Abstract

This project was made in collaboration with Mustang News, Cal Poly’s campus news group. The goal was to create an app for Mustang News that would help them reach a broader audience and readership. The app will present stories found in the Mustang News newspaper and website in a mobile friendly format.

Monica Chavez-Martinez
Computer Engineering Undergraduate, Spring 2018
# Table of Contents

1 INTRODUCTION .................................................................................................................. 2

2 REQUIREMENTS AND STAKEHOLDERS ............................................................................ 2
   2.1 CLIENT REQUIREMENTS ............................................................................................. 2
   2.2 STAKEHOLDERS ......................................................................................................... 3

3 BACKGROUND & RELATED WORK .................................................................................... 3
   3.1 COMPONENTS OF AN ARTICLE .................................................................................. 3
   3.2 APP LAYOUT REFERENCE .......................................................................................... 4
   3.2 TECHNOLOGIES USED .............................................................................................. 6
       3.2.1 REST API, JSON, & HTTP Clients. ...................................................................... 6
       3.2.2 CocoaPods .......................................................................................................... 7
   3.3 RELATED WORK .......................................................................................................... 7

4 KEY DESIGN DECISIONS .................................................................................................... 8
   4.1 COLLECTION VIEWS OR PAGE VIEW CONTROLLER .................................................. 8
       4.1.1 Collection View ................................................................................................... 8
       4.1.2 Page View Controller ........................................................................................ 8
   4.2 WEB STORIES OR CUSTOM CONTENT ......................................................................... 9
       4.2.1 Use Web Stories & Expand the REST API’s Response ......................................... 9
       4.2.2 Creating a Subdomain ......................................................................................... 10
   4.3 HANDLING MULTIPLE REQUESTS FOR A SINGLE ARTICLE ..................................... 10
       4.3.1 Use Custom Database ......................................................................................... 10
       4.3.2 MetaBox Plugin ................................................................................................. 11

5 FUTURE WORK .................................................................................................................... 11
   5.1 ENHANCED PUSH NOTIFICATION ............................................................................. 12
   5.2 ADDING IMAGES TO THE ARTICLE CONTENT .......................................................... 12
   5.3 ACCESSIBILITY ........................................................................................................... 12
   5.4 CONTINUE WORK ON AUGMENTED REALITY (AR) .................................................... 12

6 REFLECTIONS ...................................................................................................................... 13
   6.1 REACT NATIVE / ANDROID APP .............................................................................. 13
   6.2 WORK ON THE PROPOSED PROJECT GOALS ............................................................ 13

7 CONCLUSION ...................................................................................................................... 14

REFERENCES ........................................................................................................................ 15
1 Introduction

Mustang News, Cal Poly’s campus news group, pitched the idea for a mobile application at the Senior Project Pitch Event Fall 2017. The goal was to create an app that would expand their audience and help the writers at Mustang News explore journalism in a mobile setting. Their main methods of publication were through the Mustang News website and the campus newspaper. The purpose of the app would be to create a new platform for journalism students to publish their stories and to help Cal Poly students stay up to date with the most current campus news. The first set of goals focused on the design of the app’s user interface with tentative goals to add features like push notifications and AR/VR.

2 Requirements and Stakeholders

The client for this app is the Mustang News Group. The stakeholders include both the Mustang News staff and, more broadly, Cal Poly students. Austin Linthicum, the Mustang News Creative Director/Designer, is the main individual involved with communicating the requirements for the app. After meeting multiple times with Austin, we decided what the main features of the app should be.

2.1 Client Requirements

**User Interface.** The interface had to be simple and intuitive. It was also important that the interface closely resemble website’s interface in order to keep a sense of consistency with their brand. Figure 1 shows the home page of the current Mustang News website. Within the next few months, Mustang News is planning on phasing out the website in Figure 1 and replacing it with one that better showcased their brand’s new color scheme (a small sample seen in Figure 2).

The navigation bar and the Mustang News banner were the main sources of inspiration for the interface design since they would be the two major components that would remain consistent in the transition to the site’s new design.

![Figure 1 A screen capture of the Mustang News website as of June 5, 2018.](image-url)
**Figure 2** A screen capture of the new Mustang News color scheme seen on a Safari browser when the page is bookmarked to the Favorites folder.

**SHARING.** Sharing capabilities were also a priority. Users should be able to seamlessly share the stories with one another. Ideally, users would be able to share stories on social media. This would be another way Mustang News could expand their readership through the app.

