DESIGN AND IMPLEMENTATION OF MULTI-HEAD PRESENTATION SOFTWARE FOR THE iOS PLATFORM

by

Michael S. Babienco

A PROJECT PROPOSAL

Presented to the Faculty of

The School of Computing at the Southern Adventist University

In Partial Fulfilment of Requirements

For the Degree of Master of Science

Major: Computer Science

Under the Supervision of Professor Hall

Collegedale, Tennessee

August, 2014

DESIGN AND IMPLEMENTATION OF MULTI-HEAD PRESENTATION SOFTWARE FOR THE iOS PLATFORM

Michael S. Babienco

Southern Adventist University, 2014

Adviser: Tyson Hall, Ph.D.

ShareSynch, currently available for both Windows and Mac OS X, is a presentation software application tailored towards evangelistic speakers with limited experience. The software has several essential features including the use of speaker notes during a presentation, support for independent slide and speaker note language, speaker note pagination with dynamic font scaling, editing of presentations and rich text speaker notes within the application, and dynamic appeal video configuration. The ShareSynch software does not yet exist on the iOS platform, and no known iOS application contains all of the PC software's essential features. In this paper, we propose the design and implementation of ShareSynch on the iOS platform. The iOS version of ShareSynch, developed in conjunction with ShareHim, will mirror functionality currently available in the PC software and will introduce a new file format and database schema that provides for reduced storage consumption.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

ShareHim's ShareSynch presentation software has been designed to help speakers learn to easily practice and participate in evangelism, both locally and internationally [1]. Although the software has many advantages and unique features when compared to similar presentation software, speakers of ShareSynch sermons can only use ShareSynch on a Windows or Mac OS X computer and cannot present from any other operating system or device due to its proprietary sermon file format. The call for a mobile version of the software has grown in recent years due to the increased popularity and convenience of presenting sermons from mobile devices such as tablets or phones. Creating a mobile version of the ShareSynch software will give speakers the ability to use ShareSynch sermon materials and take advantage of all of ShareSynch's unique features while presenting from a mobile device.

ShareSynch includes several unique features that help novice speakers actively participate in the evangelism process. A set of sermon presentations make up a sermon series, and each series contains several different languages for both sermon slides and speaker notes. The language used for speaker notes does not need to match the language used for sermon slides, and the languages used can be changed from within the application. Speakers can easily adjust the font size of speaker notes during a presentation, and notes are dynamically paginated depending on this font size. When a speaker advances the presentation, the next slide is not shown if there are more pages of notes to show for the current slide. When there is another page of notes to show, the speaker notes are scrolled smoothly downward to the next page. Videos shown for the sermon appeal can change automatically dependent upon the speaker's choice for the appeal type and song without the need for the speaker to edit the overall sermon.

Android and iOS are the two largest mobile application markets. Although Android has a bigger market share when compared to iOS, many Android devices do not support external displays, and more than 50% of currently active Android devices cannot utilize the external display APIs [2, 3, 4, 5]. All iOS products since the release of the iPhone 4 support external displays, and at least 99% of iOS devices can use the external display APIs [6, 7, 8]. We propose the development of an iOS version of ShareSynch that will mirror functionality currently available in the PC version of ShareSynch. The iOS version will incorporate a new file format and storage mechanism that has the potential to greatly reduce consumed storage space for a sermon series.

The rest of this paper discusses background material in Chapter 2, defines the proposed project in Chapter 3, and describes plans to test the application in Chapter 4. Chapter 5 presents a summary of project goals, the proposed solution, and deliverables.

The iOS version of ShareSynch is being developed in conjunction with the Center for Innovation and Research in Computing (CIRC) at Southern Adventist University.

Chapter 2

Background

Emerging research in mobile development is quite disparate, including issues and techniques for mobile application development, the choice of application deployment type and platform, and user expectations. In this chapter, we will discuss research pertinent to the development of the iOS version of ShareSynch and perform a competitive analysis of the available features in currently available mobile presentation applications.

2.1 Mobile Development Issues and Techniques

Mobile software development has several unique challenges when compared with the development of traditional software applications. These include the handling of sensor data, invoking remote services, creating one application on a wide variety of hardware devices, different security concerns, adherence to user interface (UI) guidelines, and power consumption [9]. Developers must create a UI that allows users to perform all functions in a given application, remains easy to utilize by all types of users, and adheres to UI guidelines; they must also make decisions regarding how code or UI differences will be handled between different platforms. More unique challenges for mobile development include universal user interfaces, software reuse across platforms, context-aware applications, and requirements uncertainty in mobile environments [10].

Biel and Gruhn [11] suggest two new design patterns for mobile development. Although the patterns focus on Android devices, the first design pattern, pertaining to the support of a wide variety of hardware screen sizes and resolutions, is still applicable to iOS and the ShareSynch project due to the multitude of screen resolutions and sizes for iOS [12]. In the mobile market, users have a variety of device types, each with their own screen size and resolution. Devices fit into three different pixel density classes: high, medium, or low density classes. An application needs to function properly and look attractive on each class of device as well as on every device within each class. Biel and Gruhn's design pattern suggests that developers design their UI for a specific screen size and resolution in each class, then use a minimal amount of runtime code to scale this general UI to a UI that functions properly and looks acceptable on each type of device within that class. This technique saves storage space and improves user experience across many devices, but can increase media design efforts and programming effort when compared to designing for a single screen size and resolution.

2.2 Mobile Platforms

Developers of mobile applications often must decide between building a fully-native application for each mobile operating system, building a mobile web application that can run on multiple platforms, or using a third-party abstraction tool to transform a single set of code into native code for each device [9]. Although cross platform tools seem promising, a market analysis performed by Smith's Point Analytics [13] states that limitations in these tools, specifically in the realms of performance and device-specific functionality, are causing developers to struggle and reducing efficiency gains normally obtained by using cross platform tools. The market analysis also suggests that although the market for cross platform tools is growing, the rate at which the market is growing is rapidly decreasing. When a single code base is used for a large variety of devices, irregularities often exist due to device-specific conditions [14].

Charland and Leroux [15] investigated whether native applications can really be replaced by web applications. The main benefit of web applications is their ability to run across different platforms, thus saving development costs and reducing development time. Although web application capabilities are increasing, they are still slower than native applications, and it can be difficult to replicate the native user interface in a web application. A user expects a speedy application that has an interface design similar to other native applications, and a web application cannot always meet those needs.

