
**University of Arkansas – CSCE Department**
**Capstone II – Final Report – Spring 2020**

# Campus Events

**William Bissett, Colby Crowne, Keaton Dalquist,**
**Manuel Delarosa, Alexander Frey, Kassell Harris**

## Abstract

One of the most important aspects of the college experience is the social gatherings of students. This is often done through clubs, guest speakers after class, career fairs, celebrations, and university sponsored events. Since there are so many gatherings that students can elect to attend, promoted and hosted by a large variety of groups, students can often feel overwhelmed. Inboxes are flooded with emails promoting important university sponsored events, clubs pass out flyers and send messages via group messaging applications, and professors and friends spread event details by word of mouth. Sometimes events are barely promoted, leading to a small turnout. With many different events happening daily on campus, students are bound to forget about events they may be interested in attending, or miss out on exciting events that weren't promoted as heavily as others. This is where Campus Events comes into play.

Campus Events aims to solve the problems mentioned above by streamlining all the information about events being hosted on campus in one place. Students are able to elect to automatically receive notifications from their favorite departments or Registered Student Organizations without having to dig through old emails or be handed a flyer while walking to class. With all the event information consolidated into an app, all the necessary information the student needs to attend will be present. Clear location and starting times, event details, and itineraries can be accessed quickly by the student. Students who have a free afternoon or evening can quickly search to see if any events are being held that they may wish to attend. Event planners and clubs will also have a one stop location to advertise their events.

## 1.0 Problem

The problem is the unorganized nature of campus events and activities. Students often get many different emails informing them of guest speakers, companies visiting campus, and club events that crowd their inbox. This results in many students missing or possibly not even opening the messages. When students don’t look at these emails, don't pay attention to flyers around campus, or are just overwhelmed by the number of events happening, it's important to try consolidating the information into one location for students to quickly access.

Having this application would increase student awareness and turn out to the many events on campus. A second perk is the integration of event information with something most students have, a smartphone. The ability to sync events compiled by the app to the user's calendar allows for students to receive notifications and reminders to help students remember events they wish to attend. With calendar sync, they can plan around classes or other commitments. This would result in a more engaged student body, leading more companies and guest speakers to be interested in hosting events at the University of Arkansas, knowing the school has large turnouts.

## 2.0 Objective

The objective is to create an application that consolidates all campus events into one central location. Users will be able to create new events and display them for all users. This master list of events can also be filtered by the user to only show events they may be interested in, such as clubs or departments they are a part of. The app will provide push notifications to the user to remind them of events that are happening. User's will also have the option to sync the events to their Google Calendar.

## 3.0 Background

### 3.1 Key Concepts

For this application a few key concepts are needed. One concept is a dynamic user login database. This allows for students or groups to have unique accounts for the application using an email and password. Each user having a unique account allows for repeated use of the app as long as the database stands, in addition to connecting specific data to the user's account.

Another key concept is syncing the event with the user's Google Calendar by using the Google Calendar API. Google allows applications to use this feature, and this can be implemented by pressing a sync button on the client side of the application. This needs to be included in the design process, leading to the event appearing on the user's Google Calendar. A similar feature could be created to export calendar information to the user's calendar of choice.

### 3.2 Related Work

The most similar and accomplished application within our project area would be a calendar app like Google Calendar or Microsoft Outlook. Both applications handle a much broader range of issues as they are designed to organize a person's entire day-to-day schedule for every facet of one’s life. The goal of this application is much more specific. The application targets on-campus events. This allows a student to manage this part of their life. From this

micro-perspective, there are some shortcomings when compared to big applications like Google Calendar or Microsoft Outlook, and this application looks to improve upon these examples.

One common drawback of a calendar application such as Google Calendar is that its design inhibits efficiency [1]. Google Calendar can often become too cluttered with multiple calendars and general information being displayed all at once. For some situations, this is understandable and beneficial in terms of managing and displaying specific information clearly. This application will make sure a student's events are displayed with no clutter at all, ensuring the user can check their events and its details quickly and efficiently.

A second negative some scheduling apps have is the need to search through the entire calendar to see upcoming events. This application will improve upon this by providing an "upcoming" page that shows all events in the near future.

