Flagellant-SRP

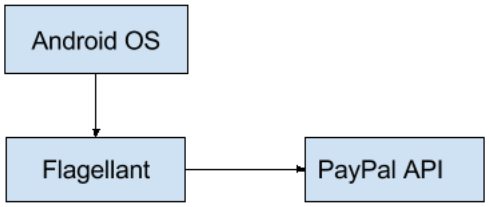
Regina Weeks - Chase Denecke - Brice Ng - Tyler Farnham - Qibang Liu

February 9, 2018

1 Project Description

Our group is planning to make an Android application called Flagellant. Our target demographic are people who recognize they spend too much time on their phone and want tools to help them control their own behavior. In support of this goal, our app will allow a user to create a list of "time-wasting apps", and every time the user opens one of these apps, Flagellant will make an automatic donation of a pre-selected amount to Charity.

This involuntary donation is what really separates our app from the rest of the crowd. As any economist can tell you, financial incentives have a notable effect on people's behavior. Penalizing our users for using free apps will almost certainly reduce their use of said apps. It is this strong penalty that sets our app apart from our competitors like "forestapp"1, which simply grows a virtual tree on your phone while you stay off of it. Below is a simple diagram showing how our app interacts with existing resources.



All other functionality is contained within the application itself. We will use Android Studio, and program our application in the Java programming language. Java is the standard language for android apps, and Android Studio is a well supported IDE with many tutorials and tools for developers.

2 User Requirements Definition

1. Mandatory Requirements
   (a) The app shall donate an amount of money to charity using PayPal every time a time-wasting app is opened
   (b) The app shall provide a means of linking to the user's PayPal account so that the app has a pool of money from which to draw donations.
   (c) The app shall have a running total showing the amount of the user's money donated to charity.
   (d) The app shall play a sound effect every time the user opens up a time wasting app to let them know that they are losing money.
   (e) The app shall include a "regret button" for testing purposes, which allows a user to refund the money they have lost to time-wasting apps
2. Stretch Goals
   (a) The app shall have a mean by which the user may select which apps he or she consider "time-wasting" and wishes to be penalized for using.
   (b) The app shall have a method by which the user may select a charity to donate money to.
   (c) The app shall allow the user to select the amount of money they wish to be donated to charity for every minute they spend on a time-wasting app.
   (d) The app shall provide a simple tutorial when first opened that guides the users through the process of selecting time-wasting apps, charities, donation amounts, etc.
   (e) The app shall contain an option to donate some percentage of their penalty payment to the developers.

3 Non-Functional Requirements

1. Mandatory Requirements
   (a) The app shall be stable and not randomly crash.
2. Stretch Goals
   (a) The app shall be intuitive enough that at least 50% of new users can link their PayPal account, select a list of apps they consider "time-wasting", and select a rate at which they wish to donate without anyone telling them how to use the app.
   (b) We plan to provide at the very least a page entitled "tutorial" within the app that the user may click on and read through to understand how to use the app, and at most a step-by-step tutorial guiding the user through the process of linking their PayPal account to the app, selecting time-wasting apps, selecting a charity they wish to donate to, and setting a rate at which they wish to donate.

4 Major Features

The major features of our program shall include the following:

1. A means of linking the user's PayPal account for the purpose of withdrawing money to be donated to charity every time the user opens a time-wasting application.
2. A means of selecting a list of said time-wasting applications.
3. A means of selecting a charity to receive the donations mentioned above
4. A means of selecting an amount of time during which the above donation mechanism will be active (A "focus session" during which the donation mechanism is active)
5. A summary screen showing at least the total amount of money the user has donated

5 Planning

1. Milestones and Tasks:

   (a) Upload a "Hello World" app to the team GitHub repository. This is a major starting point in the project, as it begins our introduction to Java, and begins the process of developing the app. This milestone depends upon our GitHub knowledge, and knowledge of introductory Java. The resources required for this are at least one developer, a GitHub account and repository, a computer, and references to begin learning Java in depth.

