

# Team No. 3

Team Members: Jan Polzer, Hamza Hameed, Ron Huff, Ethan Malin, Alejandro Melgar.  
Project Name: Today  
Project Synopsis: A mobile application for the University Daily Kansan

## ***Project Synopsis***

We will create a mobile application for the University Daily Kansan to provide news stories, videos, images, and other media produced, to the users, in a user friendly format.

## ***Project Description***

The project was undertaken at the request of the UDK marketing director.

The application should draw more people to their website and the newspaper, as well as to become another media for people who prefer mobile application over the others. Creating a mobile application for the UDK gives us the opportunity to make it easier for students to stay up to date with what is currently happening on campus.

The app could also potentially house data produced by KUJH, the broadcast journalism branch of the journalism department. This may also come with the ability to livestream broadcasts produced by the KUJH news room.

A finished project will consist of a functioning app which will be available to anyone with access to a mobile device running an Android or iOS operating system. This mobile application will give the UDK the ability to distribute their articles to a larger audience.

## ***Project Milestones***

### *First Semester*

- Create Structure of App
  - Estimated completion date :10/5/2018
- Use diagrams created
  - Estimated completion date :10/12/2018
- Create Alpha build
  - Estimated completion date : 11/2/2018
- Create Build Tests
  - Estimated completion date : 11/9/2018
- Define proper outline for project in the following semester
  - Estimated completion date : 11/30/2018

### *Second Semester*

- Fix issues found in Alpha testing
  - Estimated completion date : 2/1/2019

- Create Beta build
  - Estimated completion date : 2/20/2019
- Polish Final Build
  - Estimated completion date : 4/26/2019

## ***Project Budget***

### *Estimated cost*

\$124.00 + \$99/year to host the app on both app stores. This cost will be the responsibility of the UDK.

### *Hardware, software, and/or computing resources*

N/A

### *Vendor*

The app will be hosted on the Google Play store and the Apple App store

### *Special training*

React Native

Server side programming with BLOX CMS

### *When they will be required*

React Native - 10/7/18

BLOX CMS - 10/8/18

## ***Work Plan***

Ron Huff	backend
Alejandro Melgar	frontend
Hamza Hameed	frontend
Ethan Malin	backend
Jan Polzer	backend

## Gantt Chart

	Start Date	End Date	Timeline	Status
<b>UDK Mobile App</b>	9-15-2018	5-10-2019		
First meeting with UDK	9-15-2018	9-21-2018		Completed ▾
Get familiar with backend and software	9-22-2018	10-26-2018		In progress ▾
Define structure of functioning App	9-29-2018	10-5-2018		Completed ▾
Use diagrams created	10-6-2018	10-12-2018		In progress ▾
Initial design completed	10-13-2018	10-26-2018		Planned ▾
Review with UDK	10-20-2018	10-26-2018		Planned ▾
Create alpha build	10-27-2018	11-9-2018		Planned ▾
Create build tests	11-3-2018	11-16-2018		Planned ▾
Work out bugs	11-10-2018	11-23-2018		Planned ▾
Define outline for project for next semester	11-17-2018	12-6-2018		Planned ▾
Winter break	12-7-2018	1-21-2019		Planned ▾
Refine additional features	1-22-2019	2-1-2019		Planned ▾
Start beta build	1-22-2019	2-6-2019		Planned ▾
Add commenting	2-2-2019	2-22-2019		Planned ▾
Add advertising	2-16-2019	3-8-2019		Planned ▾
Add KUJH live streaming	3-2-2019	3-22-2019		Planned ▾
Review with UDK	3-18-2019	3-22-2019		Planned ▾
Add last features	3-18-2019	4-26-2019		Planned ▾
Polish final build and release to download	4-20-2019	5-3-2019		Planned ▾

## Preliminary Project Design

### Backend

The UDK publishes any content they produce to a *Content Management System* called BLOX CMS, which is owned by TownNews.com. As far as we can tell right now, there is no way to directly access the data programmatically once the UDK has published it. Whether via SQL query over the network, or some similar data request. The only usable interface is the RSS feed generation capability.

