Calvary Chapel Corvallis

Technology Review

CS 461 Fall 2016

Kevin Stine, Courtney Bonn, Maxwell Dimm

Abstract

The purpose of this project is to produce an iOS/Android application for Calvary Chapel of Corvallis that will allow members to access a plethora of information all in one localized space. The Church's current website does not provide an interface where current members of the church can very quickly access important information such as events, bulletins, and messages from the service. The desired application will be simple enough for anyone to use while providing back end access for staff to easily upload new information to the app. The priorities lie in maximizing the usability of the app and providing bulletin, schedule, video, and giving functionality. We will work with the existing Calvary Chapel web development team to create a product that is seamlessly integrated with their already existing network.

CONTENTS

I. Introduction

This document will break down our application in 9 different parts. There will be an outline on 3 different types of technology that can be used to implement the 9 different parts of the system. We will detail each type of technology as well as discuss why it could be useful for the project. We will then compare each type of technology against each other and analyze which one would be the best for us to use in production. The 9 pieces of the app are divided among the three group members. iOS development platform, iOS user interface organization, and integrating the e-bulletins from the current website will be researched and handled by Courtney Bonn. Android development platform, Android user interface organization, and integrating the current giving platform will be researched and handled by Kevin Stine. Finally, cross-platform development, integrating the recent sermons, and integrating the existing calendar will be researched and handled by Max Dimm.

II. iOS Development Platform

A. Xcode using Swift

The main technology used to develop iOS applications is Xcode. Xcode is a program that can be downloaded on any Mac OS platform and is not able to be downloaded on Windows PCs. It makes use of developer tools to allow users to create an iOS app that will work with all updated iOS devices. This tool uses the language of Swift, which is a new programming language that has been released by Apple [1]. For people who have never developed an app before, Swift is considered the easier language to learn [2]. The tool allows the developer to use a simulator that will display the graphical interface that the user will see and makes the process of developing an app for iOS devices go smoothly.

B. AppCode

AppCode is a smart IDE that is used for iOS/Mac OS development [3]. There are many features for this IDE that allows app developers to easily and quickly develop an application without dealing with any hassle from their IDE. Some of the features include efficient project navigation, smart completion, reliable refactoring, thorough code analysis, and unit testing. Additionally, AppCode not only supports Swift, but Objective-C, C, C++, JavaScript, XML, HTML, CSS, and XPath. This wide variety of languages allows this IDE to be used for more than just iOS development all in one program. AppCode also works directly with Xcode so developing an app for iOS devices is made simple. This IDE is a paid subscription, costing $199 for the first year and the price decreasing after that. There is a free 30-day trial available to determine whether or not the program is the right tool for the wanted product.

C. iPhone App Builder

The third option for a toolkit to build an iOS app is a program that will essentially build the app for you, leaving no need to learn how to code. One program, AppyPie [4], boasts that it will take all of the work out of creating an app and allow an unskilled user to get a working app quickly uploading to the App store. This option would be useful for people who have no prior programming experience and are just looking to get an app produced with little effort.

D. Goals

The goal for comparing iOS development tool is to acquire which way is most acceptable and makes most sense for developing an iOS application. We want to determine which option is relevant to current applications and which option will produce a valid application.

E. Criteria Evaluation

The criteria we will use for determining the iOS Development Platform includes:

1) **Price:** What is the cost associated with the program?
2) **Difficulty:** Will the program do all the work for you? Is there room for learning how to build the app from scratch?

F. Table Comparison

TABLE I
COMPARISON OF IOS DEVELOPMENT PLATFORMS

Tool	Brief description	Rationale
Xcode using Swift	Using the Swift programming language to write the application using the program Xcode	Best option as it is the most recent and will allow us to actually learn how to develop an app
AppCode	Using the AppCode IDE to write an iOS app	Great option if price were not an issue as it does incorporate multiple languages into the IDE
iPhone App Builder	Using a third-party program to automatically build the app	A good option for people who have no programming experience and just want to quickly produce an app

G. Discussion

Using Xcode with Swift is a strong option for building iOS–some would say that it is one of the only options. Using a third-party IDE like AppCode is also a strong option as it does have many other features, like the ability to program in other language, that Xcode does not have. Not only that, it does integrate itself with Xcode. However, at $199 for 1 year, the price is steep and may only make sense for those who develop apps professionally.

The idea of using a third-party program to build the app automatically is tempting, but would defeat the purpose of learning how to develop an iOS app. If the objective of this project was to produce an iOS app within a matter of days or weeks, then this option could be considered useful.

H. Selection

Because our main objective for this project is to learn the detail of producing an iOS app, it is best to choose Xcode with Swift. Swift is the most updated language for iOS development and using Xcode simplifies the process of building the application. With Xcode and Swift, we will be able to properly produce an application that can be submitted to the App Store for download. It will increase the likelihood of developing an app that is testable, usable, and visually appealing on an iOS device. Considering all of the options we have for building a native iOS application, it makes sense to choose Xcode and Swift.