   (b) Have everyone contribute at least one useful file to the GitHub repository. This ensures that everyone is working on an assigned task, and that the project is smoothly under development. This milestone depends upon team member's productivity, and requires developers, computers to write and save code, and a GitHub repository.

   (c) Implement PayPal secure transactions into app to ensure user security when dealing with bank information. This is vital with anything dealing with personal information and is a very high priority. This PayPal will, at first, only donate to one charity, but more charities could become available later on in the project. This is dependent upon the app accepting a user, having correct bank information and using that information correctly, and correctly applying a donation transaction. Resources include developers' time and energy, online references, and a computer to write and save code.

   (d) Have a successful donation transaction. This is a major milestone because it means the successful implementation of a core feature of the project. A successful donation depends on a working API that allows for secure transactions, and a "time-waster" tagged app to be open. Required resources include developers and computers to write and save code, a GitHub repository to save to.

   (e) Allow selection of charity. This allows for more user interaction in the app and choose their preferred charity or organization to donate to. This is an important task because it demonstrates the completion of the core project and embellishes upon it by adding another feature. This feature depends upon being able to donate, meaning the PayPal API needs to be properly implemented, and transactions can occur. Only then will alternate donations be added. The resources required for this milestone are developers to implement the feature, computers to write code on, and a GitHub repository to save to.

   (f) Allow selection of "time-waster" apps. This is another major milestone in that makes Flagellant more useful. With this feature, users can tag multiple apps as "time-wasters", and donate whenever any of those apps are used. This feature, much like the milestone before it, relies on being able to have a set app that triggers a donation, and then adapting that to allow for a user input of additional apps. The resources required for this milestone are developers, computers, and a GitHub repository.

   (g) Put app on app store. This major milestone signals the completion of the project, aside from maintenance. When this milestone is achieved, that means that the project has been competed and the clients are satisfied with the results. The resources required for this are a computer and a developer to upload.

   This order of Milestone and tasks was chosen because it seemed like a natural progression into a final product. First, the design must be flushed out, then the actual coding will take place. The most progress will be made during this phase, because this is where the project will take shape. This phase relies heavily on learning to use coding utilities and applications, and so learning how to use the proper tools logically follows the planning phase. Many of our milestones occur in this phase, such as implementing secure transactions and allowing a user to add to a personalized list of time wasting apps. Our last milestone, putting Flagellant on the app store, is the final milestone, as it releases the final product into the market, signaling that the project is complete.

6 Schedule

	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
(1/21-1/27)		Finish Vision Statement (Group)				Group Meeting	
(1/28-2/3)	Group Meeting	Finish Requirements Overview (Group)	Team GitHub repository (Alea) Install Android Studio (Group) Java Tutorial (Group)				
(2/4-2/10)	Group Meeting (individualized deliverables)	Basic "Hello World" app developed (Tyler)		App Design Finalized (Chase)			Users and personal info stored (Group)
(2/11-2/17)	Group Meeting	UI Testing (Group)		Research PayPal API (Alea)			Transfer funds via PayPal (Austin/Alea)
(2/18-2/24)	Group Meeting	UI Testing (Group)				Select "time-waster" apps (Brice)	
(2/25-3/3)	Group Meeting	Unit Testing (Group)	Unit Testing (Group)	Unit Testing (Group)	Unit Testing (Group)	Unit Testing (Group)	All unit testing Done
(3/4-3/10)	Group Meeting	App testing (Group)	App testing (Group)	App testing (Group)	App testing (Group)	App testing (Group)	Submission of Project

1. Project Tracking To stay on schedule, we have decided to hold weekly group meetings to check our progress and maintain a healthy workflow. We will peer review each other's work and edit based on the group's decisions.

2. Risk Management

   (a) The biggest risk is not allocating enough time to the project, due to either time constraints or lack of procrastination. This is the biggest risk, as delivering an unfinished or incomplete product is not acceptable. To minimize this, we recognized the scope of the project immediately and are making efforts to start the project as soon as possible. Weekly group meetings and updates will greatly encourage productivity.

