Three Tri∆ngles

Software Requirements Specifications

Alex Anderson
Kurt Anderson
Trevor Gentner
Nate Hochstetler
Laura May

03/01/2018

Preface

This is version 1.0.0 of the requirements document of “Three Tri∆ngles” app. This document is intended to represent and capture the vision of the app. It will also provide a reference guide as well as the process of development throughout.

Change History

2/10/2018	Entries for the features section updated	Alex Anderson
2/12/2018	Competition section completed	Laura May
2/13/2018	Class Diagram created	Kurt Anderson
2/13/2018	Motivation, scope, and goals in the introduction written	Nate Hochstetler
2/15/2018	Definitions written	Trevor Gentner
2/22/2018	Activity Diagrams created and descriptions added	Trevor Gentner, Nate Hochstetler, Alex Anderson, Laura May, Kurt Anderson
2/22/2018	Use Case Diagrams created and descriptions added	Trevor Gentner, Nate Hochstetler, Kurt Anderson, Laura May, Alex Anderson
2/22/2018	Functional/Non Functional requirements section added	Kurt Anderson and Nate Hochstetler

Table of Contents

Introduction

Motivation/Purpose

The purpose of this project is to provide a gamified approach to habits. Whether it be breaking a bad habit or creating a good habit, this allows the user to track progress towards their goal. The team's hope is that by presenting the app in a fun way, people will be more inclined to check in with the app, and by doing so, break or create the habit. This document will provide the necessary details and specifications for the project.

Scope

In this project we hope to achieve the minimum viable product before the deadline, allowing us time to fine tune the application and potentially add any additional features we feel necessary. Our application is an easy and fun way to break habits that can be utilized by any age group or person. Everyone has bad habits to break or good habits to create.

Goals

The goal of this project is to create a complete Android application with the features and specifications detailed in this document. Our team wants to provide as many features as we can in the small time frame, both to better the app and to better our knowledge of areas we have little experience. Through this process, we will learn about the development process and lifecycle of Android applications, as well as how to see a project through completion and testing.

Key Definitions

Habit	A settled or regular tendency or practice, especially one that is hard to give up.In Three TriΔngles it is something that a user is keeping track of by logging actions.
Accountability Partner	Person who is capable of logging another user's habit to keep them on track

Google play games	Real-time multiplayer gaming capabilities, cloud saves, social and public leaderboards, and achievements
Push notification	A way for an app to send information to your phone (via a badge, alert, or pop up message) even when the app isn't in use.
Sidebar	Drawer that contains settings related to habits and user settings
Streak	How many days in a row that a user completes their habit tracking
Local storage	Data that is stored with the appdata on the user’s device

Project Description

Features

Accountability partner

Accountability partners are people that a person who is trying to form a habit or break a habit uses to keep them accountable for an action. An example of this could be a fit friend that goes to the gym with their friend who is trying to go to the gym daily. That person verifies that they are indeed going to the gym everyday and doing what they set out to do. We want to implement this idea into the app to give the same benefits to people keep track of their habits from the digital world. Giving the ability to select another user as an accountability partner will allow that person to determine if the person has completed their goal. This prevents the user from inflating their own habits while not actually doing them and gives them more of an incentive to continue pursuing their goals.

Push Notifications

Having the ability to send push notifications to the user will prevent them from forgetting to do their intended goal. If their goal is to go to the gym everyday, then this feature can send them a notification at a specified time to remind them to go to the gym at that time.

Achievements

The idea of gamifying the habit building and goal pursuing experience can help keep people on the correct path. We want the same neurons that light up when a user advances a level in a video game to occur when they are using this app. The use of achievements can help to gamify the experience by giving a badge or token to symbolize the struggle that the user has been through and overcome.

Cloud Synchronization

In order to give users the highest chance to achieve their goals, we need to make the app as convenient as possible. People are always looking for excuses to not complete their habits. In order to help with this we want the users appt to have the exact same data no matter which device they are currently using the product on. This can be achieved through synchronizing their data on the cloud in order to easily transfer it between devices.

Google Play Games Integration

Google offers a system for measuring progress and awarding users with achievements. We would like our achievements to be integrated with Google in order to help standardize the achievements. This makes them easier to share with friends and increases the gamify experience. If the user gets achievements from something they are already familiar with and using, it can help make the whole idea of habit making and goal setting seem more like an expansion pack to a game.

Facebook Login

In helping the user with simplicity and convenience, we want the user to use an account they are already familiar with. By allowing and implementing Facebook logins, we hope to help the user transition into the app easier. It will also allow for super easy sharing of achievements or to look for an accountability partner.

Google+ Login

With this being an app developed for android, the ability to quickly login with a user's Google account will streamline the onboarding process for a user. By having the ability to login with Google, the user's transition into the app will be more seamless as it will just be another extension of their Google account. It will also make achievement sharing with Google Play Games much easier.

Local Storage

In the event the user doesn't want to sign up for an account with our app or login with any of our integrations, we want the user to still be able to use the app. Therefore, we wish to implement a local storage that is stored on the phone for the user. This would not however be backed up by the cloud as verifying what user has what data wouldn't be possible. It would allow someone to just simply use the app on their phone.