III. iOS User Interface Organization

A. Sketch

Sketch is a paid program exclusively available on Mac OS X 10.10+ [5]. This program gives app designers a platform for organizing the user interface. There are many features that are available in this program, including precision, objects, the inspector, tools, reusable elements, and exporting. Precision allows scalability to be present in the app. Objects allow each shape created to be easily findable and completely editable. The inspector is the tool for dimensions, positioning, opacity and everything else you would need in order to control the design of the app. Because Sketch is exclusively available on Mac, it uses Apple's frameworks so the app will be completely supported. Most importantly, there is a Sketch app available for download on iOS devices which allows the designer to view the app design on an actual iOS device, rather than just on the computer. This program is $99.00 but there is a free trial and education discounts.

B. Interface Builder

Within the Mac native program Xcode, there is a built-in editor called Interface Builder that allows the designer to build a user interface [6]. It uses a Model-View-Controller pattern which allows the designer to build the interface without worrying about the implementation of the actual app. This program will connect the user interface to the code for the app automatically. There are multiple views in an app, and this builder uses storyboards to keep the multiple views organized. This tool gives the user the option to preview the user interface without actually running the app which saves time. Users are automatically given access to the Interface Builder when downloading and using Xcode.

C. Marvel App

Marvel App is a program that allows designers to design, prototype, and collaborate on their app designs [7]. Marvel App allows users to create two projects for free, and has paid subscriptions for any additional projects. You can use images from Sketch or Photoshop to work into your design, or you can design your interface directly using Marvel. It allows for prototyping that makes your app look and feel like it is a real app–which allows user testing to begin on just the basic design of the app. The tool used for designing is called Marvel Canvas and it can be used on any computer without downloading. This tool allows users to sync their designs to Dropbox or Google Drive so designs are automatically saved. One great feature is being able to upload sketches that you have done on paper by using the Marvel app on the iPhone.

D. Goals

The goal of comparing different ways to design the user interface is to obtain the best way to organize our iOS app. There are many programs out there that will design the app for you, but we want an app that is simple, yet sophisticated, and works for the general public.

E. Criteria Evaluation

The criteria we will use to determine iOS UI Organization includes:

1) **Price:** What type of price is associated with the program?
2) **Easy integration:** Will the UI program easily integrate with our chosen app development IDE?

F. Table Comparison

TABLE II
COMPARISON OF iOS UI ORGANIZATION

Tool	Brief description	Rationale
Sketch	A paid program that allows for designing a user interface that has many features	A great option if price were not a factor
Interface Builder	Built in with Xcode so it allows for the designing of the app and programming to be done all in one program	Best option due to price and simplicity of only using one program
Marvel App	A program that combines design, prototyping, and collaborating for iOS applications	Good option despite its pricing because it allows two free projects

G. Discussion

All three options presented here would be extremely useful in designing an iOS app. The Interface Builder makes a lot of sense purely because it allows us to use one program for both the development and design of the app, rather than having to switch between programs. Sketch has wonderful reviews for designing an iOS app and could work well if price was not a factor. Unfortunately, $99.00 is not in the budget for this project. Marvel App allows for two free projects so price ends up not being a factor since we are only working on one project. The prototyping feature of Marvel App is attractive and would be useful for this project.

H. Selection

Despite the obvious positive attributes of Sketch and Marvel App, using Interface Builder still makes the most sense for this particular project. Our app is not going to be overly complicated in its design, so using the storyboards and views offered in Interface Builder should be enough tools to build the simple design we are looking for.

IV. ANDROID DEVELOPMENT PLATFORM

A. Android Studio

Android Studio is the official integrated development environment (IDE) for Android. [8] It provides the fastest tools for building apps on every type of Android device. Android Studio comes with tons of tools that will make development for Android much easier. The IDE has a feature called Instant Run, which allows you to push code and resource changes to your app while it is running on a device or emulator to see the changes instantly come to life. This feature can drastically help speed up development, as we will not have to constantly rebuild the app and start up the emulator each time we want to test a new change. Android Studio also features an intelligent code editor, allowing us to write better code, work faster and be more productive. Android Studio is built on IntelliJ and is capable of advanced code completion, refactoring and code analysis. It also has a fast emulator which will allow us to quickly get our app up and running for testing and debugging. Android Studio also has the following features: a flexible Gradle-based build system, GitHub integration, extensive testing tools and frameworks and lint tools to catch performance, usability, version compatibility and other problems.

