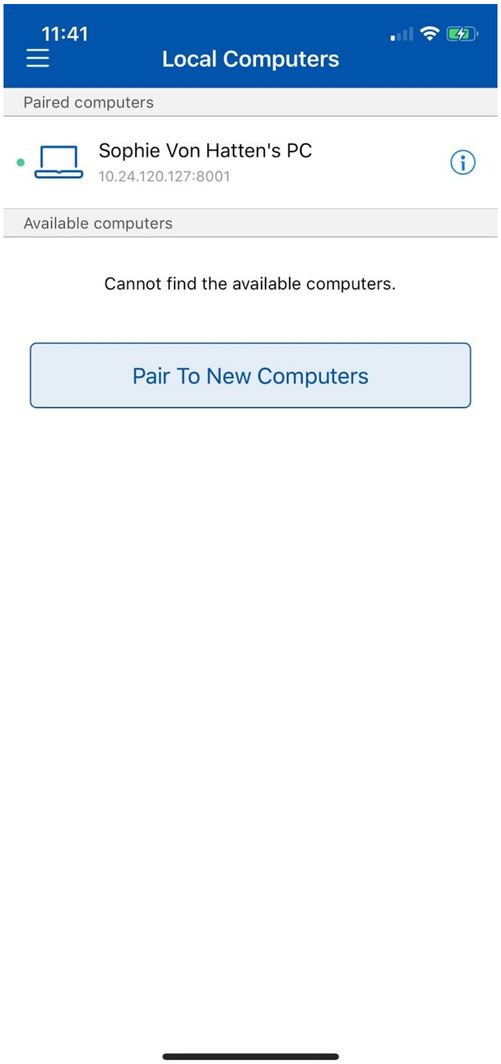
## Testing using Monaca Debugger:

To check on the functionality, CSS, and the overall user-flow of *Host-It*, we decided to use *Monaca Debugger*, an application used for just as its name states, debugging, as well as testing the functionality of a given app on a real device. This process is done via connection between a PC and handheld device. The following images and descriptions refer to testing and debugging that occurred during the implementation of, as well as the active use of, *Monaca Debugger*.

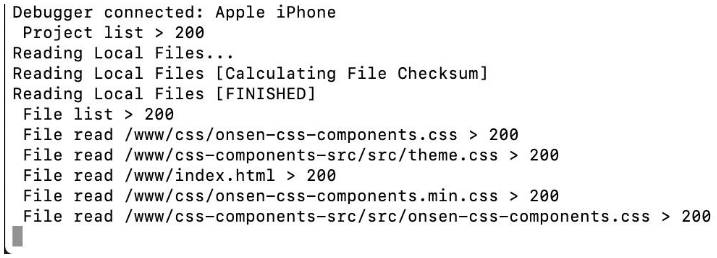
By implementing *Monaca Debugger* into our app, we were able to debug and fix any vulnerabilities that may have remained within our packages; this in addition deleted packages left over in our code that were deemed unnecessary to our final product, making our code a bit cleaner.



Sophie Von Hatten's PC
10.24.120.127:8001

Cannot find the available computers.

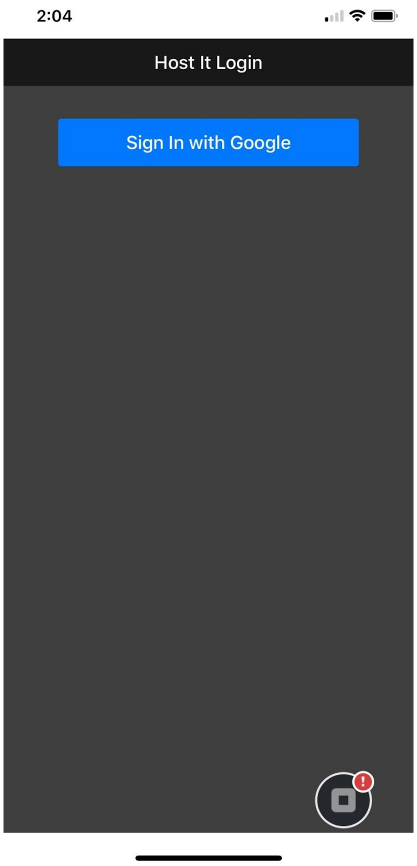
Once the connection between our designated PC and handheld device had been successfully made via IP address and cloud connection between the PC's login command and the cellphone's *Monaca Debugger* application login, we were able to view and interact with *Host-It* through the cellphone's *Monaca* app.

