



ISLAMIC UNIVERSITY OF TECHNOLOGY

Memory Lane: An iOS Based Smart Photo Editing Application

By:

Sheik Tanvir Ahmed (094411)

Md. Iftekhar Hossain (084411)

Supervised by:

Kashif Nizam Khan

Assistant Professor

Department of Computer Science and Engineering

*A project submitted in partial fulfilment of the requirements
for the degree of Bachelor of Science in Computer Science and Engineering*

Academic Year: 2012-2013

Department of Computer Science and Engineering

Islamic University of Technology.

A Subsidiary Organ of the Organization of Islamic Cooperation.

Dhaka, Bangladesh.

October 23, 2013

Declaration of Authorship

We, Sheik Tanvir Ahmed & Md. Iftekhar Hossain, declare that this project titled, 'Memory Lane: An iOS Based Smart Photo Editing Application' and the work presented in it are our own. We confirm that:

- This work was done wholly while in candidature for a Bachelor degree at this University.
- Where any part of this project has not been submitted previously for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.

Submitted By:

Sheik Tanvir Ahmed (094411)

Md. Iftekhar Hossain (084411)

Memory Lane: An iOS Based Smart Photo Editing Application

Approved By:

Prof. Dr. M.A. Mottalib
Head of the Department,
Department of Computer Science and Engineering,
Islamic University of Technology.

Kashif Nizam Khan
Project Supervisor,
Assistant Professor,
Department of Computer Science and Engineering,
Islamic University of Technology.

ISLAMIC UNIVERSITY OF TECHNOLOGY

Abstract

CSE

Department of Computer Science and Engineering

Bachelor of Science in Computer Science and Engineering

Memory Lane: An iOS Based Smart Photo Editing Application

by

Sheik Tanvir Ahmed

Md. Iftekhar Hossain

The need for a good and stable platform for mobile device programming was felt from the introduction of small handheld devices specially mobile phones. iOS is a mobile operating system developed and distributed by Apple Inc. The user interface of iOS is based on the concept of direct manipulation, using multi-touch gestures. iOS technology allows us to interact and manipulate our screen in a variety of ways.

In our project we will be using iOS platform to develop a Photo Editor. The name of our photo editor application is "Memory Lane". Since people take pictures to keep remembering past memories in form of image where life is a straight lane we walk through and by editing images we keep them how we want to remember our memories.

Acknowledgements

Special thanks to:

Prof. Dr. M.A. Mottalib

Head of the Department

Department of Computer Science and Engineering

Special thanks to:

Kashif Nizam Khan

Project Supervisor

Assistant Professor

Department of Computer Science and Engineering

Contents

Declaration of Authorship	i	
Abstract	iii	
Acknowledgements	iv	
List of Figures	vii	
1 Introduction	1	
1.1 Background		1
1.2 Why iOS?		2
1.2.1 Earn More Revenues		
1.2.2 Easy to test		
1.2.3 Less Resources		
1.2.4 Great Experience		
2 Literature Survey	4	
2.1 Documentation		4
2.1.1 iOS Platform		
2.1.2 Objective C		
2.1.3 Xcode-IDE		
2.1.4 Cocoa-Touch Framework		
2.1.5 JSON Framework		
2.2 Description Of Some Existing Project		11
2.2.1 Features Of Afterglow - Photo Editor By Simon Filip	11	
2.2.2 Features Of Wood Camera - Photo Editor By Bright Mango	13	
3 Mobile Programming	15	
3.1 Platforms for Mobile Programming		15
3.2 Java MicroEdition		15
3.3 Symbian and Maemo		16
3.4 Windows Mobile Programming		17

3.4.1	Programming Languages for Windows Mobile Programming	18	
3.5	Android		18
4	Our Project	20	
4.1	Memory Lane-Photo Editor		20
4.2	Features Of Our Project		21
4.2.1	Picture Frames		
4.2.2	Pre-packegeed Filter		
4.2.3	Default-packegeed Filter		
4.2.4	Social Networking		
4.2.5	Digital Face Editing		
4.2.6	Transformation Tools		
5	How To Use?	37	
5.1	Framing and Uploading		37
5.2	Main Menu		38
5.3	Import From Facebook		39
5.4	Editing A Picture		40
6	Scope & Goal	41	
6.1	Scope		41
6.2	Goal		41
7	Conclusion	43	
8	Reference	44	

List of Figures

4.1	Photo Frames	21
4.2	Posterization	22
4.3	Saturation	23
4.4	Brightness	24
4.5	Contrast	25
4.6	Gamma	26
4.7	Noise	27
4.8	Sharpening	29
4.9	Sepia	30
4.10	LOMO	31
4.11	Vignetting	32
4.12	Polaroidish	32
4.13	Default-packeged Filter	33
4.14	Digital Face Editor	35
4.15	Transformation Tools	36
5.1	Framing and Uploading	37
5.2	Main Menu	38
5.3	Import	39
5.4	Editing	40

Dedicated to our parents...

Chapter 1

Introduction

1.1 Background

iOS (previously iPhone OS) is a mobile operating system developed and distributed by Apple Inc. Originally released in 2007 for the iPhone and iPod Touch, it has been extended to support other Apple devices such as the iPad and Apple TV. Unlike Microsoft's Windows Phone (Windows CE) and Google's Android, Apple does not license iOS for installation on non-Apple hardware. As of September 12, 2012, Apple's App Store contained more than 700,000 iOS applications, which have collectively been downloaded more than 30 billion times. It had a 21% share of the smartphone mobile operating system units shipped in the fourth quarter of 2012, behind only Google's Android. In June 2012, it accounted for 65% of mobile web data consumption (including use on both the iPod Touch and the iPad). At the half of 2012, there were 410 million devices activated.[According to the special media event held by Apple on September 12, 2012, 400 million devices have been sold through June 2012. The user interface of iOS is based on the concept of direct manipulation, using multi-touch gestures. Interface control elements consist of sliders, switches, and buttons. Interaction with the OS includes gestures such as swipe, tap, pinch, and reverse pinch, all of which have specific definitions within the context of the iOS operating system and its multi-touch interface. Internal accelerometers are used by some applications to respond to shaking the device (one common result is the undo command) or rotating it in three dimensions (one common result is switching from portrait to landscape mode). iOS is derived from OS X, with which it shares the Darwin foundation. iOS is Apple's mobile version of the OS X operating system used on Apple computers. In iOS, there are four abstraction

layers: the Core OS layer, the Core Services layer, the Media layer, and the Cocoa Touch layer. The current version of the operating system (iOS 6.1.3) dedicates 1-1.5 GB of the device's flash memory for the system partition, using roughly 800 MB of that partition (varying by model) for iOS itself. iOS currently runs on iPhone, iPad, iPod touch and Apple TV.

1.2 Why iOS?

iOS is Apple's mobile operating system which was originally developed for the iPhone, but has now expanded to support various other Apple devices including the iPad and iPod touch.

iOS technology allows us to interact and manipulate our screen in a variety of ways. This makes it much more user friendly because we can alter the screen size by simply placing two fingers in the centre of the screen and dragging them away from each other. This is particularly useful if we have a visual impairment, or if we have just forgotten our glasses, because within an instant, everything can be made much bigger. Double tapping the screen returns the page to its normal size, or if we wish to zoom out, simply reverse the zoom in motion. There are plenty of other touch screen motions, such as sliding our finger to the left or the right to scroll through photos, as well as dragging and dropping items to customize our apps. And below these are the more practical reasons why we chose to work on iOS.

