man ser en stor marknad för webbutvecklare och personer som saknar kunskaper i programmering.

Begränsningarna hos cross plattformarna beror inte enbart på företagen bakom dessa, utan de beror till stor del på att de som äger de nativa utvecklingsmiljöerna försöker på olika sätt förhindra att de utvecklas vidare. Apple har begränsat utvecklingsmiljön och dess verktyg till endast Mac OS X. Detta märks även av i cross plattformar eftersom man inte kan simulera eller felsöka iOS applikationer i Windows. Detta kan även ha med Apples inställning till cross plattformar och medgrundaren Steve Jobs gav sin åsikt i ett öppet brev med namnet *Thoughts on Flash* med följande innehåll där han öppet kritiserar tredje parter.

" We know from painful experience that letting a third party layer of software come between the platform and the developer ultimately results in sub-standard apps and hinders the enhancement and progress of the platform. If developers grow dependent on third party development libraries and tools, they can only take advantage of platform enhancements if and when the third party chooses to adopt the new features. We cannot be at the mercy of a third party deciding if and when they will make our enhancements available to our developers. This becomes even worse if the third party is supplying a cross-platform development tool. The third party may not adopt enhancements from one platform unless they are available on all of their supported platforms. Hence, developers only have access to the lowest common denominator set of features. Again, we cannot accept an outcome where developers are blocked from using our innovations and enhancements because they are not available on our competitor's platforms. Flash is a cross-platform development tool. It is not Adobe's goal to help developers write the best iPhone, iPod and iPad apps. It is their goal to help developers write cross-platform apps. And Adobe has been painfully slow to adopt enhancements to Apple's platforms. For example, although Mac OS X has been shipping for almost 10 years now, Adobe just adopted it fully (Cocoa) two weeks ago when they shipped CS5. Adobe was the last major third party developer to fully adopt Mac OS X. [40]. "

I denna redogörelse påpekade han att tredje parter alltid ligger steget efter när nya tjänster lanseras och detta kan man se tydligt hos vissa cross-plattformar eftersom de inte har stöd för alla Apples iOS-tjänster eller som exempelvis med MoSync som inte har stöd för den senaste versionen av operativsystemet som är iOS 5.

6.2 Slutsats

Den frågan vi ställde i början av rapporten: "*Finns det någon kraftfull universal utvecklingsmiljö som kan förenkla utvecklingen av applikationer samt portera dessa till flera mobila plattformar samtidigt?*" Denna fråga kan inte besvaras med ett enkelt ja eller nej. Delvis ja, eftersom de cross plattformar vi använt klarar utmärkt att skapa enklare kommersiella applikationer. Ett tillämpningsområde är klient- och serverapplikationer, med användning av dessa verktyg kan utvecklingstiden reduceras. Detta passar bra för andra målgrupper förutom mjukvaruutvecklare eftersom man här har en chans att skapa mobila applikationer, vilket inte är lätt med de nativa utvecklingsverktygen.

För mjukvaruutvecklare anser vi att man borde kunna använda dessa verktyg som komplement, i de fall man inte arbetar med resurskrävande applikationer med krav på hög prestanda. I dagsläget bör applikationsutvecklare inte ställa om till att enbart arbeta med cross plattformar. Om man vill arbeta med dessa verktyg bör man i första hand innan projektet påbörjas noggrant undersöka om alla funktioner och finesser som

skall finnas i applikationen verkligen stöds i API: erna. Det visade sig att stödet för små detaljer som en liten bock i en lista varierade mellan de utvärderade cross-plattformarna. Detta förarbete kan minimera utvecklingstiden avsevärt.

En annan aspekt som vi anser viktigt vid val av cross plattformar om man vill arbeta med detta är att identifiera vilka slags kunder man utvecklar applikationer åt. Är applikationerna inriktade mot webbtjänster eller enklare typer, så kan ett cross plattform verktyg som Titanium eller PhoneGap var utmärkt att använda. Däremot om man jobbar med applikationer som kräver hög prestanda eller spelinriktade är Marmalade och MoSync utmärkta då man arbetar med C++. Den enda nackdelen som även nämns i boken *Vision Mobile Cross-Platform Developer Tools* är att man bör ha många års erfarenheter av C/C++ programmering för att kunna utnyttja potentialen i dessa verktyg. Det visade sig även att inlärningskurvan är hög för utvecklare som endast har kunskaper i de moderna språken som Java och C#.

Utav de cross plattformar vi undersökt anser vi att den bästa för mjukvarutvecklare som arbetar med nativutveckling i både iOS och Android är Mono for Android. Denna plattform erbjuder samma funktionalitet och finesser som finns i Android SDK. Likadant är det för MonoTouch som är motsvarigheten till iOS. Dessa verktyg gör att man behöver lära sig minimalt och man kan fortsätta att utveckla på samma vis som man gör på nativ plattform. Den stora fördelen trots att det inte är byggt på principen *"write once, run everywhere"* är man kan skapa plattformsberoende kod som exempelvis logik och databashantering som kan direkt användas i iOS och Windows Phone 7 utan kompromisser.

6.3 Kritisk Diskussion

Examensarbetet har stundtals gånger drabbats av med- och motgångar under arbetsprocessen. Om vi skulle få samma uppdrag på nytt skulle vi hanterat det på ett annorlunda sätt. Trots att vi började med en research innan vi påbörjade arbetet skulle vi idag ha studerat mera ingående om vad användarna av olika cross plattformar skriver om olika funktioner i diverse communities. När vi utvecklade applikationen så fanns det oftast ingen information i fall vissa funktioner hade stöd för en viss plattform. Denna information finns oftast i communities där användare rapporterar om detta.

Vi kände också att vi kanske har varit för ambitiösa när vi valde att evaluera tre plattformar istället för två. Då skulle vi haft mera tid för att ta oss an diverse problem som vi stötte på under vägen och försöka lösa dem med att ta fram tidskrävande lösningar, vilket inte var möjligt med den befintliga tidsplanen. En annan sak som vi också kunde ha gjort annorlunda gäller vår approach att göra en evaluering av en utvecklings verktyg i taget. I stället borde vi initialt ha arbetat med samtliga cross plattformarna på en gång för att avgöra om någon av dessa är alldeles för krävande eller har en hög inlärningskurva. Vi hamnade i detta problem vid evaluering av cross plattformen MoSync, detta gjorde att det tog uppemot en vecka extra av projektiden att hitta en alternativ plattform.

Vi har haft relativt lätt att sätta oss in nya programmeringsspråk och miljöer på grund av erfarenheter från tidigare kurser under vår utbildning och vi antog att samma princip gällde för att lära oss utveckla applikationer med cross plattformar. Det visade sig att så var inte fallet med MoSync, där applikationen byggs upp av en speciell

struktur och man använder C++, att hinna med att evaluera detta verktyg på två veckor var inte möjligt.

Slutligen för att öka kvalitén ytterligare på utvärderingen skulle vi förutom emulatorerna och en äldre Android telefon med Android 2.2, också haft tillgång till en nyare Android telefon med Android 4.0 samt fått tillgång till en utvecklarlicens för att kunna installera vår applikation på en iPhone istället för enbart använda en simulator. Vi märkte att vår Android telefon hade dålig prestanda och fick applikationerna att krascha ofta, med de ovan nämnda åtgärderna hade scenariot möjligtvis varit annorlunda.

6.4 Fortsatt forskning

Då vi hann med mycket av det vi hade planerat att åstadkomma finns det nu med facit i hand, en del saker vi skulle vilja undersöka ytterligare. En vidare utveckling av den här evalueringen skulle vara att skapa en mycket större och mer komplex applikation med hjälp av enbart en cross plattform. Denna applikation skulle kunna innefatta en klient- och serverlösningar, eller en implementation av mer specifika tjänster så som exempelvis användning av Game Center.

Det vi skulle göra med Mono for Android, MonoTouch och generellt för andra cross plattformar som inte följer ordspråket “write once, run everywhere” är att kartlägga vilka områden i en applikation som kan bli plattformsberoende. Denna identifiering är svår att utföra och det behövs en strategi eller ett designmönster för att lyckas med uppgiften. Vi saknade denna insikt när vi började utveckla applikationen exakt på samma sätt som vid nativutveckling.

7 KÄLLFÖRTECKNING

7.1 Referenser

- [1] Vision Mobile Cross-Platform Developer Tools, 2012 sida 11.
- [2] COMPUTER WORLD.
[http://www.computerworld.com/s/article/9217437/Android vs. iPhone and i Pad by the numbers](http://www.computerworld.com/s/article/9217437/Android_vs._iPhone_and_i_Pad_by_the_numbers)
- [3] WebProNews
<http://www.webpronews.com/android-market-share-half-by-2012-2011-04>
- [4] ANDROID // UI PATTERNS
<http://www.androiduipatterns.com/2012/03/multi-platform-frameworks-destroy.html>
- [5] ZDNet
<http://www.zdnet.co.uk/blogs/jacks-blog-10017212/smartphones-run-html5-a-lot-slower-than-pcs-10026239/>
- [6] iOS Developer
<https://developer.apple.com/technologies/ios/>
- [7] iOS Developer Library
<https://developer.apple.com/library/ios/#documentation/Cocoa/Conceptual/ObjectiveC/Introduction/introObjectiveC.html>
- [8] iOS Developer Program
<https://developer.apple.com/programs/ios/distribute.html>
- [9] Users are not worried about iOS restrictions by Alex Knight
<http://zerodistracton.com/blog/2012/2/7/users-are-not-worried-about-ios-restrictions.html>
- [10] Apple Developer Program Enrollment
<https://developer.apple.com/programs/start/ios/>
- [11] iOS5 Developers
<https://developer.apple.com/technologies/ios5/>

- [12] iOS Data Management
<https://developer.apple.com/technologies/ios/data-management.html>
- [13] iOS Networking and Internet
<https://developer.apple.com/technologies/ios/networking.html>
- [14] iOS Developer Tools
<https://developer.apple.com/technologies/tools/>
- [15] iOS Graphics and Animation
<https://developer.apple.com/technologies/ios/graphics-and-animation.html>
- [16] MFMessageComposeViewController Class
https://developer.apple.com/library/ios/#documentation/MessageUI/Reference/MFMessageComposeViewController_class/Reference/Reference.html
- [17] Android Developers
<http://www.android.com/developers/>
- [18] Telekom idag
<http://www.telekomidag.se/nyheter/artikel.php?id=37856>
- [19] IDG
<http://computersweden.idg.se/2.2683/1.372435/android-i-topp-aven-for-mobilutvecklare>
- [20] Android Developers, philosophy and goals
<http://source.android.com/about/philosophy.html>
- [21] Android Developers, what is Android
<http://developer.android.com/guide/basics/what-is-android.html>
- [22] Android Developers, Installing the SDK
<http://developer.android.com/sdk/installing.html>
- [23] Android Developers, what is NDK
<http://developer.android.com/sdk/ndk/overview.html>
- [24] Google Play, for developers
<http://support.google.com/googleplay/android-developer/bin/answer.py?hl=en&answer=113468>

- [25] Amazon Appstore Developer Portal
<https://developer.amazon.com/help/faq.html>
- [26] Android Developers, Android 4.0 Plattform Hights
<http://developer.android.com/sdk/android-4.0-highlights.html>
- [27] Android Developers, The AndroidManifest.xml file
<http://developer.android.com/guide/topics/manifest/manifest-intro.html>
- [28] Android Developers, Data Storage
<http://developer.android.com/guide/topics/data/data-storage.html>
- [29] CHIPERSOFT
http://www.chipersoft.com/view/510/Why_I_STILL_prefer_Prototype_over_j_Query
- [30] Vision Mobile Cross-Platform Developer Tools, 2012 sidorna 6-27.
- [31] Github
<https://github.com/sintaxi/PhoneGap/blob/master/android/framework/src/com/PhoneGap/PhoneGap.java>
- [32] Github
<https://github.com/sintaxi/PhoneGap/blob/master/android/js/device.js>
- [33] Vision Mobile Cross-Platform Developer Tools, 2012 sidan 4.
- [34] PC Magasin
http://www.pcmag.com/encyclopedia_term/0,1237,t=runtime+engine&i=56079,00.asp
- [35] Vision Mobile Cross-Platform Developer Tools, 2012 sidan 24.
- [36] Vision Mobile Cross-Platform Developer Tools, 2012 sidan 23.
- [39] W3 schools, HTML5
http://www.w3schools.com/html5/html5_intro.asp
- [40] Apple Inc, Thoughts on Flash
<http://www.apple.com/hotnews/thoughts-on-flash/>
- [41] The HTML5 test
<http://html5test.com/about.html>

- [42] PhoneGap:Build, What is PhoneGap?
<https://build.PhoneGap.com/>
- [43] Vision Mobile Cross-Platform Developer Tools, 2012 sidan 53.
- [44] PhoneGap, Wiki
<http://wiki.PhoneGap.com/w/page/16494768/Debugging%20PhoneGap%20Apps>
- [45] Apache Cordova Documentation
http://docs.PhoneGap.com/en/1.7.0/guide_getting-started_android_index.md.html#Getting%20Started%20with%20Android
- [46] PhoneGap, Tools
<http://PhoneGap.com/tools>
- [47] JQuery
<http://jquerymobile.com/gbs/>
- [48] Appcelerator
<http://www.appcelerator.com/platform/titanium-sdk>
- [49] Vision Mobile Cross-Platform Developer Tools, 2012 sidan 60.
- [50] GitHub
<https://github.com/guilhermechapiewski/titanium-jasmine#readme>
- [51] Appcelerator
<https://marketplace.appcelerator.com/home>
- [52] Appcelerator
[http://www.appcelerator.com/plans-pricing/?__utma=1.1479156654.1332614439.1337691628.1338047405.5&__utmb=1.1.10.1338047405&__utmc=1&__utmz=1.1338047405.5.5.utmcsr=developer.appcelerator.com%7Cutmccn=\(referral\)%7Cutmcmd=referral%7Cutmct=/blog/2011/05/titanium-mobile-intro-series-sockets.html&__utmv=-&__utmk=167165502](http://www.appcelerator.com/plans-pricing/?__utma=1.1479156654.1332614439.1337691628.1338047405.5&__utmb=1.1.10.1338047405&__utmc=1&__utmz=1.1338047405.5.5.utmcsr=developer.appcelerator.com%7Cutmccn=(referral)%7Cutmcmd=referral%7Cutmct=/blog/2011/05/titanium-mobile-intro-series-sockets.html&__utmv=-&__utmk=167165502)
- [53] Vision Mobile Cross-Platform Developer Tools, 2012 sida 89
- [54] Xamarin, store
<https://store.xamarin.com/>
- [55] Xamarin, Trial

