PiSonal Trainer: Weight Lifting Performance Tracker
Detailed Design

Birunthaa Umamahesan Micaela Estabillo Simarpreet Singh

April 1, 2017

Prepared for Computer Science 4ZP6: Capstone Project
Instructor: Dr. Wenbo He Fall/Winter 2016-2017

Contents

List of Figures

List of Tables

Revision History

Date	Version	Primary Author	Comment
04/01/2017	1.00	Birunthaa Umamahesan	Final editing and proofreading for revision 1
04/01/2017	1.00	Micaela Estabillo	Adjust technical specifications based on changes
01/10/2017	0.00	Micaela Estabillo	Final editing and proofreading for revision 0
01/09/2017	0.00	Birunthaa Umamahesan	Update user Interface Elements Descriptions
01/09/2017	0.00	Simarpreet Singh	Update Module decomposition
01/09/2017	0.00	Micaela Estabillo	Initial skeleton version

Table 1: Revision history

1 Templates, Symbols and Conventions Used

The modules' functions are specified in the format shown in Table 2.

`nameOfFunction()`	
Input:	List of input variables or actions
Output:	List of updated variables or system's response
Description:	Description of what the function does

Table 2: Function specification format

2 User Interface Elements Descriptions

Pisonal trainer's user interface (UI) design is presented in this section. Specifically, the UI's navigation flow and the major UI elements are described. Each element is explained with the support of Norman's design principles and illustrated with screen images from a mockup.

2.1 Navigation Flow

Refer to Figure 1. Users are directed to login page. From there existing users can login, and new users can registers for an account and then return back to the login page. After logging successfully, the user will land on the explorer page. From the profile page, the user can access the navigation bar locate to the progress, log, and settings page.

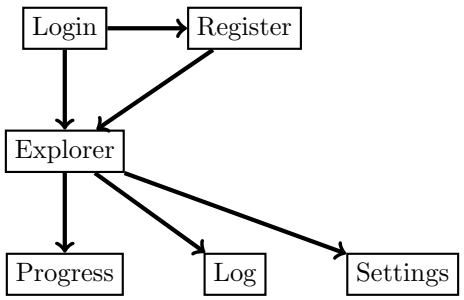

Figure 1: Navigation Flow diagram

2.2 Norman's Design Principles

Don Norman lists six principles to support software usability. These principles are affordance, constraints, conceptual model, feedback, mapping, and visibility[1]. The UI decisions for PiSonal trainer comprises of these design principles which are briefly explained below.

- Affordance is the attribute or control that helps the user to determine how it can be used[1].
- Constraints are the limits of an object or control[1].
- Conceptual model is a physical understanding of an interaction technique based on real-world experience[1].
- Feedback is when the user is provided with information of the results of their actions and directs what actions can be taken next[1].
- Mapping is the relationship between a control and its effects[1].
- Visibility conveys to the user their current state and possible actions[1].

2.3 Pages

2.3.1 Login

The login page is the first page that appears when the application is initially opened. It is designed to have strong visibility, such that a new user should know exactly what this page is for and what to do next. There is two empty text fields that indicate user input, and a Login button to proceed. The login button is a constraint as it is disabled until both fields are filled. See Figure 2.

2.3.2 Register

The register page is designed similarly to the login page, with strong visibility and a simple and uncluttered design. The empty fields afford input and the register button is disabled until valid input. See Figure 3.

2.3.3 Explorer

The explorer page is the main page of the application. From here every other page can be accessed via the navigation bar. The explorer consists of suggested workouts, diet plans, and other interesting articles that may appeal to the user. The buttons within the page affords to be clicked to access the details about that specific section. See Figure 4.

2.3.4 Log

The log page is used to log specific workouts. This page can be accessed using the navigation bar. The page layout is similar to the login and register page, where there are empty text fields for user entry and an add button to submit the results to their workout log. See Figure 5 6.