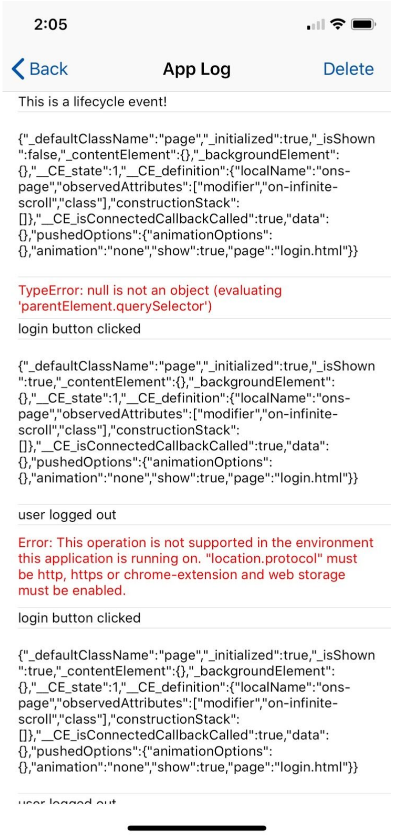


```
Debugger connected: Apple iPhone
 Project list > 200
Reading Local Files...
Reading Local Files [Calculating File Checksum]
Reading Local Files [FINISHED]
 File list > 200
 File read /www/css/onsen-css-components.css > 200
 File read /www/css-components-src/src/theme.css > 200
 File read /www/index.html > 200
 File read /www/css/onsen-css-components.min.css > 200
 File read /www/css-components-src/src/onsen-css-components.css > 200
```

*note: the “File read [...]” lines from the PC’s computer were from *Monaca Debugger* taking in the css changes from a light them to a dark theme, as well as various other css changes made. (see login image with dark theme implemented below)

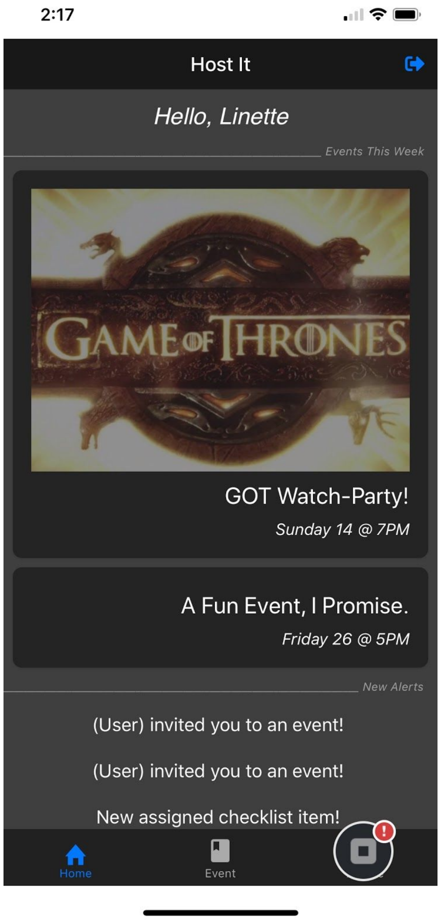
**Login Page:**





```
This is a lifecycle event!

{"_defaultClassName":"page","_initialized":true,"_isShown
":false,"_contentElement":{},"_backgroundElement":
{},"__CE_state":1,"__CE_definition":{"localName":"ons-
page","observedAttributes":["modifier","on-infinite-
scroll","class"],"constructionStack":
[]},"__CE_isConnectedCallbackCalled":true,"data":
{},"pushedOptions":{"animationOptions":
{},"animation":"none","show":true,"page":"login.html"}}

TypeError: null is not an object (evaluating
'parentElement.querySelector')
login button clicked

{"_defaultClassName":"page","_initialized":true,"_isShown
":true,"_contentElement":{},"_backgroundElement":
{},"__CE_state":1,"__CE_definition":{"localName":"ons-
page","observedAttributes":["modifier","on-infinite-
scroll","class"],"constructionStack":
[]},"__CE_isConnectedCallbackCalled":true,"data":
{},"pushedOptions":{"animationOptions":
{},"animation":"none","show":true,"page":"login.html"}}

user logged out
Error: This operation is not supported in the environment
this application is running on. "location.protocol" must
be http, https or chrome-extension and web storage
must be enabled.
login button clicked

{"_defaultClassName":"page","_initialized":true,"_isShown
":true,"_contentElement":{},"_backgroundElement":
{},"__CE_state":1,"__CE_definition":{"localName":"ons-
page","observedAttributes":["modifier","on-infinite-
scroll","class"],"constructionStack":
[]},"__CE_isConnectedCallbackCalled":true,"data":
{},"pushedOptions":{"animationOptions":
{},"animation":"none","show":true,"page":"login.html"}}

user logged out
```

