Spring Midterm Progress Report

Brandon Lee, Rutger Farry, Michael Lee

May 16, 2017

Abstract

The following report evaluates and recaps the purposes and project goals. The following will evaluate project progress, milestones, challenges, problems, proposed solutions, design, and remaining work. The following document will describe these aspects in the technical implementation level as well as in an overall higher level for clarity.

1 Introduction/Recap

As a quick recap, our project, C7FIT is an iOS health app built for the local Portland gym, Club Seven Fitness. This project is built in collaboration with eBay's mobile team located in Portland as well. The application's purpose is to allow customers of a gym to easily be able to access their health information already on their iPhones as well as integrate that data into something that will help them work with trainers at Club Seven Fitness. Our goal for this application was to build something that would be an effective tool to do such that.

2 Brandon Lee

2.1 Current Progress

As we delegated through various agile methods such as kan-ban boards, sprint meetings, and etcetera, we've decided to delegate the work based off of the tabbed views in the iOS application. Thus, I've been put in charge of the Store tab and the Profile tab. Since the last time we've put out a progress report, I've significantly cleaned up the profile screen's code In the process, I've fixed numerous bugs and minor UI issues as well. For example, the profile section's first cell was previously a mere table view cell. However, in refactoring the code for this, I've moved it over into being a UITableViewHeader instead. Additionally, I've started a UI code refactor process which makes the code a bit less verbose and more simplified as displayed here:

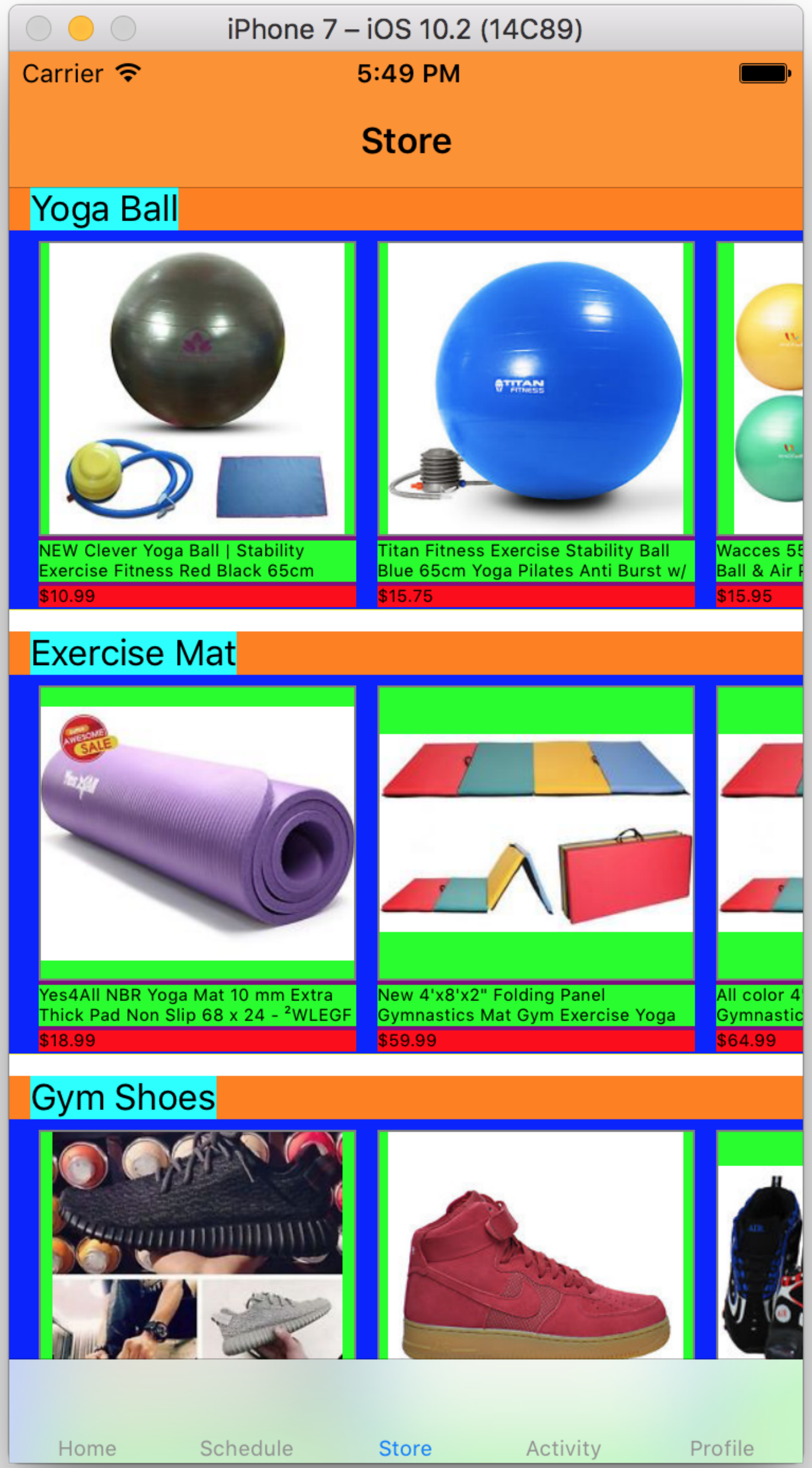
Original

```
scheduleLink.translatesAutoresizingMaskIntoConstraints = false
let linkLead = scheduleLink.leftAnchor.constraint(equalTo: leftAnchor, constant:0)
let linkTrail = scheduleLink.trailingAnchor.constraint(equalTo: trailingAnchor, constant:
let linkTop = scheduleLink.topAnchor.constraint(equalTo: topAnchor, constant: 0)
let linkBot = scheduleLink.bottomAnchor.constraint(equalTo: bottomAnchor, constant: 0)
NSLayoutConstraint.activate([linkLead, linkTrail, linkTop, linkBot])
```