2.3.5 Progress

The progress page is used to track the user's workout history. This page can also be accessed using the navigation bar. The progress page consists of charts to visually illustrate the user's performance as well as a log history below the charts. The there are option buttons the users can choose on how they want to display the data, which acts as the constraints. See Figure 7.

2.3.6 Settings

The settings page can be used to logout of the account and also change the user's profile picture. This page can also be accessed using the navigation bar. There are two functionality within this page, allowing the page layout to be simple and not cluttered. The buttons on the page aford to be clicked. Feedback is given to the user is they have successfully logged out or changed their picture.

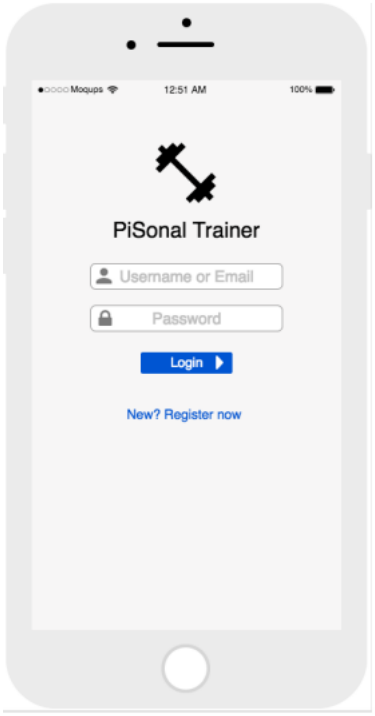

Figure 2: Login page

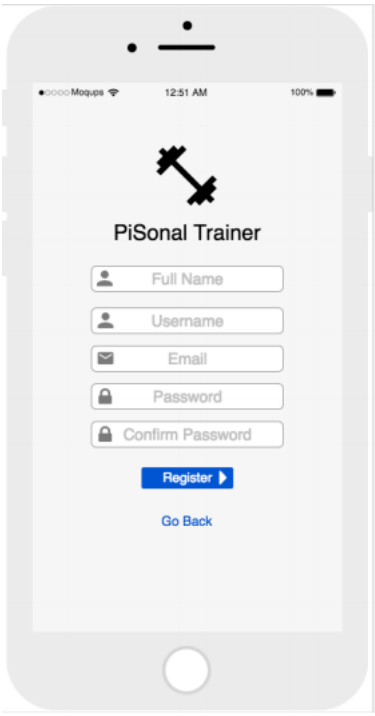

Figure 3: Register page screenshot

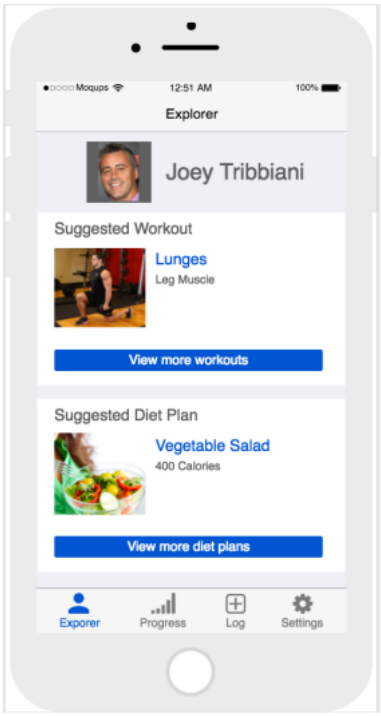

Figure 4: Explore page screenshot

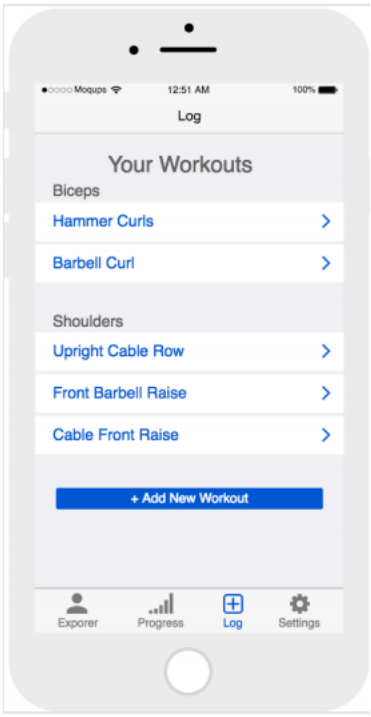

Figure 5: Log workout page

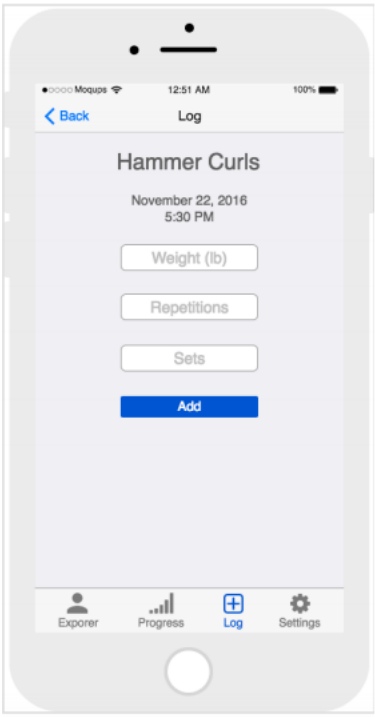

Figure 6: Log workout page screenshot

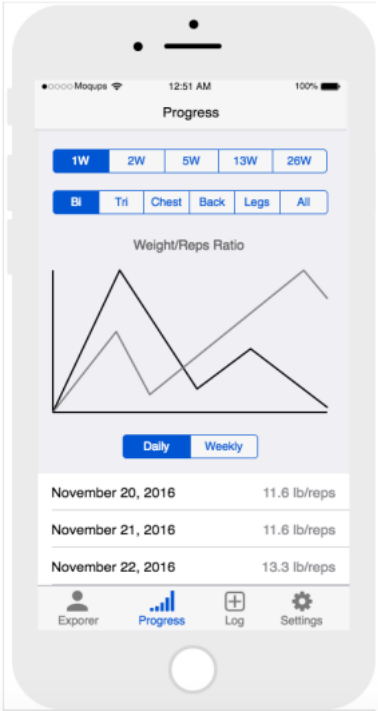

Figure 7: Progress page screenshot

3 Module Decomposition

3.1 Login

`login (email, password)`	
Input:	Email and password to validate user authentication
Output:	If the response confirms credentials as correct, the user is redirected to the dashboard on the app
Description:	Allows users to login into their PiSonal Trainer Account using their username/email and password

