Senior Project Church App

David G Vidic

North Central University

CSCI495: Senior Project

Neely Tamminga

5/31/2024

Senior Project Abstract

For this project, I worked with my home church Christian Center Slovenia (KCS) to create a Church App. This project serves as the initial phase of the mobile application. This project's primary goal is to have a functional version of a mobile application that the church can use along with the primary requested features. Among the requests is the compatibility between both Android and IOS devices. To achieve this, I chose to use the React Native framework, which was completely new to me at the start of this project. Many features have now been implemented with some still to come. The target audience are KCS members and the Slovenian population, which means that the app will be primarily in the Slovenian language.

Outline

Size and Scope

For this class I am aiming to complete phase 1 of the project, which is to deliver a working prototype. This phase will serve as a minimum viable product for the final app. The goal of the final app is to be a central hub where church members can access all the church resources and information. I will be working with the Church to continue development on the App outside of class. This is due to the sheer size of the project, which would typically be done by a team of developers. I am also learning several new technologies to create this project, which adds to the time required for development. I am not expecting to have completed all the requested features by the end of this class. I will be focusing on the events calendar feature and go from there.

Product Goals – Deliverables

This mobile application will be used by Christian Center and their church members. This app will serve as an all-in-one place where people can see upcoming events, recordings, and other useful information. KCS requested 16 features for this application of differing levels of importance. For this project I will be focusing on the most important features for the app.

Requested Features

KCS initially provided me with a list of 16 features that they would like in their app. This full list of features is unrealistic for such a project with a single developer. For this reason, I communicated with KCS and got a narrowed down list of features that I can focus on. For this project I will only implement some of them due to time constraints. Here I go through the requested features in order of importance and explain their function and purpose.

Events

The first and most important feature for the KCS app is the Calendar of Events. This feature will be the primary focus for the application and will be the landing screen. This page will allow users to browse through upcoming events. It will also allow the users to click on events and receive information about the event such as the time, date, and location. There are a lot of ways this feature could be implemented, and I am left with the creative freedom to choose how to do so.

Sharing

Social media sharing capability is the next requested feature. This will allow users to share video and event links from the application across social media. KCS has the biggest Christian online presence in Slovenia and is dedicated to spreading the Gospel on social media.

Notifications

Push notifications for events. This is a standard feature of modern apps and will be included in this one. The reason KCS requested this feature is because they want to be able to let people know about upcoming events and live streams. It can also allow KCS to alert and notify others about a start time change of an event, a canceled service, sermon updates, prayer requests, a last-minute special event, change in service times.

Contacting

This feature may be more difficult to implement and may not fall into the scope of this project or phase of development. KCS would like to allow users to interact with each other on the app and for members to message each other and in messaging groups. This would be useful for leaders planning events and house group members. This feature could be a whole project on its own.

Video and Radio player

KCS currently upload church sermons and other videos to YouTube and would like to integrate their videos into the app. The videos could also be tied to events such that when a user looks at a past or live event, they could see the corresponding recording. The church also runs an online Christian radio which currently can only be accessed on third-party sites as well as at radio.sprejmi.si which does not have a good-looking front end yet. Having the radio built into our app would allow listeners to listen in the background and avoid any unwanted advertisements from other third-part applications.

Other

Some of these features should be easy to implement and others could be a separate project. Other than the previously stated features KCS also requested Mobile Giving, Social Walls, Daily Scriptures, a Contact feature, Audio Player, Note Taking ability, Bible reading plans, Livestreaming, Digital Forms, and TV casting. Which of these features I choose to implement will depend on the technology I choose to use to build the app as well as the ease of implementation.

Project Goals – Learning Experience

This project will give me an opportunity to grow as a developer. In this section I detail how I can benefit from this project from the possible challenges to the lessons and experience gained.

Challenges

This will be a challenging project and experience for me to complete. The first challenge I have had so far is narrowing down the goals for this project from the church. I had to figure out the church's desires for the application and my own for the project.