To the right of the “Host-It Login” page, an image of the “App Log” is shown to showcase console logs for logging in and out (e.g. “login button clicked” and “user logged out”), as well as overall page navigation and activity. You get to this log by hitting the floating button in the bottom right hand corner.



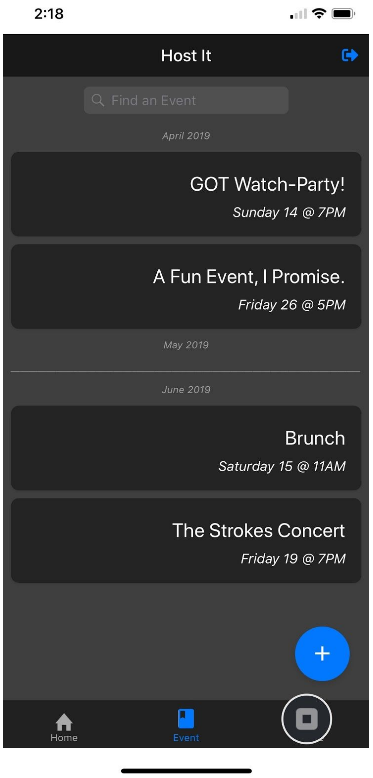
(User) invited you to an event!

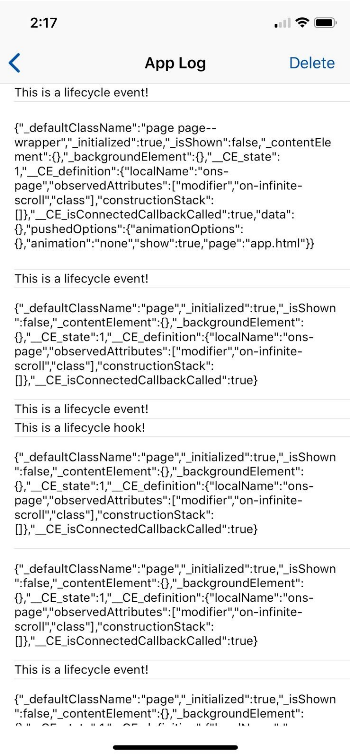
(User) invited you to an event!

New assigned checklist item!

## Home Page/Events Page:

Once logged in via Gmail account, we are brought to the home page, where we see our upcoming, already made events for the user (in this case, Linette’s account). We used console logs that show up in the App Log in the *Monaca Debugger* : “This is a lifecycle event!” This proves that what is being shown, i.e. the “GOT Watch-Party!”, “A Fun Event, I Promise.”, “Brunch”, and “The Strokes Concert” (see Events Page and associated App Log).



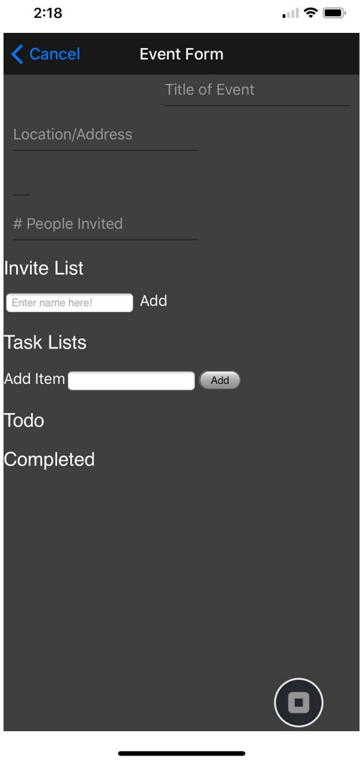


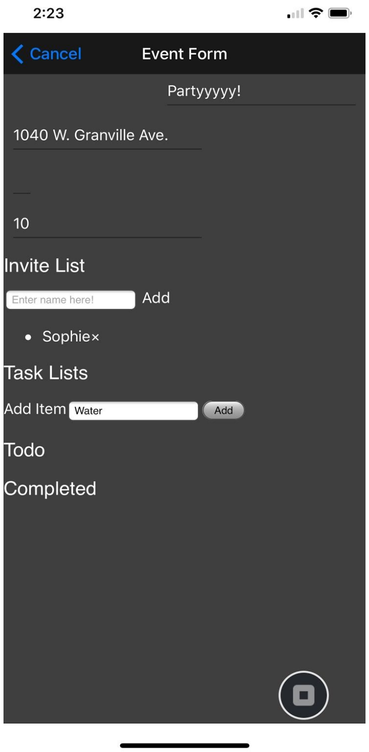
```
This is a lifecycle event!

{"_defaultClassName":"page page--
wrapper","_initialized":true,"_isShown":false,"_contentEle
ment":{},"_backgroundElement":{},"__CE_state":
1,"__CE_definition":{"localName":"ons-
page","observedAttributes":["modifier","on-infinite-
scroll","class"],"constructionStack":
[]},"__CE_isConnectedCallbackCalled":true,"data":
{},"pushedOptions":{"animationOptions":
{},"animation":"none","show":true,"page":"app.html"}}

This is a lifecycle event!

{"_defaultClassName":"page","_initialized":true,"_isShown
":false,"_contentElement":{},"_backgroundElement":
{},"__CE_state":1,"__CE_definition":{"localName":"ons-
page","observedAttributes":["modifier","on-infinite-
scroll","class"],"constructionStack":
[]},"__CE_isConnectedCallbackCalled":true}

This is a lifecycle event!
This is a lifecycle hook!

{"_defaultClassName":"page","_initialized":true,"_isShown
":false,"_contentElement":{},"_backgroundElement":
{},"__CE_state":1,"__CE_definition":{"localName":"ons-
page","observedAttributes":["modifier","on-infinite-
scroll","class"],"constructionStack":
[]},"__CE_isConnectedCallbackCalled":true}

{"_defaultClassName":"page","_initialized":true,"_isShown
":false,"_contentElement":{},"_backgroundElement":
{},"__CE_state":1,"__CE_definition":{"localName":"ons-
page","observedAttributes":["modifier","on-infinite-
scroll","class"],"constructionStack":
[]},"__CE_isConnectedCallbackCalled":true}

This is a lifecycle event!

{"_defaultClassName":"page","_initialized":true,"_isShown
":false,"_contentElement":{},"_backgroundElement":
```

*We eventually made the decision to make the Events Page the actual Home Page due to the redundancy of having both in *Host-It*.

## New Event Creation Page:

Once you hit the floating blue button in the bottom right hand corner of the device's screen, it navigates to another page for creating a new event, which has been tested (succeeds in making that navigation happen, as well as being able to cancel the creation of the new event).





Partyyyyy!

