

San Francisco State University School Project

Internet App Dev CSC667/867 Spring 2019

## **EventUp**

### Final Documentation

Team 4

**Cory Lewis**(Team Lead, [clewis9@mail.sfsu.edu](mailto:clewis9@mail.sfsu.edu))

Mitul (Front End/GitHub Master)

Chintan Sanjay Puri (Front End)

Alex Wolski (Front End/Back End)

Vincent Santos(Front End/Back End)

05/18/19

<https://github.com/csc667-02-sp19/csc667-sp19-Team04>

## 1. Project Introduction

EventUp is a social app for iOS designed to bring SFSU students together. Users can create, browse, and RSVP for events on campus. The app has many convenient features to help students find, remember, and get to different events.

There is a category filter to help students find the kinds of events they are looking for. There is a button that adds the event to your calendar. And apple maps are integrated into the app, so users can get directions to where the event is taking place. Additionally, users can leave comments on an event in case they need to communicate with the host of the event.

The inspiration for this app came from the pre-existing SFSU app. We found that the SFSU app was very limited and only showed campus news. Additionally, the UI for the SFSU app was confusing and not friendly for students. Our goal was to expand on the idea of hosting events on campus by allowing students to post and connect with each other via RSVP and commenting.

## 2. Software Stack Used

- a. Server: Ubuntu Server 18.04.2 on AWS
- b. Web Server & Version: Node.js (v10.15.1 LTS)
- c. Database Server Used: MySQL (v8.0.13)
- d. Server-side language: JavaScript
- e. Frameworks Used:
  - i. React Native by Facebook
  - ii. Express.js
- f. Additional Libraries on Front-end side:
  - 1) UI/UX
    - a) React-Native-Elements
    - b) React-Native-Paper
    - c) React-Native-Material-Dropdown
  - 2) Image Selector
    - a) React-Native-Image-Picker
  - 3) Navigation/App Flow
    - a) React-Navigation
  - 4) Map Integration for Direction
    - a) React-Native-Maps

3. Tools used for communication
  - a. Google Calendar
    - i. Used to keep track of deadlines and submissions
  - b. Discord
    - i. Used for team collaboration
    - ii. Functions Used: webhooks for github pings for everyone to be notified of project progress.
    - iii. Google Calendar linked to discord for deadline updates
    - iv. Trello tasks linked to Discord so members would get notified when tasks are created in trello
    - v. Team Lead notified members of meetings and any urgent messages in global ``announcements`` chat.
    - vi. Server and Database channel used to keep reference of any important commands or fixes used throughout the course of the project

4. Tools used for task management
  - a. Trello
    - i. Used to assign tasks to team members and describe what needs to be done and how to do it
    - ii. TODO/Doing columns were created for Back-End and Front-End so each side of the project knows what they need to do and if this task is urgent or not
  - b. Used Throughout Development