# 4.0 Design

## 4.1 Requirements and/or Use Cases and/or Design Goals

Requirements

1. SQL database
    a. Login table to store user login information
        i. Username, password, date created, type (student, faculty, employer, event organizer)
    b. User information table to hold additional information on user
        i. Link to login table, store user's UARK email, etc
    c. Events table to store information around events
        i. Store type of event (company, guest speaker, club meetings, other)
2. Easy to navigate user interface
    a. A simple and clean interface for maximum efficiency
        i. User logs into home page displaying upcoming events
            1. If no upcoming events are followed by user, display the events closest to current date
    b. Several tabs: home, upcoming, search, create
        i. Home page displays user's followed events
        ii. Upcoming page displays events near current date
        iii. Search page allows user to search for a certain event, event organizer, or guest speaker by name, date, or type
        iv. Create page allows the user to create an event

3. Integration and sync with Google Calendar

    a. When a user follows/accepts an event, the user has the option to link their Google Calendar for the event to be automatically added.

        i. The event is added at the exact time and date to the user's calendar, with specified reminders so the user does not forget the event

4. Ability for users to add events

    a. Club owners, companies, and other organizers have the ability to create their own event within the application

        i. On the create page, if the user is certain type, they can manually create upcoming events to be included in the application

        ii. The user inputs the time, date, location, description, title, etc for the event and notifies moderators/admins to check on listing

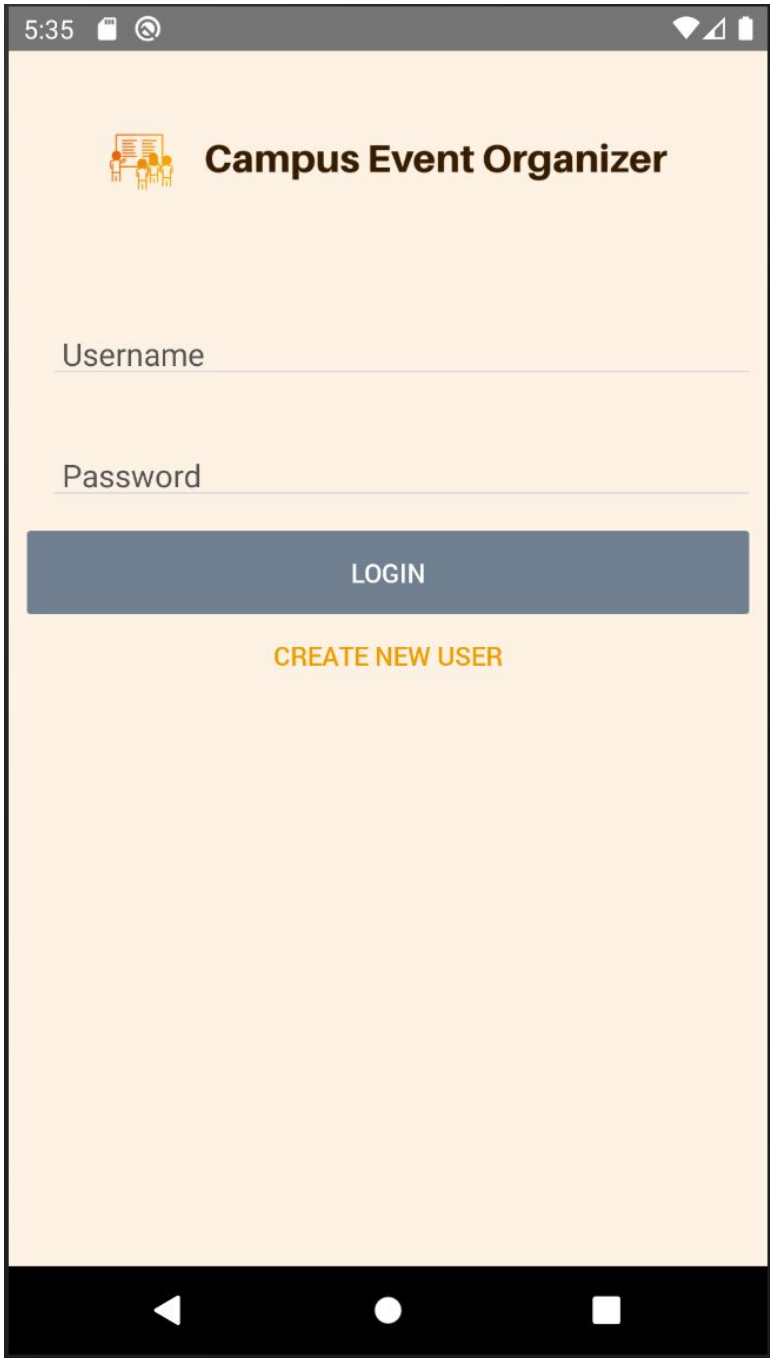
5. Notifications

    a. Ability for event organizers to send out notification to a group of people about the event