1.2.1 Earn More Revenues

As per Distimos Annual Report for 2011, iOS Developers made more money than Android Developers (source <http://bit.ly/z3Bj0E>). As per the report Apple App Store for iPhone generates about four times the revenue that is generated in the Google Android Market. That's a good reason to choose iOS over Android.

1.2.2 Easy to test

There are only a handful of iOS Devices that we will need to test our apps. Thus, we can be sure that when we release our app it is going to run smoothly on each and every iOS Device. Also, all these devices are very popular, that's why it's easy to find someone using an iPhone/iPod to test our app.

1.2.3 Less Resources

If we are going to develop for iOS platform we have to consider only two possible screen sizes (320x480 and 640x960) unless we are also developing for iPad. Also, even if we support just one screen size (320x480) it would run perfectly on higher resolution iPhone/iPod.

1.2.4 Great Experience

The touch response of all the iOS Devices is great and we will be very satisfied when we use your app on them. Apple products are always built using the best quality display and hardware components. Thus, we can be sure that our app will look great and respond quickly.

Chapter 2

Literature Survey

2.1 Documentation

2.1.1 iOS Platform

iOS is the world's most advanced mobile operating system. With its easy-to-use interface, amazing features, and rock-solid stability, iOS is the foundation of iPhone, iPad, and iPod touch. And even as others try to catch up, the technologies and features built into iOS to keep Apple devices years ahead. Together, the iOS SDK and Xcode IDE make it easy for developers to create revolutionary mobile apps. Derived from core OS X technologies, the amazing user experience of iOS has been streamlined to take maximum advantage of iPhone, iPad, and iPod touch hardware. Technologies shared between iOS and OS X include the OS X kernel, BSD sockets for networking, and Objective-C and C/C++ compilers for native performance. iOS devices share several unique characteristics that influence the user experience of all apps that run on them. The most successful apps embrace these characteristics and provide a user experience that integrates with the device they're running on.

2.1.1.1 The Display Is Paramount, Regardless of Its Size

Different iOS devices can have displays of different dimensions and resolutions, but in all devices the display affects the user experience in the same ways:

1. 44 x 44 points is the comfortable minimum size of a tappable UI element.

2. People are very aware of the quality of app artwork.
3. The display encourages people to forget about the device and to focus on their content or task.

2.1.1.2 Device Orientation Can Change

1. On iPhone and iPod touch, the Home screen is displayed in portrait orientation only, with the Home button at the bottom. This leads users to expect iPhone apps to launch in this orientation by default.
2. On iPad, the Home screen is displayed in all orientations, so users tend to expect iPad apps to launch in the device orientation they're currently using.

2.1.1.3 Apps Respond to Gestures, Not Clicks

People make specific finger movements, called gestures, to operate the unique Multi-Touch interface of iOS devices. For example, people tap a button to activate it, flick or drag to scroll a long list, or pinch open to zoom in on an image. The Multi-Touch interface gives people a sense of immediate connection with their devices and enhances their sense of direct manipulation of onscreen objects. People are comfortable with the standard gestures because the built-in apps use them consistently. Their experience using the built-in apps gives people a familiar set of gestures that they expect to be able to use successfully in most other apps. In addition to these gestures, people can move the device itself to perform certain tasks. Examples of these tasks are to activate Siri (on iPhone 4S) or to calibrate the compass. People expect all these gestures to work the same, regardless of the app they're currently running.

2.1.1.4 Interact with One App at a Time

Only one app is visible in the foreground at a time. When people switch from one app to another, the previous app transitions to the background and its user interface goes away. This feature, called multitasking, allows apps to remain in the background until users relaunch them or until they are terminated. Most apps enter a suspended state when they transition to the background. Suspended apps are displayed in the multitasking UI, which provides a convenient way for people to switch to recently used apps. The multitasking UI

appears at the bottom of the screen, below the UI of the currently running app or the Home screen. When people return to a suspended app, it can instantly resume running from the point where it went to the background, without having to reload its UI. Some apps might need to continue running in the background while users run another app in the foreground. For example, users might want to continue hearing the song that's playing in one app while they're using a different app to check their to-do list or handle email.

2.1.1.5 Preferences Are Available in Settings

People set certain preferences for an iOS app in the built-in Settings app. They must switch away from the current app when they want to access those preferences in Settings. Preferences in the Settings app are of the "set once and rarely change" type. Although some of the built-in apps provide preferences of this type, most apps do not need them, so they don't have preferences in the Settings app.

2.1.1.6 Onscreen User Help Is Minimal

Mobile users have neither the time nor the desire to read through a lot of help content before they can benefit from an app. What's more, help content takes up valuable space to store and display. iOS devices and the built-in apps are intuitive and easy to use, so people don't need onscreen help content to tell them how to use the device or the apps. This experience leads people to expect all iOS apps to be similarly easy to use.

2.1.1.7 Most iOS Apps Have a Single Window

An iOS app has a single window, unless it supports an external display. An app's window fills the device's main screen and provides an empty surface that hosts one or more views in which you present your content. It's important to realize that a window in an iOS app is very different from a window in a computer app. For example, an iOS window has no visible components (such as a title bar or a close button) and it can't be moved to a new location on the device display. It's also important to realize that most users are unaware of the windows and views in the iOS apps that they use. For the most part, users experience an iOS app as a collection of screens through which they navigate. From this perspective, a

screen generally corresponds to a distinct visual state or mode in an app. In the Contacts app on iPhone, for example, users think of their contact list—regardless of its length—as one screen and an individual contact’s details as a different screen.

2.1.1.8 Two Types of Software Run in iOS

There are two types of software that you can develop for iOS devices:

1. iOS apps
2. Web content

An iOS app is an app we develop using the iOS SDK to run natively on iOS devices. iOS apps resemble the built-in apps on iOS devices in that they reside on the device itself and take advantage of features of the iOS environment. People install iOS apps on their devices and use them just as they use built-in apps, such as Photos, Calendar, and Mail. Web content is hosted by a website that people visit using their iOS devices. There are three types of web content:

1. Web apps. Webpages that provide a focused solution to a task and conform to certain display guidelines are known as web apps because they behave similarly to iOS apps. A web app often hides the UI of Safari on iOS so that it looks more like a native app. Using the web clip feature, a web app can also supply an icon for people to put on the Home screen. This allows people to open web apps in the same way that they open iOS apps.
2. Optimized webpages. Webpages that are optimized for Safari on iOS display and operate as designed (with the exception of any elements that rely on unsupported technologies, such as plug-ins, Flash, and Java). In addition, an optimized webpage correctly scales content for the device screen and is often designed to detect when it is being viewed on iOS devices, so that it can adjust the content it provides accordingly.
3. Compatible webpages. Webpages that are compatible with Safari on iOS display and operate as designed (with the exception of any elements that rely on unsupported technologies, such as plug-ins, Flash, and Java). A compatible webpage does not tend to take extra steps to optimize the viewing experience on iOS devices, but the device usually displays the page successfully. An iOS app might combine native UI elements with access to

web content within a web content-viewing area. Such an app can look and behave like a native iOS app, without drawing attention to the fact that it depends on web sources.