2.3 User Expectations

Customer expectations for mobile applications are reviewed by Haller [16] in an examination of more than 1,000 application reviews in three app stores. Complaints fell into four different categories: functional problems, such as bugs or crashes; technical issues, such as performance and battery consumption; GUI interaction problems, including poor design or missing languages; and business decision complaints or suggestions. Improvement ideas, bugs, and crashes ranked among the top five reasons for poor application reviews. Haller believed that testing procedures were not carried out by many projects, as shown by the multitude of crashes and bugs. Khalid et al. [17] also investigated the reasons for user complaints regarding mobile applications in a study of 6,390 one and two-star reviews. Khalid et al. found that functional errors, feature requests, and application crashes are the reason for more than 50% of complaints, thus supporting the work by Haller [16].

The work on privacy expectations by Jung et al. [18] showed that users do not want mobile applications to gather and share unnecessary or private information. Jung et al. enlisted twenty Android smartphone users and monitored the data that their smartphone applications collected and transferred for three weeks. Participants were interviewed after the three week period to learn about their privacy expectations and reactions to the actual data collected and transferred. Several users expressed surprise at the amount of data that several apps collected, and over 50% of users were concerned by applications that shared location information with third parties. Users expected applications to collect data in a way that coincided with the way the application was used.

2.4 Current Mobile Presentation Applications

A large number of presentation applications are currently available in the iOS App Store. Each of these applications have different features and capabilities. We performed a search on the keyword “presentation” in the iPad iOS App Store on August 7, 2014 and located the top five presentation applications by popularity and by relevance as determined by the App Store. One application, Prezi [19], was listed in both groups. This section organizes these applications according to their features and first reviews full featured applications, then discusses template-based applications, and ends with an examination of the remaining presentation applications.

2.4.1 Full Featured Applications

Keynote [20], created by Apple Inc., is available for free for new iOS owners who have activated their device after August 2013 [21]. Keynote lets users create, edit, and give presentations via an external device or screen. All device orientations are supported, and the application will let the user create duplicates of presentations. Unfortunately, presenter notes are only available in presentation mode when the iOS device is connected to an external display; when the device is not connected to an external display, only the presentation's slides are shown on the speaker's device in presentation mode. Furthermore, a speaker cannot view both notes and a preview of upcoming slides at the same time while in presentation mode.

Microsoft PowerPoint for iPad [22] brings the popular PC program to iPad devices. The application is free to download, but in order to create and edit presentations, the user must have an Office 365 subscription, which requires a monthly or yearly payment. A user can import and give already-created presentations without an Office 365 subscription. When an Office 365 subscription is active, the application has many of the same capabilities as the desktop version of PowerPoint. Speaker notes can be edited in rich text, and these notes can be seen while presenting from the iPad regardless of whether an external display is connected. CloudOn [23] lets users create and edit Microsoft Word, Excel, and PowerPoint documents through the cloud. A user must have internet access in order to use CloudOn. CloudOn mirrors the functionality and feel of the PC Microsoft Office software, including the ability to give and edit PowerPoint presentations. Unfortunately, many of CloudOn's mobile features are behind a paywall, including the ability to present PowerPoint presentations.

WPS Office [24] is a completely-free application that allows users to create and edit Microsoft Word, Excel, and PowerPoint documents. The application runs locally on the device without requiring an internet connection. A small number of rich text capabilities are available when editing slides, but speaker notes can only be written in plain text. Presentation mode shows the current presentation progress, a presentation timer, the current slide, and optionally shows speaker notes and slide preview thumbnails.

2.4.2 Template-based Applications

Flowboard [25] utilizes templates for quick presentation creation. Individual slides are made from a pre-defined or blank layout, and users can easily insert text, pictures, or videos into the slide layouts. Each slide element can link to another presentation slide, allowing users to easily create presentations that quickly jump between slides. Users can pay for extra features, such as the ability to export slides to a PDF document. Flowboard does not offer speaker note functionality.

Haiku Deck [26] and Prezi [19] offer more templates for users to quickly create appealing presentations. Speaker notes are visible in Haiku Deck while in vertical device orientations, and the application lets speakers use iPhones as presentation remotes. Haiku Deck also helps you easily share your presentations with others through a multitude of sharing options. Prezi creates presentations that consist of one giant slide. As a presentation continues, Prezi zooms in and out of the slide in order to give the appearance of having individual slides. Within the provided templates, the Prezi slides are easily configurable in theme and structure. Like

Flowboard, Prezi has no speaker note capabilities.

2.4.3 Other Presentation Applications

SlideShark [27] lets users store their presentations in the cloud for easy access across several iOS devices. SlideShark only allows users to give presentations; no editing capabilities are available within the application. When presenting from SlideShark, animations, fonts, and other media are shown in the same manner as in the desktop version of PowerPoint. Speakers can add annotations to slides while they are being presented. Presentation mode can show speaker notes, a thumbnail list of slides, timers, and the current slide on the iPad version of the application; however, speaker notes can only be seen when presenting from an iPad.

Nearpod [28] is a unique presentation application that is built for the education market. A speaker creates an interactive presentation filled with items such as quizzes or polls. Audience members join the presentation via the internet and their own electronic device and can interact with the presentation when required, such as by answering poll questions. As audience members submit results, the speaker views these results in real-time. Unfortunately, the Nearpod application pushes users to an external web browser for several important tasks, such as presentation creation, and it does not offer speaker note functionality.

None of the applications discussed in this section allow for pagination of speaker notes, changing the speaker note font size while presenting, changing the presentation or speaker note language independently of one another, or hot-swapping of certain slides.

Chapter 3

Proposal

ShareSynch is an extensive presentation software system that is currently available for Windows and Mac OS X systems. The proposed project will take the core features of the PC software and implement them in the iOS environment. The application will allow for presentation of ShareSynch materials on an external display via an Apple TV or a VGA/HDMI cable. Users will be able to edit the sermon slides and speaker notes from within the application. New features that are not currently available in the PC software will be created for this project, including the ability to download sermon series from the internet and a new file format. The new file format makes use of iOS storage policies to allow for easy backups of user data; this format and its corresponding database will be used for storing all sermon series and sermon information, including pictures, videos, and notes. In this chapter, we discuss the software and hardware requirements, proposed approach, task delineation, final deliverables, and development prerequisites for the iOS version of ShareSynch.

3.1 Summary of Requirements

The ShareSynch iOS application will have direct support for the iPad Air, iPad 2, and iPod Touch 5th generation devices. Tangential support will be provided for the iPhone 5, iPad Mini, iPad with Retina Display, and a small number of other iOS devices. Each iOS device must run iOS 7.1 or 8.x, and the device needs to be compatible with an adapter that allows for VGA or HDMI output. The application will support external displays via these adapters or via an AirPlay device, such as an Apple TV.