Requirements

Functional

- App will keep track of Google profile information
- Notifications will be sent to the user to remind them about their habits
- User will be able to receive achievements from Google Play Games to reward their progress
- User will be able to keep track of multiple habits at once
- User will be able to view their habit progress in multiple views (ie calendar, streak, etc)
- User will be able to add an accountability partner to track each others progress

Non-Functional

Google Play Services

- Billing, achievements, and login shall be accomplished using Google Play services

Amazon AWS / EC2 / Parse

- AWS will be the main service to host the database for our app. EC2 is the plan under AWS we will be using. The data for the user will be saved in this database
- Accountability partner will be another user of the app. This needs to be considered with security of the app.

Android Push Notifications

- Will utilize push notifications to send reminders to check-in for a habit, as well as send status updates on progress.

Competition

	Habitica	HabitBull	Productive	Momentum	HabitMinder
Accountability Partner	🟢	🟡	🔴	🔴	🔴
Push Notifications	🟢	🟢	🟡	🟢	🟢
Achievements	🟢	🟡	🟡	🟡	🟡
Cloud Synchronization	🟢	🟢	🔴	🔴	🔴
Google Play Games Integration	🔴	🔴	🔴	🔴	🔴
Facebook Login	🟢	🟢	🔴	🔴	🔴
Google+ Login	🟢	🔴	🔴	🔴	🔴
Local storage	🟢	🟢	🟢	🟢	🟢

Android	🟢	🟢	🔴	🔴	🔴

Habitica

Habitica is an accountability app that incorporates gamification. Successfully completing your goals results in achievement badges and currency that can be exchanged for gear. It does not use Google Play Games Integration. Habitica is also very complicated game, and requires a lot of time to register every habit.

HabitBull

HabitBull's design includes a calendar view and a streak progress view. Achievements are only kept track of by streaks. The social component of the app is that it shows people who are wishing to form the same habit as you, but they are not people that you have a relationship with in real life, instead they are strangers. HabitBull does not have Google integration.

Productive

Productive is a strictly iOS app that organizes habits based on when they are to be done in the day. It sorts them into morning, afternoon, and evenings. The free version of the app is very limited. To get access to stats, reminders, and a different interface, you have to pay to upgrade. Achievements are only measured by streaks. There is not Google nor Facebook integration.

Momentum

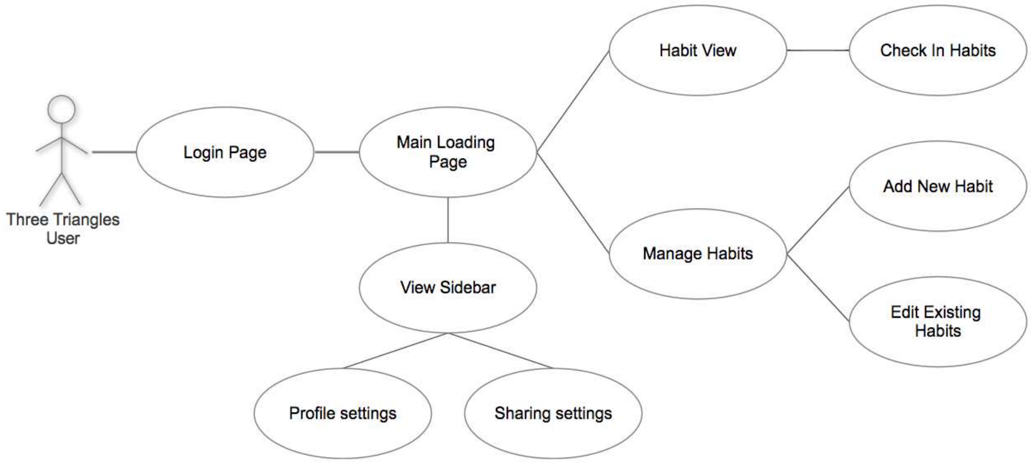
Momentum is a strictly iOS app that has a very minimal design. The app has a simple, calendar view where habits are formed using streaks as achievements. It is strictly local storage and does not require any form of username or login.

HabitMinder

HabitMinder is an iOS app that tracks habits with streaks and percentages. The interface is simple, but colorful. It is strictly local storage and does not require any form of username or login.

