Institiúid Teicneolaíochta Cheatharlach

INSTITUTE of TECHNOLOGY CARLOW

At the Heart of South Leinster

Research Manual

For

iPhone Application

Submission Date: -

1st December 2009

Prepared by: -

Tuna Erdurmaz (C00115609)

Supervisor: -

Paul Barry

Table of Contents

1. Introduction

This Research Manual is about 4th year Software Engineering project for developing iPhone application. The manual contains general view about the different mobile phone development platforms which mainly reflects the significant aspects of the iPhone which is a smartphone designed and marketed by Apple that we are aiming to develop application for this mobile platform. There is then description about the iMac which is a range of all-in-one Macintosh desktop computers designed and build by Apple that is the only computer makes us available the development kit called iPhone SDK in order to develop iPhone application. There is also section about Objective-C which is a programming language that we are going to use and other section called Xcode which is a development environment we will also use for developing iPhone app on Mac OS X which involves Interface Builder, iPhone Simulator and Instruments & Shark in it. Finally, there is section about categories of different types of iPhone applications. Paper also contains detailed appendix which contains the information about the relevant sections.

2. The Mobile Phone as a Development Platform

After investigations about the mobile phone as a development platform, I can conclude that current desktop application developers have an advantage while choosing the platform (or Operating System) to develop by comparison with the mobile application developers since desktop developers have limited choices of OS and each compatible with the others. Mobile application developers are forced to choose between Android, BlackBerry, iPhone, Palm webOS, Symbian, and Windows mobile. So, in my opinion it is very difficult and crucial to determine the mobile platform to develop. Difficult since there are many platforms which has little commonality in between and each is incompatible with the others. Crucial since the mobile platforms limits the choice of tools and languages that are available, so it can be a factor to decrease your productivity in the meantime. If you want your application gain popularity and to earn good amount of money from your application, platform you have chosen will also be important in order to build your application (the widely used platform will bring more popularity to your application). Nowadays, there are plenty of iPhone apps available in the Apple store and continue to increase everyday since the most popular mobile platform today. Since my project is developing iPhone app, it is great opportunity for me to jump into that popular mobile development field.

I should take into consideration, distinction between the mobile applications and desktop applications. The mobile application is not small portable desktop application that the mobile platform has less, limited resources (Memory, CPU, etc.) than the desktop platform. So, I should be careful about the memory, my application will deal with. Mobile phones are also task-focused that applications intended to spend minimal user time. The user should not spend lots of time while using the application. For instance, the user just start the application which shows the main functionality immediately- task-focused, perform his/her task and put it in his pocket (game apps are slightly different). iPhone can only run one application at a time while Symbian OS can perform multiple-task which is currently used by most of the mobile manufacturers. Therefore, it is also important to take into account for my project that if the phone rings, a text message comes in, etc., my application will exit and the user will move another application. I must be able to put users back into reasonable spot when they come back to my application.

iPhone and Blackberry softwares are the only ones that not open source and free in comparision with Symbian, Android and Palm webOS. These open source platforms allow developers to access their end product’s source materials. Even if iPhone OS is not open source, this platform offers very powerfull development environment that I will use to build my application. Environment includes Mac computer is the compulsory for the iPhone app development (iMac which we are going to use), Xcode IDE, Interface Builder which is the part of Xcode, iPhone Simulator, Instrument and Shark which will allow me to assess my application performance.

3. <u>iMac</u>

As I mention at the previous section, an Intel-based Mac is compulsory in order to develop iPhone applications as the iPhone SDK requires OS X to run applications such as Xcode and the iPhone Simulator. There are some ways but not legal ways to develop iPhone applications on anything that isn't a Mac. Also those illegal solutions do not as efficient, reliable and robust as the original one. It is obvious fact that developing on iMac will provide me the huge benefits as I will learn another desktop operating system technology called Mac OS X. Great screen resolution of iMac with the built-in 21.5 inch or 27 inch LED backlit glossy wide screen will inevitably make contribution to my productivity while developing my iPhone application. iMacs have the quad-core power into the 27-inch iMac with a 2.66GHz Intel Core i5 processor or 2.8GHz Intel Core i7 processor that will enable me to perform my task with the high performance and efficiently. There are some differences but not huge differences between Microsoft OS which currently my PC operates and iMac's Mac OS X. As we are developers, this will not cause any problem when moving to any desktop platform.

4. Development Tools

4.1. Objective-C:

Objective-C is an object-oriented programming language that is used by Apple for developing applications for iPhone and Mac. It is an extension of ANSI C programming language that it actually adds the object-oriented approach on to the ANSI C language. I believe it will not be the huge problem to learn and code by using this language for the people who has object-oriented programming background. Knowledge of C programming language will also be additional advantage before moving to Objective-C since it is extension of ANSI C. It is actually simple, well-structured and powerful language. It eliminates the complexity of the some of the object-oriented approaches, different than Java programming language; there are not multiple inheritances which inherit from the same parent class. Objective-C also saves the programmer time by auto-generating the accessor (setter and getter) methods. This is one of the features that I found very handy in Objective-C. I had an experienced from my previous projects that accessors in other object-oriented languages have not got that kind of feature, so they cause to spent your precise time by copying and pasting the staff around. Objective-C automatically generated the accessor methods by looking at the “property” reserved word in Objective-C which is kind of virtual member variables that describe the state of an object. It is also important to note that after declaration of a property, you have to tell the compiler to instantiate it by using @synthesize directive.

Objective-C is well-structured since it uses two distinct classes in order to organise your code efficiently, Header files (classes with .h extension) and Implementation files (classes with .m extension). The header files contain forward declarations of your interfaces, class fields, methods, and properties. As the name implies, the implementation files contain the basic implementation code for your declared components in the header files scope.

There is one very important fundamental issue that I have to keep in mind while developing my iPhone app that Objective-C has not got the garbage collection. I have to keep an eye on my memory by releasing and retaining the objects to not to cause memory leak. It is important because Apple store rejects the iPhone apps which have memory leaks.

Objective-C uses the object reference counting approach to keep track of how much memory space in use. So, there are two crucial rules to keep my memory straight;

- I must release objects I create with alloc, new, copy or mutableCopy.
- I must consider everything else to have a retain count of 1 (creation of object with alloc (malloc in C language) increases the retain count by 1) and in the autorelease pool.

In conclusion, these are the fundamental aspects of the Objective-C language from my point of view after investigations on this object-oriented language.

4.2. Xcode:

Xcode is another necessary Apple development environment that I will be using to develop my iPhone application. Xcode IDE integrated with the other applications (tools) like Interface Builder, iPhone Simulator, and Instruments & Sharks. It is actually suite of development tools and the main application of the tools is Xcode IDE which will allow me to implement code for my iPhone app. It also supports a variety of languages beyond just Objective-C. For example, it is possible to add C or

C++ code in to your project since it is extension of ANSI C; however, it is necessary to use Objective-C programming language if you are aiming to build convenient iPhone app. So, I will keep that in my mind while developing my application.

One of the graphical applications (or tools) that I will be dealing with most of the time while developing my application is iPhone Simulator (I will talk more about it later). Once you compile your application, Xcode will install it on the iPhone Simulator (or real device); it will enable me to evaluate my application visually.

I have been experienced with some of the powerful IDEs (Integration Development Environment) on my previous projects like Eclipse and NetBeans. So, after I did essential investigations about Xcode, it will not cause any problem for me while migrating to Xcode IDE.

4.3. Interface Builder

Interface Builder is an application (or tool) of the Xcode which I will use to drag and drop any of the basic library elements (Buttons, labels, text fields, etc.) into my view, edit them, and work with the connections between the code and these elements. There is an important custom UI framework for the iPhone and iPod Touch called Cocoa Touch, all of these elements come from this custom framework.

Interface Builder has the similar functionality as NetBeans IDE’s Palett (which I’ve an experienced with) which allows you to select a component and placing it in the Design Editor. However, Interface Builder is different at the background. When you drag and drop elements from NetBeans Palett, it is automatically generates code for you into your relevant class although you have not got permission to amend these auto-generated codes. In contrast, Interface Builder is not generating any Objective-C code into your class. It is creating the files called “nib” which is an special XML files that contains description of the GUI (Graphical User Interface) you are building, and Cocoa Touch framework uses that actually create buttons, labels, text fields, etc. for your application at runtime.

One of the good features I found handy in Interface Builder, it allows you to hook up the components from your view to your code by using visual wire. It is very nice feature that makes you configure your view and controller relationship with an effective way.

4.4. iPhone Simulator

The iPhone Simulator is another useful tool of the Xcode, I will be using that while testing my application on iMac. As I mentioned before, once you compile your application, Xcode will automatically install it on the iPhone Simulator that will enable me to evaluate my application visually without need any device (iPhone or iPod Touch). One of the shortage of the iPhone Simulator; it does not come with all of the applications that a real phone does; however, for the most part it behaves the same way. So, it will be enough to test my application for the most of the part.

There is another important drawback, I have to take into consideration that memory, performance, camera, GPS, and other characteristics (Shaking, rotating, etc.) cannot be reliably tested using the Simulator. iMac has so many resources in comparison with iPhone. Thus, I need to install my application on to one of the device (iPhone or iPod Touch), depends on characteristics I am using in order to test my application fully. This means joining one of the paid development programs to install application on an actual device.

4.5. Instruments and Shark

Instruments and Shark are also applications (or tools) of the Xcode that will help me to control my application behaviour and performance. Instruments are for analysing the performance of iPhone application. It collects and visually output the data like memory retains and releases which will cause memory leaks, CPU activities, user events, and etc. I will use Instruments mainly to control the memory leaks which are the most crucial point for my application since iPhone has limited memory. Memory should not be wasted that will also cause the application to be rejected by Apple store, if you attempt to publish it.

I will also use Shark to profile whole system and checking my application code performance in order to make it as efficient as possible. Shark specifically indicates the block of code or the system which cause the performance weaknesses. Nowadays, code efficiency is one of the most important thing which has directs affect of quality of the final product. Thus, Shark will be useful tool due to the fact that it will increase quality of your application.

In conclusion, I will be using both Instruments and Sharks to improve my application performance, behaviour and most essentially to keep my memory straight.

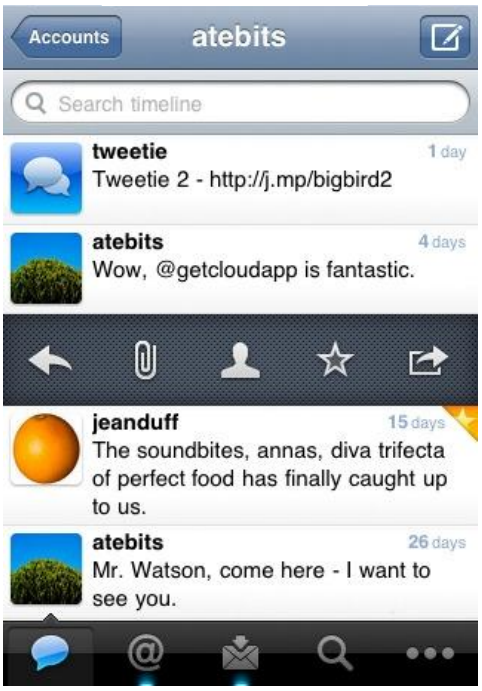
5. <u>Categories of iPhone Apps</u>

In spite of the fact that I could not find the chance to install and play with the following iPhone apps on the actual device (iPhone or iPod Touch), as best I can I will try to highlight some of the features I like and possibly take an advantage of these features while building my application.

5.1. Social Networking

Tweetie 2

Figure-1



tweetie 1 day
Tweetie 2 - http://j.mp/bigbird2

atebits 4 days
Wow, @getcloudapp is fantastic.

jeanduff 15 days
The soundbites, annas, diva trifecta of perfect food has finally caught up to us.

atebits 26 days
Mr. Watson, come here - I want to see you.

Nowadays, Tweetie 2 is one of the most popular twitter clients currently available and widely used by the iPhone users which was also awarded for 2009 Apple Design prize.

Firstly, design of the application seems really aesthetic, well-structured and easy to learn for the users. I believe that If you aim to create widely used application, the first thing you should consider is, how handy to use your application for the user. Users are generally showing a tendency to straight-forward applications that they can quickly get the usage idea and perform their tasks efficiently. Thus, I will try to build my application in this manner.

Tweetie 2 is persistent that when the application shut down by the interruption of other application (telephone rings, message comes) or you close the application manually, it remembers the last time you were browsing and bring you back when you start the application again. For example, you were on a user’s Twitter profile and you exit from the application. You start the application and you find yourself on a user’s Twitter profile, where you were looking last time. I can use this feature while building my application. For instance, the user is dealing with calculator and entering some figures in order to calculate his/her budget. Suddenly, the user recognizes that he/she has to call somewhere before closed. So, the user shuts down the application immediately, and calls the necessary place. When the user comes back the calculation, he/she will see that the figures (Ex: numbers) he entered the last time still there. What a handy feature.

There is a draft manager that you can save your tweets to send at a later time. I believe it can be beneficial for the people who want to tweet something after specific amount of time and do not want to forget he/she will tweet.

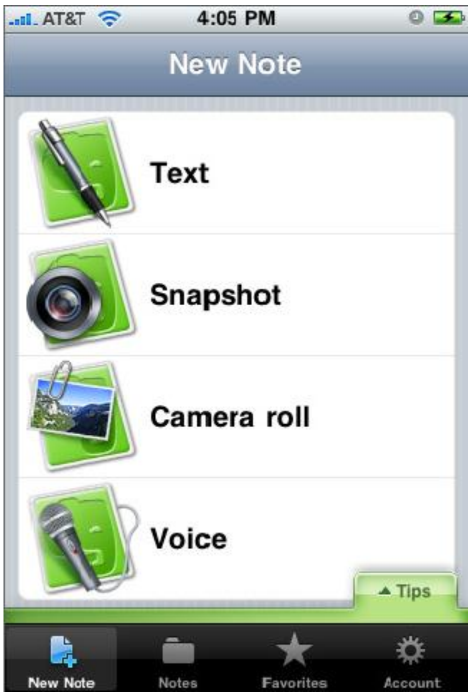
One of the features I found very creative is; when you swipe a tweet to the right, it brings you set of options that you can do with this tweet like add tweet into your favourite tweets, go to person profile, retweet, translate tweet, post link to tweet, quote tweet and mail tweet.

Another creative idea I found at Tweetie 2; there is not additional reload button for refreshing your tweets stream rather than this you just scroll up, and hold for a second, it will reload the newly added tweets if they are exist.[1] So, possibly I should add some innovative futures that will make my application unique.

5.2. Productivity

Evernote

Figure-2



Evernote is enables you to create notes, get the snapshot of photos, save photo notes and record voice memos which you can access them from your iPhone, computer or the internet.

