|  |  |
| --- | --- |
|  |  |

Document Control

Review Versions

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Date | Changes by | Comments |
| 0.1 | 07/08/2015 | Simon Cheng | First Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Distribution Authorisation

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Review Date | Authorised By | Position |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1 Summary 1](#_Toc426795277)

[2 Technical Design 2](#_Toc426795278)

[2.1 Overall Design 2](#_Toc426795279)

[2.2 Key technical points 3](#_Toc426795280)

[2.3 UML Sequence Chart for Key Scenarios 3](#_Toc426795281)

[2.4 UI Design 4](#_Toc426795282)

# Summary

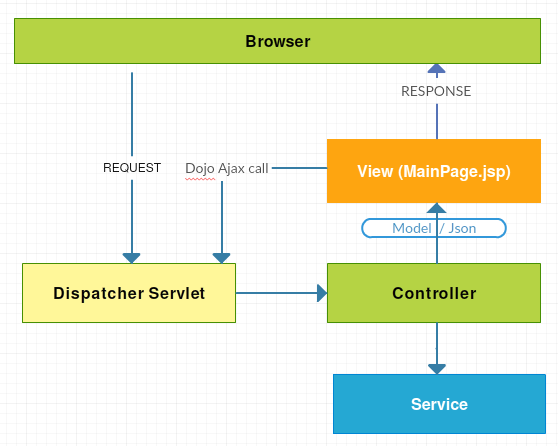
The purpose of this document is outline the design and implementation of City Weather Lookup software, a demo software developed based on the **Code Test - Get Current Weather** from Pactera Technology International Limited.

# Technical Design

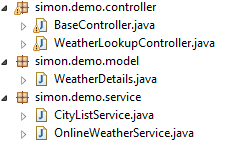
## Overall Design

A web based application has been developed to meet the requirements. Spring MVC pattern is used as the main structure of the software. Behind Spring MVC, there is another service layer to provide specific services to the MVC controller. There is no data persistence layer needed in the design, all the configuration data is fed from Spring Context XML.

The structure can be shown in the diagram below:



The following classes and web pages are created:





BaseController class: Controller for the main UI page.

WeatherLookupController class: Controller to supply the weather details for a given city.

WeatherDetails class: Data model of weather details displayed in the UI form.

CityListService class: Provide the city list to BaseController, so that user could see and choose from the City drop down list in the UI form.

OnlineWeatherService class: Provide city weather lookup service to WeatherLookupController.

MainPage.jsp: The presentation view of this web app.

## Key libraries and tools used

Spring framework 4.1.1 is used to manage the dependencies of object in this web application.

JUnit test 4.11 is used to test the services.

Dojo 1.10 is used to make Ajax call to WeatherLookupController once user changed the value in the City drop down list from the UI.

Jackson 1.9.10 and flexjson 2.0 libraries are used to parse the WeatherDetails model object and return to the browser form.

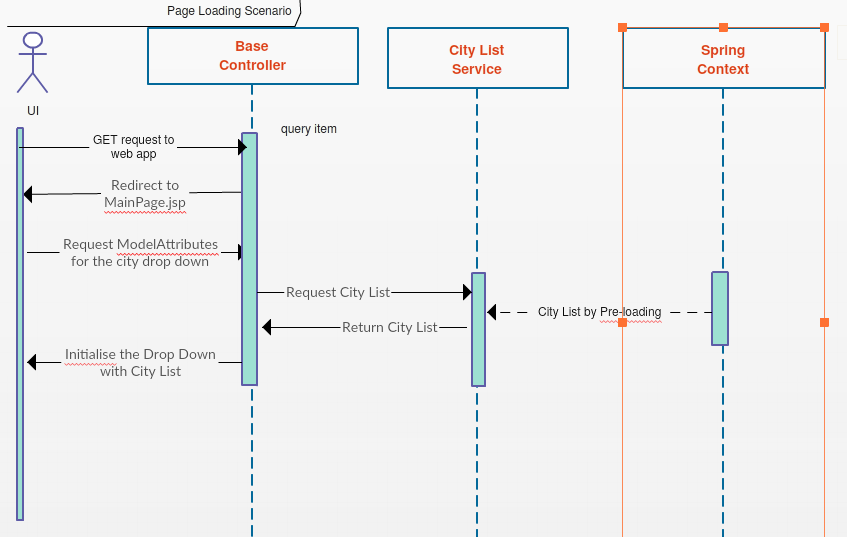
Maven 3.3.3 is used to retrieve the dependency libraries, compile, package, test and also run the web app.

Github is used to do source control.

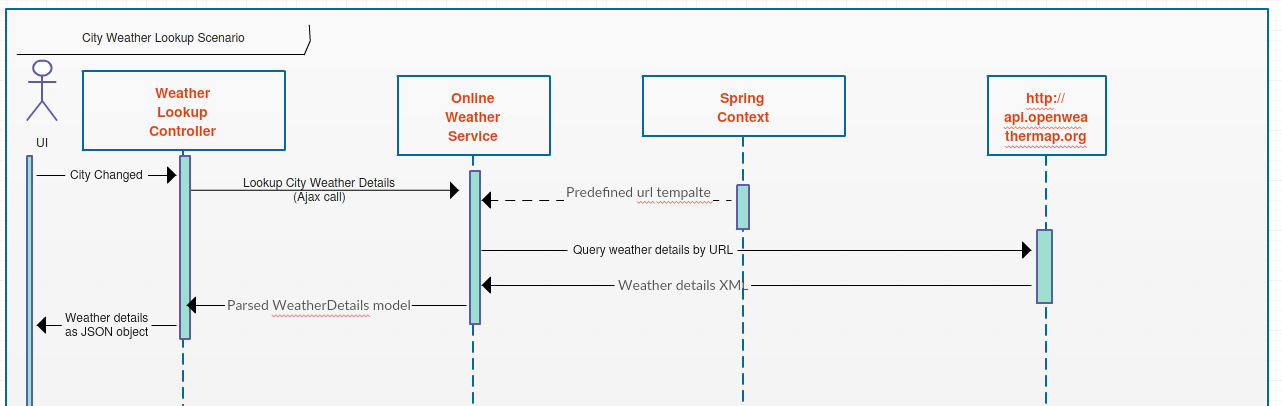
## UML Sequence Chart for Key Scenarios

Below are the two key scenarios in this demo software.

Page Loading Scenario: shows how to load the city list to the drop down list in the UI form.



City weather lookup scenario: shows how to populate the weather details in the UI form from online weather services, once user changed the city in the drop down list.



## UI Design

UI are implemented using Spring form tag lib as below:

