

Assignment 6 - FIS

Alea Weeks - Chase Denecke - Brice Ng - Tyler Farnham - Qibang Liu

March 5, 2018

1 Product Release

Instructions:

- Download Android Studio: <https://developer.android.com/studio/install.html>
- Download the zip file of our github repository: <https://github.com/aleaw/cs321-flagellant>
- Extract the zip folder to somewhere you can find it (downloads)
- Open an existing project in android studio. Navigate to the folder you extracted our project to, select it, and open it. Click on the “SDK Manager” button near the top right corner of the android studio window.
- In the menu that this button pulls up, navigate to “Appearance & Behavior”, then “Android SDK.”
- Under the “SDK Platforms” tab, download “Android API 27”
- Get the Pixel 2 emulator
- At the top of your Android Studio window, select “Run”.
- In the dropdown menu, select “Run ‘app’”
- Select the emulator that you chose if it prompts you to. This should build and run the app.

2 User Story

User Story: A large-fingered user

- Pair Programmers: Alea and Tyler
- Problems: None
- Time required: 4 hours
- Current Status: Implemented, and tested
- Left to be completed: N/A (fully implemented)
- The spike diagram was not useful because the user story was such an easy fix. The user required only a minor adjustment to the application, and did not need any functionality to be changed, only the icon sizes. With such a simple user story, no other spike diagrams were necessary for this user story.

User Story: Set Donation Amount

- Pair Programmers: Brice and Alea
- Problems: N/A
- Time required: 10 minutes
- Current Status: Done
- Left to be completed: N/A (fully implemented)
- The Spike and UML sequence diagram wasn't very useful since this is also no hard to implement and no much problem encountered, so Spike and UML sequence were not necessary.

3 Design Changes and Rationale

Questions that we asked the customer:

- Implementing the PayPal API is going to be very difficult and will take a long time. We believe that using the Google Pay API will be beneficial to the rate at which we can complete the project. Do you want us to make the switch to the Google Pay API or stay our course with the PayPal API?

The customer wants us to make the switch over to the Google Pay from PayPal.

- Do you want the list of applications that the user can select from to be dynamically created based on the applications that the user has on their phone, or statically designated based on known distracting apps?

The customer wanted a dynamic list of applications.

- Do you want user settings to be saved, or require the user to reset their settings each time they open the app?

The customer wants the user settings to be saved between instances of the app.

Changes to our requirements and/or design specification:

The biggest change we made was making the switch over to the Google Pay API from the PayPal API. This changes much of the design of our app. We spent a lot of time researching the PayPal API and our conclusion was that it was not designed to do what we want it to do. After inspection of our other possible options for payment APIs that we could use we realized that the Google Pay API was probably the best fit. Because our app is android exclusive, we do not need to worry about google account creation. Instead of automatically donating money, at the end of each flagellation period we will now prompt the user to donate their money. There may be some other minor design changes that we make along the way based on this API switch.

We made a few changes to our design specifications. The list of applications that the user can declare distracting will be entirely pulled from the users device instead of being hard-coded into our application as was originally planned. In the process of researching how we would compare each app that the user launches against the apps that they select from the list of apps on their phone, we realized that we would need a specific permission to be granted by the user to detect the application. This permission cannot be granted by the system to third party applications, so we need to redirect the user to the settings app screen that enables them to give Flagellant permission to get their application usage data. This is a major design change and will impact usability.

In testing the functionality of our UI, we realized that it was really annoying for the user to input their settings every time they used the app. This led us to asking the customer if they wanted us to save the settings of the user or not. We enabled saving the user's settings as per the customer's requirement update request.

4 Tests

Describe at least one unit test for at least one major existing component/task in your system:

Unit test for settings screen

- Open up the check-marks app, navigate to the settings screen
- Attempt to change charity. Click on the “change charity” button and select one of the options
- Click on the “Dollars to Donate” button and enter an amount in dollars
- Click the for each app you wish to list as time-wasting.
- Click on the back arrow, then click on the settings cog again. Ensure that the settings you just entered a moment ago are still there
- Close the app and re-open it. Ensure that the settings you entered a moment ago are still there.
- If the above tasks can be successfully accomplished, the unit test for the settings screen will be considered a success

Describe at least one unit test for at least one major existing user story in your system:

Unit test for redirecting user to PayPal:

- Open app
- On login screen, click the link that says “Don’t have a PayPal account?”
- Click here to create one.”
- If clicking on this link redirects the user to the PayPal “create account” page, then the unit test has succeed.

Unit test for large fingered user:

- Attend thumb wrestling world championships in the UK
- Hire winner to test our app
- If they can successfully navigate our application, our unit test will be
- considered a success.
- In case the above method proves too costly, give a phone with the app installed to a random person in the valley library at OSU. If they can successfully navigate the screens, the test will be considered a success.

Describe at least one unit test for at least one part of the system that is not yet implemented.

Unit test for login with PayPal

- Open the app
- After setting a donation amount, a charity, and at least one time-wasting app, open said time wasting app
- Check PayPal account balance. If it has decreased by the donation amount set in the settings menu, the unit test will be considered a success

5 Meeting Report

Our meeting this week went SWIMMINGLY. Chase and Qibang worked on user story “User wants to set the amount they donate”. Chase also worked on writing the “Tests” section of assignment 6, and Qibang worked on the “User Stories” section, specifically the user who wants to set the amount they donate. Alea and Tyler worked on the user story “Large fingered user”. Alea also worked on UI improvements. Tyler worked on the “User stories” section of assignment 6, specifically the “Large fingered User” story.

Next week we hope to start finally making progress on login via PayPal (or possibly some other payment application if it turns out login with PayPal on an Android app is not feasible). We also hope to make progress on getting a list of apps that are currently running on the phone working so that we know when the app should make a donation. Tyler has done some research in this area and has discovered that there is a function that checks which app was most recently opened every 10 seconds. We may be able to use this function to meet the specifications of our app.

Our customer met with us on Sunday and worked on the project with us. Our customer was very reasonable, as they are part of our group.

6 Github url: <https://github.com/aleaw/cs321-flagellant/tree/Assignment-6-FIS>