Simplified

```
scheduleButton.translatesAutoresizingMaskIntoConstraints = false
NSLayoutConstraint.activate([
    scheduleButton.leftAnchor.constraint(equalTo: leftAnchor),
    scheduleButton.rightAnchor.constraint(equalTo: rightAnchor),
    scheduleButton.topAnchor.constraint(equalTo: topAnchor),
    scheduleButton.bottomAnchor.constraint(equalTo: bottomAnchor)
])
```

As you can see from the code snippet, the simplified version yields a much clearer, self documenting piece of declarative UI code. Keeping in terms of UI. We've cleaned up the previously set background colors for Autolayout anchoring constraint purposes. Additionally, we've changed the color scheme and added some app icons and default images for things such as profile picture and loading image. As a result the app now looks a lot nicer and professional. Here are some before/after screenshots of the UI:


iPhone 7 – iOS 10.2 (14C89)

Carrier 5:49 PM

Store

Yoga Ball

NEW Clever Yoga Ball | Stability Exercise Fitness Red Black 65cm
$10.99

Titan Fitness Exercise Stability Ball Blue 65cm Yoga Pilates Anti Burst w/
$15.75

Wacces 55
Ball & Air
$15.95

Exercise Mat

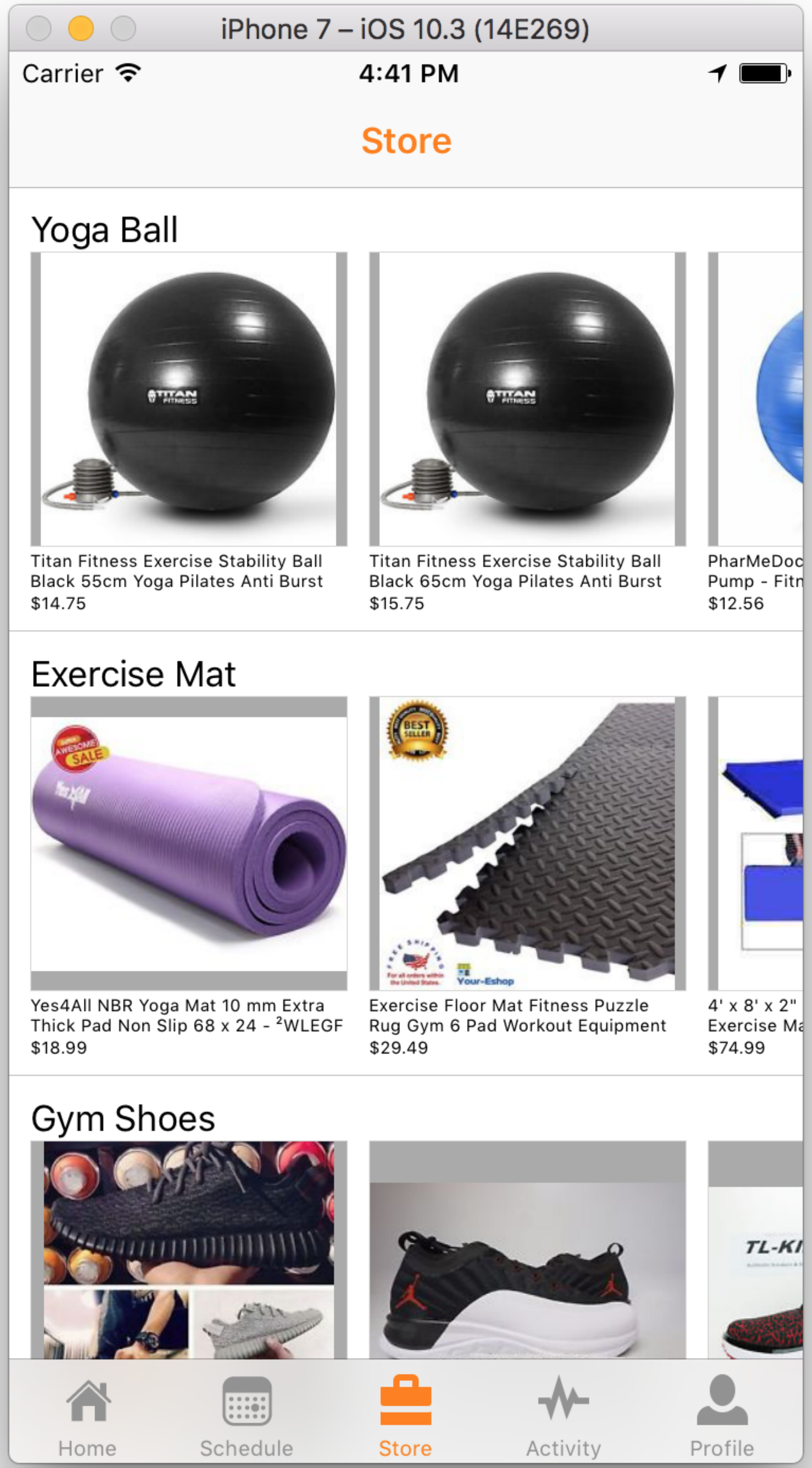
Yes4All NBR Yoga Mat 10 mm Extra Thick Pad Non Slip 68 x 24 - ²WLEGF
$18.99