   (b) Another project risk is the loss of code. Should the newest version of the project suddenly be deleted, and no previous version is found, then the project would have to be restarted. To minimize the risk of that happening, using a repository and then forking and merging would help to ensure there is always at least one working version of the project.

   (c) One risk, particular to this specific project, is the security aspect of the project. With very little app development experience as a whole, developing an app that can securely get personal information like bank account numbers will be very slow going. To develop an app that does everything that it was envisioned to do will require the correct amount of time to be dedicated to the specific core functions. To reduce the risk of a product that is not as intended, we will allocate more time to the core features of the app to ensure a working product.

7 Use Cases

1. New User for the Flagellant app

   (a) Goal: A new user downloads and signs up for the Flagellant app. They configure the app's settings to their personal preferences.

   (b) Actor: A new user of Flagellant who wishes to stay focused longer.

   (c) Preconditions: The new user has an android smart phone, a PayPal account, and has downloaded the Flagellant app from the android app store. Now they need to create an account and configure their settings.

   (d) Postconditions: Due to the app's intuitive UI, the new user has successfully logged into Flagellant with their PayPal account and has configured the settings in a way that suits them.

   (e) Flow of Events:

      i. The new user opens up the Flagellant app.They are greeted with a login screen and are prompted to login via their PayPal account.

      ii. They click the "login with PayPal" button new secure screen pops up where they enter their PayPal username and password.

      iii. The login is successful and their PayPal account is now linked.

      iv. From here a new screen appears in which the user chooses the charity they wish to donate to

      v. They choose the Wildlife Conservation charity as the one they wish to donate to.

      vi. From here they are sent to a Settings screen where they set the amount of money they wish to donate each time they open a time wasting app while Flagellant is running, the amount of time they wish to focus, the charity they want to donate to, and they can add apps to their list of "time wasting" apps.

      vii. Now they are sent to a Summary screen where they can review their account information and press a button to start their Flagellant session

      viii. The new user is pleased with how easy it was to set up their account and they close the app, excited to use it next time they need to remain focused

   (f) Quality Requirements:

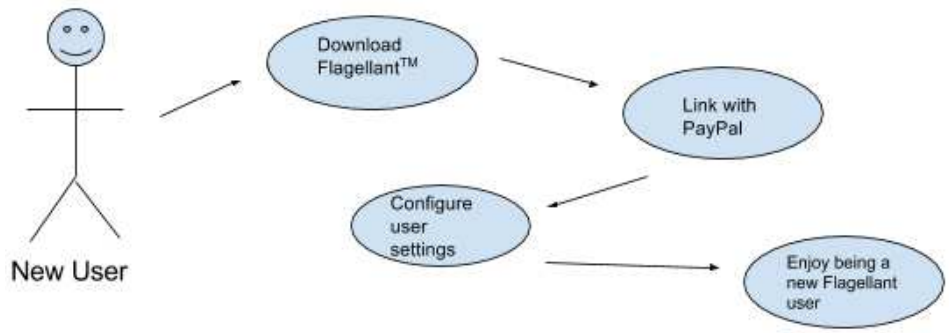
      i. Flagellant must be able to login in with PayPal account.

      ii. Flagellant's UI must be clean and intuitive so the user can navigate the app with ease

      iii. The settings the user sets must be stored and the app must update the settings when the user changes them

   (g) Error Scenario:

   A main error scenario for this use case would be if Flagellant didn't properly link to the user's PayPal account in which case, the main motivating factor of the app (losing money) would be sacrificed. The best remedy for this would be consistently testing the app and making sure logging in and linking with PayPal flawlessly executes.



2. Motivation while cramming an assignment

   (a) Goal: to complete an assignment that's due in 1 hour without getting distracted

   (b) Actor: sleep deprived college student

   (c) Preconditions: Student has and is using the Flagellant app.

