Alisha A.
Rohit T.
John W.

Buckets README

Operating instructions:

Download the ".ica" file from the CS 147 class Apple Developer Account and run it using XCode. Test the app on iOS 9.1, screen size 4.7" (iPhone 6 or 6S).

Limitations:

As with any prototype, there are several limitations to our Hi-Fi app implementation. Some limitations were more general, such as the absence of a settings page for the app, which others were specific to our core tasks. Almost every limitation was by choice: we simply didn't feel like the functionality was necessary to the core tasks of our project. Fortunately, we feel like these limitations don't affect the core functionality/tasks of our app. For one or two limitations (specified below if this was the case), there wasn't a reasonable way to implement it on iOS so we were forced to leave it out.

In terms of managing your bucket list, there are 2 main limitations. First, once you add an activity to your bucket list under a particular time frame, there is no way to change the time frame. The workaround for this is to delete the activity and then re-add it under the correct time frame. Second, there are a limited number of hard-coded suggestions for bucket list activities, so once the user adds all selected items to their personal bucket list the suggestions area will be empty.

Managing communities is the area of the app that has the fewest opportunities for customization, primarily because we haven't implemented an online backend for the app. Integration with the Facebook API has not yet been completed, so we pre-filled a sample of data that would normally be pulled from a user's Facebook profile. For example, the 'Join Communities' page in the initial setup flow would normally be filtered to show communities with which a user has a significant number of Facebook friends (or just nearby communities if the user chose not to log in with Facebook). However, the communities that show up are hard-coded, as well as all of the members, allowing us to focus on the app design and UI rather than dealing with web requests. Other limitations of community management are the inability to create additional communities or leave communities that you are currently a member of. Finally, users do not have public profile pages that show which communities they are members of.

Lastly, since our app is completely local, we don't have the ability to actually send or receive notifications from "other users". To make it seem like the person testing our app is being invited to activities with another user, we had to create fake invites to show up in the 'My Plans' screen. These invites appear every time the user adds one or more items to their bucket list, indicated by a notification badge above the 'My Plans' navigation icon. The biggest limitation of the invite system in our implementation is that you will never get responses to invites that you send, since all of our users are fake.

Wizard of Oz techniques

In order for testers to experience the full functionality of the app, we needed to make sure that they could invite people to do any activity on their bucket list. Therefore, every time an item is added to your bucket list, we give 4 randomly chosen fake users the same item. Then, when you go to make plans for that activity, there will be at at least 4 possible people to do it with.

As mentioned above, all incoming invites are generated programmatically to give the impression that you are being invited to activities (however, please don't actually show up - nobody will meet you there…).