Upon opening the application for the first time, users will have access to a demo sermon series that contains a single sermon. Users gain access to more sermon series by registering with ShareHim; once registered, users can download new sermon series via resumable downloads over the internet or install series via iTunes application management. A user can have multiple series installed at one time. Before entering presentation mode, users can adjust the slide or speaker note language, generate a PDF of a specific sermon, or create variations of a sermon series.[1]

A speaker can begin presenting a ShareSynch sermon after loading that sermon in presentation mode. Both images and videos can be shown on the external display. The presentation interface shows a scrolling list of slide preview images, current presentation progress in terms of the number of slides presented, speaker notes, and a toolbar that lets a speaker make quick changes to presentation settings. The presentation can be advanced via a Bluetooth remote or keyboard, buttons on the screen, or by tapping in or swiping over the speaker notes. Speakers can quickly jump to any slide in the presentation using the slide preview images.

[1] A sermon series variation is a copy of a given sermon series that contains altered versions of sermon slides or speaker notes.

Speaker notes take up the majority of the presentation mode screen; note font size can be changed from the toolbar with a simple tap of the increase or decrease font buttons. Notes for each slide are paginated, and there is a visual indicator on the screen when there are more notes to display for the current slide. When a speaker attempts to advance the presentation, the application first checks to see whether the last page of notes are being displayed for the current slide. If the last page is being shown, the sermon is moved to the next slide. If there are more notes to show, the speaker notes are smoothly scrolled to the next page, and the next slide is not shown. The external display smoothly animates from one image or video to the next when appropriate.

An appeal may be given at the end of a sermon. Three appeal types exist: vocal, instrument only, or text only. Vocal and instrumental appeals project videos, and a text-only appeal projects a single image. As the presenter advances the speaker notes for an appeal, text is displayed on top of the projected video or image in order to communicate with the congregation. The appeal song selection and type can be changed from the presentation mode toolbar. When a user changes the appeal song selection or type, the application automatically swaps the appropriate slides into and out of the presentation; the user does not need to enter the editing interface to manually perform these changes.

The user can access the sermon editing capabilities from the presentation interface for that sermon. Speaker notes can be edited in rich text with the following options: bold, italic, underline, strikethrough, and bulleted lists. Slides can be hidden or rearranged, and new slides can be created from pictures or videos on the user's device. Slides can also be duplicated or imported into the current sermon from other sermons within the same sermon series. Changes to the sermon are saved automatically, and undo/redo buttons are available for the user should they need them.

Details on all ShareSynch features and requirements appear in the ShareSynch Requirements Specification located in Appendix A.

3.2 Proposed Approach

The ShareSynch iOS application will be implemented as a native application in Objective-C in order to take advantage of all available device functionality; the application will be built for both the iPod Touch and iPad platforms. When applicable, we will be following the Model-View-Controller design pattern [29] and will make substantial use of the Delegation design pattern [30] for communication between different objects. We will also utilize the recommendations set forth in the iOS App Programming Guide [31] for general guidelines and principles to use while developing the application. The application will not share any user information with third parties, will not gather personal user information, and will only perform data gathering on the length of time that sermons are presented.

Files for the application will be stored in two different locations. All pictures, videos, and other pertinent non-user data will be stored in the Application Support directory. This directory is not directly accessible to users and will thus prevent users from easily accessing any ShareSynch pictures or video. Keeping large files in a separate directory will also allow us to add the com.apple.MobileBackup attribute to the folder, which prevents the backup of these large files through iCloud or iTunes [32]. Pictures, videos, and other sermon or series content are stored in large archive files for easy access; archive files also allow for file compression, which will save valuable storage space. These archive files will be protected by passwords, which will also help to protect proprietary images or video. The SQLite database that houses series and sermon information will be stored in the application's Documents folder. The Documents folder is accessible to the user from iTunes by default; making the database file accessible to the user will let the user manually backup or restore their sermon settings and notes. When sermon series files are copied into the application's Documents directory via iTunes and subsequently detected by the application, the application will move these data files into the Application Support directory, which will then let the user present the sermon series from the application.

A unique feature of the iOS version of ShareSynch is the ability to dynamically draw text on top of images. The same image can be used for the background of a large number of slides, and each of these slides can have text drawn on top of its image. This feature will allow ShareSynch series creators to design one slide image to be used across a large number of slides and then designate text and text attributes for that slide separately. Text in a specific language is drawn onto the image at runtime. Text attributes include position, width, height, font name, font size, and font color. If this capability is utilized, the application can use one image for many different slides and languages, which drastically reduces the sermon storage cost when compared to storing a unique image for every slide in every language.

The ShareSynch application utilizes an SQLite [33] database, and the proposed database schema can be seen in Figure 3.1. The database stores all information on each series and sermon as well as speaker notes and sermon slide data.

We will utilize a large assortment of open-source libraries to help speed up the development of this project. Some examples of open-source projects we will use include:

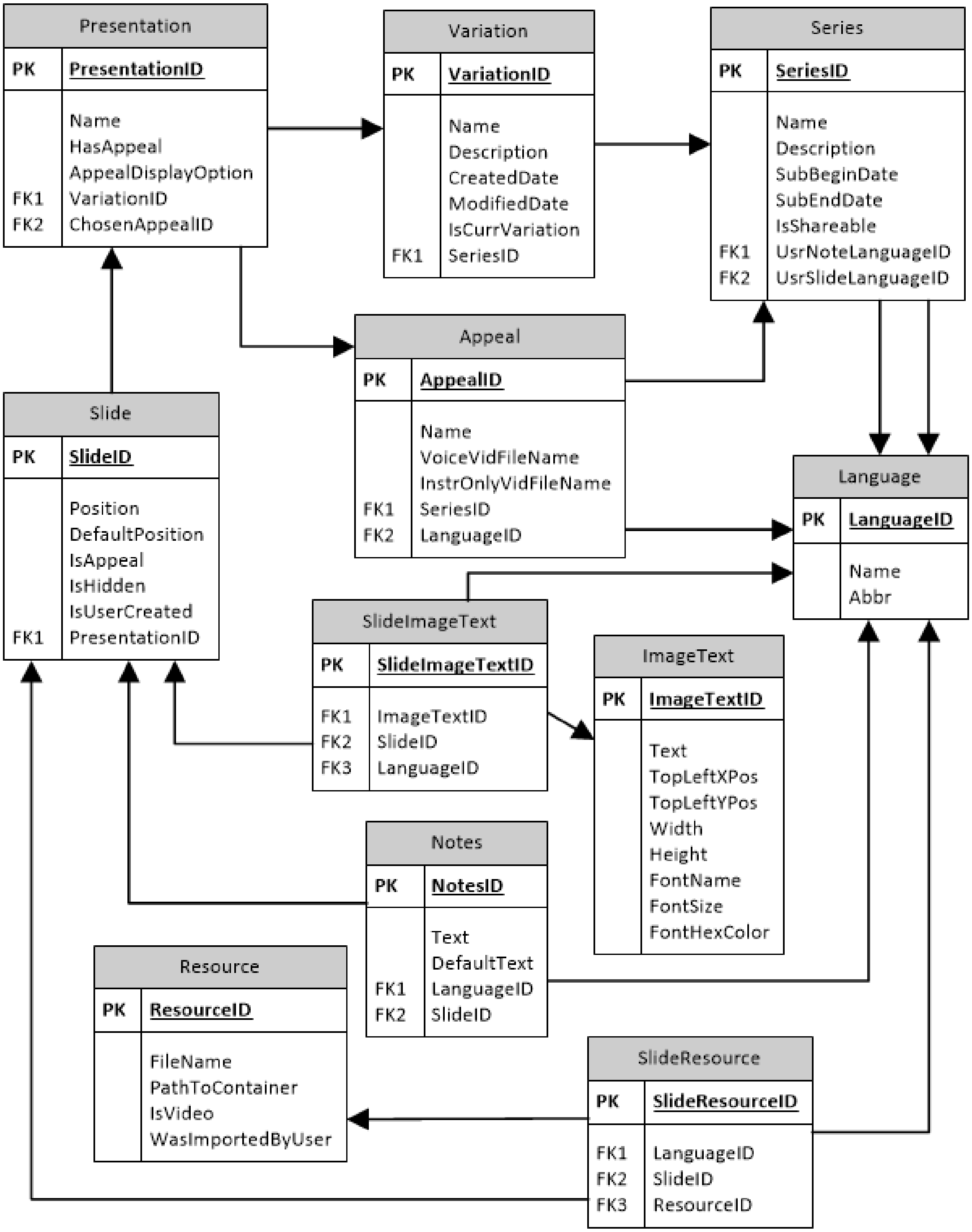

Figure 3.1: The proposed schema for the ShareSynch iOS application allows for the storage of all sermon series information and user data.

- iCarousel [34] — allows for a scrolling thumbnail interface used on the presentation screen for navigation
- The iOS Rich Text Editor with lists, available at [35] — takes care of the rich text editing for slide notes
- KSLabel [36] — eases the process of drawing borders around label text
- ASIHTTPRequest [37] — allows for asynchronous web requests and resumable downloads

3.3 Task Delineation

The ShareSynch project has been broken into seven major task areas for development and testing. A summary of major tasks, including estimated work hours for each task, can be seen in Table 3.1. Most requirements in the requirements document have been assigned a task number corresponding to the tasks listed in this table. Approximately one hundred hours of work towards the completion of these tasks were completed by August 22, 2014. During the Fall 2014 semester, work will progress at the rate of approximately twenty hours per week. This section details items completed in each task along with the features available for users when a given task is completed.

3.3.1 Presentation Mode

The first task of development will be the creation of the speaker's presentation view. This portion of the application lets the speaker project slides and video onto an external monitor while viewing notes and slide thumbnails on the iOS device. This task includes initial project and source control setup, displaying and scrolling

Table 3.1: Task Delineation for the Development of the ShareSynch iOS Application

Task	**Estimated Work Hours**	**Estimated Date of Completion**
Presentation Mode	100	August 22, 2014
File and Database Implementation	70	September 15, 2014
Sermon Editing System & Settings	160	November 10, 2014
Sermon Slideshow Installation	80	December 08, 2014
UI Design and Implementation	60	December 29, 2014
Beta and User Testing	60	January 19, 2015
Final Changes, Bug Fixes, Testing, and Documentation	40	February 2, 2015
Project Completion	**570**	**February 2, 2015**

speaker notes and slides, overlaying text on an image, and playing an appeal video with text on top of the video.

3.3.2 File and Database Implementation

The second task includes the creation of the application database and the implementation of file formats, file storage, and file manipulation. Sermons, slides, variations, and all other data will be loaded as appropriate from data files and the database when this task is complete. The user will not see many obvious changes after this task is complete because most changes occur in the application back end. Because the final file and database implementation details were not finalized before project coding began, this task is placed after presentation mode implementation.

3.3.3 Sermon Editing System & Settings

The sermon editing system is the most intricate part of the ShareSynch system. The user will be able to edit slide notes in rich text; move slides from one position to another; and add, hide, or remove slides from a given sermon. The speaker can undo or redo their changes to the sermon. PDF documents of a sermon can be generated and printed; these PDFs contain the speaker's own notes along with their corresponding slides. The settings page for the application will also be created during the course of this task.

3.3.4 Sermon Slideshow Installation

When this task is complete, users will be able to download a sermon from ShareHim servers or install it via iTunes after registering their device with ShareHim. Wireless downloads will be resumable, and a sermon series will not be accessible until the entire series has been downloaded. A demo sermon is available for users who have not yet registered their device with ShareHim.

3.3.5 UI Design and Implementation

Because the application is being created for both iPod Touch and iPad devices, a task must be specifically devoted to the creation and implementation of graphical user interfaces that function properly on the widely varying screen sizes. User interfaces will be designed according to the interface guidelines laid out by Apple for iOS applications [38]. Two main interfaces will be created: one for the larger iPad screen size and density and a second for the significantly smaller iPod Touch. The application will manipulate these interfaces at runtime to adjust for differing screen sizes and densities. Once the UI design and implementation task is complete, users will be able to move through the feature-complete ShareSynch application via an intuitive and clean interface on all directly supported device types.

3.3.6 Beta and User Testing

The application now undergoes thorough testing and evaluation. All features are tested; optimizations and bug fixes are made. Along with GUI and feature testing, this testing phase also includes device testing, battery-consumption testing, and user testing.

3.3.7 Final Changes, Bug Fixes, Testing, and Documentation

This final task envelopes the final details and work of the project. Documentation is created as necessary, any last-minute bugs that have been found are fixed, and more testing is completed. When this task is complete, the application is ready for use by the public and can be submitted to the iOS App Store.

3.4 Final Deliverables

Once the project has been completed, we will deliver the following materials to ShareHim:

- Application source code
- Administrative access to project source control
- Documentation on the general structure of the application code
- File format documentation
- Final project report

3.5 Development Prerequisites

In order to develop and test the ShareSynch software, we will require the hardware and software listed in Table 3.2. Git [39] and Bitbucket [40] will be utilized for source control. As shown in the table, no additional resources will need to be purchased or provided by the School of Computing for this project.