- <http://xamarin.com/trial>
- [56] Xamarin. Documentation
http://docs.xamarin.com/ios/tutorials/Introduction_to_iCloud
- [57] Xamarin, Documenation
http://docs.xamarin.com/ios/tutorials/Introduction_to_the_Twitter_API
- [58] Marmalade, About Us
<http://www.madewithmarmalade.com/aboutus>
- [59] Vision Mobile Cross-Platform Developer Tools, 2012 sida 69
- [60] Marmalade, Documentation
<http://www.madewithmarmalade.com/devnet/docs#/main/introduction.html>
- [61] Marmalade, Documentation
<http://www.madewithmarmalade.com/devnet/docs#/main/native/creatinganapplication/buildingforarm.html>
- [62] Marmalade, Documentation
<http://www.madewithmarmalade.com/devnet/docs#/api/s3eapidocumentation.html#ios>
- [63] Marmalade, Buy
<http://www.madewithmarmalade.com/buy>
- [64] Marmalade, Support
<http://www.madewithmarmalade.com/devnet/support>
- [65] MoSync
<http://www.mosync.com/what-is-mosync>
- [66] MoSync, Documentation
<http://www.mosync.com/files/imports/doxygen/latest/html5/index.html>
- [67] MoSync, Reload
<http://www.mosync.com/reload>
- [68] MoSync, Downloads
<http://www.mosync.com/download>
- [69] MoSync, Documentation

- <http://www.mosync.com/documentation/manualpages/getting-started-mosync-ide>
- [70] MoSync, Documentation
<http://www.mosync.com/documentation/manualpage/emulating-device>
- [71] MoSync, Documenation
<http://www.mosync.com/content/html5-javascript-wormhole>
- [72] MoSync, Forums
<http://www.mosync.com/content/connection-ios-game-centre-0>
- [73] MoSync
http://www.mosync.com/the_company
- [74] MoSync, Documentation
<http://www.mosync.com/documentation>
- [75] Vision Mobile Cross-Platform Developer Tools, 2012 sida 59
- [76] Corona, Developer
<http://developer.anscamobile.com/content/corona-20-tools-guide>
- [77] Corona, Project Manager v.3
<http://coronaprojectmanager.com/>
- [78] InformationWeek
<http://www.informationweek.com/news/personal-tech/smart-phones/222000116>
- [79] Corona, Developer
<http://developer.anscamobile.com/resources/apis>
- [80] Corona
<http://www.anscamobile.com/pricing/?ref=nav>
- [81] Corona, Developer
<http://developer.anscamobile.com/community/>
- [82] Vision Mobile Cross-Platform Developer Tools, 2012 sida 38

7.2 Bildkällor

Figur nr:

3 VisionMobile Cross-Platform Developer Tools, 2012

4 VisionMobile Cross-Platform Developer Tools, 2012

7.3 Tabellkällor

Tabell nr:

2 http://en.wikipedia.org/wiki/IOS_version_history

4 [http://en.wikipedia.org/wiki/Android_\(operating_system\)#Version_history](http://en.wikipedia.org/wiki/Android_(operating_system)#Version_history)

5 VisionMobile Cross-Platform Developer Tools, 2012

Appendix A - Ordlista

I det här appendixet ges en översikt över ofta använda termer och förkortningar.

ADT: Android Development Tools

ANSI: American National Standards Institute

API: Application Programming Interface

ARM: Application Response Measurement

BSD: Berkeley Software Distribution

commonJS: Standardiserad struktur av API:er för JavaScript

CPU: Central Processing Unit

CSS: Cascading Style Sheets

DDMS: Debug Monitor Server (DDMS)

HTML: HyperText Markup Language

IDE: Integrated Development Environment

JSON: JavaScript Object Notation

LLVM: fd. Low Level Virtual Machine

MVC: Model-View-Controller

NDK: Native Development Kit

OS X: Operativsystem X (Bokstaven "X" i OS X anspelar såväl på siffran tio som på Unix)

SDK: Soft Development Kit Dalvik

Appendix B

app.js

```
1 /*
2  * DigitalDiary
3  * Author: Joseph Hejderup & Eghogho Ileso
4  * File: app.js
5  * Version: 1.0
6  * Description: This is the bootstrap for the application
7  * This is mainly based from the Titanium Forging Tutorials
8  */
9
10
11
12 /*
13  * Self-calling function that will add the required commonJS modules
14  * to create the tabs used in this application. The tabs are added as
15  * an argument to the AppTabGroup module. The function has its own private
16  * scope to prevent polluting the global scope
17  */
18 (function() {
19
20     //Decalare the required modules to initate tabs
21     var AppTabGroup = require('ui/AppTabGroup'),
22         NewEntryWindow = require('ui/NewEntryWindow'),
23         ListWindow = require('ui/ListWindow'),
24         DiariesWindow = require('ui/DiariesWindow');
25
26
27     //Window used with the entries page,
28     // this is required in order to make the window listens for events
29     var list_window = new ListWindow();
30
31     //Initiates a tabgroup with specified properties
32     var app = new AppTabGroup(
33         {
34             title: 'Diaries',
35             icon: 'images/tab_diaries.png',
36             window: new DiariesWindow({title: 'Diaries', backgroundColor: 'white'})
37         }, {
38             title: 'New Entry',
39             icon: 'images/tab_add.png',
40             window: new NewEntryWindow()
41         }, {
42             title: 'Entries',
43             icon: 'images/tab_entries.png',
44             window: list_window
45         }
46     );
47
48     //Open the tab
49     app.open();
50 })();
51
52
```

AppTabGroup.js

```
1 /*
2  * DigitalDiary
3  * Author: Joseph Hejderup & Eghogho Ileso
4  * File: AppTabGroup.js
5  * Version: 1.0
6  * Description: This written with the commonJS approach
7  * This module is used to initiate the tabgroup and add
8  * the specified tabs from the arguments
9  */
10
11 function AppTabGroup() {
12
13     //Create the tabgroup
14     var instance = Ti.UI.createTabGroup();
15
16
17
18     //Get the length of the arguments added
19     //Notice: you don't need to put 'arguments' as a parameter to this function
20     //this is automatically done
21     var array_length = arguments.length;
22
23
24     for (var i = 0; i < array_length; i++) {
25
26         //Creates a tab window with the given properties
27         var tab = Ti.UI.createTab(arguments[i]);
28
29         //We want the second tab to display first
30         if (i === 1) {
31
32             instance.currentTab = tab;
33         }
34         //Add the created tab to the tabgroup
35         instance.addTab(tab);
36     }
37
38     //listen for the event 'focus', each time a tab is highlighted
39     //the application will set the tab as the current one.
40     instance.addEventListener('focus', function(e) {
41         instance.currentTab = e.tab;
42     });
43
44     return instance;
45 };
46 module.exports = AppTabGroup;
```

DiariesForm.js

```
1 /*
2  * DigitalDiary
3  * Author: Joseph Hejderup & Eghogho Ileso
4  * File: DiariesForm.js
5  * Version: 1.0
6  * Description: This module is a simple form
7  * used with the DiariesWindow form to add a
8  * new diary
9  */
10 //TODO: Before submitting: check the value in textfield so it dosen't contain any
    whitespace and check if the diary already exist in the database
11 exports.createForm = function(){
12
13 ////////////////////////////////////////////////// FORM/WINDOW
14
15
16     var form = Ti.UI.createView({
17         backgroundColor:'white',
18     });
19
20 ////////////////////////////////////////////////// UI COMPONENTS
21     var text_field = Ti.UI.createTextField({
22         top: 110,
23         height: 40,
24         width: 250,
25         color: '#222',
26         borderStyle:Titanium.UI.INPUT_BORDERSTYLE_ROUNDED
27     });
28     text_field.keyboardType = Ti.UI.KEYBOARD_DEFAULT;
29
30     var add_button = Ti.UI.createButton({
31         top:170,
32         left:80,
33         title: 'Add',
34         height: 40,
35         width: 70
36     });
37
38     var cancel_button = Ti.UI.createButton({
39         title: 'cancel',
40         top:170,
41         left:160,
42         height: 40,
43         width: 70,
44
45     });
46     var label = Titanium.UI.createLabel({
47         text:'Add a new diary',
48         height: 40,
49         width: 248,
50         top: 60,
51         shadowColor:'#aaa',
52         shadowOffset:{x:5,y:5},
53         color:'#222',
54         font:{fontSize:22},
55         textAlign:'left'
56     });
57
58 ////////////////////////////////////////////////// EVENTLISTENERS
59
60
61     add_button.addEventListener('click', function(e) {
62
63
```

DiariesForm.js

```
64 //NOTICE: hastext method is not working
65 if(text_field.value != ''){
66     //Fire an event, listener for "submit" will handle the event
67     form.fireEvent('submit', {title: text_field.value});
68 }else{
69     alert('Please enter a title before submitting');
70 }
71 });
72
73 cancel_button.addEventListener('click', function(e) {
74
75     //Send an close event
76     form.fireEvent('cancel', {event:'cancel'});
77
78 });
79
80 ////////////////////////////////////////////////// ADD COMPONENTS TO FORM
81
82 form.add(label);
83 form.add(text_field);
84 form.add(add_button);
85 form.add(cancel_button);
86
87
88 return form;
89
90
91 };
92
```

DiariesWindow.js

```
1 /*
2  * DigitalDiary
3  * Author: Joseph Hejderup & Eghogho Ileso
4  * File: DiariesWindow.js
5  * Version: 1.3
6  * Description: This module will handle the user-specified diaries and populate all
the user-entered diaries in the tableview
7  * Each diary will be assigned a specific table in the database
8  */
9 //TODO: Remove all database code to a seperate file, although some attempts have
been done without success.
10 //TODO: Add an edit-feature to edit the title of the database, modifications also
has to be done in DiariesForm.js
11 function DiariesWindow(args) {
12
13 /*
14  *      GLOBAL VARS AND INITIALIZATION
15 */
16     //Properties is almost the same as with preferences with android
17     Ti.App.Properties.setString('table_name', null);
18     //This variable will keep track of the last row clicked
19     var previousRowClicked;
20
21     //Open/create database
22     //The database will be installed once, no matter how many times you install
23     var db = Ti.Database.open('diaries.db');
24     //Create a table to track all the tablenamees if it don't exist
25     db.execute('CREATE TABLE IF NOT EXISTS tableNames(name TEXT PRIMARY KEY)');
26 /*
27  *      UI COMPONENTS - Window and TableView
28 */
29     //Create a windows according with the properties given in the parameter
30     var instance = Ti.UI.createWindow(args);
31
32     //Create a tableview and call the internal function populateDataForTableView
33     var diary_tv = Titanium.UI.createTableView({
34         data:populateDataForTableView(),
35         editable:true
36     });
37     //Add the tableview to the window
38     instance.add(diary_tv);
39
40 /*
41  *      Eventlisteners
42 */
43     //Normal click events will select the current diary only
44     diary_tv.addEventListener('click', function(e) {
45
46         //Set the recent row to unchecked
47         if(previousRowClicked)
48             previousRowClicked.hasCheck = false;
49         //Checkmark the current row
50         e.row.hasCheck = true;
51         //Store the current row for use next time
52         previousRowClicked = e.row;
53         //Save the diary title to accomdate for use in the other windows
54         Ti.App.Properties.setString('table_name', e.rowData.title);
55     });
56
57     //Long click event will delete the selected diary. This will only work with
android
58     diary_tv.addEventListener('longclick', function(row_tv){
59
60
```

DiariesWindow.js

```

61     //Create an alertdialog for deletion
62     var alertDialog = Titanium.UI.createAlertDialog({
63         title: 'Are you sure?',
64         message: 'Do you want to delete ?',
65         buttonNames: ['Yes', 'No'],
66         cancel: 1
67     });
68
69     //Add an eventlistener for dialogbox
70     alertDialog.addEventListener('click', function(e) {
71
72         //user clicked yes
73         if (e.index == 0) {
74             //Delete the selected row in the database
75             deleteDiary({tablename: row_tv.rowData.title});
76             //Update tableview
77             diary_tv.setData(populateDataForTableView());
78         }
79     });
80
81     //display the alert dialog for the user
82     alertDialog.show();
83 });
84
85
86 if(Titanium.Platform.osname == 'android'){
87
88     androidMenuUI({window: instance, tv: diary_tv});
89
90 }else{
91
92     iphoneMenuUI({window: instance, tv: diary_tv});
93
94 }
95
96 return instance;
97
98
99
100 };
101 /*
102  * This internal method will launch a modal window so the user
103  * can add a new library
104  */
105 var addDiary = function(args){
106
107     //aquire the required module
108     var forms = require('ui/DiariesForm');
109     //Create the form
110     var forms = forms.createForm();
111     //Create a new window
112     var win_form = Ti.UI.createWindow({backgroundColor: '#fff'});
113     //var forms is a simple view, add it to the window
114     win_form.add(forms);
115
116     forms.addEventListener('submit', function(e) {
117
118         //id is date
119         //Create a new table in the database
120         var db = Ti.Database.open('diaries.db');
121         db.execute('CREATE TABLE IF NOT EXISTS ' + e.title
122             + '(id INTEGER PRIMARY KEY, title TEXT, story TEXT, image TEXT,
123             sound TEXT, longitude TEXT, latitude TEXT)');
123         db.close();

```

DiariesWindow.js

```

124         //Save table name in a separate table
125         saveTableName({name: e.title});
126         //update the tableview with new values
127         args.tableview.setData(populateDataForTableView());
128
129         //Close the form
130         win_form.close();
131     });
132
133     forms.addEventListener('cancel', function(e) {
134         //Close the form
135         win_form.close();
136     });
137     //Open a modal window
138     win_form.open({modal:true, fullscreen:true, navBarHidden:true,
    animation:true});
139 };
140 /*
141  * Creates an android menu which will allow the user to add a new diary
142  */
143 var androidMenuUI = function(args){
144
145     args.window.activity.onCreateOptionsMenu = function(e) {
146         var menu = e.menu;
147         var menuItem = menu.add({title: 'Add'});
148         menuItem.addEventListener("click", function(e) {
149             addDiary({tableview: args.tv});
150         });
151     }
152 };
153 /*
154  * This will add native iphone features to add and delete entries in the tableview
155  */
156 var iphoneMenuUI = function(args){
157
158     // iphone button for add
159     var add = Titanium.UI.createButton({
160         systemButton:Titanium.UI.iPhone.SystemButton.ADD
161     });
162     // iphone button for edit
163     var edit = Titanium.UI.createButton({
164         systemButton:Titanium.UI.iPhone.SystemButton.EDIT
165     });
166
167     args.window.setRightNavButton(add);
168     args.window.setLeftNavButton(edit);
169
170     //Add a new diary
171     add.addEventListener('click', function(e) {
172         addDiary({tableview: args.tv});
173     });
174     //Make the tableview editable so the entries can be deleted
175     edit.addEventListener('click', function(e) {
176
177         if(!args.tv.editing){
178             args.tv.editing = true;
179         }else{
180             args.tv.editing = false;
181         }
182     });
183
184     args.tv.addEventListener('delete',function(e) {
185
186         deleteDiary({tablename: row_tv.selectRow.title});