2.1.1.9 Safari on iOS Provides the Web Interface

Safari on iOS provides the interface for browsing web content on iOS devices. Although Safari on iOS is similar in many ways to Safari on the computer desktop, it is not the same. For the most part, users can't change the size of the viewport (the area that displays content). On the desktop, users resize the viewport when they resize the browser window. On iOS devices, the viewport doesn't resize unless the device orientation changes. iOS users can change the scale of the viewport by zooming in and out, and they can pan the webpage. On iPad, users are much less likely to zoom web content than they are on iPhone.

2.1.2 Objective C

The Objective-C language is a simple computer language designed to enable sophisticated object-oriented programming. Objective-C is defined as a small but powerful set of extensions to the standard ANSI C language. Its additions to C are mostly based on Smalltalk, one of the first object-oriented programming languages. Objective-C is designed to give C full object-oriented programming capabilities, and to do so in a simple and straightforward way. These are the various features of Objective C

- An object-oriented programming language
- A library of objects
- A suite of development tools
- A runtime environment
- Classes are objects
- Dynamic typing and optional static typing
- Categories
- Message sending

- Expressive message syntax
- Introspection
- Dynamic run-time
- Automatic garbage collection
- C inside
- C++ fluent
- Simplicity
- Access to Apple technologies

2.1.3 Xcode-IDE

Xcode is an Integrated Development Environment (IDE) containing a suite of software development tools developed by Apple for developing software for OS X and iOS. The Xcode suite includes a modified version of the GNU Compiler Collection as well as, in Xcode 3.1 and later, the `llvm-gcc` compiler, with front ends from the GNU Compiler Collection and a code generator based on LLVM, and, in Xcode 3.2 and later, Apple's LLVM Compiler, with the `clang` front end and a code generator based on LLVM, and the Clang Static Analyzer. It supports C, C++, Objective-C, Objective-C++, Java, AppleScript, Python and Ruby source code with a variety of programming models, including but not limited to Cocoa, Carbon, and Java. Xcode developer tools package provides everything we need to create great applications for Mac, iPhone, and iPad. Xcode is tightly integrated with the Cocoa and Cocoa Touch frameworks, creating a productive and easy-to-use development environment that is powerful enough to be the same tools used by Apple to produce OS X and iOS. The Xcode toolset includes the amazing Xcode IDE, with the Interface Builder design tool and Apple LLVM compiler fully integrated. The Instruments analysis tool is also included, along with dozens of other supporting developer tools. Designed from the ground up to take advantage of the newest Apple technologies, Xcode integrates all the tools we need. The unified interface smoothly transitions from composing source code, to debugging, and even to designing our next stunning user interface, all within the same window. The Xcode workspace is all about keeping us focused. As we type, Live Issues will immediately alert us to coding mistakes, displaying a message bubble beside our code for more detail. Hit the

Run button to launch our Mac app, or upload the app to our test device, and immediately start debugging. Hover the mouse pointer above a variable to inspect its value at runtime, never having lost our place in the editor.

2.1.4 Cocoa-Touch Framework

Cocoa Touch is the programming framework driving user interaction on iOS. Using technology derived from Cocoa and the gorgeous Mac desktop, Cocoa Touch and the iOS interface were completely re-designed for multi-touch. Buttons, table lists, page transitions, and gestures on the iPhone are unique for the pocketable form factor, and all this UI power is available to developers through the Cocoa Touch frameworks. Built upon the Model-View-Controller paradigm, Cocoa Touch provides a solid foundation for creating state-of-the-art applications. When combined with the Interface Builder developer tool, it is both easy and fun to use drag-and-drop to design the next great iOS application. Strong low-level foundations enable fantastic high-level frameworks such as Game Kit for multiplayer gaming, Core Data, which offers high performance, yet easy-to-use data management, Core Animation for stunning effects, and the most innovative browser engine on mobile devices in WebKit. Working together, the Cocoa Touch frameworks and powerful foundation provide a truly unique canvas upon which to create a new work of application art. The Cocoa Touch frameworks that drive iOS apps share many proven patterns found on the Mac, but were built with a special focus on touch-based interfaces and optimization. UIKit provides the basic tools we need to implement graphical, event-driven applications in iOS. UIKit builds on the same Foundation framework infrastructure found on the Mac OS X, including file handling, networking, string building, and more. The unique interface of iOS means that Cocoa Touch has a unique design to match. Using UIKit we have access to the special GUI controls, buttons, and full-screen views on iOS. We also get to control our application with the accelerometer and the multi-touch gesture. Much of Cocoa Touch is implemented in Objective-C, an object-oriented language that is compiled to run at incredible speed, yet employs a truly dynamic runtime making it uniquely flexible. Because Objective-C is a superset of C, it is easy to mix C and even C++ into our Cocoa Touch applications. As our application runs, the Objective-C runtime instantiates objects based on executing logic - not just in ways defined during compilation. For example, a running Objective-C application can load an interface (a nib file created by Interface Builder), connect

the Cocoa objects in the interface to our application code, then run the correct method once the UI button is pressed. No recompiling is necessary.

2.1.5 JSON Framework

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language. JSON is built on two structures:

- A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

These are universal data structures. Virtually all modern programming languages support them in one form or another. It makes sense that a data format that is interchangeable with programming languages also be based on these structures. Features of JSON Framework:

- Streaming Support.
- Configurable recursion depth limit.
- Automatic Reference Counting (ARC).
- Optionally sort dictionary keys in JSON output.
- Optional pretty-printing of JSON output.

2.2 Description Of Some Existing Project

2.2.1 Features Of Afterglow - Photo Editor By Simon Filip

Afterglow is the perfect image editing application for quick and straight forward editing. It has a simple design, paired with powerful and snappy tools, will give

users the look they want in seconds. Like most photo apps, Afterglow allows users to either snap a new image or import from their Photo Library. Once we have an image, we can begin editing it to our heart's content with the bottom toolbar. Afterglow has adjustment tools, filters, textures and light leaks, frames, and cropping and transformation tools.

2.2.1.1 13 Adjustment Tools

Use any of the 13 uniquely made adjustment tools to enhance images to perfection with ease. Adjustment tools include stuff like Brightness, Contrast, Saturation, and Exposure. There are a total of 12 different adjustment tools to choose from, so you can completely fine-tune our image until it is perfect. Each tool will give us a slider to adjust the amount of change.

2.2.1.2 34 Filters

Afterglow features a total of 34 filters that are split up into two sets: Original and Guest. The Original set will have 21 filters from the developers, each adding some personality to your image. It has 13 "Guest Filters" created by various Instagram users. There are more guest filters coming in future updates. All filters can be tailored to the user's specific wants, so if an effect is too strong, we have to slide it down a few notches.

2.2.1.3 23 Texturess

Afterglow includes a range of real and natural light leaks being created with 35mm film, to simplistic scratchy film textures.

2.2.1.4 Cropping and Transformation Tools :

It can crop photos with quick and easy cropping tool with 13 different presets. Transform photos with rotating tool, vertical and horizontal mirroring tools, and even straightening.

2.2.1.5 Frames

It provides a list of 13 different simplistic and adjustable frames, paired perfectly with Instagram.

2.2.2 Features Of Wood Camera - Photo Editor By Bright Mango

2.2.2.1 Lightbox

We can easily import and view photos all in one place. Tap once to enter full-size preview mode, or double-tap to go straight to the editing screen.

2.2.2.2 Live Lenses

We can use the in-app camera to shoot with each of our 32 lenses in real-time. Each lens is precisely crafted to emphasize a different part of each photo's dynamic range.