Table 3.2: ShareSynch Development Prerequisites

Prerequisite	**Provided By**
Mac Mini	CIRC
iPad Air	CIRC
iPad 2	ShareHim
iPod Touch 5th Generation	ShareHim
Apple TV	Dr. John Beckett
Bluetooth keyboard	CIRC
Bluetooth remote	CIRC
Xcode	Free in Mac App Store
Git & Bitbucket	Free

Chapter 4

Testing & Evaluation Plan

A large variety of test types will be executed during the course of ShareSynch's development. Xcode's XCTest framework, the iOS emulator, and physical devices will be used to perform unit tests, battery usage analysis, and device testing. Other test types will be executed in Instruments, a program included in Xcode for performance and graphical testing [41].

Performance tests in Instruments let the developer see exactly how much time is spent in a given function in comparison to all other functions that are called during the course of an application's usage. ShareSynch has several areas where the application must perform tasks within a given time frame so that the application does not feel sluggish to the user. Table 4.1 shows these areas and their maximum time for completion. If the application cannot perform some task in the given time frame, the application must lazily perform this task or execute it in the background so that the user can continue to utilize the application while lengthy tasks are running. For example, if the entire sermon cannot be loaded within ten seconds, the application will show the available content after ten seconds and continue to load the rest of the sermon in the background as quickly as possible.

Table 4.1: ShareSynch Performance Testing

Item	Maximum Time for Completion
Saving Changes to Sermon Order or Slide Visibility	250 milliseconds
Saving Note Edits	250 milliseconds
Application Boot	2 seconds
Loading Sermon Series Details	2 seconds
Saving New Slides	2 seconds
Creating a Variation	5 seconds
Restoring a Sermon	7 seconds
Loading Sermon	10 seconds

One of ShareSynch's requirements states that it must be able to present multiple hour-long sermons on a single battery charge. The application will be tested for battery usage while in presentation mode. Each device that is supported by ShareSynch will be used for two hours at full, half, and minimum brightness levels. Battery tests will pass if there is at least 15% battery left after two hours of consistent application usage.

Memory leaks can be discovered while an active application is being surveyed by Instruments. Once a memory leak has been detected, Instruments tells the developer which function the memory leak occurred in or where the creation of the leaked object took place. We aim to have no memory leaks in the application code that we create. Memory leaks may occur within Apple's framework code, but because this code is out of our control, we will not be able to prevent those memory leaks. ShareSynch must also not crash. In order to facilitate the random inputs and events that may lead to ShareSynch crashing, we will utilize the UI AutoMonkey framework [42]. The UI AutoMonkey framework gives an iOS application rapid, random events and inputs in order to test for application stability.

Unit tests will focus on data model functionality. These tests will ensure that application data is loaded, used, and stored properly during normal application function calls. The Tuneup JS framework [43] will be used for the graphical side of testing. Tuneup JS simplifies the creation and execution of GUI tests within Instruments; it also allows for graphical comparisons between what is currently shown on the screen and screenshots depicting what the screen should actually show.

Along with the aforementioned test types, a mobile application must be tested in its usage environment and on all supported device types [16]. Device testing will ensure that the application runs efficiently on each device type and will also ensure that the GUI scales as appropriate on each supported screen resolution. Device testing will not take place on tangentially supported devices. User and acceptance testing will be performed after all features have been fully implemented in the application. These final two types of testing shall be performed by ShareHim and will verify that the application meets not only its listed requirements but also its functional use by speakers. If ShareSynch passes all aforementioned tests and is found to be acceptable to ShareHim, the project should be deemed a success.

Chapter 5

Conclusion

ShareSynch is a proprietary presentation application currently available for Windows and Mac OS X systems. Unique features of ShareSynch include the ability to have different languages for sermon slides and speaker notes, pagination of speaker notes in various font sizes, and the easy adjustment of appeal selections. No currently known iOS application mirrors these unique and essential features. The proposed iOS version of ShareSynch will be developed in Objective-C and will be compatible with the iPad and the iPod Touch with tangential support for several related devices; the iOS application will also introduce a new file format and storage scheme that allows for the ability to draw text onto an image so that one image can be used for many different slides. We aim to have the application fully developed and tested by February 2, 2015, and we will write the final project report during the Winter 2015 semester. Source code, documentation, and the final project report will be delivered to ShareHim upon completion of this project.

Appendix A

Requirements Specification

A.1 Task Delineation

Most of the requirements in Section A.3 have been assigned to one of the following task areas for organizational purposes. The applicable task area is found in brackets before the requirement is given.

1. Presentation Mode
2. File and Database Implementation
3. Sermon Editing System & Settings
4. Sermon Slideshow Installation
5. UI Design and Implementation
6. Beta and User Testing
7. Final Changes, Bug Fixes, Testing, and Documentation

A.2 Hardware Requirements

- The application shall have direct support for iPod Touch 5th Generation, iPad 2, and iPad Air with tangential support for iPhone 5, iPhone 5S, iPhone 5C, iPad Mini, iPad Mini with Retina Display, iPad with Retina Display.
- The iOS device shall run iOS 7.1 or 8.x.
- The iOS device shall be compatible with an adapter that allows for VGA or HDMI output.

A.3 Application Requirements

A.3.1 General

- [1] Upon opening a sermon in presentation mode, the application shall ask users to turn off battery-consuming services when the network can be reached.
- [1] Upon opening a sermon in presentation mode, the application shall ask users to turn off battery-consuming services when the network can be reached.
- [1] The application shall automatically remove the warning to turn off battery-consuming services after 5 seconds.
- [5] The application shall support both landscape and portrait layouts in all modes.
- [2] The application shall support separate different languages for application text, sermon notes, and presentation slide content.

- [2] The application shall support JPEG or PNG formats for presentation slide images.
- [1] The application shall support presentation slides that are text overlays on images. The application shall support overlaying text of a given font, font size, and font color onto an image at a given starting point, width, and height for demoing purposes.
- [2] If an image or video uses the overlay text feature, the image or video shall not be tied to a specific language so that multiple languages can use the same image or video.
- [4] Any sermon series downloaded or installed shall be verified for integrity via an accompanying checksum or hash.
- [4] The application shall check for newly downloaded (via iTunes or WiFi) sermon series upon launch and install all new series upon launch.
- [6] The application shall support the presentation of multiple hour-long sermons on a single battery charge when battery-consuming services (such as WiFi and cellular data) are turned off.
- [1] The application shall be compatible with a Bluetooth remote for advancing slides.
- [2] The application shall be compatible with multiple sermon series.
- [1] The application shall allow for presentations via an external video adapter and through AirPlay.
- [4] The application shall verify that the users subscriptions are valid upon opening the application.