```

DiariesWindow.js

```

187         //The tableview need to be updated after a deletion
188         args.tv.setData(populateDataForTableView());
189
190     });
191 };
192 /*
193  * A seperate table in the databse will maintain the all the diaries
194  */
195 var saveTableName = function(args){
196
197     var db = Ti.Database.open('diaries.db');
198     db.execute('INSERT INTO tableNames (name) VALUES(?)',args.name);
199     db.close();
200 };
201 /*
202  * This method wil fetch all the diaries in the database and return all diaries in
    an array
203  */
204 var populateDataForTableView = function(){
205
206     var db = Ti.Database.open('diaries.db');
207     db.execute('CREATE TABLE IF NOT EXISTS tableNames(name TEXT PRIMARY KEY)');
208     var rows = db.execute('SELECT * FROM tableNames');
209     var data = [];
210
211
212     while(rows.isValidRow()){
213
214         data.push({color:'#000',font:{fontFamily:'Trebuchet
MS',fontSize:16,fontWeight:'bold'},leftImage:'../images/diaries.png',title:rows.fie
ldByName('name'),hasCheck:false});
215         rows.next();
216     }
217
218     rows.close();
219     db.close();
220
221     return data;
222 };
223 /*
224  * Delete the diary in both tables
225  */
226 var deleteDiary = function(args){
227
228     var db = Ti.Database.open('diaries.db');
229     //Delete the entire table
230     db.execute('DROP TABLE IF EXISTS '+args.tablename+');
231     //Delete from the table list as well
232     db.execute('DELETE FROM tableNames WHERE name=?',args.tablename);
233     db.close();
234 };
235
236 module.exports = DiariesWindow;

```

ListWindow.js

```
1 /*
2  * DigitalDiary
3  * Author: Joseph Hejderup & Eghogho Ileso
4  * File: ListWindow.js
5  * Version: 1.0
6  * Description: This window willl display all the user-entries from the selected
   diary
7  */
8 //TODO: Database logic needs to be removed and inseted in a seperate file
9 //TODO: Many methods is similar to those in DiariesWindow.js. The code is not DRY
10 //      and therefore needs to undergo refactoring before more changes or done.
11 function ListWindow(title) {
12
13 /*
14  * UI
15 */
16     var entry;
17     //Main window
18     var self = Ti.UI.createWindow({
19         title:title,
20         backgroundColor:'white',
21     });
22
23     // create table view
24     var tableview = Titanium.UI.createTableView({
25         editable:true,
26         data:populateDataForTableView()
27     });
28     self.add(tableview);
29
30
31 /*
32  * Native UI
33 */
34     if(Titanium.Platform.osname == 'iphone'){
35         iphoneMenuUI({window: self, tv: tableview});
36     }
37
38 /*
39  * EVENTLISTENERS
40 */
41 //When the tab is focused an update is made
42 self.addEventListener('focus',function(){
43     tableview.setData(populateDataForTableView());
44 });
45 //longclick events will delete the selected entry in the database. This event
   is only supported for android
46 tableview.addEventListener('longclick', function(row_tv){
47     var title = row_tv.rowData.title;
48     deleteEntry({title: title});
49 });
50 //This event will open a new modal window and will display the information
   specified in the entry
51 tableview.addEventListener('click',function(e) {
52     var db = Ti.Database.open('diaries.db');
53     var rows = db.execute('SELECT * FROM
   '+Ti.App.Properties.getString('table_name')+ ' WHERE title=?',e.rowData.title);
54     //Check if it is a valid row first
55     if(rows.isValidRow()){
```

ListWindow.js

```

61         //Open a new modal window with the user-entered data
62         var ShowEntry = require('ui/ShowEntry')
63         entry = new
        ShowEntry({date:rows.fieldByName('id'),title:rows.fieldByName('title'),text:rows.fi
        eldByName('story'),image:rows.fieldByName('image'),
64             longitude: rows.fieldByName('longitude'),latitude:
        rows.fieldByName('latitude')});
65     }
66     rows.close();
67     db.close();
68
69     entry.open({modal:true, fullscreen:true, animation:true});
70
71
72
73     });
74     //Since the tableview won't update automatically, listeners have been added
75     Ti.App.addEventListener('app:updateTables', function() {
76         tableview.setData(populateDataForTableView());
77     });
78
79     Ti.App.addEventListener('app:closeEntry', function() {
80         entry.close();
81     });
82
83     return self;
84 }
85 /*
86  * This method follows the same principles as defined in DiariesWindow.js
87  */
88 var populateDataForTableView = function(){
89
90
91     var data = [];
92
93     if(Ti.App.Properties.getString('table_name') != null){
94         var db = Ti.Database.open('diaries.db');
95         var rows = db.execute('SELECT * FROM
        '+Ti.App.Properties.getString('table_name'));
96
97         //Ti.API.debug('row count: '+rows.rowCount);
98
99         while(rows.isValidRow()){
100             data.push({color:'#000',font:{fontFamily:'Trebuchet
        MS',fontSize:16,fontWeight:'bold'},leftImage:'../images/entry.png',title:rows.field
        ByName('title')});
101             rows.next();
102         }
103
104         rows.close();
105         db.close();
106
107     }
108     return data;
109
110 };
111 /*
112  * This method only updates the listview and fetches the current records in the
        database
113  */
114 var updateLV = function(args){
115
116     args.tableview.setData(populateDataForTableView());
117

```

ListWindow.js

```

118
119 };
120 /*
121  * Delete the selected entry
122  */
123 var deleteEntry = function(args){
124
125     var alertDialog = Titanium.UI.createAlertDialog({
126         title: 'Are you sure?',
127         message: 'Do you want to delete this ?',
128         buttonNames: ['Yes', 'No'],
129         cancel: 1
130     });
131     //Add an eventlistener for dialogbox
132     alertDialog.addEventListener('click',function(e) {
133
134         if (e.index == 0) {
135
136             var db = Ti.Database.open('diaries.db');
137             db.execute('DELETE FROM
'+Ti.App.Properties.getString('table_name')+' WHERE title=?',args.title);
138             db.close();
139             //Update the tableview, instead of passing the tableview in the
arguments, this is a smoother way
140             Ti.App.fireEvent('app:updateTables');
141         }
142     });
143     //display the alert dialog for the user
144     alertDialog.show();
145 };
146 /*
147  * Iphone UI features with both add and delete functions for the tableview
148  * Follows the same pattern as in DiariesWindow
149  */
150 var iphoneMenuUI = function(args){
151
152     // iphone button for edit
153     var edit = Titanium.UI.createButton({
154         systemButton: Titanium.UI.iPhone.SystemButton.EDIT
155     });
156
157     args.window.setLeftNavButton(edit);
158
159     edit.addEventListener('click', function(e) {
160
161         if(!args.tv.editing){
162             args.tv.editing = true;
163         }else{
164             args.tv.editing
165         }
166     });
167
168     args.tv.addEventListener('delete',function(e) {
169         deleteEntry({tablename: e.selectRow.title});
170         args.tv.setData(populateDataForTableView());
171     });
172 };
173 module.exports = ListWindow;
174

```

NewEntryWindow.js

```

1 /*
2  * DigitalDiary
3  * Author: Joseph Hejderup & Eghogho Ileso
4  * File: NewEntryWindow.js
5  * Version: 1.1
6  * Description: This window will allow the user to take a photo, record voice, add
  gps position
7  */
8 //TODO: Database logic need to be removed and placed in a seperate file
9 //TODO: The "Reset field function" needs to be implemented for iphone
10 //TODO: The "Reset field functino" also has to reset all the paths and end
  positioning: UPDATE PATHS ARE DONE!
11 //TODO: Error handling if the user adds another entry for the same day
12 function NewEntryWindow(title) {
13     /*
14      * LOCAL VARIABLES
15      */
16     var addLocation = false;
17     /*
18      * UI COMPONENTS
19      */
20     //////////// WINDOWS
21
22     var entrywin = Ti.UI.createWindow({
23         title:title,
24         backgroundColor:'white',
25
26     });
27     //A scrollview is added so the keyboard won't cover the textfields
28     var scrollView = Titanium.UI.createScrollView({
29         contentHeight:'auto',
30         contentWidth:'auto',
31         showVerticalScrollIndicator:false,
32         showHorizontalScrollIndicator:false,
33     });
34
35     //////////// TEXT FIELDS
36
37     //This is a custom textfield that will display a straight line
38     var txt_title = Ti.UI.createTextField({
39         hintText:'Title',
40         font:{fontFamily:'Trebuchet MS',fontSize:16,fontWeight:'bold'},
41         editable:true,
42         enabled:true,
43         borderWidth: 0, borderStyle: Titanium.UI.INPUT_BORDERSTYLE_NONE,
44         borderColor: 'white',
45         backgroundImage:'../images/bg_title.png',
46         backgroundDisabledImage:'../images/bg_title.png',
47         backgroundFocusedImage: '../images/bg_title.png',
48         top:0,left:0,height:48,width:320
49     });
50     scrollView.add(txt_title);
51     //This is also custom made
52     var txt_area = Titanium.UI.createTextArea({
53         hintText:'What happened today?...',
54         height:150, width:315, top: 170,
55         font:{fontFamily:'Trebuchet MS',fontSize:16,fontWeight:'bold'},
56         keyboardType:Titanium.UI.KEYBOARD_DEFAULT,
57         textAlign:'left',
58         backgroundImage:'../images/bg_txt.png',
59         backgroundDisabledImage:'../images/bg_txt.png',
60         backgroundFocusedImage: '../images/bg_txt.png',
61         borderWidth: 0, borderStyle: Titanium.UI.INPUT_BORDERSTYLE_NONE,
62         borderColor: 'white',

```

NewEntryWindow.js

```

63     });
64     scrollView.add(txt_area);
65
66     //////////// BUTTONS (CUSTOM ICONS)
67     var camera_btn = Ti.UI.createButton({
68         backgroundImage: '../images/camera_btn.png',
69         top:75,
70         left:20,
71         width: 80,
72         height:80
73     });
74     scrollView.add(camera_btn);
75     var position_btn = Ti.UI.createButton({
76         backgroundImage: '../images/position_btn.png',
77         top:75,
78         left:120,
79         width: 80,
80         height:80
81     });
82     scrollView.add(position_btn);
83     var speech_btn = Ti.UI.createButton({
84         backgroundImage: '../images/speech_btn.png',
85         top:75,
86         left:220,
87         width: 80,
88         height:80
89     });
90     scrollView.add(speech_btn);
91
92
93     //The scrollview with all the UI Components is added to the main window
94     entrywin.add(scrollView);
95
96     /*
97     * EVENT HANDLERS
98     */
99     //Open the camera for the user to snap a photo
100     camera_btn.addEventListener('click',function(){
101
102         //Set the current image path to 'none'
103         Ti.App.Properties.setString('camera_file','none');
104         require('../io/utilities').camera();
105
106     });
107     //Allow the user to record voice
108     speech_btn.addEventListener('click',function(){
109
110         alert('Not available for android devices');
111     });
112     //Update the position
113     position_btn.addEventListener('click',function(){
114
115         require('../io/utilities').updatePosition();
116     });
117     /*
118     * DEVICE SPECIFIC UI FEATURES
119     */
120     if(Titanium.Platform.osname == 'android'){
121
122         entrywin.activity.onCreateOptionsMenu = function(e){
123             var menu = e.menu;
124             //This will save the diary
125             var menuItem = menu.add({title: 'Save'});
126             //This will reset all fields

```

NewEntryWindow.js

```

127         var menuItemReset = menu.add({title: 'Reset'});
128
129         menuItem.addEventListener("click", function(e) {
130             //Get the position
131             require('../io/utilities').savePosition();
132             //Save the information in the database
133             saveOnDB({title_view: txt_title,text_view: txt_area,title:
txt_title.value, text: txt_area.value, image: file,sound:'none',
134                 longitude:Ti.App.Properties.getString('longitude'),
latitude: Ti.App.Properties.getString('latitude')});
135         });
136
137         menuItemReset.addEventListener("click", function(e) {
138
139             var alertDialog = Titanium.UI.createAlertDialog({
140                 title: 'Are you sure?',
141                 message: 'Do you want to reset ALL fields ?',
142                 buttonNames: ['Yes','No'],
143                 cancel: 1
144             });
145
146             alertDialog.addEventListener('click',function(e){
147                 if (e.index == 0)
148                     resetAllFields({title_v: txt_title,text_v: txt_area});
149
150             });
151             //display the alert dialog for the user
152             alertDialog.show();
153         });
154     }
155
156     }else{
157
158         // iphone button for add
159         var add = Titanium.UI.createButton({
160             systemButton:Titanium.UI.iPhone.SystemButton.ADD
161         });
162         entrywin.setRightNavButton(add);
163         add.addEventListener('click', function(e) {
164             saveOnDB({title: txt_title.value, text: txt_area.value, image:
file,sound:'none',
165                 longitude:Ti.App.Properties.getString('longitude'), latitude:
Ti.App.Properties.getString('latitude')});
166         });
167     }
168
169     return entrywin;
170
171 }
172 /*
173  * This internal method will save the user entered data to the database for the
selected diary
174  */
175 var saveOnDB = function(args){
176
177     //Check if the user has selected a diary
178     if(Ti.App.Properties.getString('table_name') != null){
179         //Format the date to have a specific format
180         var date = new Date();
181         var date_format = ""+date.getDate()+(date.getMonth()+1)+date.getFullYear();
182
183         //Save the text from the story textfield as a textfile
184         var fileName =
"story_"+Ti.App.Properties.getString('table_name')+'_'+date_format+'.txt';

