

Live Fit Mobile Fitness Trainer App

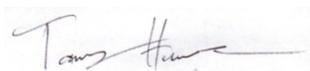
by

Tony Hume, Chris Hinds, Kenny Nissel and Jeremy Yap

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

© Copyright 2014 Tony Hume, Chris Hinds, Kenny Nissel and Jeremy Yap

The authors grants to the School of Information Technology permission
to reproduce and distribute copies of this document in whole or in part.



Tony Hume

4/15/2014

Date



Chris Hinds

4/15/2014

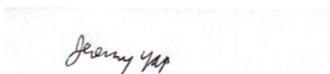
Date



Kenny Nissel

4/15/2014

Date



Jeremy Yap

4/15/2014

Date

Russell E. McMahon

Russell McMahon, Faculty Advisor

4/17/2014

Date

University of Cincinnati
College of
Education, Criminal Justice, and Human Services

April 2014

Acknowledgments

For their guidance and assistance in this project the development team of Chris Hinds, Tony Hume, Kenny Nissel and Jeremy Yap would like to thank all of the UC staff, fellow developers and personal trainers who assisted with the development of the Live Fit mobile application. Specifically, the team would like to thank Patrick Striet and Holly Mitchell for giving us the opportunity to work with them to produce this product. The team would also like to thank James Scott for introducing Patrick and Holly to us. For providing guidance and introducing us to new technologies and techniques used in the field the team would like to thank Zack Martinsek; Zack was introduced to us by Bill Mote who also advised us on the direction of our application. Our thanks go to Russell McMahon as the project's advisor; he assisted us in deciding the technologies that are used in development and helped manage the team's efforts. And thanks to Patrick Kumpf for guiding us in producing our deliverables.

Table of Contents

Acknowledgements.....	1
Table of Contents.....	2
List of Figures.....	3
List of Tables.....	4
Abstract.....	5
1. Introduction.....	6
1.1. Project Description.....	6
1.2. Project Statement.....	6
1.3. Solution.....	7
1.4. User Profiles.....	8
1.4.1. Personal Trainers.....	8
2. Design Protocol.....	9
2.1. Design Objectives.....	9
2.2. Methodology/Technical Approach.....	9
2.3. Research.....	9
2.4. Technical Elements.....	9
2.4.1. Titanium/Alloy.....	9
2.4.2. Database.....	10
2.4.2.1. Database Sync between Local and Server.....	11
2.4.3. Windows Azure.....	11
2.5. User Interface.....	12
3. Deliverables.....	12
3.1. iOS App.....	12
3.2. Android App.....	12
4. Project Planning.....	13
4.1. Project Timeline.....	13
4.2. Project Software.....	13
4.3. Project Budget.....	14
5. Testing.....	14
5.1 Emulator.....	14
5.2. Hardware.....	15
6. Conclusion.....	15
Sources.....	16

Table of Figures

Figure	Page
Figure 1: Use Case Diagram.....	8
Figure 2: Database Diagram	10
Figure 3: Gantt Chart	13

List of Tables

Table	Page
Table 1: Budget.....	14

Abstract

Many Independent Personal Trainers are using spreadsheets, pen & paper or various other methods to organize, track and invoice their clients. Holly Mitchell and Patrick Streit, owners of Live Fit Cincinnati, commissioned a design to create a mobile app that would specifically target independent personal trainers. This application will allow them to add, modify and track their clients from both a training perspective and a business perspective. Although there are other mobile applications that allow trainers to track their clients, there are none that allow you to generate invoices, track sessions, payments and store detailed client information. There is desktop software that has this functionality but this software requires the trainers to either take notes and then enter it in at the end of the day, or try to remember and enter from memory. This mobile app allows the entering of that data. Another feature that is lacking among these apps is a way to easily share appointments and gym space with other trainers who may be using the same gym space to train their respective clients.

1. Introduction

This report was created to provide an in depth explanation of the process used to develop the Live Fit personal trainer client management mobile application. The team of Tony Hume, Chris Hinds, Kenny Nissel and Jeremy Yap will explain the design and development of the application. Development of this system made use of a cross platform mobile development framework in conjunction with a server-side API connected to a “cloud” database, all of which will be detailed in this report.