I can describe this application as it is kind of your second memory that collecting and organizing information about your real life. Imagine you have camera device and you are taking photos of images; however, camera device have not got any features to add any notes on a photo while it is possible with Evernote iPhone app. This is one of the nice features that Evernote brings you.

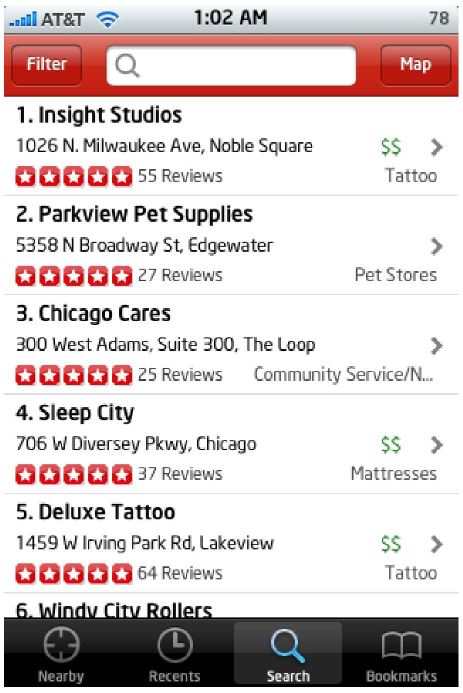
One of the incredibly good features I should highlight is with a character recognition technology it recognizes the text in snapshots and makes the text searchable by the user. It is awesome feature although after investigation about Evernote I found that character recognition from snapshots does not work for every surface like wine labels while it is working well for book snapshot.[2] If you type the text (or subtext) in order to search, Everynote will show you the corresponding snapshot and the attached note on it.

There is “Favorites” option that allows the user to save his/her favorites as the name implies. This is one of the common and nice features that most of the iPhone apps have. So, some sort of save feature that I would like to use while building my application.

5.3. Utility

Yelp

Figure-3



AT&T 1:02 AM 78

Filter Map

1. Insight Studios
1026 N. Milwaukee Ave, Noble Square $$
55 Reviews Tattoo

2. Parkview Pet Supplies
5358 N Broadway St, Edgewater
27 Reviews Pet Stores

3. Chicago Cares
300 West Adams, Suite 300, The Loop
25 Reviews Community Service/N...

4. Sleep City
706 W Diversey Pkwy, Chicago $$
37 Reviews Mattresses

5. Deluxe Tattoo
1459 W Irving Park Rd, Lakeview $$
64 Reviews Tattoo

6. Windy City Rollers

Nearby Recents Search Bookmarks

Yelp is an application for searching places to eat, drink, shop and play. Once you have searched the place to go, application allows you to review the people's comments about these places.

One of the features I found really nice in Yelp is It is possible for you to find businesses near you by using iPhone GPS technology (from Nearby Tab). After Yelp have found the appropriate businesses near your location, it allows you to filter them by location, distance, price, type, and etc.[3] Choosing the place gives some necessary informations like opening hours, reviews about place, address, phone number and prices.

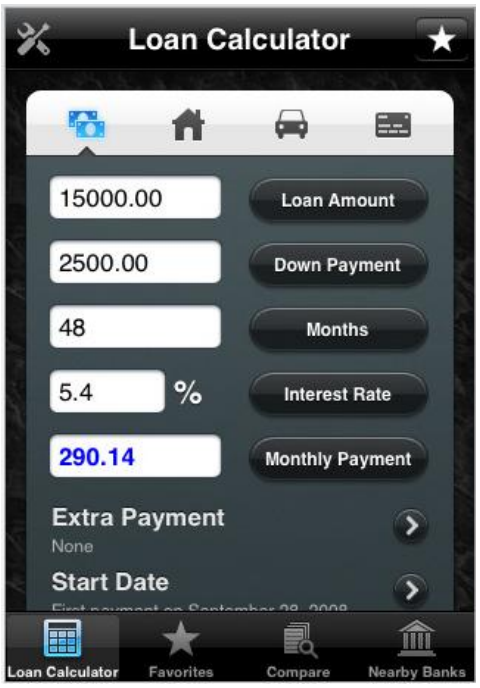
If I buy an iPhone, this will be the first application that I will download. The first reason, Yelp is free application that you can directly download from iTunes. It is very useful application since it eliminates the where to go problem. For example, you are in another location that you are not familiar with and you want to eat or drink something with friends. So, it is the best time to use Yelp to figure out the appropriate places to go.

The application also has a good design pattern that it seems to conform to the Apple's iPhone Human Interface Guidelines (iPhone HIG) for productivity type. If your application does not conforms this guideline for specific type (productivity, utility, immersive) then it will be rejected by Apple store after review process.

5.4. Finance

Loan Shark

Figure-4



Loan Shark is an application that integrates loan calculator with the iPhone. It allows you to calculate cash, home, car, and credit card loans.

Once you have opened the Loan Shark, you can start loan calculation immediately. You just need to field the required fields then the application will automatically calculate the loan for you without needed interaction with any button (there is not button like “Calculate”).[4] So, this is one of the nice features that I will probably use while building my application.

Loan Shark looks more than the calculator to me. Application allows you to compare the loans side-by-side, using the iPhone’s built-in location sensor Loan Shark is listing the banks near your location, showing you full amortization table, etc.[5]

Loan Shark’s one of the handiest feature is it has “Favorites” tab like most of the iPhone apps. You can save loan scenarios after calculation for future use. As I mentioned before, this is the tab that I want to use in my application.

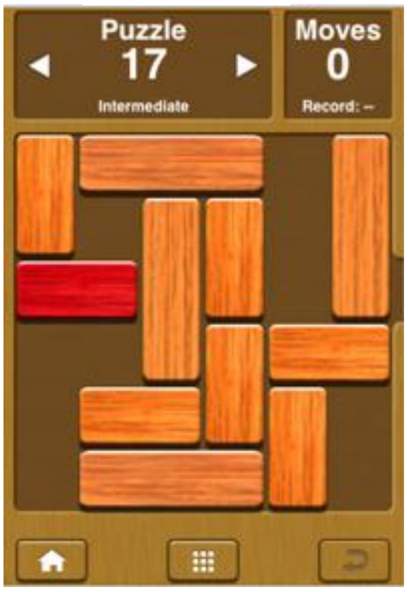
Loan Shark allows using your local currency by setting from iPhone general preferences. I have to keep in my mind that some of the features such this should be the configurable always from the iPhone general preferences rather from my application itself.

Loan Shark has also great layout that I will definitely take as an example while designing my application.

5.5. Game

Unblock Me

Figure-5



Unblock Me is a iPhone game app that the purpose of the game is trying to get the red block out from an area by sliding the other blocks elsewhere from the red block.

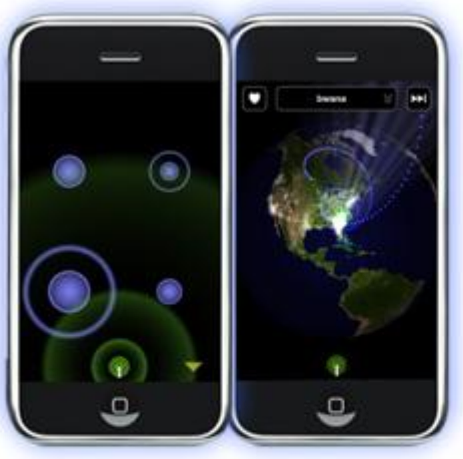
UnBlock Me seems very straight-forward game which likely makes you addictive after couple of plays. I put this application as an example because I believe this type of games fit into small devices very good rather than the first person shooter games such as Doom. In my opinion, it is an obvious fact that the game players prefer and enjoy more while playing the first person shooter games, or car games with the computer device which has widescreen and high resolution instead of tiny screen.

Unblock Me has very aesthetic GUI (Graphical User Interface) that I like. Colour harmony and the wood looking blocks are pretty cool.

5.6. Music

Figure-6

Ocarina



Ocarina is a very impressive iPhone app that it replaces the musical instrument with a mobile device. You just need to blow your microphone to play music. Besides this, Ocarina is sensitive to touch and movements. Ocarina claims that these sensitivities making it more versatile than the original musical instrument.[6]

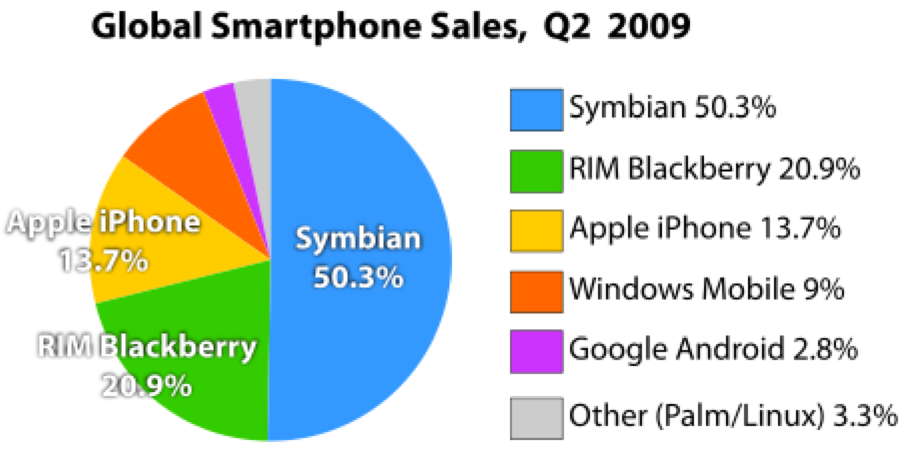
In my opinion, Ocarina is a great example of how it is possible to build an application which integrates between the iPhone technologies with the application. There are several technologies that iPhone has such as accelerometer, GPS (built-in location sensor), multi-touch screen and etc. While developing an iPhone app, the developers should take into account existence of these technologies and try to take an advantage of them if they want to create challenging iPhone app.

Appendix

Appendix 1 “The Mobile Phone as a Development Platform”

At the first glance, mobile phone is a small, portable communication device that enables people to make phone calls.[7] Nowadays, it is more than that. The mobile phone as a development platform since the first mobile devices (or handled devices) of the 1980s, the popularity of these platforms has risen substantially. Many mobile phone models after the late 2000s gain the capability to run user-installed application (or software) and mobile phones are getting more and more sophisticated everyday.

Today’s desktop application developers have it easy. Desktop application developers essentially have three OS platforms to choose from: Windows, Linux and Mac OS X. There are also often ways to make software written for one platform run on the others (Platform Independence). Even if the mobile developers limit the choices to smartphone platforms alone, mobile developers must choose between iPhone, Android, BlackBerry, Palm webOS, Symbian, and Windows Mobile. Each platform is incompatible with the others (Platform Dependence). In turn, the choice of platform limits the choice of tools and languages that are available and each mobile application can only run on the device depends on which platform the application developed in. This means, iPhone application can only run on the iPhone but not on the Blackberry.[8]



http://en.wikipedia.org/wiki/File:Smartphone_2009.svg

Above diagram does not include Palm WebOS, which was introduced in June, 2009.

Mobile applications are not just ported small desktop applications.

- **Mobile phones have a small screen and are task-focused**

  Mobile phones have a relatively small screen. You need to put real thought into every screen and keep it focused on the specific task the user is doing.

- **Mobile phones have limited CPU and Memory**

  There is no virtual memory and every bit of CPU power you use means more battery drain.

- **Only one application can run at a time**

  If anything else happens, like the phone rings, a text message comes in, the user clicks on a link, etc., your app gets shut down and the user moves on to another application. You need to able to gracefully exit at any time and be able to put users back into a reasonable spot when they return.[9]

These are the factors that the mobile developers need to consider when they are working on a mobile phone as a development platform while developing mobile applications, in general.

Now, we will mention about our main platform called iPhone and we will also mention some of the alternative platforms for mobile development.

1.1. iPhone

Figure-7

1.1.1. Brief History:

iPhone (Figure-7) is a smartphone that development started with the direction of the Apple CEO Steve Jobs to the Apple engineers to investigate touch screens. During that time, he was considering whether or not working on tablet PCs. However, he made his decision in order to develop mobile phone rather than tablet PC. He states that traditional PDAs and tablet PCs did not really seem like feasible and high-demand product industries that Apple could enter. He believed that mobile phones are essential devices for information access to go. So, Apple was announced the development of the smartphone called iPhone and its features and applications during the Macworld convention on 9th of January 2007. Then the first generation iPhone became available to purchase at Apple stores and AT&T Mobility on 29th of June 2007 in the United States before being marketed worldwide.[10]

1.1.2. Many devices in one:

It is smart since iPhone is more than just a phone. It combines many devices in one;

Phone

It is possible to make a call by tapping a name or number in your contacts, your call log or just about anywhere.

iPod

iPhone enables you to listen music and watch a video. It displays your music, movies, TV shows, and more on a 3.5 inch display screen. You can enrich your collection by downloading music and video wirelessly from the iTunes Store.

Internet Device

iPhone uses 3G and Wi-Fi wireless connections to connect you internet. It delivers rich HTML email, Maps with GPS and Safari web browser. It also provides you two well-known search engines Google and Yahoo. iPhone is multi-threaded (multi-tasks) that it allows you to make a phone call while surfing on the web.[11]

Camera

You can also take pictures and video (using an iPhone 3GS) with a built-in camera. View them on iPhone, email them, send them in an MMS message (iPhone 3G or later), or upload them to your computer. Trim and save video clips. Upload videos directly to YouTube. Take a friend’s picture and set iPhone to display it when that person calls you.[12]

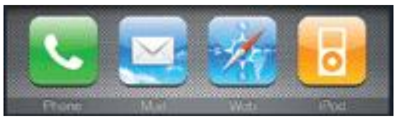
1.1.3. iPhone Overview:

One of the most significant features of the iPhone, functionalities bases on the user multi-touch events. iPhone hardware interface lack of physical keyboard or a mouse that the multi-touch screen renders a virtual keyboard when necessary. iPhone uses virtual buttons and controls that appear on its screen and requires you to use your fingers different than other smartphones (PDA or a Nintendo DS) which requires you to use a slender, pointed stylus.

The front surface of the Apple iPhone has only one button (Home button).

When you press the Home button, iPhone brings you to the main screen of the iPhone's graphical user interface. (Figure-1 above on page 4)

There, you can choose from the device's four primary functions using icons at the bottom of the phone.



Phone brings you 3G, GSM or EDGE cellular phone service as well as a visual voice mail menu.

Mail allows you to POP and IMAP email access, including HTML capabilities, in-line pictures and push email from Google or Yahoo mail.

Web opens the Safari web browser to surf on the internet and more.

iPod turns into media player to listen Music and watch a video.