2.2.2.3 Flexible Editing

Layer 32 lenses, 28 textures, and 16 frames for thousands of possible combinations. We can sharpen and straighten, and also adjust brightness, contrast, saturation, and hue.

2.2.2.4 Tilt-Shift and Vignettes

We can pinch to customize tilt-shift radius and vignette position and intensity. Use gestures to control linear tilt-shift for that perfect professional look.

2.2.2.5 One Touch Preview

Tap and hold to quickly compare your edited image to the original shot.

2.2.2.6 Non Destructive

We can enjoy the freedom to go back and modify any changes you made to the original image, even after exporting your work.

2.2.2.7 Full Resolution

Import and export images at the full, 8 MP resolution of the iPhone 4S and iPhone 5 cameras.

2.2.2.8 Share To Instagram

One touch sharing to our favorite mobile photography network

2.2.2.9 Social Networking

Quickly share our photos to the Camera Roll, Facebook, Twitter, Flickr or Dropbox.

Chapter 3

Mobile Programming

3.1 Platforms for Mobile Programming

The need for a good and stable platform for mobile device programming was felt from the introduction of small handheld devices specially mobile phones. Small processing power, limited battery life and relatively small graphical displays put a lot of constraint for the application developers. The conventional programming languages which are used to develop programs for normal computers and laptops are not suitable for small handheld devices like smart phones. Now a days, almost all the popular programming languages provide the support for embedded/mobile device programming. In addition, a good number of new platforms have been developed solely for the mobile device application development.

3.2 Java MicroEdition

Java is one of the most popular and flexible programming languages. Java MicroEdition which is well known as J2ME in the developers' arena, provides the same essence of Java with a flexible, stable and reliable environment for mobile application programming . It has been designed keeping the constraints of the embedded devices in mind. Java ME is not merely a programming language. It is a combination of the technology and its specifications. With the help of this platform, it is easy to produce a Java runtime environment full filling the requirement of limited memory and power. J2ME provides configurations with the essential libraries and virtual machine and APIs. With the current

advancement in mobile technologies, now there is also a large gap between the capabilities of various mobile phones. Some of the mobile devices now possess the capability of extra processing power and memory than the other limited ones. Keeping this difference in mind, J2ME provides two different types of configurations for mobile device programming: Connected Limited Device Configuration (CLDC) and the Connected Device Configuration (CDC). The basic features of J2ME includes cross-platform compatibility, better user experience, security and safety, rich class libraries for easy application development and last but certainly not the least, is to allow the basics of Java technology so that a skilled Java developer can easily grasp its basics.

3.3 Symbian and Maemo

Symbian and Maemo are two very famous OSs for mobile device programming. These two OSs are mainly developed and maintained by Nokia with the help of some other open source communities such as Debian and GNOME. Although these two OSs were initially designed and developed keeping specially Nokia mobile devices in mind, both of them have gained much popularity resulting in some other manufacturers producing Symbian or Maemo based devices. Symbian OS is a very organized and subtle software system. Symbian uses C++ as the native programming language for developing applications. It is a very smartly built software environment which allows the developers to produce flexible, efficient and high performance software products for mobile/handheld devices. Initially mobile devices had very little graphical support with the inputs mainly controlled by keypads. Symbian OS provides a very efficient way of talking to the hardware. It provides both the low-level APIs as well as the high-level APIs for developing technological components for different types of devices. Till now, the most popular Symbian OS phones that have conquered the market are Nokia 80 series, Nokia 60 series and User Interface Quartz (UIQ) series phones . Symbian provides developer tools and free emulators for the developer to design new software products. By maintaining an open source community, it also provides numerous useful and efficient APIs and source codes for mobile application development. Maemo is a similar type of open source project like Nokia and it is also a great initiative taken by Nokia to provide a proficient platform for advanced smart phones like Nokia N810 internet tablet . Like Symbian, Maemo also provides a Software Development Kit (SDK) for developing application on top of the platform. Maemo OS is essentially based and inspired by Linux OS.

This OS not only provides some very useful components to the developers, it also allows the easy integration of components from various online repositories.

Developers can choose the programming languages like C, Java, Python or Rubi for developing applications in Maemo. However, C and Java are the two most popular choices among the Maemo developers. In case of Java development in Maemo, a Java Virtual Machine (JVM) named Jalimo is available. It has a very rich class library. With the support of Debian and GNOME, Maemo offers a scalable, reliable and stable platform for mobile application development.

3.4 Windows Mobile Programming

Windows based mobile devices have gained much popularity among the mobile users and as such windows enabled embedded devices have a good share of the products in the market. Microsoft provides a set of OSs for mobile/embedded device programming known as Windows Embedded. Among others, Windows Embedded CE is arguably the most popular OS available for windows enabled embedded devices. It is also sometimes called as Windows Mobile OS. It was previously known as Pocket PC. Windows CE is a highly customizable real-time operating system specially designed for embedded devices or smart phones that have constrained power and memory. It provides tools and emulators that allow the developers to build custom environment for specific devices. One of the interesting features of Windows CE is that it resembles a small Windows 32-bit OS. As such it supports most of the Win32 APIs and it also supports the x86 and ARM like processor architectures . Microsoft has developed a number of popular mobile OSs based on the Windows CE kernel. Among those, Pocket PC, Windows Mobile and Smartphone are widely used. Microsoft also provides numerous useful tools and plugins for easy integration and development over the Windows CE. Visual Studio 2005 and 2008 support the development of Windows CE applications based on emulators. A .NET framework is available for windows based embedded device programming which is based on the Just-in-Time (JIT) compiler. The .NET framework is equipped with a large library and run-time management system.

3.4.1 Programming Languages for Windows Mobile Programming

There is also a good set of choices for developers when it comes to the selection of suitable programming language for windows enabled embedded/-mobile device programming. Visual C/C++ is often preferred as the programming language in this case because with these languages, it is possible to talk directly with the hardware without any intermediary layers. With Visual C++ it is possible to develop faster, light-weight and flexible mobile applications. Visual C++ can communicate with the Win32 APIs very efficiently and it is much easier to interact with the conventional Windows Desktop APIs using Visual C++. That is why Visual C++ is sometimes referred to as the native language for windows mobile application development. However, when it comes to debugging and error-handling, Visual C++ environment may be much more challenging than other Visual languages like Visual C#. It is also a bit complex to design high level interfaces for mobile application using Visual C++. Visual Basic .NET and Visual C# are also very popular programming languages for mobile application development specially for windows enabled devices. These two programming languages are very much preferred for the new programmers because these are easier to learn. A lot of underlying plumbing is performed by the .NET compact framework, so it is very easy for a developer to build application on top of the framework. It is also very flexible when it comes to designing nice high-level user interfaces. .NET framework greatly simplifies the task of the developer as there are many useful components already available for application development. It is also much easier to convert a standard Visual C# or Visual Basic Code for windows mobiles compared to other programming languages. As these two languages do not talk to the hardware directly but with the help of intermediary layers, they are also sometimes referred as managed languages. Apart from these popular languages Java is also used for windows mobile development. J2ME is a popular language for windows mobile application development for traditional Java developers. For mobile web application development Jscript or ASP.NET is commonly used.