An RSS feed is a request-response interaction in which a client requests data using a URL such as `https://yourdomain.com/?q=<parameters>`. The server then responds with XML formatted content. It is returned pre-formatted because RSS feeds are traditionally embedded on websites directly so a browser would then handle presenting the data. However, we are not a browser, and our content will not be presented using XML, but instead in JSX, which is the markdown language used by React Native.

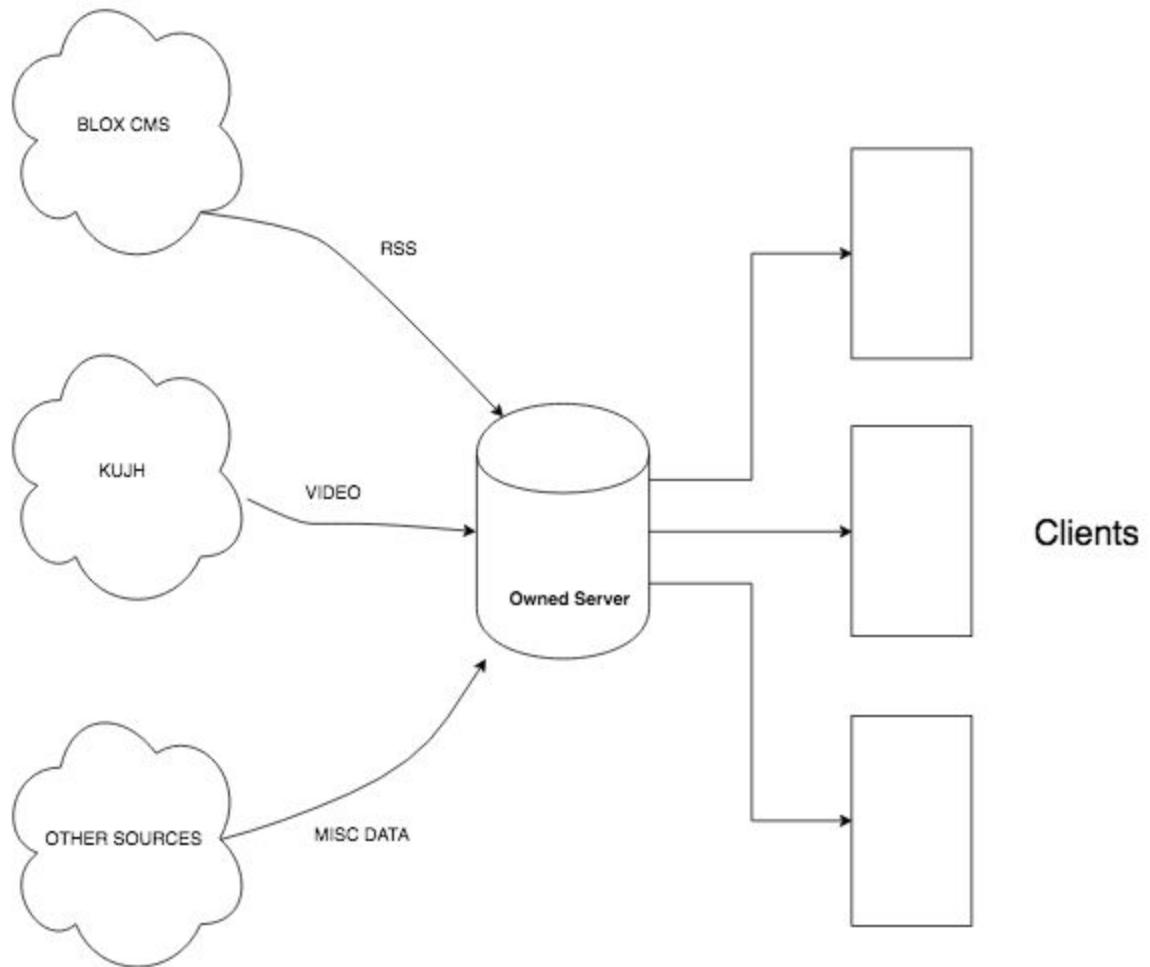
What we will have to do is have a server of our own that keeps a database of content produced by the UDK. Ideally, the UDK would upload the raw data to our server directly to be easily formatted for use by our clients. However, this would require the journalists to upload content twice for every story they produced, and would make it difficult to make changes to content after it has been published. To avoid this, we will have to get the data from BLOX CMS, and it seems as if we will be getting it in XML format. If we can use a library to parse the

returned document into usable data, we will consider it. But it seems likely that this is a job we will have to do ourselves.