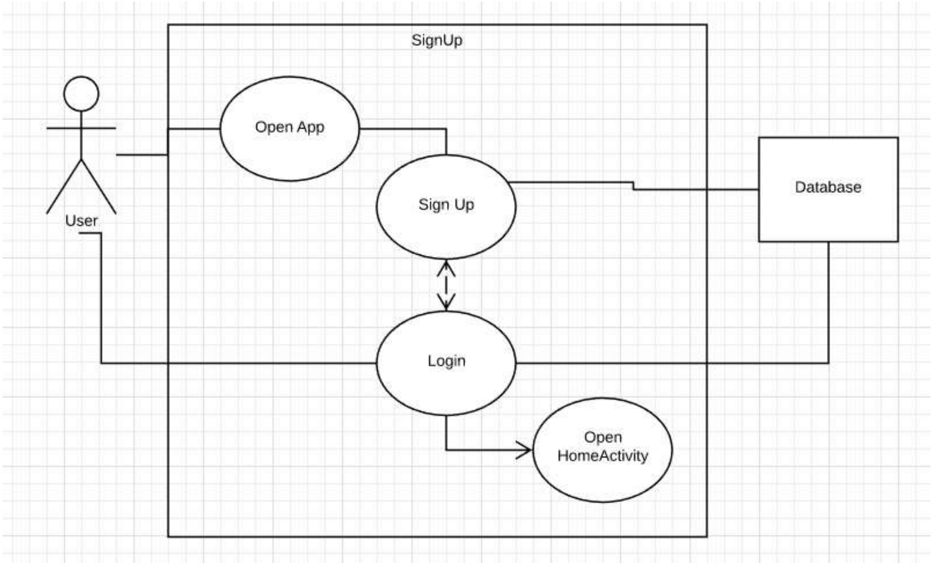
Diagrams

Overarching Use Case Diagram



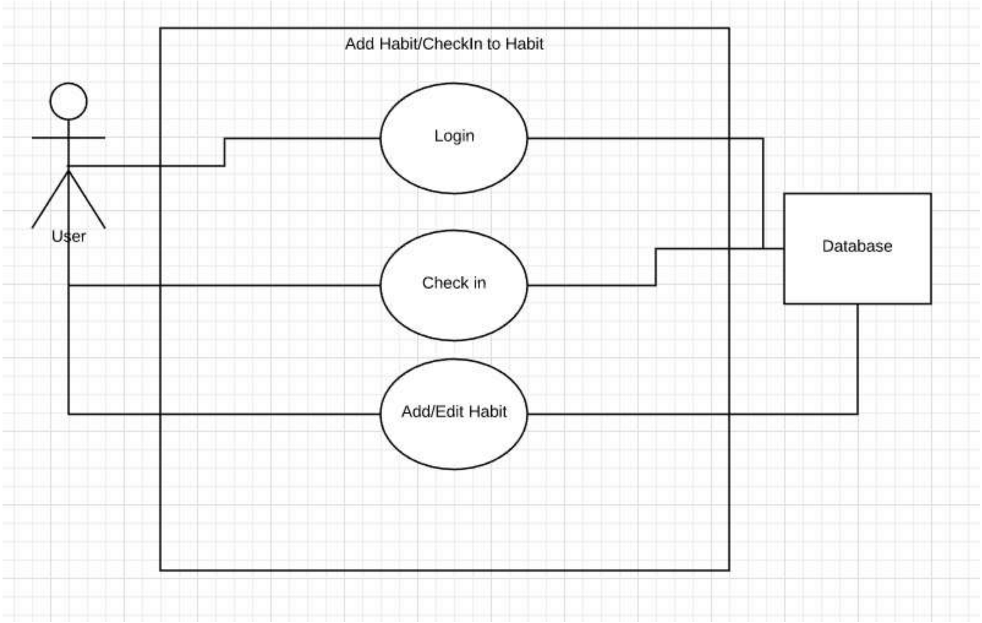
The main use case diagram. This is a single user app, therefore there won't be many connections. The user interacting with the app and the app interacting with the database are the two main cases. Some of these relationships are broken down below.

Use Case Diagram 1



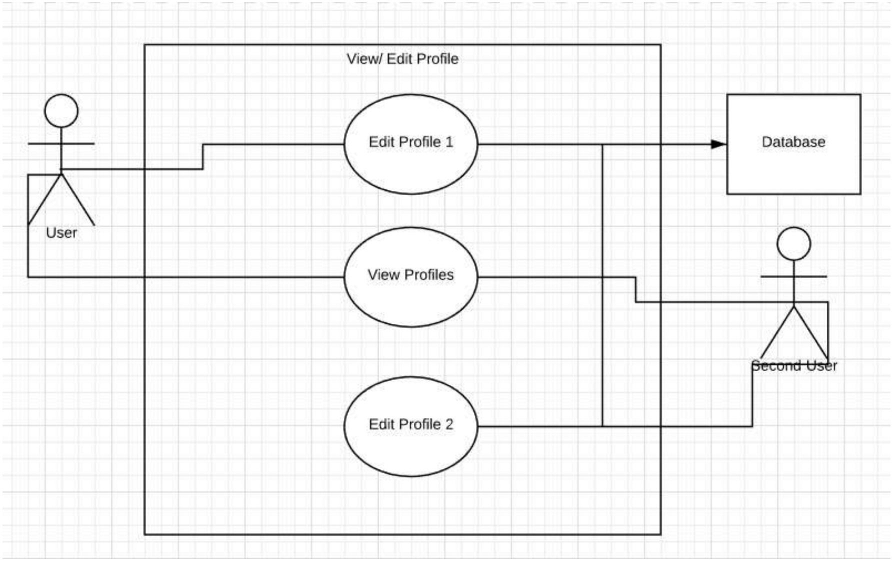
The first use case diagram. The user at the start of the app will be able to sign up or login. Both of these actions will require a check with the database. Once verified, the user will move on to the HomeActivity, propagated with user specific data.