```

NewEntryWindow.js

```

185     var file = Ti.Filesystem.getFile(Ti.Filesystem.applicationDataDirectory,
file_name);
186     file.write(args.text);
187
188     //Save data to the database
189     var db = Ti.Database.open('diaries.db');
190     db.execute('INSERT INTO '+ Ti.App.Properties.getString('table_name')+
(id,title,story,image,sound,longitude,latitude) VALUES(?,?,?,?,?,?,?)',
191     date_format,args.title,file_name,Ti.App.Properties.getString('camera_file'),a
rgs.sound,args.longitude,args.latitude);
192     db.close();
193
194     //Reset all fields
195     resetAllFields({title_v:args.title_view, text_v: args.text_view});
196
197 }else{
198
199     alert('You need to choose a diary first!');
200 }
201 };
202 /*
203  * Resets all fields and sets camera path to "none"
204  */
205 var resetAllFields = function(args){
206
207     args.title_v.value = "";
208     args.text_v.value = "";
209     Ti.App.Properties.setString('camera_file','none');
210
211 }
212
213 module.exports = NewEntryWindow;

```

ShowEntry.js

```
1 /*
2  * DigitalDiary
3  * Author: Joseph Hejderup & Eghogho Ileso
4  * File: ShowEntry.js
5  * Version: 1.1
6  * Description: Displays the user-specific entered data such as gps position,
   image etc
7  */
8 //TODO: Add a button so iphone devices also is able to close
9 //TODO: Handle errors if the image file cannot be retrieved
10 //TODO: Add function to play recorded voice
11 //TODO: Add an edit button so the exsisting information can be transfered
12 //      to the new entry window
13
14 function ShowEntry(args) {
15
16     /*
17      * UI COMPONENTS
18      */
19
20     //////////// VIEW AND WINDOWS
21
22     var entrywin = Ti.UI.createWindow({
23         title:args.title,
24         exitOnClose:true,
25         backgroundColor:'white',
26
27     });
28     var scrollView = Titanium.UI.createScrollView({
29         contentHeight:'auto',
30         contentWidth:'auto',
31         showVerticalScrollIndicator:false,
32         showHorizontalScrollIndicator:false,
33     });
34     //////////// TEXT FIELDS
35
36     //Add both date and title
37     var title = args.date + ' | ' + args.title;
38     var txt_title = Ti.UI.createTextField({
39         value: title,
40         font:{fontFamily:'Trebuchet MS',fontSize:18,fontWeight:'bold'},
41         editable:false,
42         enabled:true,
43         borderWidth: 0, borderStyle: Titanium.UI.INPUT_BORDERSTYLE_NONE,
44         borderColor: 'white',
45         backgroundImage:'../images/bg_title.png',
46         backgroundDisabledImage:'../images/bg_title.png',
47         backgroundFocusedImage: '../images/bg_title.png',
48         top:0,left:0,height:48,width:320
49     });
50     scrollView.add(txt_title);
51     var txt_position = Ti.UI.createTextField({
52         value: 'Latitude: ' + args.latitude + ', Longitude: ' + args.longitude,
53         font:{fontFamily:'Trebuchet MS',fontSize:16,fontWeight:'bold'},
54         editable:false,
55         enabled:true,
56         borderWidth: 0, borderStyle: Titanium.UI.INPUT_BORDERSTYLE_NONE,
57         borderColor: 'white',
58         backgroundImage:'../images/bg_title.png',
59         backgroundDisabledImage:'../images/bg_title.png',
60         backgroundFocusedImage: '../images/bg_title.png',
61         top:50,left:0,height:48,width:320
62     });
63     scrollView.add(txt_position);
```

ShowEntry.js

```

64     var txt_area = Titanium.UI.createTextArea({
65         value: loadTextFromFile(),
66         font:{fontFamily:'Trebuchet MS',fontSize:16},
67         editable:false,
68         height:150, width:315, top: 422,
69         keyboardType:Titanium.UI.KEYBOARD_DEFAULT,
70         //font:{fontSize:12,fontFamily:'Bradley Hand'},
71         textAlign:'left',
72         backgroundImage:'../images/bg_txt.png',
73         backgroundDisabledImage:'../images/bg_txt.png',
74         backgroundFocusedImage: '../images/bg_txt.png',
75         borderWidth: 0, borderStyle: Titanium.UI.INPUT_BORDERSTYLE_NONE,
76         borderColor: 'white',
77     });
78     scrollView.add(txt_area);
79
80     //////////// IMAGE VIEW
81     var image_view = Ti.UI.createImageView({
82         image: loadImageFromFile(),
83         top: 100,
84         width: 320,
85         height: 320
86     });
87     scrollView.add(image_view);
88
89     //Lastly add the scrollable view to the main window
90     entrywin.add(scrollView);
91     /*
92      * EVENT LISTENETS
93      */
94     entrywin.addEventListener('android:back',function(){
95
96         //This event will close this modol window when the physical back button is
entered
97         Ti.App.fireEvent('app:closeEntry');
98
99     });
100
101     return entrywin;
102     /*
103      * Retrieves the text file using the path provided
104      */
105     function loadTextFromFile() {
106         var file = Ti.Filesystem.getFile(Ti.Filesystem.applicationDataDirectory,
args.text);
107         return file.exists() ? file.read().text : "";
108     };
109     /*
110      * Retrieves the image file using the path provided
111      */
112     function loadImageFromFile() {
113
114         return
Titanium.Filesystem.getFile(Titanium.Filesystem.applicationDataDirectory,args.image)
;
115     };
116 }
117
118 module.exports = ShowEntry;

```

utilities.js

```
1 /*
2  * DigitalDiary
3  * Author: Joseph Hejderup & Eghogho Ileso
4  * File: utilities.js
5  * Version: 1.0
6  * Description: Utilities to access camera, gps and voice recording
7  * is accesible here. Most of the methods are based on the Kitchen Sink example
8  */
9 //TODO: Make the code DRY getPosition and savePosition are the same
10 //TODO: Add voice recording capabilities
11 //TODO: It would be much better if values could be returned rather than
12 //      being saved using Ti properties, however returning values is not
13 //      working
14
15 /*
16  * This function will allow the user to use the integrated camera
17  */
18 exports.camera = function() {
19
20     Titanium.Media.showCamera({
21
22         success: function(event) {
23             var cropRect = event.cropRect;
24             var image = event.media;
25
26             var date = new Date();
27             //Notice month: 0-11, jan:0 and dec:11
28
29             //Save the image file
30             var fileName =
31 'camera_'+Ti.App.Properties.getString('table_name')+'_'+date.getDate()+(date.getMonth()+1)+date.getFullYear()+'.png';
32             var f =
33 Titanium.Filesystem.getFile(Titanium.Filesystem.applicationDataDirectory, fileName);
34             f.write(image);
35             //Store the path for later use
36             Ti.App.Properties.setString('camera_file', fileName);
37
38             }, cancel: function() {
39                 return;
40             }, error: function(error) {
41                 // create alert
42                 var a = Titanium.UI.createAlertDialog({title: 'Camera'});
43
44                 // set message
45                 if (error.code == Titanium.Media.NO_CAMERA) {
46                     a.setMessage('Device does not have video recording
47 capabilities');
48                 } else {
49                     a.setMessage('Unexpected error: ' + error.code);
50                 }
51                 // show alert
52                 a.show();
53                 return;
54             }, allowEditing: false, autohide: true
55             });
56 };
57
58 /*
59  * This function turns on GPS tracking
60  */
61 exports.updatePosition = function() {
62
63     Ti.Geolocation.preferredProvider = "gps";
64 }
```

utilities.js

```
61 //The purpose needs to be set, iOS only
62 Ti.Geolocation.purpose = "GPS demo";
63 Titanium.Geolocation.accuracy = Titanium.Geolocation.ACCURACY_BEST;
64 //This tells how often events should be triggered based on differences in meter
65 Titanium.Geolocation.distanceFilter = 10;
66
67 //Listens for event, event are triggered according to the set distanceFilter
68 Titanium.Geolocation.addEventListener('location', function(){
69
70     //Internal function is called
71     getPosition();
72
73 });
74
75
76 }
77 /*
78  * This method will save the position, the method updatePosition needs to be called
79  * first in order
80  * to start tracking the position
81  */
82 exports.savePosition = function(){
83
84     Titanium.Geolocation.getCurrentPosition(function(e)
85     {
86         if (!e.success || e.error)
87         {
88             Ti.API.info("Code translation: "+translateErrorCode(e.code));
89             alert('error ' + JSON.stringify(e.error));
90             return;
91         }
92
93         var longitude = e.coords.longitude;
94         var latitude = e.coords.latitude;
95
96         //Save the position for later use, the information can be accessed everywhere
97         Ti.App.Properties.setString('longitude',longitude+"");
98         Ti.App.Properties.setString('latitude',latitude+"");
99     });
100     //Remove the listener and stop tracking position changes
101     Titanium.Geolocation.removeEventListener('location', function(){
102     });
103
104
105 });
106 /*
107  * This method simply reads the current position
108  */
109 var getPosition = function(){
110
111     Titanium.Geolocation.getCurrentPosition(function(e)
112     {
113         if (!e.success || e.error)
114         {
115             Ti.API.info("Code translation: "+translateErrorCode(e.code));
116             alert('error ' + JSON.stringify(e.error));
117             return;
118         }
119
120         var longitude = e.coords.longitude;
121         var latitude = e.coords.latitude;
122
123     });
```

utilities.js

```
124
125 };
126 /*
127  * A set of error codes are translated into text to now what sort of error it is
128  */
129 function translateErrorCode(code) {
130     if (code == null) {
131         return null;
132     }
133     switch (code) {
134         case Ti.Geolocation.ERROR_LOCATION_UNKNOWN:
135             return "Location unknown";
136         case Ti.Geolocation.ERROR_DENIED:
137             return "Access denied";
138         case Ti.Geolocation.ERROR_NETWORK:
139             return "Network error";
140         case Ti.Geolocation.ERROR_HEADING_FAILURE:
141             return "Failure to detect heading";
142         case Ti.Geolocation.ERROR_REGION_MONITORING_DENIED:
143             return "Region monitoring access denied";
144         case Ti.Geolocation.ERROR_REGION_MONITORING_FAILURE:
145             return "Region monitoring access failure";
146         case Ti.Geolocation.ERROR_REGION_MONITORING_DELAYED:
147             return "Region monitoring setup delayed";
148     }
149 };
```

```
1  using System;

2  using Android.App;
3  using Android.Content;
4  using Android.Runtime;
5  using Android.Views;
6  using Android.Widget;
7  using Android.OS;

8  /*
9   * DigitalDiary
10  * Author: Joseph Hejderup & Eghogho Ileso
11  * File: TabHandler.cs
12  * Version: 1.0
13  * Description: This activity will handle
14  * the three tabs in the application
15  * TODO: change the icon and make a common method instead of repeating the same 7
16  * thing three times
17  namespace DigitalDiary
18  {
19      //mainlauncher = true, means its registred as launchable
20      [Activity (Label = "DigitalDiary", Theme=" 7
21          @android:style/Theme.Light.NoTitleBar")]
22      public class TabHandler : TabActivity
23      {
24          protected override void OnCreate (Bundle bundle)
25          {
26              SetContentView(Resource.Layout.Main);
27              TabHost.TabSpec spec;      // Resusable TabSpec for each tab
28              Intent intent;              // Reusable Intent for each tab

29              // Create an Intent to launch an Activity for the tab (to be reused)
30              intent = new Intent (this, typeof (Activities.DiariesActivity));
31              intent.AddFlags (ActivityFlags.NewTask);

32              //DIARIES TAB
33              // Initialize a TabSpec for each tab and add it to the TabHost
34              spec = TabHost.NewTabSpec ("Diaries");
35              spec.SetIndicator ("Diaries", Resources.GetDrawable (Resource.Drawable. 7
36                  ic_tab_artists));
37              spec.SetContent (intent);
38              TabHost.AddTab (spec);

39              // Do the same for the other tabs
40              intent = new Intent (this, typeof (Activities.NewEntryActivity));
41              intent.AddFlags (ActivityFlags.NewTask);

42              spec = TabHost.NewTabSpec ("New Entry");
43              spec.SetIndicator ("New Entry", Resources.GetDrawable (Resource.Drawable. 7
44                  ic_tab_artists));
45              spec.SetContent (intent);
46              TabHost.AddTab (spec);
```

```
45     intent = new Intent (this, typeof (Activities.EntriesActivity));
46     intent.AddFlags (ActivityFlags.NewTask);

47     spec = TabHost.NewTabSpec ("Entries");
48     spec.SetIndicator ("Entries", Resources.GetDrawable (Resource.Drawable. ic_tab_artists));
49     spec.SetContent (intent);
50     TabHost.AddTab (spec);

51     //Set the current tab
52     TabHost.CurrentTab = 1;
53 }

54 }
55 }
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.IO;

6  using Android.App;
7  using Android.Content;
8  using Android.OS;
9  using Android.Runtime;
10 using Android.Views;
11 using Android.Widget;
12 using Android.Locations;
13 using Android.Util;
14 using Android.Preferences;

15 /*
16  * DigitalDiary
17  * Author: Joseph Hejderup & Eghogho Ileso
18  * File: NewEntryActivity.cs
19  * Version: 1.0
20  * Description: The user is able to record audio, take a picture,
21  * getting the gps coordinates
22  *
23 namespace DigitalDiary.Activities
24 {
25     [Activity (Label = "NewEntryActivity")]
26     public class NewEntryActivity : Activity, ILocationListener
27     {
28         //Local variables
29         private Button btn_submit;
30         private Button btn_camera;
31         private Button btn_voice;
32         private Button btn_location;
33         private Button btn_reset;
34         public static string MY_PREFS = "MY_PREFS";
35         private EditText editTxt_Title;
36         private EditText editTxt_Story;
37         private string Filename = null;
38         private string camera_file = null;
39         private string sound_file = null;
40         private LocationManager _locationmanager;
41         private Location currentLoc = null;

42         protected override void OnCreate (Bundle bundle)
43         {
44             base.OnCreate (bundle);
45             SetContentView (Resource.Layout.NewEntry);
46             //Attach all UI components to the code
47             InitiateLayoutItems();

48             //TODO: use this code for edit mode
49             //TODO:Fix the filename for use when submitting the content
50             /*
51             string path = System.Environment.GetFolderPath
```

```

        (System.Environment.SpecialFolder.Personal);
52     Filename = "story_"+GetCurrentMilliseconds()+".txt"; --> read the filename
        i db instead
53     string filename = Path.Combine (path, Filename);
54     if (File.Exists (filename)) {
55     using (var f = new StreamReader (OpenFileInput (Filename))) {
56     string line;
57     do {
58     line = f.ReadLine ();
59     } while (!f.EndOfStream);

60     editTxt_Story.Text = line;
61     }
62     }
63     */
64     //Get the location manager
65     _locationmanager = (LocationManager)this.GetService(Context.
        LocationService);

66     //If the camera button is selected, open the native camera
67     btn_camera.Click += delegate(object sender, EventArgs e) {

68     Intent intent = new Intent(Android.Provider.MediaStore.
        ActionImageCapture);
69     //after the camera activity has ended, file management etc is dealt here
70     StartActivityForResult(intent,0);

71     };
72     //When the user click the "submit" button the user-entered data is saved
in the database
73     btn_submit.Click += delegate(object sender, EventArgs e) {

74     //Get the current diary(selected in DiariesActivity)
75     ISharedPreferences prefs = this.GetSharedPreferences(MY_PREFS,
        FileCreationMode.Private);
76     string curTable = prefs.GetString("TableName",null);

77     //Check if a diary is selected
78     if(curTable != null){
79         //Create a new file
80         string path = System.Environment.GetFolderPath (System.Environment.
        SpecialFolder.Personal);
81         Filename = "story_"+GetCurrentMilliseconds()+".txt";

82         //Get the content of story and save it as a txtfile
83         using (var f = new StreamWriter (OpenFileOutput (Filename,
        FileCreationMode.Append | FileCreationMode.WorldReadable))) {
84             f.WriteLine (editTxt_Story.Text);
85         }

86         //The date is the id of each entry
87         DateTime time = DateTime.Now;

88         time.ToShortDateString();
89         //Store the pathnames as a contentvalue

