4 ½ FRAT BOYS - RUSHME

FINAL RELEASE

07 DECEMBER 2017

TEAM MEMBERS

Ben Crevier, Jim Hines, Adam Kuniholm, Andie Labgold, and Stephen Notley

FINAL RELEASE

Below are the instructions for installing and hosting both forms of the application. The code is this zip file.

RushMe Website Setup Guide

1. Download XAMPP
    - ➔ https://www.apachefriends.org/index.html

2. Find the XAMPP folder on your computer

3. Move project files inside /XAMPP/htdocs

4. Run Server
    - ➔ Within the XAMPP folder, open the ‘Manager’ application
    - ➔ Click the ‘Manage Servers’ tab
    - ➔ You should see ‘Apache Web Server’ listed. Select this server, and click the ‘Start’ button.
    - ➔ The server should update its status to ‘Running’

5. Load the page
    - ➔ Open a Chrome browser window
    - ➔ In the web address bar, go to localhost/RushMeWeb/index.php

6. The page should take a few seconds to load, then you will be ready to go!

RushMe iOS Application Setup Guide

1. Give Adam Kuniholm your email

2. Follow instruction in email to download TestFlight and redeem the app code that is found in the email

FINAL TEST RESULTS

Overview:

The team conducted usability testing as described in the revised testing plan submitted. This allowed for the simultaneous testing of user experience and the promised functionality, as each task corresponds with our listed test cases (shown below). Usability testing was conducted on volunteers who had never had any previous interaction of the application. Testing was conducted separately for the website and app, and so the results and recommendations for deployment are documented here separately. Five users tested the website and five additional users tested the app. Both female and male, affiliated and unaffiliated users were tested.

Test Cases:

Test Case:	Steps:	Expected Results:
Test Case 1: Use Case: **SeeFraternityDescription**	1. View the page of fraternity X	The description of fraternity X is the same as expected/pulled from the database
Test Case 2: Use Case: **SeeFraternityCalenderImage**	1. View the page for fraternity X	The calendar image data is the same as the calendar image stored on the database for fraternity X
Test Case 3: Favorite one Fraternity Use Case: **FavoriteFraternity**	1. Load RushMe 2. Select the favorite option for Chi Phi	Chi Phi is on the favorites page
Test Case 4: Favorite multiple fraternities Use Case: **FavoriteFraternity**	1. Load RushMe 2. Select the favorite option for Chi Phi 3. Select the favorite option for AEPI	Chi Phi, Alpha Epsilon Pi and Pi Kappa Alpha are all on the favorites page

	4. Select the favorite option for Pi Kappa Alpha	
Test Case 5: View Custom Calendar empty Use Case: **ViewCustomCalendar**	1. Load RushMe 2. Select the Calendar option	Calendar should have no events listed on it.
Test Case 6: View Custom Calendar Use Case: **ViewCustomCalendar**	1. Load Rush Me 2. Favorite Fraternity X 3. Navigate to Custom Calendar	Assert events on custom calendar match events for fraternity X
Test Case 7: View next rush Event Use Case: **SeeNextEvent**	1. Load RushMe 2. Select Fraternity X	Fraternity information page displays the next rush event for this fraternity
Test Case 8a: Export Custom Calendar Web Use Case **ExportCustomCalendarWEB**	1. Load RushMe 2. Favorite Fraternity X and Fraternity Y 3. Navigate to the calendar tab and select download	Every event stored on the database for Fraternity X and Fraternity Y is included on the downloaded ics file
Test Case 8b: Export Custom Calander App Use Case: **ExportCustomCalendarAPP**	1. Load RushMe 2. Favorite Fraternity X and Fraternity Y 3. Navigate to calendar tab hit share button	Every event stored on the database for Fraternity X and Fraternity Y is included on the downloaded ics file
Test Case 9: Unfavorite Fraternity Use Case: **FavoriteFraternity**	1 . Load Rush Me 2. Select the favorite option for Fraternity X 3. Verify Fraternity X is added to the favorites page 4. Select the unfavorite option for Fraternity X	No fraternities are on the favorites page

Usability Testing Tasks:

TASK 1 (TEST CASES 1 & 2)

- Scenario: You are a student who is interested in rushing a fraternity. You remember meeting Chi Phi at the IFC kickoff, but would like to learn more about them.
- Task: Please use the application to get more information about the Chi Phi Fraternity
- Success: You have found more information about the Chi Phi fraternity.

TASK 2 (TEST CASE 7)

- Scenario: You would like to know when Chi Phi is holding its next event.
- Task: Use the application to find out when Chi Phi is holding its next event.
- Success: You know when the next event is.

TASK 3 (TEST CASE 3 & 4)

- Scenario: You have gotten to know the fraternities at RPI better and have come to like Chi Phi, Alpha Epsilon Pi, and Pi Kappa Alpha. You would like to distinguish your favorites from the rest of the fraternities.
- Task: Use the application to designate these three fraternities as you favorites.
- Success: You have three fraternities favorited.

TASK 4 (TEST CASE 6)

- Scenario: You would like to know when your favorite fraternities are holding rush events.
- Task: Use the application to find out when your favorite fraternities are holding rush events.
- Success: You know when the fraternities are holding events.

TASK 5 (TEST CASE 8)

- Scenario: You would like to have the rush events from your favorite fraternities synced to your google calendar.
- Task: Use the application to download the calendar.
- Success: You have downloaded the calendar.

Results- Website:

Michelle (Unaffiliated)

Task	Rating	Succes s	Comments
1	10	Yes	“Really straightforward”
2	10	Yes	“really easy, all the information is in one place”
3	6	No	She did not find the favorites tab “It didn’t look like the buttons changed anything”
4	9	Yes	“Hard to see tabs, I was focused on the cards”
5	10	Yes	“Very visible button”
Overall	“I like the pictures, and how they are all set up the same way. It’s very neat, and I like looking at it. I think the favorite button should change when you click it. I liked it though.”		

Hunter (Affiliated)

Task	Rating	Success	Comments
1	10	Yes	Suggested search bar to find the frats easier
2	10	Yes	
3	10	Yes	“Straightforward”
4	9	Yes	Was excited to see only favorite frats in calendar
5	10	Yes	“Can’t miss it”
Overall	“For a student looking to join a fraternity, its essential. So much better than handing out calendars. Gives you all the event and frat infor you would want. I really like the favorite feature. Doesn’t get easier than that. If that existed when I rushed I would have absolutely used it. Would recommend.”		

Riley (Unaffiliated)

Task	Rating	Success	Comments
1	10	Yes	Tried to click on picture on card to pull up dialog. “Icon is hard to see at first”
2	7	Yes	Went to calendar first because expected events to be under calendar
3	9	Yes	Didn’t know where they would show up when favorited. Didn’t realize there was a page for favorites “The heart should be highlighted when favorited so they stand out more”
4	10	Yes	“Maybe if there is only one event, you just show that instead of ‘1 Event’”
5	10	Yes	
Overall	“Very useful, not for me though because I’m not interested in rush. I liked the alphabetical order. I feel like this could be extended to other clubs.” Had issue recognizing tabs.		

Nate (Unaffiliated)

Task	Rating	Success	Comments
1	8	Yes	“Finding the frat was easy” Thought you could open dialog by clicking on the frat name on card
2	8	Yes	Expected to see them under calendar tab first
3	10	Yes	“Very easy, the heart is a universal favorite sign”
4	10	No	Didn’t go to calendar tab Went to individual frat pages instead to see them one-by-one
5	10	Yes	“Its right there. It makes sense”
Overall	“Looks good, very helpful. It's a thing I could see people using. It's hard to keep track of all the events during rush. The user interface is nice. I like the blue”		

Jake (Affiliated)

Task	Rating	Success	Comments
1	10	Yes	“Nice to see everything on one page”
2	10	Yes	
3	10	Yes	“It would be nice if the heart lit up when you pressed it so you know you favorited it.”
4	10	Yes	
5	10	Yes	“Straightforward nice big button”
Overall	“It was pretty easy. It would definitely be a good tool for freshman to use as far as getting to know when fraternities are having events. The heart should light up and highlight so you know you have favorited it.”		

Recommendation- Website:

Overall, the testing went very well with the website. The functionality passed all test cases, as no errors/unexpected behavior was encountered during the testing. There was some feedback on some usability features, namely making the event information more obvious on the page and making a more noticeable indication of favoriting.

The team feels that the website is **ready for deployment** based on this testing, as all desired functionality worked as expected and the only feedback indicated minor UI updates would be helpful, which are changes which can easily be rolled out in patches and the like.

Results - App:

Jon (Affiliated)

Task	Rating	Success	Comments
1	10	Yes	
2	7	Yes	“Needs some sort of header indicating where it is”
3	9	Yes	“Needs an indicator showing you have successfully favorited”
4	8	Yes	
5	10	Yes	
Overall	“It is a great tool for anyone interested in rushing multiple houses. It clearly outlines all the events from the favorited houses in one calendar.		

Brendan (Affiliated)

Task	Rating	Success	Comments
1	10	Yes	
2	10	Yes	
3	5	No	“Side Menu froze while I was trying to figure out how to do it”
4	10	Yes	
5	2	No	“Would like a more prominent button”
Overall	“App was overall well designed and looked nice. Not all of the features were as easy to use as I would have wanted.”		

Andrew (Affiliated)

Task	Rating	Success	Comments
1	10	Yes	
2	10	Yes	
3	8	Yes	
4	7	Yes	“Didn’t know there was a slideout menu”
5	9	Yes	
Overall	“Calendar should indicate what day I’m highlighting, and fraternities should indicate the day by which you have to sign. I do not like that fraternity GPAs are listed.”		

José (Unaffiliated)

Task	Rating	Success	Comments
1	10	Yes	
2	8	Yes	“Difficult to tell what date the first event is on so it’s unclear that it’s the first event”
3	10	Yes	“Would like common abbreviations or nicknames for the fraternities because I don’t know all the letters”
4	9	Yes	
5	10	Yes	
Overall	“Enjoyed the app, though I am unsure I would use it.”		

David (Affiliated)

Task	Rating	Success	Comments
1	10	Yes	
2	10	Yes	
3	8	Yes	“Finding the favorites was a little unintuitive.”
4	9	Yes	
5	8	Yes	“Wasn’t too bad.”
Overall	Pretty good app, much better way to stay organized than having rush calendars scattered all over your room.		

