DESIGN DOCUMENT

Project of Software Engineering 2

WEATHER-CAL

Authors:

PAOLO POLIDORI

MARCO EDEMANTI

Contents

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 analisys 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

[1] IEEE, *IEEE Std 1016-2009,IEEE Software Design Descriptions*, IEEE Computer Society 1998.

[2] Raffaela Mirandola, *Design and software architecture - slides from SE2 course*, 2014.

[3] Paolo Polidori, Marco Edemanti, *Requirement analisys and specification document for WeatherCal project*, 2014.

[4] Jesse James Garrett, *Ajax: A New Approach to Web Applications*, February 18th, 2005.

Chapter 2

System

2.1 System Description

The system will implement a calendar as a web application, splitted into client-side and server-side (motivations discussed in section 2.2 and section 2.3). The former will be used for implementing the asyncronous 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. The first one is the use of J2EE as server-side application implying the use of a storage for persisting the data (events, users, invitations, etc.). This constraint entails that the client-server architecture will be adopted in the WeatherCal system.

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 constraints.

2.3 System Architecture

As said in section 2.2 the system will use J2EE for implementing the server-side application and thus the system will rely on a client-server achitecture. The server will also implement the MVC design pattern by means of the Java Server Faces framework, which will facilitate the development of the structure taking advantage of both the design pattern and the facilities brought.

Another choice is to implement a web client because, instead of a traditional application, it provides universal access and no need of being in possess of a dedicated client application.

The application also needs to persist data, so we decided to use <NAME OF DBMS> free RDBMS to accomplish this task.

The related client side will be developed using both the Web tier provided by JSF and Marionette.js, a Javascript framework, with its dependecies, for making the client more responsive and interactive. This framework implements MVC, so the

client will have the same design pattern of the server, with same necessity of storing data. This target will be accomplished by the Java API for RESTful Web Services integrated in J2EE, giving the opportunity of sharing and syncronizing the models. This path was chosen because traditional web application, which are synchronous, need to change the webpage everytime some data need to be exchanged with the server, giving the user a worse experience [4] and it offers more capabilities than the AJAX facilities provided by JSF.

The graphic environment of the web pages will be managed by PrimeFaces, a JSF component suite. It was chosen among other similar libraries both for its features and performances in combination with the availability of support from the client.

Chapter 3

Design

3.1 Persistance 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