**INTEGRATION WITH WORDPRESS.** In order to make this project more useful to the client, it was necessary to incorporate the technology that the Mustang News staff was already comfortable with. All stories are posted on the Mustang News website using WordPress, so it was important to keep that interface available to the staff.

### 2.2 Stakeholders

Mustang News is the main stakeholder in this project. They would benefit by reaching a greater readership. Cal Poly students are also stakeholders in this project. More students can be informed about campus news and community events. While Mustang News currently has both a website and a newspaper, the mobile application would make the stories more accessible to students. Journalism students would also benefit by using this new platform to explore digital journalism.

### 3 Background & Related Work

#### 3.1 Components of an Article

Before any development started I had to understand the content I would be displaying on the app. I met with some of the Mustang News staff to gather what kind of information would be valuable to display in the app. They outlined the important features of an article that are used to grab the attention of the reader to encourage them to read the full article. These features are: the title, a featured image, and an excerpt (i.e. a brief summary of the article). Once the reader has an overview of the article’s content and chooses to read the article, we decided on the features of the more detailed view of the article. The first feature is the section name the story came from. The app will feature the following sections: Home (general news), Arts, Sports, Opinion (seen as Voices on the website), and San Luis Obispo (a section unique to the app). The remaining features are the date published, the
title, the featured image, the author, content, and an option to share the article. The app will share a link to the story on the Mustang News website, not the subdomain (more on the subdomain in Section 4.2.2).

Understanding these components of the story were fundamental to the design process. It was important to keep these components in mind when designing the user interface.

3.2 App Layout Reference

Throughout this paper I reference different components of the final app design. This section is a reference of the terminology that will be used in this paper.

**HOME PAGE VIEW.** Figure 3 shows home page view of the app. It is first screen the user sees when the app launches. Figure 4 highlights the key features that are used to summarize a story for users.

![Figure 3 Screen capture of the home page view of the app.](image-url)
**Article View.** Figure 5 shows article view of the app. Once the user clicks on the article summary this view comes to the front. This keeps the section title at the top of the screen in the app’s navigation bar. The sharing button is purposefully placed on a navigation bar that does not hide when the user swipes up. This addresses the client requirement to make sharing accessible. All the key features of the article are present on this screen. Images can have captions even though the image in Figure 5 does not. The remainder of the page is just plain text.
3.2 Technologies Used

3.2.1 REST API, JSON, & HTTP Clients.

Mustang News uses WordPress as their content management system. WordPress has a REST API that provides API endpoints for WordPress data types. These endpoints give me the ability to interact with site remotely by sending and receiving JSON (JavaScript Object Notation) objects. Using the WordPress REST API documentation and an HTTP client, I was able to map out the JSON objects that contained the data for each post [5].
3.2.2 **CocoaPods**

CocoaPods is a dependency manager for Swift and Objective-C Cocoa projects. It makes installing other open source libraries easy and fast. I used 3 pods for this project: Cache, DKScrollingTabController, and PushNotifications. [1].

- Cache was a library that helped efficiently cache any objects that conformed to Codable protocol. I created a Codable-protocol conforming struct to organize the information of each article that was requested from the site. Cache made it easy to cache the articles instead of trying to save all the data that came in.
- DKScrollingTabController was a pod written in Objective-C. The documentation could have been more detailed but overall it was easy to use. It is a horizontally-scrolling bar that is a highly customizable. This pod helped create the section bar in the app it seamlessly fits in with the design of the app.
- PushNotifications is pod released by Pusher. This company had very easy to read documentation. This is the only service that has the potential of requiring future payments. They have monthly-subscription-fee model, however, there is an option to use the service for free. The free version allows unlimited push notification for 2,000 devices. The Mustang News team decided this was sufficient for the volume of traffic they are anticipating early on.

3.3 **Related Work**

**The Mustang News Website.** One of the goals of this project was to expand the ability of Cal Poly journalism students to explore digital journalism. The Mustang News website already lets them do that. It also gives them helps give them broader readership than their newspaper alone would. The website, therefore, overlaps with some of the goals of this project. It does not take away from the website’s ability to meet the same goals it simply adds a new medium that can be used to reach the same goals.

**Previous Mustang News Mobile Application.** Cal Poly previously published an app with similar goals. This app, however, has not been updated in over 2 years. It was created by advertising students and is not regularly maintained. For obvious reasons, while this previous app overlaps with the goals of this current project, it is still beneficial to make this new app. Unfortunately, the app is still in the App Store under the name Mustang News.
4 Key Design Decisions

4.1 Collection Views or Page View Controller

One of the first major design decisions revolved around developing the user interface, specifically, the home page of the app.