## 5. Build Instructions for Application

### a. Packages Needed:

#### i. Front End Packages:

aws-sdk: 2.442.0

axios: 0.18.0

date-fns": "^1.30.1

expo: 32.0.0

json-server: 0.14.2

moment: 2.24.0

React:16.5.0

React-native-elements:1.1.0

react-native-fs: 2.13.3

React-native-image-picker:0.28.1

react-native-maps: 0.24.2

React-native-material-dropdown:0.11.1

react-native-modal-datetime-picker: 7.4.0

React-native-open-maps:0.3.3

react-native-paper: 2.15.2

react-navigation: 3.5.1

#### ii. Back End Packages:

bcrypt: 3.0.4

body-parser: 1.18.3

express: 4.16.4

file-system: 2.2.2

jsonwebtoken: 8.5.1

mongoose: 5.4.19

morgan: 1.9.1

multer: 1.4.1

mysql: 2.16.0

mysql2: 1.6.5

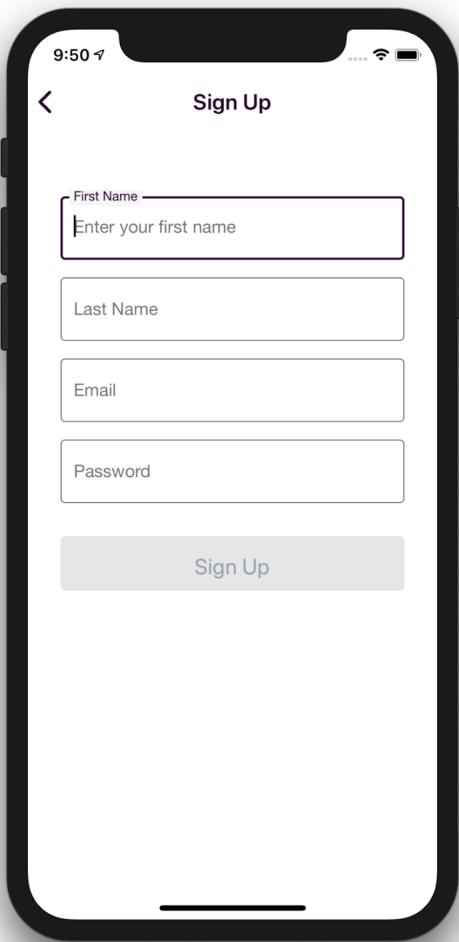
- b. To run your application

<https://github.com/csc667-02-sp19/csc667-sp19-Team04#getting-started>

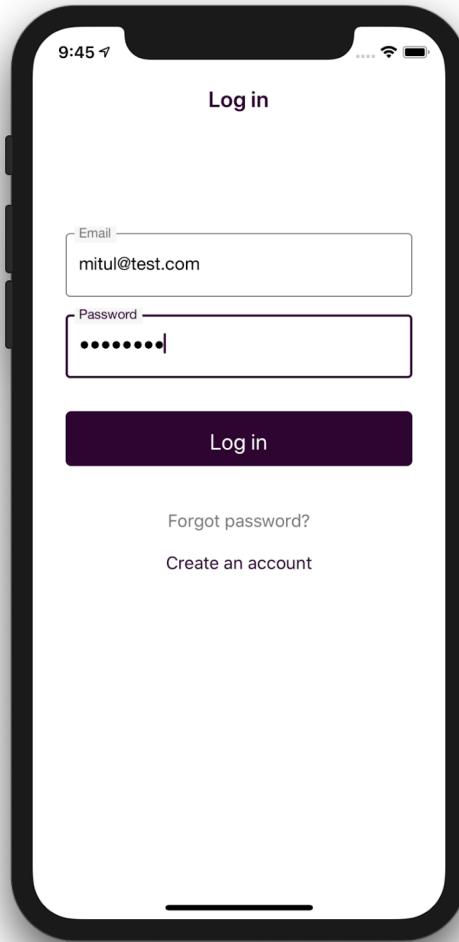
- c. Any other information needed to run your application

<https://github.com/csc667-02-sp19/csc667-sp19-Team04/blob/master/README.md>

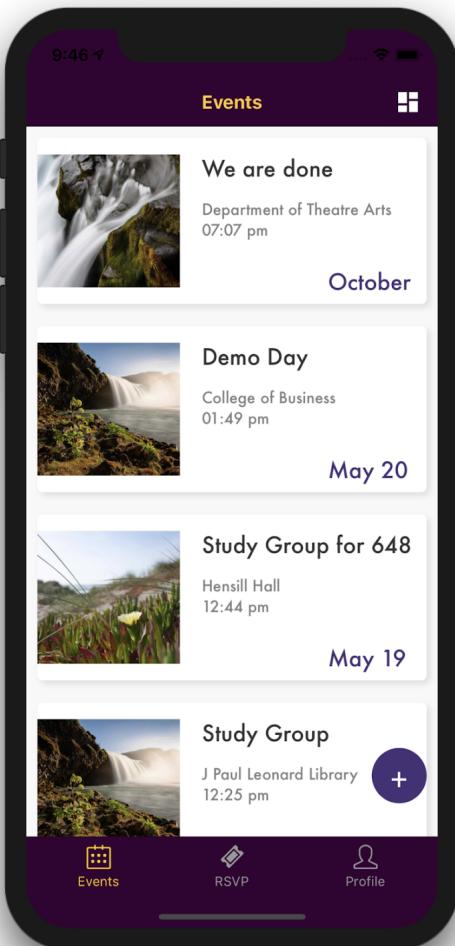
- d. Pictures of each page in your application