Use Case Diagram 2



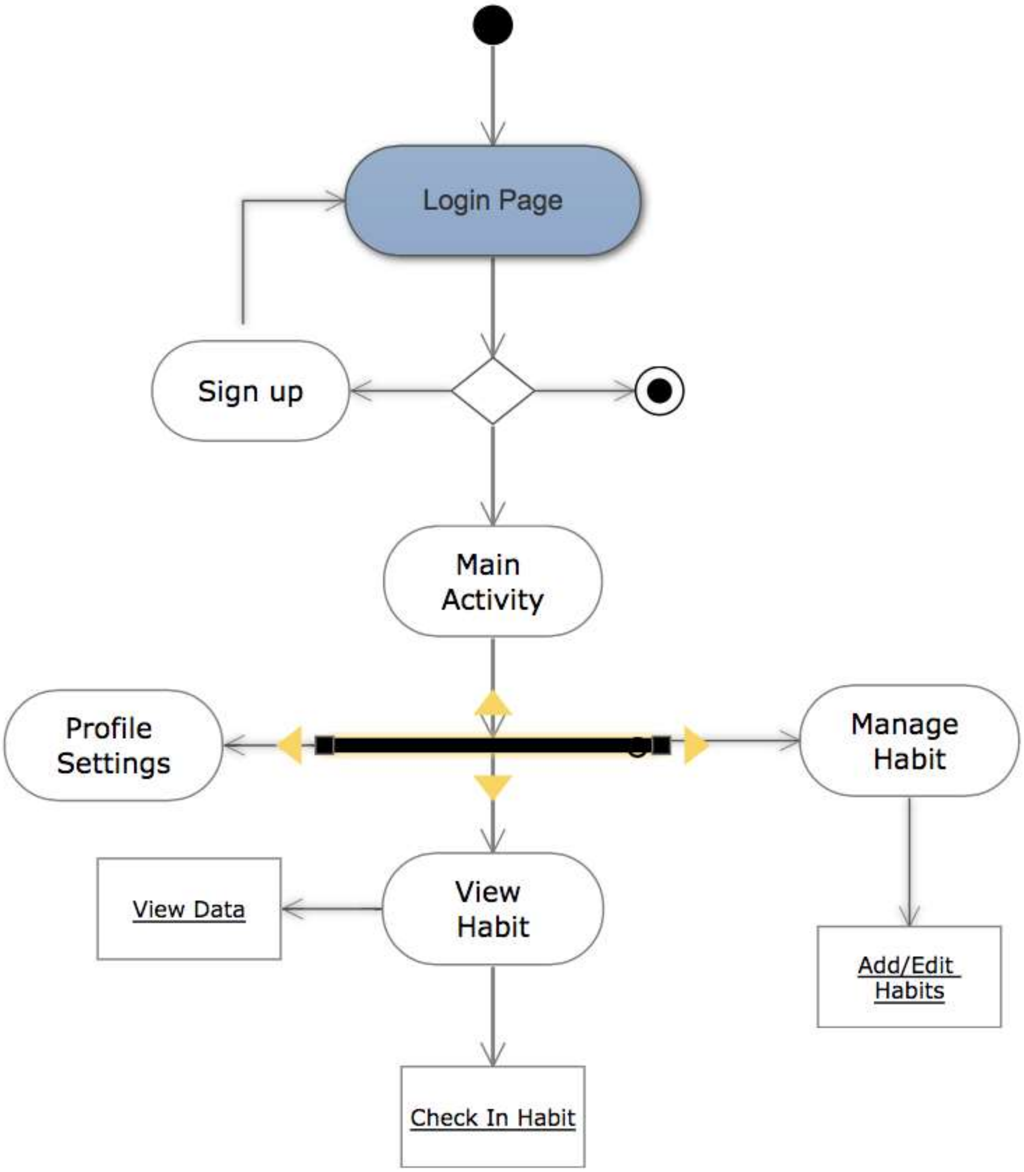
The second diagram details adding a habit to the app The user will login. He then has the ability to check in and then to add or edit a habit. All of these commands will communicate with the server to make sure the changes are propagated correctly.