B. Appcelerator

Appcelerator is a platform which provides everything you need to create native mobile applications - all from a single JavaScript code base. [9] Appcelerator offers direct access to native APIs using Hyperloop, delivers fully native apps for rich user experience, immediate support for each new OS release, and seamless integration to existing continuous delivery systems. Appcelerator also comes bundled with your own MBaaS (Mobile Backend as a service). This feature, called Arrow, is a powerful opinionated framework for building and running APIs. Arrow would allow us to deliver data to any app client, engage users with pre-built notification capability, trigger notifications based on user location or predetermined schedule and view notification history and details. This could be extremely helpful for the utilization of push notifications for Church members. We could have schedules that would send out a notification at 10:15 AM with the morning bulletin for users that are at church that morning. This would be very useful as we would not be sending out notifications to every app user since not everyone might be able to attend that particular day. Appcelerator also comes with real-time mobile analytics for every native app - whether built on the Appcelerator Platform or directly via native SDK (iOS & Android). This provides a mobile lifecycle dashboard for visibility into all apps and performance and crash analytics. This feature could be extremely helpful for the church so they can monitor who is using the app and get some visual feedback on its success.

C. Adobe PhoneGap

PhoneGap is a platform used to create mobile apps that are powered by open web technology. [10] PhoneGap allows you quickly make hybrid applications build with HTML, CSS and JavaScript. It also allows you to create experiences for multiple platforms from a single codebase so you can reach your audience no matter their device. PhoneGap includes PhoneGap Build, which takes the pain out of compiling PhoneGap apps. Build allows you to get app-store ready apps without maintaining native SDK's and compiles it for you in the cloud. PhoneGap includes a desktop app for creating apps without using the command line, and a mobile app to connect your device to your development machine to see changes instantly. This could be a great resource to utilize since it is free, and the apps are created using HTML, CSS and JavaScript rather than Java.

D. Goals

The goal is to determine which development platform we would like to utilize for creating our Android version of the app. Since we need to create an app for both Android and iOS, it is important to determine how we want to go about that. We can create applications using the native SDK's through Android Studio and Java, or we could utilize a third party application such as Appcelerator or PhoneGap which will work using HTML, CSS and JavaScript. The goal is to determine which method we think would be most effective for creating our application.

E. Criteria Evaluation

The criteria for determining which development platform we use will be based off:

1) **Usability:** Is the development platform user friendly, have the correct tools we need to utilize and allow for quick development.
2) **Price:** Is the development platform free and open source, is there an upfront one time cost, or a recurring cost.
3) **Language:** Does the development platform utilize programming languages which we are already familiar with, or will we have to learn a new language to create a mobile app.

F. Table Comparison

TABLE III
COMPARISON OF ANDROID DEVELOPMENT PLATFORMS

Tool	Brief description	Rationale
Android Studio	Using Android Studio and the native Android SDK	Best option for creating a native app built in Java that will be platform specific
Appcelerator	Using the Appcelerator Platform to create an iOS and Android app in JavaScript	Best option for the number of features, can be written in JavaScript but is not free
Adobe PhoneGap	Using the Adobe PhoneGap platform to create an iOS and Android app build in HTML, CSS and JavaScript	Great option for usability since we can utilize previous Web Development skill

G. Discussion

Android Studio offers the most pure Android development experience as it utilizes the native Android SDK. This makes it a strong contender for the platform we will choose for development. In addition to utilizing the Android SDK, it has features such as Instant Run which would allow us to constantly make changes to the app while it is running to see the changes occur instantly. This makes Android Studio very powerful and helpful through the development process.

Appcelerator is another strong contender as it provides a ton of functionality since it comes bundled with a few other features such as Titanium, Arrow and user analytics. While this could provide a lot of good information for the church once the app is up and running, it does cost roughly $400 dollars a year which is a bit pricey. To get the most out of this platform and the best return on investment, it would require the church to have someone using all these features by actively monitoring analytics and crash reports. We think that even though this looks like it is an incredibly powerful platform, it would be underutilized and provide more functionality than needed for the scope of this app.

Adobe PhoneGap does seem like a very promising option as it allows the app to be developed in HTML, CSS and JavaScript. This is a plus since our team is already familiar with these languages and there would be less of a learning curve then utilizing Java. PhoneGap comes with a desktop app and mobile app for easy development. Rather than having to deal with compiling the application PhoneGap handles that in the cloud. This could be a really good option as it might be simpler to use with less of a learning curve than other platforms.

H. Selection

Based on the scope of this project, we have decided that utilizing Android Studio with the native Android SDK will be the best option moving forward. Since the team joined this project in hopes of learning more about mobile app development, we think that utilizing the native Android SDK will give us the best learning experience. While developing the app in Java may have a steeper learning curve than HTML or JavaScript, we believe that we will learn the most by using the language that most Android apps are written in. We hope that we will also be able to create a smooth and well polished app using Android Studio and features that it comes with. As Android Studio if the official IDE for Android, we believe it is the best platform to use.

