Spring 2016

ClockWork Memo

MARCH 6, 2016

BY CULLIN LAM, ANDRY LORA, SON NGUYEN

PROJECT GOALS REVIEW

ClockWork is a hybrid mobile application that seeks to stream line the planning it takes to hang out with your friends. ClockWork provides its users the ability to quickly publish and join events. It removes the hassle of having to call, text, or message your friends to find who is available to hang out.

PROJECT FEATURES

Alpha

The features required for Alpha release are as follows:

A. Event Feed (Public Access)
B. Event Detail (Public Access)
C. Event Add
D. Sign in Account System

Event Feed

The Event Feed view shall serve as the main page of the ClockWork Application. Here, users will be able to view published events. In addition, users will be able to navigate to additional views, such as the Event Detail and Event Add views.

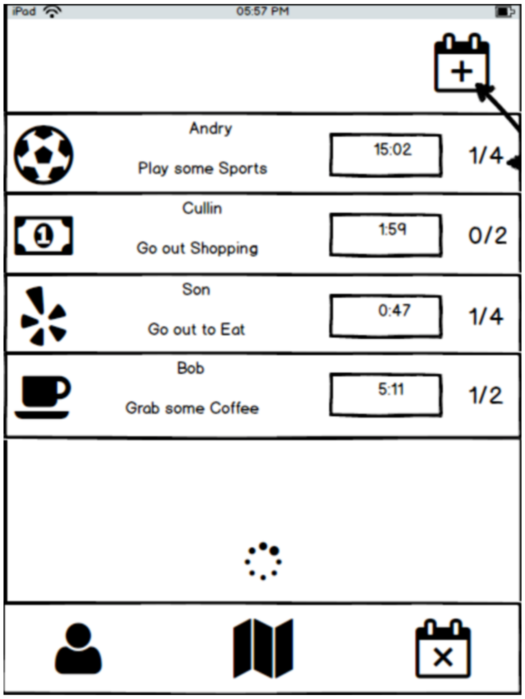


Figure A. Event Feed View

Event Detail

The Event Detail view shall provide users with more information about published events on the Event Feed view. An expanded description of the event shall be displayed. Also a list of users attending shall appear. Lastly the attend button shall be placed here so that those interested may notify the event owner that they will join.

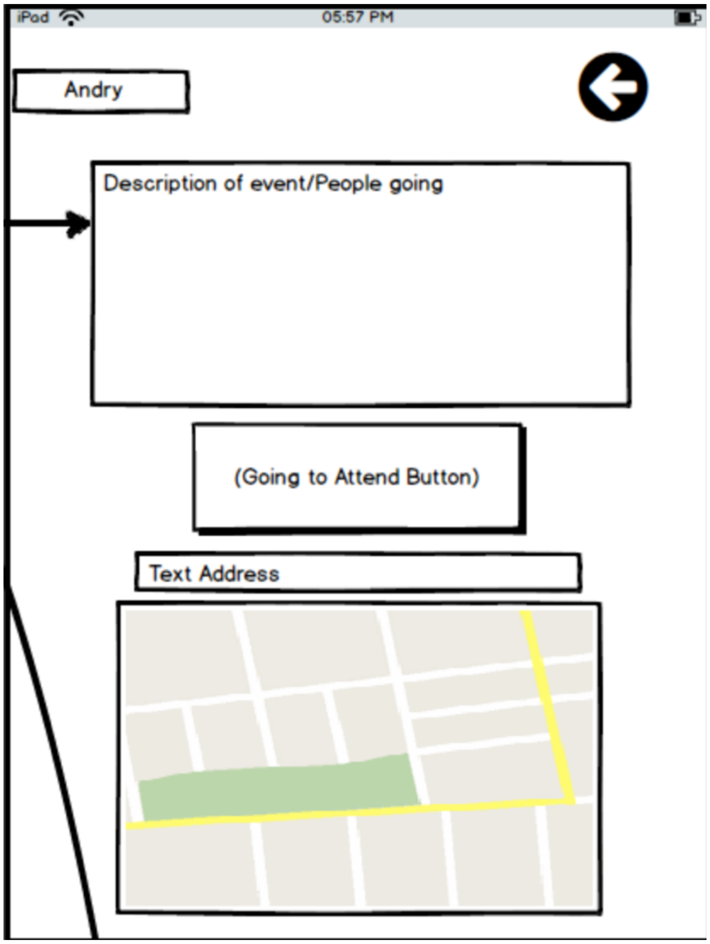


Figure B. Event Detail View

Event Add

The Event Add view shall display to the user a form that will be used to create a new event. A drop down selection for event type i.e. food, game, movie, sport, etc. will constitute one of the fields. In addition, the user will be required to provide a description of the event. We have decided to allow the address field to be optional, as it will allow flexibility for interested users to agree on an appropriate location for the event. Lastly event expiration time is a required field.