    b. This could be done using Firebase notifications

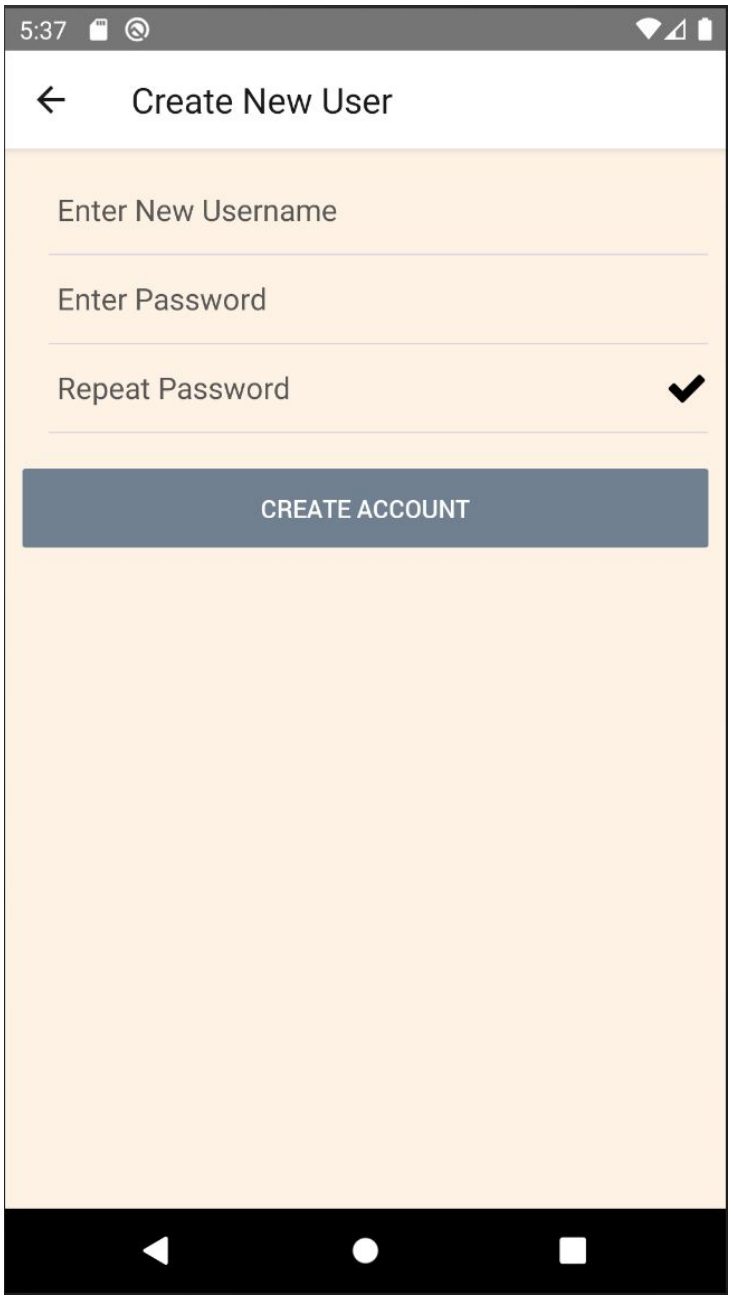
### 4.2 [High Level / Detailed] Architecture

Login Screen:

- 

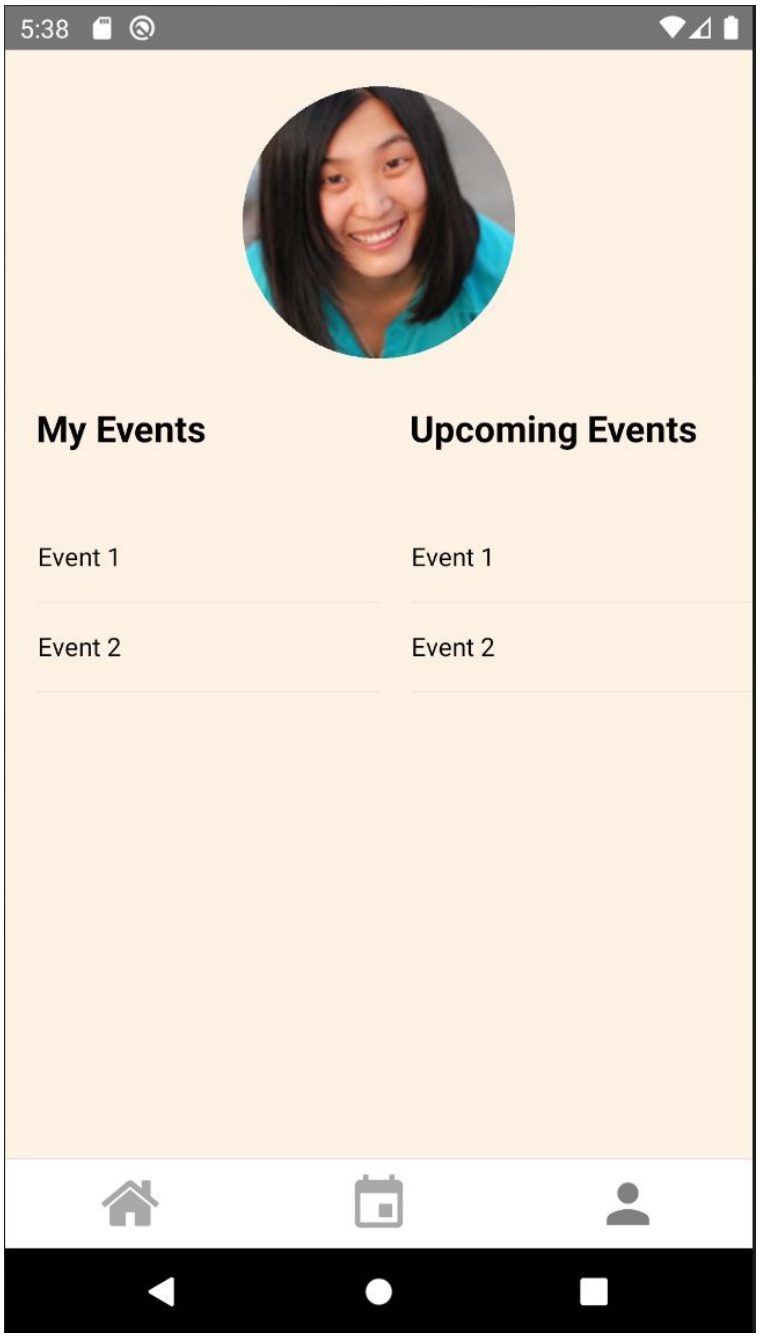
- Login screen for the application. This page takes the user's username and password so the user can successfully login. A “Create New User” button is included for first time users to create a new user in the system.

Create New User Screen:

- 

- Create New User screen for the application. This page requires the user to fill out a simple form to create an account within the application. The user is required to enter a new username and a new password, and to confirm their password selection. Password authentication is also included. It checks if the password and password confirmation are identical, and displays a check mark if the passwords are found to match.

Profile Screen:

- 

- Profile Screen for the application. This page of the application displays the user's unique profile information. This page shows a profile picture, events the user has created, and upcoming events the user is subscribed to. There is also a navigation bar at the bottom of the screen. From left to right is Home, Upcoming, and Profile, and they navigate the user to different views within the app.

### 4.3 Risks

| Risk | Risk Reduction |
|---|---|
| Multiple users create the same event, leading to conflicts in the database | Only allow one event creator per event. |
| Users could overload the database. Resulting in the application backend crashing. | Need to make the database able to handle many users so the application does not crash |
| Risk of privacy; users could hack into the database and get usernames and passwords | Need to hash user passwords and encrypt the database so attackers can not read the information. |

### 4.4 Tasks –1. Decide on technologies to be used, and overall architecture of the project

2. Design of backend:
    - Create Database:
        - Develop schemas for all functionality
            - Firebase
            - Event creation
            - Event followers/group members
            - Relevant user information
    - Create Service to make calls to Google Calendar API
3. Design of Front end to connect to database:
    - Develop authentication portion of the app first
        - Login screen, with password authentication
    - Develop main app stack
        - Home page
            - Settings will be available through this page
        - Upcoming events page
        - Profile page
        - Create event page
4. Write unit tests and documentation

