

# Calvary Chapel Corvallis

## Progress Report

### CS 462 Winter 2017

Kevin Stine, Courtney Bonn, Maxwell Dimm

Group #62

#### **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.

## I. PURPOSE AND GOALS

The purpose of our project is to create an application for Calvary Corvallis Church that will act as a connection between the congregation and the administration. The church already has a website that has some of this information, but they want the website and app to serve different functions. The website will be to introduce people to the church. The app will be used for the existing congregation as the go to place to access the most commonly used or needed information. Some of the features being provided within the app are: having sermons available, listing the bulletin, having the church schedule, and allowing members to donate to the church. Our client explained to us that these were the features that they wanted in the app as they are the most needed services by their members.

Our client has requested that the app be as automated as possible in regards to updating the information hosted within it as to reduce any upkeep as much as possible. So we will be working with their existing infrastructure as much as possible to pull our information from. We are also creating both an iOS and Android application and we want to make the applications as functionally similar as possible. This will allow for greater understanding of the app across users who may or may not be super tech savvy. Our final goal in this project is to reduce costs wherever possible for our client. If that means suggesting newer cheaper infrastructure or setting up our app in a way that reduces how often it will need to be updated, we want to do it.

## II. FALL 2016

### A. *Progress report*

Currently, we are still in the planning phase of the project. We have just finished the design document which detailed how we plan on implementing our app and how it will be designed. However, we have not begun the actual implementation of either app quite yet. Our immediate plans include extensive research on iOS and Android app development which we will focus on during Winter break.

### B. *Problems*

One of the first issues we ran into was the fact that there are multiple platforms to run an app on. At first our client did not know, but it became quickly apparent that they wanted their app to be available to all their members. This meant that we had to switch from creating just one app, to developing two apps that will perform the same task. This adds a substantial amount of work to our project. We are looking into coding platforms that will allow us to reuse some of our code if possible but it seems we will just need to budget more time to this project to develop for both Android and IOS.

The rest of our problems ended up being small things like needing to change the name of the project in the requirements doc or figuring out which platform we will be coding on. These all had simple solutions like going in and adjusting the name and looking up the pros and cons of the different platforms. Occasionally we had the problem of running a little close on the due date of papers and had a struggle of getting them out and signed in time. This was solved by communicating with our client in advance so they knew we were on a tight timeline and having quick turnaround on when the documents were sent and received.

Finally, our last thing we need to work on is learning about app development. Notice, I did not say that this was a problem. I would say it is more of a hurdle that we need to get over. We have all of winter break to get the fundamentals down along with starting the framework of our app. I think come early January that we will be set up well with our project.

### III. WINTER 2017

#### A. Courtney Bonn

At the beginning of this term, I had a hard time getting started on this project for many different reasons. First off, though it was encouraged, I ended up not being able to work on research or the project itself during Winter break due to family obligations and working. Not doing any of the research I had hoped to do meant I felt behind before the term even began. Second off, the research and learning how to use Xcode and Android Studio has been much harder than I expected.

During the first week of school I was able to get Android Studio downloaded and began working with the program. Because I'm also in Mobile Development, we are using Android Studio in that class so I began learning Android studio in an assignment for that class. We also began the iOS app for our project. At first, we decided to focus on one app at a time so we would not overwhelm ourselves and we decided to go with iOS first. Kevin was able to get a skeleton iOS app that we were able to pull to our local computers from Git and successfully run and view updates from each other. This week I got in touch with our client, whom we had not spoken to the entirety of Winter break. We updated her on our progress, letting her know that we have just started on the projects. At this point we agreed that we did not need to meet face-to-face because neither the client nor us had any deliverables to show each other at this time.