Upper portion of the Home screen makes available to the user other iPhone applications. (Figure-8)

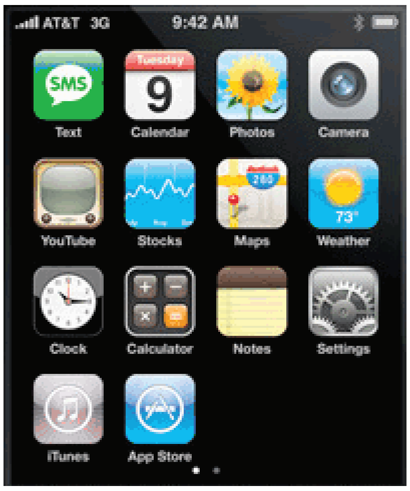


Figure-8

There are some built-in applications are coming with the iPhone. These include a calculator, calendar, widgets, and notepad. You can also purchase or freely download applications which made specifically for the iPhone from the iTunes like apps for fun and games, apps for social networking, apps for sports, apps for productivity and so on. (More about downloadable iPhone applications later on this paper) [13]

So, it is possible to have multiple home screens, if you have many applications. You can arrange them over multiple Home screens and switch to another home screen by flick left or right, or tab to the left or right of the row of dots.[14]

Other than the built-in applications we did mention above, there are also some built-in ones come with iPhone 3G S (Latest generation iPhone currently available):

You can get current weather conditions and a six-day forecast with Weather app.

You can watch your favourite stocks which updated automatically from the internet with Stocks app.

You can see a street map, satellite view, or hybrid view of locations around the world with Maps app.

Even though the iPhone does not support flash, you can play videos from YouTube's online collection with YouTube app.

You can use digital compass to determine your heading by getting current coordinates and choosing between true north and magnetic north with Compass app. [15]

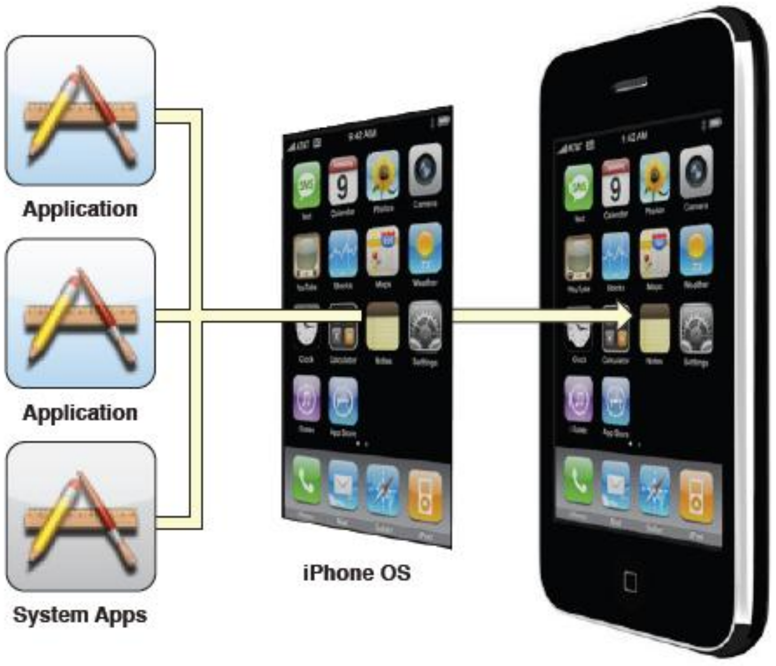
1.1.4. iPhone OS Technology Overview:

iPhone OS is the operating system that runs on iPhone and iPod touch devices. This operating system manages the device hardware and also provides the basic technologies needed to implement native applications on the phone. Depending on whether it is installed on an iPhone or an iPod touch, the operating system also ships with several system applications, such as Phone, Mail, and Safari that provide standard system services for the user.

iPhone OS Architecture:

iPhone OS Architecture is similar to the basic architecture of Mac OS X. At the high level, iPhone OS acts as an intermediary between the iPhone and iPod touch hardware and the applications that appear on the screen (as shown in Figure-9 below). Applications that you create never interact directly with the hardware. Instead of that, go through system interfaces, which interact with the appropriate drivers. This abstraction protects your application from changes to the underlying hardware.

Figure-9 Applications layered on top of iPhone OS

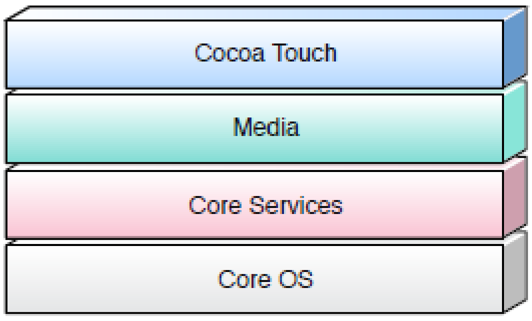


iPhone OS uses fairly straightforward stack. The Mach kernel and hardware drivers locate at the very bottom of this stack that manage the overall execution of programs on the device. There are additional layers on top of that layers, core technologies and interfaces that developers use for development.

iPhone OS Technologies:

iPhone OS technologies consists of a set of layers (as shown in Figure-10 below). When we look the lower-level layers these are the fundamental services on which all applications rely on and the higher-level layers contain more sophisticated services and technologies.

Figure-10 Layers of iPhone OS



Developers should prefer to use of higher-level frameworks over lower-level frameworks when writing code for their applications. The higher-level frameworks exist to provide object-oriented abstractions for the lower-level frameworks. These abstractions generally make much easier to write code since they reduce the number of lines of code you have to write and encapsulate potentially complex features, such as sockets and threads. If the developers want to use the lower-level frameworks, they are still available for developers who want to use aspects of those frameworks that are not exposed at the higher level.

Cocoa Touch Layer

Cocoa Touch is one of the most important layers for iPhone OS. It contains the crucial frameworks, which provide the infrastructure you need to implement applications in iPhone OS. These are the first frameworks that developers should always start with and drop down to lower-level frameworks only when needed.

UIKit Framework is very important.

The UIKit framework (UIKit.framework) contains Objective-C programming interfaces that provide the key infrastructure for implementing graphical, event-driven applications in iPhone OS. Every application in iPhone OS uses this framework to implement its core set of features such as:

- Application management
- Cut, copy, and paste support
- Graphics and windowing support
- Support for handling touch and motion-based events
- User interface management
- Objects representing the standard system views and controls
- Support for text and web content
- Integration with other applications on the system through URL schemes
- Support for the Apple push notification service
- Accessibility support for disabled users

In addition to providing the fundamental code for building your application, UIKit also incorporates support for some device-specific features, such as the following:

- Accelerometer data
- The built-in camera (where present)
- The user’s photo library
- Device name and model information
- Battery state information
- Proximity sensor information

Media Layer

In the Media layer are the graphics, audio, and video technologies geared toward creating the multimedia experience available on a mobile device. These technologies were designed to make it easy for you to build applications that look and sound great. The high-level frameworks in iPhone OS make it easy to create advanced graphics and animations quickly and the low-level frameworks provide you with access to the tools you need to do things. Media layer consists of:

<u>Graphics Technologies:</u> core graphics framework, Quartz core framework, OpenGL ES framework.
<u>Audio Technologies:</u> AV foundation framework, Core Audio family of frameworks, OpenAL.
<u>Video Technologies:</u> iPhone OS provides support for full-screen video playback through the Media Player framework. This framework supports the playback of movie files with the .mov, .mp4, .m4v, and .3gp filename extensions.

Core Services Layer

This layer provides the fundamental system services that all applications use. Many parts of the system are built on top of these fundamental services, although developers use these services directly. Fundamental services such as; Address Book framework, Core Data framework, Core Foundation framework, Core Location framework, Foundation Framework, Store Kit framework, SQLite library and XML support.

Core OS Layer

This layer is the very core layer that consists of CFNetwork framework, which is a set of high performance, C based interfaces that provide object-oriented abstractions for working with network protocols; external accessory framework, which provides support for communicating with hardware accessories attached to an iPhone or iPod touch device; security framework, which are built-in security features that developers use to guarantee the security of the data your application manages; and the system level, which encompasses the kernel environment, drivers, and low-level UNIX interfaces of the operating system.[16]

1.2. Alternative Platforms

1.2.1. Android:

Android is a mobile operating system that running on the Linux kernel. It was initially developed by Android Inc. Then, Google purchased a Android Inc which was a little-known company, co-founded by Andy Rubin, now director of mobile platforms at Google. Lately purchased by the Open Handset Alliance, open mobile platform (Android) development organization. [17] [18]

Figure-11

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. Android is an alternative mobile development platform that Android SDK enables the developers to begin developing applications by providing the tools and APIs necessary using the Java programming language.[19]

Android is open source that it can be liberally extended to incorporate new cutting edge technologies as they emerge. Android mobile platform will maintain to evolve as the developer community works together to build reformer mobile applications.[20]

Features

- Application Framework enables to reuse and replacement of components.
- Integrated browser available in Android is based on the open-source WebKit application framework.
- SQLite database technology is used for data storage.
- Optimized graphics powered by a custom 2D graphics library, 3G graphics based on the OpenGL ES 1.0 specifications, and traditional smartphone layouts.
- Dalvik virtual machine optimised for mobile devices.

- Media support for common audio, video, and still image formats like MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF.
- GSM Telephony hardware dependent.
- Connectivity technologies are Bluetooth, EDGE, 3G, and WiFi.
- Additional hardware support with Camera, GPS, compass, and accelerometer.
- Development environment including device emulator, tools for debugging, memory and performance profiling, and a plugin for the Eclipse IDE.

Android Architecture

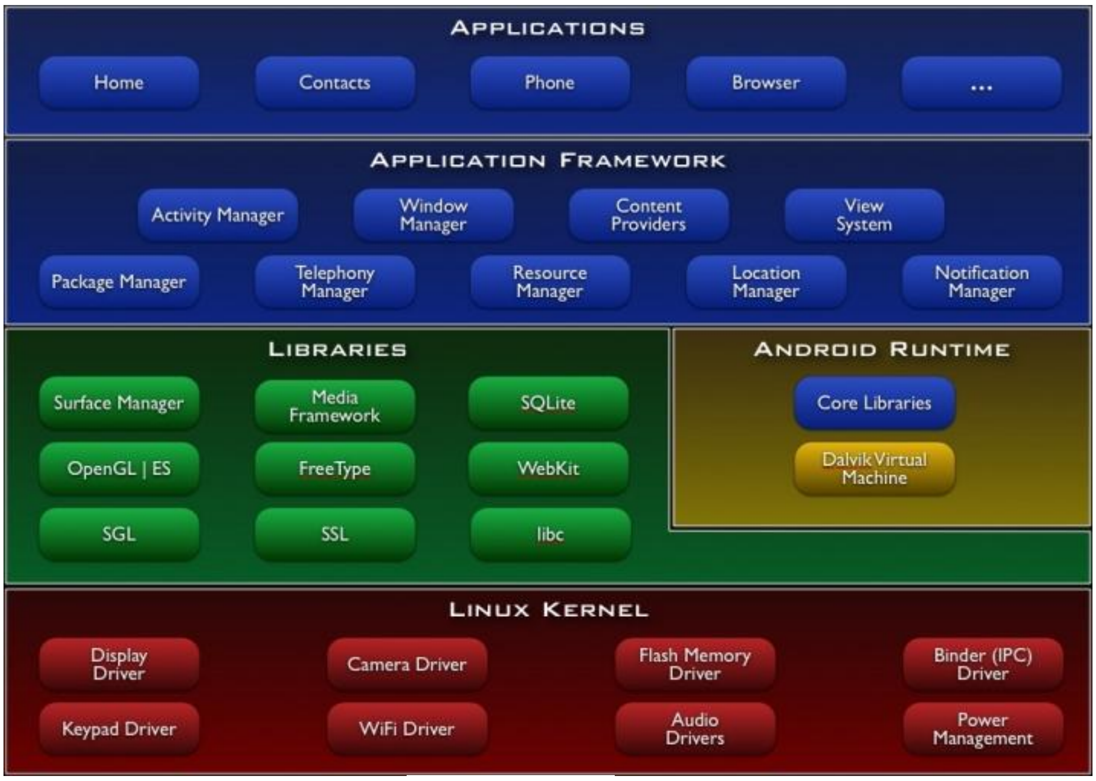


Figure-12

The diagram above visualises the major layers of Android OS.

Applications

Android comes with the set of built-in applications including SMS program, maps, calendar, email client, browser, contacts, and many others. Those built-in applications are written with the Java language.

Application Framework

Android developers allow to access the same framework APIs used by the core applications. The architecture of the application simplifies the components reusability. For instance, any application can publish its capabilities and any other application may then make use of those capabilities. This same mechanism enables components to be replaced by the user.

Libraries

Android includes a set of C and C++ libraries and those libraries used by miscellaneous components of the Android platform. Developers are able to access these capabilities through the Android application framework.

Android Runtime

Android includes a set of core libraries that makes available to developers most of the functionality in the core libraries of the Java language. The Dalvik Virtual Machine depends on Linux kernel that underlying functionality such as low-level memory management and threading.

Linux Kernel

Android uses the Linux version 2.6 for core system services such as process management, memory management, security, network stack, and driver model. The kernel also plays role as an abstraction layer between the harware and the rest of the software stack. [21]

1.2.2. BlackBerry:

Figure-13

The BlackBerry is a smartphone that is widely used in the enterprise for its wireless email handling capability. The device is often referred to as the CrackBerry because of its compelling nature. It is an example of convergent device that developed by the Canadian company Research In Motion (RIM).[22] It consists of smartphones integrated with software that enables access to a variety of data and communication services. The BlackBerry solution supports leading enterprise email platforms and can be customized to meet the needs of one person or an entire enterprise, to any scale, in any industry.[23] The BlackBerry smartphone is a pure Java device with all built-in applications and APIs written in Java.

RIM provides a personal multi-tasking operating system for the BlackBerry, which makes heavy use of the device's specialized input devices. The operating system provides support for Java MIDP 1.0 and WAP 1.2. The current OS 4 provides a subset of MIDP 2.0, and allows complete wireless activation and synchronization with Exchange's e-mail, calendar, tasks, notes and contacts. It also supports Lotus Notes and Novell GroupWise.