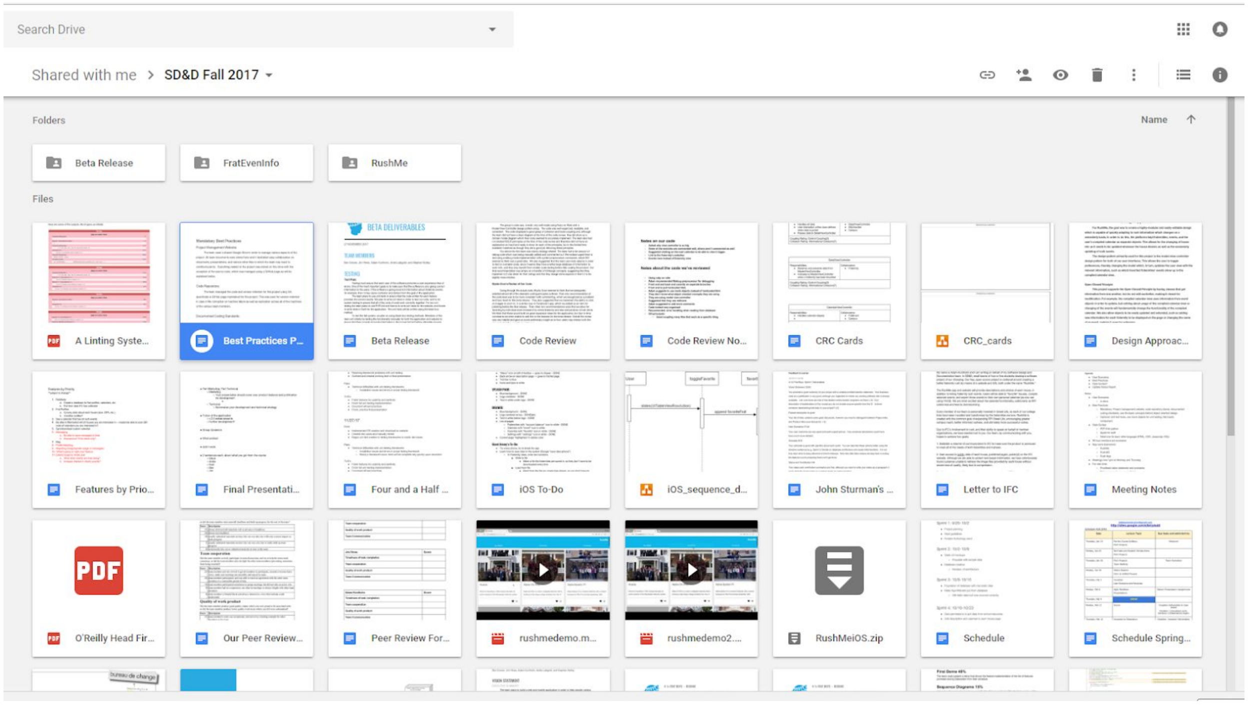
Recommendation- App:

The feedback from the app was very positive. The users suggested some UI changes to make some actions more intuitive, but tasks were largely able to be completed nonetheless. There was one bug found, which the team was unable to recreate, however, it was not a fatal bug and the app restored to normal function in a matter of seconds. Because these are things that could easily be rolled out in patches and do not interfere with the functionality of the app as a whole, the team’s recommendation is that the app is **ready for deployment**.

MANDATORY BEST PRACTICES

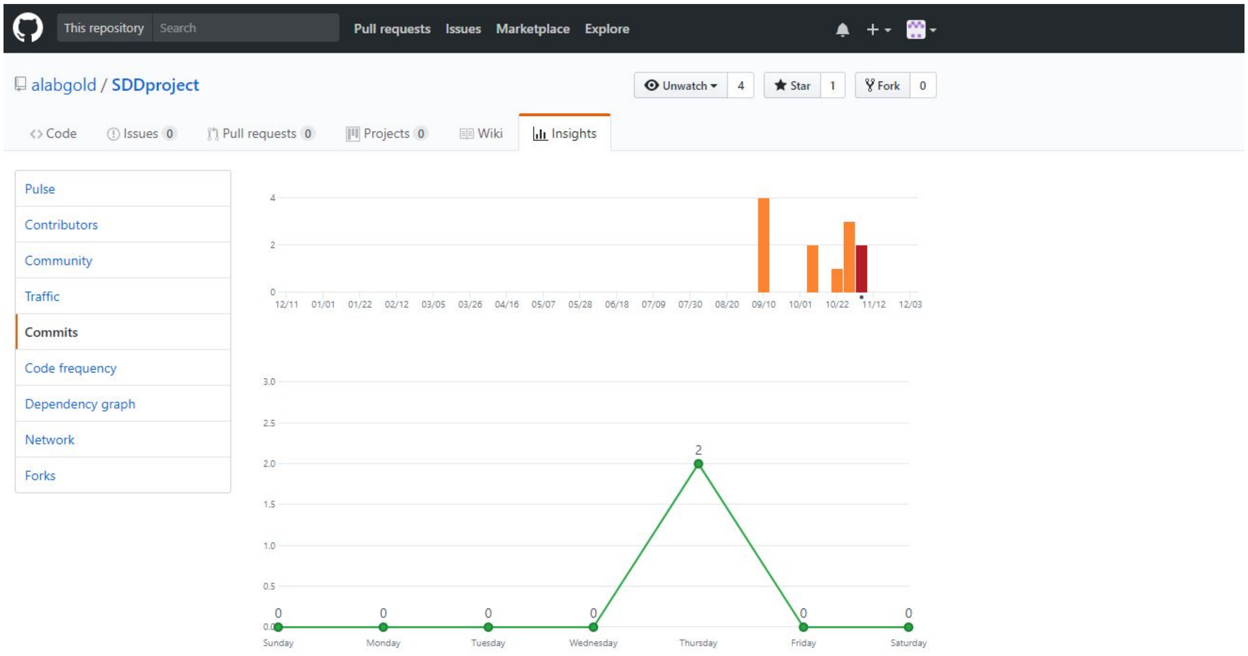
Project Management Website:

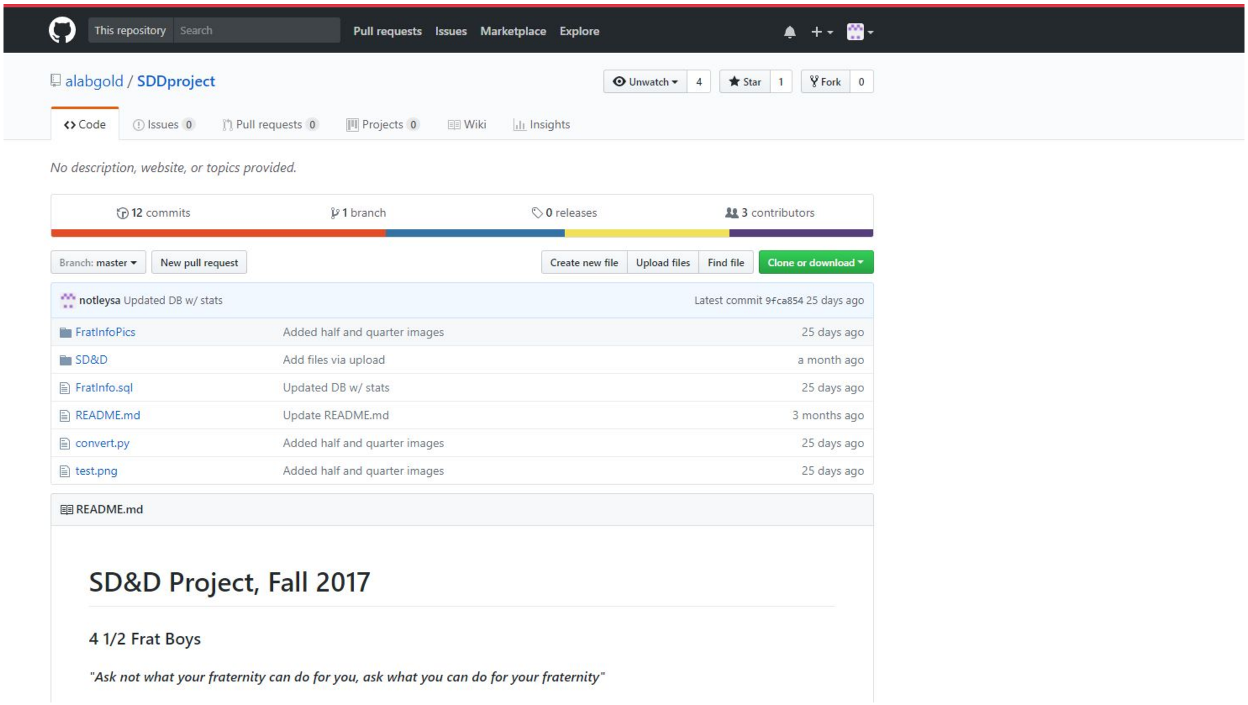
The team used a shared Google Drive in order to manage various aspects of the project. All team documents were stored here and it facilitated easy collaboration on documents, presentations, and various other files to which the team may need to contribute jointly. Everything related to the project was stored on this drive with the exception of the source code, which was managed using a GitHub page as will be explained below. A snapshot from the shared Google Drive can be seen below.



Code Repository:

The team managed the code and version retention for this project using Git, specifically a GitHub page maintained for this project. This was used for version retention in case of file corruption or machine failure as well as replication across all of the machines of the various team members. The images below show some commit statistics and a screenshot of the homepage of the GitHub repository.





No description, website, or topics provided.

12 commits | 1 branch | 0 releases | 3 contributors

notleysa Updated DB w/ stats — Latest commit 9fca854 25 days ago

FratInfoPics	Added half and quarter images	25 days ago
SD&D	Add files via upload	a month ago
FratInfo.sql	Updated DB w/ stats	25 days ago
README.md	Update README.md	3 months ago
convert.py	Added half and quarter images	25 days ago
test.png	Added half and quarter images	25 days ago

SD&D Project, Fall 2017

4 1/2 Frat Boys

"Ask not what your fraternity can do for you, ask what you can do for your fraternity"

Documented Coding Standards:

Apple Style Guide (Swift)*:*

https://developer.apple.com/library/content/documentation/General/Conceptual/DevPedia-CocoaCore/CodingConventions.html

https://developer.apple.com/library/content/documentation/Cocoa/Conceptual/CodingGuidelines/CodingGuidelines.html

JSLint Style Guide (AngularJS):

https://github.com/douglascrockford/JSLint

http://www.jslint.com/

Both style guides were closely followed and enforced by linting systems (more on that later). The only exception here is that for loops and single quotes were tolerated in AngularJS code, though not traditionally allowed in the style guide. This was done for simplicity and due to the database's use of single quotes.