I am practically creating three different applications at once. To accomplish this, I will be using a multi-platform development framework. Since I have not done this in the past, I will have to learn new technologies and workspaces. Even with the help of a framework I will still need to test on separate platforms and implement platform specific features individually. As for the project scope for this project, I will start from scratch and need to complete each of the development stages on my own, which includes planning, design, implementation, testing, and launching the application. I may seek additional help if I get stuck at any of these stages.

Opportunities

This project is an opportunity to face any challenges and come out with a product at the end. Allowing me to link it on my website and resume and show off my real-world developer experience. With this project I am given creative freedom which will allow me to implement features how I see fit and in accordance with KCS. Planning out this project so far has already challenged me creatively to create a front-end vision of the app, and I am excited to implement it. In addition to having a project for my resume I will also have mobile developer experience in a technology I have not used before opening the door to job opportunities in that field.

Initial Research

For this project I have been researching different technologies I can use to accomplish the goals of KCS and my own. I have decided to go with a multiplatform framework that will allow me to write the codebase once and run it off each platform. There are interesting approaches that the frameworks take to accomplish this which will impact my decision on the one I chose to use. I have narrowed it down to Metas React Native and Googles Flutter. The other two options were to either code natively separately on both platforms effectively splitting the project into two code bases or use Kotlin Multiplatform which has a smaller community and less documentation aka learning tools. The main factor for consideration will be which approach best allows me to implement desired features.

React Native

React is a JavaScript library developed by Meta and has a wide variety of use cases. React Native is a framework that runs React code Natively on devices. React Native allows the developer to write code in an abstracted version of JavaScript and then translates it to the corresponding programs for each platform. The main advantage of React Native is that it uses the “Native” components of a device. Which means that if you have a button, on the Android version it will use the built in Android button and on IOS it will use the Apple built in button. This means that if Apple decides to change how buttons work on IOS the change will automatically take place on our app which allows for a smoother and more integrated experience. Another key advantage is that there are a lot of job openings for "React developers" in the job market. Which means that if I enjoy this technology, I already have experience to get a job. The main difficulty with this approach is that I have not programmed in TypeScript before and only have some experience in JavaScript.

Flutter

Flutter is a software development kit made by Google. It is also designed for cross platform integration. To use Flutter, I will have to learn to program in Dart, a language developed by Google. Dart is a C-style language and may be difficult to pick up at first. If I chose to go with Flutter, I would likely use a tool like FlutterFlow which provides a template and some basic prebuilt features. Unlike React Native Flutter, it does not use native operating system components. Instead, flutter simulates every pixel to the screen and recreates these components. This approach has better performance but can lead to a less integrated experience that becomes outdated quicker. There are Flutter developer positions, however they are less common and tend to be for smaller companies. The main advantage of using Flutter for me would be the ease of use of FlutterFlow.

Each of the two approaches has a large developer community and good documentation. This is vital for me to be able to learn the technologies and accomplish my goals. I believe each feature can be made in either technology, but I am still researching the specifics which will allow me to make the ultimate decision. At the end of my research, I chose to go with React Native. This decision was primarily due to the framework's superior compatibility, the ability to code native elements, and wider support for development.

Implemented Features

Phase one of the project is now complete. I have completed the functionality of several features and deliverables. This section serves as a list and description of each feature I have implemented so far.

Events Calendar

The Events Calendar is the main feature of this application and is the most developed deliverable of this prototype. The Events Calendar is given its own page. When you land on this page the current day is automatically selected. On the top of the screen, you can see a list of days of the week that are marked with a dot if there is an event on that day. You can swipe down from the week to see a bigger calendar that allows you to see months at a time and select future and past weeks. In this view you can see the current day, events on each day, up to a year ahead. Once a week is selected, you see a list of events happening on the day selected and upcoming events. If an event is live it is highlighted in red. You can click on any event to see more details about it. So far you will see the location address of the event, the time of the event, and a contact number associated with the event location. In addition, I plan to add map functionality to the location, more information about the event, and a link to virtual events and recordings of the event. For this feature, I used the React-Native-Calendars API, and wrote the data into a JSON file.