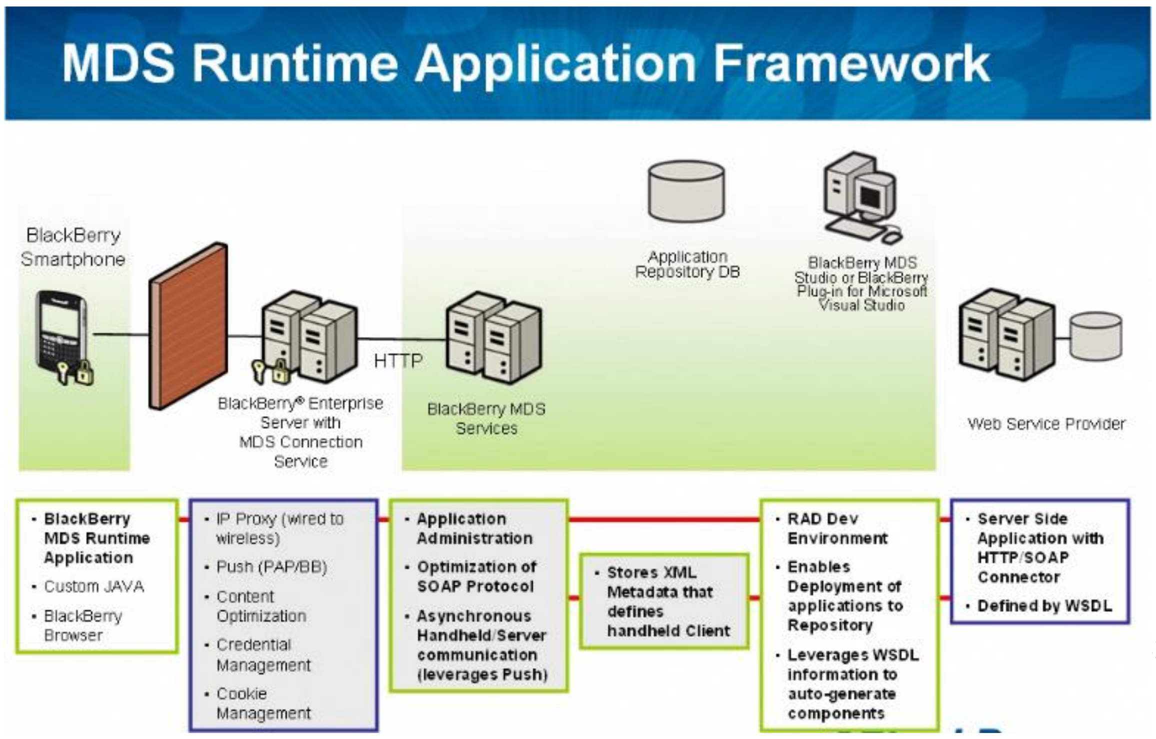
The latest BlackBerry 9000 series use Intel Xscale 624 MHz CPU, which makes it the fastest BlackBerry to date. [24]

Features

- Data extracted from a BlackBerry to a host computer is stored in a single file in a BlackBerry specific format known as IPD. [25]
- Leverages the core strengths of BlackBerry to mobilize additional business processes and applications; Security & Encryption, managed connectivity & push, centralized management, designed for wireless.

- From an application perspective, BlackBerry MDS provides end-to-end security, managed wireless connectivity, standard protocol interfaces and device & network independent push.
- Blackberry smartphones support MIDP Standard APIs for cross platform development, CLDC Standard APIs for connectivity and BleckBerry specific Java APIs.
- Development environment including device simulators, tools for debugging, themes and animated graphics, issue tracker, and BlackBerry Java Development Environment Plugin for Eclipse or BlackBerry JDE.
- Two-way email integration with Outlook, Notes and GroupWise, respectively and instant messaging.
- Go beyond communications with a built-in camera, multi-media and social networking capabilities.
- Instant information with technologies BlackBerry browser, GPS, Organizer, BlackBerry Maps, Tethered Modem.
- Entertainment environment with Media Player, BlackBerry Media Sync and Camera & Video Recording.[26]

Figure-14



MDS Runtime Application Framework
BlackBerry Smartphone
BlackBerry® Enterprise Server with MDS Connection Service
HTTP
BlackBerry MDS Services
Application Repository DB
BlackBerry MDS Studio or BlackBerry Plug-in for Microsoft Visual Studio
Web Service Provider
• BlackBerry MDS Runtime Application • Custom JAVA • BlackBerry Browser
• IP Proxy (wired to wireless) • Push (PAP/BB) • Content Optimization • Credential Management • Cookie Management
• Application Administration • Optimization of SOAP Protocol • Asynchronous Handheld/Server communication (leverages Push)
• Stores XML Metadata that defines handheld Client
• RAD Dev Environment • Enables Deployment of applications to Repository • Leverages WSDL information to auto-generate components
• Server Side Application with HTTP/SOAP Connector • Defined by WSDL

1.2.3. Palm webOS:

Palm webOS is a mobile operating system running on the Linux kernel with proprietary components developed by Palm. It was released on June 6, 2009 that was introduced to the public at the Consumer Electronics Show in Las Vegas.[27] It is Palm's next generation operating system that designed around a fast and beautiful user experience and optimized for the multi-tasking user. Palm webOS integrates the power of a window-based OS with the simplicity of a browser. Applications are built using standard web technologies and languages, but have access to device-based services and data.[28]

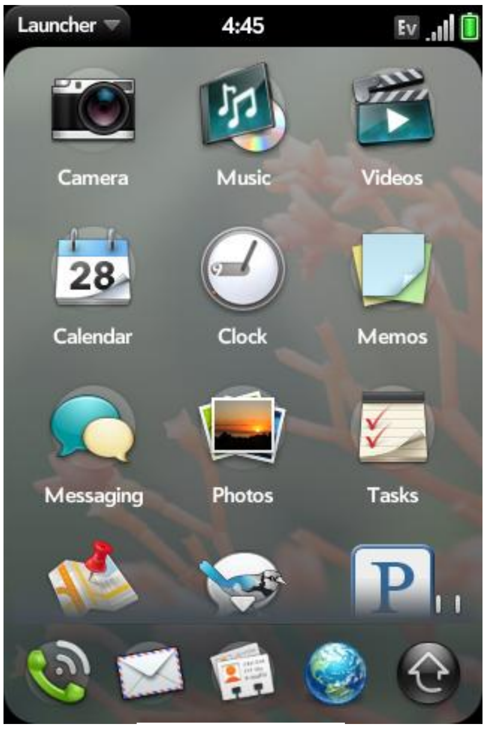


Figure-15

Features

- Palm webOS is designed to run on a variety of hardware with different screen sizes, resolutions and orientations, with or without keyboards and works best with a touchpanel though doesn't require one.
- It includes set of applications for personal information management and you can think of webOS applications as native applications, but build from the same standard HTML, CSS, and Javascript that you would use to develop web applications.
- The user experience is optimized for launching and managing multiple applications at once.
- Palm uses a webOS software development kit called Mojo, development environment including Palm emulator, tools for debugging, command-line tools, and Palm Plugin for Eclipse IDE.

- Palm Services automatically backs up your phone so you have a copy of your important data. You can also remotely erase your data if your phone is lost or stolen.
- Additional hardware support with Camera and GPS (built-in location). [29] [30]

Palm WebOS Architecture:

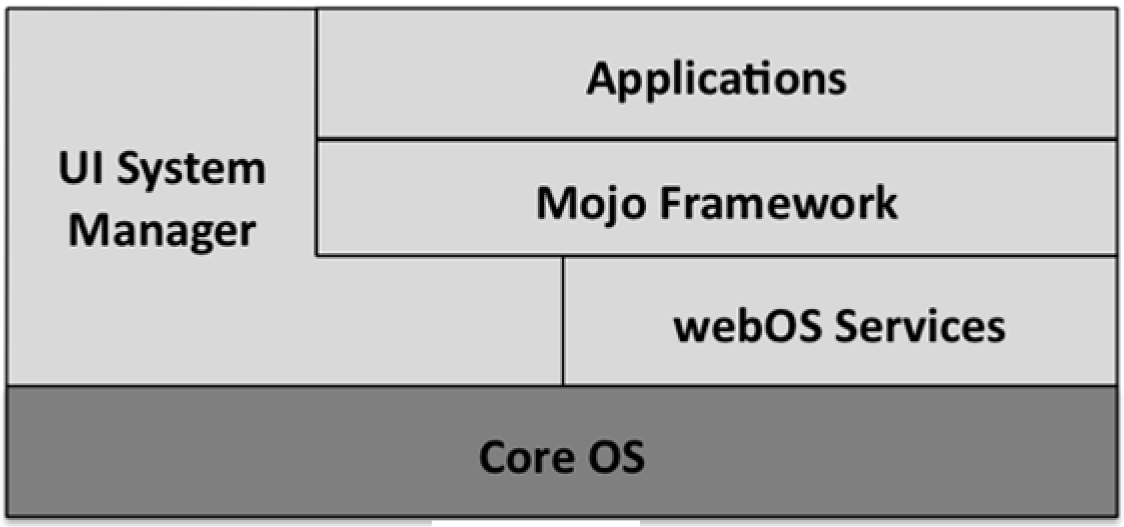


Figure-16

The Palm webOS platform is based on the Linux 2.6 kernel, with a combination of open source and Palm components providing user space services, referred to as the Core OS.

You have not got any direct interaction with the Core OS and the end users. Instead of that, your interaction is through Mojo and the various services. Users interact with the various applications and the UI System Manager, which is responsible for the System UI.

Application Environment:

The application runtime environment is managed by the UI system manager that also presents the System UI manipulated by the user. The framework provides access to the Palm Services and the UI Widgets. Supporting this environment is the Core OS environment, an embedded Linux OS with some custom sub-systems handling telephony, touch and keyboard input, power management, storage and audio routing. All these Core OS capabilities are managed by the Application Environment and exposed to the end user as System UI and to the developer through Mojo APIs.[31]

1.2.4. Symbian OS:

Symbian is an operating system designed for mobile devices and smart phones developed by Symbian Ltd. It includes associated libraries, user interface, frameworks, and reference implementation of common tools.[32] When Symbian OS was initially released, it was known as EPOC and it is still used to refer to the kernel.[33]

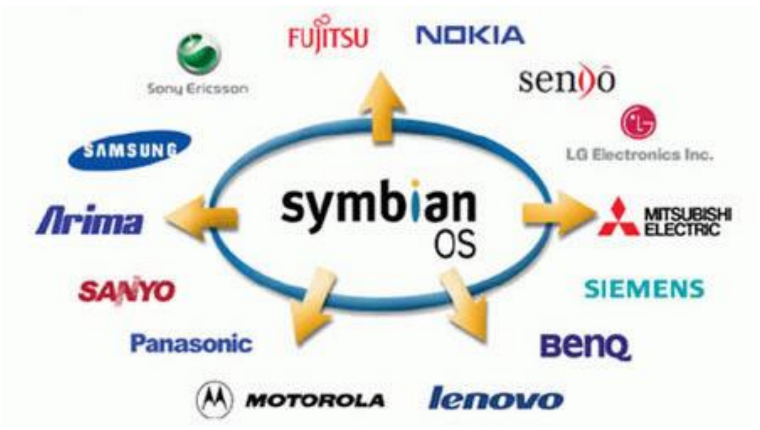


Figure-17

Symbian operating system was build with three system design principles;

- User time is precious and must not be wasted,
- All resources are limited,
- The integrity and security of user data is paramount. [34]

Symbian OS has very small memory footprint and low power consumption. So, it is important for the users who do not want to recharge their phone daily and to allow running on small devices with limited memory. It was originally closed source but it moved to open source in order to enable third party developers to build and install applications independently from the device manufacturers. [35]

Device manufacturers which currently use the Symbian OS are;

Ericsson, Fujitsu, Mitsubishi, Motorola, Nokia, Samsung, Sendo, Sharp, Siemens, and Sony Ericsson.[36]

Features

- Symbian is the world's most popular mobile operating system, accounting for 50% of smartphone sales.
- Symbian OS can run on more than one hardware platform, so it can be used on a variety of device types including those touch screens and those with pens or keyboards.
- Some devices that run Symbian OS may not be switched off for years, therefore the OS was designed so applications could run for years without loosing the user data.
- Symbian OS is highly optimized, heavily asynchronous, pre-emptive, multitasking operating system.
- Symbian OS supports the client-server architecture. OS provides a class framework and a suite of organising and communicating applications.
- All the system services are directly running from ROM. The user applications reside in RAM. Multi-threading and context switch are not encouraged at the user application level.
- Symbian OS can be run on devices with different screen sizes (Screen size independent). It is also user input (touchscreen, pen, keypad, etc.) independent. [37] [38]

Symbian OS Architecture

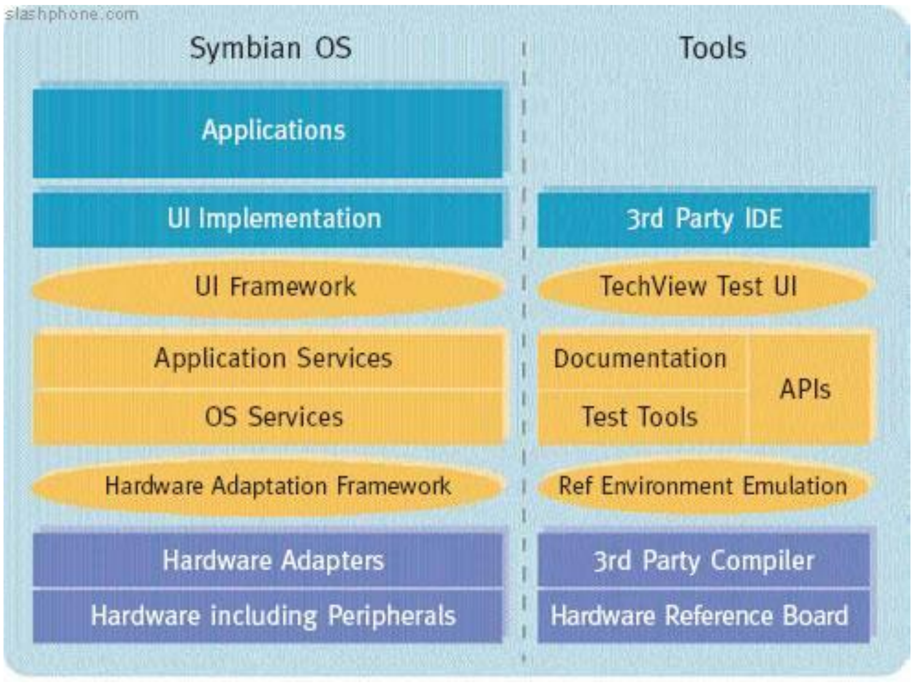


Figure-18

The diagram above visualises the major layers of Symbian OS.

UI Framework Layer

It is the topmost layer of Symbian OS, the UI Framework layer provides the frameworks and libraries for constructing a user interface, including the basic class hierarchies for user interface controls and other frameworks and utilities used by user interface components.

Application Services Layer

The Application Services layer provides support independent of the user interface for applications on Symbian OS.

These services divide into three broad groupings:

- System-level services used by all applications, for example the Application Architecture or Text Handling.

