CPSC 490: Family Archives

iOS Application to Preserve and Archive Family History

By: Zoe Henry

Advised By: Professor Ruzica Piskac

Can be found on Github:
https://github.com/zoehenry/seniorproject
Abstract

For my project, I have created an iOS application that helps users preserve and archive family history. The application has two main features—uploading photos and uploading audio. The user can upload either type of media, name the event or person featured in the media, and write a description about it. This will allow users to click through family photos and listen to stories across generations with losing family history due to damaged or lost old photos or the overwhelming volume of photos stored on their personal device.

I chose this project because I wanted to expand my knowledge of software engineering by learning mobile development. I chose to focus on learning Swift because the language is so well documented and most of the mobile development I’ve seen in industry has been iOS development. As a result, I felt that I would have plenty of resources to use for independent study, while also preparing myself professionally for working with iOS if I chose to pursue mobile development in the future. This project has helped me expand the list of tools I feel comfortable using, as well as given me a project that I am excited to share with my family.

The idea for this application was inspired by many conversations I have had with my parents about the importance of preserving family stories and how the digital age has made it too common to take photos and forget about them. This application allows users to organize photos and audio recordings in one place without being overwhelmed by the quantity of documents.

Introduction

Background

With the plethora of technology today, taking photos and videos to document moments and share them with the world is easier than ever before. Unlike the physical
family photo albums containing pictures from specific events with labels of what event it was from and where it occurred, phones tend to generalize photos simply by day, the people who are in them, and the geolocation. Because of the constant availability of cameras and immediate storage of photo albums on computers and phones, it is easy to become overwhelmed with the volume of photographs on a given device or in the cloud and lose track of when the photo was taken and the story behind it.

The virtual album lacks the personal touch of handmade photo albums that are able to create a story of a family member or family event. In addition, physical family photo albums are becoming less and less popular, and while they may exist, they are susceptible to damage.

For my senior project, I wanted to create a solution to this problem through a mobile application that helps individuals preserve and archive family history. The idea was to create an application that could virtually provide the personalization and documentation of a physical family album.

Goals of the Project

I wanted to work on this project to broaden my software engineering skills. Many of the computer science classes at Yale focus on the theoretical aspects of computer science, so many of my practical skills come from outside work and projects. I had worked with web applications several times, but had no experience with mobile applications. In addition to learning Swift for basic iOS development, I also researched and implemented technologies that allow photo uploads, audio recordings, and file storage. My goal was not only to create an application that makes it easier to document family history, but also to further my own learning. While I have learned on the job during my internships, I believe that being able to teach myself new technologies is a valuable skill. Most of my learning has always been at least somewhat guided, so I felt that this independent research and implementation would be helpful to my own professional development.
Learning Process

Resources

To begin this process of learning Swift from scratch I used several resources. In my project proposal I outlined courses and tutorials that I thought would be helpful, which served as a useful jumping off point. I also did a lot of research as I ran into roadblocks and changes in syntax due to upgrades along the way (StackOverflow was very helpful here!). Please the final section of this project report for a full list of the resources I used.

Learning Journey

I used February and March to teach myself the basics of Swift and create a few practice mobile applications. I started by watching the first half of the Stanford CS193P Course, Developing Apps for iOS. These lectures were very insightful and helped me understand the basics of Swift. I was struck by how UI focused iOS development is. Watching the professor’s demonstrations and explanations showed me that a lot of the development is done in the storyboard, the UI side of the application, and then connected back to the code through a series of clicks and declarations so that each part of the UI has a programmatic function.

These lectures also showed examples of the Model View Controller framework in Swift. I was familiar with the MVC framework from web development, so was happy to see the same concept repeated even with different technology. This reinforced for me what is often said about why the Yale computer science curriculum is so theoretical. Which is to say that while the technology may change, the underlying concepts remain the same. Because I had worked with MVC before, I felt that I would be able to translate those skills to mobile development. Knowing that these concepts are transferable gave me more confidence going forward in my learning. However, there were also concepts that were different. The practice of making the UI first and then using outlets and
actions to create a function was very new for me. In my experience with web development, I have often focused on the model and controller first before creating the view. I was starting to learn that in iOS, the flow of development was reversed from what I was used to.

As I continued to watch these videos, I felt that I needed to actually start implementing the concepts about which I was learning to really learn and understand how everything worked. I first implemented my new learnings in an example calculator application. This practice app was helpful for me because I learned how to directly connect a button click to a resulting action of populating a screen and then making mathematical functions.

