## 

## **Deliverable 3**

## **Novus**

## **November 28, 2017**

## 

## 

## 

## Instructor: Hyunsook,Do Contributors: Daniel Jimenez, Tsung-han Hsieh, Brandi Werner, Kaitlyn Lafleur

## 

## 

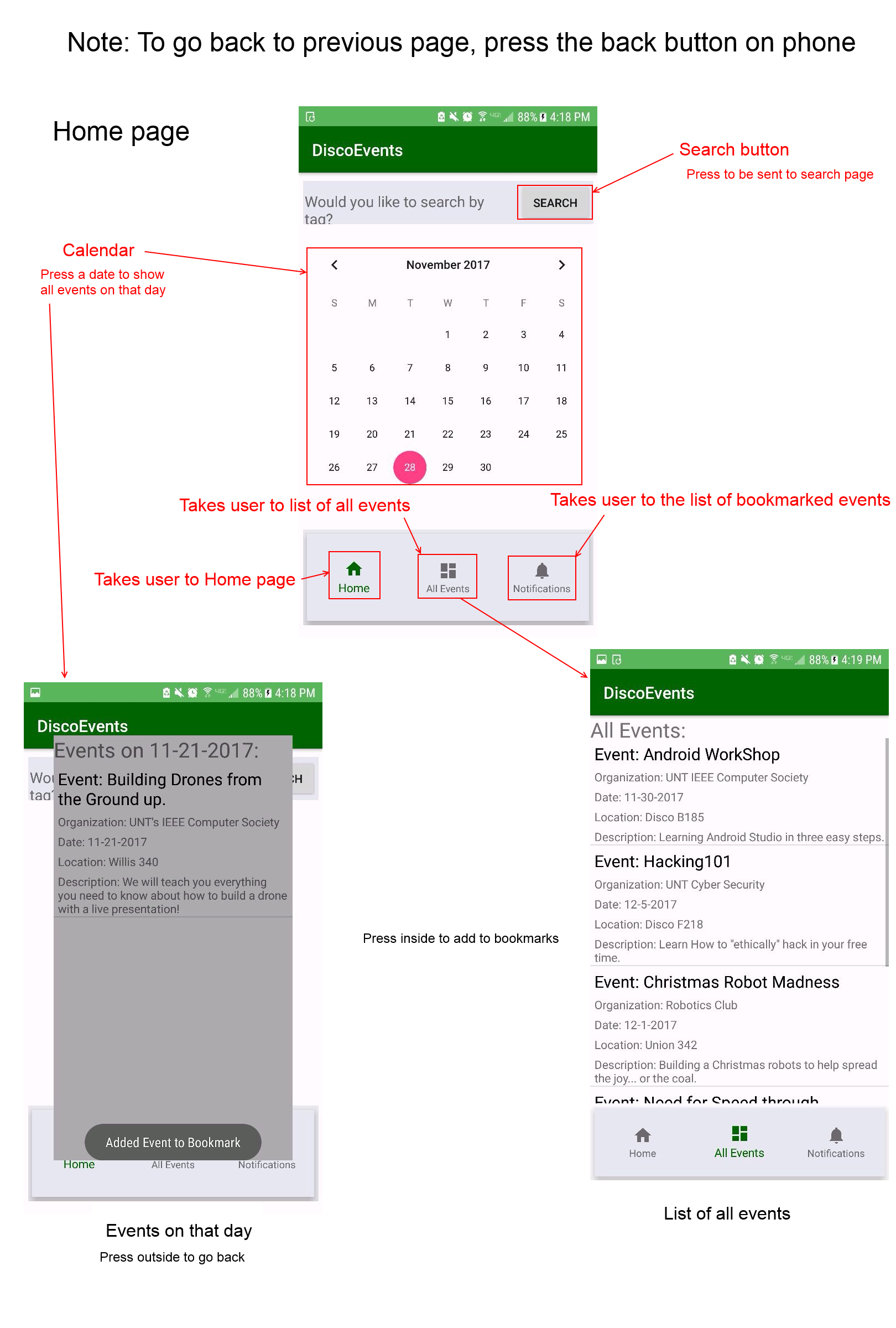
## Table of Contents

1. Manuals ………………………………………………….3   
    - DiscoEvents Android App Manual………………………………3  
    - DiscoEvents Website Manual……………………………………6
2. How to compile and run the programs and test   
   suites……………………………………………………..9  
    - DiscoEvents Website………………………………………….....9  
    - DiscoEvents Android App………………………………………..9  
    - How To Run Test Suites……………………………………..…13
3. Automated tests and experiences with testing……..14  
    - Testing for the Website Portion………………………………..14  
    - Testing for the Android Portion………………………………...15
4. Accomplishments and limitations…………………….16  
    - Accomplishments………………………………………………..16  
    - Limitations………………………………………………………..16  
    - Future Development…………………………………………….16
5. Meeting minutes……………………………………….17
6. Contribution table……………………………………...19

## 1 | Manuals

## 1.1 | DiscoEvents Android App Manual

DiscoEvents is a go-to app for students to use as a reference for any events going in Discovery Park. The homepage for the app is a calendar with a search option above it and a navigation bar below it.

