

Electronic weather Tracking System

Advanced Programming 2015



4/06/2015

James Cook University Cairns

Contents

Project Scope 2

Project Tools 2

Requirements Analysis 2

Implementation 2

Design 2

Modularity 2

Database 2

Bibliography 3

# Project Scope

The goal of this project is to create a client-server system, in which users can use the PCduino to measure the weather and climate in their vicinity. A web interface will be used to allow for portability, ease of use and accessibility.

The device will measure environmental variables such as temperature, humidity and light. The device could be used in conjunction with cellular or satellite networks to provide more weather coverage and reduce production costs.

The device can also be used in other types of scientific research, whether current or potential research. Making the device simple to use allows for a larger number of people to have access to the tools, which means more people can conduct research.

The data will be saved in to a MySQL database, which can be used to generate statistics and conclusions. With the right tools, the user can access the database from comfort of their office or home.

Planting these devices in a potentially dangerous area with remote access allows for the safe study of places, which were not available to study for an extended period of time.

Setup is also simple which allows for simple and timely execution of the device.

# Project Tools

The software used to write the server-side code was JetBrain’s IntelliJ IDEA 14, the server-let was written in Java and was used in conjunction with Apache TomCat 8. The server handled requests using PHP 5 and JavaScript

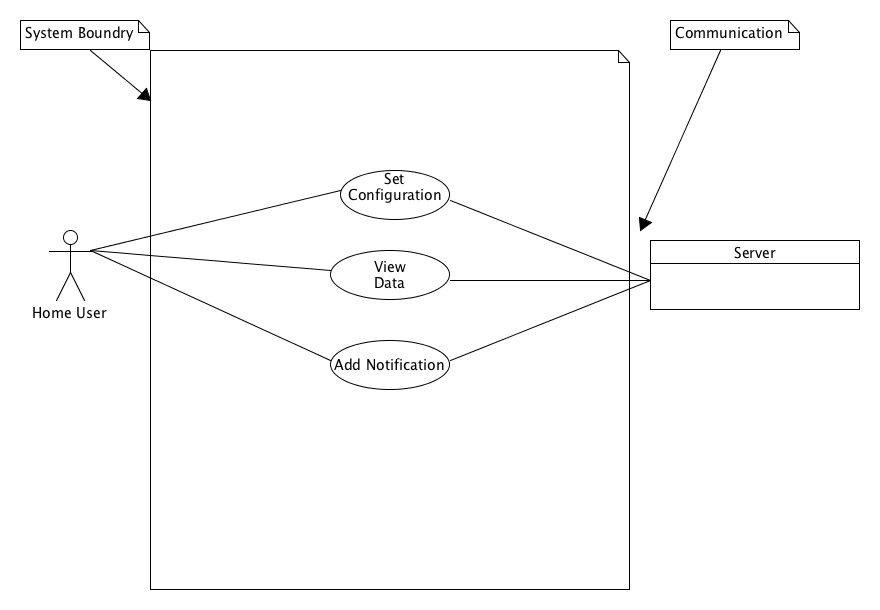
The client-side interface is web based and was written using JetBrain’s WebStorm. The web interface was created using a combination of Hyper Text Mark-up Language, Cascading Style Sheets and JavaScript.

All data stored for use, was stored in the open source database software Oracle MySQL. The data will be stored in tables, which will be appropriately named after the type of data in which they are stored.

All user-interface related software will be used to create a system, which follows the model-view-controller software architecture pattern.

# Requirements Analysis

## Weather Tracking Use Case Diagram



## Fully-dressed Use Cases

Use-case UC1: Set configuration

Primary Actor: Home User

Stakeholders: Companies, Individuals, Groups, Scientific institutions, Governments.

Preconditions: The home user has a working browser & a connection to the server.

Main scenario:

1. The user loads the browser.
2. The user turns on all the sensors.
3. The system changes the values generated to something other than -1.

Use-case UC2: View Data

Primary Actor: Home User

Stakeholders: Companies, Individuals, Groups, Scientific institutions, Governments.

Preconditions: The home user has a working browser & a connection to the server.

Main Scenario:

1. The user loads the browser
2. The user views the data
3. The system pulls the values from the database
4. The system displays the data in a generated web page.

Alternate Scenario:

2a. There is no data in the database.

1. The user waits for the servlet to generate values.

3a. There is no data in the database.

1. The user waits for the servlet to generate values.

4a. There is no data in the database.

1. The user waits for the servlet to generate values

Use-case UC3: Add Notification

Primary Actor: Home User

Stakeholders: Companies, Individuals, Groups, Scientific institutions, Governments.

Preconditions: The home user has a working browser & a connection to the server.