Use Case Diagram 3



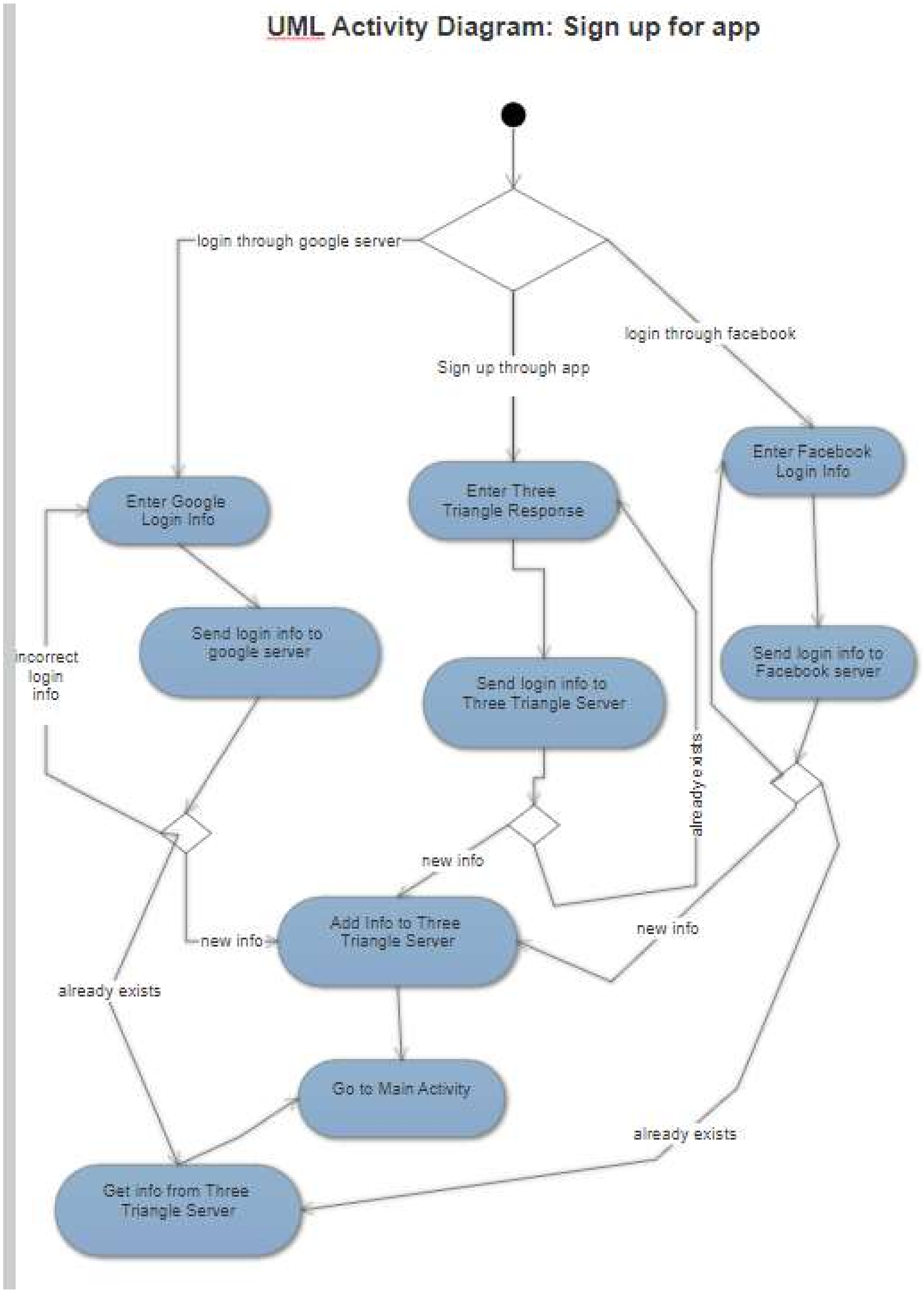
The third use case details the interaction with viewing and editing a profile. In this situation, we have two users. Both users are allowed to view whatever profile they want. However, user 1 will only have access to editing profile one, and user 2 will only be able to edit profile two. Both edits will interact with the server.

Overarching Activity Diagram



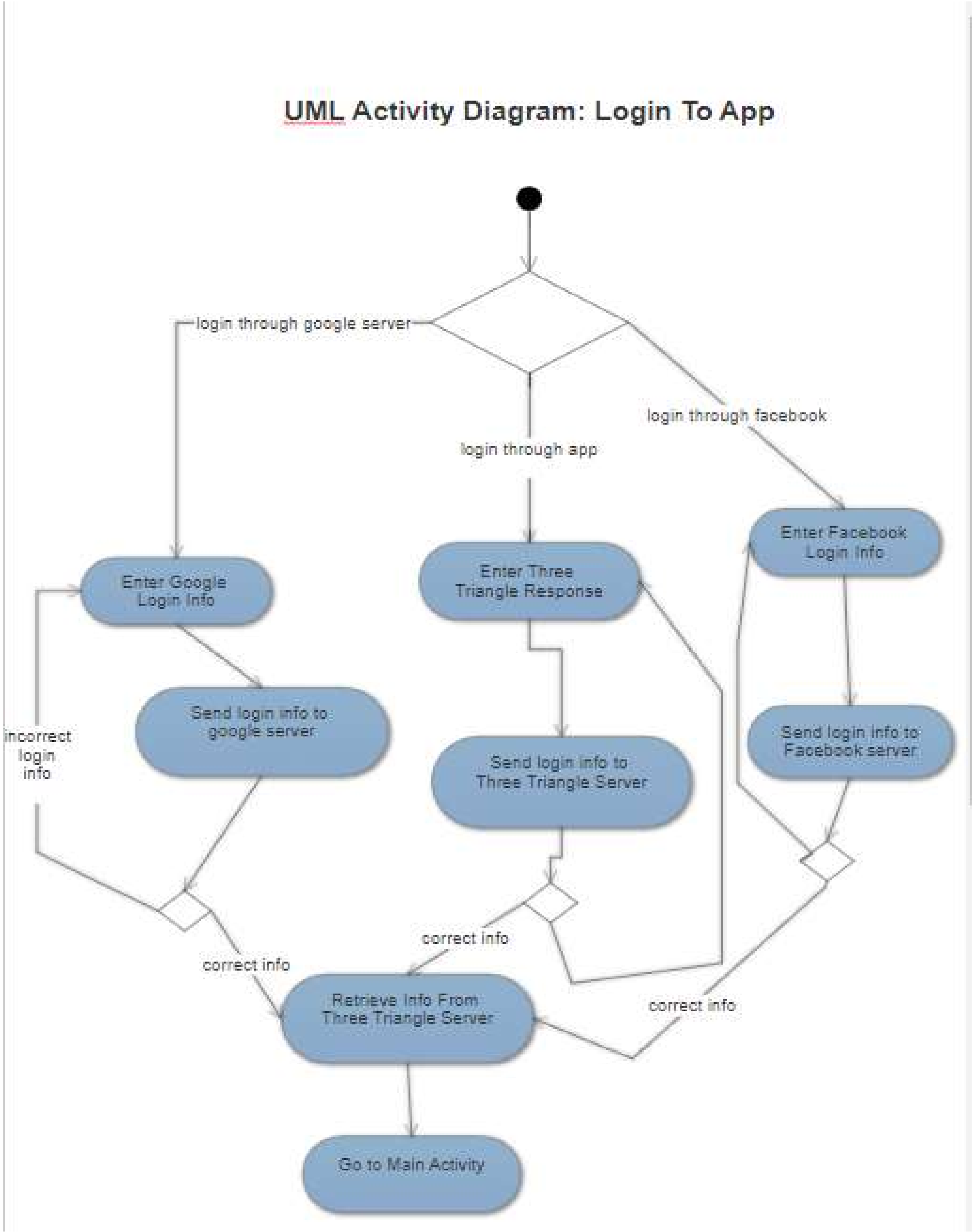
The diagram walks through the high level use of the app and how any activity would flow through it. The user would start on the Login page and have the option to create a new account or sign in. Creating a new account will bring them back to the login page. After logging in they will be brought to the main activity where they will have multiple options of next activities. From the main screen they can view habits, manage habits and go to their profile settings.