```

```

90     ContentValues content = new ContentValues();
91     content.Put("id",time.Day+"_"+time.Month+"_"+time.Year);
92     content.Put("title",editTxt_Title.Text);
93     content.Put("story",Filename);
94     content.Put("image",camera_file);
95     content.Put ("sound",sound_file);
96     content.Put("longitude",currentLoc.Longitude);
97     content.Put("latitude", currentLoc.Latitude);

98     //Save the entry in the database
99     ((DigitalDiaryApplication)Application).EntryRepository.AddEntry(
        curTable,content);

100    //Reset the fields
101    editTxt_Story.Text = "";
102    editTxt_Title.Text = "";

103    }else{
104        //TODO: ask the user to select a diary
105    }

106    };
107    //Resets the content
108    btn_reset.Click += delegate(object sender, EventArgs e) {

109        //Reset the fields
110        editTxt_Story.Text = "";
111        editTxt_Title.Text = "";
112        //TODO: remove camera and sound file

113    };
114    //Launch the soundrecorder activity so the user is able to record voice
115    btn_voice.Click += delegate(object sender, EventArgs e) {

116        Intent intent = new Intent();
117        intent.SetClass (this, typeof (SoundRecorder));
118        StartActivityForResult(intent,1);

119    };
120    //Button to start tracking the position
121    btn_location.Click += delegate(object sender, EventArgs e) {
122        //Set the criterias, in this case no requirements are set
123        var locationCriteria = new Criteria();

124        locationCriteria.Accuracy = Accuracy.NoRequirement;
125        locationCriteria.PowerRequirement = Power.NoRequirement;

126        //Select the provider that best meets the criteria
127        string locationProvider = _locationmanager.GetBestProvider(
        locationCriteria, true);

128        //Request updates based on the set timerintervall and distances changes
129        _locationmanager.RequestLocationUpdates(locationProvider, 2000, 1, this
        );

130    };

```

```
131     }

132     /*
133     * This method is called after an activity is finished. In this application
134     * after the camera and sound recorder
135     * the path or the URI is passed to this method. The result code for the
136     * default camera activity is 0 and
137     * for the sound recorder it's 1.
138     *
139     protected override void OnActivityResult(int requestCode, Result resultCode,
140     Intent data){
141     base.OnActivityResult(requestCode, resultCode, data);
142     //Camera
143     if(requestCode == 0 && resultCode == Result.Ok){
144
145         //Create a name for the camerafile
146         camera_file = "/sdcard/Pictures/image_"+GetCurrentMilliseconds()+".png";
147         //Get the image and save as bitmap
148         var bitmap = (Android.Graphics.Bitmap) data.Extras.Get("data");
149         //Create the image file
150         using (var stream = File.Create(camera_file)){
151
152             bitmap.Compress(Android.Graphics.Bitmap.CompressFormat.Png, 0, stream);
153
154         }
155     }
156     //Soundrecorder
157     if(requestCode == 1 && resultCode == Result.Ok){
158
159         //Save the pathname of the file
160         sound_file = data.GetStringExtra("SoundPath");
161
162     }
163 }

164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```

```
168     /*
```

```
169      * In Resume if the location tracking is off, switch it on
170      * TODO: make a boolean to check that
171      *
172      protected override void OnResume ()
173      {
174          base.OnResume ();

175          //TODO: make the following lines of code to a method
176          var locationCriteria = new Criteria();
177          locationCriteria.Accuracy = Accuracy.NoRequirement;
178          locationCriteria.PowerRequirement = Power.NoRequirement;

179          string locationProvider = _locationmanager.GetBestProvider(
180              locationCriteria, true);
181          _locationmanager.RequestLocationUpdates(locationProvider, 2000, 1, this);
182      }

182      /*
183      * On Pause, if the device is stilling tracking for position, turn it off
184      * TODO: use a boolean as metioned in method OnResume
185      *
186      protected override void OnPause ()
187      {
188          base.OnPause ();
189          _locationmanager.RemoveUpdates(this);
190      }

191      /*
192      *Called when a location change has occured, this will store the current
193      position in
194      *the global var currentLoc
195      *
196      public void OnLocationChanged(Location location){
197          currentLoc = location;
198          /* DEBUG USE
199          editTxt_Story.Text = String.Format ("Latitude = {0}, Longitude = {1}",
200          currentLoc.Latitude, currentLoc.Longitude);
201          */
202      }

202      /* DEBUG USE
203      editTxt_Story.Text = String.Format ("Latitude = {0}, Longitude = {1}",
204      currentLoc.Latitude, currentLoc.Longitude);
205      *
206      public void OnProviderDisabled(string provider){
207      }

208      public void OnProviderEnabled(string provider){
209      }

210      public void OnStatusChanged(string provider, Availability status, Bundle
211      extra){
```

```
211     }

212     /*
213     *This method get the number of milliseconds elapsed since the system
    started. This is used to
214     *make every filename unique by including the time.
215     *
216     private int GetCurrentMilliseconds ()
217     {
218         return System.Environment.TickCount;
219     }
    
```

```
220     }
221 }
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;

5  using Android.App;
6  using Android.Content;
7  using Android.OS;
8  using Android.Runtime;
9  using Android.Views;
10 using Android.Widget;
11 using System.Timers;
12 using System.Threading;
13 using Android.Media;

14 /*
15  * DigitalDiary
16  * Author: Joseph Hejderup & Eghogho Ileso
17  * File: SoundRecordery.cs
18  * Version: 1.0
19  * Description: This Activity will display the
20  * elapsed time of the voice recording, when the user
21  * wants to end an recording the activity is closed.
22  *
23 namespace DigitalDiary
24 {
25     [Activity (Label = "SoundRecorder")]
26     public class SoundRecorder : Activity
27     {
28         private System.Timers.Timer timer;
29         private TimeSpan _offset;
30         private DateTime _startTime;
31         private MediaRecorder recorder;
32         private string FileName = null;
33         bool isStarted = false;
34         private TextView tv_timer;

35         protected override void OnCreate (Bundle bundle)
36         {
37             base.OnCreate (bundle);

38             SetContentView (Resource.Layout.soundrecorder);
39             //Create the filename
40             FileName = "/sdcard/sound_"+GetCurrentMilliseconds()+".aac";
41             Button btn_Start = FindViewById<Button> (Resource.Id.btn_Start);

42             //Start the recording
43             //TODO: make a method of this instead
44             btn_Start.Click += delegate(object sender, EventArgs e) {

45                 //Check if SD card is mounted
46                 if (Android.OS.Environment.ExternalStorageState.Equals (Android.OS.
                    Environment.MediaMounted)) {
47                     -if (isStarted) {
```

```

48         timer.Stop();
49         recorder.Stop();
50         recorder.Reset();
51         recorder.Release();
52         recorder = null;
53         //Create a new intent and insert the path name
54         Intent returnIntent = new Intent();
55         returnIntent.PutExtra("SoundPath", FileName);
56         //The activity who has started this activity, will receive data to
OnActivityResult
57         SetResult(Result.Ok, returnIntent);
58         Finish ();
59     }
60     else {
61         //Create a new recorder
62         recorder = new MediaRecorder();
63         recorder.SetAudioSource(AudioSource.Mic);
64         recorder.SetOutputFormat(OutputFormat.Mpeg4);
65         recorder.SetAudioEncoder(AudioEncoder.Default);

66         recorder.SetOutputFile(FileName);
67         recorder.Prepare();
68         //Start the timer of the recording
69         _offset = TimeSpan.Parse (tv_timer.Text);
70         _startTime = DateTime.Now;
71         timer = new System.Timers.Timer (1000.0);
72         timer.Elapsed += new ElapsedEventHandler (elapsed_timer);
73         timer.Start ();
74         recorder.Start();
75         isStarted = true;
76         btn_Start.Text = "Stop";
77     }
78 }

79 };

80 tv_timer = FindViewById<TextView> (Resource.Id.textview_timer);
81 Button btn_Stop = FindViewById<Button> (Resource.Id.btn_Stop);
82 //Finish the recording
83 btn_Stop.Click += delegate(object sender, EventArgs e) {
84     this.Finish ();
85 };

86 _offset = new TimeSpan (0);
87 _startTime = DateTime.Now;
88 //Set the buffer as normal
89 tv_timer.SetText ("00:00:00", TextView.BufferType.Normal);
90 }

91 /*
92  * This method is called every time the timer has elapsed
93  * This will update the textview with the new time
94  */
95 private void elapsed_timer (object sender, ElapsedEventArgs e)
96 {

```

```
97     TimeSpan span = (DateTime.Now - _startTime) + _offset;
98     RunOnUiThread (() => tv_timer.SetText (String.Format ("
99         {0:00}:{1:00}:{2:00}",
100         (int)span.TotalHours, span.Minutes, span.Seconds), TextView.BufferType.
            Normal));
100     }

101     //Stop the activity from being rotated so that the timer/recording isn't
102     //stopped.
103     public override void OnConfigurationChanged (Android.Content.Res.
104         Configuration newConfig)
105     {
106         base.OnConfigurationChanged (newConfig);
107     }

108     /*
109     *This method get the number of milliseconds elapsed since the system
110     started. This is used to
111     *make every filename unique by including the time.
112     */
113     private int GetCurrentMilliseconds ()
114     {
115         return System.Environment.TickCount;
116     }
117 }
```

```
1  using System;
2  using Android.App;
3  using Android.Runtime;
4  using DigitalDiary.DataMangement;

5  /*
6   * DigitalDiary
7   * Author: Joseph Hejderup & Eghogho Ileso
8   * File: DigitalDiaryApplication.cs
9   * Version: 1.0
10  * Description: This will allow all the activites to
11  * gain access to the database from one place
12  *
13  *
14  namespace DigitalDiary
15  {
16      [Application]
17      class DigitalDiaryApplication: Application
18      {
19          public IEntryRepository EntryRepository{
20              }

21          public DigitalDiaryApplication(IntPtr handle, JniHandleOwnership transfer):
22              base(handle, transfer){

23          public override void OnCreate(){
24              base.OnCreate();
25              EntryRepository = new AndroidEntryRepository(this);
26          }

27      }
28  }
```

```
1  using System.Collections.Generic;
2  using DigitalDiary;
3  using Android.Content;

4  /*
5   * DigitalDiary
6   * Author: Joseph Hejderup & Eghogho Ileso
7   * File: IEntryRepository.cs
8   * Version: 1.0
9   * Description: This interface contains the nescessary methods
10  * that are needed to manage diaries and entries.
11  */
12  namespace DigitalDiary.DataMangement
13  {
14      public interface IEntryRepository
15      {
16          //Get the entries according to entered diary(tablename)
17          IList<Diary> GetEntries(string tablename);

18          //Adds a new entry to the specified diary
19          long AddEntry(string tablename, ContentValues contents);

20          //Adds a new diary to the table containing diaries
21          long AddTable(string name);

22          //Get all the diaries in the system
23          IList<string> GetTables();

24          //Deletes an entry according to the specified diary and the date(id) (date is
           the primarykey)
25          long RemoveEntry(string tablename, int id);

26          //Remove a specified table
27          long RemoveTable(string name);

28          //Create a new table
29          void CreateTable(string name);
30      }
31  }
```

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;

5  using Android.App;
6  using Android.Content;
7  using Android.OS;
8  using Android.Runtime;
9  using Android.Views;
10 using Android.Widget;
11 using Android.Graphics;

12 /*
13  * DigitalDiary
14  * Author: Joseph Hejderup & Eghogho Ileso
15  * File: DiariesActivity.cs
16  * Version: 1.0
17  * Description: This activity allows the user to
18  * select a diary to later view the entries or write
19  * a new entry or add/delete a new diary to the system
20  */
21 //TODO: add an edit feature to change the title of a diary
22 namespace DigitalDiary.Activities
23 {
24     [Activity (Label = "DiariesActivity")]
25     public class DiariesActivity : Activity
26     {
27         private ListView list_view;
28         public static string MY_PREFS = "MY_PREFS";

29         protected override void OnCreate(Bundle bundle)
30         {
31             base.OnCreate(bundle);

32             SetContentView(Resource.Layout.List);

33             //The listview is used to display all the exsisting entries in the system
34             list_view = FindViewById<ListView>(Resource.Id.List);

35             Button btn_addDiary = FindViewById<Button>(Resource.Id.btn_addTable);

36             btn_addDiary.Click += delegate(object sender, EventArgs e) {
37                 //If the user wants to add a new diary, the Adddiary activity is started
38                 StartActivity(typeof(Activities.AddDiary));
39             };

40             //Single clicking will select a diary
41             list_view.ItemClick += new System.EventHandler<AdapterView.
42                 ItemClickEventArgs>(listItemClick);

43             //Clicking for a diary for a long time will ask the user to delete or not
44             delete the diary
45             list_view.ItemLongClick += delegate(object sender, AdapterView.
46                 ItemLongClickEventArgs e) {

