



# Campus Events



# Objective

- Create an application that consolidates all campus events into one central application
- Create clean and organized UI and UX designs
- Allow users to sync their selected events to their Google Calendar, or export to their calendar of choice.

# Problem

- The main problem we are looking to solve is the unorganized nature of campus events and activities.
- As students, we get a lot of different emails talking about guest speakers, companies visiting campus, and club events that crowd our inbox and makes many of us not even open the messages.
- Because a lot of students don't look at these emails, don't pay attention to posters around campus, or are just get overwhelmed by the amount of stuff going on, we thought it would be important to consolidate all this information into one central application.

# Related Work

It's important to discuss similar products and their drawbacks to see where we could make improvements and provide an innovative solution.

- **Google Calendar**

- Too much clutter with multiple facets of someone's life being displayed from a very macro-perspective.
- Our Solution: Display campus events only and in a clean and efficient manner so as to quickly relay information to the user.

- **EventZilla**

- No easy and fast way to see events in the near future.
- Our Solution: "Upcoming Events" tab will display events within the week for user convenience

# Requirements

## 1. **SQL Database**

- login information, Event information

## 2. **Easy to navigate UI**

- Several tabs, Home, upcoming, search, create

## 3. **Integration and sync with Google Calendar**

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

## 4. **Ability for users to add events**

- Club owners, companies, etc.

## 5. **Notifications**

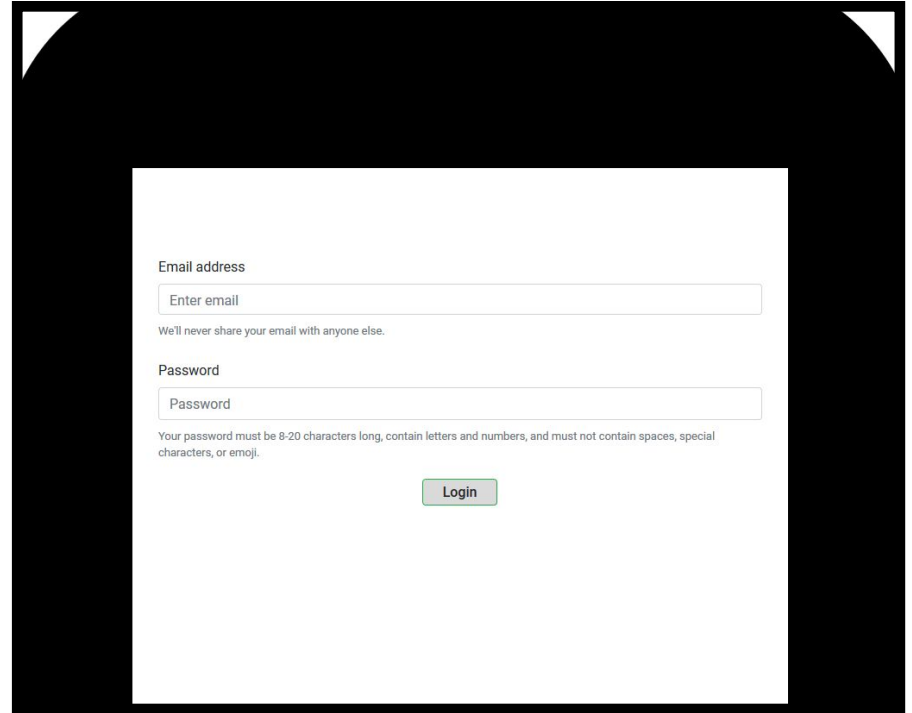
- For event organizers to send out messages



# Architecture

## 1. Login Page

- a. Username/email textbox
- b. Password textbox
- c. Login button



The image shows a login page architecture within a black frame. It features two text input fields and a button. The first field is labeled "Email address" and contains the placeholder text "Enter email". Below it is a small line of text: "We'll never share your email with anyone else." The second field is labeled "Password" and contains the placeholder text "Password". Below it is a line of text: "Your password must be 8-20 characters long, contain letters and numbers, and must not contain spaces, special characters, or emoji." At the bottom center is a button labeled "Login".

Email address

We'll never share your email with anyone else.

Password

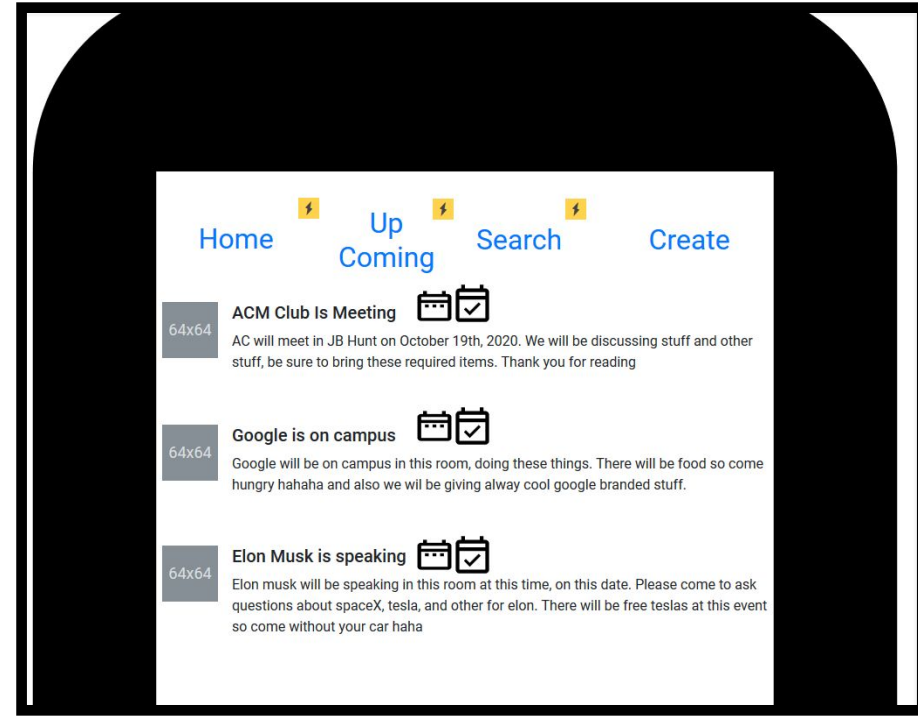
Your password must be 8-20 characters long, contain letters and numbers, and must not contain spaces, special characters, or emoji.