New 4'x8'x2" Folding Panel Gymnastics Mat Gym Exercise Yoga
$59.99

All color 4
Gymnastic
$64.99

Gym Shoes

Before:



iPhone 7 – iOS 10.3 (14E269)

Carrier 4:41 PM

Store

Yoga Ball

Titan Fitness Exercise Stability Ball Black 55cm Yoga Pilates Anti Burst
$14.75

Titan Fitness Exercise Stability Ball Black 65cm Yoga Pilates Anti Burst
$15.75

PharMeDoc Pump - Fitn
$12.56

Exercise Mat

Yes4All NBR Yoga Mat 10 mm Extra Thick Pad Non Slip 68 x 24 - ²WLEGF
$18.99

Exercise Floor Mat Fitness Puzzle Rug Gym 6 Pad Workout Equipment
$29.49

4' x 8' x 2" Exercise Ma
$74.99

Gym Shoes

After:

As an overview recap of functionally developed for the profile screen, the following illustrates the functionality aspect of the code. The profile tab is essentially a profile page where a user is able to input various forms of health data including their name, bio, profile picture, height, weight, repetition records of various workout routines, mile time, and maxes of various workout routines. Essentially, after much trial and error in implementing our application, I was able to hook up a Firebase backend with the client side code to enable many core features such as a login/auth system, account creation, profile fetching and perform batch updating. In pursuit of this, I developed various classes such as a Firebase Data Manager, User model, and the respective profile views and view controllers. I developed the respective table view cell classes, setup Firebase on the server side, and developed the schema in which users would be interpreted. All of the following work has been merged from their respective feature branches, into our main development branch.

I have also developed the store screen. The store tab is essentially a tab where users can purchase various types of workout equipment from yoga balls to dumbbells. The store tab is geared towards using the eBay APIs and should deep link the listings on eBay. In order to implement this, much work has been done on developing request, response, and data manager classes for the eBay API as well as developing the client side code such as the view controllers, models, and views for the various listings we support. These items and listings are curated upon request of the client.

2.2 Remaining Work

Our initial requirements for the application have been essentially met. However, there can always be room for improvement in terms of minor details. Starting off with the major objectives. We want to deploy this application onto the App Store. In order to do so, we require to meet with our client eBay and Club Seven Fitness as well. This is to ensure everyone's on the right page of things before we go public. Firstly, we need to obtain all the correct assets from the gym before we ship. That way, we have all the actual data that would be relevant to the user. Secondly, we would need to set up some additional automated testing so that we will be able to find out what exactly goes wrong at a given point in time and target the issue in a quick manner. Secondly, we need to prepare the application as much as we can before meeting with the clients. This includes the retrieval of assets and adding them to our app. Finally, we also have the engineering expo to attend. This step should be fairly straightforward.

2.3 Impeding Problems

Our main impeding issue currently is the lack of assets. We are still waiting for some resources to be supplied by our client, hopefully before the engineering expo. Additionally, another issue is the remaining term schedule. As this is my final term, I have a lot of things to take care of before departing. As the midterm season begins to dawn, hopefully we will be able to integrate everything together in time. Finishing this project in completion remains a primary objective for me.