This field sets the duration of how long the event will remain active on the event feed. After the allotted time the event will be closed to new attendees, and will be removed from the feed. The event owner and the attendees will receive a notification that the event is now closed and the event members should begin communicating on event specifics or begin their event.

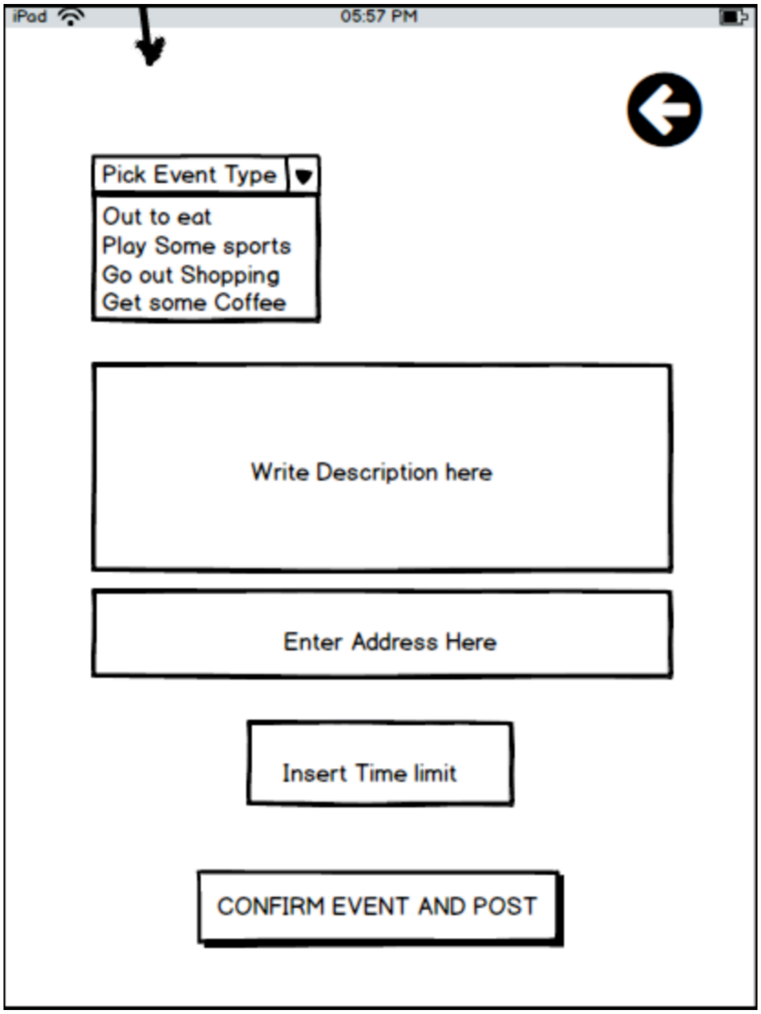
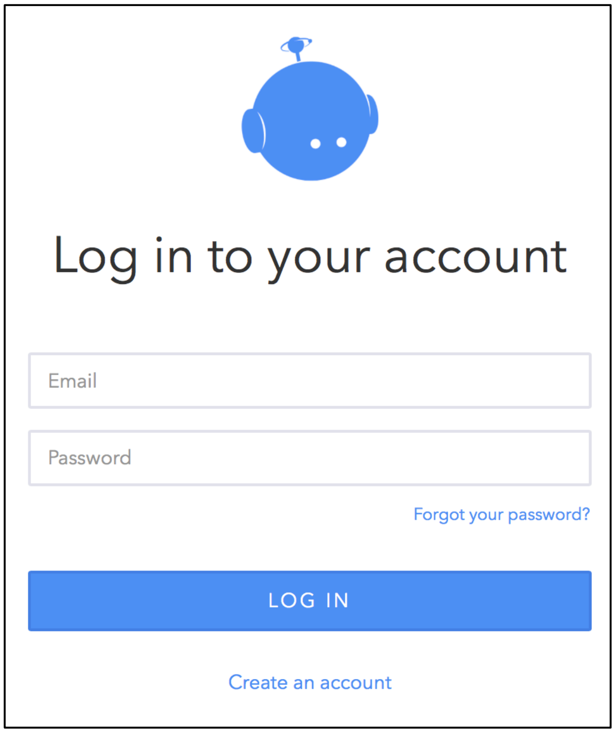