1040 W. Granville Ave.

10

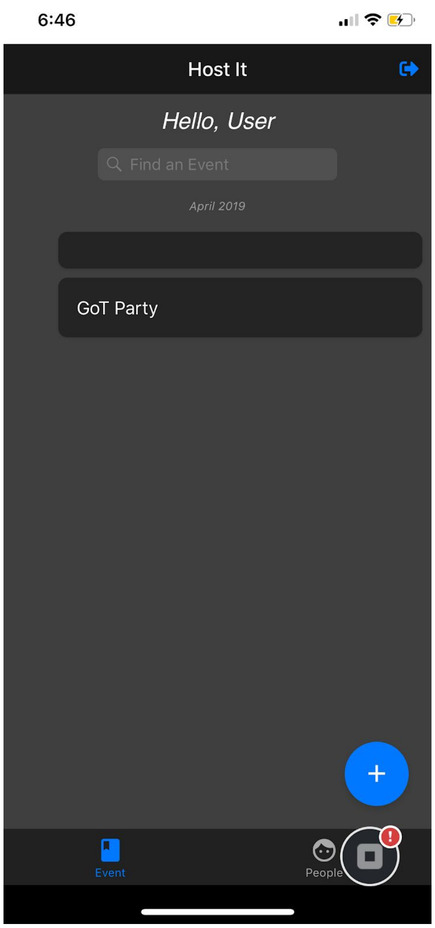
- Sophie×

Water

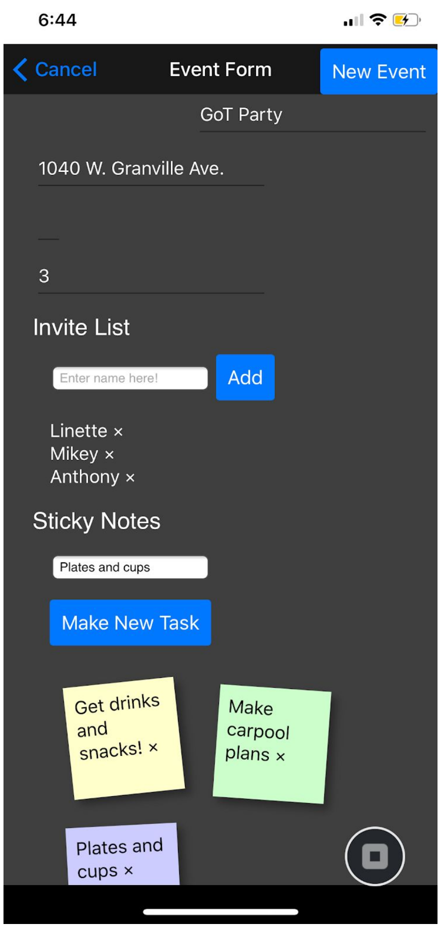
*Note: Shows that the Event Form is editable, being able to designate an event name, a location, a number of invites and an invite list, as well as a task list that categorizes said list into what still needs to be done, the "Todo," and what has been "Completed."

## Final Testings:

Leading into our final submission due date, we completed final testing of functionality and UI, making sure that everything works and looks as it should be for final submission. The following screenshots are said tests:



GoT Party



GoT Party

1040 W. Granville Ave.

3

Enter name here!

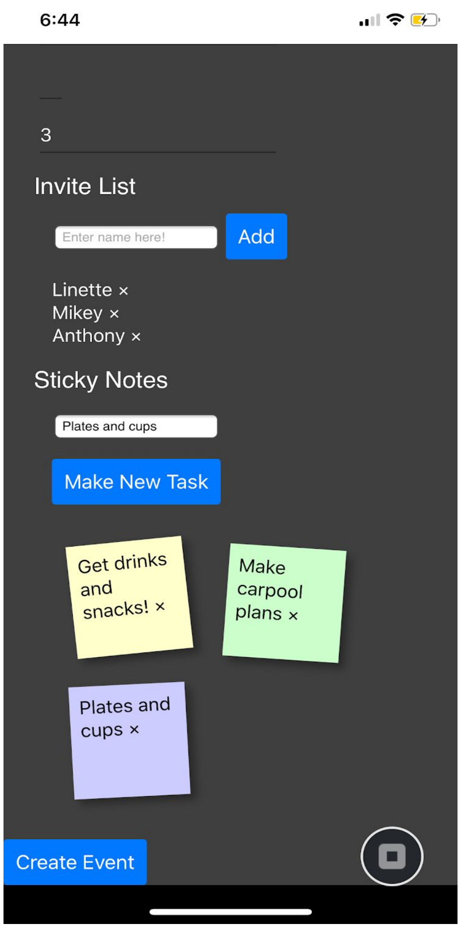
Linette ×
Mikey ×
Anthony ×

Plates and cups

Get drinks and snacks! ×

Make carpool plans ×

Plates and cups ×



3

Enter name here!