1.1. Project Description

Personal trainers provide their clients with guidance towards a healthier and happier way of living. In order to do this the trainer needs to retain information about their clients’ progress and appointments. Often the trainers resort to paper and pencil solutions to record this information. This becomes increasingly more difficult for trainers working in gyms with shared workspaces. Management of clients is difficult for private personal trainers, as a solution to this issue the team has designed a mobile application to manage client information, session scheduling and invoicing of completed sessions. Unlike similar mobile applications that are on the market the Live Fit mobile application focuses on sharing scheduled training sessions allowing multiple trainers to better coordinate usage of shared space.

1.2. Problem Statement

Live Fit is a local gym where multiple fitness trainers use rooms in the gym to conduct sessions with their clients. The trainers have been using paper documents and Excel spreadsheets to record client information and to calculate payment for sessions. This process can be messy and

unorganized. The process of recording information is not the same for all trainers. When these trainers share gym space scheduling becomes difficult.

The owners of the gym would like a solution to these issues in the form of a mobile phone application that includes a simple data entry interface with support of a server-side database on iOS and Android devices. While the owners of the gym want to use this system at their own gym, they would like the system to be flexible enough for other trainers and gym owners to use. In order to assist in scheduling sessions in a shared space, the gym owners want a shared calendar of appointments.

The final objective is to figure out the best way to develop the application itself. The team has researched cross-platform frameworks that have allowed us to develop both iOS and Android apps at the same time. Because these frameworks use HTML, CSS, and JavaScript the team has looked into web based technologies to make the app responsive for various screen sizes.

1.3. Solution

The team has developed a mobile application that intuitively stores all needed information for later use as well as invoices clients after completed sessions. The front end of this application has been developed using the cross-platform framework Titanium Alloy. Titanium is a JavaScript framework which gives the ability to create native applications using JavaScript instead of the platforms native language. Using this cross-platform technology allowed for rapid development on multiple mobile operating system platforms with the same code base. The information is stored on the local device's SQLite database until the local data is synced to a remote, server-side SQL database for secure backup storage and later use. This syncing is done through the use of a restful API service that the team developed.