You can see this is not an ideal situation for a few reasons. If BLOX CMS changes one thing about how they format their content generated for RSS requests, our server could fail to correctly store the data, resulting in undefined behavior. If the UDK changes content it published in the past, our server will have to correctly update that content as well. We are also storing every story the UDK produces twice, it seems as if there should be some better way to solve this problem, given that the data we need is already obviously housed by the BLOX CMS system, we just cannot seem to get it, with TownNews.com being a company with company-like interests.

At any rate, our server parses and stores the content in a usable manner, and sends it to clients (phones) whenever it gets a request. Some thought will have to be put in as to how to validate the requests, and how to protect our server from malice.

## Backend Design



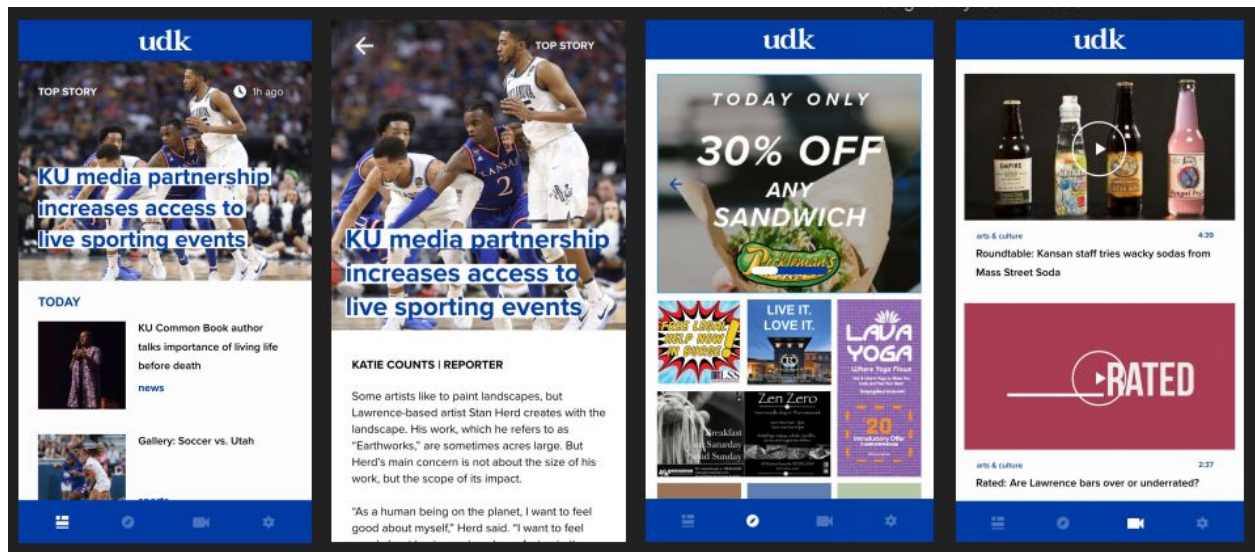
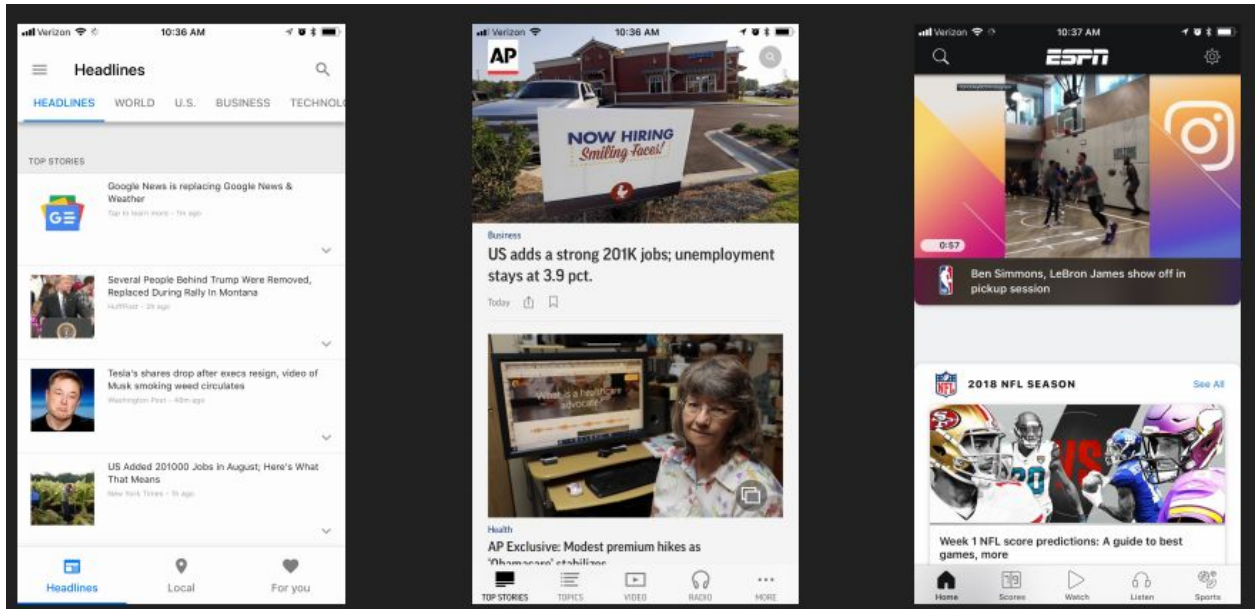
## Frontend