Activity Diagram 1


UML Activity Diagram: Sign up for app
login through google server
Sign up through app
login through facebook
Enter Google Login Info
Enter Three Triangle Response
Enter Facebook Login Info
Send login info to google server
Send login info to Three Triangle Server
Send login info to Facebook server
incorrect login info
already exists
new info
new info
already exists
new info
Add Info to Three Triangle Server
already exists
Go to Main Activity
already exists
Get info from Three Triangle Server

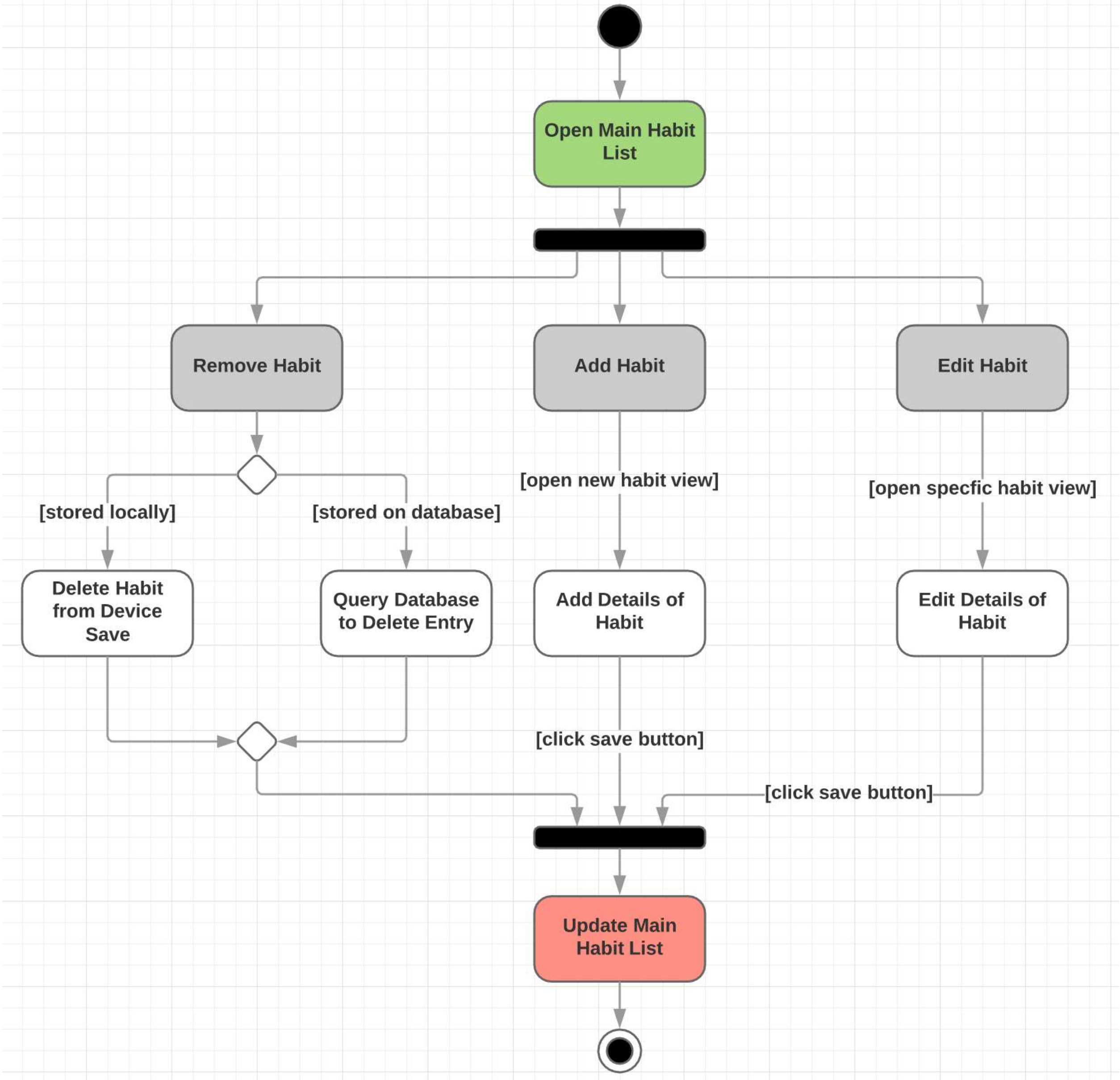
This diagram goes through the sign up activity. A user will be presented with three different methods of signing up to the application through the help of facebook logins and native google logins. After information is collected the user database is updated and a new account is created.