1.4. User Profile

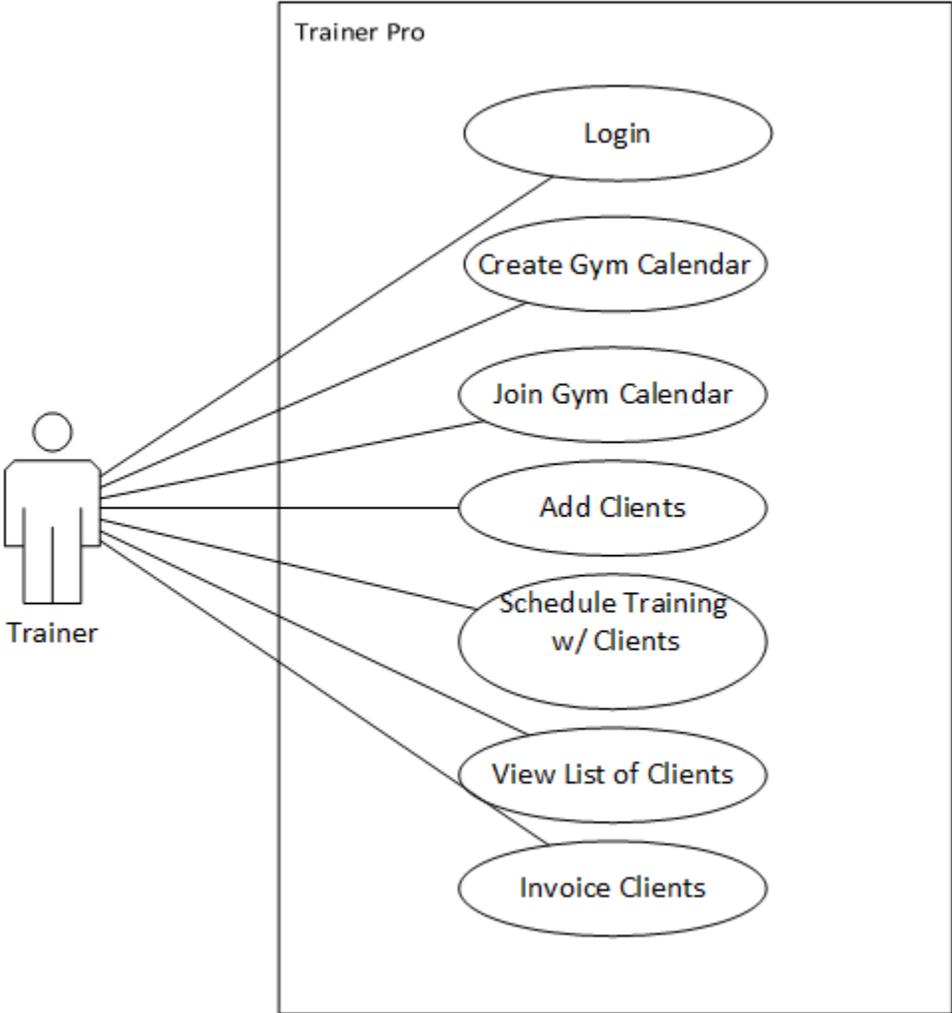


Fig. 1 User Profile

1.4.1. Personal Trainers

The core users of this app would be personal trainers. The age of a trainer can vary from their twenties to their seventies. It is not expected for them to be very familiar with mobile technology. It is expected that the independent personal trainers have mobile devices with a basic understanding of how to navigate mobile applications.

2. Design Protocol

2.1. Design Objectives

The goal for this mobile application was to make a reliable storage platform for client information with an intuitive user interface for easy data entry and reading of information.

2.2. Methodology/Technical Approach

As opposed to developing for multiple mobile device platforms the team opted for a mobile development framework that would produce applications on all targeted platforms. This also allowed the team to develop user interfaces of the application using web-based technologies.

2.3. Research

In research of a mobile development framework to use the team compared Apache Cordova and Appcelerator Titanium. Due to the lack of documentation of Apache Cordova and the fact that Appcelerator Titanium makes use of the native controls of the mobile OS, the team felt Appcelerator Titanium would provide a better experience for the user and would be less problematic to develop for.

2.4. Technical Elements

2.4.1. Titanium Alloy

Appcelerator Titanium allows the team to develop an app that can be used on multiple mobile device platforms. Titanium uses a JavaScript runtime and utilizes API of various mobile platforms. Using this, developers can reuse code for apps across different platforms.