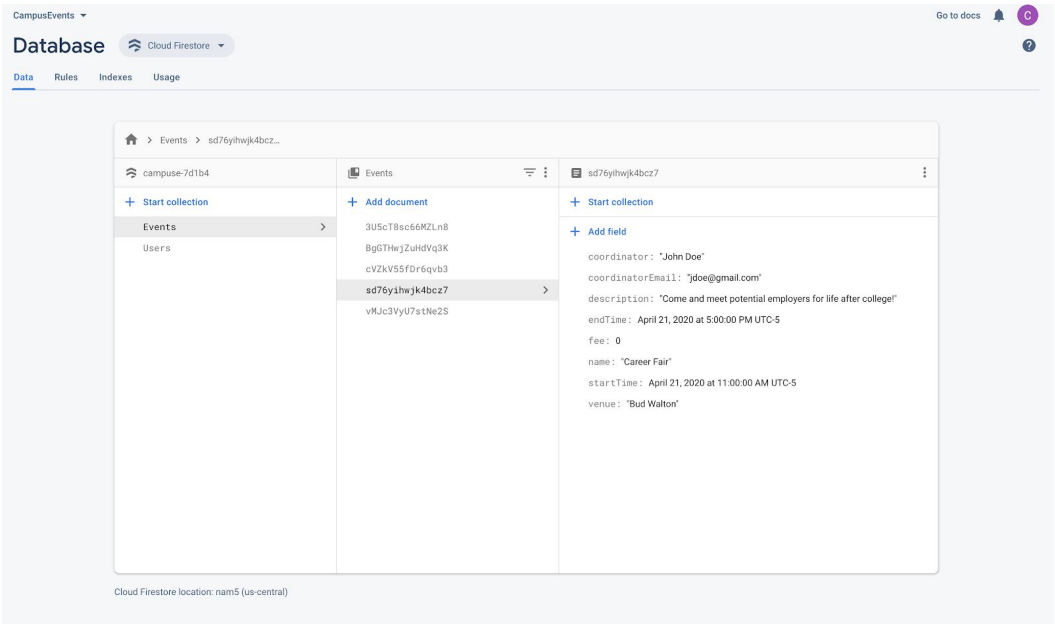
### 4.5 Schedule –

| Tasks | Dates |
|---|---|
| 1. Preliminary work<br>1. Set up database accounts<br>2. Set up API Keys<br>3. Set up dev environments | 1/13-1/17 |
| 2. Design database schema<br>1. Implement database tables<br>2. Start designing service to connect database to calendar | 1/20-1/31 |
| 3. Start implementing front end<br>1. Design UI/UX for app<br>2. Start designing functionality for front end | 2/3-2/14 |
| 4. Connect front end to database<br>1. Connect user authentication to DB<br>2. Connect event creation to DB | 2/17-2/28 |
| 5. Implement calendar functions<br>1. Use API keys to connect DB to user calendar<br>2. Implement notification features | 3/2-3/13 |
| 6. Finish front end<br>1. Complete UI/UX<br>2. Finalize app functionality and start debugging | 3/16-4/3 |
| 7. Documentation<br>1. Start report<br>2. Document any remaining code | 4/6-4/17 |

| | |
|---|---|
| 3. Finish debugging and prepare final product<br><br>8. Complete any loose ends<br><br>9. Update website, finish report, and film presentation video | 4/20-4/30 |

### 4.6 Deliverables –

- Documentation: All relevant reports, documents, and .pdf's will be included in website
- Database scheme and initial data: Firebase will be used for usernames and passwords




- Application code: React Native code is hosted on Colby's Github account.
- Final Report

## 5.0 Key Personnel

**William Bissett IV** - Bissett is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed all the relevant courses. He interned at Dialexa as a Software Engineering Intern and a User Interface developer at Systemware. He will be responsible for the user interface and experience through the whole application.

**Colby Crowne** - Crowne is a senior Computer science major at the University of Arkansas. He has completed all pre-requisites for Capstone. He is currently an intern at JB Hunt. He will be responsible for architecting the database as well as navigation through the app.

**Keaton Dalquist** - Dalquist is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed all the relevant courses. He interns as a Software Engineering Intern at Onestone Ecommerce. He will be responsible for the user interface and experience through the whole application. He will also work in calendar integration and anywhere else needed.

**Manuel Delarosa** - Delarosa is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has currently an internship offer with J.B. Hunt for the upcoming summer . He will be responsible for the client side of the program and whatever the team needs him to complete.

**Alex Frey** - Frey is a senior Computer Science major at the University of Arkansas. He has completed all relevant courses in order to accomplish the project from beginning to end. He has interned with the Data Analytics team at Andrews Distributing in Dallas, Texas. Alex will mainly be working on back-end development and helping the team anywhere else needed.

**Kassell Harris** - Harris is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas with a minor in Mathematics. During his time at the University of Arkansas he has taken coursework which includes Mobile Programming, Databases, Software Engineering, and Programming Paradigms. He currently interns as an applications developer at J. B. Hunt Transport. Harris will be responsible for the development of the functionality of the front end and its relation to the user experience.

## 6.0 Facilities and Equipment

Laptops and computers will be needed for software development. Additionally, meeting locations will need to be discussed so that work can be accomplished in a proper and efficient environment.

## 7.0 References

[1] GetApp.com Google Calendar Reviews,
https://www.getapp.com/collaboration-software/a/google-calendar/reviews/

[2] Wordpress Web page,
https://wordpressua.uark.edu/capstone/create-a-page/team-5-bulletin-board-application-campus-events/

[3] Github (Hosted on Colby's account),

https://github.com/ColbsSTR/CapstoneProject