Montana Pistell

N0145744 | mpistell@northislandcollege.ca

The Project: Kind Words

An iOS mobile application designed to encourage and motivate you through daily affirmations delivered either in-app, through notifications, or in a screen widget.

Final Project Summary

DGL 204 Prog. Mobile App Development

# Project Summary

## Project Overview

Kind Words is an iOS mobile application designed to encourage and motivate users through daily affirmations delivered either in-app, through notifications, or in a screen widget depending on the user’s preferences.

The app will consist of three screens: a Home screen, a Categories screen, and a Settings screen. The app will also offer widget functionality, which will include a home screen widget that displays a daily affirmation.

## Code Structure

The back end of this app was written in two playgrounds, containing code pertaining to the Home and Categories screens, and the Settings screen. The code in these playgrounds is responsible for the various data types, methods and functionality, and logic of the app. In a future iteration of this project, these files will interact with “Controller” code, connected to a front-end user interface, where the user can easily and intuitively navigate the features of the app.

## Swift Features Utilized

In this project I used a variety of Swift features, such as classes, enumerations, guard statements, functions, optionals, control flow features, protocols, and polymorphism. Self-documenting code was used as often as possible throughout the program, and a large focus was set on creating efficient code in terms of both time and memory, with the appropriate programming features being chosen for various needs in the program.

## Project Difficulties & Hurdles

The largest hurdles I encountered during this project were a lack of time, and missing knowledge pertaining to the front-end code. Although the scope of this project is to write a back-end only, I feel that once I have the knowledge to create a user interface and controller, the ‘larger picture’ perspective will help me write more efficient and organized back-end code.

In particular, a struggle I came across was having my code return a random affirmation from a specific category of affirmations. Using a function, the program will pass in a category type, and the overall collection of affirmations, and return a random affirmation that belongs to that category. When testing this code, I was running into crash. I added a guard to protect against returning an affirmation from an empty category, and noticed that my guard was always evaluating to empty, even though the category wasn’t. After some debugging, I noticed that I had used a ‘<’ instead of a ‘>’.