



DESIGN DOCUMENTS

Project of Software Engineering 2

WEATHER-CAL

Authors:

PAOLO POLIDORI

MARCO EDEMANTI

Contents

1	Introduction	5
1.1	Purpose	5
1.2	Scope	5
1.3	References	6
2	System	7
2.1	System Description	7
2.2	Design Constraints	7
2.3	System Architecture	8
3	Design	9
3.1	Persistence design	9
3.2	MVC modeling	9
3.2.1	MVC structure	10
3.2.2	MVC behaviour	10

Chapter 1

Introduction

1.1 Purpose

This document describes the high level design and the technology involved in the development of the WeatherCal software. The target will be accomplished by the use of a description of the architecture which comes after the analysis of the problem and the constraint, explained in chapter 2. The final design and how the application will be developed is shown in chapter 3. This document is a supplement of the RASD formerly redacted. Provide an overview of the entire document.

1.2 Scope

This document is intended for the stakeholders of the system, the developers and reviewers/testers.

1.3 References

IEEE, *IEEE Std 1016-2009, IEEE Software Design Descriptions*, IEEE Computer Society 1998

Raffaella Mirandola, Design and software architecture - slides from SE2 course, 2014

Paolo Polidori, Marco Edemanti, Requirement analysis and specification document for WeatherCal project, 2014

Chapter 2

System

2.1 System Description

The system will implement a calendar as a web application, splitted into client-side and server-side. The former will be used for implementing the asynchronous facilities delivered in the calendar, while the ladder will be used both as an interface for the former to interact with the persistency and for providing web pages to the client.

2.2 Design Constraints

The application, first, will have some constraints on the system proposed by the client explicitly. These include the use of J2EE as server-side application. To this requirement we decided to add Java Server Faces framework for facilitating the development of the structure taking advantage of the MVC design pattern and the facilities brought with the framework. The application also needs to persist data, so we decided to use <NAME OF DBMS> free RDBMS to accomplish this task. The related client for this server application will be developed using both the Web tier

provided by JSF and Marionette.js, a Javascript framework, with its dependencies, for making the client responsive and interactive.

Client constraints even include the time for the system development, which is due to January 25th, 2015.

Constraint imposed by the client does not include any strict restraint on the hardware and the software over which the system will need to be deployed and any further requirement will be added, giving the possibility to be platform independant. Anyway the system on which the platform will be deployed on will have an impact on the server-side application performances and both the client-side environment and the network connecting the client and the server will impact the client-side application performances. Even though both the client and the server software involved in this project have some requirements on the hardware and the software to be used, so they will make our constraint.

2.3 System Architecture

Describe the chosen system architecture. For instance, architectural design patterns can be used to describe the system: peer-to-peer, client/server, stand-alone or embebbed systems, etc. Include also discussion or description of alternative designs

Chapter 3

Design

3.1 Persistence design

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>

3.2 MVC modeling

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

3.2.1 MVC structure

3.2.2 MVC behaviour

Time Reporting

	Paolo Polidori	Marco Edemanti
RASD writing	19 hours	19 hours

List of Figures

Listings