- [1] When transitioning between slides, the application shall smoothly fade from one slide to the next during the transition.
- [1] By default, the application shall mirror the users screen when connected to an external screen.
- [1] The application shall start a timer when the user enters presentation mode.
- [1] When the user leaves presentation mode, the application shall end the timer. If the timer length is 20 minutes or longer, the presentation time length and name of the presented sermon shall be saved in a JSON format.
- [4] The presentation time length along with the presented sermon name shall be uploaded to ShareHim servers once every seven days.

A.3.2 Download & Installation

- The user shall be able to download the ShareSynch software for free from the iTunes app store.
- [4] When opening the application for the first time, the application shall prompt the user for their activation code or allow the user to use the demo sermon series.
- [4] A successful activation code entry shall result in access to a specific set of sermon series that the user can now download or install.
- [4] When downloaded from the app store, the app shall contain a single sermon series entitled Demo that contains a single sermon for demo purposes.

- [4] The demo sermon series shall not be available after the user has downloaded a complete sermon series from ShareHim.

- [4] The application shall verify the users information with the ShareHim website before granting access to non-demo sermon series downloads when an internet connection is available.

- [4] If an internet connection is not available, the application shall prompt the user for an activation code, which can be manually obtained from ShareHim via phone or the internet. Any information that the user will need to give ShareHim for activation shall be displayed along with this prompt.

- [4] The user shall have access to the Download Sermon Series page, which contains a list of ShareHim sermon series available to subscribe to. Series not currently available for the current user shall be grayed out.

- [4] The user shall be able to sort the list on the Download Sermon Series page by name or by available series.

- [2] Slides, pictures, audio, and video shall not be accessible from outside the iOS application. These resources shall be encapsulated in large archive files that are password protected so that they are not easily accessible from outside the application.

- [4] The user shall be able to install new sermon series through iTunes application management or wirelessly over WiFi and cellular data.

- [4] The user shall see the size of a sermon series before downloading it over WiFi and cellular data.

- [4] The user shall be able to view the amount of available space left in the application for sermon series from the Download Sermon Series page.
- [4] The application shall give a warning if there is not enough space on the device or in the application to download a given sermon series.
- [4] The user shall be able to start, pause, and resume the download of a sermon series at any time.
- [4] The user shall see the progress, speed, and estimated download completion time of the sermon series download on the Download Sermon Series page.
- [4] The user shall not be able to access any sermon of a series until the entire series has been downloaded and installed.

A.3.3 Opening and Managing a Sermon Series

- [2] Upon opening the app, the user shall be presented with the following options on the main menu: Load Last Used Sermon, Load Sermon, Download Sermon Series, and Settings.
- [2] The user shall be returned to slide 1 of their last-used sermon upon choosing Load Last Used Sermon on the main menu.
- [4] Selecting the Download Sermon option shall take the user to the Download Sermon Series page.
- [2] Upon selecting Load Sermon from the main menu, the user shall be provided with a list of available sermon series to choose from.

- [2] When the user chooses a sermon series, the user shall be taken to a new view that contains thumbnail previews of all sermons within that series. This view is the sermon selection screen.

- [2] From the sermon selection screen, the user shall see data on the selected sermon series (slide language, note language, etc.), picture thumbnails of each sermon in the series, and a toolbar at the bottom of the screen.

- [2] From the sermon selection screen toolbar, the user shall have the option to change the series language for slides.

- [2] From the sermon selection screen toolbar, the user shall have the option to change the series language for notes.

- [2] From the sermon selection screen toolbar, the user shall be able to enter variation management mode

- [2] The variation management mode shall show a list of all sermon variations for a given sermon series.

- [2] In variation management mode, the user shall be able to create a new variation based on the default, unedited variation.

- [2] In variation management mode, the user shall be able to create a new variation based on another variation.

- [2] In variation management mode, the user shall be able to delete variations of a sermon series, but shall not be able to delete the original, unedited variation.

- [2] In variation management mode, the user shall be able to switch the currently loaded sermon series to a different variation of that series.

- [3] From the sermon selection screen toolbar, the user shall be able to generate a PDF of a sermon. This PDF shall contain all user notes (as edited by the user) along with a small picture of each slide.
- [3] The user shall be able to print or email the generated PDF.
- [2] From the sermon selection screen, the user shall be able to choose a sermon in order to enter presentation mode.

A.3.4 Presentation Mode

- [2] The user shall see a loading progress indicator dialog when loading a sermon will take longer than half a second.
- [2] The user shall be able to cancel loading a sermon via the progress indicator dialog.
- [1] The application shall alert the user if there is no secondary display available or if a secondary display is disconnected while in presentation mode.
- [1] The speaker shall have button controls to control whether the external screen mirrors the current display or projects the sermon slides and videos.
- [1] The user shall see a thumbnail preview of the previous, current, and next slides along with the current slides notes.
- [1] The thumbnail previews of slides shall show that slides position in the overall sermon.
- [1] The user shall be able to move forward (advance) and backward (go back) by swiping left or right, by touching the right or left sides of the screen, by

swiping right or left, or by tapping the next/previous buttons, as defined by a user-specified setting for advancing a sermon.

- [1] When sermon notes for a single slide do not fit on a single page during a presentation, the user shall see a visual indicator when there are more notes to display for the current slide.
- [1] When sermon notes for a single slide do not fit on a single page during a presentation, advancing will scroll the sermon notes for a single slide smoothly without moving the presentation to the next slide. If there are no more notes to show, advancing shall move the presentation to the next slide.
- [1] The user shall be able to skip to any slide by scrolling the thumbnail previews to the right (forward) or to the left (backward) and selecting a slide by tapping on its thumbnail.
- [1] When scrolling the thumbnail previews, the previews shall not wrap from end to beginning or vice versa.
- [1] The user shall be able to tap the thumbnail of the next or previous slide in order to instantly jump to the next/previous slide without smooth scrolling of notes.
- [1] The user shall be taken to the first slide after advancing on the final slide.
- [1] The user shall be taken to the last slide after going back on the first slide.
- [1] If the user is using an external keyboard or has a Bluetooth remote with number capabilities, the user shall be taken to a specific slide number after typing in the slide number and hitting Enter (↩).