Figure C. Event Add View

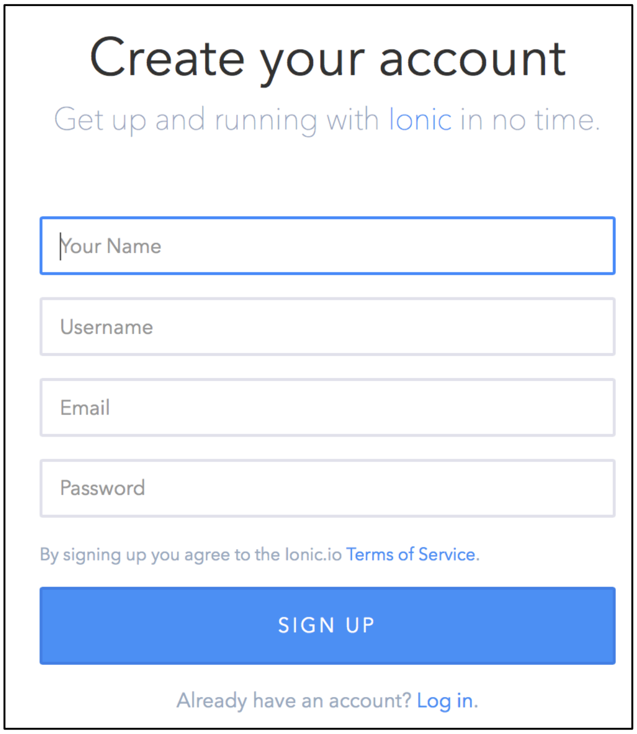
Sign-In Account System

This application shall have a user account system that stores user information such as login email and encrypted password. In addition, profile information shall also be stored. Users will be required to login in order to use this application. A login in view shall be displayed if the user is not signed in. From this view the user may also sign up for an account.



Log in to your account

Figure D. Login View



Create your account

Get up and running with Ionic in no time.

By signing up you agree to the Ionic.io Terms of Service.

Already have an account? Log in.

Figure E. Sign Up View

img src: https://apps.ionic.io/

Beta

The features required for Beta release are as follows:

I. Event Timer
II. Event Feed Infinite Scrolling
III. User Profile View
IV. Ability to Add friends

Event Timer

We hope to implement a live timer on the Event Feed View for each event. Similar to an auction this timer will count down the time remaining for users to join an event before it is closed. Server code shall check active events and close any expired events (See Figure A).

Event Feed Infinite Scrolling

Once the number of active events becomes quite large, it will be impractical to force the client to load all the events in the event feed view. Instead we shall implement infinite scrolling and have the client fetch additional events as the user scrolls down.

User Profile View

We hope to implement a public user profile view for each user, so that other users may be able to easily identify and recognize each other. Users will be able to share a short bio or description about themselves.

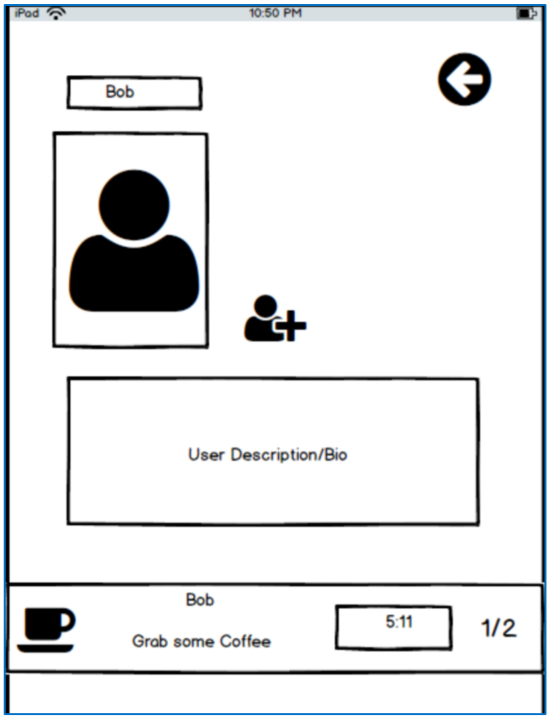


Figure F. User Profile View with add friend button

Add Friends

We hope to implement an add friend feature. Adding friends will allow the user to see their friends' events on the Event Feed view. This feature will necessitate a notification system to approve friend requests. The alpha build will allow any user to see all the Events on the Event Feed.

Final Release

Additional features required for a full final release are as follows:

I. Ability to leave event
II. Symbols
III. Event Sorting
IV. Event Feed Search Bar
V. Friend List View
VI. Flaky User Rating
VII. In app Messaging
VIII. OAuth login

ASSESSMENT OF PROJECT TIMELINE

DATE	ITEM	Responsibility
[Feb-02-2016]	Set Up Sub Directories in Git	[Cullin/Andry/Son]
[Feb-10-2016]	Web Template of UI	[Cullin/Andry/Son]
[Feb-12-2016]	Research Unlimited Scrolling	[Cullin]
[Feb-14-2016]	Research Image Upload	[Andry]
[Feb-15-2016]	Implement Sign In	[Son]
[Feb-16-2016]	Implement Event List	[Cullin]
[Feb-20-2016]	Usability Testing	[Cullin/Andry/Son]
[Mar-11-2016]	Implement Event Detail	[Cullin/Andry/Son]
[Mar-12-2016]	Implement User Profile	[Andry]