V. Android User Interface Organization

A. Tab Navigation

There are many different ways in which we can structure our Android app. Navigating between pages and windows is a large part of the User Interface, and we want to make sure we are using the most intuitive method for navigation. One method would be to use tabs which make it easy to explore and switch between different views. Tabs enable content organization at a high level and can be used for switching between views, data sets, or functional aspects of an app. [11] The utilization of tabs would allow for swipe gestures between the various tabs of the application. Because it uses swipe gestures, we would need to make sure to not use content which also supports swiping. Tabs provide an easy to use and recognizable view which immediately lets users know that they can swipe between the content listed at the top. Tabs are a good choice because it provides all of the content directly on the first page for users to see. This would prevent users from getting confused since the navigation pane will not be hidden.

B. Side Bar Navigation

Side bar navigation is another UI layout which creates a navigation bar to the left of the content. Side nav bars can be pinned for permanent display or can float temporarily as overlays. [12] Temporary nav drawers overlay the content canvas, which is likely how we would use a side nav bar in our application. Side nav bars are an excellent option for navigation since it essentially hides the navigation pane until the user swipes in from the left of their screen. This allows for the focus of the page to be on the content and would allow us to maximize the space that the user interacts with. Instead of taking up screen space for a navigation pane, we can hide the navigation panel and have it accessible to the user from any page. Side nav bars are extremely popular in most material design inspired applications, and would fit perfectly in our app as well.

C. Bottom Navigation

Bottom navigation make it easy to explore and switch between top-level views in a single tap. Tapping on a bottom navigation icon takes you directly to the associated view or refreshes the currently active view. [13] This is a pretty standard UI layout that can be found on the majority of iOS applications. This would be a great choice for being able to match the layout we use for our iOS application. All of the pages are neatly laid out at the bottom of the screen, so it is easy for the user to know exactly which pages are available and how to access them. The only downside of this layout would be for Android phones that have on-screen soft buttons rather than off-screen hardware buttons. This could potentially provide an issue for users trying to access certain content but rather then clicking the page they want, they accidentally click the home button instead. Despite this, bottom navigation would be a great option for continuity across both of our mobile applications.

D. Goals

Our goal is to determine which UI layout we want to use for our Android application. We want to narrow down the three main UI layouts to determine which one will be most user friendly and most intuitive to app users.

E. Criteria Evaluation

The criteria for determining which UI layout we use will be based off:

1) **Continuity:** Does this layout provide a similar look and feel to our iOS application.

2) **Usability:** Is this layout easy to use and does it provide enough context for the user to know how to navigate the application.
3) **Intuitiveness:** Is this layout simple and easier for anyone to understand and are they able to navigate the various pages without being walked through the process.

F. Table Comparison

TABLE IV
COMPARISON OF ANDROID UI ORGANIZATION

Tool	Brief description	Rationale
Tab Navigation	Layout that includes tabs at the top of the page	A great option for being able to quickly swipe between pages
Side Bar Navigation	Layout that includes a nav drawer overlay	A great option to have a hidden navigation bar that is accessible from any page
Bottom Navigation	Layout that includes bottom buttons	Col 3 A great option for continuity with our iOS app and easily viewable information

G. Discussion

The use of tabs for the navigation of the app is a very intuitive and easy layout for people that are familiar with Android devices. On Android, almost everything can be done by a simple swipe motion. This is less common on iOS devices, so it could be confusing for people that are used to iOS instead of Android. This is also true of the sidebar navigation pane, which is hidden be default unless the user swipes in from the left or clicks on the hamburger menu. For continuity, the use of the bottom navigation would make the most sense as a user switching from iOS to Android will find it similar. This does however cause issues with some Android users that have on-screen soft buttons.

H. Selection

While the sidebar can be a little less intuitive than other UI layouts, we think this will be the best option to use in our application. With the navbar off to the left side of the page, it will allow us to focus on the clarity of the content that we are presenting to the user. This frees up screen space by not using it for tabs or buttons. This also allows the user to easily get to the settings and access all other pages by swiping in from the left. We believe that the side navigation bar will be the most intuitive and easiest to use once people use it for the first time.

VI. CROSS-PLATFORM DEVELOPMENT

A. Sencha

Sencha is a platform with the purpose of creating web and application platforms for developers using a code base of HTML5 [14]. The benefits of creating apps in HTML5 can be pretty major when you look at them. The first and biggest benefit is that it provides a single codebase for all platforms, Android, IOS, and even web viewing. This means that when you want to update your code or something about your application, that you do not need to edit multiple codebases. A

strength of this is when people open your app, they will be having the same experience regardless of which platform they are viewing it on. This provides consistency between your user bases which leads to less confusion. Sencha being a HTML5 development framework also has the benefit of hosting developers and many code review services which is good for the longevity of the project. This would mean that we could get the project completed and functional to the client's desires, but later it would give our client a resource to maintain and update the app after we are gone. One of the downsides to this platform and method is that often times, cross platform interfaces can be difficult. When working with different hardware and peripherals there can become redundancies within your app. Also, when working with HTML5, JavaScript and CSS, the applications speeds might not be as responsive as native applications.