4.1.1 Collection View

The first version of the home page used a Collection View Controller as the main component. This collection view had 5 horizontal cells since there were 5 new sections: Home, Arts, Sports, Videos, and Opinion. Each cell in the Collection View Controller had an embedded Collection View that grew vertically. Each cell in this Collection View contained the details of a single story. Each of the 5 sections loaded up to 10 stories, so there were 10 vertical cells per section.

This method met the functionality requirements; however, it caused a significant amount of delay on start up. All stories were requested when the app started up causing a long wait time from opening the app to seeing the content.

Another problem with this approach is that it had lots of code duplication. Each section cell had its own class file that would make similar function calls. Ultimately, the only positive of using this method was that it met the functionality requirements. Because of the delay on startup and lack of code reusability and readability this method was not used in the final product.

4.1.2 Page View Controller

The next approach to this problem was using a Page View Controller. This component handles the horizontal scrolling, so there is no need to have the Collection View Controller with 5 cells. “Page View Controllers present content in a page-by-page manner. They manage a self-contained view hierarchy. The parent view of this hierarchy is managed by the page view controller, and the child views are managed by the content view controllers that you provide”. [2]

For this project, there was only one content view controller. It was a Table View Controller that would only request stories from the database one section at a time. This method solved the delay on startup problem introduced by Section 4.1.1. It also cleaned up the code significantly since both the Page View Controller and Table View Controllers are easy to manage using the storyboards on the Interface Builder in XCode. Most of the functionality of these components could be controlled through the Interface Builder, thereby also reducing the amount of coded needed. Due to increased performance and code readability, this method was used in the final product.
Figure 6 shows a screen capture of the table view controller that holds the articles and its relationship to the view controller that hold the article’s content.

![Diagram of table view controller and article view](image)

**Figure 6** The relationship between the table view controller and the article view.

### 4.2 Web Stories or Custom Content

Originally, the app was designed to pull articles directly from the Mustang News website using the WordPress REST API. The site returned a JSON object with the article title and the content field as an HTML attributed string. This meant the HTML tags on the string had to be either stripped off or used to render the text on the screen. Another problem introduced with the attributed strings was the lack of customizability and originality. Simply displaying the attributed string meant that the app could become just a mirror of the website. According to Apple’s App Store Review Guidelines, “[apps] should include features, content, and UI that elevate it beyond a repackaged website”.[3] Keeping the attributed strings could make the app fall into a category possibly violating the App Store Review Guidelines. I attempted to address these problems in the following ways.

#### 4.2.1 Use Web Stories & Expand the REST API’s Response

While XCode supports rendering attributed strings, there was a problem when the string had attributes indicating an embedded image. Upon tapping on the story on the home
page of the app, the article view would come up and render the attributed text. This would trigger a request for any embedded images in the string. The images would render too large, too small, off-center, or not at all. Also, there were cases where the article contained not only embedded images but also interactive maps or graphs, causing more inconsistency when rendering an article’s content. Overall, lack consistency of all these forms of multimedia was a major problem. One quick proposed solution was to remove only this multimedia content from the string, however, the problem with this solution is that the text in the article often referenced those images/graphs/maps. The story would become unreadable if the content was just stripped off.

The WordPress REST API provides a few ways to edit the JSON response it sends to the mobile application. So, instead of using the provided “content” field in the JSON object, it is possible add a field to the object that has only the plain text string instead of the HTML attributed text. This, however, has the same problem as simply removing the multimedia; the articles could become unreadable.

4.2.2 Creating a Subdomain

A different approach to this problem was to create a subdomain for stories that would be posted to the app. There are some inevitable problems of this method. One, is that each story would have to be posted twice, once on the main Mustang News cite and once in the subdomain. Another, is that each story that had embedded media would have to be reworded in certain sections to account for the lack of embedded media. After discussing these issues with the Mustang News Group, they decided the subdomain was the better option. They mentioned they had enough staff, so the added work was not that great of a problem. The stories for the app are posted to app.mustangnews.net. The links shared through the app point to articles on the original webpage, not the subdomain.

4.3 Handling Multiple Requests for a Single Article

The JSON returned by the REST API contains the article’s title, date published, and content. However, the featured image associated with the story is not available, instead, a media id is returned. This media id is needed to create a second request for the source URL. Once the app retrieves the source URL it can download the image. This means that for each story a total of two requests must be made just to get the contents necessary to populate the home page view. This causes significant delay in rendering the content.