App Navigation

I based the tab navigation on social media apps and used the default Expo navigation settings. The navigation bar contains five pages that can be accessed at any time. The current page is highlighted in the nav bar. Each page has an Icon and text label. However, the text labels may be removed at some point. The five pages are the home page, video page, contact page, events page, and radio page. These are the major features that demand their own page. Each one of them is explained further in their own section. There is also a modal page accessible in the top right corner of the screen.

Home Page/Announcements

The home page serves as the landing page when you first open the application. The main page will hold announcements, general information for new users, and information on current events.

Video Page

This page is currently in progress and currently opens to our welcome video. In the future it will serve to display livestreamed events. It will also allow users to watch past YouTube videos from the KC YouTube channel.

Radio Page

This page holds a radio player that plays KCS web radio, which is currently hosted on radio.sprejmi.si but does not have a graphical front end. This page allows users to listen to this radio station straight from the app, listen in the background, and even with a turned off screen. This allows listeners to have a seamless experience and listen without ads which they might get on other web radio pages.

Contact Page

The contact page serves as a place for people to find important information. On this page people can find contact information for each individual church campus. On this page they can also access a prayer line and help line. Other information includes the locations and addresses of each campus, a church membership forum, and links to the church social media.

Light and Dark mode

This is a feature I implemented that was not requested by KCS. However, it is easy to implement using expo go, and the template I used automatically included this feature. I chose to keep it because it is a feature that I personally enjoy in most applications and believe that other users do too. It currently defaults to the devices theme colors. In phase two I plan to add a toggle switch for this option that will be accessible in an upcoming settings page.

Platform Compatibility

The app is compatible with both major mobile platforms (Android and IOS). React Native additionally allows for an in-browser web app. However, some of the APIs are not as compatible with the browser and will require some platform specific changes and will support less functionality.

Upcoming Features

Phase one is complete. These are the left-over features that will be implemented in phase two.

Notifications

Mobile Notifications is one of the main reasons the church wanted a mobile app over a new website. With a mobile app I will be able notify people of upcoming events and other important updates such as an event cancelation, moved date, or other changes. Users will be able to choose what notifications they receive depending on the location they attend. They will also be notified of updates on events that they selected to attend.

The other use of Notifications will be for out of app audio controls. The user will be able to pause or mute the radio station, navigate on a YouTube recording, or livestream, and interact with functions on virtual events. Although, this last option is not fully planned yet.

Users And Accounts

In order to personalize a person's experience, I will need to implement user accounts into the application. This will allow for personalized notifications and interactivity with events and other users. This will also allow differing levels of access for users. Allowing organizers to add and update events, and announcements from within the app.

Visual Updates

Most pages still need visual changes. The changes will be implemented in phase two. One of these changes is the color coding of links and contacts. Other UI changes will also need to be implemented for aesthetics.

Event Interactivity

Each event will be able to be interacted with. The first interactive feature will be an attendance checkbox. This will allow event organizers to estimate how many people will attend any event. It will also allow for personalized updates on the events. Another interactivity feature is upvoting an event and commenting or chatting on it. This will allow interested users to ask questions about the event and discuss it with other users. This chat feature will require moderation and differing levels of access.

Settings Menu

Every decent app has a settings menu. This menu will include an option to toggle light and dark mode, language preferences, and notification settings. It may also include other options currently unforeseen.

Assumptions and Revaluation

At the start of the project, I had no idea how far I could get and how much time and effort it would take to get this far. This is due to the nature of the project and its novelty. At the end of phase one I ended up exceeding my own expectations for the project. Because of my unsureness, I spent more time in the research phase of this project than I would have liked. I struggled to decide what technologies to use for this project. Once I chose the technology to go with, I enjoyed learning it and working on the project. After using it for a while I believe that I made the right choice, and I could have made it sooner and with less information. The main challenge was looking through the conflicting information and recommendations from online sources.