- Services that support generic types of application and application-like services, for example personal productivity applications (vCard and vCal, Alarm Server) and data synchronization services (OMA Data Sync, for example); also included are a number of key application engines which are used and extended by licensees (Calendar and Agenda Model), as well as legacy engines which licensees may choose to retain (Data Engine).

- Services based on more generic but application-centric technologies, for example mail, messaging and browsing (Messaging Store, MIME Recognition Framework, HTTP Transport Framework).

OS Services Layer

The OS Services layer is the middleware layer of Symbian OS that provides the servers, frameworks, and libraries that extend the bare system below it into a complete operating system.

The services are divided into four major blocks, by broad functional area:

- Generic operating system services.
- Communications services.
- Multimedia and graphics services.
- Connectivity services.

The Base Services Layer

The foundational layer of Symbian OS, the Base Services layer provides the lowest level of user-side services. In particular, the Base Services layer includes the File Server and the User Library. The microkernel architecture of Symbian OS places them outside the kernel in user space. (This is in contrast to monolithic system architectures, such as both Linux and Microsoft Windows, in which file system services and User Library equivalents are provided as kernel services.)

Kernel Services and Hardware Interface Layer

The lowest layer of Symbian OS, the Kernel Services and Hardware Interface layer contains the operating system kernel itself, and the supporting components which abstract the interfaces to the underlying hardware, including logical and physical device drivers and variant support which implements pre-packaged support for the standard, supported platforms (including the Emulator and reference hardware boards). [39]

Appendix 2 "iMac"

Figure-19

The iMac (Figure-18) is a range of all-in-one Macintosh desktop computers designed and built by Apple Inc. It has been the primary part of Apple's consumer desktop offerings since its introduction in 1998, and has evolved through four distinct forms. In its original form, the iMac G3, the iMac was gum drop- or egg-shaped with a CRT monitor, mainly enclosed by colored, translucent plastic. The second major revision, the iMac G4, moved to a design of a hemispherical base containing all the main components and an LCD monitor on a freely moving arm attached to the top of the base. The iMac G5 and the Intel iMac placed all the components immediately behind the display, creating a chunky unified design that tilts only up and down on a simple metal base. The current iMac shares the same form as the previous models, but is now thinner and uses anodized aluminium and black-bordered glass for its case.

Apple updated iMac on March 3, 2009 with new nVidia chipsets, the new Mini-DisplayPort and a new keyboard without any numeric pad that has become standard in all new Apple computers.[40]

Apple made further update to the iMac on October 20, 2009 with two new LED-backlit 16:9 widescreen sizes in 21.5” and 27” models. These new models replace with the 20” and 24” 16:10 aspect ratio screens of the previous generation. In addition, Apple is increased the memory and hard drive capacity of the iMac. The 21.5-inch and 27-inch iMac models offer fast Intel Core 2 Duo processors up to 3.33GHz. Quad-core power comes to the 27-inch iMac with a 2.66GHz Intel Core i5 processor or 2.8GHz Intel Core i7 processor. Four cores deliver up to 2x faster performance for just about everything you do: managing your photos, editing HD video, even playing graphics-intensive 3D games. [41]

For your daily task.

A Mac comes with all the applications you need to send email, create calendars, and more. It is all part of the Mac operating system: Mac OS X Snow Leopard.

Hardware and software made to work together.

Apple makes the Mac, the software that comes with it, and the operating system that powers it. That makes features like the wireless Magic Mouse possible. Snow Leopard enables Multi-Touch gestures, so you can scroll with one finger or swipe with two.

Works with windows.

The Mac works with Microsoft Office applications such as Word and Excel, and even runs Windows, so it’s compatible with the PC world. That includes sharing documents with people who use PCs, printing to just about any printer, and working over Windows networks.

Lack of (minimal) viruses.

Designed with security in mind, Mac OS X Snow Leopard is not plagued by constant attacks from PC viruses and malware. Every Mac is highly secure right out of the box, so you can safely go about your work or play without interruption.

Robust operating system.

Every Mac comes with Mac OS X Snow Leopard, an operating system custom-made for the advanced technology inside. Built on a rock-solid UNIX foundation and designed to be simple and intuitive, it's what makes the Mac innovative, highly secure, compatible, and easy to use. [42]

Figure-20

Latest iMac Technical Specifications

Size and weight

21.5-inch iMac

Height:	**17.75** inches (45.1 cm)
Width:	**20.8** inches (52.8 cm)
Depth:	**7.42** inches (18.85 cm)
Weight:	**20.5** pounds (9.3 kg)[1]

27-inch iMac

Height:	**20.4** inches (51.7 cm)
Width:	**25.6** inches (65.0 cm)
Depth:	**8.15** inches (20.7 cm)
Weight:	**30.5** pounds (13.8 kg)[1]

Connections and expansion

- One FireWire 800 port; 7 watts
- Four USB 2.0 ports
- SD card slot

Display

- Built-in 21.5-inch (viewable) or 27-inch (viewable) LED-backlit glossy widescreen TFT active-matrix liquid crystal display with IPS technology
- Resolution
  - 21.5-inch models: 1920 by 1080 pixels
  - 27-inch models: 2560 by 1440 pixels
- 16:9 aspect ratio
- Millions of colors at all resolutions
- Typical viewing angle: 178° horizontal; 178° vertical
- Typical brightness: 320 cd/m^2 (21.5-inch models); 375 cd/m^2 (27-inch models)
- 27-inch model can be attached to a wall mount, articulating arm, or other VESA-compliant mounting solution using the optional VESA Mount Adapter Kit

Processor and memory

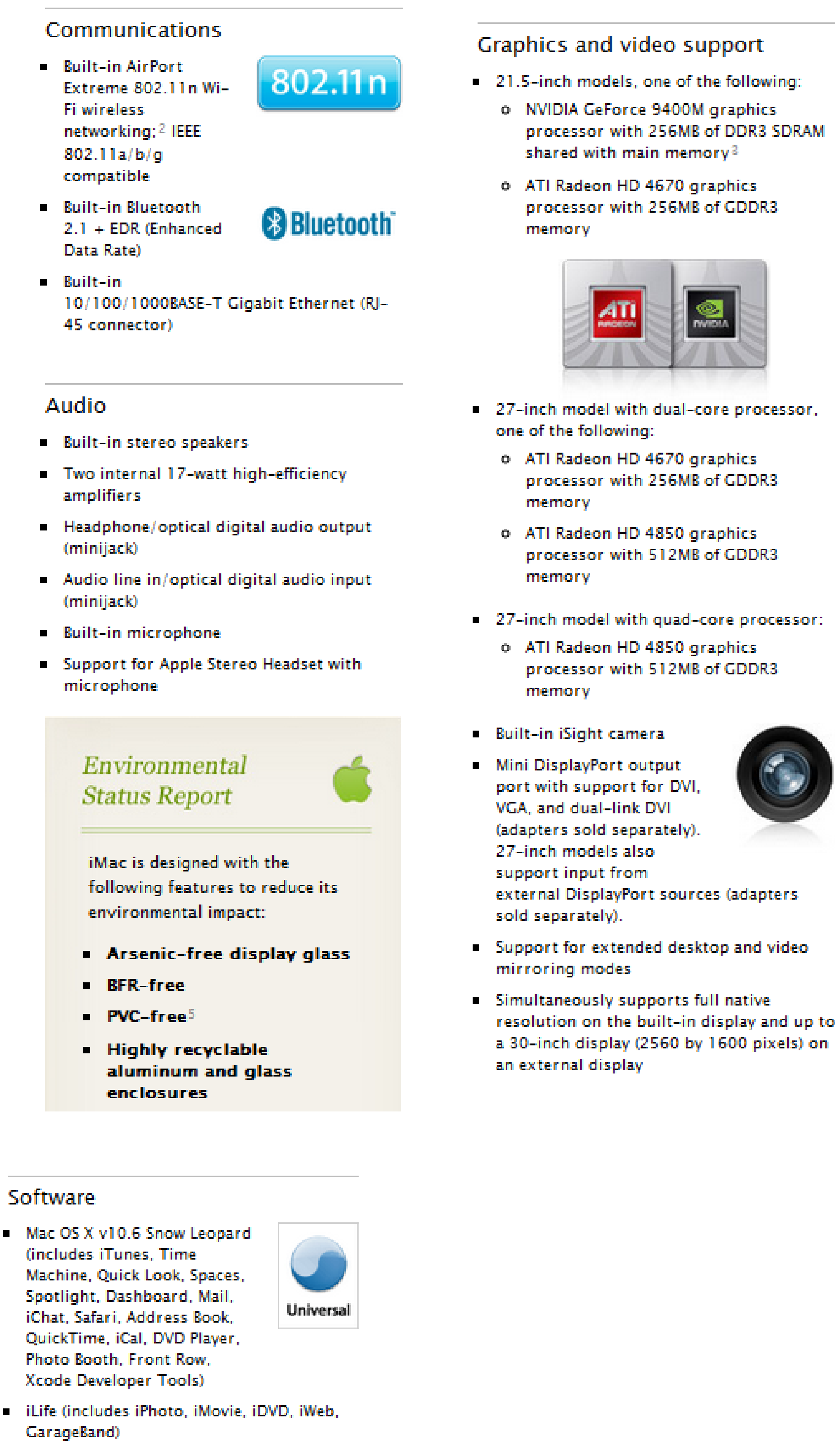
- 21.5-inch and 27-inch models, one of the following:
  - 3.06GHz Intel Core 2 Duo processor with 3MB shared L2 cache
  - 3.33GHz Intel Core 2 Duo processor with 6MB shared L2 cache
- 27-inch models only, one of the following:
  - 2.66GHz quad-core Intel Core i5 processor with 8MB shared L3 cache; Turbo Boost dynamic performance up to 3.2GHz
  - 2.8GHz quad-core Intel Core i7 processor with 8MB shared L3 cache; Turbo Boost dynamic performance up to 3.46GHz; Hyper-Threading for up to eight virtual cores
- 4GB (two 2GB SO-DIMMs) of 1066MHz DDR3 SDRAM; four SO-DIMM slots support up to 16GB

Configure your iMac now, only at the Apple Online Store.

Storage[4]

- 21.5-inch models:
  - 500GB or 1TB 7200-rpm Serial ATA hard drive
  - Optional 2TB 7200-rpm Serial ATA hard drive
- 27-inch models:
  - 1TB 7200-rpm Serial ATA hard drive
  - Optional 2TB 7200-rpm Serial ATA hard drive

Figure-21



Communications

- Built-in AirPort Extreme 802.11n Wi-Fi wireless networking;[2] IEEE 802.11a/b/g compatible
- Built-in Bluetooth 2.1 + EDR (Enhanced Data Rate)
- Built-in 10/100/1000BASE-T Gigabit Ethernet (RJ-45 connector)

Audio

- Built-in stereo speakers
- Two internal 17-watt high-efficiency amplifiers
- Headphone/optical digital audio output (minijack)
- Audio line in/optical digital audio input (minijack)
- Built-in microphone
- Support for Apple Stereo Headset with microphone

Environmental Status Report

iMac is designed with the following features to reduce its environmental impact:

- **Arsenic-free display glass**
- **BFR-free**
- **PVC-free**[5]
- **Highly recyclable aluminum and glass enclosures**

Graphics and video support

- 21.5-inch models, one of the following:
  - NVIDIA GeForce 9400M graphics processor with 256MB of DDR3 SDRAM shared with main memory[3]
  - ATI Radeon HD 4670 graphics processor with 256MB of GDDR3 memory

- 27-inch model with dual-core processor, one of the following:
  - ATI Radeon HD 4670 graphics processor with 256MB of GDDR3 memory
  - ATI Radeon HD 4850 graphics processor with 512MB of GDDR3 memory
- 27-inch model with quad-core processor:
  - ATI Radeon HD 4850 graphics processor with 512MB of GDDR3 memory
- Built-in iSight camera
- Mini DisplayPort output port with support for DVI, VGA, and dual-link DVI (adapters sold separately). 27-inch models also support input from external DisplayPort sources (adapters sold separately).
- Support for extended desktop and video mirroring modes
- Simultaneously supports full native resolution on the built-in display and up to a 30-inch display (2560 by 1600 pixels) on an external display

Software

- Mac OS X v10.6 Snow Leopard (includes iTunes, Time Machine, Quick Look, Spaces, Spotlight, Dashboard, Mail, iChat, Safari, Address Book, QuickTime, iCal, DVD Player, Photo Booth, Front Row, Xcode Developer Tools)
- iLife (includes iPhoto, iMovie, iDVD, iWeb, GarageBand)

Appendix 3 “Development Tools”

3.1. Objective-C:

The Objective-C language is a simple computer language, which designed to enable sophisticated object-oriented programming. Objective-C extends the standard ANSI C language by providing syntax for defining classes, and methods, as well as other constructs that promote dynamic extension of classes. Object-oriented concepts, such as encapsulation, inheritance, and polymorphism, are all present in Objective-C with a few important differences (such as there is no multiple inheritance in Objective-C different than Java language).

Extension of ANSI C

Objective-C is an extension of the ANSI version of the C programming language and supports the same basic syntax as C. As with C code, you define header files and implementation (or source) files to separate public declarations from the implementation details of your code. Objective-C header files use the file extensions listed in Table-1.

Table-1 File extensions for Objective-C code

Extension	Source type
.h	Header files. Header files contain class, type, function, and constant declarations.
.m	Implementation (or source) files. This is the typical extension used for source files and can contain both Objective-C and C code.
.mm	Source files. A source file with this extension can contain C++ code in addition to Objective-C and C code. This extension should be used only if you actually refer to C++ classes or features from your Objective-C code.

You typically use #import directive, when you attempt to include header files into your implementation file in Objective-C. This is similar to #include in C language, except that it makes sure that the same file is never included more than once.

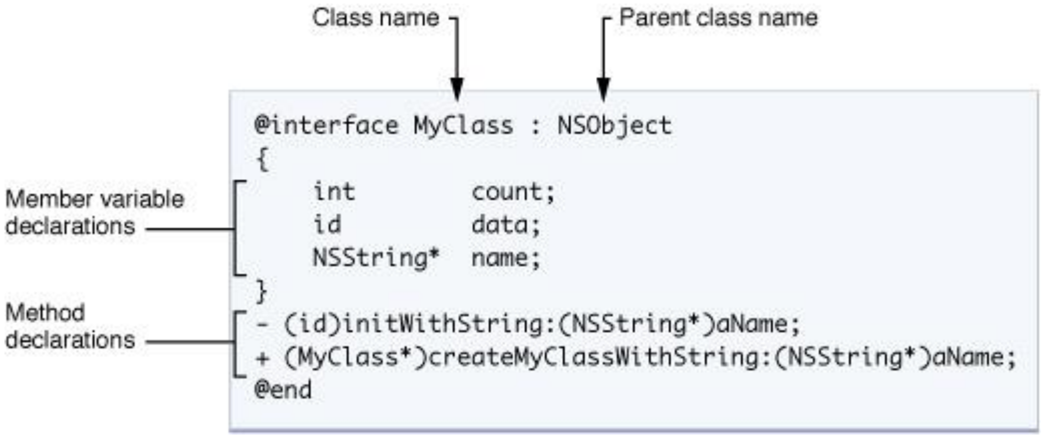
Classes in Objective-C

Classes in Objective-C provide the basic construct for encapsulating some data with the actions that operate on that data as in most other object-oriented languages. An object is a runtime instance of a class, and contains its own in-memory copy of the instance variables declared by that class and pointers to the methods of the class.