   (d) Postconditions: The student completes their assignment just in time because they weren't tempted to open up time wasting apps, thanks to the punishment their Flagellant app would have inflicted upon their bank account.

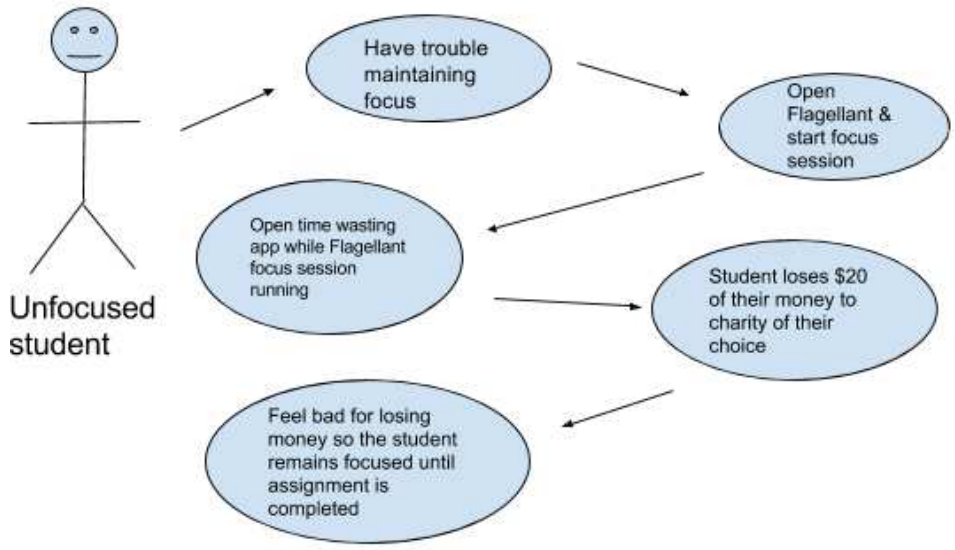
   (e) Flow of Events:

      i. A student has procrastinated completing an assignment that is due in 1 hour.
      ii. They are sleep deprived and have a history of an inability to remain focused and motivated
      iii. They often waste time by browsing time wasting apps instead of doing their work
      iv. Because of this, they downloaded the productivity app Flagellant
      v. They set the amount of time they wish to be productive to 1 hour
      vi. 30 minutes of quality work time have gone by and the student is making progress on their assignment
      vii. The student becomes tempted to open up tinder and mindlessly swipe left and right on local singles near them
      viii. The student resists this temptation because if they do, Flagellant will donate $15 of their money to charity
      ix. The student regains focus and completes their assignment on time and feels accomplished

   (f) Quality Requirements: Flagellant must be a functional app in order to be a motivating force. So it must be able to donate the user's money to charity and be able to tell if they have opened up ay time wasting apps in their specified amount of focus time.

   (g) Error Scenario:

   A main error scenario for this use case would be if the user was penalized for opening an app that was not on their "time wasting" apps list. A remedy for this would be refund the user their money through a troubleshooting screen.



3. Save the money

   (a) Goal: Quit the bad habit of browse and buying thing they don't need.

   (b) Actor: New graduate that just get a job and don't have a good idea to save money

   (c) Preconditions:

      i. Bad habit of buying useless thing on a phone app

      ii. Likely to spend time on browsing and buying

      iii. Flagellant app installed

   (d) Postconditions: The company employee less likely to spend time on that app and put more time on his/her work because he/she can't afford to spend more than expected money on buying non-useful product or even just browsing them.

   (e) Flow of events:

      i. When the Flagellant app in running on the background

      ii. User open the buying app and browse on it

      iii. As soon as it open, Flagellant start counting

      iv. Before the user find an interesting product and trying to click on it, user realize that he/she already waste money on just browsing on the app, then user quit on the app and start doing something useful.

   (f) Quality Requirements:

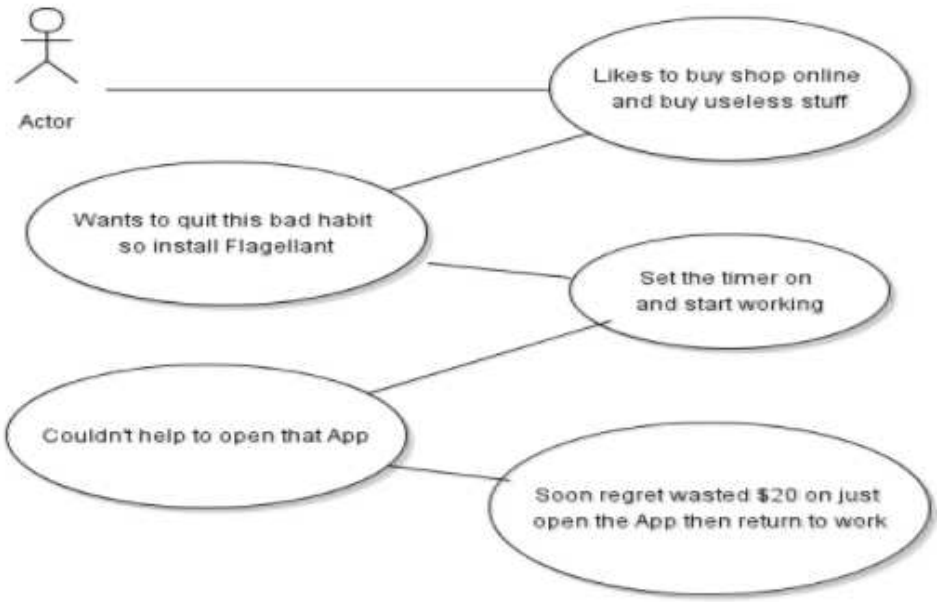
      i. Make sure it works

      ii. Make sure it can detected the app

   (g) Error Scenario:

   If the user doesn't have a Paypal account or Flagellant doesn't properly link to Paypal account then it will fail to punish user for wasting time, the remedy is sign up for a Paypal account or retest to make sure Flagellant links to the Paypal.

   These three use cases cover the most important scenarios because our app is not a very dynamic one. The main actions out users will be performing with our app are setting up a new account (Use Case 1) and then using the app with their preferred settings (Use Cases 2 and 3). Other than that, there is not much variety in the actions within our app.



8 Meeting Report

As of Sunday, 1/28/18, our group has met once and completed assignment 2. We have outlined both the functional and non-functional project requirements. We have outlined 3 use-cases, and created a preliminary schedule including future group meetings at which we will hash out the details of the remainder of the project. Our report also contains a list of risks that could potentially derail the development cycle as well as strategies to mitigate those risks. We have all downloaded and installed Android studios, the IDE on which we plan to write and test the code for our application. We have also created a GitHub repository, and given all group members the ability to access and contribute to. By the end of next week, 2/4/2018, we plan to have finished delegating tasks to everyone in the group. The reason we are waiting until next week to do this is that we need some time to learn the basics of developing an app and figure out what we actually have to do. Between Sunday, 1/28/2018, and Sunday 2/4/2018, we plan for everyone to complete a basic "Hello World" programming task for a Java app. Our goal is to be ready to delegate tasks to individual group members by next week. Chase Denecke contributed to the project by writing up a the meeting report and the project description. Alea Weeks contributed by writing 2 use cases, creating the group GitHub, and doing research on developing Java apps. Qibang Liu contributed by writing a use case for a person who spends too much time and money shopping online. Brice Ng contributed by creating a development schedule, listing out risks and mitigation strategies, and making a list of milestones and descriptions of said milestones. Tyler Farnham contributed by formatting our SRP assignment in Latex, doing research on PayPal fees, and helping to diagram out the app.

9 References

References

[1] Forest. (2018). Forestapp.cc. Retrieved 28 January 2018, from https://www.forestapp.cc/en/