3 Rutger Farry

3.1 Current Progress

A lot has changed since the last progress report. We've delegated the remaining tasks using GitHub's built-in feature-tracking tools and are set to release C7FIT to the App Store as soon as we finish some minor tasks and get an Apple developer account for the Club Seven Fitness Gym. The purpose of the application is to allow clients of Club Seven Fitness in Portland, OR to stay on top of events at the gym and to allow trainers to better connect with said clients. Our app is feature-complete according to our requirements document, although after performing user tests and presenting our app to our sponsors at eBay, as well as the owner of Club Seven Fitness, we've decided that there are some tweaks that we'd still like to make to the app before it is published to the App Store.

The requested changes are mostly just numerous small UI tweaks that are too minor / specific to mention. There was also a request to host trainer biographies on the backend as well, which should be very easy to implement, considering nearly all of our app's data is loaded from Firebase already.

Since the last progress report, we've made a huge amount of progress, completing the Home, Schedule, Store, Activity and Profile tab to our specifications. We've done significant work on the backend and have even performed some optimizations to batch / reduce our number of API calls. User functionality is implemented - new users can sign in with an email account and returning users can access their data by logging in.

3.2 Impeding Problems

Since we have completed all of our written specifications, the biggest bottleneck to our development right now is communication with our clients at eBay and Club Seven Fitness. So far we haven't met the gym owner of Club Seven Fitness and have only communicated with him through our contacts at eBay. eBay intends to set up a meeting between our group and Club Seven Fitness soon though, and I look forward to seeing what our "effective clients" think about their new app.

Until then though, this communication bottleneck still exists. One downside is the slowness of feedback on UI changes. Since the gym owner is unable to build and run the application without installing the Apple iOS development toolchain, our clients at eBay need to build and install the app to show it to him. This results in a long feedback loop, meaning receiving feedback on incremental changes is impossible. Another downside has manifested in the quality of assets we've received so far. Some of the images we have received for display in the app are strangely colorized, pixelated, or otherwise unfit for releasing on the iOS App Store. Our partners at eBay are attempting to retrieve higher-quality assets, but this process takes awhile.

One final problem is perhaps code consistency. In our development model, we've sort of siloed out different parts of the app to different team members. This has resulted in different code styles in different parts of the application. While this probably won't be an impediment to future development, we believe this app could be a solid contribution to the open-source community, since there are not many full-featured Swift apps on the App Store with fully openly-available source code. Cleaning up code styles and keeping weird idioms to a minimum could enhance this open-source contribution.

3.3 Interesting Code

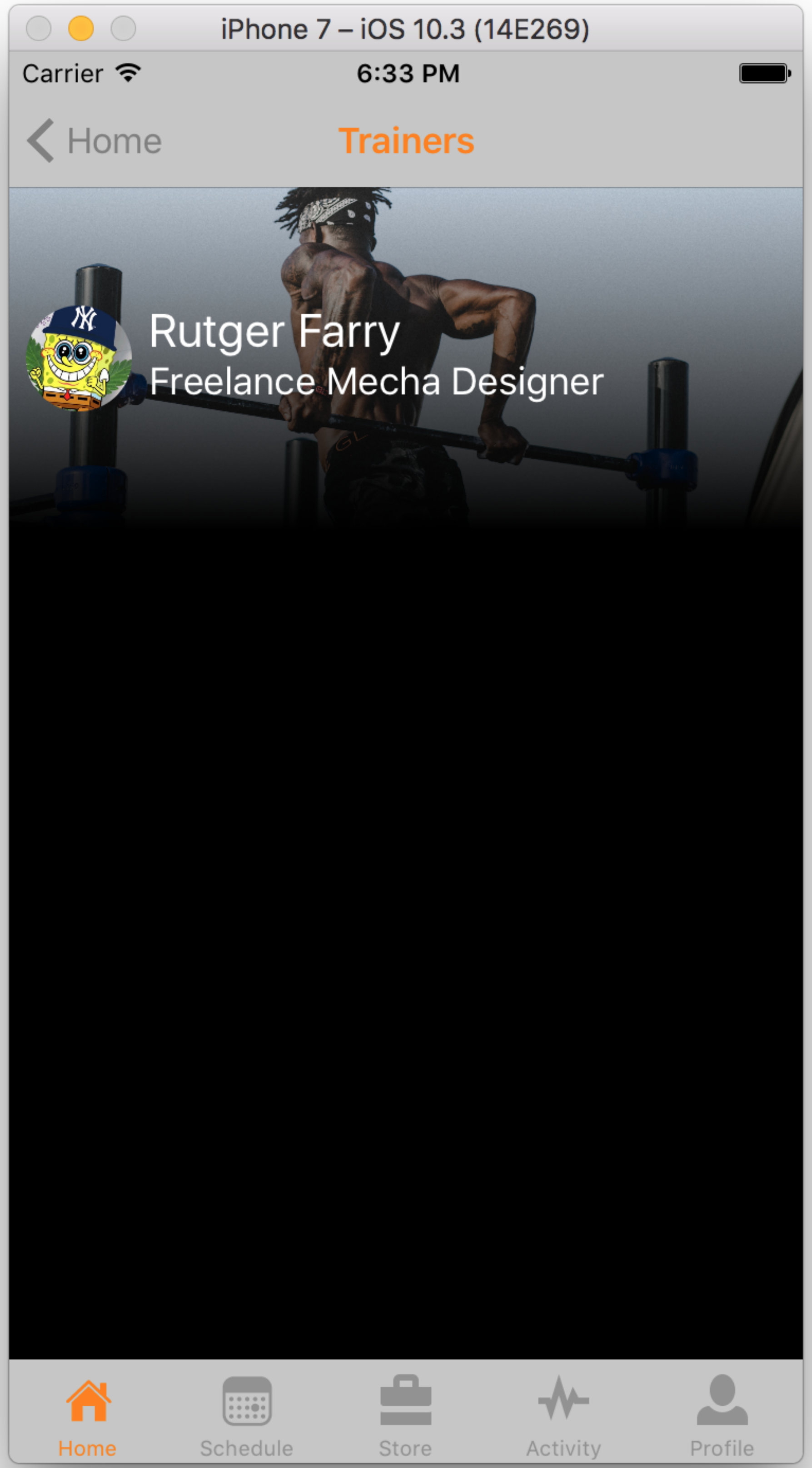
One of the interesting development goals of our project was to develop the entire UI in code. This is relatively unusual in iOS development, as Apple provides some very intuitive drag-and-drop tools (called Interface Builder) for developing UIs in macOS and iOS apps. Sadly, the ease-of-use of these tools comes with a drawback when working on large teams. Interface Builder stores view information in cryptic XML that is unwieldy to edit by hand. Therefore, whenever two people are working on the same interface file, huge merge conflicts arise that are very painful to resolve.

