Project Title: GoodFriend
Team Members: Thomas Ermis, Tejas Roysam, Taiyi Ouyang, Christopher Frazier
UT EIDs: tfe95, tbr388, to3879, clf2328
Repository URL: https://github.com/DCodezy/EE461L-Project

Motivation

Sometimes all you need is a friend. Everyone has their weak moments, whether it be walking past that taco stand on your way home, surfing through your phone before going to bed, or even as severe as recovering from nicotine addiction. The caring word of a friend in those times of weakness can make all the difference. Unfortunately - that friend friend cannot always be by your side every second of the day. Fortunately - your phone is.

What we are proposing is a companion application to help in those times of weakness and give the user greater control over their lives and habits. The Android application GoodFriend will look at the current stage of the user in their progress towards goals and recognized key triggers in a timely manner, which might deter them. It does so by giving the user friendly recommendations such as “Hey Chris, I noticed you are pretty stressed today. How about we go for a walk outside for five minutes?” Thereby giving the user that little extra push to avoid temptation, all with purely friendly and compassionate interests in mind.

Feature Description

Initialization:
User downloads app for first time, screen pops up saying greeting and purpose of app, user selects habit to break, user is redirected to status screen (days since program started, risk level, current day of plan, etc).

User Stories:
User Story 1: Alice has a smoking problem. She wants to stop smoking, but she always relapses after a week or two of going cold. Finally, Alice decides to try out the GoodFriend app. She selects “smoking” as a habit she wants to break and reads through a simple description of the program. Over the next couple of days, GoodFriend asks Alice about her general emotions and stress levels. Alice avoids situations that have the ‘biggest risk’ of her relapsing as recommended by the app. After a week of going strong, Alice starts to feel the urge to relapse into smoking. She inputs her stress level as ‘high’ the next time GoodFriend asks how she is doing. After a few minutes, Alice’s phone buzzes and she receives the notification “Hey Alice, I’ve noticed you’ve been stressed today. How about we go outside and get some fresh air?”. Alice follows the app’s advice and feels refreshed enough to continue her day without incident. Over the next few weeks, Alice is able to stay away from smoking with the help of her GoodFriend. She is continually encouraged by friendly reminders and is able to check her status of completing the program in real time along the way. She completes all stages of her program a couple of months later smoking free.

User Story 2: Bob has a habit of skipping class. It's not always intentional. Sometimes it just happens. One day, Bob decides to try and break his habit by downloading the GoodFriend app. Bob selects the "skipping class" habit and goes through the welcome screen. Every weekday morning, GoodFriend posts notifications reminding Bob to go to class. Bob updates the app with his mood throughout the weeks. One day, Bob decides to skip class. He pushes the "relapse" icon on his app. The app asks him a couple of questions about how he is currently feeling and then restarts the program to day 1. Over the next few weeks, Bob is tempted to skip class, but the app seems to recognize this desire and suggests friendly advice to convince him to go to class. Bob completes each stage of his program over the next few months with the help of this app. Bob breaks his habit of skipping class and finishes the semester with a 3.7 GPA.

Non-Functional Requirements:

GoodFriend is designed to be a background app, and most of its interactions with the user are notification based. Therefore, consistent performance and minimal persistent resource usage is important to ensuring a positive user experience. Since the app is designed to be a stress aid, its usability and simplicity to the client is also a key factor.

