**Operations Manual**

*Team 17*

*Project: Smart Phone Barometric Mapping*

*Sponsor: Risk Management Services*

*Members:*

*David Minter Dayreal Brown*

*Xuanchen Xiang Ziheng Zhang*

**OPERATIONS MANUAL**

**TABLE OF CONTENTS**

Page #

Contents

[Article I. Introduction 3](#_Toc479460568)

[Article II. Components 3](#_Toc479460569)

[Article III. Setup 3](#_Toc479460570)

[Article IV. Operation 4](#_Toc479460571)

[Article V. Troubleshooting 5](#_Toc479460572)

[Appendix 5](#_Toc479460573)

# Introduction

This is the operations manual of the Smart Phone Barometric Project.  The project uses sensors in iPhones to gather information.  Then the data is sent to a database where it is cleaned up and ready to use as information in developing weather maps.

  Something important to know about this project is that it uses several different languages, APIs, and technological environments.  This is important because if you want to change anything you must make sure that it does not affect the bridge between any two types of functionality.  While breaking things should be easily avoidable as long as the interfaces of the connecting functions are not altered, there is always a chance that a change causing problems further down the line when changes are made.  So, depending on the change desired, there will potentially be a lot of research involved to learn about each of the various parts of this project and how to alter them so that the program will remain functional at each stage.

# Components

The major parts for this project consists of the iOS application and the MySQL database. The iOS application was designed to use the barometer built into smartphones, namely Apple iPhones 5 and later, to gather pressure readings. The MySQL database was designed to receive values from the iOS application and store them for data review.

The iOS application was created using Xcode, Apple’s IDE, integrated development environment, for creating apps for iPhone, iPad, Mac Apple Watch and Apple TV. The MySQL database was created using PHP Storm, an IDE for writing PHP files and a web server with XAMPP to store the database.

The two components are connected through code written inside of the iOS application that sends the values read by the barometer to the database while also displaying it on the iPhone screen. The connection is done by using a PHP connection. Through XAMPP, the PHP files can connect to the web server and execute functions.

If there is an error in the code, there is no quick fix besides examining the code. There should be no errors currently present in the code If the error is in Xcode, Xcode may be able to direct you to the exact location and line of the error and place a warning symbol. If the error is in the PHP code, PHP Storm will highlight the code that generated the error.

# Setup

Largely all the setup is done internally within the iOS application. Firstly, you will need a Mac computer and install Xcode. To install the application onto an iPhone, bring up the project in Xcode, plug the iPhone into the computer, and from within Xcode set the simulation device to the iPhone plugged into the computer. Run the project and if there are no errors Xcode will automatically install and start the application on the phone. The iPhone can now be unplugged from the computer.

To view the database, in a web browser go to the URL: *localhost:80/phpmyadmin*. You will need to have XAMPP installed on your computer and running before you try to access, otherwise you will not be able to view the page. This provides the connection between the PHP code and the webserver with the database. You will need to have the PHP files in the correct folder within the XAMPP application files. The PHP files should be in the following relative location, ..*/Applications/XAMPP/htdocs/MyWebService*. The folder name *MyWebService* is arbitrary and can be anything you decide. The important aspect is that the PHP files are within this folder. Inside of the *MyWebService*folder, there two folders *api* and *includes*. The *api* folder contains the PHP file for creating the data entry for the database, *creatingreading.php,* and *includes* contains the files related to connecting to the database, *config.php, DbConnect.php,* and *DbOperation.php*

*.*

One portion of the code that may need to be modified is the URL for the web server containing the database. The location of the URL variable is in the file *ViewController.swift* in the Xcode project. The URL may need to be changed because it must contain the IP address of the machine you are running on. For example, from the computer that was used for testing, the used URL was: [*http://216.162.146.246/MyWebService/api/createreading.php*](http://216.162.146.246/MyWebService/api/createreading.php)*.*

There are no hazards associated with the setup of the application or the database. The software programs Xcode, PHP Storm and XAMPP are all safe and can be download from the Mac App store for Xcode or the official websites for PHP Storm and XAMPP.

# Operation

Once everything has been setup, operation is very simple. Open the application on the iPhone and you will be taken to the main screen. On this screen, at the top, you will see a start button on the right side of the screen and a stop button on the left. Simply press the start button and the application will begin gathering pressure readings using the installed barometer. If the app is installed on an iPhone that does not have a barometer, iPhone 5 and earlier, then the app will display an error and will return you to the main screen.

For iPhones with a barometer present, the next screen will display 7 values: time interval, relative altitude, elevation, pressure in millibars and kilopascals, longitude and latitude. The time interval is displaying the time the application has been running. Relative altitude is displaying, in meters, the change in altitude detected since starting the app. The value should initially be 0 and as the phone goes to higher elevations, the value will rise and fall as the user goes to lower elevations. Elevation shows the user their current height above sea level in meters. The pressure value gives the value for the pressure in kilopascals and its equivalent value in millibars. The longitude and latitude give the user their current location coordinates. A new reading will be taken every second and each reading will be sent to the database. It should be noted that the phone will require either a data or WIFI connection to allow for sending the readings to the webserver. Whenever you want to stop gathering readings, simple press the stop button on the top left and you will be taken back to the main screen. When the application is stopped, all values are reset.

When the values are sent to the database, there are fields: id, time, relative altitude, elevation, pressure in both millibars and kilopascal, longitude and latitude. Each reading from the iPhone will be a new entry. The entries will be appended to an ongoing list so last as the app is running. The entries will be ordered by the id field as it serves as the primary key for the table. The database can be queried to find averages of pressure for a given time or other relationships.

# Troubleshooting

* **The App won’t open/ crashes upon starting.**
  + *Try reinstalling the app from Xcode.*
* **URL for Database web server won’t open/connect.**
  + *Make sure XAMPP is running. Go to the “Manage Servers” tab and check that MySQL Database, ProFTPD, and Apache Web Server are all running. If they are not, click the “Start All” button at the bottom of the window.*
* **App gives error message of “No Barometer Detected”.** 
  + *The phone being used may not have a barometer install or possibly the barometer is damaged. In both cases, it would be advisable to try using a different phone.*
* **Readings are not being added to the database**
  + *check* createreading.php *and* DbConnect.php *files and run the following address in Postman or a similar program to check for correct operation.* <http://216.162.146.246/MyWebService/api/createreading.php>

Maintenance would include reinstalling the application back on to the phone from Xcode occasionally. It was noticed that after the phone was off for an undetermined amount of time, the application would refuse to open without crashing. The reason for this issue is unknown and had only been observed twice.

# Appendix

**Purpose of the following files:**

*config.php* – Setup and define configuration for database login and table selection.

*DbConnect.php* – Connect to database using information in config.php and notify if error occurs.

*DbOperation.php* – Open connection to database and create entry for data to be passed from a source to the database. The information is the variables defined in createreading.php are sent their associated field in the database.

*Createreading.php* – Define variables and assign them to their corresponding field in the database. Produce message whether data entry was successful or not. Also display message if the request method sent to the database is not “POST”.

**Website for downloading programs used:**

Xcode: Can be found on the Apple App Store

PHP Storm: <https://www.jetbrains.com/phpstorm/download/#section=windows>

XAMPP: <https://www.apachefriends.org/download.html>