3.2 Register

`signup(email, username, fullname, password, confirm-password)`	
Input:	Sends a request to the server with all the fields entered (email, username, fullname, password, confirm password)
Output:	The response from the server confirms that the account was created, the user is redirected to the login screen of the app and a congratulations alert box is displayed.
Description:	Allows users to signup for a PiSonal Trainer Account

3.3 Explorer

`getWorkoutSuggestions(userid)`	
Input:	Sends a request to the server with the userid
Output:	Respond is an array of suggested workouts, the array is rendered as list view on the Explorer tab page.
Description:	Provides personalized workout suggestions

`getDietSuggestions(userid)`	
Input:	Sends a request to the server with the userid
Output:	Respond is an array of suggested diet plans, the array is rendered as list view on the Explorer tab page.
Description:	Provides personalized diet suggestions to the user

3.4 Progress

`getLogs(userid)`	
Input:	Sends a request to the server with the userid (randomly generated string on signup), to retrieve the user's workout logs
Output:	Response is an array of json object with timestamp, exercise name, number of repetitions and sets. This information is rendered as a graph and as a list view of these stats.
Description:	Gets existing log history

`changeGraphView(weeksMode)`	
Input:	weeksMode is the input paramater (eg. 1w, 2w, 4w, 8w, 10w, 12w)
Output:	Changes how many weeks of data is plotted on the graph.
Description:	Displays graph depending on input type.