Basic Object-Oriented Design Concepts:

As will be mentioned in the "Design Pattern Use" section, the iOS app employed a Model-View-Controller design pattern, which greatly facilitated many object oriented programming principles by allowing for high cohesion, low coupling, data encapsulation, among others, Polymorphism is greatly leveraged among inherited classes. These object oriented design principles can be seen throughout the code.

CHOSEN BEST PRACTICES

Design Pattern Use:

The app employed a Model-View-Controller design pattern. This design pattern is standard in iOS development in Swift and greatly facilitated by the language. This pattern is a very logical way to represent the information flow throughout our system and its use is very evident in the file structure of this iOS code. The relations here can be seen simply in app file and class names, which are clearly indicative of the function and portion of pattern contained within.

The app made good use of the Singleton pattern to ensure data consistency and simplicity. For example, within the Campus class there is a static, computed variable called "shared," which provides access to the file private shared instance. The design pattern reduced unnecessary and bug-ridden data passing between components of the app, in addition to reducing memory usage and computation time.

The Observer pattern played a large role in the app's functionality, as "didSet" blocks were used in almost every class. In Swift, a didSet block is attached to a variable and called when the variable's value was set, such that if, for example, a Fraternity is added to the list of favorite fraternities, a block is called that saves all favorited fraternities to a file to reflect the change.

Third Party Component or Tool Usage:

SWRevealController (https://github.com/John-Lluch/SWRevealViewController):
SWRevealController is a framework used to simply and efficiently add a pull-out menu from the left hand side of an iOS App. It is very high-speed, making full use of Objective-C's memory management features, and very attractive, as it allows for advanced customization. Quite simply, the framework is used to provide the drawer menu accessed using a swipe from the left, or from the little "burger" buttons provided on the top left of most app screens.

OHMySQL (https://github.com/oleghnidets/OHMySQL):
OHMySQL is a small SQL query wrapping framework. It manages all connections to a remote database, in addition to error checking, sending, and receiving results of queries. Within the RushMe app, the framework is used to connect the device with the database to retrieve fraternity data in text, and images by providing URLs to their locations.

iCalKit (https://github.com/kiliankoe/iCalKit):
iCalKit is used to generate .ics files for export. It provides a simple interface to translate events into the plain text that describes them within the .ics format. The framework

enables the app to export fraternity events to almost any calendar platform, such as Google Calendar.

Google Maps (https://www.google.com/maps)

Google Maps was used in the website platform in an HTML tag that takes a Google Maps url as input for an iframe to display the desired location. This was used in the website in order to embed Google Maps locations in the various fraternity pages.

Ics.js (https://github.com/nwcell/ics.js/)

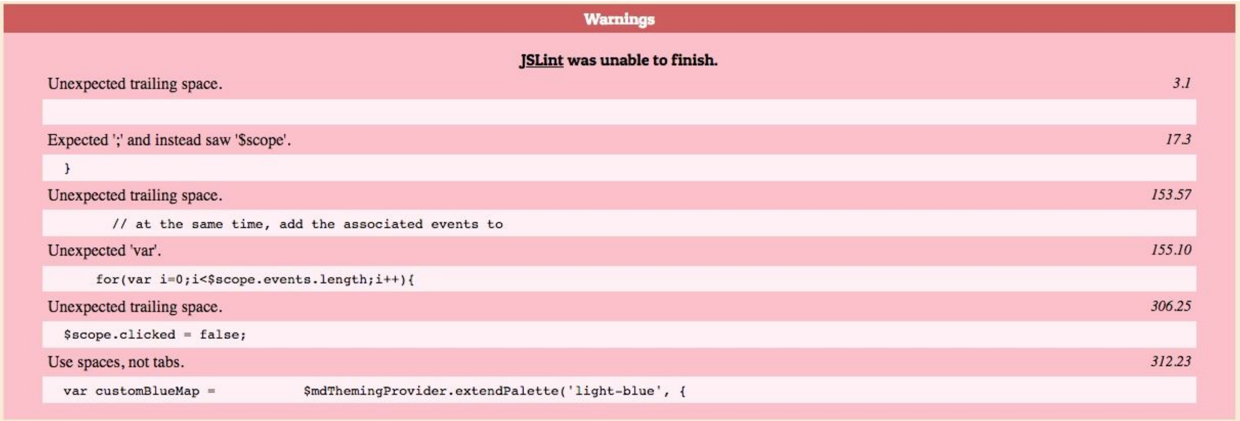
This was used for the ICS file creation on the website. When the download button on the website is pressed, it triggers a function from this package that creates an ICS object and then adds all of the events from favorited fraternities to it.

A Linting System:

JSLint (For Website's JavaScript, http://www.jslint.com/):

For the web platform, the JavaScript linting system JSLint was used in order to ensure we were maintaining compliance with the style guide. Below is the initial linting output showing all errors, followed by the clean linting output after the code was brought up to compliance.