The app will be written using a javascript framework *React Native*, which translates javascript and JSX (a markdown language) into native code for each of the two platforms of interest for us. This avoids attempting to learn how to develop for two separate mobile platforms, and implementing the same design twice.

The plan for the UI is to make it look similar to other news apps of this nature. This means that the most important stories will show up at the top (as headlines) and the user scrolls down to see more stories. Clicking on a headline will allow the user to read the rest of the story. Additional buttons (back button, search button, bottom navigation bar) will be available depending on the exact functionality that is decided on for the app. The UDK already gave us a template for what they want the UI to look like, but they have allowed us generous leeway as to how to design it so that we aren't burdened in any way. We plan on making it look slick and

modern like most successful apps do and making heavy use of straight lines and boxes so that it looks clean and is easily readable. Since the app is an extension of the UDK website, we will be utilizing some of the design choices that they already have such as font choice or the colors used. If React Native works out like it should, both the Android and iOS should look exactly the same, with only minor cosmetic differences at best.

(Below: UDK Designs)



### Design constraints

The main constraint we have is our platform. We are developing a mobile app for Android and iOS, so we are already forced to have native codebases for both of those platforms. This leads us to our framework of choice, React Native. With React Native, we can

write one codebase, in javascript, and translate it to native code for either of the two platforms. This cuts our work in half.

This choice further constrains us by roping us into a javascript framework. As mentioned above we will be using React Native, which means we are constrained by what libraries, components, etc, we can use, and we are forced to use Javascript for most of the front end development.

The University Daily Kansan uses BLOX CMS to upload Articles to their website, the Kansan.com. As a result they would like to be able to use BLOX CMS to upload articles onto a mobile device. Another constraint lies within this as we do not have much experience with mobile development and using BLOX CMS and the use of RSS Feeds.

We are also constrained by time. Having to develop a functioning mobile app by May severely limits the scope of a project we will actually be able to finish. We must design an application which can be completed in this timeframe.

## ***Ethical and Intellectual Property Issues***

### **Security**

It is our responsibility to make sure all content hosted on the app is controlled entirely by the UDK. No outside content will be allowed in, and no secure content (passwords, internal communication, etc) is allowed out. Hopefully we will not be interacting much with sensitive information, so having solid security should be less difficult.

### **Content**

All content related ethical issues will have to be handled by the UDK at the time they occur. We assume no responsibility for the media posted by the UDK or the comments posted by readers of the newspaper.

### **Privacy**

We will not 'monitor' the user, or otherwise access any information that we do not have a clear, essential need for.

### **IP**

This topic has yet to be clearly resolved. Currently it seems that the UDK will possess all intellectual property for this project however the possibility for developer team accreditation, ad revenue sharing or some other form of remuneration is still a topic for discussion.