2.4.2. Database

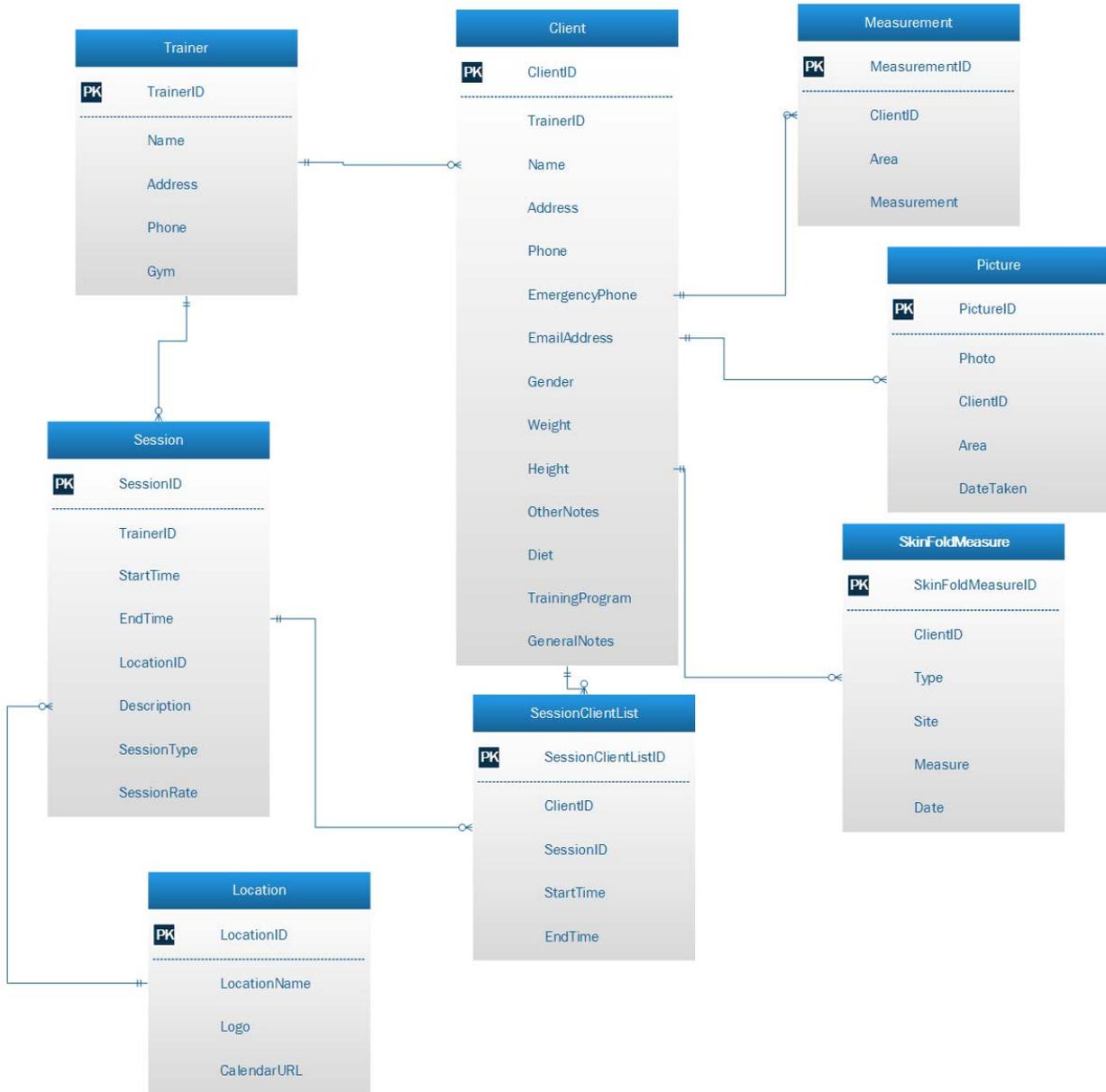


Fig. 2 Database Diagram

A concern that the stakeholders had was that their gym did not have Wi-Fi Internet access. This would leave the trainer unable to update their client's information to the web server. During times where the user is unable to send their information to the web server, the

information will be stored on the local mobile device's database. The mobile application then makes use of the SQLite database to store that information (Kreibich 187). The SQLite database structure has been designed to mimic the structure of the SQL database. The only difference being the date that is added to the records in the databases. These date markers allow the system to tell when a record was last updated on the server-side SQL database and when the record was last synced on the local SQLite database. This allows the system to determine what to update when the time to sync data comes around. This will also give the application the flexibility to be used on multiple devices by one user.

2.4.2.1. Database Sync between Local and Server

A challenge the team faced in development was how to achieve storage offline and keep saved data on the server-side database. On initially loading the app the system will do a full sync of the two databases. The local database will store the data of the app until this sync happens. In the event that the trainer is running the application without a data connection the device will store that information until a data connection is detected.

When a data connection is detected the system will do a sync of all information associated to that Trainer. As a means of security, the API calls need to have a unique identifier for the trainer passed in to the API call. This allows for the JSON packets that are returned to include information specific to that trainer.

2.4.3. Windows Azure

In order to host the SQL database, develop API calls as well as build a companion web interface to the app the team turned to the Windows Azure platform. Windows Azure provided a useful Web-based interface to easily develop the SQL database on. The API calls were part of a

code project that was held on the Azure server. And in development of the API calls and using the SQL database tables, Azure provided an easy means of creating a web interface with only small amounts of customization by the team.

2.5. User Interface

As the size of the mobile device can vary greatly, the user interface of the app was designed to be easy to use and flexible for all different sized screens. As our user base is not expected to have thorough experience with complex interfaces. As such, the app was designed so that the user interface would be easily accessible. This was done by breaking up our sections of user input into small groups. The team wanted to reduce the need for unnecessary scrolling or small text.

3. Deliverables

3.1. iOS App

As the stakeholders explained to the team, the majority of trainers at their gym use iOS devices. That made the iOS application a priority for the team. The mobile application is supported by the latest version of the mobile OS, iOS 7.

3.2. Android App

Though the majority of the current trainer's use iOS devices, an Android version of the mobile application was also developed as well, since Android is a widely used mobile application platform.

