Thermometer App

High Level Design

1.0

November 19, 2013

Ray Yi

Software Engineer

Prepared for

WUBS UI Thermometer Assignment

**Table of Contents**

1. General Description 1

2. Hardware 1

3. Software 1

3.1. node.js and express framework 1

3.2. Twitter bootstrap 1

3.3. JQuery and angular.js 1

3.4. Jasmine 1

4. Sequence Diagram 2

4.1. Main View 2

4.1.1. Diagram script: 2

4.2. Setting View 3

4.2.1. Diagram script: 3

5. Component Design 5

5.1. Default page 5

5.2. Main View 5

5.3. Main Controller 5

5.4. Main Controller Helper 5

5.5. Setting View 5

5.6. Setting Controller 5

5.7. Threshold View 5

5.8. Threshold Controller 6

5.9. Setting Model 6

5.10. Temperature Service 6

5.11. Temperature API 6

# General Description

This document describes high level design for the project Thermometer App.

# Hardware

This app is web-based application, it will run in any browser with HTML 5 support.

We use node.js as www server, it can run on Windows/Mac/Linux.

# Software

## node.js and express framework

I use node.js as www server, and also create temperature API by using express framework.

## Twitter bootstrap

I use twitter bootstrap for UI styling.

## JQuery and angular.js

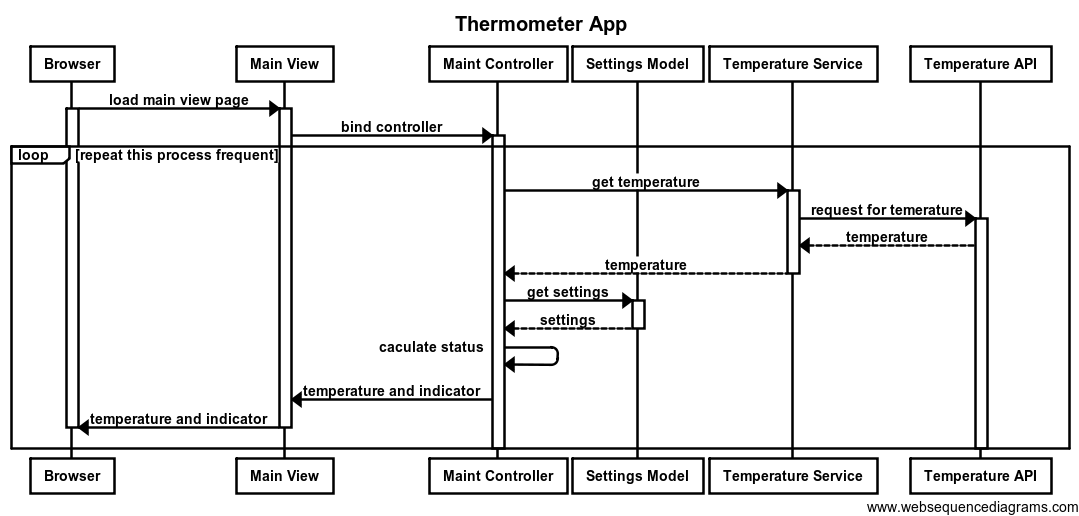
I use JQuery and angular.js to develop this single-page-application.

## Jasmine

I use Jasmine test framework for unit test.