Activity Diagram 2


UML Activity Diagram: Login To App
login through google server
login through app
login through facebook
Enter Google Login Info
Enter Three Triangle Response
Enter Facebook Login Info
Send login info to google server
Send login info to Three Triangle Server
Send login info to Facebook server
incorrect login info
correct info
correct info
correct info
Retrieve Info From Three Triangle Server
Go to Main Activity

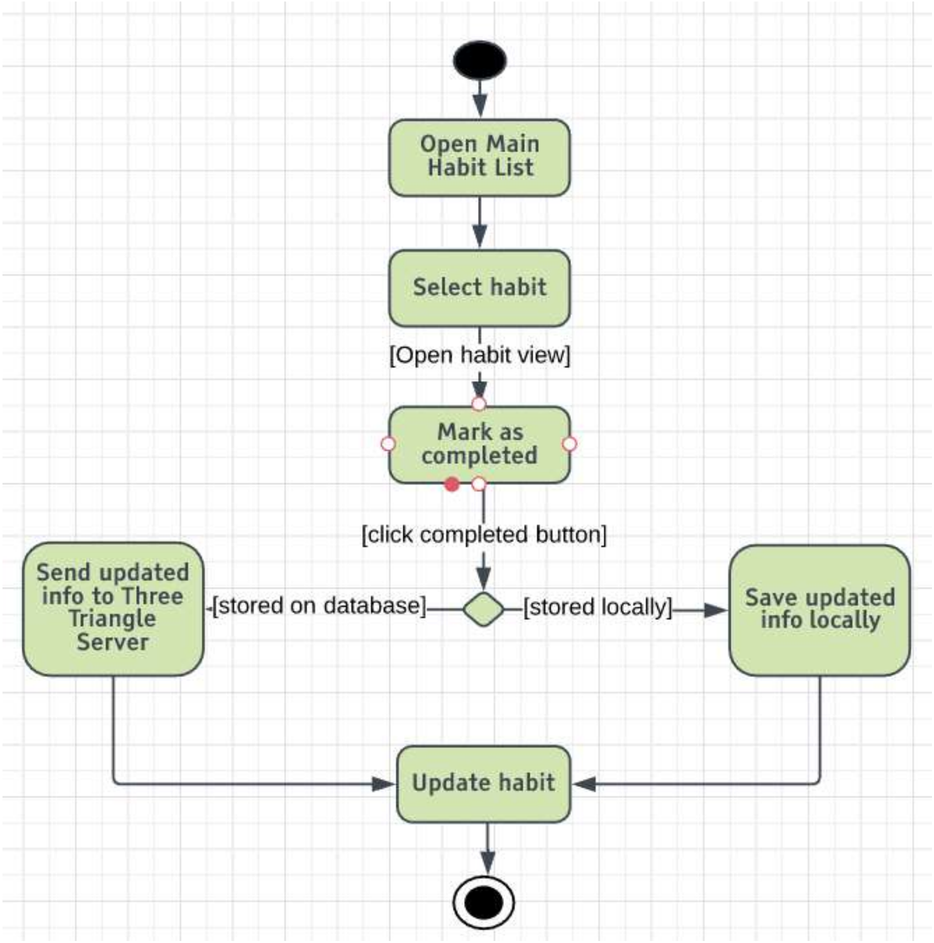
The third diagram models the activity of signing in to the account and verifying that account data is valid and exists in the database. It again uses either facebook, google, or our own account system and checks user data and either accepts or rejects a login attempt. On a successful login the user will be directed to the main activity after their data has been retrieved.