In week 2 I shifted my focus to Xcode and learning Swift. While Swift is similar to other languages I have learned, I have never coded in it and still need to learn the proper syntax. Not only did I need to learn Swift, I also had to learn how to use a new interface—Xcode. I began following a Swift tutorial provided by Apple that helps users learn how to use Xcode and build a Food Tracker app. I did not finish the tutorial, but I got far enough in it to assist me in working on our own application. I also did some additional research on the church's new website. The new website seemed to be a Wordpress website, which would help in terms of pulling information from the website directly to the app. Before I did anymore research on using Wordpress' API, I emailed our client to verify that I was correct and that we would be able to have access to the Wordpress site if needed. If I am correct and we are able to pull from Wordpress' API, the web development team would only have to update their website as normal and the changes would be reflected in the app. At this point I decided to wait until I heard back from the client and then I would continue researching how to use Wordpress' API.

The third week of term is the first week I actually began working on the iOS app. My goal was to get Calvary's logo up on each page. While this seems like it would be an easy task, it actually was rather difficult and took me two days to accurately do it. In order to have the images show up centered on every device and not just the device I was working with, I had to use constraints on the images. I was having trouble setting the constraints correctly and the images were not in the same spot on every page. Eventually, I was able to successfully get the logo centered on the top of each page in the app and it will stay centered no matter what the screen size is. I received confirmation from our client that the new website is Wordpress and we can have access to its API. The client will work on getting us access to the Wordpress site and until then I will continue researching how to use its API.

As of week 4, we are still waiting on the Wordpress credentials from the client. In order to not lose much more time and get behind, I began working on getting a JSON parser up on the bulletin page. After much research online, I was able to find a simple JSON parser. Using the parser to have the JSON data print to the console was easy, but figuring out how to display the data on the app itself took quite a while. Eventually I figured out how to display the data and now it is printing onto a large TextView. For now, I am pulling the data from <https://jsonplaceholder.typicode.com/todos/1> which is just a website that allows users to test pulling JSON data. Currently all that is displayed is a title and description, though they are both

in latin. Even though it is not usable information, once we obtain the Wordpress API we will be able to replace the above URL and pull the correct information.

At this point in the term, our group started to struggle a little bit with the development. We had not started the Android app so we were feeling very behind and not sure how we would successfully finish both apps in less than two months. Max discovered an issue with our storyboard that was greatly affecting his ability to add code to any app page past page two and this problem took a while to diagnose. Once we figured out what was wrong, Max was easily able to finish his task on the iOS app. We gathered our confidence back, but we still knew we were behind in the Android portion. I started the Android app and created all five pages we needed. This took longer than I expected because adding pages to an Android app is not as straightforward as it is in Xcode. I was also able to implement a sidebar navigation which pointed to each page. I am still doing research on how to parse and display JSON data using a REST API. The information I find regarding this subject is more complex than is needed for our project, so I have not been able to solve this part quite yet.

Coming up in our project is finishing each page and working on getting the Wordpress API information from the client. We also need to set up user studies to test our interface. It is too early right now to set these up since the basic functionality of each app has not been completed yet. My prediction is we will finish the iOS app sooner than we will finish the Android app, which will allow us to focus our attention on the Android app. Our client would also like us to research if we would be able to offer the applications in different languages. At the very least they would like the welcome message on the homepage to be presented in several different languages. After we have the basic functionality finished on the apps, we can decide if we are able to offer different languages for the app.

#### *B. Max Dimm*

Ok to recap briefly what our project is about, we are working with Calvary Corvallis to create both an IOS and Android app that will work independent of the website to be the connection tool between the congregation and the church. This app will deliver the most pertinent information to the members of the church in one easy to understand and centralized location. We are making both of the apps natively and will be trying to keep the interfaces as similar as possible.

All of our group members started this project with little to no app development experience. This made starting the project pretty daunting as we had not taken on something like this before. Our progress was slow at first but in retrospect I think that makes sense. It takes a while to get your reigns about a project and start to understand the language you are using. It took until about week 3 to start seeing major changes in our skeleton version of the app. I spent a lot of time personally watching Youtube tutorials and reading online resources about coding in swift.