While I definitely got an understanding of how typical iOS development works, I still had a long way to go in terms of understanding how to make my UI actually look good.
Seeing that my frontend skills still needed a lot of work, I did research on AutoLayout and Stack Views which help objects in the UI stay aligned with each other even when changes occur on the screen. I also had to practice making constraints so that the objects would stay where I wanted them to be in relation to the edges of the screen. I implemented these findings in my final project, and the design and feel of my UI was significantly improved.

I also found an Apple tutorial which gave me insight on how to upload photos and persist data. These skills were directly transferable to my final project, and were used as the basis for my photo upload page.

Application Walk Through

In this section, I will go through each of the pages in my application and describe the functionality. In addition, I will briefly discuss what technologies and plugins made this functionality possible.
These are the login in pages. When you first open the application it takes you the orange page on the left which invites you to login. Pressing the login button takes users the Firebase Authentication service, which has a built out authentication functionality for email, phone number, Google, Facebook, and more. I chose to authenticate using email because I felt that it would be easiest for people who are older who might be using my application. The screenshot above on the right shows an example of a user who has already made an account and just needs to sign in. The Firebase functionality for creating a new account is quite similar.
Once logged in, you are taken to the page on the left which is a wrapper page for the application. Once the user clicks on “Your Archives,” they are taken to the homepage of the application. This page houses the list of all entered events and people, a search bar to look through the events, and buttons to add new media. The events can be searched by name and are persisted between sessions. They can also be deleted when the edit button is clicked if desired.
These last two pages show what it is like to both upload and edit new media. On the left is the photo view and the right is the audio view. The two views are quite similar in layout. They both have a label on top to name the media being uploaded and a space underneath the media for a description of the photo or audio recording. The photo view allows the user to access their Photo Library and select any photo to be uploaded. This photo is then used in the list item shown on the homepage. The audio view allows the user to record a conversation or story through the internal iPhone microphone and play back this recording both immediately and when the page is revisited.
Development Process

Initial Thoughts

As I was starting my project, I had a range of thoughts and questions. My main focus was the flow of my application. How I would go from launch screen to menu to edit and view pages and how many controllers I would need. It helped me to draw out an index card for each page and lay out the flow from page to page on my desk. On these cards I wrote down things I needed to add. For example, the page for uploading photos had “ImagePicker, add TextField for Name, segue to main page?”. These were just quick notes for me to remember the flow of my application as I developed. I also had to make decisions about what devices I was optimizing my application for. I decided to make my app available solely in the portrait orientation and optimized for the iPhone X; however, it also works on iPads for users who prefer a bigger screen.

Implementation

I started off by working on the upload pages for photo and audio. I implemented the photo page via storyboard and connecting buttons and text fields back to the controller after the UI design. In contrast, I implemented the audio page UI programmatically. By developing the views in two different ways, I was able to see how they are different and which one I would prefer to do in the future. Using storyboard was much easier, because I was able to get all of the sizing and relative positioning correct fairly quickly. The programmatic implementation was much harder. I had to explicitly say how far apart I wanted each button and text field instead of just placing them directly into the UI. It took significantly more trial and error.

After setting up those pages, I made a table view to function as the menu for each of the photo and audio objects created by the user. I then added search, edit, and deletion functionality to this page so that users can interact with their already created objects. This part was actually much easier than I anticipated. Making a list to display
stored objects only requires using a TableView object and then populating it. The most
difficult part of this section was figuring out how to have each TableViewCell segue to
the correct view depending on if it had audio content or photographic content which I
will talk about more in the Challenges section below.

The next and most important addition I made was persisting the data. To do this,
I had to make a PersonEvent object with the properties I wanted to save. Then for my
audio and photo controllers I had to make sure to create a new object or access the
object being edited and save any changes made. The object is then saved and used to
populated the table view whenever the application is opened. This was pretty
straightforward for the photo controller, but was more challenging for the audio
controller which is less well documented and involved much more set up due to the
medium and number of parts involved in the process of setting up both the audio
recorder and the audio player.

Then I implemented a Login functionality using Firebase Authentication. I was
really intimidated by having users set up an account and storing emails and passwords.
However, Firebase makes it really easy to add sign up and login functionality to an iOS
application. After setting it up, I was able to insert Firebase’s pages and database into
my application between my landing page and my menu page so that each user has more
security around their family archives.