Activity Diagram 3



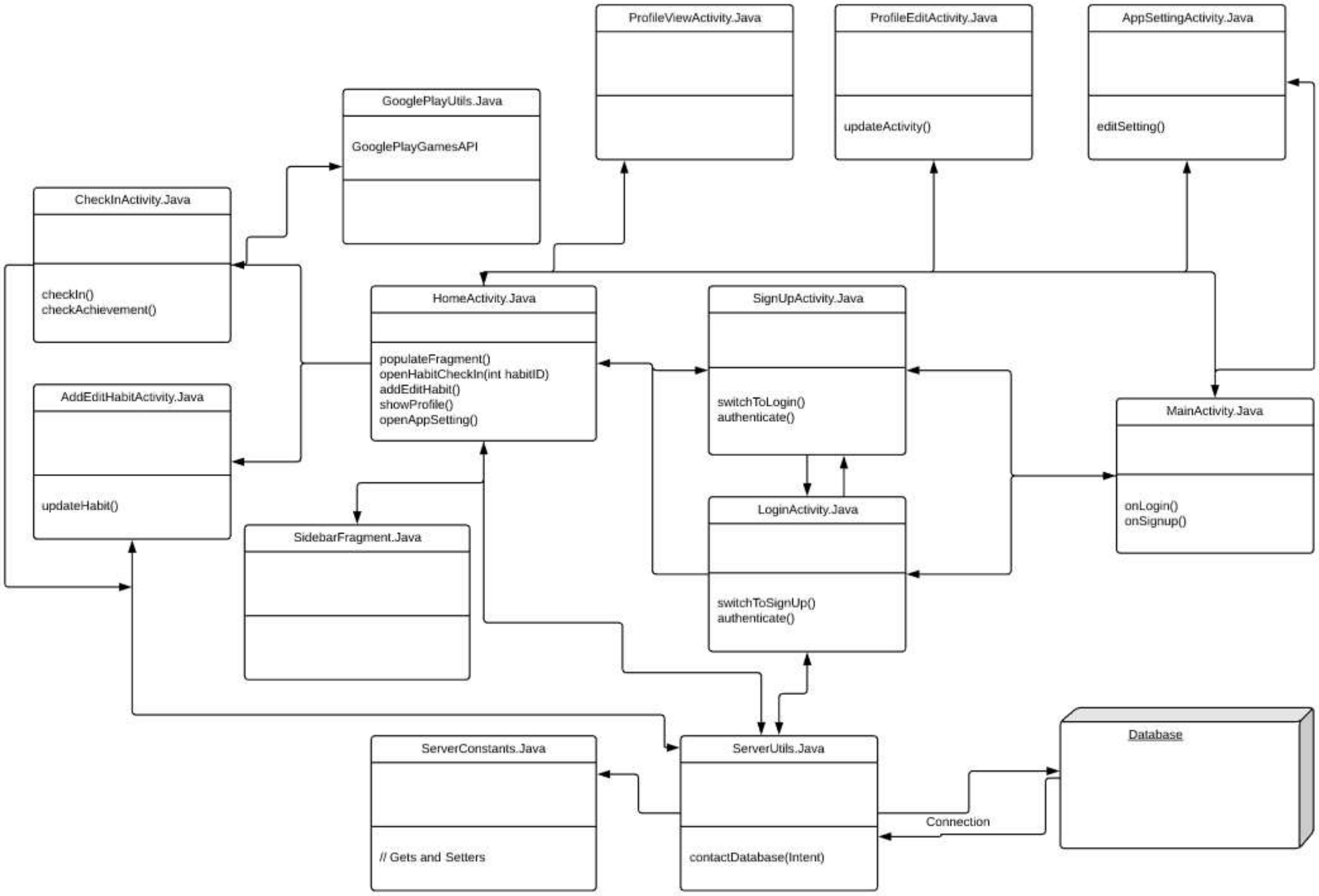
This diagram models the behavior of adding, editing, and removing a habit from the user's main habit list. When removing a habit, if the user is signed up, the app will remove the entry from the database. If the user is not signed up, however, the data is stored locally on the device and the data will be deleted from there. Adding and editing a habit both will take you to the same view in the app, except that editing a habit will populate the existing details into the view. Then the user will click the save button to either store the new habit or update the existing. Each path then requires the main habit list on the app to update.

Activity Diagram 4



This diagram models the behavior of marking a habit as completed from the user's main habit list. When looking at the main habit list, the user will select which habit they want to check in to. From there, they will click a button that marks the habit as complete for that day. If the user is logged in, the information will be sent to the Three Triangles database. If the user is not logged in, the information will be saved locally.

Class Diagram



Class Descriptions

This is a top down class diagram of the app. It incorporates the MVP of the design of the app. Some things like inheritance, methods, and trivial classes are omitted here. We will add these in with the refinement to the class diagram in the design document.

MainActivity.java - Starting point of the Application. All control flows begin through here.

SignUpActivity.java - The activity where users will sign up. Can also switch to the LoginActivity

LoginActivity.java - A user can login through this activity. Can switch to the SignUpActivity. Once logged in, the user moves to the HomeActivity.

HomeActivity.Java - Displays the home activity. This will have a list of habits to track.

SidebarFragment.Java - A sidebar that lets you get to the other sections of the app.

ServiceUtils.Java - Will contain a set of server commands for the rest of the app. Will be the only entity talking with the database.

ServerConstants.Java - A set of constants that will be used to create commands to the database.

AddEditHabitActivity.Java - Used to edit or add habits to the database

CheckInActivity.Java - Opens the prompt to check a habit off.

GooglePlayUtils.Java - A set of commands for talking to the Google Play Services

ProfileViewActivity.Java - An activity for allowing the user to view their own, or another user's profile

ProfileEdit.Java - Allows a user to update/ Edit their own profile

AppSettingsActivity.Java - An activity which allows the user to edit the various settings within the app.