The specification of a class in Objective-C requires two distinct pieces: the interface and the implementation. The interface portion contains the class declaration and defines the instance variables and methods associated with the class. The interface is usually in a .h file. The implementation portion contains the actual code for the methods of the class. The implementation is usually in a .m file.

Figure-22 below shows the syntax for declaring a class called MyClass, which inherits from the NSObject base class. The class declaration always begins with the @interface compiler directive and ends with the @end directive. Following the class name (and separated from it by a colon) is the name of the parent class. The instance (or member) variables of the class are declared in a code block that is delineated by braces. Following the instance variable block is the list of methods declared by the class. A semicolon character marks the end of each instance variable and method declaration.

Figure-22



```
@interface MyClass : NSObject
{
    int         count;
    id          data;
    NSString*   name;
}
- (id)initWithString:(NSString*)aName;
+ (MyClass*)createMyClassWithString:(NSString*)aName;
@end
```

When storing objects in variables, you always use a pointer type. Objective-C supports both strong and weak typing for variables containing objects. Strongly typed pointers include the class name in the variable type declaration. Weakly typed pointers use the type id for the object instead. Weakly typed pointers are used frequently for things such as collection classes, where the exact type of the objects in a collection may be unknown. They provide tremendous flexibility and allow for much greater dynamism in Objective-C programs.

The following example shows strongly and weakly typed variable declarations:

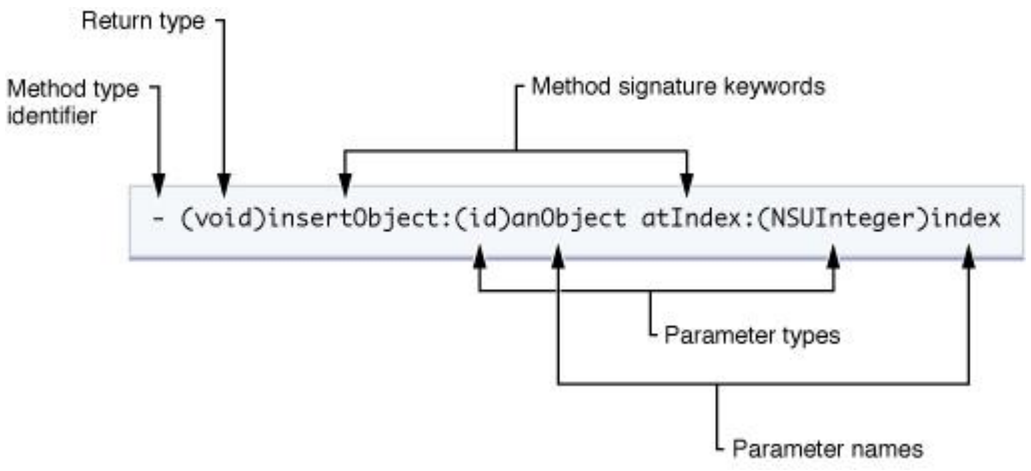
```
MyClass *myObject1;   //Strong typing
id       myObject2;   //Weak typing
```

Methods and Messaging in Objective-C

A class in Objective-C can declare two types of methods: instance methods and class methods. An instance method is a method whose execution is scoped to a particular instance of the class. In other words, before you call an instance method, you must first create an instance of the class. Class methods, by comparison, do not require you to create an instance.

The declaration of a method consists of the method type identifier, a return type, one or more signature keywords, and the parameter type and name information. Figure-23 shows the declaration of the `insertObject:atIndex:` instance method.

Figure-23


Return type
Method type identifier
Method signature keywords
- (void)insertObject:(id)anObject atIndex:(NSUInteger)index
Parameter types
Parameter names

This declaration is preceded by a minus (-) sign, which indicates that this is an instance method. The method’s actual name (insertObject:atIndex:) is a concatenation of all of the signature keywords, including colon characters. The colon characters declare the presence of a parameter. If a method has no parameters, you omit the colon after the first signature keyword. In this example, the method takes two parameters.

When you attempt to call a method, you do so by **messaging** an object. A message is the method signature, along with the parameter information the method needs. All messages you send to an object are dispatched dynamically, thus facilitating the polymorphic behavior of Objective-C classes.

Messages are enclosed by brackets. Inside the brackets, the object receiving the message is on the left side and the message (along with any parameters required by the message) is on the right.

For example, to send the `insertObject:atIndex:` message to an object in the myArray variable, you would use the following syntax:

```
[myArray insertObject:anObject atIndex:0];
```

To avoid declaring numerous local variables to store temporary results, Objective-C lets you nest messages. The return value from each nested message is used as a parameter, or as the target, of another message. For example, you could replace any of the variables used in the previous example with messages to retrieve the values. Thus, if you had another object called myAppObject that had methods for accessing the array object and the object to insert into the array, you could write the preceding example to look something like the following:

```
[[myAppObject theArray] insertObject:[myAppObject objectToInsert]
atIndex:0];
```

Objective-C also provides a dot syntax for invoking accessor methods. Accessor methods get and set the state of an object, and typically take the form -(type)*propertyName* and -(void)set*PropertyName*:(type). Using dot syntax, you could rewrite the previous example as:

```
[myAppObject.theArray      insertObject:[myAppObject      objectToInsert]
atIndex:0];
```

You can also use dot syntax for assignment:

```
myAppObject.theArray = aNewArray;
```

This is simply a different syntax for writing, [myObject setTheArray:aNewArray];.

Although the previous examples sent messages to an instance of a class, you can also send messages to the class itself. When messaging a class, the method you specify must be defined as a class method instead of an instance method. You can think of class methods as something akin to (but not exactly like) static members in a C++ class.

You typically use class methods as factory methods to create new instances of the class or for accessing some piece of shared information associated with the class. The syntax for a class method declaration is identical to that of an instance method, with one exception. Instead of using a minus sign for the method type identifier, you use a plus (+) sign.

The following example illustrates how you use a class method as a factory method for a class. In this case, thearray method is a class method on the NSArray class and inherited by NSMutableArray that allocates and initializes a new instance of the class and returns it to your code.

```
NSMutableArray *myArray = nil;  // nil is essentially the same as NULL


// Create a new array and assign it to the myArray variable.
myArray = [NSMutableArray array];
```

Figure-24 below shows the implementation of MyClass from the preceding example. Like the class declaration, the class implementation is identified by two compiler directives, @implementation and @end. These directives provide the scoping information the compiler needs to associate the enclosed methods with the corresponding class. A method definition therefore matches its corresponding declaration in the interface, except for the inclusion of a code block.

Figure-24

```
@implementation MyClass


- (id)initWithString:(NSString *)aName

{

    if (self = [super init]) {

        name = [aName copy];

    }

    return self;

}


+ (MyClass *)createMyClassWithString: (NSString *)aName

{

    return [[[self alloc] initWithString:aName] autorelease];

}

@end
```

Declared Properties in Objective-C

Declared properties are a convenience notation used to replace the declaration and the implementation of accessor methods.

You include property declarations with the method declarations in your class interface. The basic definition uses the @property compiler directive, followed by the type information and name of the property. You can also configure the property with custom options, which define how the accessor methods behave. The following example shows a few simple property declarations:

```
@property BOOL flag;

@property (copy) NSString *nameObject;  // Copy the object during assignment.

@property (readonly) UIView *rootView;  // Declare only a getter method.
```

Each readable property specifies a method with the same name as the property. Each writable property specifies an additional method of the form set*PropertyName*:, where the first letter of the property name is capitalized.

In your class implementation, you can use the @synthesize compiler directive to ask the compiler to generate the methods according to the specification in the declaration:

```
@synthesize flag;
@synthesize nameObject;
@synthesize rootView;
```

You can combine the @synthesize statements in a single line if you want:

```
@synthesize flag, nameObject, rootView;
```

Properties useful for reduce the amount of redundant code you have to write. Because most accessor methods are implemented in similar ways, properties eliminate the need to implement a getter and setter method for each property exposed in the class. Instead, you specify the behaviour you want using the property declaration and then synthesize actual getter and setter methods based on that declaration at compile time.

Strings in Objective-C

Objective-C supports the same conventions for specifying strings as C. In other words, single characters are enclosed by single quotes and strings of characters are surrounded by double quotes. However, most Objective-C frameworks do not use C-style strings very often. Instead, most frameworks pass strings around in NSString objects.

The NSString class provides an object wrapper for strings that has all of the advantages you would expect, including built-in memory management for storing arbitrary-length strings, support for Unicode, printf-style formatting utilities, and more. Because such strings are used commonly though, Objective-C provides a shorthand notation for creating NSString objects from constant values. To use this shorthand, all you have to do is precede a normal, double-quoted string with the @ symbol, as shown in the following examples: [43]

```
NSString *myString = @"My String\n";

NSString *anotherString = [NSString stringWithFormat:@"%d %s", 1,
@"String"];


// Create an Objective-C string from a C string

NSString *fromCString = [NSString stringWithCString:"A C string"
encoding:NSASCIIStringEncoding];
```

3.2. Xcode:

Xcode is Apple's premiere development environment for Mac OS X.[44] It is a mature and advanced suite of developer tools used to create software products. This suite includes GUI based applications (Interface Builder, iPhone Simulator), command-line tools, and documentation to assist you in the software development process.

The main application of the suite is the integrated development environment (IDE), also named Xcode, which provides an elegant and powerful user interface for creating and managing software development projects. You use Xcode to organize and edit your source files, view documentation, build your application, debug your code, and optimize your application’s performance.[45]

Figure-25

Developing with Xcode is all about keeping you focused. You simply click the green Build and Go button to start the build, debug, and test cycle. Build errors are displayed within your source code as Message Bubbles. Once your project is built, the debugger bar appears in the editor window, and hovering your mouse reveals variable values as Data Tips. If you are developing for iPhone, Xcode automatically installs your application on the device and attaches the debugger over USB. Throughout, Xcode keeps your code front and center.[46]

3.3. Interface Builder:

The Interface Builder application lets you design your application's user interface graphically and save those designs as resource files (special XML files) that your application loads at runtime. [47]

> "*When you open any *.xib file in Interface Builder, it will automatically show the Main window, your view, and a library of UI elements. Interface Builder allows you to drag and drop any of the basic library elements into your view, edit them, and work with the connections between the code and these elements. All of these elements come from the Cocoa Touch framework, a custom UI framework for both the iPhone and iPod Touch.*" [48]

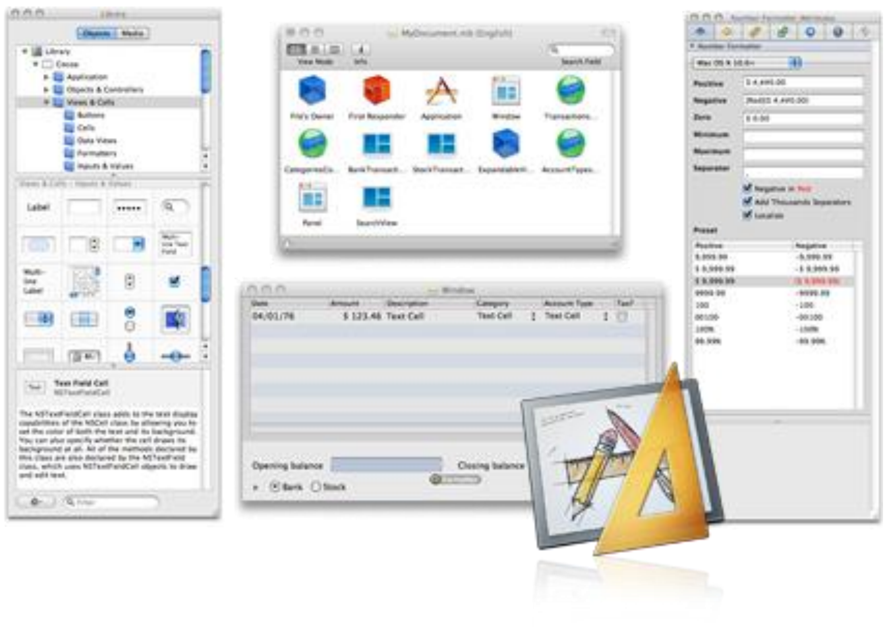


Figure-26

You can easily build user interfaces because Cocoa is built using the Model-View-Controller pattern. In fact, the user interfaces are actually archived Cocoa objects that require no code generation. It is creating an XML description of the GUI you are building, and the Cocoa Touch framework uses that to actually create the buttons. Cocoa interface objects are dynamically connected to your implementation code at runtime. Changes to the user interface do not require you to recompile your code, and changes to your code do not require you to recompile the user interface. [49]

3.4. iPhone Simulator:

The iPhone simulator application implements the iPhone OS API, providing an environment that closely resembles the environment iPhone OS devices provide. It allows you to run your applications in Mac OS X, letting you test most of their functionality without the need for an iPhone OS device. [50]

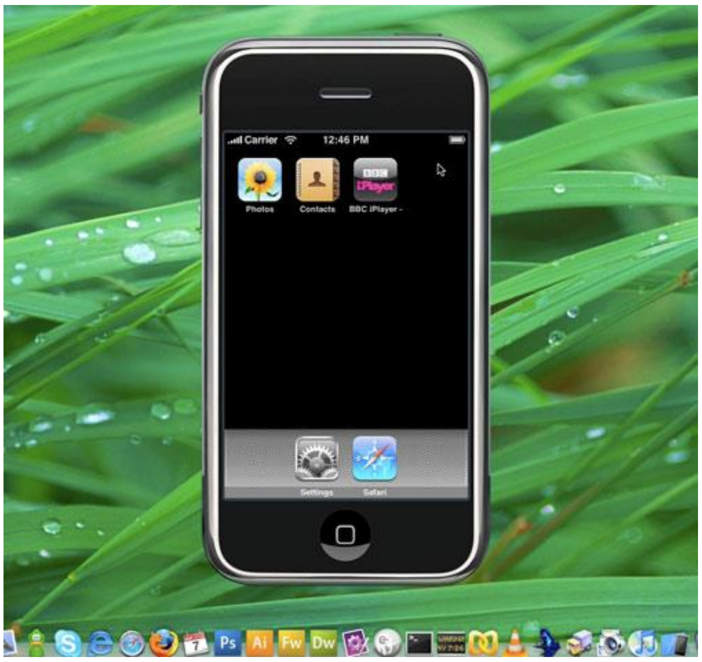


Figure-27

Run, test, and debug your application locally on your Mac using a simulated iPhone.