On my end I am in charge of primarily working on the sermons page and partially on the donations page. The purposes of those pages are to allow the users to view live and past sermons from within the app, and to be able to donate to Calvary Corvallis directly. Originally I had a good idea of what I wanted to do with our project but struggled with an issue of our viewcontrollers not being connected to our story board in x-code. This made implementing what I wanted to do difficult as I spent over a week trying to diagnose the issue which was more difficult than expected because of my lack of background knowledge. Week four I was able to figure out the problem and get it set back on track. As of right now I have the video functionality up and working on our app and am working on the users ability to navigate past sermons. One issue Im currently facing is the live functionality is something that the video hosting service will not allow without a more premium membership from Calvary Corvallis. So I spoke with their tech team and they are working on finding a solution to that for

me. They explained that they will either be upgrading their account to allow live functionality, or also hosting their videos on another platform like Youtube to allow the users to watch their sermons live.

So as stated above I am still working on getting the navigation of past sermons onto the page below the player, but believe I have recently discovered the method I will be using to implement it. I want to test my method on a separate project I have been working on alongside our primary one. This is what I would describe as my testing grounds and allows me to remove code and make more drastic changes without affecting our main project.

Looking forward to what I have to do once I finish up the sermons page I'll start looking into both the donations page and beginning the android development on android studio. My personal preference would be to finish up the donations page first so I could focus completely on the android app. Our plan with the donations page is to open up a web view of the existing donations page. The reason we're going with that is that we don't have to mess with the back end of the authorization of our app. This would mean we would be in contact with their company information and we would like to avoid that if possible. Their tech team agreed on this matter.

In regards to starting up the development of the android portion, I need to do some preliminary research on the differences between x-code and android studio. Because we want the app to function and feel as similar to the IOS app, I want to implement the video and the donations the same way as much as that is possible. This is doubly advantageous because it will be the most familiar way for me to accomplish the implementation. I ran into the issue of the android studio simulator not working with my AMD processor, but we quickly figured out a solution. We found that if I have an android phone, that I can plug in and use my phone to test the app. This is a huge improvement on even the x-code simulator which takes ages to run on my five year old Mac.

### *C. Kevin Stine*

I started some initial development for this project towards the end of Winter break since I was pretty busy traveling and working for the first few weeks. Initially I wanted to have the basic framework for our iOS application laid out so that when we returned from break, everyone was able to have an idea of the basic layout. I started off by creating a simple tabbed application which had 5 individual pages for each of the sections that our project needs. Getting the tabbed application setup was pretty simple, however I ran into a few issues when I created the various ViewControllers. When you initially create a tabbed iOS app, it creates two view controller for just two tabs. Since we needed to have 5 total pages, I had to go in and manually create the swift files. Once I created them however, there weren't communicating properly with the main storyboard. It took some tweaking to make sure that they were all working properly but eventually I got it to work.

I began development by finding some icons that we could use for our tab buttons rather than using the built in square and circle icon. I found some icons for each of the pages and set those as the default icons. Once I actually got started on development, I chose to work on the events page. I began by following the iOS Swift tutorial which setup an application for a FoodTracker app. The food tracker layout was pretty much exactly what I was envisioning for the events page, so it worked out perfectly to have a guide. As I followed the swift guide I ran into a few issues getting the tableViewController to work properly with the eventViewController that I had created. Since I haven't had much experience working with mobile development in the past, I struggled to figure out how the two were supposed to interact with each other. Despite trying to troubleshoot and debug for a while, I was still unable to get the tableViewController working correctly, so I just need to spend some time reading through documentation to figure out how the table class should interact with the view controller.

After getting frustrated with the `tableViewController` not working with the `tableViewCell`, I started working on the second part of the events page which was the code to parse the xml data from CCB. We got the admin credentials from our client which allowed us to pull information from the public calendar of Church Community Builder. I wrote a quick python program that connected to CCB and got the file and was able to parse the event names, dates and times. I began writing this in swift using a playground for testing purposes which allowed me to see quick updates as I made changes. The basic thought behind the XML parser is that I need to be able to first connect with the CCB api in order to access the public calendar. To do this, I'm using the builtin `URLSession` and passing in the full length URL to connect to the database. The way that the API for the public calendar is setup, I need to pass in the start date (or end date if needed) to the URL so that it knows which date to grab the events from. In order to do this, I'm using the `NSDate()` and `NSDateFormatter()` to get the date to be formatted in a yyyy-MM-dd configuration (year, month, day). This is the way that the CCB api needs to have the date, so I get the current days date and append it in the correct format the URL. Swift has a little more complex way of parsing through data than python, so I'm still currently working on getting the functionality behind the actual parse to work. I'll need to implement a way to parse through the XML and grab the event name, month and day for the initial list view. If the user decides to click on the event to see more details, I'll parse through the XML and grab any more detailed information that might be relevant.