B. Appcelerator Titanium

Appcelerator is a platform to write mostly cross platform apps using JavaScript [9]. The platform claims to reuse 60-90 percent of code when developing cross platform but provides native app speeds and functionality. The platform boasts an intuitive visual design process for all operating systems. It also allows you to compare your different UI differences between OSs include visually impressive visual effects through JavaScript. It also has a framework for building and running different APIs with other applications and working with existing systems such as CCB. Some of the downsides being though that it is not truly a cross platform development strategy as different device platforms would have different codes. Also it seems that the use of this coding platform and their site requires a subscription model to Appcelerator which can become costly. One benefit to this though is they provide their own database storage, can complete API calls and have a built in file storage for development purposes.

C. Apache Cordova

Apache Cordova is a free, open source coding API that provides nearly 100 percent code re-use between platforms using JavaScript [15]. Cordova works in conjunction with visual studio for easy setup when starting out your project. The way it works from a high level is it takes advantage of the universality of HTML and JavaScript while simply managing the OSs specific API to interpret the page. This would mean, like Sencha, that the app would be available in many ways beyond just iOS and Android platforms. Cordova has CLI, which is a high-level tool they developed that allows you to develop for multiple platforms simultaneously by abstracting a lot of the functionality of lower level shell scripts. The downsides of this option is because it is a free and open source, it lacks a lot of the support and functionality of the above paid options. It does not have built in cloud storage, no database to use for the app post launch and no developers working on their end. Meaning that to modify the app after our development finished outside help would be necessary.

D. Goals

The goal here is to determine which platform we would use if we choose to develop our application to be writeable cross-platform. There are many frameworks and platforms available to us to code our application so it is best to scope out the different options, weigh the pros and cons of each option and make an educated decision based on that info. We want to develop our app in an efficient and low cost manner.

E. Criteria Evaluation

Our criteria for determining which platform to use will primarily come down to pricing. However it will also factor in the amount of features each tool or method will bring to the table. Along with that it will be noted the language being coded in and its strength within the scope of the different operating platforms.

F. Table Comparison

TABLE V
COMPARISON OF CROSS-PLATFORM DEVELOPMENT

Tool	Brief description	Rationale
Sencha	Using HTML5 and existing frameworks	Strong option for the longevity of the app
Appcelerator	Creating native apps for each OS while reusing 60-90 percent of code	Visual design process and backend features are amazing, but come with large pricetag
Apache Cordova	Using their custom CLI to code in HTML and JavaScript	Solid option with less features but the great price of free

G. Discussion

Appcelerator seems to be the strongest option in the perspective of what it provides to us. The amount of features, including cloud storage, a built in database, user analytics, and visual design process. However the price is very high and on a subscription model to upkeep the code base, and database.

Sencha appears to be a mix between Appcelerator and Apache Cordova. It provides less features then Appcelerator but they have a team of developers that you can hire to work on and maintain your application. This could be helpful less to us but our client for the longevity of their application.

Apache Cordova boasting a perk of being open source and free is too big to overlook. When working for a client that represents a non-profit organization it is crucial to keep costs low in the scope of the project. Apache Cordova would provide us the tools to get the job done while not costing us a cent.

H. Selection

Based on how heavily price will weigh in on our client we think that if we choose to go with a cross platform development strategy that we should go with Apache Cordova. It give us a lot of the tools we will need, is free and open source, and allows us more freedom to integrate with our clients existing database/back end.

VII. INTEGRATING THE GIVING PLATFORM

A. Authorize.net

Authorize.net is a payment gateway which enables Internet merchants to accept online payments via credit card and e-checks. [16] This giving platform allows you to accept credit cards and electronic checks from websites and deposit funds automatically into your merchant bank account. Authorize.net allows for all major credit cards, signature debit cards, linking of bank accounts, and digital payment solutions such as Apple Pay, PayPal and Visa Checkout. It also comes with fraud

prevention to identify, manage and prevent suspicious fraudulent transactions with address verification service and card code verification. This platform also gives you online access to manage and review transactions, configure account settings and download reports with the ability to sync to Quickbooks. In addition to online payments this platform accepts payments that are made on a mobile application and they also provide a free mobile app. Authorize.net charges 2.9% plus 30 cents per transaction with a $49 setup fee and a $25 monthly gateway fee. Authorize.net is currently what Calvary Chapel is already using as their platform for giving, so utilizing this platform will probably be the best option for our application.

B. Pushpay

Pushpay is another giving platform which allows you to make collections for your organization in seconds. [17] Pushpay is one of the fastest ways to give to your church and requires only three steps to give. You need to download the Pushpay app, choose who you want to pay and how much, and enter your passcode and the donation is complete. Pushpay allows you to pay any registered merchant and includes geo-location and search technology to find who to pay. Pushpay makes giving extremely easy, and provides the users with a simple, easy to interact with application that makes giving a breeze. This is a great option for giving as it provides a mobile application and website which you can give securely from. It hooks in directly to the Church's merchant account and can be up and running quickly. While this is a very simple and easy option, it is a bit pricier compared to Authorize.net.