Since we are such a small team, we probably could have used the Interface Builder tools with relatively few issues, but we decided to be forward-thinking and design the app as though we were working on a much larger team - laying out all our interfaces in Swift using native UIKit constraints.

Below is an example of this code in use along with the view it creates:

```
func setupConstraints() {
        let margins = contentView.layoutMarginsGuide
        let avatarImageViewSize: CGFloat = 50

        // Use NSLayout constraints for all subviews
        nameLabel.translatesAutoresizingMaskIntoConstraints = false
        titleLabel.translatesAutoresizingMaskIntoConstraints = false
        avatarImageView.translatesAutoresizingMaskIntoConstraints = false
```

```
    // Position nameLabel to top right of avatarImageView
    nameLabel.bottomAnchor.constraint(equalTo: margins.centerYAnchor).isActive = true
    nameLabel.leadingAnchor.constraint(equalTo: avatarImageView.trailingAnchor,
                                       constant: 8.0).isActive = true

    // Position titleLabel to bottom right of avatarImageView
    titleLabel.topAnchor.constraint(equalTo: margins.centerYAnchor).isActive = true
    titleLabel.leadingAnchor.constraint(equalTo: avatarImageView.trailingAnchor,
                                        constant: 8.0).isActive = true

    // Position avatarImageView to center left of cell
    avatarImageView.layoutMargins = UIEdgeInsets(top: 8,
                                                 left: 8,
                                                 bottom: 8,
                                                 right: 8)
    avatarImageView.widthAnchor
        .constraint(equalToConstant: avatarImageViewSize).isActive = true
    avatarImageView.heightAnchor
        .constraint(equalToConstant: avatarImageViewSize).isActive = true
    avatarImageView.centerYAnchor
        .constraint(equalTo: margins.centerYAnchor).isActive = true
    avatarImageView.leadingAnchor
        .constraint(equalTo: margins.leadingAnchor).isActive = true

    // Make avatarImageView circular
    avatarImageView.layer.cornerRadius = avatarImageViewSize / 2
    avatarImageView.clipsToBounds = true
    avatarImageView.layer.masksToBounds = true
    avatarImageView.contentMode = .scaleAspectFill
}
```


iPhone 7 – iOS 10.3 (14E269)
Carrier
6:33 PM
Home
Trainers
Rutger Farry
Freelance Mecha Designer
Home
Schedule
Store
Activity
Profile

3.4 Summary

Overall I think we've done a good job so far and haven't made any serious mistakes. With a few minor tweaks, we will be ready for deployment to the App Store, which hopefully will be relatively hiccup-free (Apple reviewers can sometimes be picky). I'm looking forward to sharing this app with our client and hopefully giving Club Seven Fitness gym-goers the ability to track their workouts and stay motivated.