Functional Requirements

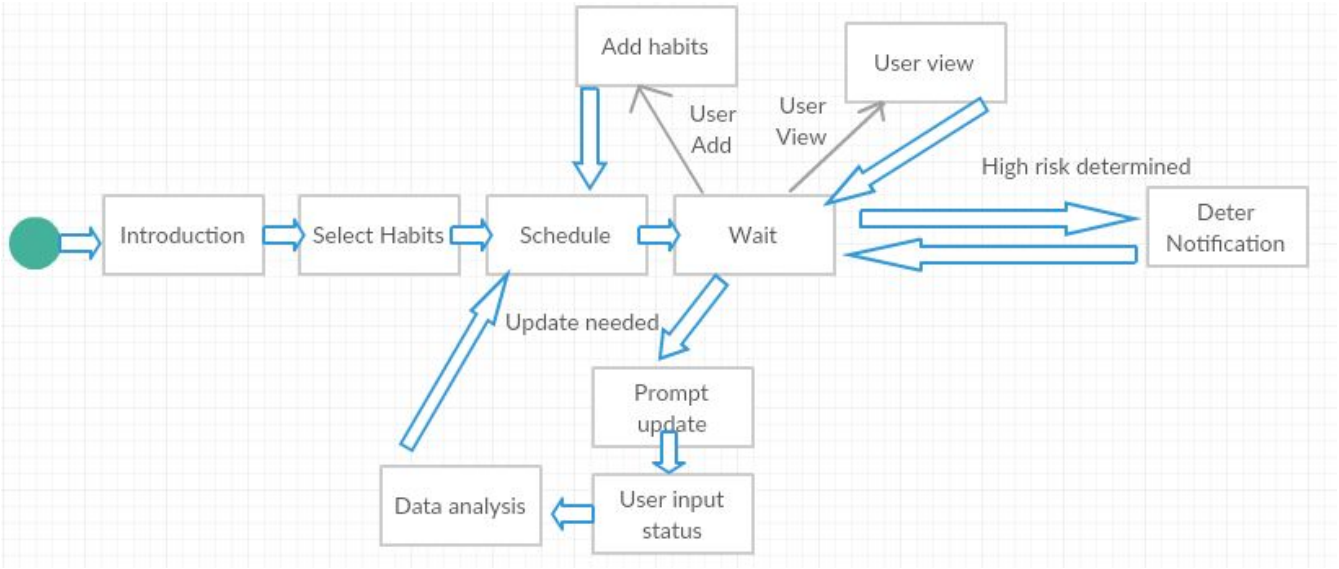

Figure 1: GoodFriends UML State Diagram

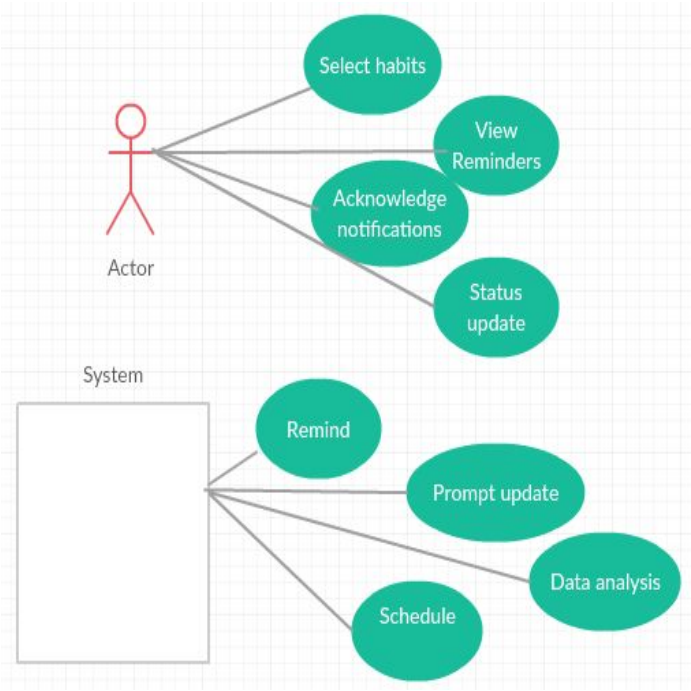

Figure 2: GoodFriends UML Use Case Diagram

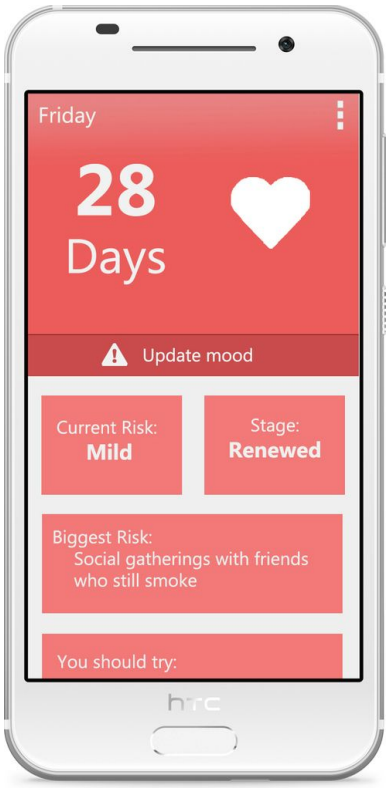

Figure 3: UI Mockup Snapshot

Design Document

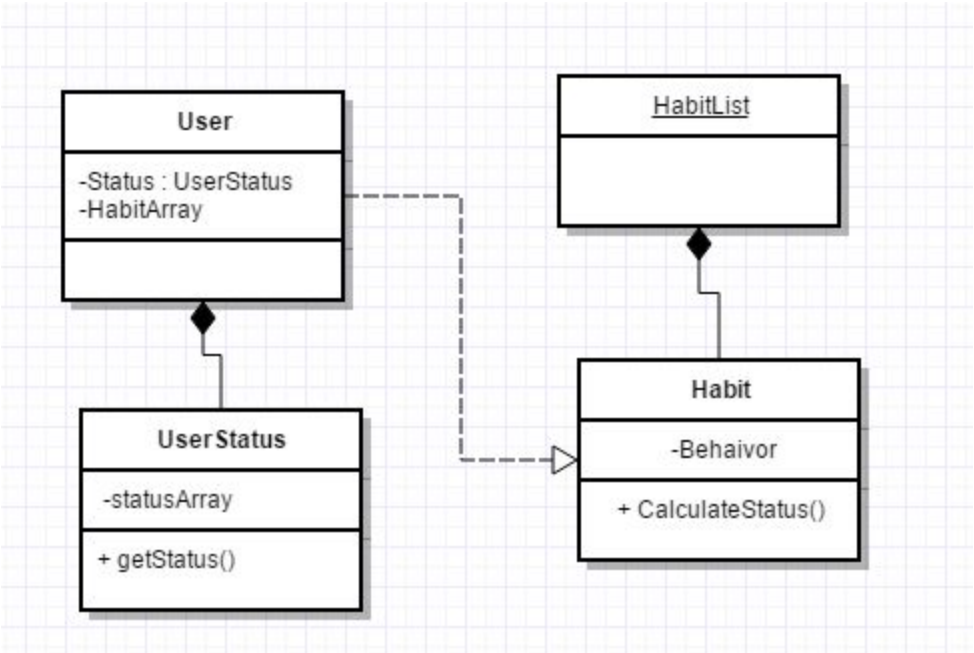

Figure 4: UML Class Diagram

In order to test this app, we will write both unit and system tests. The unit tests will confirm that each individual part of the system works. For the system tests, we will implement the app on different Android phones to ensure similar behavior across different devices. The test will consist of a single habit, with the ability to add on new habits later. The key functionalities we are looking for are the automated notifications, and proper results from the data analysis based on the user's habit patterns. The data we will be using is primarily our own behavior, with the app adjusting to what we input.

Timeline:

- 2/22 - Project proposal presentation
- 3/20 - Framework for app set up, can deploy but little behavior.
- Mid to End of April- Functions for app set up, app reaction to data and habits determined
- May- App tested in real life use cases, app is finished and has the potential to add new habits later.

Feasibility

GoodFriend is designed in a modular fashion, so that adding new functionality is independent of the user interface or experience. The basic interface will be primarily notification-driven, which is an established feature of Android and will therefore not exclude the majority of devices. The app will not require any Android functions newer than Android 11 (Honeycomb), which was released in 2011. In addition, there are no special permission requirements (such as location data, internet connection, or access to contacts), and no third party services or apps required.

We may find it necessary to obtain and use external data to create profiles for some habits. For example, in the case of drug addictions there are definite stages of relapse and withdrawal, and integrating these stages into GoodFriend's analysis of the user is important to its functionality as a helpful tool. In these cases, however, the app itself will not be fetching data in real time, but rather will be using preloaded cherry-picked data that we the developers have created or used. Therefore, there is no need for the use of unfamiliar APIs, or a reliance on web hosted data.

Finally, GoodFriend will not require any backend support. The entire user experience will be completely functional without a server-side service. This means that there will be no need to write any backend functions or host a server, and thus no need to facilitate client-server communication.

Contribution

Thomas Ermis: Feature Description
Tejas Roysam: Feasibility, Non Functional Requirements
Taiyi Ouyang: Design Document, UML Diagrams, Functional Requirements
Christopher Frazier: Motivation, UI Design Sketch