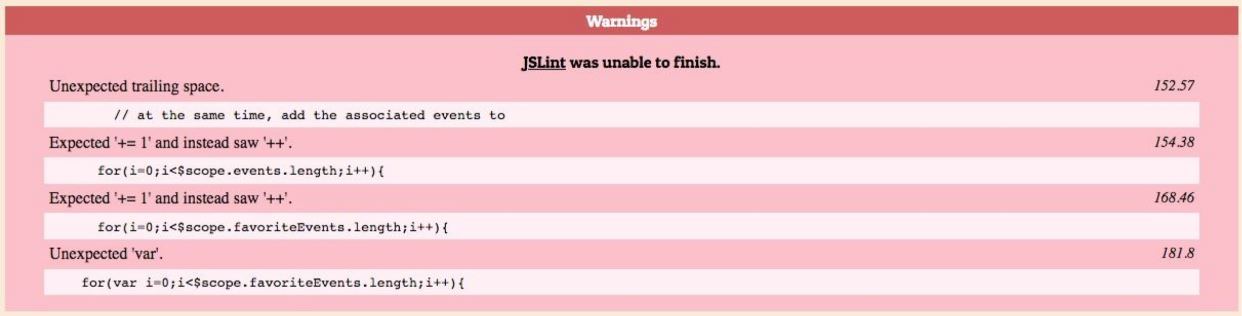
Here are some of the outputs JSLint gave us initially:



```
                                   Warnings

                          JSLint was unable to finish.
Unexpected trailing space.                                                   3.1

Expected ';' and instead saw '$scope'.                                      17.3
  }
Unexpected trailing space.                                                153.57
        // at the same time, add the associated events to
Unexpected 'var'.                                                         155.10
      for(var i=0;i<$scope.events.length;i++){
Unexpected trailing space.                                                306.25
  $scope.clicked = false;
Use spaces, not tabs.                                                     312.23
  var customBlueMap =           $mdThemingProvider.extendPalette('light-blue', {
```



Warnings

JSLint was unable to finish.

Unexpected trailing space. *152.57*

```
        // at the same time, add the associated events to
```

Expected '+= 1' and instead saw '++'. *154.38*

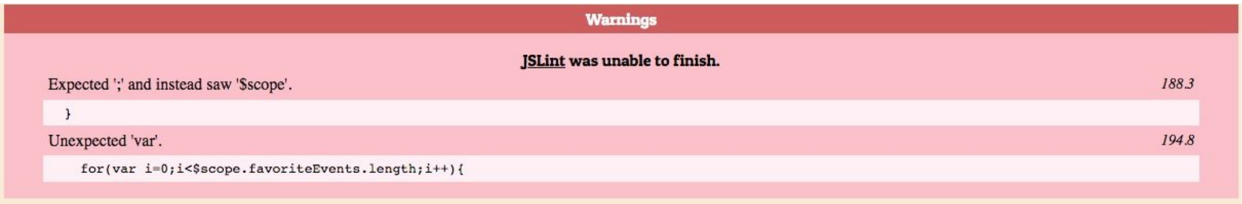
```
      for(i=0;i<$scope.events.length;i++){
```

Expected '+= 1' and instead saw '++'. *168.46*

```
      for(i=0;i<$scope.favoriteEvents.length;i++){
```

Unexpected 'var'. *181.8*

```
    for(var i=0;i<$scope.favoriteEvents.length;i++){
```



Warnings

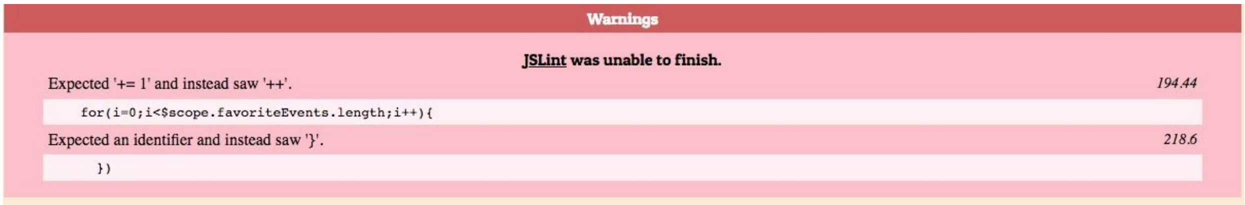
JSLint was unable to finish.

Expected ';' and instead saw '$scope'. *188.3*

```
  }
```

Unexpected 'var'. *194.8*

```
    for(var i=0;i<$scope.favoriteEvents.length;i++){
```



Warnings

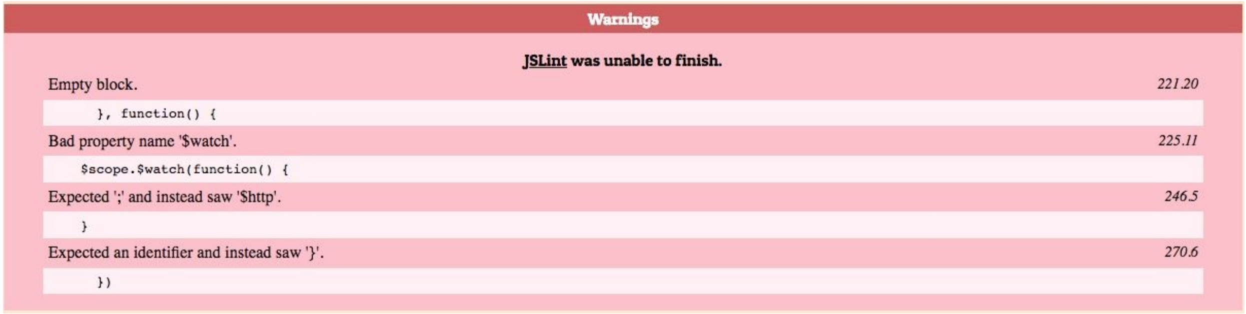
JSLint was unable to finish.

Expected '+= 1' and instead saw '++'. *194.44*

```
    for(i=0;i<$scope.favoriteEvents.length;i++){
```

Expected an identifier and instead saw '}'. *218.6*

```
      })
```



Warnings

JSLint was unable to finish.

Empty block. *221.20*

```
      }, function() {
```

Bad property name '$watch'. *225.11*

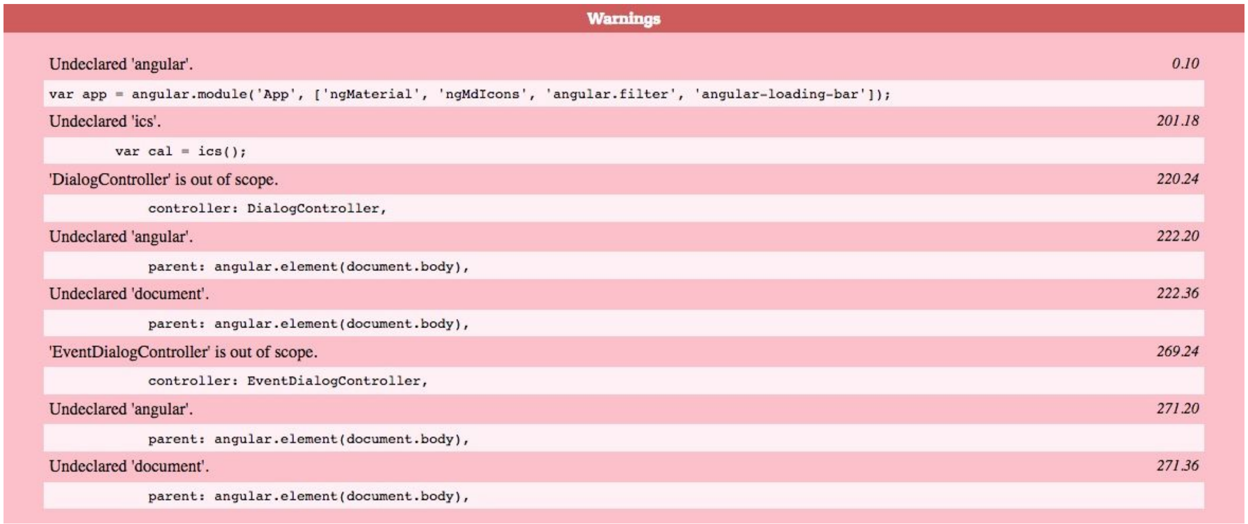
```
    $scope.$watch(function() {
```

Expected ';' and instead saw '$http'. *246.5*

```
    }
```

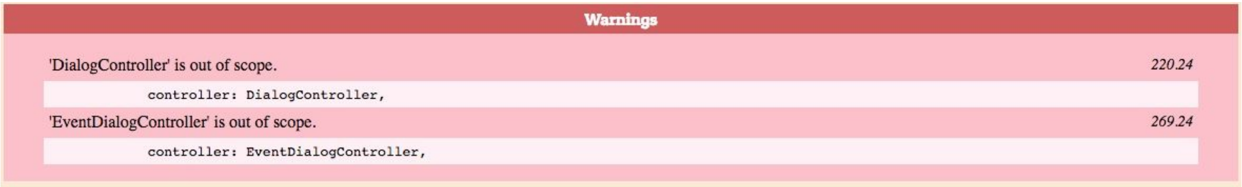
Expected an identifier and instead saw '}'. *270.6*

```
      })
```



```
Warnings

Undeclared 'angular'.                                                                0.10
var app = angular.module('App', ['ngMaterial', 'ngMdIcons', 'angular.filter', 'angular-loading-bar']);
Undeclared 'ics'.                                                                  201.18
        var cal = ics();
'DialogController' is out of scope.                                                220.24
            controller: DialogController,
Undeclared 'angular'.                                                              222.20
            parent: angular.element(document.body),
Undeclared 'document'.                                                             222.36
            parent: angular.element(document.body),
'EventDialogController' is out of scope.                                           269.24
            controller: EventDialogController,
Undeclared 'angular'.                                                              271.20
            parent: angular.element(document.body),
Undeclared 'document'.                                                             271.36
            parent: angular.element(document.body),
```