3.5 Android

Android software development is the process by which new applications are created for the Android operating system. Applications are usually developed in the Java programming language using the Android Software Development Kit,

but other development tools are available. As of October 2012, more than 700,000 applications have been developed for Android, with over 25 billion downloads. A June 2011 research indicated that over 67% of mobile developers used the platform, at the time of publication. The Android software development kit (SDK) includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.5.8 or later, Windows XP or later. The officially supported integrated development environment (IDE) is Eclipse using the Android Development Tools (ADT) Plugin, though IntelliJ IDEA IDE (all editions) fully supports Android development out of the box, and NetBeans IDE also supports Android development via a plugin. Additionally, developers may use any text editor to edit Java and XML files, then use command line tools (Java Development Kit and Apache Ant are required) to create, build and debug Android applications as well as control attached Android devices. Enhancements to Android's SDK go hand in hand with the overall Android platform development. The SDK also supports older versions of the Android platform in case developers wish to target their applications at older devices. Development tools are downloadable components, so after one has downloaded the latest version and platform, older platforms and tools can also be downloaded for compatibility testing. Android applications are packaged in .apk format and stored under /data/app folder on the Android OS (the folder is accessible only to the root user for security reasons). APK package contains .dex files (compiled byte code files called Dalvik executables), resource files, etc.

Chapter 4

Our Project

4.1 Memory Lane-Photo Editor

In our project we will be using iOS platform to develop a Photo Editor. In most cases we know that to use an apple certified application we have to go to apple store to purchase an application. And nowadays photography has become very popular among general people where a photo editor is a very essential part.

People use photo editing software to edit images according to their will and share with their friends and family along with updating their daily activities in digital format in their respective social networks. Our goal is to develop such a photo editor that will be an apple certified freeware with all the features of an apple certified Photo Editor along with some additional features. So that general people can use a well developed Photo Editor without spending extra money.

The name of our photo editor application is "Memory Lane". Since people take pictures to keep remembering past memories in form of image where life is a straight lane we walk through and by editing images we keep them how we want to remember our memories.

4.2 Features Of Our Project

4.2.1 Picture Frames

A picture frame is a decorative edging for a picture, such as a painting or photograph, intended to enhance it, make it easier to display, or protect it. Our application will have many built-in picture frames for basic decoration of any image.

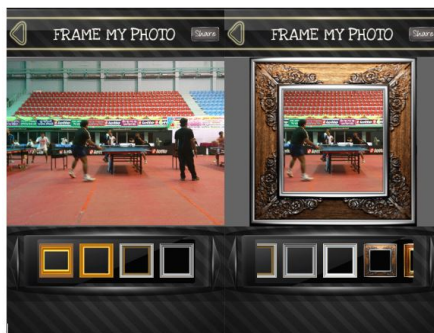


FIGURE 4.1: Photo Frames

4.2.1.1 Picture Posterization

Posterization is a process in photograph development which converts normal photographs into an image consisting of distinct, but flat, areas of different tones or colors. A posterized image often has the same general appearance, but portions of the original image that presented gradual transitions are replaced by abrupt changes in shading and gradation from one area of tone to another. Printing posterization from black and white requires density separations, which one then prints on the same piece of paper to create the whole image. Separations may be made by density or color, using different exposures. Density Separations may be created by printing three prints of the same picture, each at a different exposure time that will be combined for the final image. Typically, posterization is used for tracing contour lines and vectorizing photo-realistic images. This tracing process starts with 1 bit per channel and advances to 4 bits per channel. As the bits per channel increases, the number of levels of lightness a color can display increases. A visual artist, faced with line art that has been damaged through JPEG compression, may consider posterizing the image as a first step to remove artifacts on the edges of the image.

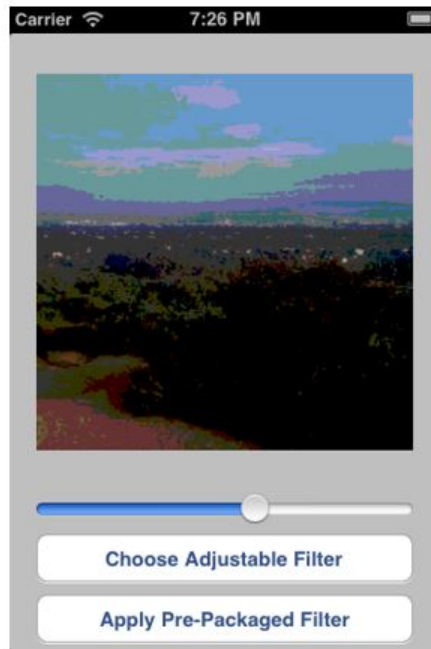


FIGURE 4.2: Posterization

4.2.1.2 Saturation

A measure of the strength of color in a photograph. Our software programs allow us to adjust saturation adding grey to make the image appear more monochrome or removing grey to produce rich, highly saturated colors. Saturation is one of three coordinates in the HSL and HSV color spaces. Note that virtually all computer software implementing these spaces use a very rough approximation to calculate the value they call "saturation", such as the formula described for HSV and this value has little, if anything, to do with the description shown here.

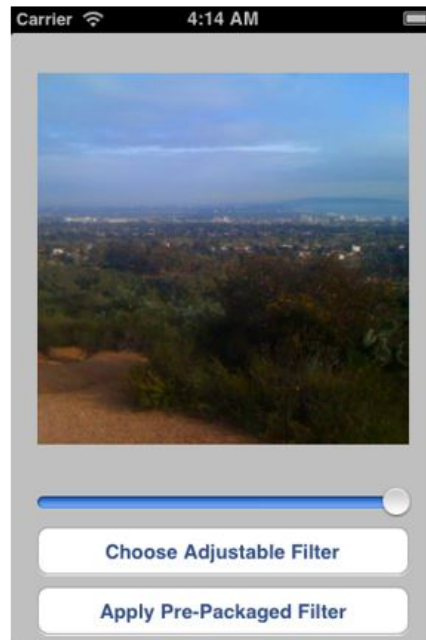


FIGURE 4.3: Saturation

4.2.1.3 Brightness

Brightness is an attribute of visual perception in which a source appears to be radiating or reflecting light. In other words, brightness is the perception elicited by the luminance of a visual target. This is a subjective attribute/property of an object being observed.

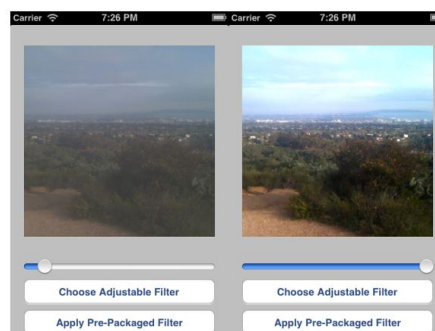


FIGURE 4.4: Brightness

4.2.1.4 Contrast

Contrast is the difference between the lights and darks of an image. The greater the difference between lights and darks, the higher the contrast. Contrast is the

difference in luminance and/or color that makes an object (or its representation in an image or display) distinguishable. In visual perception of the real world, contrast is determined by the difference in the color and brightness of the object and other objects within the same field of view. Because the human visual system is more sensitive to contrast than absolute luminance, we can perceive the world similarly regardless of the huge changes in illumination over the day or from place to place. The maximum contrast of an image is the contrast ratio or dynamic range.