[Mar-14-2016]	Timer Function	[Cullin/Andry/Son]
[Mar-15-2016]	Google Map API Location of Event	[Cullin/Andry/Son]
[Mar-20-2016]	Test Cross Platform	[Cullin/Andry/Son]
[Mar-21-2016]	Beta Testing	[Cullin/Andry/Son]
[Apr-10-2016]	Completion	[Cullin/Andry/Son]

As of March 7th 2016, the team has completed a majority of the tasks scheduled to be completed by this date. The following originally proposed tasks have been completed:

- Set Up Sub Directories in Git
- Web Template of UI
- Research Unlimited Scrolling
- Implement Event List
- Usability Testing
- Implement Event Detail
- Implement Timer Function

The two tasks, implement Event Detail and Implement Timer Function were completed ahead of schedule. These features were meant to be implemented for beta release however we were able to complete them much earlier. Unfortunately, the Sign-In View feature was unable to be delivered satisfactorily by the intended due date. In the grand scheme however we are ahead of schedule and we are on track to meet our beta release goals.

PROJECT ISSUES

To date we have encountered several problems and have had to adjust accordingly. The first issue we encountered was the disruption of our Ionic views through the use of our directive styled components. Angular 1.X best practices calls for the use of Angular directives to create components which are the combination of html templates and view controllers. Following this style caused Ionic Views to improperly render and broke some navigation features. We opted to go back to the old style which consists of defining an html view and controller separately and linking them together in the route configuration.

Another issue that one of us discovered is the lack of a string address to latitude/longitude coordinate conversion function in the google maps angular API. After doing additional research we found resources for the Google Geo Decoder API which we plan on using to convert string addresses into appropriate coordinates to display as markers on the map.

As the amount of views and features increased we realized that a tabbed layout would not be sufficient for a good user experience. We decided to convert our application from a tabbed layout to a side menu layout with the hamburger icon. We believe this will allow us to insert easy and intuitive navigation for the user to access our application’s features.

Form validation for our Event Add view has turned out to be more difficult to implement well then originally expected. Our Event Add view is found inside a modal with it’s own controller logic. We are currently struggling to figure out a way to reset the form after data has been successfully submitted upon the closing or opening of the modal.

On mobile platforms such as IOS, screen size has become an issue. Since the width of the screen is smaller than the desktop browser much of the Event Title text is truncated. We need to use CSS to allow for text overflow to a new line so users will be able to see more of the event title.

One final issue is code development patterns. Much of our work is done independently of other group members. A feature was unable to be delivered on time due to the teammate misunderstanding the project goal and thus time and resources was wasted. Moving forward we will work collaboratively and more closely together to ensure the success of each teammate's contributions.

EVALUATION OF IMPLEMENTED FEATURES

Event Feed

The Event Feed feature performs as expected. New events appear on the feed upon creation. Clicking on the event brings the user to the appropriate Event Detail view. Pressing the compose icon on the upper right corner causes the Event Add modal to appear. Lastly pressing the hamburger icon opens the side menu.

Improvements

The Event Feed should render a different icon for each event depending on the event type. This feature still needs to be completed. In addition, CSS should be applied to allow overflow of text to a new line to prevent over truncation of text.

Event Detail

The Event Detail currently displays the required information. It displays the event title and description. It also displays the event owner, event expire time, countdown timer, and the number of spots remaining for users to join. It also currently displays a radio button to attend the event.

Improvements

The Event Detail view needs to be styled better. Headers need to be added to each data item so users can easily identify what each represents. The radio button can be changed to a regular button that is more visually appealing. Lastly Google Maps is not yet implemented and thus an empty object is currently being displayed at the bottom of the page.

Event Add

The Event Add modal view has input fields for the event's title, description, type, capacity, and countdown timer duration. There is also a submit button that inserts a new record into the database and closes the modal.

Improvements

The Event Add controller currently doesn't insert the event type into the record. This needs to be corrected once the final event types are determined. Angular validation needs to be set to appear only on submit and dirty forms. The form needs to reset to pristine after the event is submitted. Better styling of error messages needs to be done.

Countdown Timer

The countdown component is used in both the Event Feed and Event Detail views. It reactively decrements the amount of seconds until the countdown reaches zero every second. A date filter is applied to convert it into H:M:S format. This feature does not need any improvements.

FEATURE FEASABILITY TESTING

The following features have yet to be tested for feasibility

- Infinite Scrolling on Event Feed

- User View & Edit Profile View
- Add Friend Feature