CS 4720 - F17 - Final Project Proposal

Device Name: iPhoneXS Platform: iOS

Name: Alex Castillo Computing ID: agc3km

Name: Derin Serbetcioglu Computing ID: ds5bb

App Name: Event Manager

Project Description:

Bounce is an application for host management. The goal of our application was to be able to simplify the process of checking a guest list to see whether somebody has been invited or not. This was inspired by seeing the long lines at parties at UVA. By creating an account on the app, a user can create events and send invites to these events. Each guest is given a QR code link which can be then be used to gain entry to the event. In order to do this, we used a QR code scanner built into the application which can be accessed by the original event-creator by clicking the "bounce" button. Events are not static, guests can be added, but not removed (as that would be quite rude).

Platform Justification - What are the benefits to the platform you chose?

We chose to use the iOS platform because we felt more comfortable developing for iPhones and because we are both iPhone users.

Major Features/Screens - Include short descriptions of each (at least 3 of these)

We have a few major features. The first is the login and create event pages. From here, you can create a login or login directly if you already have one. They are stored on Firebase and are permanent. After logging in, you have a list of events you are hosting with a short description and date for each. If you click the bounce button from any of these, it takes you to the video camera where you can scan guest qr codes. If you click on any of the events (the cell, not the Bounce button), it takes you to an edit-event page where you can add more guests or view the current guest list. Finally, if you click the "create event" button on the top right, it takes you to a page where you can add a new event.

Optional Features - Include specific directions on how to test/demo each feature and declare the exact set that adds up to ~60 pts

For our optional features, we chose to use Firebase Auth for login functionality. Logging in will populate your main page with only the events that you have created, not those of other users.

We also used Firestore to save events you have created and their individual guest lists. These are pulled every time the app starts.

We used Twilio to sent text messages to guests with a link to their QR codes.

As another optional feature, we implemented a camera when you click the “Bounce” button on each event. It can scan guest QR codes and notifies the bouncer of whether or not the qr code belongs to the event which you are bouncing. If you swipe right on the camera page, it takes you back to the main page with all your events.

Testing Methodologies - What did you do to test the app?

The majority of our testing was done on our own devices, although we also tested qr-code links on Android phones that belonged to our friends.

Usage - Include any special info we need to run the app (username/passwords, etc.)

The app must be run on an iPhone. You can create a new account, or you can log in using the following information:

Username: alex@gmail.com

Password: alex123

Lessons Learned - What did you learn about mobile development through this process?

We learned a lot about how to structure information on Firestore. We tested various methods of storing event guest lists and eventually settled on having a “guest” collection with a field for the authentication code of the person who invited you and the name of the event.

We also learned a lot about QR code generation and sending images through SMS. We were not able to achieve our initial goal of sending QR codes directly through SMS, so we had to send a link instead.

Overall, this process taught us about managing our ambitions as we were using a lot of new technology from the get go (Twilio, Firestore).