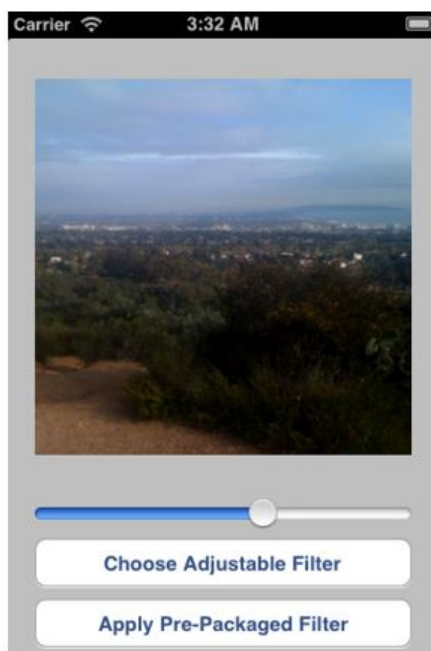


FIGURE 4.5: Contrast

4.2.1.5 Gamma

Gamma correction, gamma nonlinearity, gamma encoding, or often simply gamma, is the name of a nonlinear operation used to code and decode luminance or tristimulus values in video or still image systems. When a photographic film is exposed to light, the result of the exposure can be represented on a graph showing log of exposure on the horizontal axis, and density, or log of transmittance, on the vertical axis. Such curves are known as Hurter-Driffeld curves. Since both axes use logarithmic units, the slope of the linear section of the curve is called the gamma of the film. Negative film typically has a gamma less than 1; positive film (slide film, reversal film) typically has a gamma greater

than 1. Photographic film has a much greater ability to record fine differences in

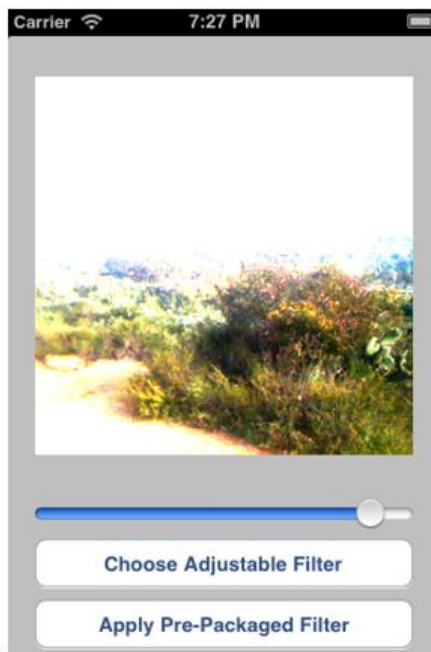


FIGURE 4.6: Gamma

shade than can be reproduced on photographic paper. Similarly, most video screens are not capable of displaying the range of brightnesses (dynamic range) that can be captured by typical electronic cameras. For this reason, considerable artistic effort is invested in choosing the reduced form in which the original image should be presented. The gamma correction, or contrast selection, is part of the photographic repertoire used to adjust the reproduced image. Analogously, digital cameras record light using electronic sensors that usually respond linearly. In the process of rendering linear raw data to conventional RGB data (e.g. for storage into JPEG image format), color space transformations and rendering transformations will be performed. In particular, almost all standard RGB color spaces and file formats use a non-linear encoding (a gamma compression) of the intended intensities of the primary colors of the photographic reproduction; in addition, the intended reproduction is almost always nonlinearly related to the measured scene intensities, via a tone reproduction nonlinearity.

4.2.1.6 Noise

Image noise is random (not present in the object imaged) variation of brightness or color information in images, and is usually an aspect of electronic noise. It can be produced by the sensor and circuitry of a scanner or digital camera. Image noise can also originate in film grain and in the unavoidable shot noise of an ideal photon detector. Image noise is an undesirable by-product of image capture that adds spurious and extraneous information. The original meaning of "noise" was and remains "unwanted signal"; unwanted electrical fluctuations in signals received by AM radios caused audible acoustic noise ("static"). By analogy unwanted electrical fluctuations themselves came to be known as "noise". Image noise is, of course, inaudible. The magnitude of image noise can range from almost imperceptible specks on a digital photograph taken in good light, to optical and radio astronomical images that are almost entirely noise, from which a small amount of information can be derived by sophisticated processing (a noise level that would be totally unacceptable in a photograph since it would be impossible to determine even what the subject was).

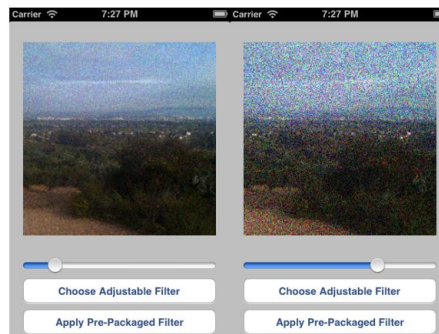


FIGURE 4.7: Noise

4.2.2 Pre-packaged Filter

4.2.2.1 Unsharp masking or Sharpening

Unsharp masking (USM) is an image manipulation technique, often available in digital image processing software. The "unsharp" of the name derives from the fact that the technique uses a blurred, or "unsharp," positive image to create a "mask" of the original image. The unsharped mask is then combined with the negative image, creating an image that is less blurry than the original. The resulting image, although clearer, probably loses accuracy with respect to the image's subject. In the context of signal-processing, an unsharp mask is generally a linear or nonlinear filter that amplifies high-frequency components. For the photographic process, a large-format glass plate negative is contact-copied onto a low contrast film or plate to create a positive image. However, the positive copy is made with the copy material in contact with the back of the original, rather than emulsion-to-emulsion, so it is blurred. After processing this blurred positive is replaced in contact with the back of the original negative. When light is passed through both negative and in-register positive (in an enlarger for example), the positive partially cancels some of the information in the negative. Because the positive has been blurred intentionally, only the low frequency (blurred) information is cancelled. In addition, the mask effectively reduces the dynamic range of the original negative. Thus, if the resulting enlarged image is recorded on contrasty photographic paper, the partial cancellation emphasizes the high frequency (fine detail) information in the original, without loss of highlight or shadow detail. The resulting print appears more acute than one made without the unsharp mask: its acutance is increased. In the photographic procedure, the amount of blurring can be controlled by changing the "softness" or "hardness" (from point source to fully diffuse) of the light source used for the initial unsharp mask exposure, while the strength of the effect can be controlled by changing the contrast and density (i.e., exposure and development) of the unsharp mask. For traditional photography, unsharp masking is usually used on monochrome materials; special panchromatic soft-working black and white films have been available for masking photographic colour transparencies. This has been especially useful to control the density range of a transparency intended for photomechanical reproduction.

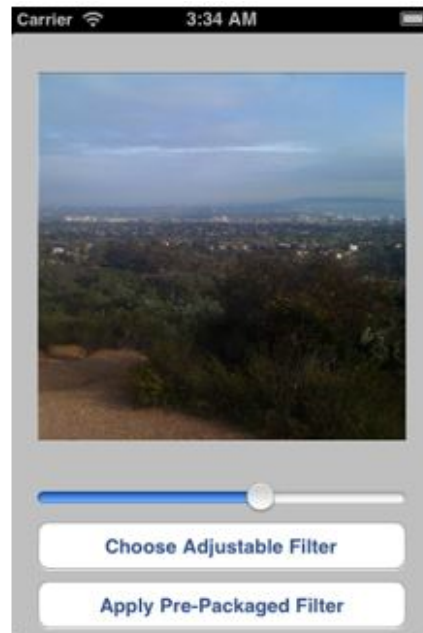


FIGURE 4.8: Sharpening

4.2.2.2 Sepia

In photographic terms, the word Sepia refers to the dark-brown color of a specially prepared photographic print. Sepia tone is a type of monochrome photographic image in which the picture appears in shades of brown as opposed to greyscale as in a black-and-white image. It was originally produced by adding a pigment made from the Sepia cuttlefish to the positive print of a photograph taken with any number of negative processes. The chemical process involved converts any remaining metallic silver to a sulphide which is much more resistant to breakdown over time. This is why many "old time" photographs are sepia toned-those are the ones that have survived until today.