4.3.1 Use Custom Database

One option to fix this problem would be to use a custom database. This would give me the freedom to store only the relevant information of a post in a such a way that would make retrieving the data simple and efficient. It could decrease the size of the data being
requested improving performance. The WordPress REST API returns JSON object with lots of metadata that increases the size of the data request without providing any useful or necessary information. While it is possible to add to the JSON response, it is highly discouraged to remove any fields from the JSON object. So, a custom database seemed like a promising option. However, keeping the WordPress interface was an important client requirement so this method was not used.

4.3.2 MetaBox Plugin

MetaBox is a WordPress plugin that makes it easy to add fields to the JSON object returned by the REST API. Figure 7 shows the fields that I created using the plugin. This page comes up inside the WordPress page that the staff is accustomed to using. This plugin made development a lot easier since all text came as plain text with no HTML attributes. It also made the client requirement of keeping WordPress possible. And finally, it was able to send the image in the initial request. So each story only required one request.

![Figure 7 Fields created using the MetaBox plugin.](image)

5 Future Work

The first version of the app was released in the App Store on June 3, 2018. This version of the app accomplishes the fundamental customer requirements. Future versions of the app would benefit from the following additions.
5.1 Enhanced Push Notification

The version on the App Store does not have support for push notifications. Future developers can use the code on GitLab to enhance the push notification capabilities of the app before releasing version 2. The code I have used Pusher Beams to send push notifications to all iOS devices that granted push notification permission. It sends a notification consisting of a title and a message. Clicking the notification simply opens the Mustang News app. Version 2 of the app could let students sign up for push notification only when articles of a certain section are posted. This can be done using Pusher Beams. Another improvement on push notifications would be to create an option for students to receive notifications of urgent information/breaking news. The staff at Mustang News could send out notifications on demand to people who are interested in staying up to date.

5.2 Adding Images to the Article Content

Sections 4.2 and 4.3 highlight one of the major problems I had with development of the application. The content of the app was difficult to format, and the final design choice resulted in a limiting design since writers must edit any stories that contain any media in the content portion. Future versions of the app should redesign article view so that the writers have more freedom to include images in the content, not just the featured image. Several news apps like The Washington Post or the New York Times are good models to look at for reference. [4]

5.3 Accessibility

Next versions of the app should focus on accessibility. Users should be able to increase or decrease the font size to make reading easier. Most apps that require a large amount of reading have settings that make the screen easier on the eyes. Future versions of the app could add functionality to dim the screen or change it to a “dark theme” that would have light text on a dark background.

5.4 Continue Work on Augmented Reality (AR)

There are several mainstream news outlets that are experimenting with AR in journalism. A good goal for the Cal Poly Mustang News app would be to use AR to integrate the app with the campus newspaper. This would be done by having an AR feature on the app that would let users point the camera to an image printed on the newspaper and trigger an AR effect. There are other methods AR can be included in journalism, this is just one method that would support the newspaper. There are code samples of some of the AR libraries I researched in the GitLab repo.
6  Reflections

Most of the first quarter was spent refactoring the code from using collection views to using the page controller and perfecting the user interface. The second quarter was spent fixing caching problems, adding push notification feature, and researching AR libraries. After a full year of working on the app here are a few things I would do differently.

6.1  React Native / Android app

I originally started developing this project as a native iOS app because my only exposure to app development was from Cal Poly’s Mobile App Development course. I have a Mac so development was very easy the first quarter of senior project since all testing could be done using the Simulator. The second quarter testing push notification and AR libraries was more difficult since it could no longer be done on just the simulator. I had to schedule meetings with friends to lend me an iPhone to do some of my testing.

So, now that I understand some fundamentals of app development I would have chosen to do this project exploring React Native or Android Studio. This would have made development and testing a lot easier.

6.2  Work on the Proposed Project Goals

On my senior project proposal, I listed a set of goals for the second quarter that included some of the functionality mentioned in Section 5: Future Work. In the beginning of the second quarter I chose to change my plans and focus this quarter on exploring AR libraries and improving performance. In hindsight, I would choose to work on the original set of goals, like implementing the accessibility settings and designing a way to dynamically add media to the body of an article.

Most of that functionality would have been testable on the Simulator which would make development easier. Also, throughout this senior project experience I realized I really enjoy working on the interface design and visible functionality so those features would have been more enjoyable.
7 Conclusion

Overall, I had a positive experience with my senior project. The app has the functionality that Mustang News needed, and it was designed to meet the constraints set up by the client requirements. I enjoyed collaborating with students who were from outside my major and feel like it enhanced my learning experience. This was my first app and I am excited to see it in the App Store. This project also helped me realize that I thoroughly enjoy UI/UX design and I hope to continue some version of mobile app development in the future.
References


