PiSonal Trainer: Weight Lifting Performance Tracker

System Architecture

Birunthaa Umamahesan Micaela Estabillo Simarpreet Singh

January 11, 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**
1/11/2017	1.00	Birunthaa Umamahesan	Proofread document for revision 0
1/11/2017	1.00	Micaela Estabillo	Finalize document draft
1/9/2017	1.00	Micaela Estabillo	Create document outline

Table 1: Revision history

1 Introduction and Overview

This document provides a description of how PiSonal Trainer will be built. The following are covered in this document:

- Diagram and individual descriptions of the overall component decomposition of the system
- Anticipated and unlikely design changes
- The Requirements and Design Traceability Matrix relating the system's components to the project requirements established in the Software Requirements Specifications (SRS) document

2 Templates, Symbols and Conventions Used

The modules' functions are specified in the format shown in Table 2. This is used in the Detailed Design document.

`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

3 Decomposition into Components

The system is decomposed based on its different elements in order to allow each component to be implemented and developed independently. Nevertheless, each component interacts with at least one other.

1. **User Interface (UI)**: The UI encompasses the frontend and backend of the mobile app used to track workouts and display performance history. The app communicates with the database containing a collection of authorized users to authenticate. It also sends exercise performance statistics to the database. A mobile platform is ideal for PiSonal Trainer so that users can easily bring their device to their gyms, and record or view their performance.

2. **Camera and Counting Algorithm**: Since the camera in use is a Macbook webcam, the movement counting algorithm is implemented on the same machine using OpenCV-python. This is done to reduce resource overhead that would be generated by sending image information for processing to a remote server. The algorithm is discussed in detail in the Detailed Design document.

3. **Database**: The database keeps unauthorized users from using the app and stores counts pertaining to a user's exercises. User performance graphs are dynamically generated by querying the database and sending data back to the user's phone. All the counts are stored in the database in order to avoid taking up too much space in the phone's disk.

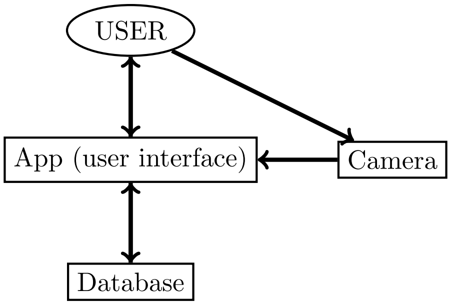

Figure 1: Interaction of PiSonal Trainer Components

4 Anticipated Changes

The expected changes will be caused by changing the system camera from a Macbook webcam to the mobile phone's built-in camera. The new interaction diagram of the components is illustrated in Figure 2.

1. **Use phone's built-in camera**: Merge the camera component into the mobile app so vision and movement detection can be performed within the app instead of an external webcam and computer. To track movement using the user's phone, the counting algorithm would have to be ported to JavaScript as well.

2. **Add a camera tab to existing mobile app**: The app currently only allows manually typing in the number of reps/sets for an exercise and a weight. This feature will be retained, but a new tab for recording performance metrics using the phone camera will be added. Using the camera for inputting counts assumes that the gym equipment will be colour-coded according to weight, thereby removing the need for manual input.

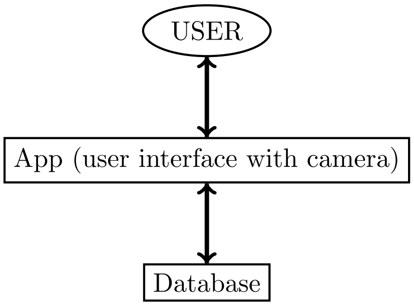

Figure 2: Interaction of PiSonal Trainer Components After Changing Camera

5 Unlikely Changes

1. **User interface layout**: The general look of the mobile app is unlikely to change, even after adding the camera tab (see Anticipated Change 1).

2. **Movement detection algorithm**: The algorithm behind the rep/set counting mechanism of the system will likely stay the same even after porting the Python code to JavaScript in order to use a mobile camera.

6 Requirements and Design Traceability Matrix

The following table contains the functional requirements identified in PiSonal Trainer's Software Requirement Specification document, their descriptions and their corresponding design component/s.

Requirement	**Description**	**Design Reference**
R1	A QR code attached to gym equipment shall be scanned using the app	UI
R2	The app shall be able to identify the equipment's weight	UI
R3	Only registered users may be able to use the app	UI, Database
R4	The app shall provide meaningful error messages	UI
R5	The app shall display the user's progress statistics through visual graphs	UI, Database
R6	A camera shall track the motion of the gym equipment	Camera
R7	The user's exercise performance data shall be stored for future retrieval	Database
R8	The user's performance shall be calculated using data from the camera and the equipment's weight	UI, Database, Camera

Table 3: Requirements and Design Traceability