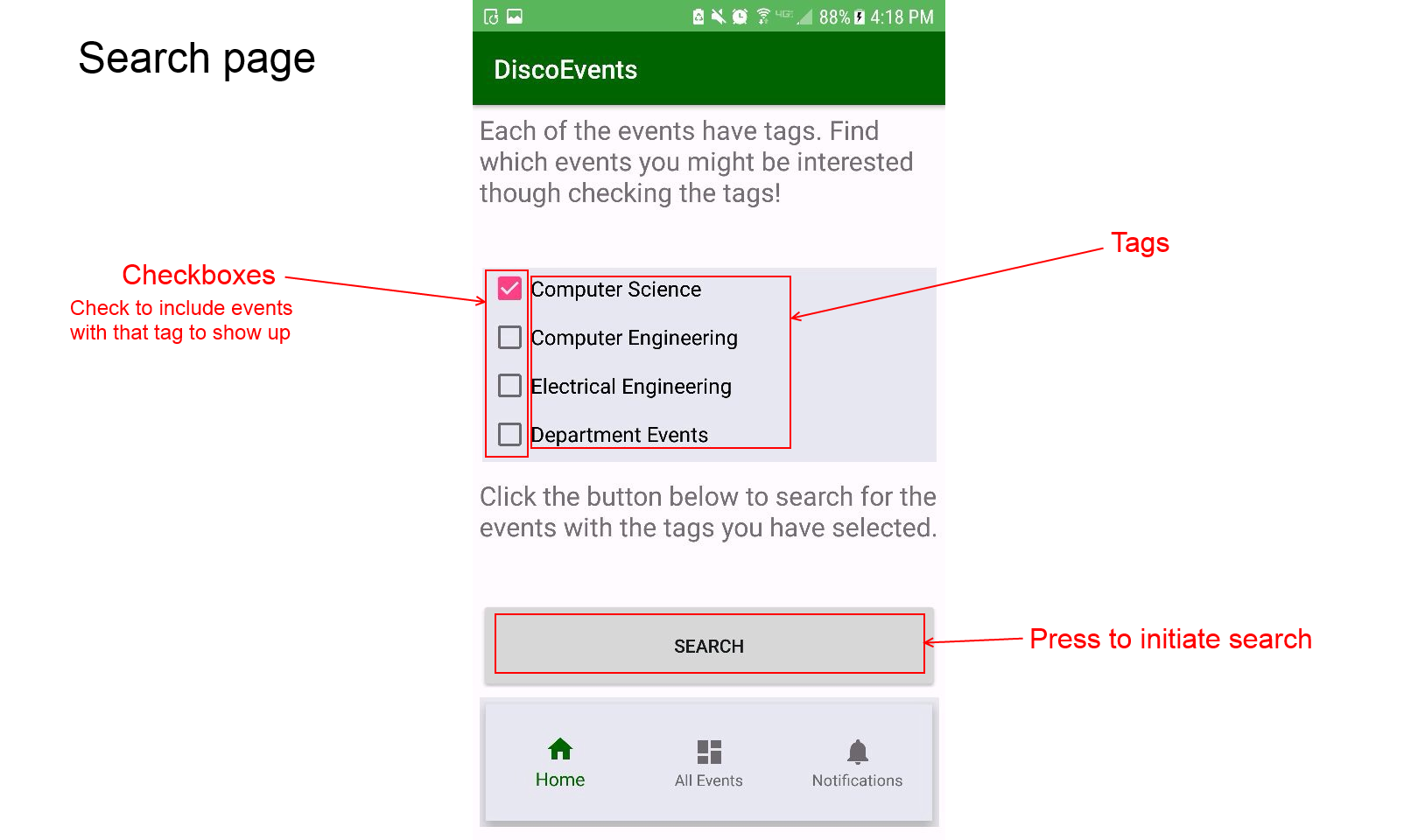


The calendar is interactive and swipeable. To display an event that is occurring on any given day, tap on that day in the calendar. The app will create a pop up displaying the relevant event information.

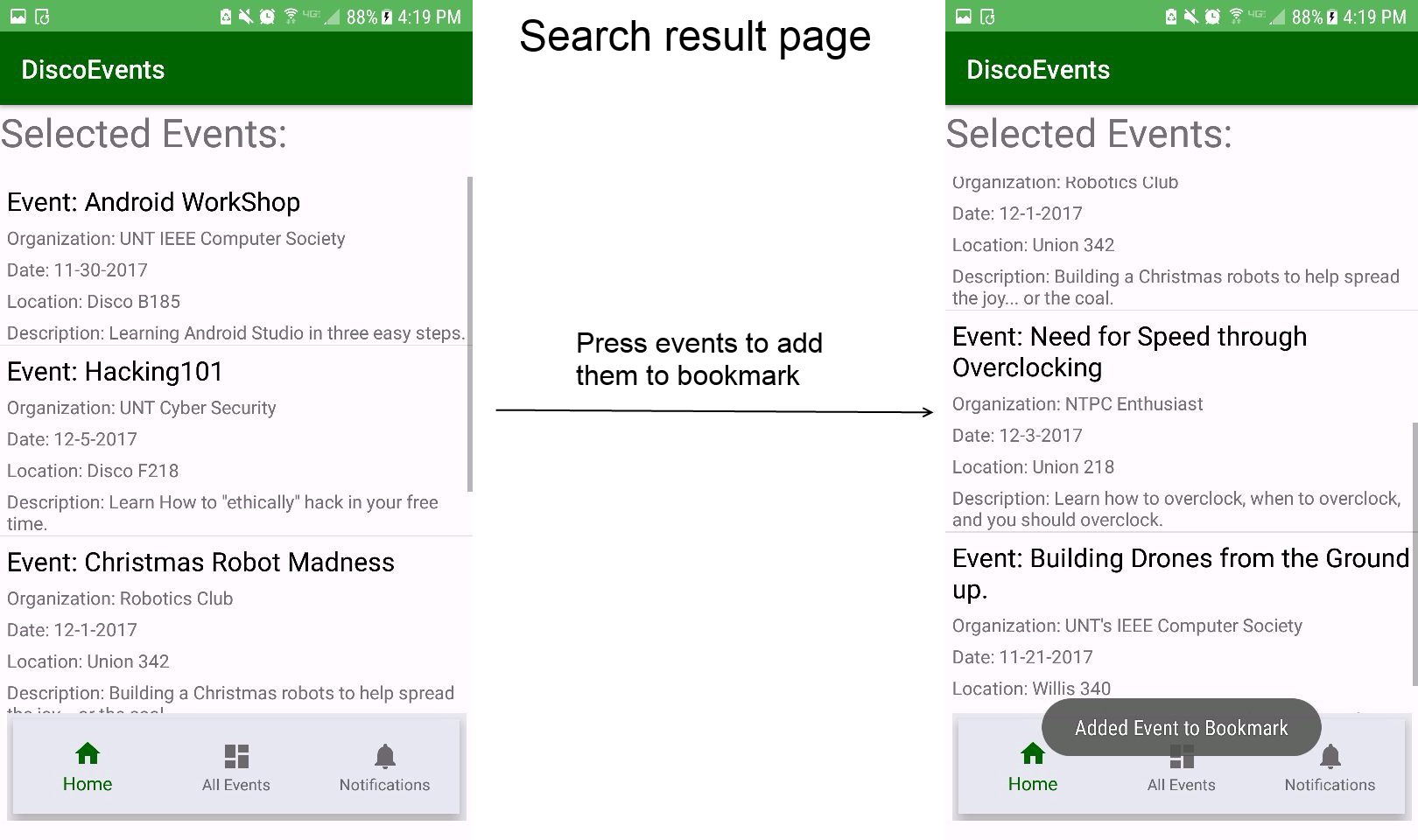
The All Events navbar is invoked by tapping on the all events button. It will then display a page that lists all of the events scheduled in discovery park.

The bookmarks navbar is invoked by tapping on the bookmarks button. It will then display a page that lists all of the user's individual bookmarked events for future reference.