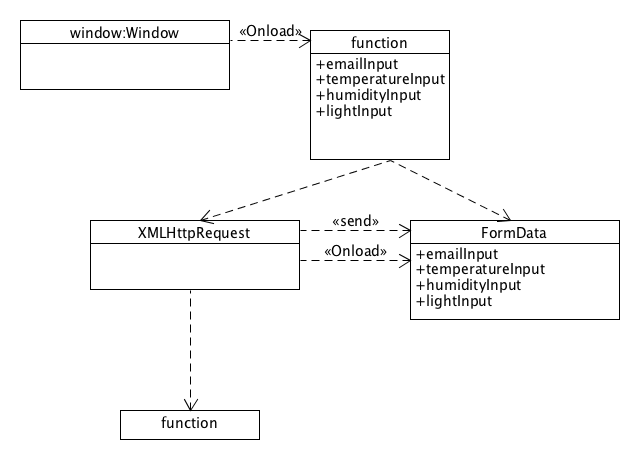
Main Scenario:

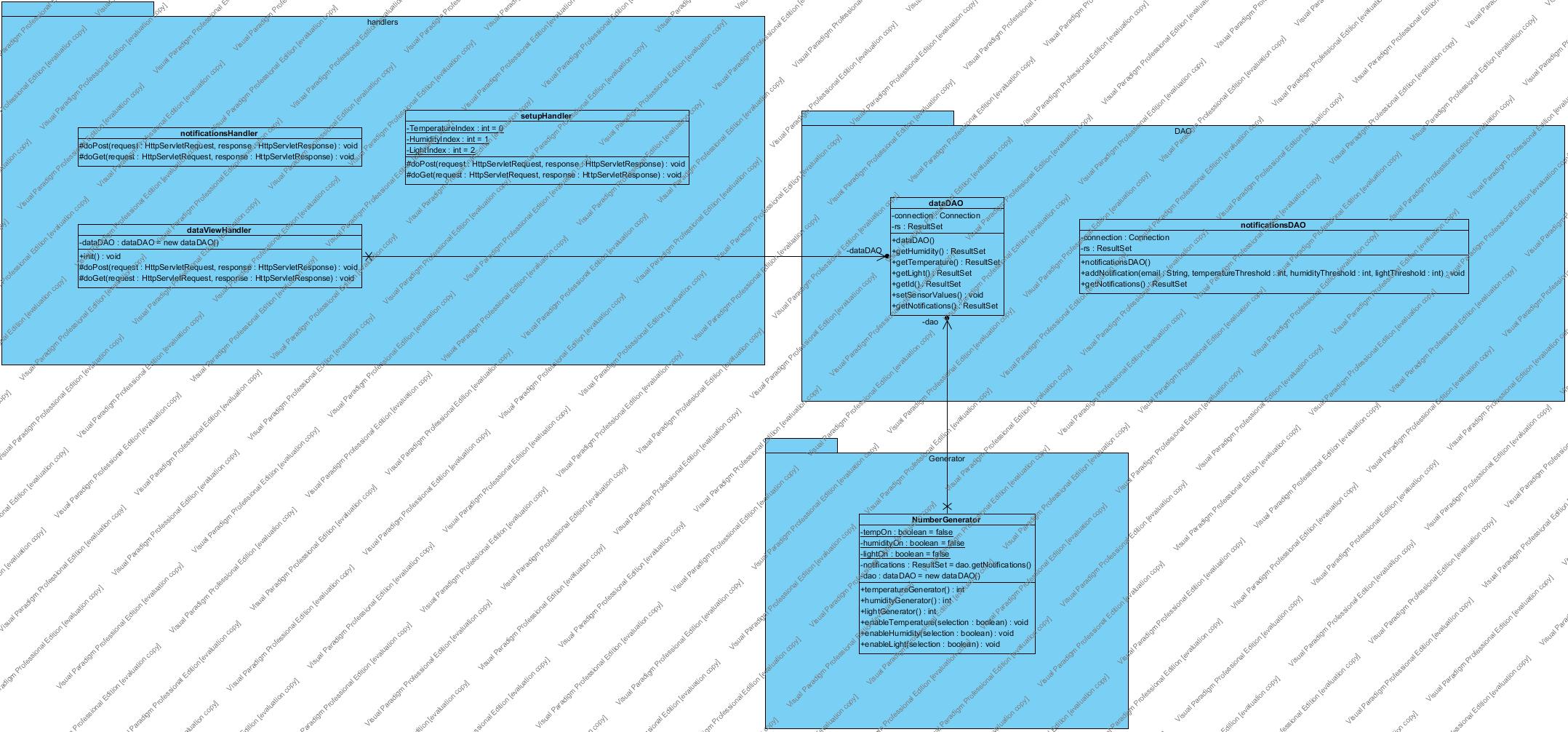
1. The user loads the browser
2. The user adds their E-mail and Thresholds
3. The system generates values and check if the user’s threshold has been reached.

# Implementation

1. The user opens their browser.
2. The user goes to the homepage of the weather tracking system.
3. The user selects the configuration from the navigation bar.
4. The user chooses which values they want to display.
5. Selects the relevant checkbox, which will display the values.
6. The user selects submit.
7. The user selects the notifications link in the navigation bar.
8. The user enters their e-mail and thresholds, which they want.
9. The user receives e-mails when the thresholds have been reached.
10. The user reads their e-mails to check, which thresholds have been met.
11. The user views the data.

# Design





# Modularity

## The model view controller

* The model is the number generator; the view is the html, css, javascript and Servelet class handler.

## Monostate

* The setupHandler is a monostate class.

# Database

# 

# Bibliography

**There are no sources in the current document.**