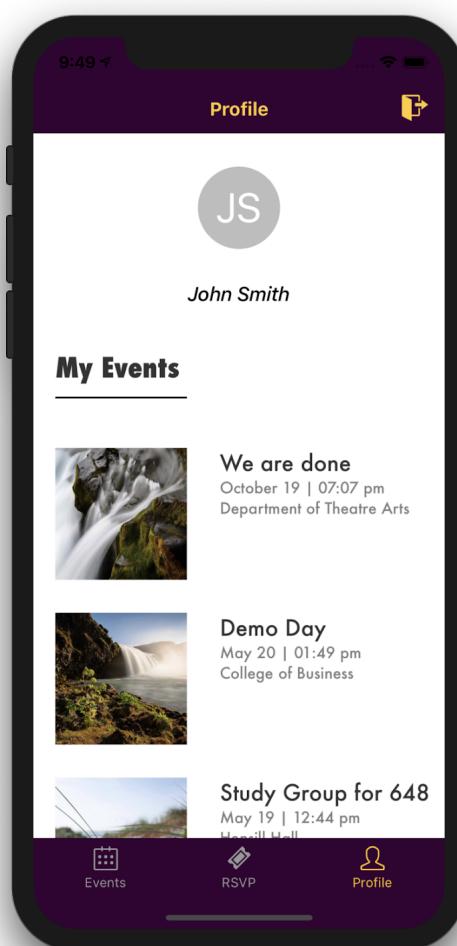
iPhone X - 12.1



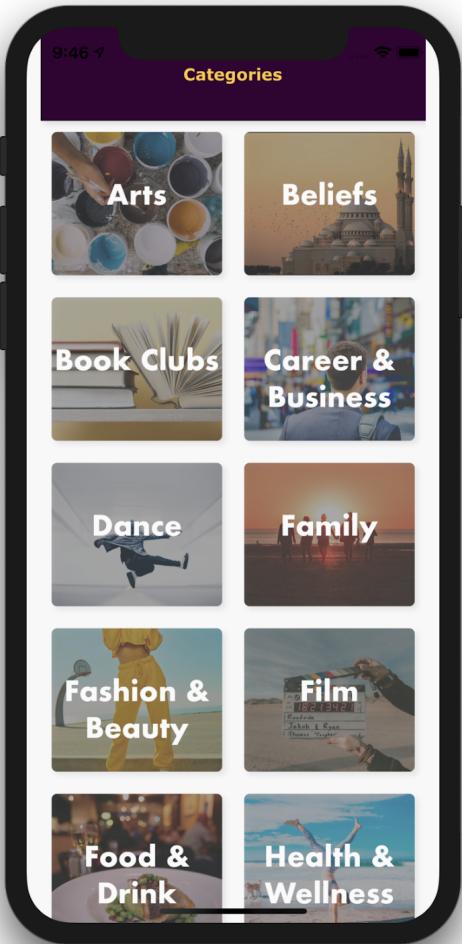
iPhone X - 12.1



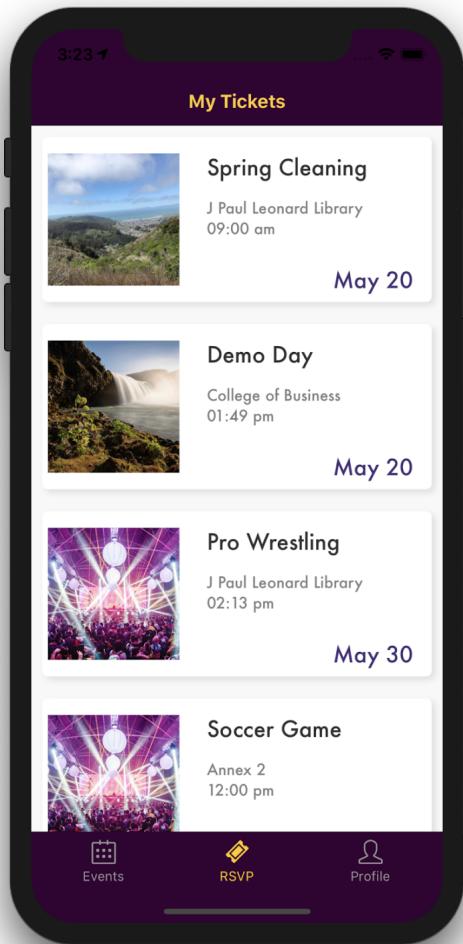
iPhone X - 12.1



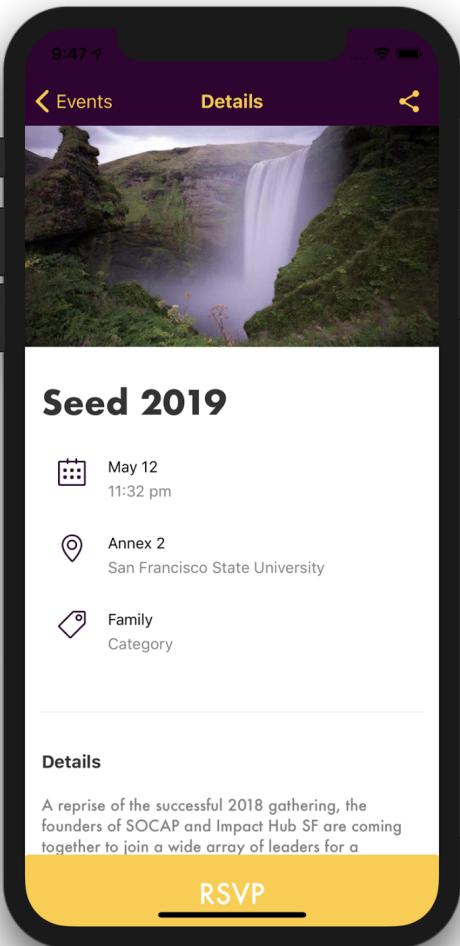
iPhone X - 12.1



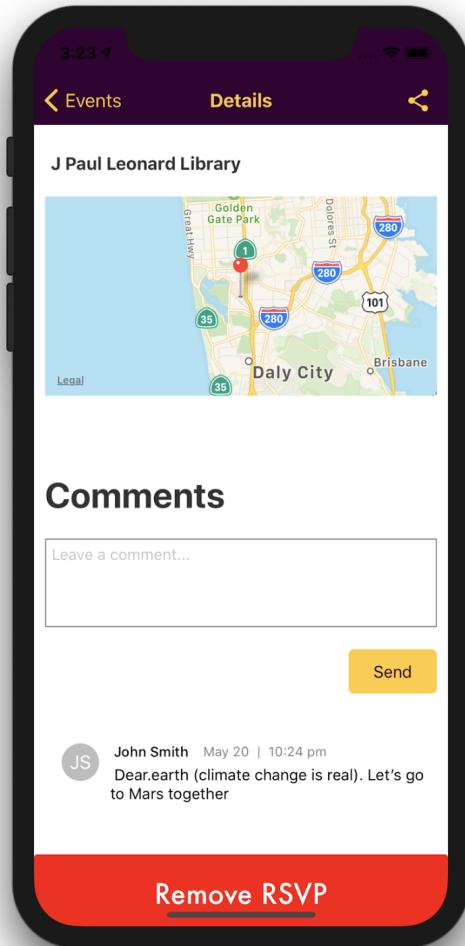
iPhone X - 12.1



iPhone X - 12.1



iPhone X - 12.1



iPhone X - 12.1

e. List of API routes and their descriptions.

\*for more details on the endpoints look at the full documentation here:

<https://github.com/csc667-02-sp19/csc667-sp19-Team04/blob/milestones/milestones/m5/EventUp%20REST%20API.pdf>

Routes Simplified:

1. User Routes

1.1. users/Login - POST

Login user and dispense JWT token

1.2. users/Register - PUT

Register user, insert into DB, and dispense JWT token

1.3. users/ - GET

Select user from DB using given id

1.4. users/RSVP - POST

Create RSVP and insert into join table

1.5. users/RSVP - DELETE

Delete RSVP entry

1.6. users/RSVP/:EventId(\d+) - GET

Grab RSVP using event id

1.7. users/posts - POST

Grab all posts for given user id

## 2. Event Routes

2.1. events/ - GET

Grab all events

2.2. events/ - POST

Create event and insert into DB

2.3. events/:id - DELETE

Delete event by id

2.4. events/:id - GET

Grab event by id

2.5. events/filter/:id - GET

Grab events according to a filter

## 3. Message Routes (1 to 1 relationship)

3.1. messages/send - POST

Create message and insert into DB

3.2. messages/:EventId - GET

Grab all messages for a given event id

## 4. Category

4.1. categories/ - GET

Grab all categories

## 5. Location

5.1. locations/ - GET

Grab all locations

## 6. Team Member Contributions

- a. **Cory Lewis** (Team Lead, GitHub Master)
  - i. Developed Routes on Back End
  - ii. Designed database Schema with Vincent
  - iii. Used best practices for Relational database queries (JOIN, LEFT JOIN, etc..)
  - iv. Performed roles of UX to check UI for simplicity
  - v. Back End Documentation editor
  - vi. Milestone editor
  - vii. Managed AWS Server restarts and pulling new versions of back end code
  - viii. Managed Trello for task management
  - ix. Intermediary of Back End to Front End
- b. **Mitul Savani** (Front End, GitHub Master):
  - i. I worked on the front-end side of the mobile application which includes integrating our application with our back-end services, along with brainstorming UI/UX design patterns for building features.
  - ii. Understand and analyze the needs of our end user(students) to bring cut-edge user experience.
  - iii. As a Github master, I monitored our GitHub repo to make sure no broken code is pushed into our stable branch (i.e) mater/production branch by reviewing all Pull Requests.

- c. **Alex Wolski** (Front End, Back End):
    - i. Helped determine what API calls would be needed on the backend and brainstormed the database schema.
    - ii. Worked on integrating the front end with features of iOS such as sharing events or saving them to the calendar.
    - iii. Discovered and fixed numerous bugs on the front end, such as the events feed not reloading or sloppy code causing errors.
  - d. **Vincent Santos** (Back End):
    - i. Created an EC2 instance on AWS to use as our web server to host our RESTful API
    - ii. Pushed the initial Back-end structure for our API following the standard MVC pattern for an express app. Helped create routes and controller functions thereafter.
    - iii. Helped with the brainstorming and implementation of database schema.
- IV. Incorporated native maps functionality in Events Detail View
- V. Completed Categorical filtering and its corresponding View for the Front-end

e. **Chintan Puri** (Front End)

- i. Responsible for developing the Front-end for and integrating with the Back-end following features : Events Feed, Create Events, Dropdown filtering, My Events, Comments sections and cancelling RSVP.
- ii. Worked on creating UI Mockups for all screens.
- iii. Assisted in integrating Login/Registration to the Back-end.

## 7. Project Reflection

We are proud of the app we were able to create. It has all of the base functionality we had planned to implement and looks great. However, we faced some challenges in the development of this app.

The first problem was that we didn't have a solid vision for the app before we started development. For example, one of our early ideas was to have a map displaying the wifi strengths around campus. And we advertised that this feature would be in the app. However as our app started to take shape, we realized that the feature wasn't a good fit and was a massive undertaking. So we weren't able to fulfill the promise to include it.

Another issue we had was dividing the work equally amongst the team. Initially, we had two developers on the frontend and three on the back. But halfway through development, we found that the backend was making much faster progress than the front. So we had a backend developer take the time to learn react native and switch to the frontend. If we had anticipated how much work the frontend would be, we could have assigned roles more efficiently at the start and saved time.

The last issue was that our timeframe was too slow. At the start of development, we were setting very easy goals for our team. And while we met all of our deadlines, our

progress was too slow. So as we approached the end of our development time, we realized that we were behind schedule and needed to speed up. Fortunately, we managed to finish the app on time. But we had to omit some features such as Wifi Map. If we had paced ourselves better, we could have added more features to the app.

## 8. Project Conclusion

This project served as a great learning experience for the entire team. Many of us had little experience in developing with React Native and the other part of our team had little experience setting up a Back End for a Rest API. Each team member was challenged in their own way to deliver code to implement necessary features. Collaborating with Discord we were able to keep decent communication despite all of us having busy schedules. This project served as a great learning experience for the team lead who tried to create, assign, and manage tasks to the best of his ability so all team members were engaged and the product was delivered on time. The Front End team had their own challenges with images and UI implementation with the added integration of using the Rest API routes. Back End started slow at the beginning of the semester because the server needed to be provisioned and the database initialized but once these things were completed we were able to design many routes for the Front End team to use efficiently and with ease. Overall this project served as a great learning experience for many of us who were able to work with SQL, Node.js, Express.js, bcrypt, jwt, React Native, and HTTP requests which can be applied in real world application.