4 Michael Lee

4.1 Current Progress

The purpose of our project remains the same at the end of the term, our main goal of creating an iOS application for the gym Club Seven Fitness still holds. While there are many fitness applications that do what ours does, our application is personalized to our clients: Club Seven Gym and eBay. We do this by incorporating the gym's website and the eBay API store within our application. As the code freeze has already been implemented, our application fulfills all of the requirements that we outlined at the beginning of this project; however, in meeting with our client they had some pieces of advice and requests for additional changes that we need to make before we submit our application to the Apple store. Our project purpose has shifted from creating a minimum viable product to finalizing all the details and getting the project ready for the general public. Next I will highlight some of the more important portions of my code within the Activity Tracker.

4.2 Current Code

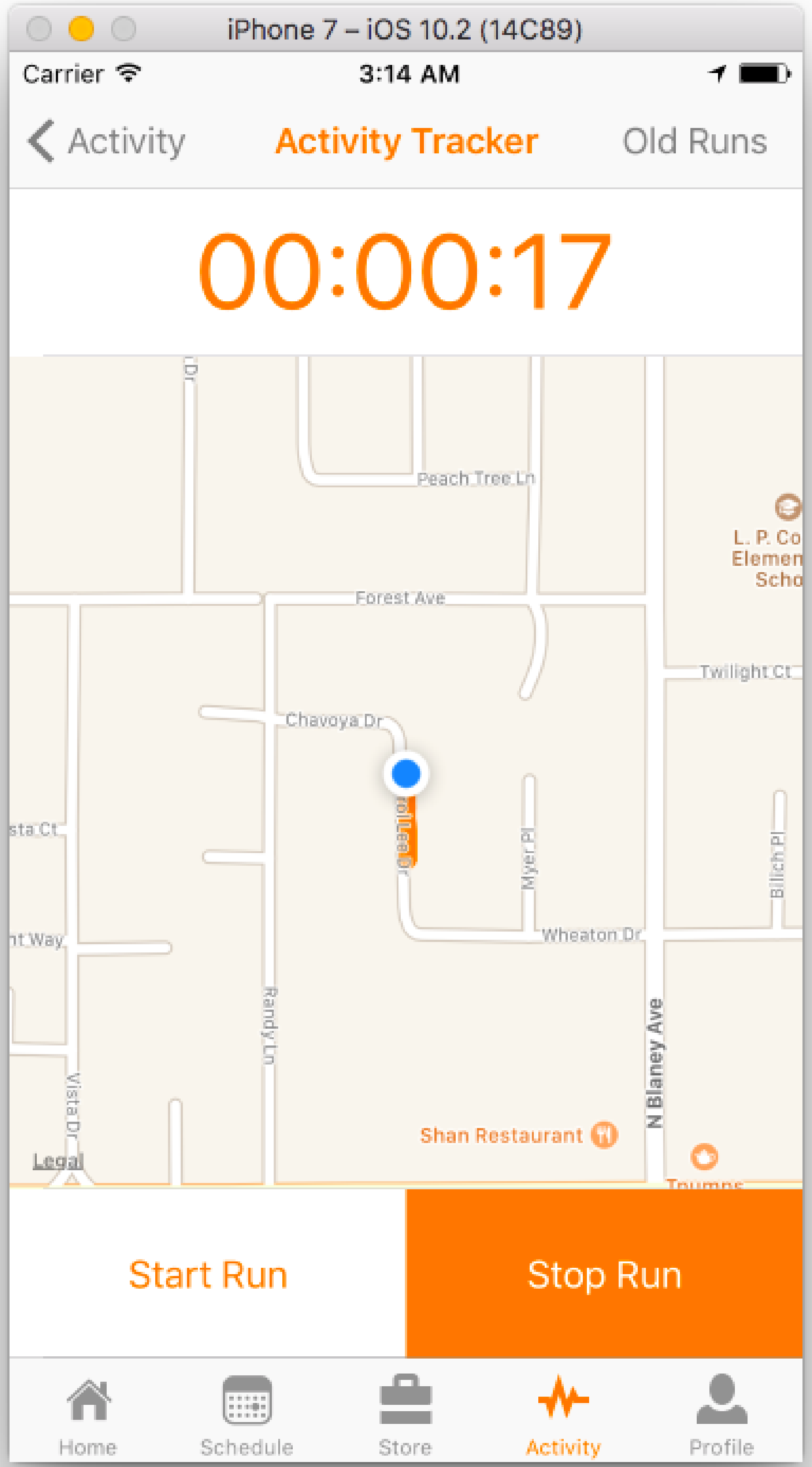
The following code snippet is responsible for the displaying all of the user's stored runs in Firebase. The first two lines make sure that the user is logged in before it pulls any data. If the user is not logged in, it will display a login pop-up with LoginViewController(). If the user is logged in it will obtain their userID and search our Firebase DB for all of their stored runs and save them to a local variable. Then it will reload the table's data source so they are displayed on screen.

