# Title

Design Document for Chris Cota, Sully Cain, and Cameron Kurtz

# Problem Description

This app will be available on the app store for usage by those who wish to use it. You will be able to enter in your levels and calculate methods to get to another level. This will offer the most efficient method. We chose this project because it is a beneficial tool that would be useful to people who play the game RuneScape. The app will calculate experience for all 23 skills currently in the game.

# Requirements

In our app we wanted to show the different methods of training a various skill after looking up a user on the high scores. The requirements for our application are to offer support for the main skills (excluding combat) for training, as these work separately from the combat related skills. Combat is more of a challenge and as such we created a combat level calculator, as trying to tell a user how many times to attack something is not feasible. The reason this is so is due to the way that combat experience is gained. The experience a user gains is done per attack, and changes depending on the enemy. Experience also changes per damage done, so we decided it would not be possible to determine that and calculate easily for users.

# System Design

Our application has 3 main screens that are used throughout the app. We designed it this way to be as modular as possible due to the amount of information related to the amount of skills (23 skills in total). When you first load up the app, it comes to a main screen that has a text input available to type in a RuneScape character name. Upon clicking “Lookup”, the app will asynchronously search the RuneScape high scores for the entered username.

If the player is found, their stats will be populated below. These skills include: attack, strength, defense, ranged, prayer, magic, runecrafting, construction, hitpoints, agility, herblore, thieving, crafting, fletching, slayer, hunter, mining, smithing, fishing, cooking, firemaking, woodcutting, farming. All these skills start at 1 (besides HP which starts at 10 to prevent you from dying easily at the beginning of the game) and go to a maximum level of 99.

Attached to the skills is a navigator (for the skills not related to combat) that takes the user to a skill detail screen displaying the current experience that the user has in said skill, as well as a spot for a target “goal” to be set. Once a target has been input, the app will reference all the various methods of training for that skill and tell the user the number of actions that are required to go from the current experience to the target experience goal.

If the user taps on a skill tile that is related to combat (attack, strength, defense, ranged, magic, prayer, hitpoints) they will be taken to a different screen that will calculate their current combat level and tell them how many levels they need to gain in each of the respective combat styles to advance a combat level. These levels use a set of functions where your melee skills (attack, strength, defense) are more heavily weighted at later levels than magic and ranged combined.

# Implementation

The only library that we used was Native Base and that was for us to use icons. It can be used to create containers and different widgets within your app but we did not have time to fully implement that.

# Lessons learned

Throughout our app we had many problems that arose from nowhere, and we learned a few important lessons. One was the time constraint, even though we had most of the semester, it was not until near the end that our group felt comfortable tackling the entire project. Another lesson that we learned that ultimately determined that we should change from our original app idea to this one. Our ambitions were high but once we started digging into React Native we realized that our hopes of the original idea would not be very likely to complete within the time restriction.