3.5. Instruments and Shark:

The Instruments application is a performance measurement tool that lets you peer into your code as it is running and gather important metrics about what it is doing. You can view and analyze the data Instruments collects in real time, or you can save that data and analyze it later. Instruments collects data such as disk, memory, or CPU usage in real time, either on your Mac or remotely from a connected iPhone. The collected data is graphically displayed as tracks over time, making it easy to pinpoint problem areas, and then drill down to the offending lines of code. When maximum speed is required, Shark lets you sample at the micro-second level with incredible detail.[51]

Shark is a code-profiling application that lets you analyze the performance of your code and allows you to profile the entire system. Shark profiles the system while your code is running to see where time is being spent and can also produce profiles of performance events. This information is an invaluable first step in your performance tuning effort so you can see which parts of your code or the system are the bottlenecks.[52]

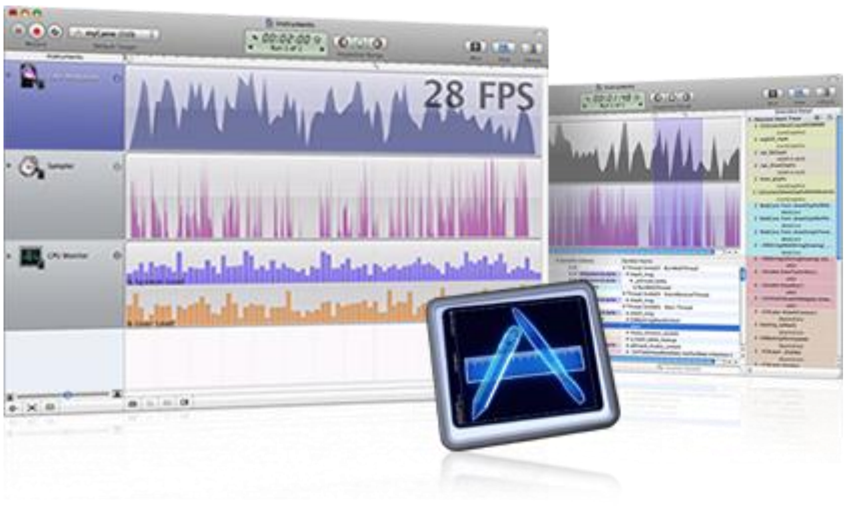


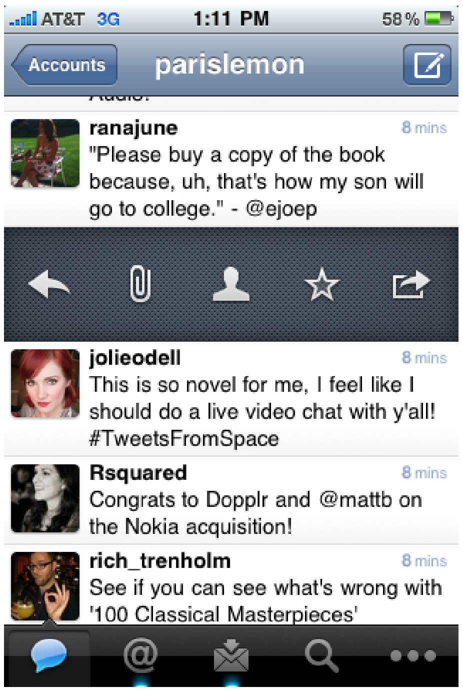
Figure-28

Appendix 4 “Categories of iPhone Apps”

4.1. Social Networking:

Tweetie 2

As the name implies, Tweetie 2 is the Twitter client for iPhone. It is the most popular Twitter client iPhone app that widely used by iPhone users. This app also be awarded with 2009 Apple design winner. 2.0 is the newest version, rewritten from the ground up with a fast and powerful new core, Tweetie 2 offers the most polished mobile Twitter experience around.



ranajune 8 mins
"Please buy a copy of the book because, uh, that's how my son will go to college." - @ejoep

jolieodell 8 mins
This is so novel for me, I feel like I should do a live video chat with y'all! #TweetsFromSpace

Rsquared 8 mins
Congrats to Dopplr and @mattb on the Nokia acquisition!

rich_trenholm 8 mins
See if you can see what's wrong with '100 Classical Masterpieces'

Figure-29

Features

- Seamlessly handle multiple Twitter accounts.
- Explore all of Twitter, from your own timeline and mentions, to the favorite tweets of your followers and friends.
- Full persistence: more than just caching tweets, Tweetie 2 restores your entire UI if you quit or get a phone call.
- Fantastic new offline mode. Read, tweet, favorite, follow, save to Instapaper and more even when you don't have a connection. Your actions will be synced as soon as you go back online.
- Full landscape support (configurable).
- Live-filter your tweet stream.
- Post photos and videos, even configure your own custom image host.
- Vastly improved compose screen with recent hashtags, @people picker, URL shortening and more.
- Compose screen multiple-attachments manager.
- Drafts manager ensures you never lose a tweet (and you can even send drafts to Birdhouse).

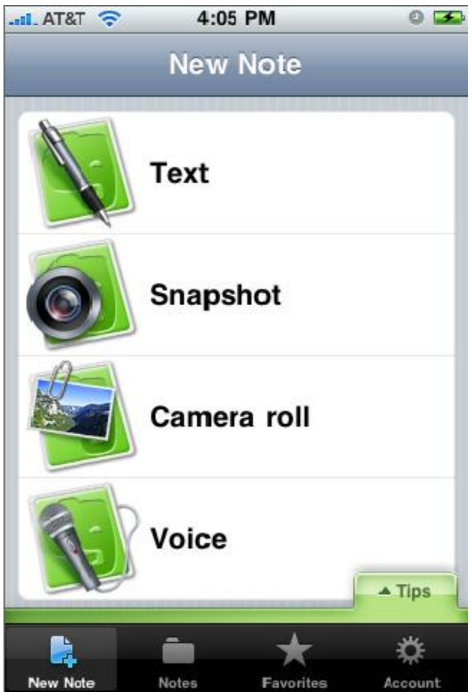
- Link Twitter contacts to Address Book contacts.
- Follow, unfollow, block and unblock from multiple accounts simultaneously.
- Saved searches sync with Twitter.com and the upcoming Tweetie 2 for Mac.
- Autocomplete recent searches and Go-to-user.
- Threaded Direct Messages and improved conversation navigation.
- TextExpander integration.
- Rich integration with Follow Cost, Tweet Blocker, Favstar.fm and more.
- Edit your own Twitter profile.
- Specify custom API roots on a per-account basis.
- Nearby map view.
- Translate tweets.
- Preview short URLs.
- Safari bookmarklet support for easily sharing links.
- In-app rich text email composition.
- Tons of little things, including improved avatar caching, auto-refresh, refresh-all, seamless Twitlonger support, hashtag definitions and more.

Last version 2.0 was released on 9 October 2009, and purchasable (downloadable) via iTunes with 2.39 € fee.[53]

4.2. Productivity:

Evernote

Figure-30



Evernote app for iPhone lets you create notes, snap photos, and record voice memos that you can then access any time from your iPhone, computer, or the web. Evernote basically turns the iPhone and iPod Touch into an extension of your brain. This multiple award winning app widely used over a million people.

Features

- Create, text, photo and audio notes.
- Auto-synchronizes your notes with Mac, PC, and Web.
- Character recognition makes text in snapshots searchable.
- Selected “Favourite” notes can be accessed quickly, even offline.
- All notes include geo-location information for mapping and search.
- Premium Feature: File Synchronization – add, sync, access, and share files (PDF, Word, Excel, PPT, and more) among the different versions of Everynote you use.

Areas to use Everynote

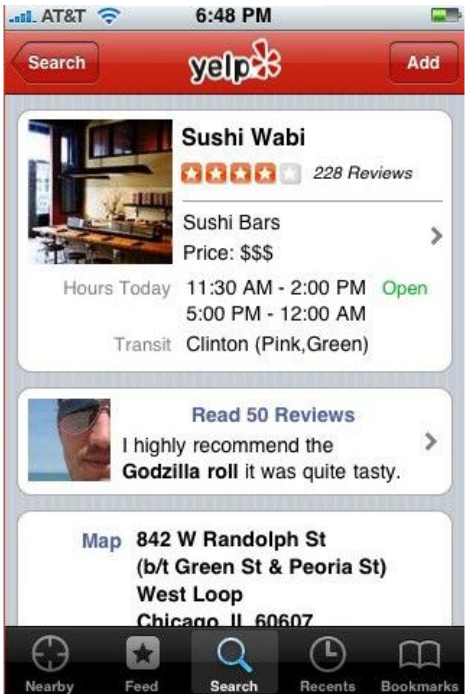
- For research and class notes.
- To capture blog ideas and design inspiration.
- To stay in-sync with your desktop notes, web clips, and files.
- To snap photos of whiteboards and wine labels.
- As part of your GTD system to help you stay organized.
- To record quick voice memos.

Last version 3.1.1 was released on 8 October 2009, and downloadable via iTunes with no fee.[54]

4.3. Utility:

Yelp

Figure-31



6:48 PM

Sushi Wabi

228 Reviews

Sushi Bars

Price: $$$

Hours Today 11:30 AM - 2:00 PM Open
5:00 PM - 12:00 AM

Transit Clinton (Pink,Green)

Read 50 Reviews

I highly recommend the **Godzilla roll** it was quite tasty.

Map 842 W Randolph St
(b/t Green St & Peoria St)
West Loop
Chicago, IL 60607

Yelp is a social review service, the first Augmented Reality (AR) iPhone app specifically for US.[55] This app also has info on Canadian and UK businesses. Yelp use to search for places to eat, shop, drink, relax and play then app allows users to reviews from an active community of locals in the know.

Features

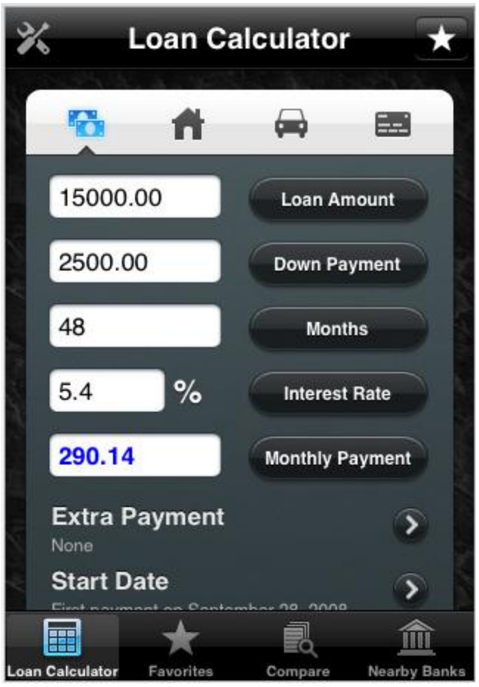
- Search for businesses near you using the iPhone's built-in location finder.
- Tab quick links to find nearby pubs, restaurants, cafes and more.
- Browse reviews to read what's great (and not so great) in your city. Look up addresses and phone numbers for thousands of businesses, then call or map them from your iPhone.

Last version 3.1.1 was released on 7 October 2009, and downloadable via iTunes with no fee.[56]

4.4. Finance:

Loan Shark

Figure-32



Navigating the waters of financial lending can be very dangerous indeed. Loan Shark helps you fight off predators and assists you in making sound financial decisions for you and your family. Loan Shark is an easy to use loan calculator that you just need to enter the information you know, click a button for the field you want calculated, and Loan Shark fills in the amount for you.

Features

- Calculate any component of a loan, including payment, interest rate, or loan amount.
- See the full Amortization Table for the loan's lifespan.
- See the effect of making an extra payment per year.
- Save loan figures for comparison shopping or examining different loan scenarios.
- Find banks in your area.
- Uses your local currency (as set in the iPhone General preferences)
- Semi-annual interest for Canada and other countries.
- Compare multiple loans side-by-side.

Great Uses

- Enter loans from different banks and save for later comparision.
- Calculate how long it will take to pay off credit cards.
- Determine how much interest loans are costing you.

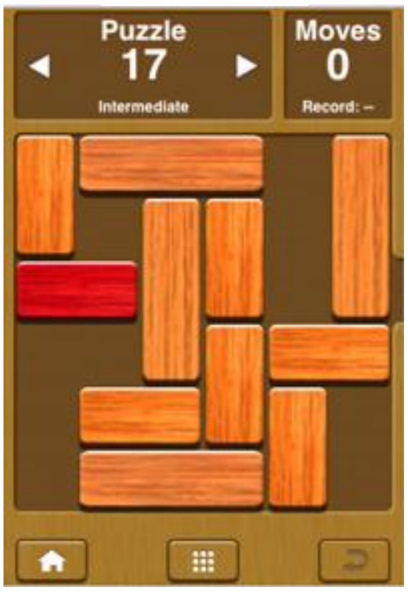
- Track the process of your mortgage.
- Calculates cash, home, car, and credit card loans.

Last version 2.0.7 was released on 2 April 2009, and purchasable (downloadable) via iTunes with 2.39 € fee. [57]

4.5. Game:

Unblock Me

Figure-33



Unblock Me is a simple and addictive puzzle game inspired by Nob Yoshigahara. The aim of the game is get the red block out of the board by sliding the other blocks out of the way. Unblock Me comes with 4 difficult levels ranging from Beginner to Expert each with 600 puzzles. There are 3000 puzzles in total worth hours of playing to keep you challenged. Free version of the game also available that offers 200 levels, most casual gamers will find that number plenty challenging.

Features

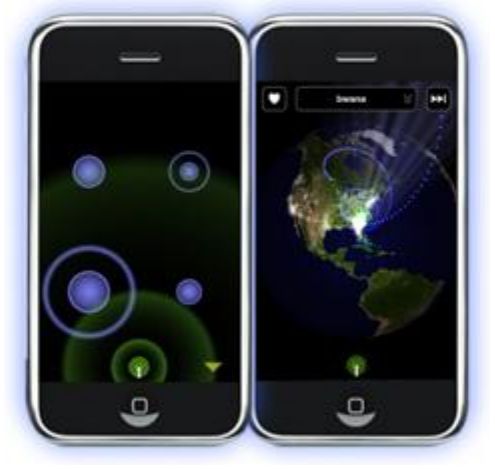
- 4 difficult levels ranging from Beginner to Expert.
- 600 bonus puzzles.
- 3000 unique puzzles in total for you to enjoy.
- Ranking system to keep you more challenged.
- Keep track of all the puzzles you’ve cleared.
- It’s addicting game.
- You can try before by with the free version.