- [1] If the user is using an external keyboard or has a Bluetooth remote with number capabilities, hitting the left and right arrow keys will move the sermon backward or forward, respectively.
- [1] The user shall see the number of the currently active slide in the current sermon along with the total number of slides in the current sermon in the following format: [# of current slide]/[total # of slides].
- [1] The user shall see a toolbar at the bottom of the screen.
- [1] The user shall be able to increase or decrease the font size of sermon notes from the toolbar.
- [3] The application shall remember any indentation as it increases font, decreases font, and performs line wrapping.
- [1] The user shall be able to change the appeal song for a given sermon from the toolbar.
- [1] The user shall be able to change the appeal options for a given sermon from the toolbar to the following options: vocal, instrument only, text only.
- [1] Changes to the appeal song or appeal song options shall not come into effect while the sermon is displaying the appeal. Changes shall come into effect the next time the appeal slide is shown.
- [1] In the preview of a video slide, the user shall see the word Video along with the length of the video in text on top of the preview for the video.
- [1] When a video is playing, the user shall see the length of time remaining for the video playback only on the users device.

- [1] Some slides may be appeal slides. An appeal slide shall have text dynamically written onto the slide that mirrors portions of the speakers notes. When the user advances the appeal notes, if there are more notes to display, both the notes on the users device and the text on the external display changes.

A.3.5 Edit Mode

- [3] The user shall be able to open sermons in edit mode from presentation mode.

- [3] The user shall be able to change slide notes, slide order, and which slides to display during a presentation in edit mode.

- [3] When editing a slides notes, the user shall have the following formatting options: bold, italic, underline, and bullets.

- [3] The user shall be able to use the rearrange slides mode while in edit mode.

- [3] In rearrange slides mode, the user shall be able to rearrange the slides of a sermon by dragging and dropping slides from one location to another while viewing thumbnail previews of all slides in a sermon.

- [3] The user shall see the numerical location of each slide in the lower right-hand corner of each slide preview.

- [3] The user shall be able to select one or more slides at one time to set them as hidden or shown when rearranging slides.

- [3] The user shall be able to insert new slides containing pictures or video from their local iOS device into the current sermon.

- [3] The user shall be able to insert new slides containing pictures, video, or slides from another sermon into the current sermon.

- [3] The user shall be able to duplicate a slide, including notes, at the current slide position.

- [3] The user shall be able to delete slides, including their accompanying resource, if the slide was created by the user and the resource was imported into the application by the user. If the resource was not imported by the user, only the slide shall be deleted.

- [3] When inserting new slides, the user shall be able to insert new slides before or after the current slide.

- [3] Changes to sermon order or contents shall be saved automatically upon the change being made.

- [3] Changes to sermon notes shall be saved automatically upon changing to a different slide in edit mode, when the user exits edit mode, or when the user leaves the app for any reason (such as via the home button).

- [3] The user shall see both the current slide and its notes while editing a slides notes.

- [3] The user shall have undo and redo buttons for the following change types: change of note text, hiding or showing a slide, changing a slides position, and adding a slide.

- [3] The maximum amount of times the user shall be able to undo or redo shall be 10 times.

- [3] The undo and redo information shall be lost when the user leaves edit mode.
- [3] The user shall be able to restore a slides notes to its initial contents.
- [3] The user shall be able to reset an entire sermon to its initial state. This operation shall be confirmed with the user before being performed. Resetting a sermon restores all slides to their default position, notes, and visibility. Resetting a sermon also deletes any custom slides created by the user.
- [3] The user shall be able to edit the text shown on top of appeal videos. This is done via the normal editing interface for notes. The notes on top of an appeal video may advance along with the speakers notes. The format for appeal text shall mirror the PC version of ShareSynch. Notes to be shown on top of the appeal video shall be inside a <C><c> tag. Advancing the notes on an appeal slide shall advance the text displayed on the external screen to the text within the next <C><c> tag, if available. A single <R> tag shall designate that the text on the external screen shall be removed, and any text after the <R> tag will be displayed as the final notes for the speaker.

A.3.6 Settings

- [3] The user shall be able to view and edit applications settings after selecting Settings on the main menu.
- [3] The user shall be able to view their series subscription details and activation code.
- [3] The user shall be able to view information about the mobile application and its creators.

- [3] The user shall be able to view information on how much storage space the application is using.
- [3] The user shall be able to permanently delete sermon series from their device.
- [3] The user shall be able to change their default method of advancing sermon notes (swipe, previous/next buttons, or touching the left/right side of the screen).

A.4 Optional Requirements

- Users should be able to import the following formats into the application in order to create new sermons and sermon series via email or iTunes: ZIP full of PNG/JPEG, PDF, PPT, and PPTX.
- The user should be able to see the current sermon length in time while presenting.
- The user should be able to reset the current sermon timer by tapping and holding on the sermon timer.
- The user should be able to pause/resume the current sermon timer by tapping the timer.
- Users should have additional note formatting options including: font type, font color, and bullet type.
- Stylized text for slide images in all languages should be generated for slides upon loading of the sermon.

- The user should be able to change which items are synced to their iCloud account (sync notes, pictures, and/or videos).

- The user should be able to automatically sync their changes to a sermon series or sermon to iCloud in order to have consistent edits across all of their devices.

- Users should be able to enable remote coaching from Settings.

- If remote coaching is enabled, the application should be capable of recording videos, uploading videos, and chatting with coaches in the same manner as the ShareHim Coaching Android application.

- The user should be able to export or import series variations from another user via email or iTunes application management if sharing is enabled for a given sermon series. Sending and receiving a series variation should not include slides created by the user with custom pictures and video.

- The application should backup current sermon notes, order, and contents by saving it as a sermon variation before importing changes from another user.

- Once the user has left edit mode, any changes to the sermons text (notes, appeal text, etc.) made to the sermon should be uploaded to ShareHim servers for viewing by an Administrator.