4. Project Planning

4.1. Project Timeline

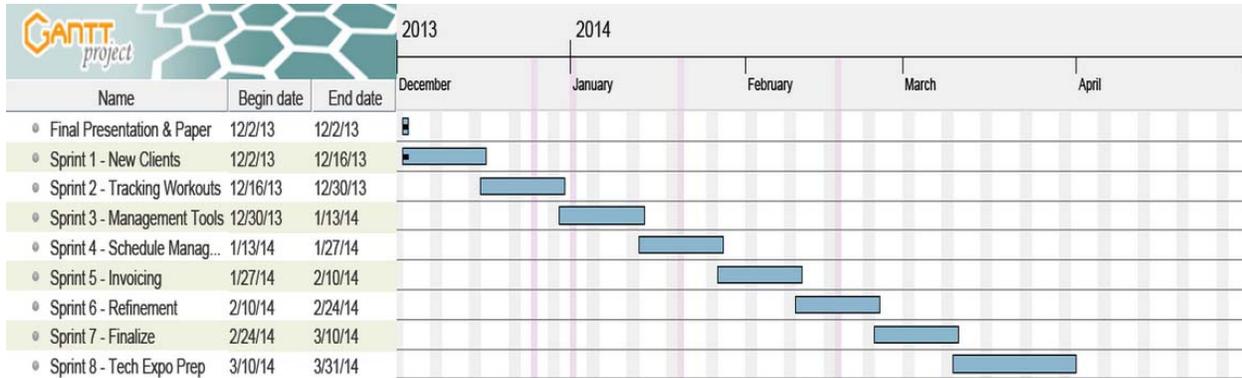


Fig. 3 Gantt Chart

4.2. Project Software

The Titanium platform uses a modified version of the Eclipse IDE to develop on. The team used this software on both Windows and iOS machines as the iOS platform cannot be developed on a Windows operating system. For both Android and iOS development the Titanium development application makes use of emulators to do debugging.

The team used Visual Studio 2013 to create a separate code project for the Web API calls. This project used entity framework to map out the database tables into models that can be easily be parsed into JSON objects.

4.3. Project Budget

Item	Estimate Cost	Actual Cost
Visual Studio express	\$0	\$0
Eclipse	\$0	\$0
Azure platform	\$0-\$1,000**	\$0
Apple Developer License	\$99*	\$99
Android Developer License	\$25*	\$25
iPhone 5s	\$649	\$0
Nexus 5	\$350	\$0
Titanium Platform	\$0	\$0
SQL Server Express	\$0	\$0
Labor Cost	\$45***	\$0
Total	\$44,368	\$124

* - Per year

** - Per month

*** - Per Hour

5. Testing

5.1. Emulator

During initial development, the team relied on software emulators of the Android and iOS operating systems to test the mobile application. As the Android emulator that comes with the Android SDK can be slow to run and possibly unreliable to test on, the team quickly moved on to testing on actual devices.

5.2. Hardware

As development proceeded, the team turned to actual devices to test on. This was mainly on various Android devices including the Samsung Galaxy S4. These devices were of varying screen size to test out the visual elements of the design. After obtaining an Apple Developers License the team also did testing on an iPhone5.

6. Conclusion

This mobile application enables independent personal trainers to better manage clients than any other application currently on the market. The unique combination of mobile portability, intuitive interface, and business functions such as shared calendar scheduling and automatic invoicing make this application the best personal trainer management system option. The project team is grateful to have been able to work on this project and take pride in the final product.

Sources

- Bergman, Neil. *Hacking Exposed: Mobile Security Secrets & Solutions*. New York: McGraw-Hill, 2013. Print.
- Bibeault, Bear. *JQuery in Action: Second Edition*. Stamford: Manning, 2010. Print.
- Coenraets, Christophe. "Tutorial: Developing a PhoneGap Application." *Christophe Coenraets*. N.p., n.d. Web. 07 Oct. 2013.
- Gliser, Shane. *Creating Mobile Apps with JQuery Mobile*. Birmingham, U.K.: Packt Limited, 2013. Print.
- Hay, Stephen. *Responsive Design Workflow*. San Francisco, CA: New Riders, 2013. Print.
- Kreibich, Jay A. *Using SQLite*. Sebastopol, CA: O'Reilly, 2010. Print.
- McWherter, Jeff, and Scott Gowell. *Professional Mobile Application Development*. Indianapolis: Wiley, 2012. Print.
- Shotts, Kerri. *PhoneGap 2.x Mobile Application Development Hotshot: Creating Exciting Apps for Mobile Devices Using PhoneGap*. Birmingham [u.a.: Packt Publ., 2013. Print.
- "Using Titanium HttpClient to Access Windows Azure Mobile Services REST API." *Appcelerator Developer Center*. N.p., n.d. Web. 15 Oct. 2013.
- Walther, Stephen. "Using Durandal to Create Single Page Apps." *Stephen Walther*. N.p., 08 Feb. 2013. Web. 07 Oct. 2013.