3.5 Log

`addToLog(userid, exercise, sets, reps, weight)`	
Input:	Sends a request to the server with the parameters exercise (String), sets (Integer), reps (Integer), weight (Integer). This adds a log entry/record under the user with userid.
Output:	Response from the server is a confirmation on whether the addition of a log entry was successfull. If an error message occurs, an error message is displayed and the user is prompted to try again.
Description:	Add Workout Log

3.6 Settings

`renderSettings()`	
Input:	Void
Output:	Renders the list view of avaliable settings on the Settings tab
Description:	Renders setting list view

`setUserProfileImage(userid, imageURL)`	
Input:	Sends the request to the server for setting a new imageURL to the user with userid
Output:	Response is a confirmation of whether the new imageURL was set for the user with userid
Description:	Method to set user profile image for account

`logout()`	
Input:	Void
Output:	The logged in session on the app is cleared. The cached data (user workout logs) is deleted from the client's device.
Description:	Method to logout of user account

3.7 Counting algorithm

`cameraLoop(camera, pts, reps)`	
Input:	A reference to the camera, position values of the object over time, number of reps
Output:	Void
Description:	Continuously checks motion detected by the camera to determine if a rep has been performed, or if the set has ended.

`detectDirection(prev, cur, reps)`	
Input:	Previous position, current position, rep count
Output:	Updated rep count
Description:	Updates the reps variable, which keeps count of the number of repetitions of a movement. It uses slopes to determine whether the object is moving upwards (positive) or downwards (negative). A consecutive up and down movement counts as one rep.

`detectEndOfSet(stop0, t0, sets, reps)`	
Input:	Time when object stopped, current time, set count, rep count
Output:	Updated values of stop0, sets and reps
Description:	Detects the end of a set if object stays in the same area for more than 10 seconds. The object does not need to stay perfectly still as the average of the last 3 position values of the object is compared to an offset value.

4 Relational Database Structure

4.1 ER Diagram

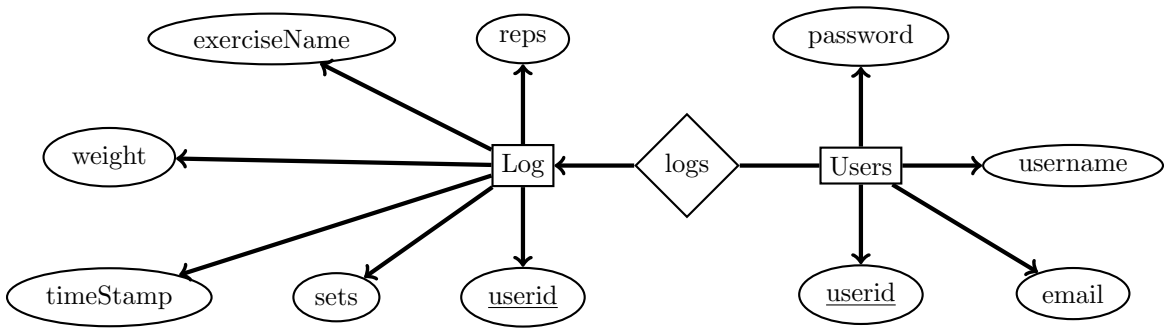

Figure 8: ER diagram PiSonal Trainer table descriptions

4.2 Table Descriptions

- **Users** table stores users' credentials and basic information.
- **Logs** table contains all the records for every user, identified by the userid.

5 Development Details

- Languages of implementation
  - Objective-C
  - JavaScript
- Supporting frameworks
  - OpenCV
  - React Native
- Supporting technology
  - Camera on iOS devices

References

[1] Norman, D. The Design of Everyday Things. Basic Books, New York, 2013.