The Alpha Release

Harry Ellerm[1] and baspeden21[1]

[1]Affiliation not available

May 27, 2018

Contents

Challenges Faced

- Learning Swift. The concept of an "optional" was particularly weird to get used to at first.
- Learning how to use dictionaries and parse JSON responses from REST web services.
- Understanding the architectural features of an IOS application. For example, how to embed a view controller inside a Navigation controller so that you can smoothly transition to another view controller without an animation that looks like it's from the early 2000's.
- Learning how to control the flow of execution throughout the application. This was relevant when figuring out how to smoothly transition from the login to the main view of the application. This process was originally pretty poorly implemented, we refactored everything into a "LoginFlowWorker" that is called from the AppDelegate in order to refine this process further.
- Learning how to embed table-views (User Controller and HealthProfileController) and then customize these with custom table cells was particularly time-consuming.
- Figuring out how to use Scene-Kit for our model (still in process). Managed to get the main structure implemented.

- Using XCode has definitely presented some challenges, we have had numerous problems with it downright just crashing to projects being backwards but not forwards compatible.

- Still, despite everything, we are relatively happy with the progress made. We hope to refine the model next and then look to implement some automated unit testing via Travis.

xCode updated today....

macOS Installation Script

We have written a script that makes it easy to clone the repo with everything necessary to get up and running. It is designed to take your machine that already has Swift, Xcode and GIT installed up to the point where you can run the project. It was tested on a machine running macOS High Sierra v 10.13.4.

It uses:

- GIT and Homebrew (https://brew.sh/) to install GIT LFS (https://git-lfs.github.com/) if it is missing.

1. Create a new file in the folder you wish the repository to be cloned into:

```
$ nano installation_script
```

2. Copy and paste the following into the editor that appears and then save it [in nano: “ctrl x” to exit, then ‘y’, then enter]

```
#! bin/bash

echo "Checking if Homebrew package manager is installed..."
if ! [ -x "$(command -v brew)" ]; then
echo -n "Homebrew not installed, install? (y/n)? "
read response
if [ "$response" != "${response#[Yy]}" ]; then
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
else
exit 1
fi
fi
echo "Homebrew installed"
echo "Performing GIT LFS install if necessary..."
brew install git-lfs
echo "GIT LFS installed"
echo "Cloning repository & ignoring LFS files"
GIT_LFS_SKIP_SMUDGE=1 git clone https://github.com/HarrisonEllerm/COSC345_SoftwareEngProj.git
cd COSC345_SoftwareEngProj
echo "Setting up LFS Hooks in repo and pulling LFS files"
git lfs install && git lfs pull
cd ..
echo "Installation finished OK"
```

3. Turn the file into an executable by running the following command:

```
$ chmod 700 installation_script
```

4. Run the executable*:

```
$.  installation_script
```

* not ./installation_script as the script itself needs to change directories while executing.

5. You should be good to go, open Xcode and find the .xcworkspace file within the folder that was created when you cloned the repository. If at any point you wish to uninstall Homebrew (which will also uninstall GIT LFS) you can run the following command:

```
ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/uninstall)"
```