C. PayPal

PayPal offers a service which allows people to donate to your non profit or organization. [18] It allows you to add one simple button to your website or app which lets you accept credit cards, debit cards and PayPal. Users do not even need to have a PayPal account in order to donate via the PayPal Donate button. This provides an extremely easy to use interface, and would allow us as developers to easily integrate giving into the application. PayPal allows you to get access to your funds quickly as the donations process within minutes. Once the donation is processed, you can then transfer it to your organization's bank account at no charge. PayPal is a great option since there are no charges associated with donating.

D. Goals

The goal is to determine which giving platform is the easiest to use by church members, provides the simplest and fastest way to donate money, and is secure.

E. Criteria Evaluation

The criteria for determining which giving platform we use will be based off:

1) **Security:** Does this giving platform have the technology to ensure that all payments are secure.
2) **Reliability:** Does this giving platform perform well and allow users to give when they want to give.
3) **Usability:** Does this giving platform have a simple, easy to understand interface for users.

F. Table Comparison

TABLE VI
COMPARISON OF GIVING PLATFORMS

Tool	Brief description	Rationale
Authorize.net	Secure and simple giving platform currently in use by the church	Great option because it is what the church is currently using
Pushpay	Simple, fast and intuitive giving platform	Great option for speed and usability for users
PayPal	Simple giving platform that does not require a fee	Great option for allowing users to give without setting up any type of account

G. Discussion

The various giving platforms that we have looked at are all pretty similar in that they all essentially provide the same functionality. When it comes to giving, you do not really want a platform with tons of features that would just dilute the process of giving. It should be something simple and easy that can be done within seconds whenever the user would like. Because giving is generally a monthly occurrence, having a platform which supports automatic recurring donations would be a great feature to have.

Authorize.net is a great option because it is what the church is currently using as their giving platform. Pushpay provides a wonderful user experience and makes giving extremely simple. Just three clicks and you are done. PayPal also is another great option as they allow you to just include a button on your website or application. This makes the integration with the giving platform very easy. This is also a great option because it does not require users to create any type of account. They can just give whenever they like using credit, debit or PayPal.

H. Selection

Despite having a lot of good giving platform options, the one that we have decided to stick with is Authorize.net. Since this is the platform that the church is already using, it makes the most sense to integrate that into our application rather than having the church switch their platform just for this application. While the other options may have great user-friendly UIs, we want to create an application which will utilize what the church currently has in place.

VIII. INTEGRATING E-BULLETINS

A. Contentful

Contentful is a content management system that allows content to be presented on a website or application through their API [19]. There are different products available from this company, but one in particular would be best in delivery the E-Bulletins to our app. The Content Delivery API is a tool that pushes read-only data to the app. This is ideal for blog updates and similar information that is not ever going to be edited within the app itself. This API takes advantage of the Contentful Web App which is the online editor that developers can use to create or update their content. Contentful has multiple pricing plans, including a free one that is described as good for small projects. An advantage that this system has is that once the API is set up, the content can be edited directly from Contentful and without any prior technical knowledge.

B. Wordpress

Wordpress is a powerful website management system that has the ability to create an entire website and manage the content all from one editor [20]. With Wordpress, you can create pages that can be uploaded to an application. This allows developers to update a page from Wordpress' editor and save the changes which will appear on the app without having to actually edit the app itself. Wordpress has an application framework that is used to integrate the pages into the application. This system is a great option for developers who are wanting to have the same information both on an application and a website.

C. Church Community Builder

Church Community Builder is a software program that is available to churches to use for their websites [21]. It uses API to integrate calendars, forms, and other features into existing websites. Because Calvary Chapel Corvallis already uses Church Community Builder, their development team is already familiar with this program. In order to integrate the e-bulletins, a group for announcements could be made and the e-bulletin could be uploaded to that group. To display the bulletin on the app, we would use the existing API network. Church Community Builder is a paid software system, but in our case the church has already paid for this service.

D. Goals

The goal is to determine the best way to integrate the church's existing information, in this case their e-bulletins, from their website onto the app. Currently they just update their website with the content whenever they need to, so we need to discover a way to push the information to the app without having to update and resubmit the app each time there is a new e-bulletin.

E. Criteria Evaluation

1) **Price:** Is the program free for our purposes?
2) **Usability:** Will people with little technology background be able to update the content?
3) **Continuity:** Will the program work well with the current church website?