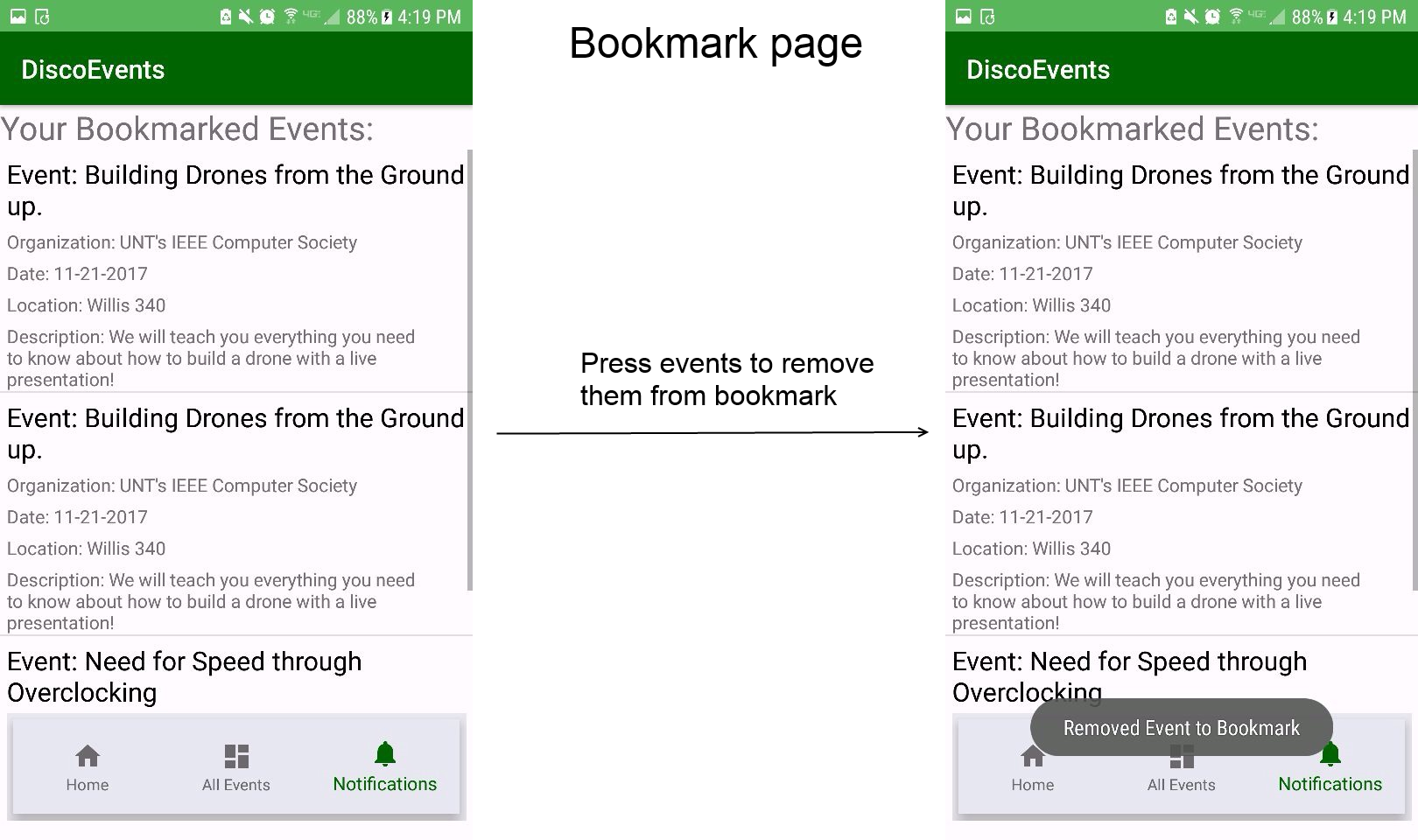
The search option is invoked by tapping on the search button. It will be further explained below.



The image above displays the search page that is displayed when the user taps the search button in the home page. To use the search feature, tap on the desired event tags to activate the checkboxes. Once all of the desired tags have been checked, tap the search button. The search button will pull up a page that displays the relevant events that match with the desired tags.

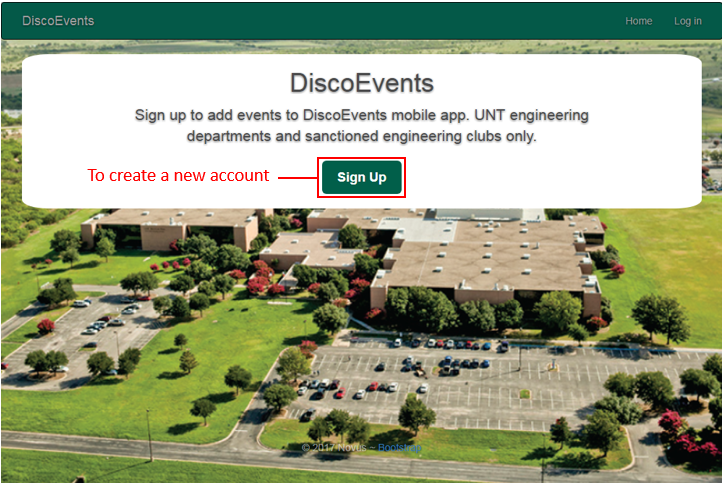


Within the search results page, the user can scroll through the events that were searched, or choose to bookmark an event by tapping the bookmark button, which will save the event for future reference.

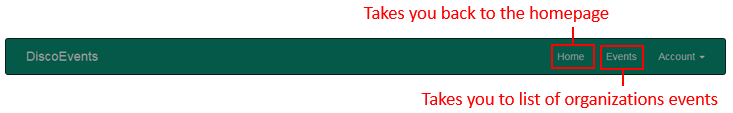


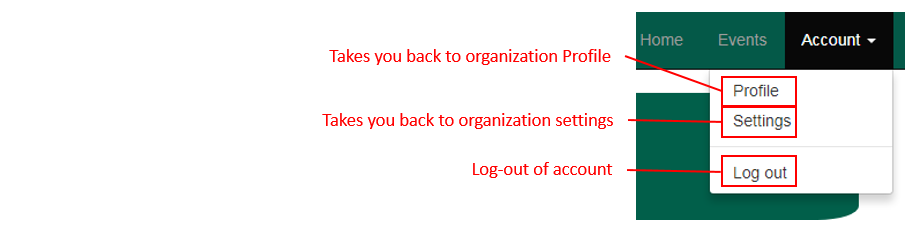
## 1.2 | DiscoEvents Website Manual

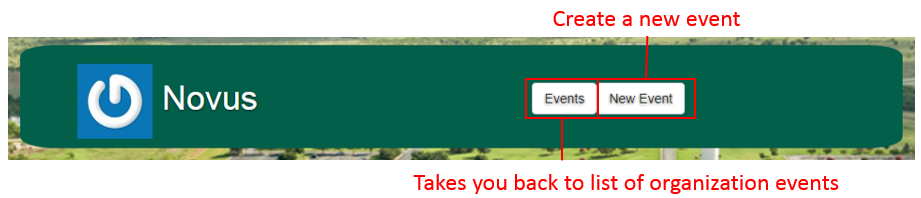
The DiscoEvents website is primarily for Organizations to create events for the database, and the events are then displayed on the DiscoEvents Android App.



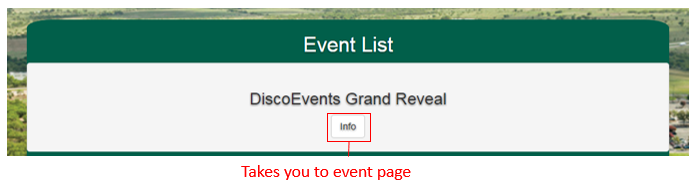




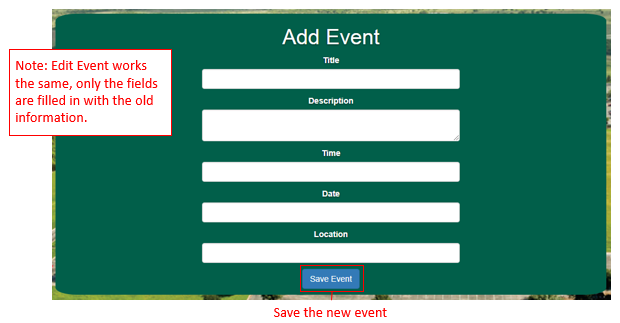


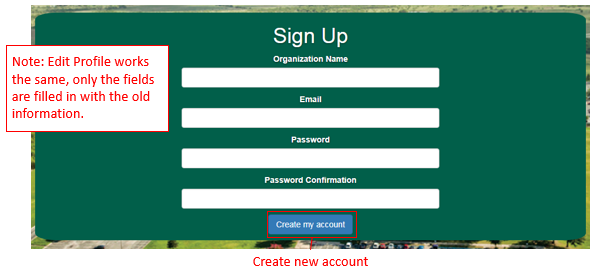


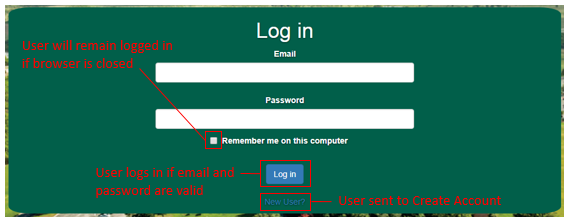
### User Profile











## 2 | How to compile and run the programs and test suites

2.1 | DiscoEvents Website

How to run program

1. Download, install, and setup Ruby and Rails on your computer.
2. Run cmd (or other command line, I use git bash) from root file of program {/site\_code}
3. In cmd line: >>rails server
4. This should start the puma web server. This window is now only running the web server.
5. In a web browser of your choice go to localhost:3001/home
6. You can now use the site