Since I haven't done any mobile development before, it's taken a bit of time for me to get comfortable using Xcode, understanding Swift and figuring out the builtin functions that we have at our disposal. Now that I feel like I have a pretty comfortable grasp on things, I just need to sit down and dedicate time to getting the XML data parsed and getting the `tableViewController` working. Once I've got that done it should be pretty simple to get those values passed over to the `IBOutlet`s and into the text placeholders which will display the information to the users. After getting that completed I'm going to begin on the Android version of the app which will hopefully go a bit smoother since we now have some basic mobile development practice. I'll likely spend most of my time getting a table view setup for Android and will hopefully be able to port over the XML parsing program without having to redesign it too much.

## IV. REFLECTION

TABLE I  
RETROSPECTIVE ON FALL 2016

Time	Positives	Deltas	Actions
Week 3	We met with our client for the first time in week 3. We found out a more detailed idea of what they wanted from us. Also we got to meet 2/3 of their development team.	No changes needed as of week 3	We needed to become more familiar with working with LaTeX.
Week 4	We had a second meeting with our client. We met the last developer on their team. He had minimal app experience but had an apple developer account.	No changes needed as of week 4	We need to start thinking about problem statement document.
Week 5	We met our TA, Vee, for the first time. We also finished up our problem statement.	No changes needed as of week 5.	We need to figure out if we need to develop one or two apps for the different platforms.
Week 6	We focused on finishing our requirements document and planning a timeline for the rest of the project.	Our client wanted us to change the app name to its official name in the requirements document.	We need to start researching our technology review and begin the design document.
Week 7	We began working on our technology review and splitting up the parts of our system.	No changes as of week 7	We need to continue researching the different technologies and keep researching our respective parts.
Week 8	We finished our technology review and began looking into the design document.	No changes as of week 8	We have to put a lot of effort in our design document in order to get it to our client on time.
Week 9	We got a headstart on the design document, but didn't get as far because of the Holiday.	No changes as of week 9	With the term coming to an end, we have to finish up the design document and send it to our client.
Week 10	We finished up the design document and met with our client for the last time until January	We do not have much knowledge in mobile development	We need to continue researching iOS and Android app development so we are more prepared for the project.

TABLE II  
RETROSPECTIVE ON WINTER 2016

Time	Positives	Deltas	Actions
Week 0	Got the framework started and began learning xcode	This notes the start of our IOS development	We all began the process of familiarizing ourselves with x-code and Kevin started the IOS project
Week 1	Courtney and Max tested our ability to build the framework Kevin had created and added our work to github	no code changes this week	Got the starting code distributed to the whole team
Week 2	We delegated which parts each of us were to complete	We now have direction and started researching relevant methods to accomplish our goals	Began the process of teaching ourselves the relevant swift functionality
Week 3	Began development of main pages	have functioning homepage, bulletin (awaiting wordpress API) and running into issues with sermons page	Closer to our goals with a few hiccups
Week 4	Figured out issues with sermons functionality and began progress report	Overall tuning of the app and still awaiting client to send us more info that we need to continue	Beginning android project and the donations page
Week 5	Met with client and talked over the next steps and our needs	Able to get additional information from our client and figure out our direction	Delivered a prototype to our client and did mid project followup
Week 6	Started working on progress report and created onenote page	We began the process of creating a digital paper trail to our work and changes	Finished our progress report, noted some changes to our tech document, and created our presentation