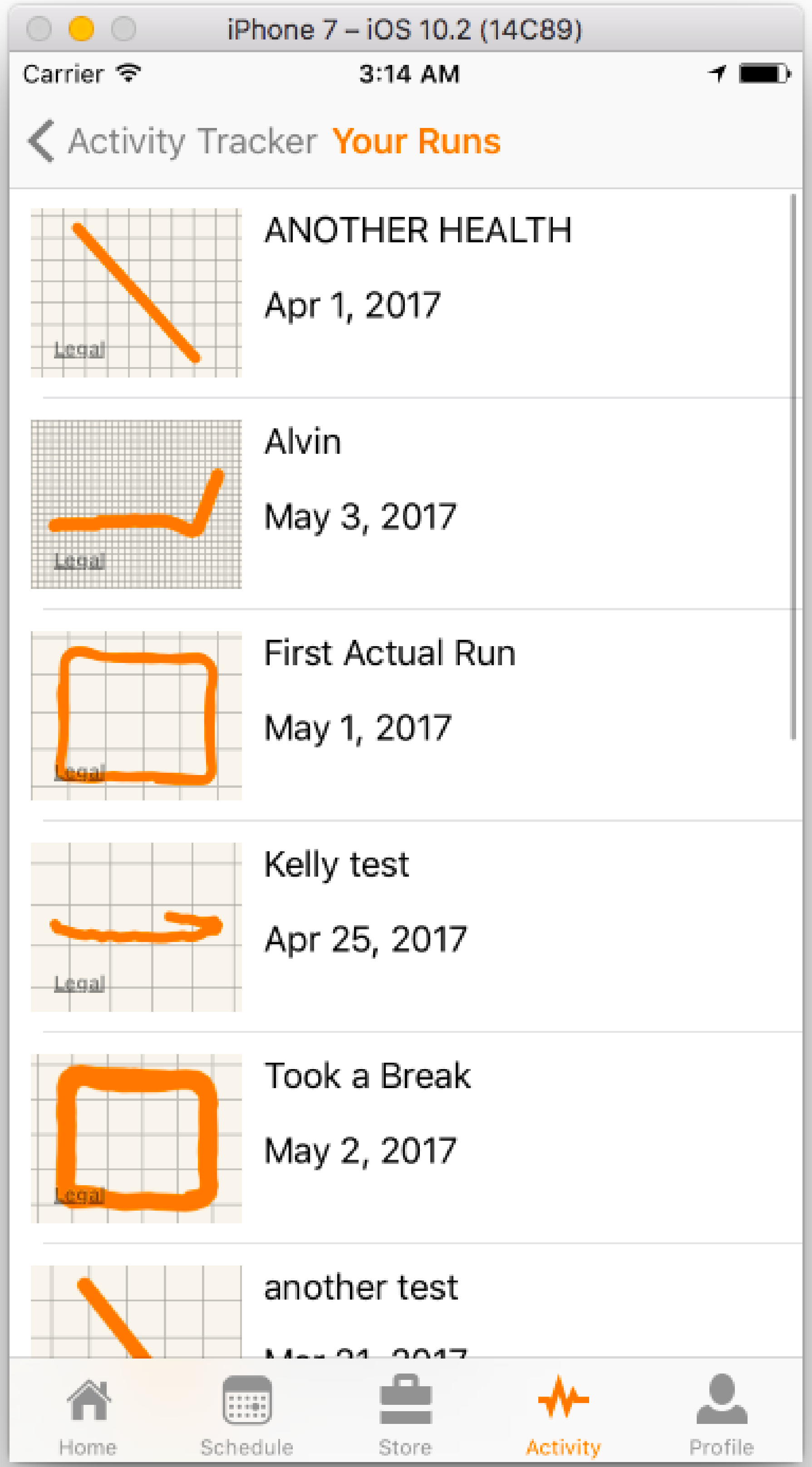
The second function initializes each of the cells in the table. They each get assigned to display a different run based on their row number. There is a blank RunListCell that gets fed the data (title, date, locations, time, etc.) to create the detailed run cells that you see in the second screen shot. Each of these cells has then their own personal mapView and MKPolyline overlay initialized and drawn.

```
firebaseDataManager.monitorLoginState { _, user in
            guard let userID = user?.uid else { return self.present(LoginViewController(),
             animated: true, completion: nil) }
            self.firebaseDataManager.fetchUserRunList(uid: userID) { data in
                self.numRows = Int(data.childrenCount)
                for run in data.children.allObjects as! [FIRDataSnapshot] {
                    let json = run.value as? [String:AnyObject]
                    let tempRun = self.firebaseDataManager.buildRunFromJson(json: json!)
                    self.listRuns.append(tempRun!)
                }
                self.tableView.reloadData()
            }
        }

override func tableView(_ tableView: UITableView,
cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    if indexPath.row < listRuns.count {
          if let runData = listRuns[indexPath.row] {
              if let cell: RunListCell =
              tableView.dequeueReusableCell(withIdentifier: runListID,
                for: indexPath) as? RunListCell {
                      cell.titleLabel.text = runData.runTitle
                     cell.valLabel.text = runData.dispDatePretty()
                   cell.mapView.isHidden = false
                        cell.mapView.delegate = self
                    cell.mapView.region = mapRegion(currentRun: runData)
                       cell.mapView.add(polyline(currentRun: runData),
                  level: MKOverlayLevel.aboveRoads)
                       return cell
                }
            }
        }
        return UITableViewCell()
}
```