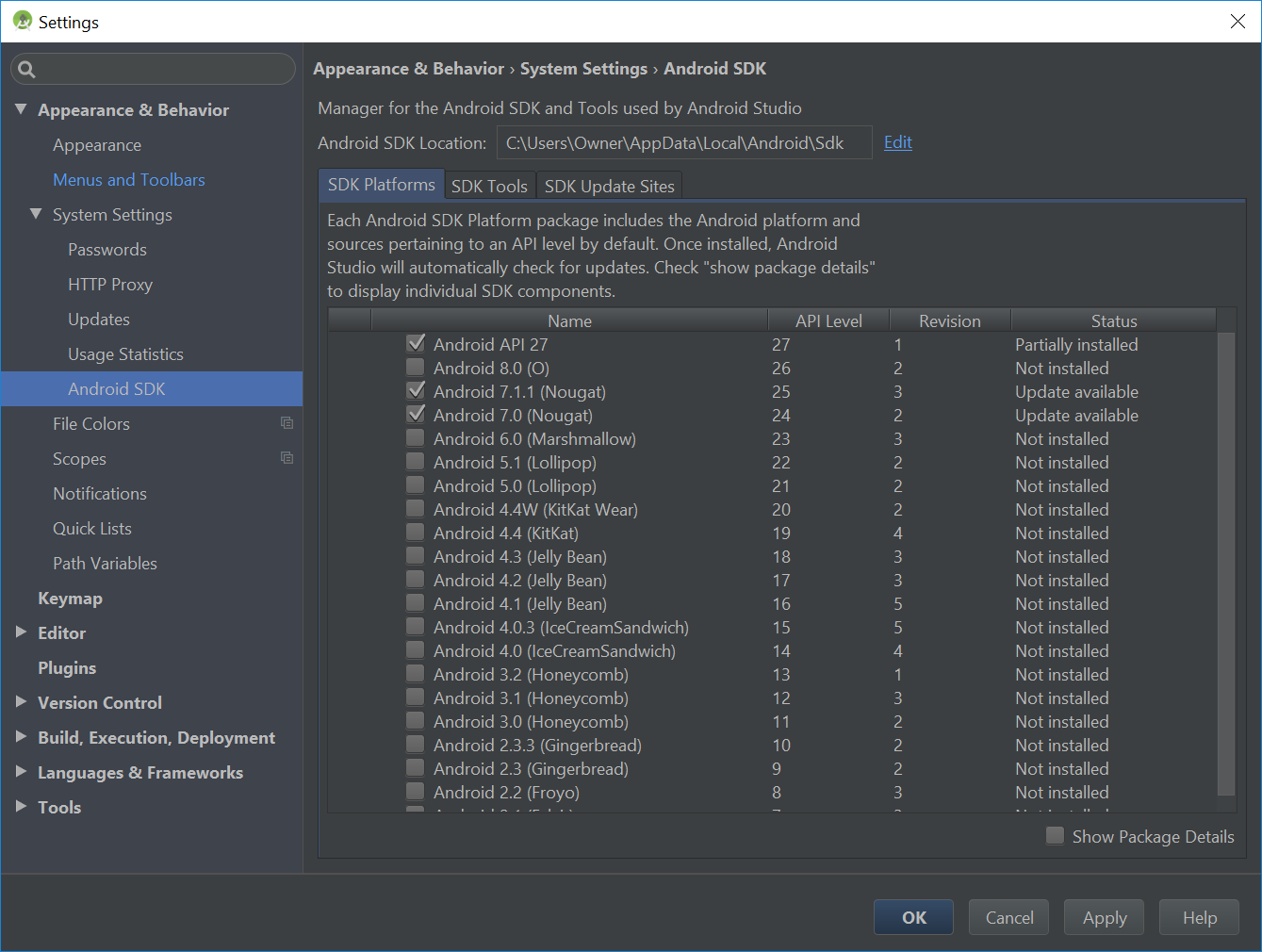
How to run test suite

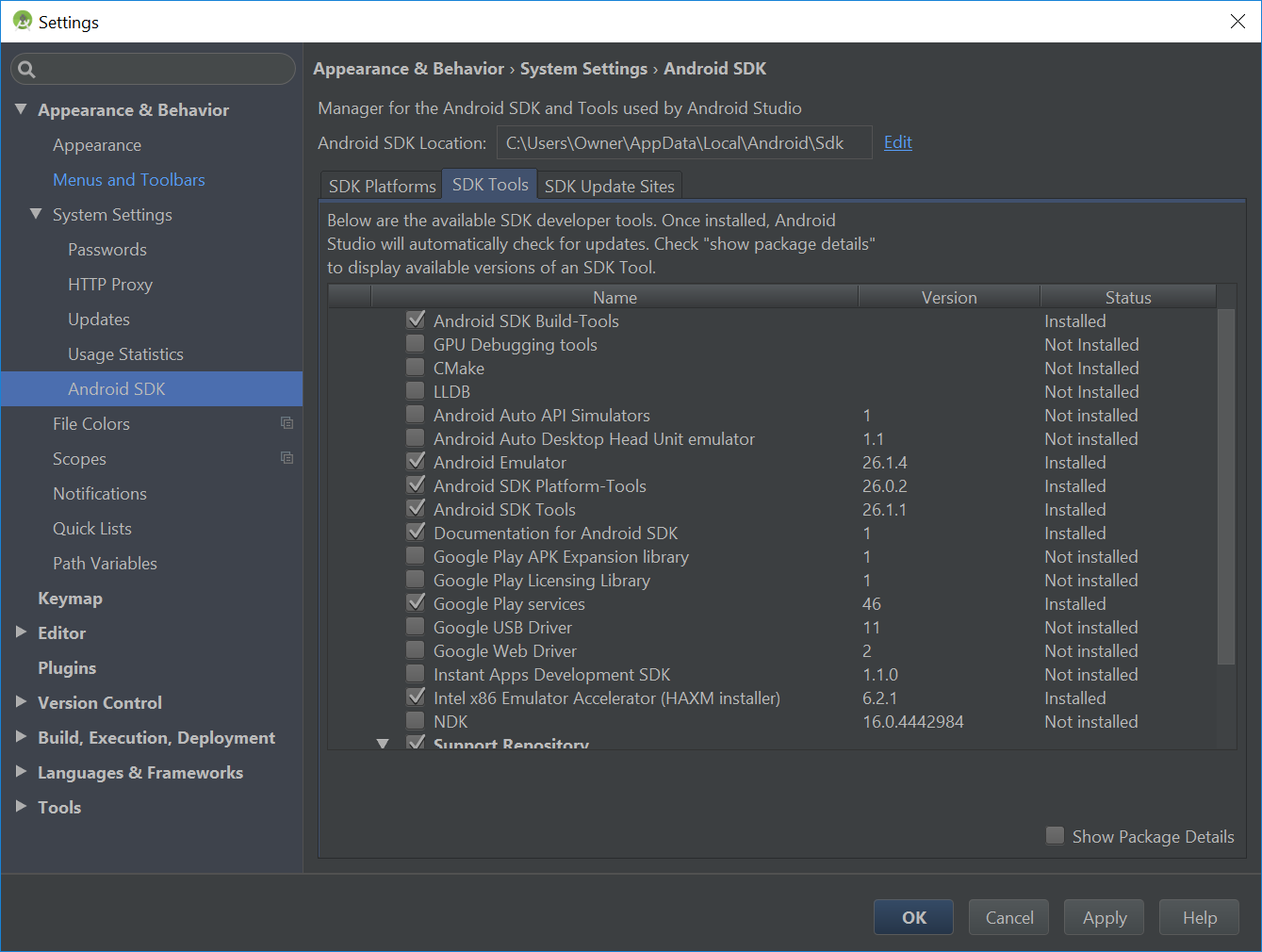
1. Follow steps 1-4 from above
2. Open a separate cmd window (also running from the root file {/site\_code})
3. In cmd line: >>rails db:migrate RAILS\_ENV=test
4. In cmd line: >>rails test
   1. This should run all tests created in the test folders.
   2. If you want to run individual tests in cmd line: >> rails test test/{path to test file}