# Sequence Diagram

## Main View



### Diagram script:

title Thermometer App

participant Browser as br

participant "Main View" as mv

participant "Maint Controller" as mc

participant "Settings Model" as sm

participant "Temperature Service" as ts

participant "Temperature API" as api

br->mv: load main view page

activate br

activate mv

mv->mc: bind controller

activate mc

loop repeat this process frequent

mc->ts: get temperature

activate ts

ts->api: request for temerature

activate api

api-->ts: temperature

ts-->mc: temperature

deactivate ts

mc->sm: get settings

activate sm

sm-->mc: settings

deactivate sm

mc->mc: caculate status

mc->mv: temperature and indicator

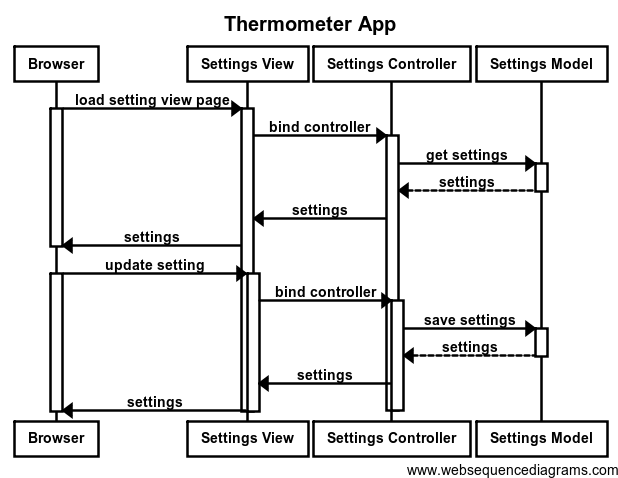
mv->br: temperature and indicator

end

deactivate mv

deactivate br

## Setting View



### Diagram script:

title Thermometer App

participant Browser as br

participant "Settings View" as sv

participant "Settings Controller" as sc

participant "Settings Model" as sm

br->sv: load setting view page

activate br

activate sv

sv->sc: bind controller

activate sc

sc->sm: get settings

activate sm

sm-->sc: settings

deactivate sm

sc->sv: settings

sv->br: settings

deactivate mv

deactivate br

br->sv: update setting

activate br

activate sv

sv->sc: bind controller

activate sc

sc->sm: save settings

activate sm

sm-->sc: settings

deactivate sm

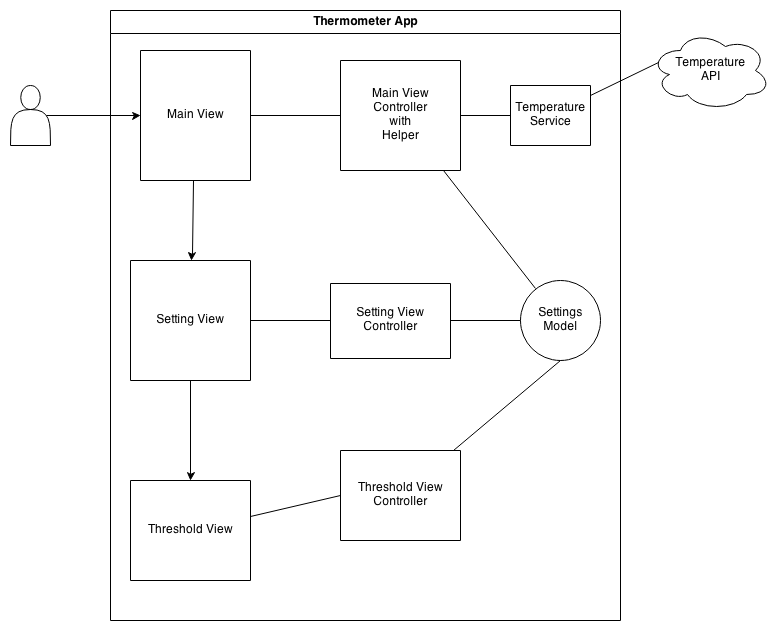
sc->sv: settings

sv->br: settings

deactivate mv

deactivate br

# Component Diagram



# Component Design

## Default page

This is default html page in public folder, which is auto loaded into browser. It is bind to angular app. It contains place holder for partial html for main view and setting view.

## Main View

This is partial html file, it binds to main controller to get temperature and indicators.

## Main Controller

This controller takes below process:

1. call method ‘getTemperature’ to get temperature from temperature API through ‘temperature service’.
2. in the call back function, it calls method ‘updateStatus’ in MainControllerHelper to update temperature and indicators on main view.
3. That method also check if current temperature reachs any threshold, if yes, popup warning modal overlay, wait for user’s response. After user close this modal overlay, go to step 4.
4. If not reach any threshold, use ‘setTimeout’ method to run ‘getTemperature’ method again to repeat step 1.

Step #2 and #3 need use values in Setting Model as parameter.

## Main Controller Helper

This is helper object for main controller, it include below methods:

1. getFormatedTemperature, it will convert temperature from C to F if the setting is F.
2. checkThreshold, if will go through the threshold list in the Setting Model to check if current temperature reach any threshold.

## Setting View

This is partial html file, it binds to setting controller to get/update settings.

## Setting Controller

It connects setting view and setting model. It include 2 event handlers:

1. addThreshold, this is event handler for the icon ‘+’, which add new threshold into Setting Model, and then redirect to threshold view to allow user to edit threshold settings.
2. deleteThreshold, this is event handler for each icon ‘-‘ in each threshold list item, and it delete current threshold from Setting Model.

## Threshold View

This is partial html file, it binds to threshold controller to display/update single threshold settings.

## Threshold Controller

This controller bind threshold view with Setting Model.

## Setting Model

It is json object, it contain all attributes used in setting page.

## Temperature Service

It is angular service, it call Temperature API to get temperature and return promise object.

## Temperature API

This is web method return temperature to client side.