F. Table Comparison

TABLE VII
COMPARISON OF E-BULLETIN INTEGRATION

Tool	Brief description	Rationale
Contentful	CMS that uses API to push updates to the app	There is a free option, but it may not be what our project needs
Wordpress	Website management system that can push updates to an app	Would be a better option if the existing website was also using this system
Church Community Builder	A Church management system that has API ability	A great option as the church already uses this system

G. Discussion

Contentful would be a good option if we were developing a more complex app. Because we are only looking to integrate this type of content for the bulletins, Contentful might be a bit too powerful for our purposes. Wordpress is a better option for a less complex app if one is already familiar with how Wordpress works and if they were utilizing that tool in their existing website. Because they are not using Wordpress at all in their website, it may be too complicated to add another tool for them to keep track of. Church Community Builder seems to make the most sense because they already use this system. Their staff already knows how to use it and there would not be as much of a learning curve to learn how to update the e-bulletins.

H. Selection

For this project, the option that will work best is Church Community Builder. While the other options offer useful features, the Church already utilizes Church Community Builder so there will be no additional costs or learning to integrate the e-bulletins using this program.

IX. Integrating the Sermon Platform

A. LiveStream

Livestream is a video broadcast and viewing platform that allows groups to distribute their events to anyone who is interested or subscribes to their content [22]. They provide the ability for their clients and users to stream their content live for the viewers to watch and then host the video for later viewing afterwards. They host integration with a variety of different systems including web access, through their app, or many popular video viewing platforms such as Roku or Chromecast. The major benefit to using livestream is that our client has existing knowledge of this platform and already is hosting a lot of their content through them. They do have an associated cost that comes with the platform, but it comes with more support than many conventional video hosting sites.

B. YouTube

YouTube is one of the world's most popular and well recognized video hosting platforms [23]. They have recently come to introduce live streaming capabilities and continue to dominate the video hosting market. Because of their sheer size they have the greatest range of integration with various video hosting platforms and a well-polished and understood API for us to use. They also have their own application that could be linked with our app for seamless viewing. They do not charge for their video hosting and can even pay the user for their content if they choose to enable adds which is YouTube's revenue source.

C. Vimeo

Vimeo is another popular video hosting service that many smaller organizations look to [24]. It allows for 500MB of weekly uploads free of charge up to a total storage of 10GB. If these totals do not allow for enough storage the user can upgrade to a paid account with fewer restrictions. They have an API for use that can be accessed upon approval from their team. They have a larger support team then the automated YouTube while having lower prices than livestream making them a mix of the two services of sorts. They also offer production assistance through their team to their team/organization accounts.

D. Goals

Our goal is to make our clients sermons available for viewing within the app in a way that is easy for the user. We also want the upload process to be as easy as possible for our client to reduce upkeep of our app. They want to continue to upload their content as they have in the past but have it available through the app.

E. Criteria Evaluation

We think the primary criteria to evaluate is how well it will integrate with their existing systems. If it requires an overhaul of how they upload videos then this would be considered a failure. The next most important criteria we believe is the pricing. Keeping costs low for non-profits is important so we are aiming to keep any recurring costs as low as possible for our clients.

F. Table Comparison

TABLE VIII
COMPARISON OF SERMON PLATFORMS

Tool	Brief description	Rationale
LiveStream	Business oriented video hosting that comes at a premium	Is the churches existing video platform
YouTube	Most popular video hosting platform with high usability	Free and well known for being a top tier platform
RVimeo	Fairly well know video hosting with high quality platform	Solid hybrid of YouTube and LiveStream

G. Discussion

So we discussed earlier that the biggest criteria for evaluation will be how well it integrates with their existing systems. LiveStream earns itself a huge leg up on the competition because it is the platform that our client is already using. This would mean that they do not have to change how they upload video from their sermons and would overall be the easiest option.

Youtube is still enticing because of the fact that it is so recognizable. On top of this they also have a giant perk of not only being free, but offering a share of their add revenue to their content creators. Their API would be usable to us as it is one of the most commonly used APIs for this style of content.

Vimeo, while being a combination of the two styles, is far less desirable because it is not already integrated with the client and it's simply not as good as YouTube. It is fairly clearly the worst option when you factor in that their API is only usable once approved so it would leave up the possibility of being denied.

H. Selection

While YouTube is such a strong option, we will be going with LiveStream. Our client is already hosting their videos through this service and they have requested that we leave their existent infrastructure as intact as possible. The price is not

awesome, but we can discount it in a way because it is not an additional cost. The client was already paying for this service so we are not adding to any costs for them. LiveStream on a functional level should complete the task just fine and work with our app excellently.

X. Integrating the Calendar Platform

A. Google Calendar

Google Calendar is one of the most widely used and recognizable calendars available to developers like ourselves [25]. It has tons of resources online to learn about it and a vocal user base to explain its integration with different apps. From what I can find we will be able to integrate it with our app without incurring any additional costs. We also know that because it is google that it will function identically with both IOS and android platforms which is a big plus. Unfortunately our client is in the process of transitioning their existing calendar off of google and onto CCB.