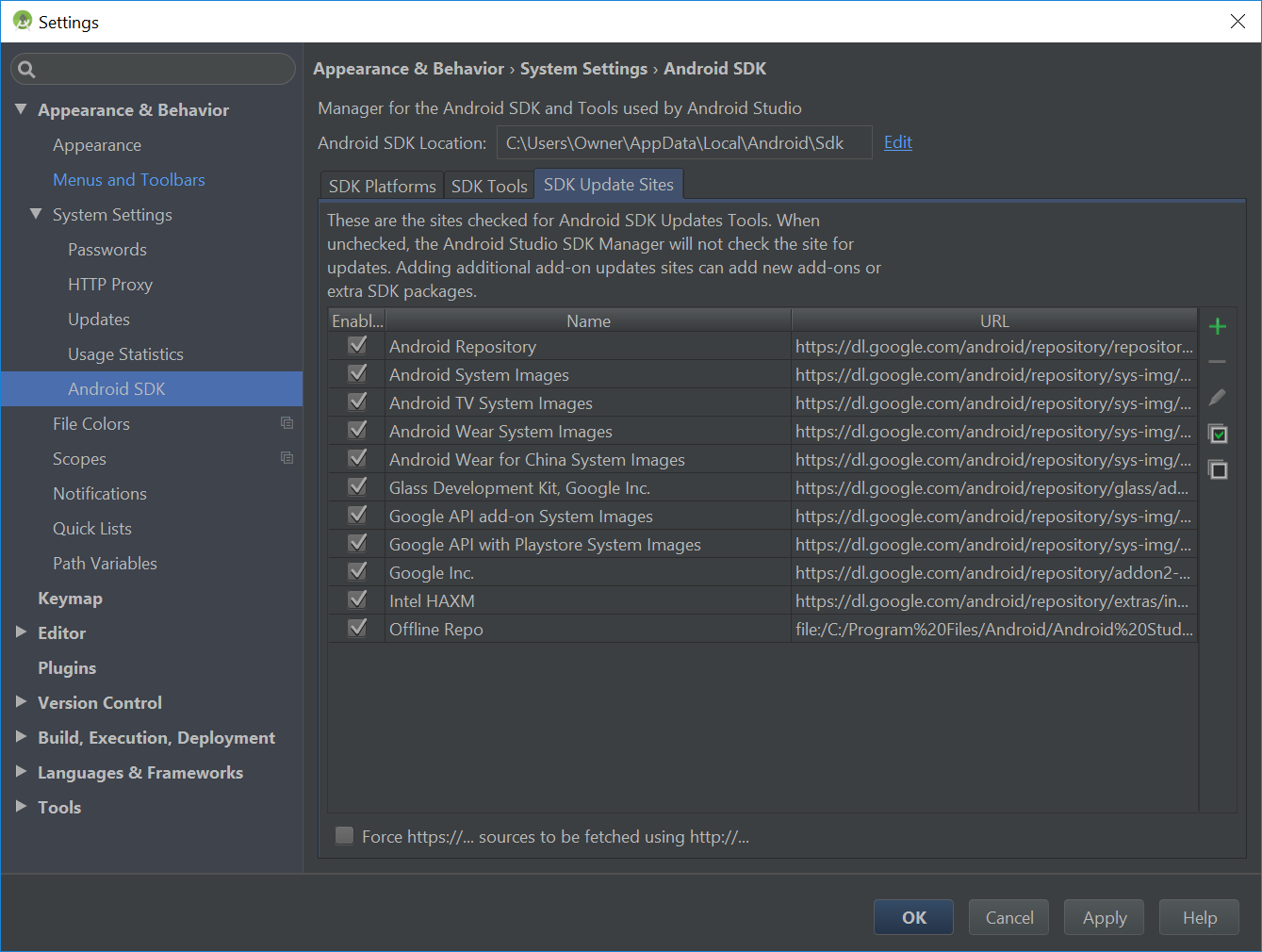
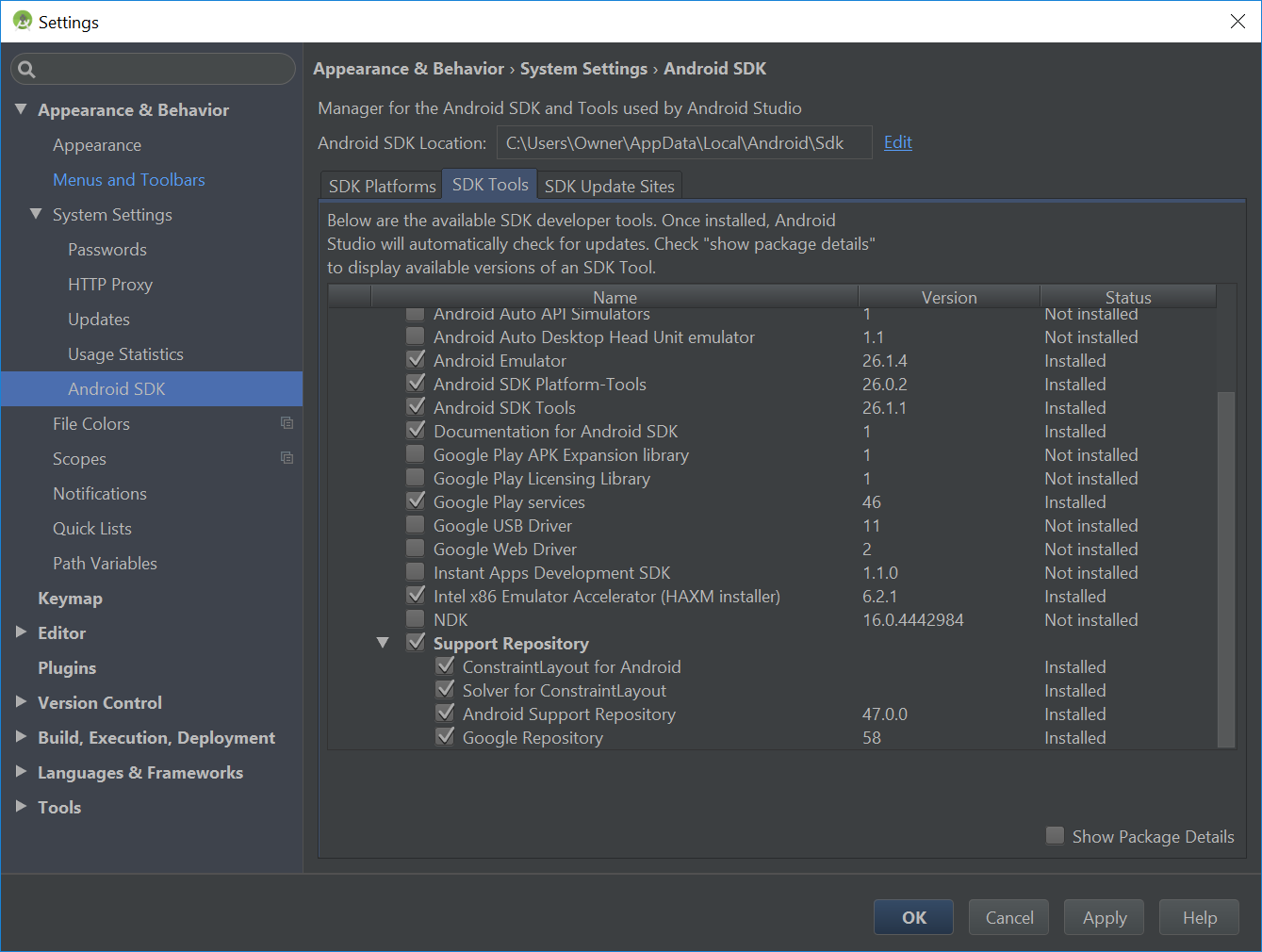
2.2 | DiscoEvents Android App