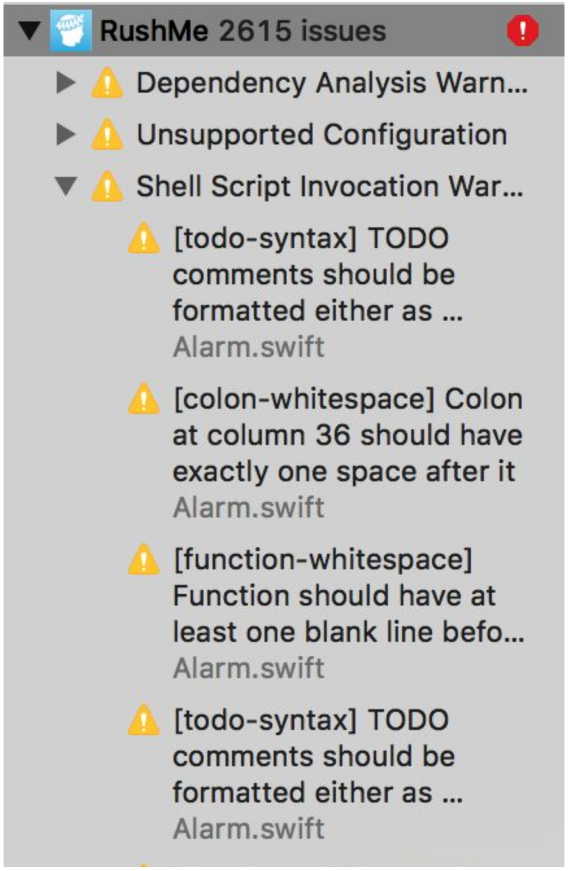
```
Warnings

'DialogController' is out of scope.                                                220.24
            controller: DialogController,
'EventDialogController' is out of scope.                                           269.24
            controller: EventDialogController,
```

The next page is the JSLint summary with no errors or warnings.

Tailor (For iOS App's Swift, https://tailor.sh/):

For the app code, the team used Tailor to ensure that the code developed adhered with the Apple Style Guide for Swift. The Tailor enforcement has some slight variation on the Apple Style Guide, which is very extensive. This version is followed entirely, and the proof of this is shown below in the form of the Tailor warnings before and after resolving them. Note that there are still a significant number of style warnings on Tailor, but this is due to the external files and third-party code used, and none of the remaining errors are in any code developed by the group.



RushMe 2615 issues

Dependency Analysis Warn...

Unsupported Configuration

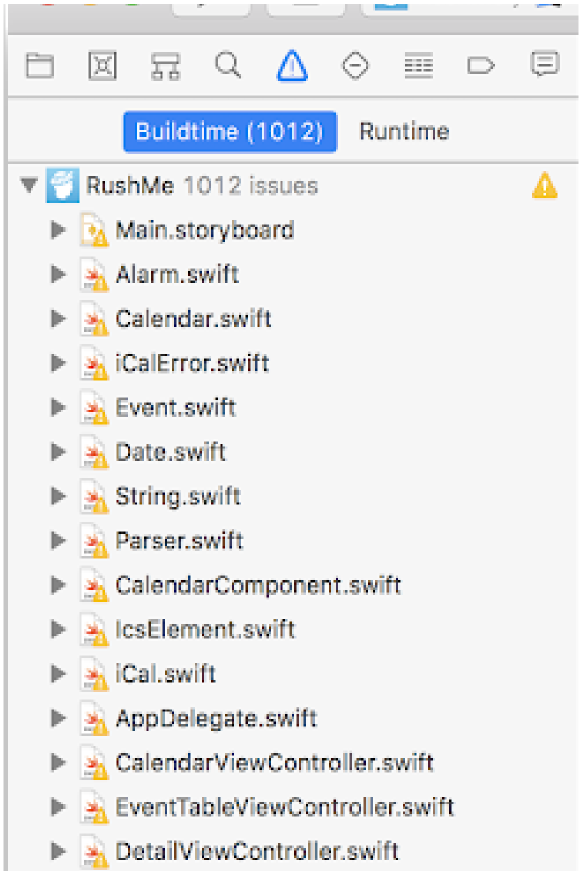
Shell Script Invocation War...

[todo-syntax] TODO comments should be formatted either as ...
Alarm.swift

[colon-whitespace] Colon at column 36 should have exactly one space after it
Alarm.swift

[function-whitespace] Function should have at least one blank line befo...
Alarm.swift

[todo-syntax] TODO comments should be formatted either as ...
Alarm.swift



RushMe 1012 issues

- Main.storyboard
- Alarm.swift
- Calendar.swift
- iCalError.swift
- Event.swift
- Date.swift
- String.swift
- Parser.swift
- CalendarComponent.swift
- IcsElement.swift
- iCal.swift
- AppDelegate.swift
- CalendarViewController.swift
- EventTableViewController.swift
- DetailViewController.swift

CONTRIBUTION SUMMARY

Ben Crevier

- Helped With Testing
- Wrote Status Report

Jim Hines

- Worked with users outside the group to complete testing
- Linted app code

Adam Kuniholm

- Tested and refined App code by adding Beta testers and trying corner cases manually
- Worked with users outside the group to complete testing
- Prepared and choreographed App demo

Andie Labgold

- Created user testing scenarios
- Completed user testing for website
- Linted js code
- Prepared and choreographed Web demo

Stephen Notley

- Compiled testing results
- Documented best practices

STATUS REPORT

DONE

Testing

- We completed usability testing. We had 5 people test the application, and 5 people test the website. We received feedback that there are a few UI things that could be changed to make the app slightly easier to use. We recommended that the product is ready for release

Linting

- We used linting software to make sure that the code for our application and website adheres to our coding standards.

Final Presentation

- We presented our final product to the class, and showed a demo of the app and website.