B. Microsoft Outlook

Microsoft has a calendar built into their email system that we could utilize for our application [26]. One of the big benefits outlook provides to us is that fact that windows is so commonly used among computer users. This means that along with our app, we could sync up the calendar with peoples computers which may be easier for some. Many users do not look to their phones exclusively for their scheduling and this would not alienate that part of our user base.

C. Church Community Builder

CCB is a development group centered on helping churches get on their feet in the complicated area of tech development [21]. Our client is moving most of their infrastructure over to CCB so integration with them in general is going to be crucial. Because most of the data we will be pulling is already from CCB it will be beneficial to us to reduce how many outside resources we are using and consolidate. They have an existing API that we can take advantage of when implementing the churches calendar.

D. Goals

The goal with the calendar integration is to make it as easy to update for our clients staff as possible. We want our app to have as minimal upkeep for our client as possible and whichever method accomplishes that while providing the necessary information will be the best option.

E. Criteria Evaluation

As stated above we want to evaluate the systems on ease of use for the clients on the backend. Alongside that the application of the calendar should be sleek and easy to interpret. As always, reducing costs wherever possible is also crucial.

F. Table Comparison

G. Discussion

In our eyes two of the big priorities when it comes to external API and including it in our project is to make it require as little work from our client as possible and work with their existing systems. This is the major benefit of using the Church Community Builder system that exists. T hey have an existing API that is functional for us to use as well.

TABLE IX
COMPARISON OF CALENDAR PLATFORMS

Tool	Brief description	Rationale
Google	Existing Google calendar integration with app	Recognizable and highly developed
Microsoft Outlook	Integration with windows and Microsoft Calendar	Allows higher integration with computers alongside the app
Church Community Builder	Church Management system with existing API	Church is already familiar and using this system

Google is a strong option and probably would have been a favorite if our client was not in the process of phasing out their existing integration with google. They have a strong background and very easy to work with API. Also the google branding is easily recognizable and would be comfortable for our users.

Finally Microsoft provides a set of tools that are interesting to talk about, but may not be super applicable to the project in front of us. Yes, it would integrate well with the windows operating system, but we are not building our app to work with computers. We are building an app that will work with IOS and Android so this may not be as relevant as you would think.

H. Selection

For the calendar we will be using CCB or Church Community Builder. Not only is it going to simplify the process by having more information being pulled from a central location, but our client specifically asked us to use this platform out of ease to their staff. Also the existence of their own API that we can make use of is excellent for us and will be a major tool in development of the app.

XI. Conclusion

Our main goal with the technology that we choose for our project is working well with the Church's current back-end system, Church Community Builder. Additionally, creating a product that the team at Calvary Chapel Corvallis can successfully continue to update and maintain is extremely valuable to this project. These goals have contributed to the technology that we have chosen to move forward with. Price of the technology was also a large factor that we had to consider.

The technology we chose for developing the iOS portion of the app is Xcode and Swift. We will use Interface Builder that is built-in to Xcode in order to organize the user interface. For the Android portion of the app, we have chosen Android Studio. We will organize the Android app using a sidebar organization layout.

Because Calvary Chapel Corvallis already uses Authorize.net for their giving platform, we have decided it is the best option for the donation page on the app as well. Similarly, Calvary Chapel Corvallis uses the Church management system, Church Community Builder, so we will continue to use this system to present the e-bulletins on the app. The technology we will use to display the sermons is LiveStream. This will work well despite its price, because Calvary Chapel Corvallis has already chosen to use this technology. Finally, we will once again be using Church Community Builder to display the calendar and events on the app.

The technology we have chosen to use will enable the church to successfully take over the maintenance of the app once the project is finished. It will also allow them to continue using their current back-end system, which will cut down on additional cost and education for the church staff. We are confident that the technology we have chosen will help us create a simple app that the members of Calvary Chapel Corvallis can use and enjoy.

REFERENCES

[1] A. Inc., "Start developing ios apps (swift)," September 2015.

[2] "How to make iphone apps with no programming experience," October 2016.

[3] "Appcode," 2016.

[4] "iphone app builder," 2016.

[5] "Sketch app," 2016.

[6] "Interface builder built-in," 2016.

[7] "Marvel app," 2016.

[8] "Android studio," 2016.

[9] "Appcelerator," 2016.

[10] "Phonegap," 2016.

[11] "Material design tabs," 2016.

[12] "Material design side bar navigation," 2016.

[13] "Material design bottom navigation," 2016.

[14] "Sencha," 2016.

[15] "Apache cordova," 2016.

[16] "Authorize.net," 2016.

[17] "Pushpay," 2016.

[18] "Paypal," 2016.

[19] "Contentful," 2016.

[20] "Wordpress.org," 2016.

[21] "Church community builder," 2016.

[22] "Livestream," 2016.

[23] "Youtube," 2016.

[24] "Vimeo," 2016.

[25] "Google calendar," 2016.

[26] "Microsoft outlook," 2016.