```

```
44         //An alertdialog will assit the user to make the right choice
45         AlertDialog.Builder builder = new AlertDialog.Builder(this);
46         builder.SetMessage("Delete");
47         builder.SetTitle("Do you want to delete the selected diary?");
48         builder.SetPositiveButton("Yes",delegate{
49
50             //deletes the diary
51             ((DigitalDiaryApplication)Application).EntryRepository.RemoveTable(e.
52                 View.FindViewById<TextView>(Resource.Id.Title).Text);
53             //Refereshes the listview
54             refreshList();
55
56             ISharedPreferences prefs = GetPreferences (FileCreationMode.Append);
57             ISharedPreferencesEditor editor = prefs.Edit();
58             editor.PutInt("CurrentPosition",0);
59             //Set tableview as null
60             editor.PutString("TableName",null);
61             editor.Commit();
62
63         });
64         builder.SetNegativeButton( "Cancel",delegate{
65             //do nothing
66         });
67         builder.Show();
68     };
69 }
```

```
66     protected override void OnResume ()
67     {
68         base.OnResume ();
69
70         refreshList();
71     }
```

```
71     /*
72     * This method will refresh the listview with records stored in the database
73     *
74     private void refreshList()
75     {
76         var tables = ((DigitalDiaryApplication)Application).EntryRepository.
77             GetTables();
78         list_view.Adapter = new TableListAdapter(this, tables);
79     }
```

```
79     /*
80     *Called upon item-click event, the clicked diary is stored in a shared
81     preference file
82     *so it can be accesed everywhere
83     *
84     private void listItemClick(object sender, AdapterView.ItemClickEventArgs e)
85     {
```

```
85      ISharedPreferences prefs = this.getSharedPreferences(MY_PREFS,
      FileCreationMode.Private);

86      ISharedPreferencesEditor editor = prefs.Edit();
87      //To track the current position
88      editor.PutInt("CurrentPosition",e.Position);
89      editor.Commit();
90      string table = e.View.FindViewById<TextView>(Resource.Id.Title).Text;
91      //save the current tablename
92      editor.PutString("TableName",table);
93      editor.Commit();

94      //To indicate which diary is selected, the selected diary is colored as
      dimgrey

95      for(int i=0; i< e.Parent.ChildCount; i++){
96          if(i == e.Position){
97              e.Parent.GetChildAt(i).SetBackgroundColor(Color.DimGray);

98          }else{
99              e.Parent.GetChildAt(i).SetBackgroundColor(Color.WhiteSmoke);
100          }
101      }
102  }
103  }
104  }
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;

5  using Android.App;
6  using Android.Content;
7  using Android.OS;
8  using Android.Runtime;
9  using Android.Views;
10 using Android.Widget;

11 /*
12  * DigitalDiary
13  * Author: Joseph Hejderup & Eghogho Ileso
14  * File: EntriesActivity.cs
15  * Version: 1.0
16  * Description: This activity allows the user to
17  * select an entry in the diary and read the content
18  * of it
19  *
20 namespace DigitalDiary.Activities
21 {
22     [Activity (Label = "EntryActivity")]
23     public class EntriesActivity : Activity
24     {
25         private ListView list_view;
26         public static string MY_PREFS = "MY_PREFS";
27         public static string SH_ENTRY = "SH_ENTRY";
28         private IList<Diary> entries;

29         protected override void OnCreate (Bundle bundle)
30         {
31             base.OnCreate (bundle);
32             SetContentView(Resource.Layout.EntryList);
33             //Connect the listview in the layout to the code and attach a click
trigger
34             list_view = FindViewById<ListView>(Resource.Id.EntriesList);
35             list_view.ItemClick += new System.EventHandler<AdapterView.
ItemClickEventArgs>(listItemClick);
36         }

37         /*
38         *Called once the user click on a entry
39         *This method will store the content of the diary in preference file
40         *TODO: When parable is supported for MonoDroid, use this technique instead
41         *
42         private void listItemClick(object sender, AdapterView.ItemClickEventArgs e){
43             //Retrieve the selected entry
44             Diary diary = entries.ElementAt(e.Position);

45             //Store all the data from the diary object in the preference file
46             ISharedPreferences prefs = this.GetSharedPreferences(SH_ENTRY,
FileCreationMode.Private);
```

```
47     ISharedPreferencesEditor editor = prefs.Edit();
48     editor.PutString("camerapath", diary.Image);
49     editor.PutString("title",diary.Title);
50     editor.PutString("story",diary.Story);
51     editor.PutString("soundpath",diary.Sound);
52     editor.PutString("longitude",diary.Longitude);
53     editor.PutString("latitude",diary.Latitude);
54     //Store the data
55     editor.Commit();

56     //Launch the showentry so the user is able to view the content of it
57     StartActivity(typeof(Activities.ShowEntry));
58 }
```

```
59     protected override void OnResume()
60     {
61         base.OnResume();

62         refreshList();
63     }
```

```
64     /*
65     *Refreshes the list view by getting the current selected diary from the
66     *and populating the content of the diary in the listview, if no diary is
67     *selected the
68     *listview is empty
69     */
69     private void refreshList()
70     {
71         ISharedPreferences prefs = this.GetSharedPreferences(MY_PREFS,
72             FileCreationMode.Private);

72         string curTable = prefs.GetString("TableName",null);

73         if(curTable != null){
74             entries = ((DigitalDiaryApplication)Application).EntryRepository.
75                 GetEntries(curTable);
76             list_view.Adapter = new EntriesListAdapter(this, entries);
77         }
```

```
78     }
79 }
```

```
1  using System;
2  using System.Collections.Generic;
3  using Android.Content;
4  using DigitalDiary;
5  using Android.Database;

6  /*
7   * DigitalDiary
8   * Author: Joseph Hejderup & Eghogho Ileso
9   * File: AndroidEntryRepository.cs
10  * Version: 1.0
11  * Description: An Android implementation of IEntryRepository
12  *
13  namespace DigitalDiary.DataMangement
14  {
15      public class AndroidEntryRepository: IEntryRepository
16      {
17          private DiaryDatabaseHelper _helper;

18          public AndroidEntryRepository(Context context){
19              _helper = new DiaryDatabaseHelper(context);
20          }

21          /*
22           * Saves a diary(table name) in the database
23           * @name: name of the diary
24           * TODO: name change
25           *
26           public long AddTable(string name){
27               var values = new ContentValues();
28               values.Put("name",name);
29               return _helper.WritableDatabase.Insert("DiaryTable", null,values);
30           }

31          /*
32           * Add an entry to the database
33           * @content: the contentvalue should contain path names of different files
34           etc
35           * @tablename: name of the diary
36           *
37           public long AddEntry(string tablename, ContentValues content ){
38               return _helper.WritableDatabase.Insert(tablename, null,content);
39           }

39          /*
40           * Returns a list containing all the entries from a specified diary
41           * @tablename: name of the diary
42           *
43           public IList<Diary> GetEntries(string tablename){
44               //Query the content of the diary
45               ICursor entryCursor = _helper.ReadableDatabase.Query(tablename,null,null,
46                           null,null,null,null);
```

```

46     var entries = new List<Diary>();
47     while(entryCursor.MoveNext()){
48         //Add each entry in the list
49         entries.Add(MakeEntryObject(entryCursor));
50     }

51     return entries;
52 }

```

```

53     /*
54     * Remove a diary(table name) in the database
55     * We are using two tables, therefore we need to delere the actual table
56     * and the table which tracks diaries
57     * @name: name of the diary
58     * TODO: name change
59     *
60     public long RemoveTable(string tablename){
61         _helper.WritableDatabase.ExecSQL("DROP TABLE "+tablename);
62         return _helper.WritableDatabase.Delete("DiaryTable","name=?", new string[]
63             {tablename});
64     }

```

```

64     /*
65     * Delete an entry based on the id and tablename
66     * @tablename: name of the diary
67     * @id: date of the diary(primary key of the db)
68     * TODO: name change
69     *
70     public long RemoveEntry(string tablename, int id){
71         return _helper.WritableDatabase.Delete(tablename,"id="+id, null);
72     }

```

```

73     /*
74     * Creates a new diary so the user can enter data
75     * @name: name of the diary
76     * TODO: include this method in AddTable instead, or make a new one
77     *
78     public void CreateTable(string name){
79         _helper.WritableDatabase.ExecSQL("CREATE TABLE " + name + " (id TEXT
80             PRIMARY KEY," +
81             "title TEXT NOT NULL, story TEXT, image TEXT," +
82             "sound TEXT, longitude TEXT, latitude TEXT)");
83     }

```

```

83     /*
84     * Get all the diaries in the syste,
85     * TODO: name change
86     *
87     public IList<string> GetTables(){
88         ICursor entryCursor = _helper.ReadableDatabase.Query("DiaryTable",null,null,
89             null,null,null,null);
90         var tablenamees = new List<string>();

```

```
90     while (entryCursor.MoveNext()) {
91         tablenames.Add(entryCursor.GetString(0));
92     }
93     return tablenames;
94 }

95     /*
96     * This helper method creates a new diary object using
97     * the queried data
98     * @curser: curser containing information about an entry
99     * TODO: name change insted of Diary call it Entry
100    */
101    private Diary MakeEntryObject(ICursor cursor) {
102        return new Diary {

103            Id = cursor.GetInt(0),
104            Title = cursor.GetString(1),
105            Story = cursor.GetString(2),
106            Image = cursor.GetString(3),
107            Sound = cursor.GetString(4),
108            Longitude = cursor.GetString(5),
109            Latitude = cursor.GetString(6)

110        };
111    }

112 }
113 }
```

```
1  using Android.Content;
2  using Android.Database.Sqlite;

3  /*
4   * DigitalDiary
5   * Author: Joseph Hejderup & Eghogho Ileso
6   * File: DiaryDatabaseHelper.cs
7   * Version: 1.0
8   * Description: This class maintains database creation
9   * and version management
10  * TODO: how to manage tables that are not created here
11  */
12  namespace DigitalDiary.DataMangement
13  {
14      public class DiaryDatabaseHelper : SQLiteOpenHelper
15      {
16          private const string DB_NAME = "DigitalDiary";
17          private const int DB_VERSION = 1;

18          public DiaryDatabaseHelper(Context context): base(context,DB_NAME,null,
19              DB_VERSION) {
20
21          }

22          public override void OnCreate(SQLiteDatabase db)
23          {
24              //This will create the table to maintain the created diaries
25              db.ExecSQL(@"CREATE TABLE DiaryTable (name TEXT PRIMARY KEY)");
26          }

27          public override void OnUpgrade(SQLiteDatabase db, int oldVer, int newVer){
28              db.ExecSQL("DROP TABLE DiaryTable");
29              OnCreate(db);
30          }
31      }
32  }
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.IO;
6  using Android.App;
7  using Android.Content;
8  using Android.OS;
9  using Android.Runtime;
10 using Android.Views;
11 using Android.Widget;
12 using Android.Graphics;
13 using Android.Graphics.Drawables;
14 using Android.Media;
15 using Java.IO;
16 using Android.GoogleMaps;

17 //Laptop: android:apiKey="0EO0KDRH2nMbc5YL56UDvEiPgkvgmCH2zwFRVig"
18 //stationÃr: android:apiKey="0EO0KDRH2nMZCIvzkrImkQYNQrJdfg5XQeXzVPw"
19 /*
20 * DigitalDiary
21 * Author: Joseph Hejderup & Eghogho Ileso
22 * File: ShowEntry.cs
23 * Version: 1.0
24 * Description: Displays the content of an entry
25 * The user is able to read the content and play
26 * the saved sound recording.
27 *
28 namespace DigitalDiary.Activities
29 {
30     [Activity (Label = "ShowEntry", Theme="@android:style/Theme.Light.NoTitleBar")]
31     public class ShowEntry : MapActivity, Android.Media.MediaPlayer.
32         IOnCompletionListener
33     {
34         private TextView showTitle;
35         private ImageView showImage;
36         private Button btn_stop;
37         private Button btn_start;
38         private TextView showStory;
39         private MediaPlayer mPlayer;
40         //Name of shared preferences containing all the paths to various files
41         public static string SH_ENTRY = "SH_ENTRY";

42         protected override bool IsRouteDisplayed {
43             get {
44                 return false;
45             }
46         }

47         protected override void OnCreate (Bundle bundle)
48         {
49             base.OnCreate (bundle);
50             SetContentView(Resource.Layout.Entry);

51             //Get the preferences
52             ISharedPreferences prefs = this.GetSharedPreferences (SH_ENTRY,
```

```

        FileCreationMode.Private);
51    //Initiate GUI components
52    InitiateComponents();
53    //Attach the mapview
54    var map = FindViewById<MapView>(Resource.Id.map);
55    //make it non-clickable
56    map.Clickable = false;
57    //Animate to the saved location
58    map.Controller.AnimateTo( new GeoPoint ((int) (Convert.ToDouble(prefs.
        GetString("latitude",null))*1E6),
59        (int) (Convert.ToDouble(prefs.GetString("longitude",null))*1E6)));
60    //Zoom x16 times
61    map.Controller.SetZoom(16);
62    //Add a marker to the saved position
63    var marker = Resources.GetDrawable (Resource.Drawable.androidmarker);
64    var androidOverlay = new AndroidItemizedOverlay (marker,prefs.GetString("
        longitude",null),prefs.GetString("latitude",null));
65    map.Overlays.Add (androidOverlay);

66    //Set the title,story and image
67    showTitle.Text = prefs.GetString("title","");
68    SetStory(prefs.GetString("story",""));
69    SetImage(prefs.GetString("camerapath",""));

70    btn_stop.Enabled = false;

71    //Start the media player
72    btn_start.Click += delegate(object sender, EventArgs e) {

73        FileInputStream fileInputStream = new FileInputStream(prefs.GetString("
        soundpath",""));
74        mPlayer = new MediaPlayer();

75        if(mPlayer != null){
76            mPlayer.SetDataSource(fileInputStream.FD);
77            mPlayer.SetOnCompletionListener(this);
78            mPlayer.Prepare();
79            btn_start.Enabled = false;
80            mPlayer.Start();
81            btn_stop.Enabled = true;
82        }

83    };
84    //Stop the media player
85    btn_stop.Click += delegate(object sender, EventArgs e) {

86        mPlayer.Stop();
87        mPlayer.Release();
88        btn_stop.Enabled = false;
89        btn_start.Enabled = true;

90    };
91 }

92  /*