Lastly, I worked on the design of my application. I do not have a natural eye for
design so this part was also new for me. I met with one of my friends who enjoys
graphic design to get tips on what I could make better. She suggested choosing a color
scheme and using minimalist buttons to make the app look clean and approachable.
During this process, I made sure that all buttons were placed where I wanted them to be
and played around with different colors that made the app look both fun and
professional. I also added a logo icon to create a complete look.
Testing

I have not had much experience with testing so I focused on the most simple form of tests, unit tests. I used these to test whether a PersonEvent object had been properly created, and whether or not it had a name property. If the name property was not filled out, I tested to make sure that the object was not created. If the name property was there, I tested to make sure that the object had been saved. This testing ensured that only objects that were named could be created. Going forward I would like to add more test coverage to make sure that the UI works as expected instead of manually testing all of the UI functionality. I found the documentation on unit testing to be more straightforward, so I would have to spend time looking at the UI testing documentation and tutorials that ensure that segues are triggered by the correct objects, move to the correct page, and occur at the correct time.

Challenges

While I really enjoyed the process of implementing the application, I ran into quite a few roadblocks ranging from big to small. One of the first issues I had was when learning how to implement the audio UI programmatically. With the photo UI, it was very easy to put placeholders in the text field because I could go to the object inspector and just type “Enter Name Here” where there was a field for the placeholder. When doing this programmatically I had to find where I could set a placeholder for the TextField and just temporarily set the TextView text to create the look of a placeholder. I then had to check for changes to the text to get rid of the placeholder and save the new text. This required more conditional statements and logic than when implementing text fields from the storyboard side first. I appreciated getting to see both ways of doing this and understanding why many iOS tutorials started in the storyboard, which at first I thought was just a weird quirk of iOS development. It’s simply an easier way of implementing the same thing.
Another challenge I had was navigating to the appropriate edit page for each person or event. Some of the objects were connected to the photo controller and view, while others corresponded with the audio controller and view. I had a lot of trouble distinguishing between the photo and audio segue when clicking on the same object to trigger both segues. In the end, I had to make a separate method to check what properties each object had, before calling the performSegue function, to make sure that each object was going to its corresponding page.

The biggest challenge I ran into was the audio page. The process of being able to playback audio after recording was challenging because I had to figure out how to set up the audio player, find the recorded file, and play it. This was even harder when the user returns to the main menu and then goes to an audio page. It took a lot of trial and error to make sure I was getting the correct path for the audio file I was searching for, but I eventually got it to play the audio even when it was not recorded immediately prior. I was surprised by how many methods were needed to make an audio recording in contrast to how easy it was to upload a photo by just dragging in a PhotoImageView.

Discussion

What I learned

Through this project, I learned about the development cycle of an iOS application from start to finish. I worked through planning out my project and the application flow in my project proposal and was able to expand and iterate on that in the execution. This process was also extremely enlightening in terms of figuring out my learning style. Because most of my learning before this project has taken place either in a classroom or from a mentor on the job, this was my first time doing a completely self guided project. I had to create my own learning and execution timeline. This project taught me that I like to first try to understand the concepts then implement them in small examples before trying a full scale application. I discovered this by making a few small test
applications before building out my final project. It also taught me that sometimes it is helpful to do things both the easy and the hard way to understand why the easy way works and not take it for granted. This lesson was made obvious to me in the storyboard versus programmatic UI case.

Future Work

Going forward if I were to continue work on this application, I would add more testing to the frontend of the application as well as add more sorting functionality to the menu page. As I mentioned in the testing section, I primarily focused on unit tests to confirm that objects were being created properly. I would add more tests on the UI section to ensure that aspects I implemented programmatically were appearing and working as expected. This would give my testing suite more coverage and give me more end to end testing. In addition, I could add a date property to my objects and allow users to search by date and not just name on the menu page.

This week, I spent some time reflecting on my project and what I have accomplished. While it is in no way the best mobile application out there, I am proud of the strides I have made in my self education and iOS development as I now feel equipped to teach myself more new skills and technologies.
Sources

Basic iOS Development

Stanford Course:
https://www.youtube.com/watch?v=Lx4Ohhsc3ho&list=PLPA-ayBrweUz32NSgNZdlo_QISw-f12Ai&index=4

Apple Tutorial:
https://www.youtube.com/watch?v=DNidR9sULfo

Search:
https://www.youtube.com/watch?v=wVeX68Iu43E

Login:
https://medium.com/@ashikabala01/how-to-build-login-and-sign-up-functionality-for-your-ios-app-using-firebase-within-15-mins-df4731f3f2f7
https://www.youtube.com/watch?v=brpt9Thi6GU
https://www.youtube.com/watch?v=JV9Oqyle3iE

Photo

https://medium.com/@deepakrajmurugesan/swift-access-ios-camera-photo-library-video-and-file-from-user-device-6a7fd66beca2

Audio

https://www.hackingwithswift.com/example-code/media/how-to-record-audio-using-avaudiorecorder
https://medium.com/@vikaskore/record-audio-in-ios-swift-4-2-a6a4d53e31b0