Politecnico di Milano

Master of science in computer science and engineering

MeteoCal

A WEATHER BASED ONLINE CALENDAR

Project Plan Document

Teachers

Raffaela Mirandola
Elisabetta Di Nitto
Marco Miglierina

Students

Alessandro Negrini 836806
Andrea Gulino 836681
Paolo Guglielmino 837055

Academic Year 2014/2015

Contents

1 Introduction

This document has the purpose of giving an overview on the project in terms of development phases. For each one, we identify the constitutive tasks and we provide a first effort estimation.

In addition, at the end of the document we will present our team and the way we organise our work in order to successfully complete and meet the deadlines.
MeteoCal is a weather based online calendar that allows people to schedule their activities avoiding bad weather conditions in case of outdoor events.

1.1 Software overview

MeteoCal software must allow its users to create, update and delete new events and add them to their personal calendar.
Each event has its own information, in terms of time and place where the event will take place and wether the event will be indoor or outdoor.
As soon as a new event is created, the system has to enrich the event with weather forecast information. In case of bad weather, the users will be notified by email or when they log into the system one day before the event takes place.
In case of indoor events, there will be no particular tasks to provide. On contrary, for outdoor events, the system should provide a smart solution in case the event can't be carried out. The solution will be presented in a more detailed way in the RASD document.
The software should also allow the users to invite other people to their personal events as guests. Only the user who has created the event is allowed to update or delete it. Guests users can either accept or decline the invitation.

The above is just a brief introduction to what the system will be able to do. In fact, in RASD document system tasks are described more in depth, and every detail is taken into account and explained.

1.2 Task To Do

We must develop the system using the Java EE platform. In particular, we will use EJBs to develop the business logic. The user interface can either be a web application or a normal Java application. We opted for a web application. The user interface has to interact with the business logic.

1.3 Deadlines

In order to complete the project, we must proceed through some steps and deadlines. The expected deadlines are the following :

- **2 November 2014** : Group presentation
  Form our group and create a google code repository
- **16 November 2014** : RASD (Requirements Analysis and Specification Document)
  RASD Document will contain the description of scenarios, use cases that describe them, and the models describing requirements and specifications.
- **7 December 2014** : DD (Design Document)
  DD must contain a functional description of the system and any other view you find useful to provide.
- **25 January 2015** : Implementation
  RASD document must be implemented respecting to requirements. We must provide source code and executable, installation and user manual, system test cases. Also a document containing information on the number of hours must be included.
- **8 February 2015** : Acceptance Testing
  Define of test cases and report on the execution of test for system developed by a different group.
- **10 February 2015** : Project reporting
  Apply Function Point approach to project and check id the result are similar to the actual size of our project.
- **Final Presentation**
  Presentation of our project providing an overview on documents and design decisions, a demo of the system. Duration 40 minutes (since we are a three student group)

2 Project Plan

In this part is discussed in a more detailed way the work load associated to each phase. This part of document will be brought at each meeting in order to cover every point

2.1 Group Registration and Project Planning

This consists just on forming a group, understanding the problem and plan the project, even if there isn't ad hoc deadline.

1. Creating of group
2. Understanding the problem
3. Defining meetings
4. Defining tools, software, ...
5. Individuating macro-tasks and writing of project plan document

2.2 Requirements analysis and specification document (RASD)

1. Identifying goals, domain, and requirements
2. Identifying actors
3. Identifying scenarios and use cases
4. Analysis class diagram
5. Analysis activity diagram
6. Analysis sequence diagram
7. Analysis activity diagram
8. Analysis state chart diagram
9. Learning, defining, implementing and testing Alloy Model
10. Writing RASD document
11. Revision of document

2.3 Design Document (DD)

1. Identifying Architecture
2. Identifying components and their responsibility inside the system
3. Identifying Technologies
4. Defining UX diagram
5. Modeling system
6. Documentation
7. Revision

2.4 Implementation

1. Setup development tools (Eclipse, Git, Maven, ...)
2. Data layer implementation
3. Business logic implementation
4. Web tier implementation
5. User interface implementation
6. Testing
7. Drawing up installation and user manual

2.5 Testing on other group project

1. Identifying test cases
2. Documentation
3. Revision

2.6 Detailed Project Plan

During the development of the system we will fill in a table in which for each activity we indicate the amount of time spent on it. To see this file, please refer to DetailedMeetings.pdf file.

3 Project Members

Our group is composed by three students, thus we have to add the following extensions to the system :

- system has to provide mechanism in order to make calendar/event public, so visible to all other registered users.
- system has to be able to find the closest sunny day in case of bad weather
- system must notify users by mail
- user can import and export their calendar
- system must avoid conflicts
- system must update weather conditions associated to events periodically.

3.1 Project Teachers and Tutors

- Raffaela Mirandola (Leader)
- Elisabetta Di Nitto (Leader)
- Marco Miglierina (Tutor)

3.2 Team Members

- Alessandro Negrini (alessandro2.negrini@mail.polimi.it)
- Andrea Gulino (andrea.gulino@mail.polimi.it)
- Paolo Guglielmino (paolo.guglielmino@mail.polimi.it)

4 Team Organization

In order to work efficiently, the workload will be splitted and distributed among the team members according to two different aspects.
First, we will try to allocate activities depending on **personal capabilities and skills**.
On the other hand, we will also take care of **time availability** of each of us, trying to find the best compromise between the two aspects.

Some activities will be carried out in a jointly way, like in the case of documents writing, because we think that debate and knowledge sharing is the right way to get the best result.
On the other hand, some activities will be accomplished in a distribute way. That's the case of coding: we will work in parallel each of us developing a different part of the system.

4.1 Team Meetings

We have planned 3 meetings per week for a total amount of 4 hours according to the needs of each of us.
During those hours we will work together on a particular common task and we will line up the work made during the week.

- Monday 15.00 - 16.00
- Wednesday 15.00 - 16.00
- Thursday 13.00 - 15.00

However, this dates are not binding, if it happens that a meeting can't be met, it will be kept in another available date.
Some other extra meetings are not excluded.
For a global view of meetings please refer to our google code, where for each meeting we stored tasks performed.

4.2 Tools for Development

- **Signavio** for designing UML diagrams (Class Diagram, Use Case, ...) and **Alloy** for RASD Document

- **Latex** and **PDF files** for the documentation
- **Eclipse IDE** and **Brackets** for write Java code, HTML, Javascript, ...
- **JEE unitary test tools** for testing

4.3 Collaboration Tools and Media

We will use the most famous tools to communicate and collaborate :

- WhatsApp
- Skype
- Email
- TeamViewer (remote control and online meeting)