How to run program

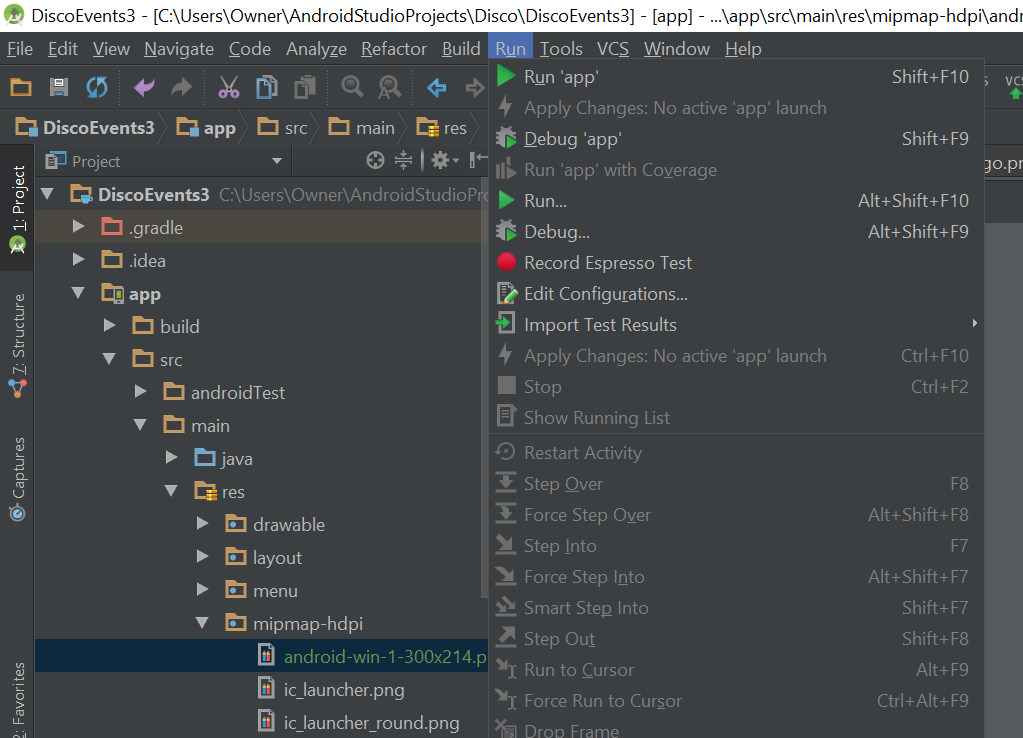
1. Download and install Android and Java
2. Go to settings and make sure you have … downloaded and up to date:



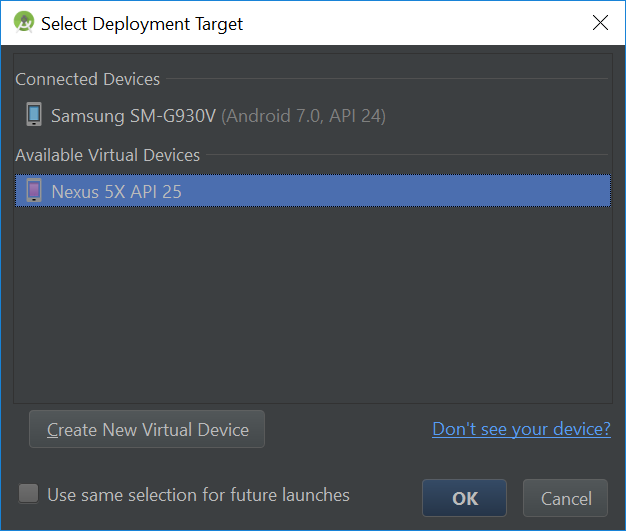




1. Click on the run tab on the top
   1. The first item in the run tab should be “Run ‘app’”, select that.



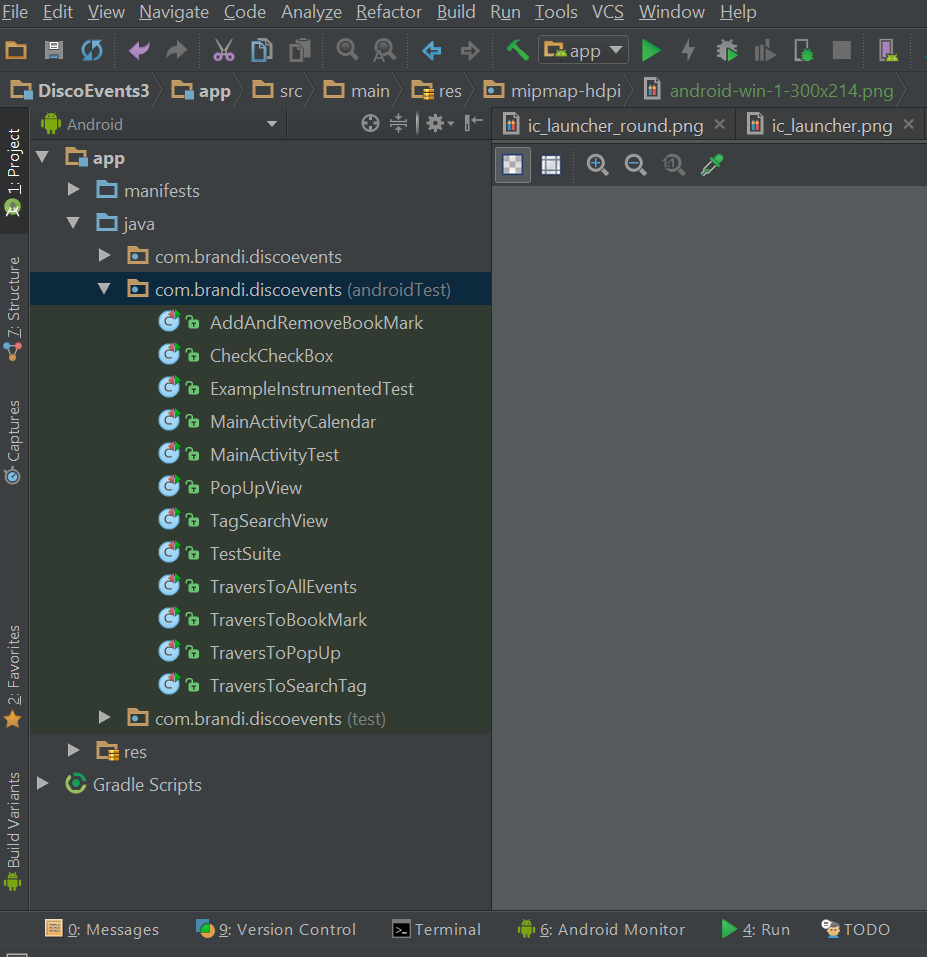
1. It should pull up a window that allows you select which device you would like to run it on.
   1. You may run it on your phone or on a virtual machine that you have created.



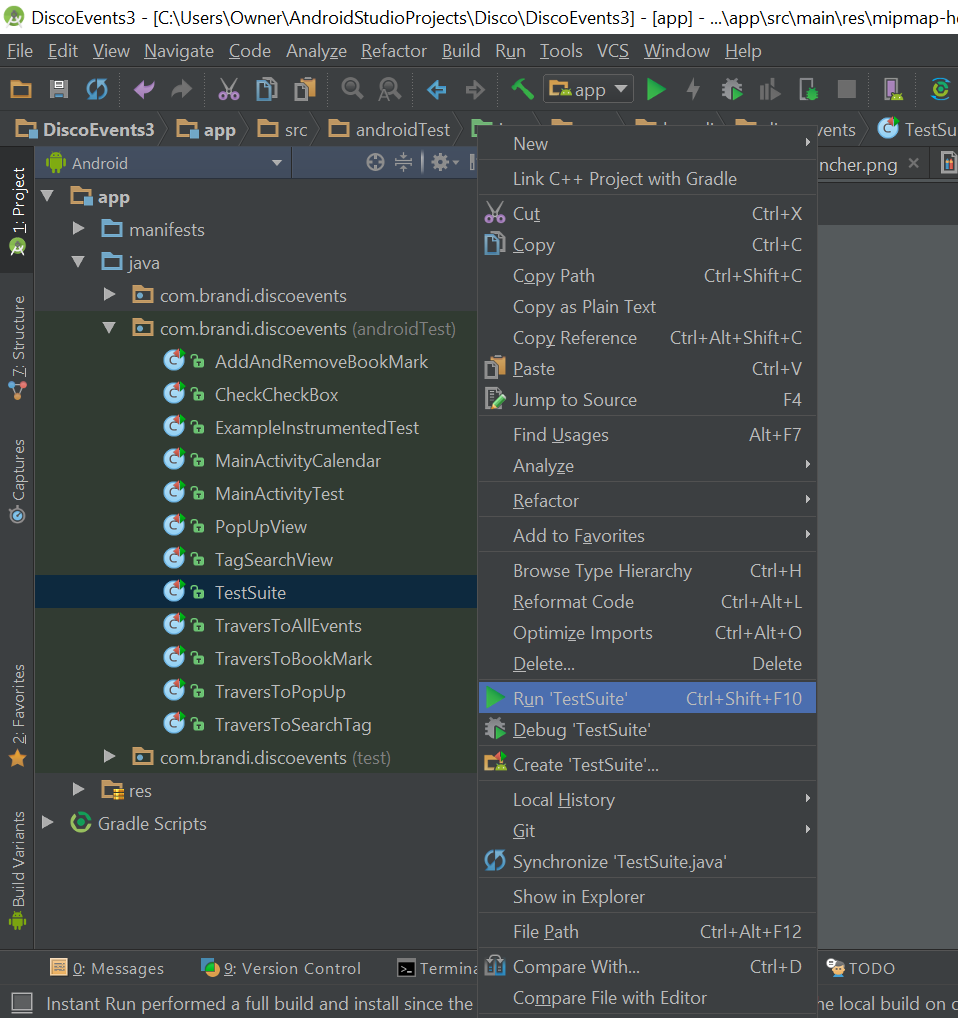
1. Once you have selected the device, press ok and the app will run

2.3 | How To Run Test Suite

1. Travers to the android test folder which holds the test classes



1. Right click on the class called “TestSuite” and click on “Run ‘TestSuite’”



1. Now you are running the test suite.

## 3 | Automated tests and experiences with testing

3.1 | Testing for the Website Portion

For the website portion, several forms of testing were used. Unit testing was not implemented until the users and sessions were being set up. This is because the web programmer was not sure how to implement unit testing, but found a resources that helped write the unit tests. The unit tests did not particularly feel useful, as the errors they showed had to do with errors in the tests. I mean this could be a good thing, maybe the site was written so well that issues weren’t found but they didn’t feel very useful. Functional Testing ended up being very useful, and ended up being very important for the forms. Users are very likely to input data you don’t want, or click buttons they don’t need to, so in using functional testing we were able to account for possible incorrect inputs.

Here at are the names of the Website Automated Test cases and a short description of each:

Controller Tests

Events Controller Test

* "should get new"

Sessions Controller Test

* "should get new"

Static\_Pages Controller Test

* “should get home”

Users Controller Test(s)

* "should redirect edit when logged in as wrong user"
* "should redirect update when logged in as wrong user"

Helper Tests

Sessions Helper Test(s)

* "current\_user returns right user when session is nil"
* "current\_user returns nil when remember digest is wrong"

Integration Tests

Users Edit Test(s)

* "unsuccessful edit"
* "successful edit"
* "successful edit with friendly forwarding"

Users Login Test(s)

* "login with valid information followed by logout"
* "login with remembering"
* "login without remembering"

Users Signup Test(s)

* "invalid signup information"
* "valid signup information"

Models Test

User Test(s)

* "should be valid"
* "organization should be present"
* "email should be present"
* "organization should not be too long"
* "email should not be too long"
* "email validation should accept valid addresses"
* "email validation should reject invalid addresses"
* "email addresses should be unique"
* "email addresses should be saved as lower-case"
* "password should be present (nonblank)"
* "password should have a minimum length"

3.2 | Testing for the Android Portion

Automated testing with android studio was simple. The reason for this is because android has a application called espresso that allows you to run through your app and created different assertions through screenshotting your app and allowing to select and define what you would like to test. After you have finished running your app with espresso, espresso creates the test file through what you did while running the app. The next step is running the test and seeing if you get any errors.