Bibliography

[1] "ShareHim programs," http://sharehim.org/about-us/, ShareHim, 2014. 1

[2] "Worldwide smartphone shipments edge past 300 million units in the second quarter; android and ios devices account for 96% of the global market, according to idc," http://www.idc.com/getdoc.jsp?containerId=prUS25037214, IDC Corporate USA, 2014. 1

[3] "What qualifies an android "display"?" http://stackoverflow.com/questions/16745935/what-qualifies-an-android-display, Stack Overflow, 2013. 1

[4] "Dashboards," https://developer.android.com/about/dashboards/index.html, Google Inc., 2014. 1

[5] "DisplayManager," http://developer.android.com/reference/android/hardware/display/DisplayManager.html, Google Inc., 2014. 1

[6] "iOS: About apple digital av adapters," http://support.apple.com/kb/HT4108, Apple Inc., 2014. 1

[7] "App store distribution," https://developer.apple.com/support/appstore/, Apple Inc., 2014. 1

[8] "Multiple display programming guide for iOS," https://developer.apple.com/Library/ios/documentation/WindowsViews/Conceptual/WindowAn

dScreenGuide/UsingExternalDisplay/UsingExternalDisplay.html, Apple Inc., 2012. 1

[9] A. I. Wasserman, “Software engineering issues for mobile application development,” in *Proceedings of the FSE/SDP workshop on Future of software engineering research*. ACM, 2010, pp. 397–400. 2.1, 2.2

[10] J. Dehlinger and J. Dixon, “Mobile application software engineering: Challenges and research directions,” in *Workshop on Mobile Software Engineering*, 2011. 2.1

[11] B. Biel and V. Gruhn, “Usability-improving mobile application development patterns,” in *Proceedings of the 15th European Conference on Pattern Languages of Programs*. ACM, 2010, p. 11. 2.1

[12] B. Lew, “iOS resolution quick reference,” http://www.iosres.com/, 2014. 2.1

[13] “Cross platform mobile development tools: Market analysis & forecast - third edition,” https://www.reportbuyer.com/product/2266370/cross-platform-mobile-development-tools-market-analysis-and-forecast-third-edition.html, Smith’s Point Analytics, 2014. 2.2

[14] Y. Ridene and F. Barbier, “A model-driven approach for automating mobile applications testing,” in *Proceedings of the 5th European Conference on Software Architecture: Companion Volume*, ser. ECSA ’11. New York, NY, USA: ACM, 2011, pp. 9:1–9:7. [Online]. Available: http://doi.acm.org/10.1145/2031759.2031770 2.2

[15] A. Charland and B. LeRoux, "Mobile application development: Web vs. native," *Queue*, vol. 9, no. 4, pp. 20:20–20:28, Apr. 2011. [Online]. Available: http://doi.acm.org/10.1145/1966989.1968203 2.2

[16] K. Haller, "Mobile testing," *SIGSOFT Softw. Eng. Notes*, vol. 38, no. 6, pp. 1–8, Nov. 2013. [Online]. Available: http://doi.acm.org/10.1145/2532780.2532813 2.3, 4

[17] H. Khalid, E. Shihab, M. Nagappan, and A. Hassan, "What do mobile app users complain about? a study on free ios apps," *IEEE Software*, vol. 99, no. PrePrints, p. 1, 2014. 2.3

[18] J. Jung, S. Han, and D. Wetherall, "Short paper: Enhancing mobile application permissions with runtime feedback and constraints," in *Proceedings of the Second ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, ser. SPSM '12. New York, NY, USA: ACM, 2012, pp. 45–50. [Online]. Available: http://doi.acm.org/10.1145/2381934.2381944 2.3

[19] "Prezi," https://itunes.apple.com/us/app/prezi/id407759942, Prezi Inc., 2014. 2.4, 2.4.2

[20] "Keynote," https://itunes.apple.com/us/app/keynote/id361285480, Apple Inc., 2014. 2.4.1

[21] "Keynote," https://www.apple.com/ios/keynote/, Apple Inc., 2014. 2.4.1

[22] "Microsoft PowerPoint for iPad," https://itunes.apple.com/us/app/microsoft-powerpoint-for-ipad/id586449534, Microsoft Corporation, 2014. 2.4.1

[23] "CloudOn," https://itunes.apple.com/us/app/cloudon-document-editor-ms/id474025452, CloudOn, Inc., 2014. 2.4.1

[24] "WPS Office," https://itunes.apple.com/us/app/wps-office-free-+-pdf-compatible/id762263023, Kingsoft Office Software, Inc., 2014. 2.4.1

[25] "Flowboard," https://itunes.apple.com/us/app/flowboard-presentation-app/id630717527, Flowboard LLC, 2014. 2.4.2

[26] "Haiku Deck," https://itunes.apple.com/us/app/haiku-deck-presentation-slideshow/id536328724, Haiku Deck, Inc, 2014. 2.4.2

[27] "SlideShark Presentation App," https://itunes.apple.com/us/app/slideshark-presentation-app/id471369684, Brainshark, Inc., 2013. 2.4.3

[28] "Nearpod," https://itunes.apple.com/us/app/nearpod/id523540409, Panarea, 2014. 2.4.3

[29] "Model-View-Controller," https://developer.apple.com/library/ios/documentation/General/Conceptual/CocoaEncyclopedia/Model-View-Controller/Model-View-Controller.html#//apple_ref/doc/uid/TP40010810-CH14-SW1, Apple, Inc., 2012. 3.2

[30] "Delegates and data sources," https://developer.apple.com/library/ios/documentation/General/Conceptual/CocoaEncyclopedia/DelegatesandDataSources/DelegatesandDataSources.html#//apple_ref/doc/uid/TP40010810-CH11, Apple, Inc., 2012. 3.2

[31] "iOS app programming guide," https://developer.apple.com/library/ios/documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/Introduction/Introduction.html, Apple, Inc., 2013. 3.2

[32] "File system basics," https://developer.apple.com/library/mac/documentation/FileManagement/Conceptual/FileSystemProgrammingGuide/FileSystemOverview/FileSystemOverview.html, Apple, Inc., 2012. 3.2

[33] "SQLite," http://www.sqlite.org/, 2014. 3.2

[34] N. Lockwood, "iCarousel," https://github.com/nicklockwood/iCarousel, 2014. 3.2

[35] A. Ghassemi, "iOS-Rich-Text-Editor," https://github.com/aryaxt/iOS-Rich-Text-Editor/tree/List_Implementation, 2014. 3.2

[36] K. Schwaiger, "KSLabel," https://github.com/vigorouscoding/KSLabel, 2012. 3.2

[37] B. Copsey, "ASIHTTPRequest," http://allseeing-i.com/ASIHTTPRequest/, All-Seeing Interactive, 2011. 3.2

[38] "iOS human interface guidelines," https://developer.apple.com/library/iOS/documentation/userexperience/conceptual/mobilehig/, Apple, Inc., 2014. 3.3.5

[39] "Git," http://git-scm.com/, 2014. 3.5

[40] "Bitbucket," https://bitbucket.org/, Atlassian, 2014. 3.5

[41] "Instruments user guide," https://developer.apple.com/library/mac/documentation/DeveloperTools/Conceptual/InstrumentsUserGuide/InstrumentsUserGuide.pdf, Apple Inc., 2014. 4

[42] J. Penn, "UI AutoMonkey," https://github.com/jonathanpenn/ui-auto-monkey, 2013. 4

[43] A. Vollmer, “Tuneup JS,” http://www.tuneupjs.org/, 2014. 4