I knew that I was going in blind even at the start of the project. My main assumption was that it would be difficult. I now know that I overestimated some of the challenges that I would encounter. Some of the features were a lot easier to implement than I expected, once I figured out how the technology worked and found good solutions for them. Other features took a lot longer to implement than I initially expected. However, this was the nature of the project. I could not expect to make accurate predictions without working on this type of project before. In the end I faced the challenges I expected to face and reached the goals I hoped to achieve.

Lessons Learned

This project taught me the full cycle of development. I learned how to plan out a project, design it, and implement it. Of course I have done most of these things separately before. Now I put them together to reach one coherent goal. I learned the importance of communicating with a client. I also learned less abstract things like how to code in TypeScript, and how to store information using JSON. The tools I learned to use are valuable to me when it comes to the workforce, or even freelancing.

In conclusion, I built a Church App. Extensively planned out a project and negotiated its deliverables. I researched the technologies that would be right for this project and learned to use them. In doing so I had to make decisions concerning what routes to take the project, the technologies to use, and what the ideal focus of the project was. I implemented many features and planned out the rest of them. I split the project into phases and took a realistic approach to completing each one.

Project Timeline

Phase 1 - By end of class – Working prototype

This is the initial phase of the project and ends at the end of class. The goal of this phase is to have a minimum viable project. To have a project complete with functional key features. The focus of this phase is not on graphics and animations but only on the functionality of implemented features.

Phase 2 - By end of July – Published on Google Play Store for testing.

During this phase I will be implementing additional features and details. I will be cleaning up the UI and implementing graphical features and settings. At the end of this phase I plan to have completed the functionality of the application, and get it published on google play. I chose to test on Android first because the majority of Slovenians use Android devices, and it is cheaper and easier to get an app published on their Appstore.

Phase 3 - By end of year – Published on Apple Store and Google Play store for public use.

As phase 3 starts, the application will be released to church members for testing and feedback purposes before a wider launch. I plan to gather feedback with an in-app support feature, word of mouth, and questionnaire.

Sources used in Research

Build apps for any screen. Flutter. (n.d.). https://flutter.dev/

Dart programming language. Dart. (n.d.). https://dart.dev/

Expo. (n.d.). https://expo.dev/

Krščanski Center. (n.d.). *Sprejmi.si*. sprejmi.si. https://sprejmi.si/ (KCS current website)

React native · learn once, write anywhere. React Native RSS. (n.d.). https://reactnative.dev/

FlutterFlow. Showcase FlutterFlow. (n.d.). https://flutterflow.io/showcase

The Six Most Popular Cross-Platform App Development Frameworks. Kotlin programming language. (n.d.). https://kotlinlang.org/docs/cross-platform-frameworks.html#kotlin-multiplatform

YouTube. (n.d.). *Fireship*. YouTube. https://www.youtube.com/@Fireship

YouTube. (n.d.). *Theo*. YouTube. https://www.youtube.com/@t3dotgg

Sources used in Development

Build amazing things. NPM. (n.d.). https://www.npmjs.com/

Expo. (n.d.). https://expo.dev/

Learning TypeScript. TypeScript. (n.d.). https://www.typescriptlang.org/docs/handbook

React native · learn once, write anywhere. React Native RSS. (n.d.). https://reactnative.dev/

Savin, V. (n.d.). *Virtual events app with react native*. Not Just Dev. https://www.notjust.dev/blog/2022-11-25-virtual-events-app

Wix. (n.d.). *React native calendar components*. GitHub. https://github.com/wix/react-native-calendars

YouTube. (n.d.). *Not Just Dev*. YouTube. https://www.youtube.com/@notjustdev