Last version 1.2.1 was released on 17 September 2009, and purchasable (downloadable) via iTunes with 0.79 € fee, also free version is downloadable. [58]

4.6. Music:

Ocarina

Figure-34



Ocarina is one of Apple’s all-time top 20 applications. This app is the first true musical instrument created for the iPhone. Ocarina is sensitive to your breath, touch and movements, making it even more versatile than the original. There are no pre-compiled riffs so musicians will find unlimited opportunities for self-expression. Advanced options allow you to choose between diatonic, minor and harmonic scales.

Features

- Blow into your microphone to play music.
- Hold down combinations of holes to change pitch.
- Tilt your phone to change vibrato rate and depth.
- Advanced users can change keys and modes, including C-Major and the new Zeldarian mode.
- Or, instead of using your breath, simply press the holes to make music in touch mode.
- Tap the globe icon to hear Smule Ocarina players from around the world.
- Show love for your favorite melodies and listen to most loved melodies.
- Manually or automatically record your melodies while you're playing.
- Share your melodies with friends and family over email right from your phone.
- Archive and display your melodies on your own "My Ocarina" webpage.

Last version 1.3.3 was released on 23 October 2009, and purchasable (downloadable) via iTunes with 0.79 € fee. [59]

6. Bibliography

[1] MG Siegler (September 2009), Preview: Tweetie 2 Takes The Best iPhone Twitter App And Ups The Sex Appeal, TechCrunch, http://www.techcrunch.com/2009/09/28/preview-tweetie-2-takes-the-best-iphone-twitter-app-and-ups-the-sex-appeal/ [Accessed Nov 29th, 2009]

[2] iPhoneHacks (January 2009), Evernote's iPhone App Helps you Organize Life's Information, http://www.iphonehacks.com/2009/01/evernotes-iphone-app-helps-you-organize-lifes-information.html [Accessed Nov 29th, 2009]

[3] Vincent (August 2009), Yelp IPhone application, Zertaprom Inc., http://www.theiphonesoftware.com/?p=253 [Accessed Nov 29th, 2009]

[4] Michael Johnston (December 2008), Loan Shark (iPhone Alley's Review), iPhone Alley, http://www.iphonealley.com/reviews/apps/foggynoggin-software/loan-shark [Accessed Nov 29th, 2009]

[5] Loan Shark(Loan advisor for the iPhone), FoggyNoggin Software, http://foggynoggin.com/loanshark [Accessed Nov 29th, 2009]

[6] Ocarina (One of Apple's "All-time top 20 Apps"), SonicMule, http://ocarina.smule.com/ [Accessed Nov 29th, 2009]

[7] Wiki.media-culture (October 2004), Mobile Phone Technology – The Development, Wiki.media-culture, http://wiki.media-culture.org.au/index.php/Mobile_Phone_Technology_-_The_Development [Accessed Nov 29th 2009]

[8] Neil Mcallister (June 2009), How to choose a mobile development platform, Infoworld Inc., http://www.infoworld.com/d/developer-world/how-choose-mobile-development-platform-077 [Accessed Nov 29th, 2009]

[9] Dan Pilone & Tracey Pilone (October 2009), Head First iPhone Development, 1st Edition, O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, ISBN: 978-0-596-80354-4, page 4.

[10] Apple-iphone-unblocking editor (2007), iPhone History and Development, http://www.apple-iphone-unlocking.com/history-and-development.html [Accessed Nov 29th, 2009]

[11] Why you'll love iPhone, Apple Inc., http://www.apple.com/iphone/why-iphone/ [Accessed Nov 29th, 2009]

[12] iPhone Applications, Apple Inc., http://www.apple.com/iphone/how-to/index.html#basics.iphone-applications [Accessed Nov 29th, 2009]

[13] Tracy V. Wilson, How the iPhone Works, A Discovery Company, http://electronics.howstuffworks.com/iphone4.htm [Accessed Nov 29th, 2009]

[14] Customizing the Home Screen, Apple Inc., http://www.apple.com/iphone/how-to/index.html#basics.customizing-the-home-screen

[15] iPhone Applications, Apple Inc., http://www.apple.com/iphone/how-to/index.html#basics.iphone-applications

[16] Apple authors (October 2009), Apple iPhone Technology Overview, Apple Inc., http://developer.apple.com/iphone/library/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/iPhoneOSTechOverview.pdf [Accessed Nov 29th, 2009]

[17] Wikipedia, Android (Operating System), Wikimedia Foundation Inc., http://en.wikipedia.org/wiki/Android_(operating_system) [Accessed Nov 29th, 2009]

[18] Gareth Beavis (September 2008), A complete history of Android, TechRadar, http://www.techradar.com/news/phone-and-communications/mobile-phones/a-complete-history-of-android-470327 [Accessed Nov 29th, 2009]

[19] Android 2.0 r1 (November 2009), What is Android?, Apache 2.0, http://developer.android.com/guide/basics/what-is-android.html [Accessed Nov 29th, 2009]

[20] Android, Open Handset Alliance, http://www.openhandsetalliance.com/android_overview.html [Accessed Nov 29th, 2009]

[21] Android 2.0 r1 (November 2009), What is Android?, Apache 2.0, http://developer.android.com/guide/basics/what-is-android.html [Accessed Nov 29th, 2009]

[22] Wikipedia (October 2009), BlackBerry, Wikimedia Foundation Inc., http://en.wikipedia.org/wiki/BlackBerry [Accessed Nov 30th, 2009]

[23] BlackBerry, What is BlackBerry?, Research In Motion Limited, http://na.blackberry.com/eng/ataglance/blackberry.jsp [Accessed Nov 29th, 2009]

[24] Wikipedia (October 2009), BlackBerry, Wikimedia Foundation Inc., http://en.wikipedia.org/wiki/BlackBerry [Accessed Nov 30th, 2009]

[25] Wikipedia (October 2009), BlackBerry, Wikimedia Foundation Inc., http://en.wikipedia.org/wiki/BlackBerry [Accessed Nov 30th, 2009]

[26] BlackBerry, Introduction to Blackberry Development, Research In Motion Limited, http://www.blackberry.com/DevMediaLibrary/view.do?name=introblackberrydev [Accessed Nov 30th, 2009]

[27] Wikipedia, webOS, Wikimedia Foundation Inc., http://en.wikipedia.org/wiki/Palm_WebOS [Accessed Nov 30th, 2009]

[28] Palm (June 2009), Overview of webOS, Palm Inc., http://developer.palm.com/index.php?option=com_content&view=article&id=1761 [Accessed Nov 30th, 2009]

[29] Palm (June 2009), Overview of webOS, Palm Inc., http://developer.palm.com/index.php?option=com_content&view=article&id=1761 [Accessed Nov 30th, 2009]

[30] Palm (June 2009), Features Details, Palm Inc., http://www.palm.com/us/products/phones/pre/ [Accessed Nov 30th, 2009]

[31] Palm (June 2009), Palm webOS Platform Architecture, Palm Inc., http://developer.palm.com/index.php?option=com_content&view=article&id=1570 [Accessed Nov 30th, 2009]

[32] Wikipedia, Symbian OS, Wikimedia Foundation Inc., http://en.wikipedia.org/wiki/Symbian_OS [Accessed Nov 30th, 2009]

[33] Jayant Kumar Gandhi (May 2006), Symbian OS Overview, Jayant, http://www.jkg.in/12-symbian-os-overview/ [Accessed Nov 30th, 2009]

[34] Wikipedia, Symbian OS, Wikimedia Foundation Inc., http://en.wikipedia.org/wiki/Symbian_OS [Accessed Nov 30th, 2009]

[35] Jayant Kumar Gandhi (May 2006), Symbian OS Overview, Jayant, http://www.jkg.in/12-symbian-os-overview/ [Accessed Nov 30th, 2009]

[36] Symbian, Symbian Devices, Symbian Foundation, http://www.symbian.org/devices [Accessed Nov 30th, 2009]

[37] Wikipedia, Symbian OS, Wikimedia Foundation Inc., http://en.wikipedia.org/wiki/Symbian_OS [Accessed Nov 30th, 2009]

[38] Jayant Kumar Gandhi (May 2006), Symbian OS Overview, Jayant, http://www.jkg.in/12-symbian-os-overview/ [Accessed Nov 30th, 2009]

[39] Introduction to the Architecture of Symbian OS, Symbian Developer Works, http://developer.symbian.com/main/documentation/books/books_files/arch/arch_chapter.pdf

[40] Wikipedia, iMac, Wikimedia Foundation Inc., http://en.wikipedia.org/wiki/IMac [Accessed Nov 30th, 2009]

[41] Apple, Features, Apple Inc., http://www.apple.com/imac/features.html [Accessed Nov 30th, 2009]

[42] Apple, Why you'll love a Mac, Apple Inc., http://www.apple.com/imac/why-mac.html [Accessed Nov 30th, 2009]

[43] Apple, Learning Objective-C: A Primer, Apple Inc., http://developer.apple.com/iphone/library/referencelibrary/GettingStarted/Learning_Objective-C_A_Primer/ [Accessed Nov 30th, 2009]

[44] Apple, Tools Xcode, Apple Inc., http://developer.apple.com/tools/xcode/ [Accessed Nov 30th, 2009]

[45] Apple, Developer Tools, Apple Inc., http://developer.apple.com/iphone/library/referencelibrary/GettingStarted/GS_Tools_iPhone/index.html [Accessed Nov 30th, 2009]

[46] Apple, Developer Tools, Apple Inc., http://developer.apple.com/technology/tools.html [Accessed Nov 30th, 2009]

[47] Apple, Developer Tools, Apple Inc., http://developer.apple.com/iphone/library/referencelibrary/GettingStarted/GS_Tools_iPhone/index.html [Accessed Nov 30th, 2009]

[48] Dan Pilone & Tracey Pilone (October 2009), Head First iPhone Development, 1st Edition, O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, ISBN: 978-0-596-80354-4, page 14.

[49] Apple, Developer Tools, Apple Inc., http://developer.apple.com/technology/tools.html [Accessed Nov 30th, 2009]

[50] Apple, Developer Tools, Apple Inc., http://developer.apple.com/iphone/library/referencelibrary/GettingStarted/GS_Tools_iPhone/index.html [Accessed Nov 30th, 2009]

[51] Apple, Developer Tools, Apple Inc.,
http://developer.apple.com/technology/tools.html [Accessed Nov 30th, 2009]

[52] Apple, Developer Tools, Apple Inc.,
http://developer.apple.com/iphone/library/referencelibrary/GettingStarted/GS_Tools_iPhone/index.html [Accessed Nov 30th, 2009]

[53] Tweetie 2 (Twitter Client for iPhone), atebits LLC,
http://www.atebits.com/tweetie-iphone/ [Accessed Nov 30th, 2009]

[54] Evernote for iPhone and iPod Touch, Evernote Corporation,
http://www.evernote.com/about/download/iphone/ [Accessed Nov 30th, 2009]

[55] Marshall Kirkpatrick (August 2009), Yelp Brings First US Augmented Reality App to iPhone Store, ReadWriteWeb,
http://www.readwriteweb.com/archives/yelp_brings_first_us_augmented_reality_to_iphone_s.php [Accessed Nov 29th, 2009]

[56] Nick Mokey (August 2009), Best iPhone Apps, Digital Trends,
http://www.digitaltrends.com/mobile/app-attack-top-15-iphone-apps/18/ [Accessed Nov 30th, 2009]

[57] Loan Shark(Loan advisor for the iPhone), FoggyNoggin Software,
http://foggynoggin.com/loanshark [Accessed Nov 30th, 2009]

[58] UnBlock Me, Kiragames (Fun with iPod and iPhone Games),
http://www.kiragames.com/games/unblockme-free [Accessed Nov 29th, 2009]

[59] Ocarina (One of Apple's "All-time top 20 Apps"), SonicMule,
http://ocarina.smule.com/ [Accessed Nov 29th, 2009]

Figure-1 retrieved from,
iphone.wareseeker.com/.../430a46115b

Figure-2, Figure-30 retrieved from,
http://www.iphonehacks.com/2009/01/evernotes-iphone-app-helps-you-organize-lifes-information.html

Figure-3 retrieved from,
http://www.decimation.com/markw/2009/01/11/recommend-iphone-apps-to-get-you-started/

Figure-4, Figure-32 retrieved from, iTunes Store, Loan Shark.

Figure-5, Figure-33 retrieved from,
http://www.kiragames.com/games/unblockme

Figure-6, Figure-34 retrieved from,
http://ocarina.smule.com/

Figure-7 retrieved from,
http://www.techcrunch.com/2009/04/15/yelps-new-and-improved-iphone-app-officially-hits-the-app-store/

Figure-8 retrieved from,
http://www.impactlab.com/2008/12/29/att-offers-refurbished-iphone-3gs-starting-at-99/

Figure-9, Figure-10 retrieved from,
http://developer.apple.com/iphone/library/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/iPhoneOSTechOverview.pdf

Figure-11 retrieved from,
http://www.techradar.com/news/phone-and-communications/mobile-phones/a-complete-history-of-android-470327

Figure-12 retrieved from,
http://developer.android.com/guide/basics/what-is-android.html

Figure-13 retrieved from,
http://tr.blackberry.com/devices/blackberrybold/

Figure-14 retrieved from,
http://www.blackberry.com/DevMediaLibrary/view.do?name=introblackberrydev

Figure-15 retrieved from,
http://en.wikipedia.org/wiki/Palm_WebOS

Figure-16 retrieved from,
http://developer.palm.com/index.php?option=com_content&view=article&id=1570

Figure-17 retrieved from,
www.inf.u-szeged.hu/sed/symgcc

Figure-18 retrieved from,
http://www.slashphone.com/

Figure-19 retrieved from,
http://www.apple.com/

Figure-20, Figure-21 retrieved from,
http://www.apple.com/imac/specs.html

Figure-22 retrieved from,
http://www.apple.com/

Table-1, Figure-23, Figure-24 retrieved from,
http://developer.apple.com/iphone/library/referencelibrary/GettingStarted/Learning_Objective-C_A_Primer/

Figure-25 retrieved from,
iphonesdktraining.com/

Figure-26, Figure-28 retrieved from,
http://developer.apple.com/technology/tools.html

Figure-27 retrieved from,
http://fourshapes.com/articles/2008/03/07/iphone-aspen-simulator-on-iplayer-site/

Figure-28 retrieved from,
http://fourshapes.com/articles/2008/03/07/iphone-aspen-simulator-on-iplayer-site/

Figure-29 retrieved from,
http://www.geardiary.com/2009/09/29/tweetie-2-ushering-in-a-new-era-of-iphone-app-abandonware/

Figure-31 retrieved from,
http://www.techcrunch.com/2009/04/15/yelps-new-and-improved-iphone-app-officially-hits-the-app-store/