```

```
93      * GUI Compnents
94      *
95      private void InitiateComponents() {
96          showTitle = FindViewById<TextView>(Resource.Id.showTitle);
97          showImage = FindViewById<ImageView>(Resource.Id.showImage);
98          showStory = FindViewById<TextView>(Resource.Id.showStory);
99          btn_stop = FindViewById<Button>(Resource.Id.btnStop);
100         btn_start = FindViewById<Button>(Resource.Id.btnPlay);
101     }
```

```
102     /*
103     * Reads the text file and displays the content of it
104     * @para: path of text file
105     *
106     private void SetStory(string storypath) {
107         string path = System.Environment.GetFolderPath (System.Environment.
108             SpecialFolder.Personal);
109         string Filename = storypath;
110         string filename = System.IO.Path.Combine (path, Filename);
111         //Validate if the file exist
112         if (System.IO.File.Exists (filename)) {
113             using (var f = new StreamReader (OpenFileInput (Filename))) {
114                 string line;
115                 do {
116                     line = f.ReadLine ();
117                 } while (!f.EndOfStream);
118                 //add each line off the steam to showEntry
119                 showStory.Text = line;
120             }
121         }
122     }
```

```
122     /*
123     * Add the image to the imageviewer
124     * @para: image filepath
125     *
126     private void SetImage(string imagepath) {
127         if (System.IO.File.Exists(imagepath)) {
128             Bitmap myBitmap = BitmapFactory.DecodeFile(imagepath);
129             showImage.SetImageBitmap(myBitmap);
130         }
131     }
```

```
132     /*
133     * After the media player has played the recording
134     * Inactivate the stop-btn and activate the start-btn
135     *
136     public void OnCompletion(MediaPlayer mp) {
137         mp.Stop();
138         mp.Release();
139         btn_start.Enabled = true;
140         btn_stop.Enabled = false;
141     }
```

```
142     }

143     /*
144     * This private class is used for setting and displaying the markers in the map
145     */
146     class AndroidItemizedOverlay: ItemizedOverlay
147     {
148         List<OverlayItem> _items;

149         public AndroidItemizedOverlay (Drawable androidmarker, string lotude, string
            latude) : base(androidmarker)
150         {
151             //Convert string to double
152             double lo = Convert.ToDouble(lotude)*1E6;
153             double lat = Convert.ToDouble(latude)*1E6;

154             //Add the position to the overlayitem list
155             _items = new List<OverlayItem>{
156                 new OverlayItem (new GeoPoint ((int)lat,
157                     (int)lo), null, null)
158             };

159             BoundCenterBottom(androidmarker);
160             Populate();
161         }

162         protected override Java.Lang.Object CreateItem (int i)
163         {
164             var item = _items[i];
165             return item;
166         }

167         public override int Size ()
168         {
169             return _items.Count();
170         }

171     }
172 }
```

```
1  using System.Collections.Generic;
2  using System.Configuration;
3  using System.Linq;
4  using Android.App;
5  using Android.Views;
6  using Android.Widget;
7  using Android.Graphics;
8  using Android.Preferences;
9  using Android.Content;
10 using DigitalDiary;

11 /*
12  * DigitalDiary
13  * Author: Joseph Hejderup & Eghogho Ileso
14  * File: EntriesListAdapter.cs
15  * Version: 1.0
16  * Description: This adapter will manage the
17  * a listview to display entries from a selected
18  * diary
19  *
20 namespace DigitalDiary
21 {
22     public class EntriesListAdapter : BaseAdapter
23     {
24         private IEnumerable<Diary> entriesList;
25         private Activity context;
26         public static string MY_PREFS = "MY_PREFS";

27         /*
28          * Number of entries
29          *
30         public override int Count{
31             get { return entriesList.Count(); }
32         }

33         /*
34          * Constructor
35          * @context: current activity
36          * @entries: list of entries from a selected diary
37          *
38         public EntriesListAdapter(Activity context, IEnumerable<Diary> entries){
39             this.context = context;
40             entriesList = entries;
41         }

42         /*
43          * The view inside the listview will only display the title of the entry
44          *
45         public override View GetView(int position, View convertView, ViewGroup
46             parent){
47             var view = (convertView
48                 ?? context.LayoutInflater.Inflate(
49                     Resource.Layout.Viewtablelist, parent, false)
50                 ) as LinearLayout;
```

```
50      //Get the entry from the list
51      Diary diary = entriesList.ElementAt(position);
52      //set the title in the list according to the title in the diary object
53      view.FindViewById<TextView>(Resource.Id.Title).Text = diary.Title;
54      return view;
55  }
```

```
56      /*
57      * Get the diary object at a selected position in the list
58      */
59      public Diary GetEntryAt(int position){
60          return entriesList.ElementAt(position);
61      }
```

```
62      public override Java.Lang.Object GetItem(int position){
63          return null;
64      }
```

```
65      public override long GetItemId(int position){
66          return position;
67      }
```

```
68  }
69  }
```

```
1  using System.Collections.Generic;
2  using System.Configuration;
3  using System.Linq;
4  using Android.App;
5  using Android.Views;
6  using Android.Widget;
7  using Android.Graphics;
8  using Android.Preferences;
9  using Android.Content;

10 /*
11  * DigitalDiary
12  * Author: Joseph Hejderup & Eghogho Ileso
13  * File: EntriesListAdapter.cs
14  * Version: 1.0
15  * Description: This adapter will manage the
16  * a listview to display diaries and also mark the
17  * selected tab
18  * diary
19  *
20 namespace DigitalDiary
21 {
22     public class TableListAdapter : BaseAdapter
23     {
24         private IEnumerable<string> tableList;
25         private Activity context;
26         public static string MY_PREFS = "MY_PREFS";

27         public override int Count
28         {
29             get { return tableList.Count(); }
30         }

31         /*
32         * Constructor
33         * @context: activity
34         * @tables: list containing all the diaries
35         *
36         public TableListAdapter(Activity context, IEnumerable<string> tables){
37             this.context = context;
38             tableList = tables;
39         }

40         /*
41         * The listview will only display the title of the diary and also mark
42         * the selected one
43         *
44         public override View GetView(int position, View convertView, ViewGroup
45         parent){
46             var view = (convertView
47             ?? context.LayoutInflater.Inflate(
48             Resource.Layout.Viewtablelist, parent, false)
49             ) as LinearLayout;
```

```
49      //Get the position of the diary the user has seleted, this is done in the
      DiariesActivity
50      ISharedPreferences prefs = context.getSharedPreferences (MY_PREFS,
      FileCreationMode.Private);
51      int curPosition = prefs.GetInt ("CurrentPosition", 0);
52      //Set the title
53      string table_name = tableList.ElementAt (position);
54      view.FindViewById<TextView>(Resource.Id.Title).Text = table_name;

55      //If the position is the selected diary, change the colot to dimgray
56      if (curPosition >= 0 && position == curPosition) {
57          view.FindViewById<LinearLayout>(Resource.Id.Layout).SetBackgroundColor (
      Color.DimGray);
58      }

59      return view;
60  }

61  public string GetTableName(int position)
62  {
63      return tableList.ElementAt (position);
64  }

65  public override Java.Lang.Object GetItem(int position)
66  {
67      return null;
68  }

69  public override long GetItemId(int position)
70  {
71      return position;
72  }

73  }
74  }
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;

5  using Android.App;
6  using Android.Content;
7  using Android.OS;
8  using Android.Runtime;
9  using Android.Views;
10 using Android.Widget;

11 /*
12  * DigitalDiary
13  * Author: Joseph Hejderup & Eghogho Ileso
14  * File: AddDiary.cs
15  * Version: 1.0
16  * Description: This activity is a modal window that
17  * will allow the user to enter a title of the diary and
18  * store this as a record in the db
19  */
20 //TODO: ADD Checks to the userinput
21 namespace DigitalDiary.Activities
22 {
23     [Activity (Label = "AddDiary")]
24     public class AddDiary : Activity
25     {
26         private EditText title;

27         protected override void OnCreate(Bundle bundle)
28         {
29             base.OnCreate(bundle);

30             SetContentView(Resource.Layout.AddDiary);

31             //EditText: user input for the user
32             title = FindViewById<EditText>(Resource.Id.Title);

33             //Button to save the diary in the database
34             FindViewById<Button>(Resource.Id.Save).Click += new EventHandler(
35                 Save_Click);
36         }

37         private void Save_Click(object sender, EventArgs e)
38         {
39             //Save the diary in the table of diaries
40             ((DigitalDiaryApplication)Application).EntryRepository.AddTable(title.Text.ToString());
41             //Create a new table so the user can add new entries to the newly created
42             diary
43             ((DigitalDiaryApplication)Application).EntryRepository.CreateTable(title.Text.ToString());

44             Finish();
45         }
46     }
47 }
```

43 }

44 }

45 }

```
1  /*
2  * DigitalDiary
3  * Author: Joseph Hejderup & Eghogho Ileso
4  * File: Diary.cs
5  * Version: 1.0
6  * Description: This class contains the properties
7  * for a Diary object. This class make it easier to transfer
8  * data to the showentry activity.
9  * TODO: change to Entry
10 */
11 namespace DigitalDiary
12 {
13     public class Diary
14     {
15         public int Id{get; set;}
16
17         public string Title{get; set;}
18
19         public string Story{get; set;}
20
21         public string Image{get; set;}
22
23         public string Sound{get; set;}
24
25         public string Longitude{get; set;}
26         public string Latitude{get; set;}
27
28     }
29 }
30
31
32
33
34
35
36
37
38
39
40
41
42
43
```

```
40     public string Title{
41     }
42
43     public string Story{
44     }
```

```
44     public string Image{  
45     }
```

```
46     public string Sound{  
47     }
```

```
48     public string Longitude{  
49     }
```

```
50     public string Latitude{  
51     }
```

```
52 }  
53 }
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;

5  using Android.App;
6  using Android.Content;
7  using Android.OS;
8  using Android.Runtime;
9  using Android.Views;
10 using Android.Widget;
11 using DigitalDiary;

12 /*
13  * DigitalDiary
14  * Author: Joseph Hejderup & Eghogho Ileso
15  * File: SplashActivity.cs
16  * Version: 1.0
17  * Description: SplashScreen
18  * TODO: Add an image
19  *
20 namespace DigitalDiary.Activities
21 {
22     [Activity (Label = "SplashActivity",MainLauncher = true, NoHistory = true)]
23
24     public class SplashActivity : Activity
25     {
26         protected override void OnCreate (Bundle bundle)
27         {
28             base.OnCreate (bundle);
29             StartActivity (typeof (TabHandler));
30         }
31     }
```

Appendix D

index.html

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4     <title>DigitalDiary</title>
5     <!-- jQuery Mobile setup -->
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7     <link rel="stylesheet" href="mylibs/jqueryMobile/jquery.mobile-1.1.0.min.css" />
8     <script src="mylibs/jquery-1.7.2.min.js"></script>
9     <script src="mylibs/jqueryMobile/jquery.mobile-1.1.0.min.js"></script>
10    <!-- Phonegap setup -->
11    <script type="text/javascript" charset="utf-8" src="js/cordova-1.6.0.js"></script>
12
13    <link rel="stylesheet" href="css/style.css" />
14    <script type="text/javascript" charset="utf-8" src="js/script.js"></script>
15
16 </head>
17 <body>
18 <!-- This is the homepage with id 'diaries' -->
19 <div data-role="page" id="diaries">
20
21
22     <div data-role="header" data-position="fixed">
23         <h1>DigitalDiary</h1>
24         <a href="#form" data-transition="slideup" class="ui-btn-right">Add</a>
25         <a href="#" data-role="button" id="diariesDelete" class="ui-btn-left">Delete</a>
26     </div>
27     <!-- The content should only contain a listview -->
28     <div data-role="content" id="mainContent" >
29         <ul id="listDiaries" data-role="listview" data-inset="true"></ul>
30     </div>
31
32     <div data-role="footer" data-position="fixed">
33     <!-- Navbar -->
34         <div id="tabs" data-role="navbar" class="nav-glyphish-example" >
35             <ul>
36                 <li><a href="#diaries" id="diaries" data-icon="custom"
37 class="ui-btn-active">Diaries</a></li>
38                 <li><a href="#newentry" id="newentry" data-icon="custom">New Entry</a></li>
39                 <li><a href="#entry" id="entry" data-icon="custom">Entries</a></li>
40             </ul>
41         </div>
42     </div>
43 <!-- This is the page with id 'diaries' -->
44 <div data-role="page" id="newentry">
45
46     <!-- Simple header with no extra information -->
47     <div data-role="header" data-position="fixed">
48         <h1>New Entry</h1>
49     </div>
50     <div data-role="content" >
51
52         <div id="formDiary">
53             <div data-role="fieldcontain" >
54                 <label>Title:</label>
55                 <input type="text" name="titleEntry" id="titleEntry" placeholder="Add a title to
56 the diary" />
57             </div>
58             <div data-role="fieldcontain" >
59                 <fieldset data-role="controlgroup" data-type="horizontal">
60                     <legend>Input:</legend>
61                     <button onclick="capturePhoto();">Camera</button>
```

```

61         <a href="#" data-role="button" >Voice</a>
62         <button onclick="getPosition();">Position</button>
63     </fieldset>
64 </div>
65 <div data-role="fieldcontain" >
66     <label>Story:</label>
67     <textarea id="descEntry" name="descEntry" rows="8" cols="50"
placeholder="Today was an amazing day.."></textarea>
68 </div>
69 <div data-role="fieldcontain" >
70     <a href="#" id="submit" data-role="button" data-theme="b">Save</a>
71 </div>
72 </div>
73 </div>
74
75 <div data-role="footer" data-position="fixed">
76     <!-- Navbar with -->
77     <div id="tabs" data-role="navbar" class="nav-glyphish-example">
78         <ul>
79             <li><a href="#diaries" id="diaries" data-icon="custom" >Diaries</a></li>
80             <li><a href="#newentry" id="newentry" data-icon="custom"
class="ui-btn-active">New Entry</a></li>
81             <li><a href="#entry" id="entry" data-icon="custom">Entries</a></li>
82         </ul>
83     </div>
84 </div>
85
86
87 </div>
88 <div data-role="page" id="entry">
89
90
91     <div data-role="header" data-position="fixed">
92         <h1>Entries</h1>
93     </div>
94     <!-- The content should only contain a listview -->
95     <div data-role="content">
96         <ul id="listEntries" data-role="listview" data-inset="true"></ul>
97     </div>
98     <div data-role="footer" data-position="fixed">
99         <!-- Navbar with -->
100         <div id="tabs" data-role="navbar" class="nav-glyphish-example">
101             <ul>
102                 <li><a href="#diaries" id="diaries" data-icon="custom" >Diaries</a></li>
103                 <li><a href="#newentry" id="newentry" data-icon="custom" >New Entry</a></li>
104                 <li><a href="#entry" id="entry" data-icon="custom" class="ui-btn-active"
>Entries</a></li>
105             </ul>
106         </div>
107     </div>
108
109
110 </div>
111 <div data-role="page" id="form">
112     <div data-role="header" data-position="fixed">
113         <h1>Add Diary</h1>
114     </div>
115     <div data-role="content" data-theme="b">
116
117         <div data-role="fieldcontain" >
118             <label>Title</label>
119             <input type="text" name="title" id="titleForm" placeholder="Add a title to the

```