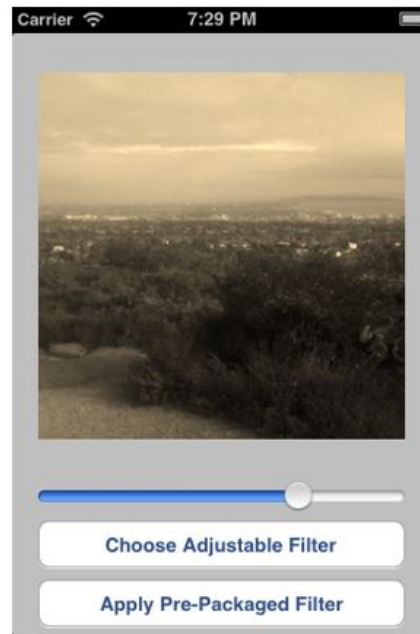


FIGURE 4.9: Sepia

4.2.2.3 LOMO

The "LOMO effect" is named for a plastic camera called the LOMO LC-A Compact Automat camera, originally made by the Russian state's LOMO optical company. The camera was made of plastic and the body was prone to light leakage, so that photos taken with the LOMO LC-A have a distinctive look that came to be known as "lomography." Digital photographers can mimic the look using our photo editing tools. The LOMO LC-A's tendency towards light leaks resulted in distinctive features including "vignetting" or darkening around the edges (which helps frame the shot), high contrast between the light and dark portions of the photo, and high saturation or intensity of colors (although the colors may not be true to life). The effects vary greatly between images taken by real LOMO LC-A cameras, but this is considered part of the charm.

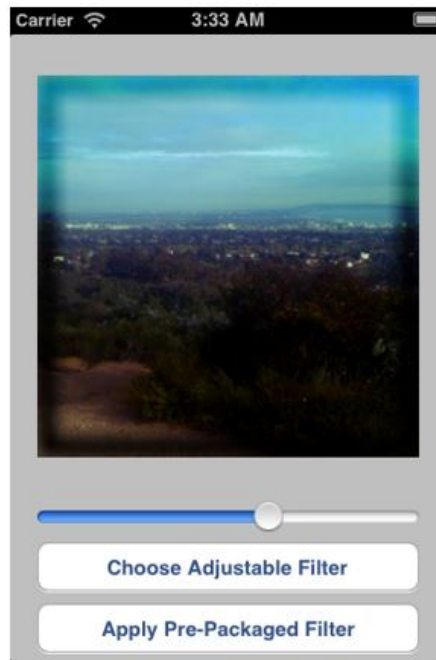


FIGURE 4.10: LOMO

4.2.2.4 Vignetting

In photography and optics, vignetting (French: "vignette") is a reduction of an image's brightness or saturation at the periphery compared to the image center. The word vignette, from the same root as vine, originally referred to a decorative border in a book. Later, the word came to be used for a photographic portrait which is clear in the center, and fades off at the edges. A similar effect occurs when filming projected images or movies off a projection screen. The resulting so-called "hotspot" effect defines a cheap home-movie look where no proper telecine is used. Vignetting is often an unintended and undesired effect caused by camera settings or lens limitations. However, it is sometimes purposely introduced for creative effect, such as to draw attention to the center of the frame. A photographer may deliberately choose a lens which is known to produce vignetting to obtain the effect, or it may be introduced with the use of special filters or post-processing procedures

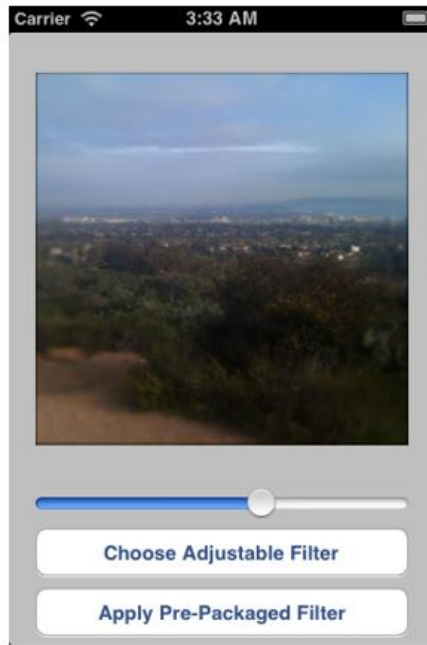


FIGURE 4.11: Vignetting

4.2.2.5 Polaroidish

It is a photographic effect found only in polaroid camera but our photo editing software will add this feature for digital image

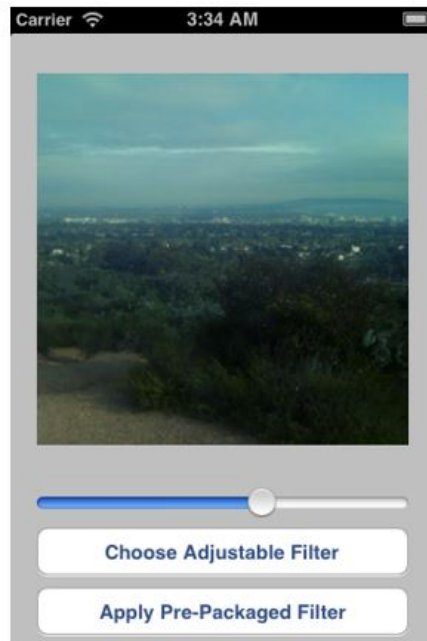


FIGURE 4.12: Polaroidish

4.2.3 Default-packaged Filter

We can also use our by Default-packaged filter. We can select any picture and use any of the 11 default filters.