Login

# Architecture

## 1. Main Page

- a. Home tab
- b. Upcoming/events tab
- c. Search tab
- d. Create tab

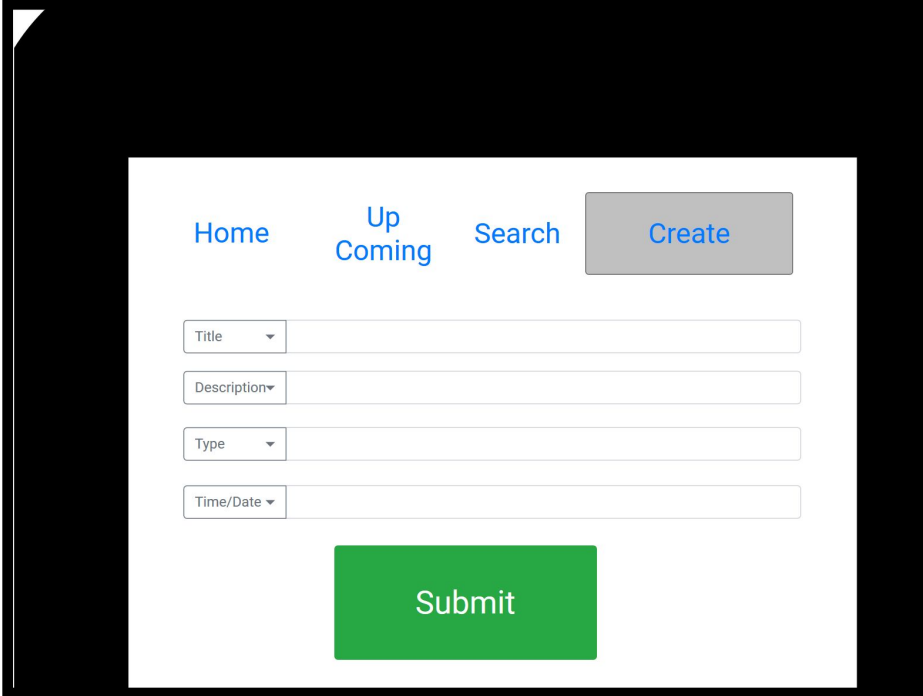




# Architecture

## 1. Create Page

- a. Title Textbox
- b. Description Textbox
- c. Type Textbox (event type)
- d. Time/Date Textbox



The screenshot shows a web interface for creating a page. At the top, there are navigation links: 'Home', 'Up Coming', 'Search', and a grey 'Create' button. Below these are four input fields, each with a dropdown menu on the left: 'Title', 'Description', 'Type', and 'Time/Date'. At the bottom center is a large green 'Submit' button.

# Design

- Simple design
- Textboxes and buttons for user input
- Top bar navigation
- Home Tab
  - Personal events, followed events, recommended events
- Events Tab
  - All upcoming events
- Search Tab
  - Allows user to search for an event
- Create
  - Allows user to create an event, invite other users

# Risks

- Users could overload the database
  - Try to use a scalable model, flexible design for the database.
- Privacy
  - Need to hash our passwords and encrypt any other sensitive information in the database.

# 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
      - 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 two factor authentication
  - Develop main app stack
    - Home page
      - Settings will be available through this page
    - Upcoming events page
    - Search page
    - Create event page
4. Write unit tests

# Schedule

<b>Date</b>	1/13-1/17	1/20-1/31	2/3-2/14	2/17-2/28	3/2-3/13	3/16-4/3	4/6-4/17	4/20-5/1
<b>Task</b>	<p><b>Preliminary work</b></p> <ol style="list-style-type: none"> <li>1. Set up database accounts</li> <li>2. Set up API Keys</li> <li>3. Set up development environment</li> </ol>	<p><b>Design database Schema</b></p> <ol style="list-style-type: none"> <li>1. Implement database tables</li> <li>2. Start designing service to connect database to calendar</li> </ol>	<p><b>Start Implementing Front End</b></p> <ol style="list-style-type: none"> <li>1. Design UI/UX for app</li> <li>2. Start designing front end functionality</li> </ol>	<p><b>Connect Front End to Database</b></p> <ol style="list-style-type: none"> <li>1. Connect user authentication to DB</li> <li>2. Connect event creation to DB</li> </ol>	<p><b>Implement Calendar Function</b></p> <ol style="list-style-type: none"> <li>1. Use API keys to connect DB to user calendar</li> <li>2. Implement notification features</li> </ol>	<p><b>Finish Front End</b></p> <ol style="list-style-type: none"> <li>1. Complete UI/UX</li> <li>2. Finalize app functionality</li> <li>3. Start product testing and debugging</li> </ol>	<p><b>Documentation</b></p> <ol style="list-style-type: none"> <li>1. Start report</li> <li>2. Document any remaining code</li> <li>3. Complete debugging and prepare final product</li> </ol>	<p><b>Complete Any Loose Ends</b></p>

# Team Members

- William Bissett IV
- Keaton Dalquist
- Kassell Harris
- Manuel Delarosa
- Colby Crowne
- Alex Frey