```

    diary"/>
120     </div>
121
122     <div data-role="fieldcontain" >
123         <label>Description</label>
124         <textarea id="descForm" name="desc" rows="4" cols="50" placeholder="Add a
description"></textarea>
125     </div>
126
127     <div data-role="fieldcontain" >
128         <a href="#diaries" id="saveDiary" data-role="button" data-theme="b">Save</a>
129     </div>
130 </div>
131 <div data-role="footer" data-position="fixed">
132     <h1></h1>
133 </div>
134 </div>
135 <div data-role="page" id="listentry">
136 <div data-role="header" data-theme="b" data-position="fixed">
137     <h1>QR Code</h1>
138     <a href="#home" data-icon="arrow-L" data-rel="back">Back</a>
139 </div>
140 <div data-role="content">
141
142 </div>
143 <div data-role="footer" class="ui-bar" data-theme="b" data-position="fixed"
data-id="footer">
144 </div>
145 </div>
146 </body>
147 </html>

```

style.css

```
1 h2{
2   text-align:center;
3   color:#060;
4   margin:0;
5   margin-bottom:30px;
6 }
7
8
9   .nav-glyphish-example .ui-btn .ui-btn-inner { padding-top: 40px !important; }
10  .nav-glyphish-example .ui-btn .ui-icon { width: 30px!important; height: 30px!important;
margin-left: -15px !important; box-shadow: none!important; -moz-box-shadow: none!important;
-webkit-box-shadow: none!important; -webkit-border-radius: none !important; border-radius:
none !important; }
11  #diaries .ui-icon { background: url(../img/33-cabinet.png) 50% 50% no-repeat;
background-size: 24px 22px; }
12  #newentry .ui-icon { background: url(../img/10-medical.png) 50% 50% no-repeat;
background-size: 24px 22px; }
13  #entry .ui-icon { background: url(../img/42-photos.png) 50% 50% no-repeat;
background-size: 24px 22px; }
14
15
16
17
```

script.js

```
1 /*
2  * DigitalDiary
3  * Author: Joseph Hejderup & Eghogho Ileso
4  * File: script.js
5  * Description: This file contains all the logic that is used
6  * in jQuery Mobile. However this file needs to be divded into several appropriate files,
7  * the purpose of this code is testing, so no programming guidelines was taken in
  consideration.
8  */
9
10 //Local variables
11 var selectedTable = null;
12 var entryArray = new Array();
13 var watchID = null;
14 var entryID= null;
15
16
17 //Standard start method
18 window.addEventListener('load', function(){
19     // Wait for PhoneGap to load
20     document.addEventListener("deviceready", onDeviceReady, false);
21 }, false);
22
23 // PhoneGap is ready
24 function onDeviceReady() {
25
26     //Open the database
27     var db = window.openDatabase("Database", "1.0", "DigitaldiaryDB", 200000);
28     //Create table if it don't exist
29     db.transaction(creatTable,errorCB, successCB);
30     //Retrieve all the diaries and add it to the listview
31     db.transaction(getDiaries,errorCB, successCB);
32
33     //Assign buttons
34     var saveBtn = $("#saveDiary");
35     var deleteBtn = $("#diariesDelete");
36     var submitBtn = $("#submit");
37
38
39     saveBtn.click(function(){
40         //Store the diary as a record in the database
41         db.transaction(saveDiaryDB, errorCB, successCB);
42         //Refresh the listview
43         db.transaction(getDiaries,errorCB, successCB);
44
45     });
46
47     submitBtn.click(function(){
48
49         if(selectedTable != null){
50
51             //Save the user-entered data to the database
52             db.transaction(saveEntryDB, errorCB, successCB);
53         }else{
54
55             alert("select a diary first!");
56         }
57
58     });
59
60     deleteBtn.click(function(){
61         //Deletes a diary
```

script.js

```
62         deleteDiary();
63
64
65     });
66
67     //When the entry tab is selected, read values to the listview
68     $('a[href="#entry"]').click(function(){
69
70         if(selectedTable != null){
71
72             db.transaction(getDiaryEntries,errorCB, successCB);
73         }else{
74
75             alert("select a diary first!");
76
77         }
78     }
79
80
81     });
82
83 }
84
85
86
87 /////////////////////////////////////////////////// DIARY PAGE
88
89 /*
90  * Creates a table where diaries are saved
91  */
92 function creatTable(tx){
93
94     tx.executeSql('CREATE TABLE IF NOT EXISTS tableNames (title TEXT PRIMARY KEY, desc
95     TEXT)');
96 }
97
98 /*
99  * Retrieve the diaries from the database, if successfull diarieslistview
100 * is called and will populate the listview
101 */
102 function getDiaries(transaction){
103
104
105     transaction.executeSql('SELECT * FROM tableNames', [], diariesListView, errorCB);
106     console.log("read all values in the table successfully created");
107
108 }
109 /*
110 * The diaries are added to the listview with radiobuttons, so the user is able to select
111 * the diary of choice
112 */
113 function diariesListView(tx,result){
114
115
116     $('#listDiaries').empty();
117
118     if (result.rows.length == 0) {
119         console.log("no values in listview");
120     }else{
121
122         $.each(result.rows,function(index){
```

script.js

```

123         var row = result.rows.item(index);
124         $('#listDiaries').append('<li><input type="radio" name="diary"
id="activeShow"+index+' value="'+row['title']+'"/><a><h3
class="ui-li-heading">'+row['title']+'</h3><p
class="ui-li-desc">'+row['desc']+'</p></a></li>');
125         $('#listDiaries').listview("refresh");
126
127     });
128
129     $('#listDiaries').listview();
130     //This needs to be done due to dynamic injection
131     activateRadioButtons();
132 }
133 }
134 /*
135  * Save the diary in a seperate table add create a new table with the title of the diary to
save user-entred entris
136  */
137 function saveDiaryDB(tx){
138     var setTitle = $("#titleForm");
139     var setDesc = $("#descForm");
140
141     tx.executeSql('INSERT INTO tableNames (title, desc) VALUES ("' + setTitle.val() +
"', "' + setDesc.val() + '")');
142     tx.executeSql('CREATE TABLE IF NOT EXISTS ' + setTitle.val() +'(id INTEGER PRIMARY
KEY, title TEXT, story TEXT,image TEXT, sound TEXT, latitude TEXT, longitude TEXT)');
143     setTitle.val("");
144     setDesc.val("");
145
146
147 }
148
149 /*
150  * This method is called in diarisListview and will listen for 'click' events
151  * when a diary is selected it is stored in the variable selectedTable
152  */
153 function activateRadioButtons(){
154
155     $("#listDiaries input:radio[name=diary]").click(function() {
156
157         selectedTable=$('#input:radio[name=diary]:checked').val();
158         console.log(selectedTable);
159     });
160
161 }
162 /*
163  * This methods deletes a diary in the database, the listview is
164  * refreshed later
165  */
166 function deleteDiary(){
167     var db = window.openDatabase("Database", "1.0", "DigitaldiaryDB", 200000);
168     db.transaction(removeDiary,errorCB, successCB);
169     db.transaction(getDiaries,errorCB, successCB);
170 };
171 /*
172  * Deletes the specified diary that is stored in the varaiable selectedTable
173  */
174 function removeDiary(tx){
175
176     tx.executeSql('DELETE FROM tableNames WHERE title="' + selectedTable + '");
177 }
178

```

script.js

```
179 //////////// ENTRY PAGE
180 /*
181 * This will retrieve all records found in the variable selectedTable
182 * if the transaction is successfull the the entrieListview function is called
183 */
184 function getDiaryEntries(transaction){
185
186     transaction.executeSql('SELECT * FROM ' +selectedTable+ ' ', [], entriesListview,
        errorCallback);
187 }
188 /*
189 * This method will populate the listview with data from the arguments
190 */
191 function entriesListview(tx,result){
192
193
194     $('#listDiaries').empty();
195
196     if (result.rows.length == 0) {
197         console.log("no values in listview");
198     } else {
199
200         $.each(result.rows,function(index){
201             var row = result.rows.item(index);
202             $('#listEntries').append('<li><a href="#listentry?url='+row['title']+'><h3
                class="ui-li-heading">'+row['title']+'</h3></a></li>');
203             $('#listEntries').listview("refresh");
204
205         });
206         $('#listEntries').listview();
207     }
208 }
209 //////////// SAVE ENTRY PAGE
210 /*
211 * Saves user-entered data to the database
212 * position, image, title and story is stored
213 */
214 function saveEntryDB(tx){
215     var setTitle = $("#titleEntry");
216     var setDesc = $("#descEntry");
217     var setImage = window.localStorage.getItem("getPhoto");
218     var setLongitude= window.localStorage.getItem("longitude");
219     var setLatitude = window.localStorage.getItem("latitude");
220
221     console.log("save to the diary "+setTitle.val()+setDesc.val());
222
223     //use specific format of the date
224     var date = new Date();
225     var date_format = ""+date.getDate()+(date.getMonth()+1)+date.getFullYear();
226
227     //Save the data in the database
228     tx.executeSql('INSERT INTO ' +selectedTable+ ' (id,title, story,image,
        sound,latitude,longitude) VALUES ("' + date_format + "', "' + setTitle.val() + "', "' +
        setDesc.val() + "', "' + setImage +
        '","/hejsan/voice/", "' +setLatitude+"', "' +setLongitude+'")');
229
230     //Remove location listener and clear the text boxes
231     navigator.geolocation.clearWatch(watchID);
232     setTitle.val("");
233     setDesc.val("");
234 }
235 //////////// GENERAL DATABASE
```

script.js

```
236
237 //This called when an error occurs during a database transaction
238 function errorCB(tx, err) {
239     alert("Error processing SQL: "+err);
240 }
241 //This called when a successfull db transaction is made
242 function successCB() {
243     console.log("table successfully created");
244 }
245 ////////// CAMERA
246
247 /*
248 * This method will initiate the native camera, the quality and source is specified
249 * The method savePhoto is called after a successfull image was captured
250 */
251 function capturePhoto(){
252     navigator.camera.getPicture(savePhoto,null,{sourceType:1,quality:60,
253     destinationType: Camera.DestinationType.FILE_URI, ,
254     sourceType: source });
255 }
256 /*
257 * This method will save the image path in localstorage so it can be retrieved
258 * when saving an entry to the database
259 */
259 function savePhoto(data){
260
261     //The date is part of the filename
262     var date = new Date();
263     var date_format = ""+date.getDate()+(date.getMonth()+1)+date.getFullYear();
264
265     var cameraFile = 'photo'+ '_' +date_format+ '_' +selectedTable;
266
267     window.localStorage.setItem(cameraFile, data);
268     window.localStorage.setItem("getPhoto", cameraFile);
269
270 }
271 ////////// POSITION
272 /*
273 * Start wathing the position with an frequency of 3000
274 */
275 function getPosition() {
276     watchID = navigator.geolocation.watchPosition(successPosition, errorPosition, {
277     frequency: 3000 });
278 }
279 /*
280 * The position is saved in the localstorage, the frequency determines how often to check
281 * and store the position
282 */
281 function successPosition(position) {
282
283     window.localStorage.setItem("longitude", position.coords.longitude);
284     window.localStorage.setItem("latitude", position.coords.latitude );
285 }
286 /*
287 * Display an error message
288 */
289 function errorPosition(error) {
290     alert(error.message);
291 }
292
293 ////////// SHOW ENTRY
294 /*
```

script.js

```
295 * Select the selected entry and add it to the show entry window
296 */
297 function getEntry(transaction){
298
299
300     transaction.executeSql('SELECT * FROM ' +selectedTable+ 'WHERE title=?'+entryID+'', [],
        addtoList, errorCallback);
301     console.log("read all values in the table successfully created-entries");
302
303 }
304 /*
305 * Save the values to an array list
306 */
307 function addtoList(tx,result){
308
309
310
311     $.each(result.rows,function(index){
312         var row = result.rows.item(index);
313         entryArray[0]=row['id'];
314         entryArray[1]=row['story'];
315         entryArray[3]=row['image'];
316         entryArray[4]=row['latitude'];
317         entryArray[5]=row['longitude'];
318
319         console.log("adding to array list");
320     });
321
322 }
323 /*
324 * The show entry page displays dynamic content, therefore before showing the page
325 * the selected content has to injected to the page.
326 */
327 function showEntry(urlObj, options) {
328
329
330     // Get the url parameter
331     var qrUrl = decodeURIComponent(urlObj.hash.replace(/.*url=/, ""));
332
333     // The id of the page we are going to write the content into is specified in the
    hash before the '?'.
334     var pageSelector = urlObj.hash.replace(/\?.*$/, "");
335
336
337     if (qrUrl) {
338         // Get the page we are going to write content into.
339         var $page = $(pageSelector);
340
341         // Get the header for the page.
342         var $header = $page.children(":jqmData(role=header)");
343
344         // Find the h1 element in the header and inject the hostname from the url.
345         $header.find("h1").html(qrUrl);
346
347         // Get the content area element for the page.
348         var $content = $page.children(":jqmData(role=content)");
349
350         entryID=qrUrl;
351
352         var db = window.openDatabase("Database", "1.0", "DigitaldiaryDB", 200000);
353         db.transaction(getEntry,errorCB, successCB);
354     }
```

script.js

```
355         //TODO: add the content to the page
356
357         $.mobile.changePage($page, options);
358     }
359
360 }
361 //Listens for 'pagebeforechange' events, this event is triggered every time a page is
    changed
362 $(document).bind("pagebeforechange", function(e, data) {
363     // We only want to handle changePage() calls where the caller is asking to load a page
    by URL.
364     if (typeof data.toPage === "string") {
365         // We only want to handle #listentryurl.
366         var u = $.mobile.path.parseUrl(data.toPage);
367         var entry = /^#listentry/;
368         if (u.hash.search(entry) !== -1) {
369             // Display the showentry page
370             showEntry(u, data.options);
371             e.preventDefault();
372         }
373     }
374 });
375
376
377
378
```