Here are some of the many test cases we created for our android app and a short description for them:

MainActivityTest.java:

A test that make sure the main activity has all of the correct content displayed on it.

[TagSearchView.java](https://github.com/DanielJimenez97/DiscoEvents.final/blob/master/app/src/androidTest/java/com/brandi/discoevents/TagSearchView.java):

Traverses to tag search view and makes sure all of the content on the view is correct.

[AddAndRemoveBookMark.java](https://github.com/DanielJimenez97/DiscoEvents.final/blob/master/app/src/androidTest/java/com/brandi/discoevents/AddAndRemoveBookMark.java):

Travers to pop up and add an event to the bookmarks. Traverse to the bookmarks and delete an event.

[CheckCheckBox.java](https://github.com/DanielJimenez97/DiscoEvents.final/blob/master/app/src/androidTest/java/com/brandi/discoevents/CheckCheckBox.java):

Travers to the Search view and check the checkboxes correctly.

## 4 | Accomplishments and limitations

4.1 | Accomplishments

Many of the feature that were specified in deliverable two for our app have been accomplished. The features that we have successfully implemented for our website are:

* creating a log-in/create account page
* having the login passwords encrypted through a hash table for security
* allowing the users to create a new event through a form which will automatically save in the database
* Allowing the user to look at the events they have created
* Allowing the user to delete any event they have created
* Creating a home view
* Creating a profile view
* Creating a drop down menu for navigation options
* Allowing the user to stay logged in when they exit the app and do not log out

Along with accomplishing many of our task for the website, we also accomplished a majority of the tasks we were assigned for our Android application which includes:

* Creating a calendar that the users can select a date and view the events on that date.
* Allowing the users to search for an event through tags
* Connecting our app to a database
* Querying from the database
* Creating a navigation bar across the whole app.

4.2 | Limitations

Despite accomplishing a majority of the task our team set out to face, we did run into some limitations. One of the main limitations we had was connecting our website and android app to the same database. The reason for this is because both applications have their standard databases they already like to write data into and changing the database of either application means that you would have to change the querying structure for the whole application.

4.3 | Future development

We have many plans for version 2 of our application. The improvements are as follows:

Next Phase for DiscoPark Website

* Adding picture for events
* Perhaps a pop up calender or calendar field button when setting event date
* Showing events on Profile page instead of a separate page
* Adjusting path, so the “home” page will be the profile page when a user is logged in, but when no one is logged in it will show the “Welcome to DiscoEvents” page with the Sign up button.
* Connect the Android app to the website, or migrate the functionalities of the android to the website.
* Further refine the appearance of the site

Next Phase for DiscoPark Android App

* Adding pictures for events
* Have the calendar indicate visually which dates have events on them
* Allowing our list views to recycle the rows that are not in use. This will allow the device to use a lot less memory when creating our list because it will not have to save all of the rows.
* Creating more tags for the user to choose from and possibly putting these tags into a list if there is a substantial amount of them.
* Further refine the appearance of the app

## 5 | Meeting Minutes

10/19/17 Meeting

Location: class

Present:   
Brandi Werner, Kaytlin Lafleur, Daniel Jimenez, Tsung-Han Hsieh

Agenda:   
Deliverable 3  
Final Project Breakdown  
Potential Database change - contingency implementation

Deliverable 3 - much in anticipation for the third deliverable, we roughly split the parts that we wanted to each do individually. Daniel and Tsung-Han have split the manuals, Brandi and Kaytlin will split the explanation of features and test suites. Daniel will also do meeting minutes. We have set goals as a group to meet the specified code amount for the first section of the deliverable 1.

Final project breakdown - for the final project, we made sure to reiterate where each team member was working. Daniel and Brandi on Android, Tsung-Han and Kaytlin will work on the database section

Potential Database change: Until now, the database group has tried to build a database using Django web framework. Until now, it has proven to be unsuccessful and too difficult to understand. After individual group research, we have all considered making a potential switch to ruby on rails to handle the database section, as it is more accessible to us than Django.

11/9/17 Meeting

Location: class

Present:   
Brandi Werner, Kaytlin Lafleur, Daniel Jimenez, Tsung-Han Hsieh

Agenda:   
Deliverable 3 part 1 progress  
Final Project progress  
Schedule analysis

Deliverable 3 part 1 progress - we discussed how much code we had each individually written. The code was glanced over to make sure that they were following the code specification example given in the class resources. Until now we are on track to have first section of the deliverable 3 complete. MAY NEED TO USE SLACK DAY

Final Project Progress: The android group discussed their recent hangups in implementing view changes along with the UI they desired to implement. Regardless of those issues, a calendar with a pop up has been created that is prepared to display events. The connectedness of the database to the android device is still under programming. Until now, ruby on rails group has proven to be more successful using this rather than Django. Working database and blog style post creations have been programmed.

Schedule analysis - we have analyzed our progress in comparison with the schedule for the project. Until now, we don't foresee any issues with the pace we are working, but anticipate we may, and should use a slack day in the future to assure quality work.

11/14/17 Meeting

Location: class

Present:   
Kaytlin Lafleur, Daniel Jimenez, Tsung-Han Hsieh

Agenda:   
Code Inspection

Code Inspection - our code was inspected by another group to assure that it meets good coding practices. The feedback we received from the group was generally positive. They did note that it would be useful if we improved the quantity and quality of our comments to allow it to be easier to follow. They also warned us of some of our switch statement that fail to have a default statement. Besides these issues, they believe we have written our code positively and in good practice.

11/21/17 Meeting

Location: class (although class cancelled)

Present:   
Brandi Werner, Kaytlin Lafleur, Daniel Jimenez, Tsung-Han Hsieh

Agenda:   
Final Project

Final project - as we near the final due date for the project, we discussed the project and what is left. Kaytlin has almost completely finished the form section of the website and the database, and is now left with the section on creating secure logins. Tsung-Han has switched group and is now in android dev. Android dev has created a calendar with pop ups on day clicks to show individual calendar data. Also, a search by tags function was created, alongside a bookmarks page and an all events display page. Until now, some functionality of the aforementioned app is not complete, however it is near the final stages.

10/24/17 Meeting

Location: phone

Present:   
Brandi Werner, Daniel Jimenez

Agenda:   
Android merge

Android merge - over a conference call, Brandi and Daniel merged their android code functionalities to near the final stages of the project.

## 6 | Member contribution table

|  |  |  |  |
| --- | --- | --- | --- |
| Member name | Contribution description | Overall Contribution (%) | Note  (if applicable) |
| Brandi Werner | Android Developer( Connecting to database, Querying from database, Creating custom Row adapter, Displaying contents of database to screen, Creating custom Rows, Creating bookmark ability of events), Team Lead | 27% |  |
| Daniel Jimenez | Android Developer (Home View, Pop-up View, Bottom Navigation Bar, Calendar View, Visual Designer) meeting minutes,, android manual | 23% |  |
| Tsung-Han Hsieh | Android developer (Created test,Created Test suite, Debugged issues, Xml code on Rows), Tester, Android Manual writer | 20% |  |
| Kaytlin Lafleur | Web Developer (All of it), Presentation Creator, Web Manual Writer | 30% |  |