The following code snippet shows the integration of the phone's GPS with the visual mapping of the user's run. On the following screenshot (first) you can see the user's current location begin shown, along with a orange vertical line denoting where they've just ran. The code below overrides the default locationManager(...) to perform a more specified purpose. Within the function on the first three lines, we can see that a region is calculated with the user at the center. This is used to center the map around the user within a certain distance. The next if statement determines when we record the user's location, in this case it will only record after the "Start Run" button has been pressed. Some error handling is performed as we want our information to be accurate, and the locations are stored in two locations: self.location and self.currentPath. currentPath is used in the following lines to draw the user's path as they run allowing for improved tracking and feedback.

```
func locationManager(_ manager: CLLocationManager, didUpdateLocations locations: [CLLocation]) {
    for location in locations {
        // Focus on the runner
        let center = CLLocationCoordinate2D(latitude: location.coordinate.latitude,
                                            longitude: location.coordinate.longitude
        let region = MKCoordinateRegion(center: center,
                        span: MKCoordinateSpan(latitudeDelta: 0.005, longitudeDelta: 0.005))
        self.mapCell.mapView.setRegion(region, animated: true)
        if self.recordUserLoc == 1 {
            // Dont record location if accuracy too low
            if location.horizontalAccuracy < 20 {
             if self.locations.count > 0 {
                  distance += location.distance(from: self.locations.last!)
               }
               self.locations.append(location)
                self.currentPath.append(CLLocationCoordinate2D(latitude: location.coordinate.latit
                                                           longitude: location.coordinate.longitud

                 if seconds > 1 {
                      self.mapCell.mapView.add(polyline(coords: self.currentPath),
                                                     level: MKOverlayLevel.aboveRoads)
                 }
            } else {
                 print("loc accuracy too low")
            }
```


iPhone 7 – iOS 10.2 (14C89)
Carrier
3:14 AM
Activity
Activity Tracker
Old Runs
00:00:17
Peach Tree Ln
Forest Ave
Twilight Ct
Chavoya Dr
Myer Pl
Wheaton Dr
Randy Ln
N Blaney Ave
Vista Dr
Bilich Pl
Shan Restaurant
Legal
Start Run
Stop Run
Home
Schedule
Store
Activity
Profile



iPhone 7 – iOS 10.2 (14C89)

Carrier 3:14 AM

Activity Tracker **Your Runs**

ANOTHER HEALTH
Apr 1, 2017

Alvin
May 3, 2017

First Actual Run
May 1, 2017

Kelly test
Apr 25, 2017

Took a Break
May 2, 2017

another test

4.3 Remaining Work

There is not too much work left to do, as I've noted before we have already fulfilled the requirements that were initially outlined for us. For my part, the client had some suggestions towards improving the heart rate calculator screen. This would involve adding a count-down clock or other visual in the empty area of the screen to help the user. Other than that there is not much work left to do on the actual application. We plan on meeting with the actual client, not eBay but the gym owner, in order to get his feedback on the current state of the application. If he is happy with it, then the remaining work would be dedicated to getting the application ready for the App store and that would complete the final goal and purpose of the application.

4.4 Impeding Problems

There weren't too may notable problems with our application. I did have to work harder to learn iOS development and swift as I had no prior experience. There were some issues in getting professional assets from our client, as the gym owner used his personal photos and nothing particularly official. Decisions in implementation were generally left up to us, so it was harder to know if they would be happy with the choices that we made. Throughout the process we would ask them for advice and review to make sure that they approved of our design choices. I would like to communicate more with the gym client as the application is, in the end, for him and his gym. His feedback would have been helpful throughout the development process, but because we mainly interacted with eBay our assigned client we missed out on this feedback.