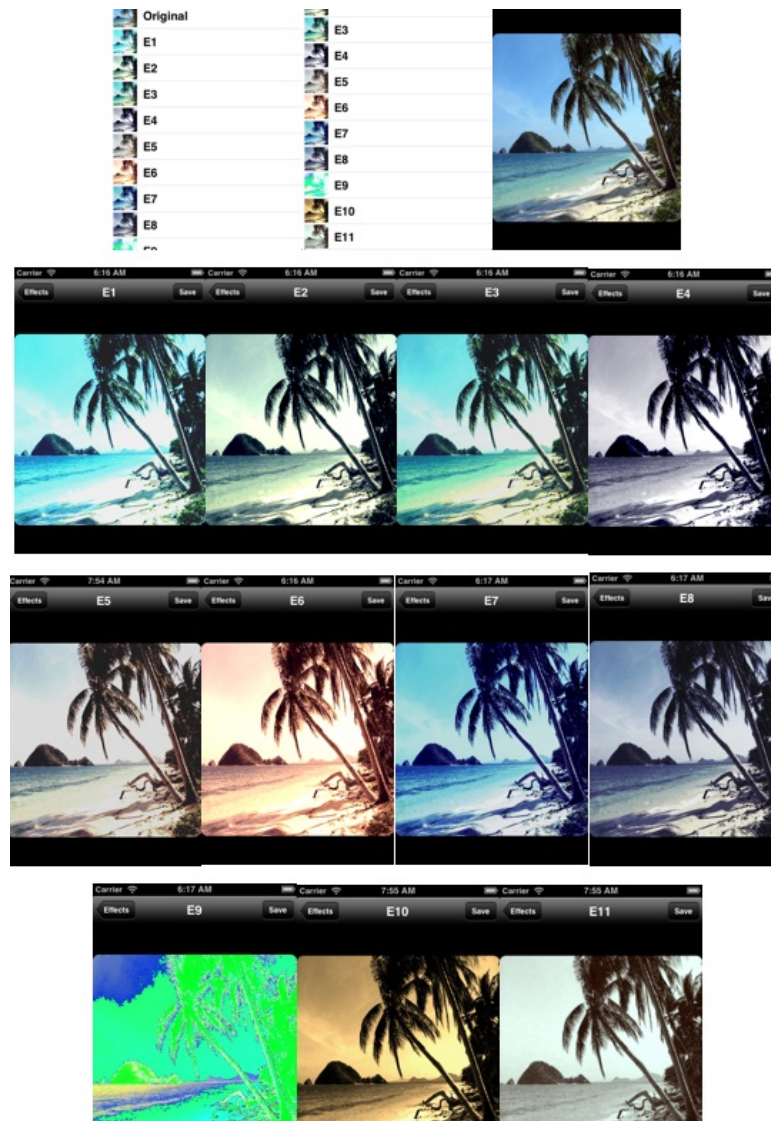


FIGURE 4.13: Default-packaged Filter

4.2.4 Social Networking

We can share our edited photo to our email, Facebook and Twitter and other Social Networks. We can Upload photos to our email or facebook or twitter account by logging in and we can also import photos from the social networks, edit those and reupload too.

4.2.5 Digital Face Editing

We can give pictures a virtual look by using our digital face editor. We can edit our face in the picture by adding digital cap, hat, wigs, mustache, sunglasses, jewelry and beard both for men and women. We can also add some emoticons in a picture. Thus we can give our memories a funny look and share with our friends through our various social networks.



FIGURE 4.14: Digital Face Editor

4.2.6 Transformation Tools

We can use our transformation tools to make gift cards with pictures we want to send to our friends and family.



FIGURE 4.15: Transformation Tools

Chapter 5

How To Use?

5.1 Framing and Uploading

There are 4 types of usage of Memory Lane-Photo Editor:

- Upload Or Take A Photo
- Choose Frame From Gallery
- Share Photo With Everyone
- Edit Pictures



FIGURE 5.1: Framing and Uploading

5.2 Main Menu

Once the application loads we will find the main menu where there will be three main options:

- We can take a photo by using the built-in camera of our smartphone.
- We can import photo from facebook.
- We can also import photos from our phone gallery.



FIGURE 5.2: Main Menu

5.3 Import From Facebook

Once we select "Import From Facebook" option with proper internet connection we will be prompted to the Figure 5.3 window. After logging in to facebook we will be prompted to the Figure 5.4 window. After that we will select our desired photo and then we will select any frame from the frame panel to frame a picture. One thing to note is once we select a image to frame, without framing the picture we cannot save or upload the picture. After framing we can save our picture to phone gallery or we can upload it to facebook, twitter or to our e-mail account.

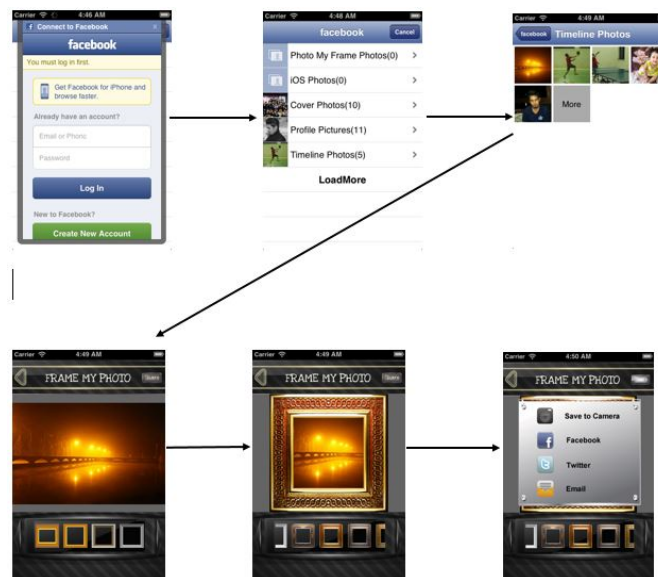


FIGURE 5.3: Import

5.4 Editing A Picture

We can edit our picture using various filters. Some are adjustable filters and some are pre-packaged filter. We can choose a photo from our phone gallery or we can import photos from social networks. Once we select a photo we will be prompted to the Figure 5.9 window. From there we select any adjustable filters and we can also apply the pre-packaged filters over the same picture at the same time.



FIGURE 5.4: Editing

Chapter 6

Scope & Goal

6.1 Scope

”Memory Lane” photo editor is right now under development. Our further development includes the following features: The application is divided into 2 parts

- Framing and uploading
- Picture editing

For further improvement we are going to merge these two parts where we can

- Import a picture from facebook or take picture by camera or import from phone gallery.
- Upload the edited & framed picture to our social networks.
- Additional Filters: We will try to add more filters.

6.2 Goal

iOS is the world’s most advanced mobile operating system. It is also the most popular smartphone technology. Smartphone is like a limb for all kinds of people, a part and parcel of our everyday life. And having the facility of taking picture via smartphone camera, a photo editor application is without a doubt a

necessary application, so that we can edit our memories in image format and share it with our friends & family any where anytime. Our Goal is

- To implement all the necessary features for not only a professional photo editor but also for any novice user.
- To connect the user with all the possible social networks, so that any user can import, upload and share their memories to their respective social networks.
- Since most of the apple product are expensive and we need to buy them from apple store it is sometimes a waste for general people to enjoy this luxury. Our goal is to upload "Memory Lane" photo editor as a freeware so that people from all class can enjoy a well developed photo editor.

Chapter 7

Conclusion

Our main objective was to develop a photo editor which will be available and useful for all kinds of user. Through various implementation process we are trying to develop such a photo editor. The name of the application itself defines a lot about the actual project. We are trying to help people to preserve their memories just the way they want them to be remembered. When they look back at their life, they will see a "Lane Of Memories" just the way they remember.

Chapter 8

Reference

- Book: The Objective-C Programming Language -Apple Developer.
- Video Tutorial: CSE 193P Lecture Videos-Paul Hegarty, Lecturer, Stanford University.
- Video Tutorial: iOS Video Tutorial-Bucky Roberts.
- Afterglow:<http://appfreedownloads.com/afterglow-photo-editor-apps/iphone-ipad-785502875.html>
- Woodcamera: <http://woodcamera.com/>
- Posterization: <http://en.wikipedia.org/wiki/Posterization>
- Brightness: <http://en.wikipedia.org/wiki/Brightness>
- Gamma: http://en.wikipedia.org/wiki/Gamma_correction
- Contrast:http://en.wikipedia.org/wiki/Phase-contrast_imaging
- Noise: http://en.wikipedia.org/wiki/Image_noise