Linette ×
Mikey ×
Anthony ×

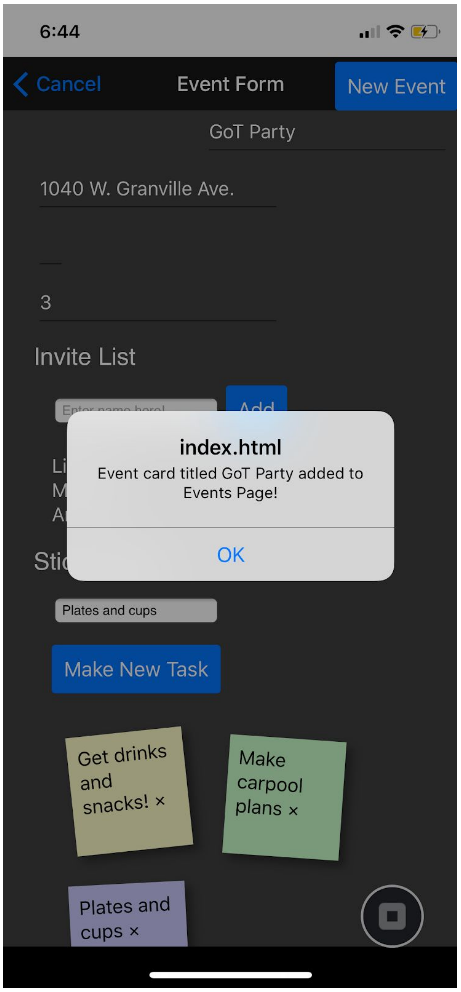
Plates and cups

Get drinks and snacks! ×

Make carpool plans ×

Plates and cups ×

*Note: since the earlier screenshots of our testing sessions, we have implemented sticky notes as the appearance for tasks. Below, we used on-screen alerts to notify us of certain actions, such as the making of an event, as well as the act of searching for a specific event in the Event List Page:



GoT Party

1040 W. Granville Ave.

3

Invite List

Enter name here!

Li
M
Ar

Stic

**index.html**

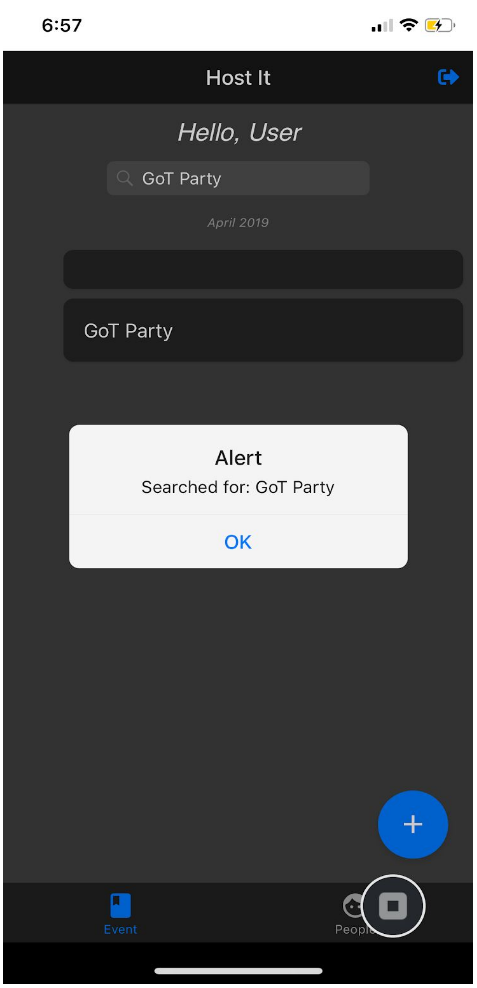
Event card titled GoT Party added to Events Page!

Plates and cups

Get drinks and snacks! ×

Make carpool plans ×

Plates and cups ×



*Hello, User*

GoT Party

*April 2019*

GoT Party

**Alert**

Searched for: GoT Party