



REQUIREMENTS ANALYSIS AND SPECIFICATION DOCUMENTS

Project of Software Engineering 2

WEATHER-CAL

Authors:

PAOLO POLIDORI

MARCO EDEMANTI

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 7 |
| 1.1 | Purpose | 7 |
| 1.2 | Scope | 7 |
| 1.3 | Glossary | 8 |
| 1.4 | References | 8 |
| 2 | Overall Description | 9 |
| 2.1 | System Environment | 9 |
| 2.1.1 | Actors | 9 |
| 2.1.2 | Scenario | 9 |
| 2.2 | Functional Requirements | 9 |
| 2.3 | Non Functional Requirements | 9 |
| 2.4 | UML Modeling | 9 |
| 2.4.1 | Use Case Diagram | 9 |
| 2.5 | Alloy Modeling | 9 |
| 3 | Requirement Specification | 11 |
| 3.1 | External Interface Requirement | 11 |
| 3.2 | Functional Requirements | 11 |

| | |
|---|----|
| 3.3 Non Functional Requirements | 11 |
|---|----|

List of Figures

Chapter 1

Introduction

1.1 Purpose

The purpose of this document is to present a description of WheaterCal system. It will explain the features of the system, the interfaces of the system, what the system will do and the constraints under which it must operate. This document is intended for both the stakeholders and the developers of the system.

1.2 Scope

We want to project and implement WheaterCal. The aim of this project is to develop a system that offers an online calendar in which user can schedule their events according to the weather conditions.

A registered user can create, delete and update an event and moreover he should provide information about where and when this event will take place and information about the invited user. Once the event is created the system should provide to its creator the weather forecast information regarding the scheduled day, and most of all

it should notify a bad weather condition one day in advance to all the participants's event.

Also a user is able to make his/her calendar visible to all other registered user showing them only the time slots in which they are busy without letting know the event information unless either the event is public. In addition in case of bad weather, three day before the scheduled data of an event, the system will inform the event's owner and propose to him the closest sunny day.

1.3 Glossary

| | |
|----------------------|---|
| User | La pagina non subisce variazioni e rimane ferma jjkjkjkkjklklldldldlldldldldl |
| Event | La pagina si sposta verso l'alto |
| Event's Owner | La pagina si sposta verso destra |
| Participant | La pagina si sposta verso il basso |
| Calendar | La pagina si sposta verso sinistra |

- User
- Event's Owner
- Event
- Calendar
- Participant

1.4 References

IEEE, *IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications*, IEEE Computer Society 1998

Chapter 2

Overall Description

2.1 System Environment

2.1.1 Actors

2.1.2 Scenario

2.2 Functional Requirements

2.3 Non Functional Requirements

2.4 UML Modeling

2.4.1 Use Case Diagram

Use Case Diagram

2.5 Alloy Modeling

Chapter 3

Requirement Specification

3.1 External Interface Requirement

3.2 Functional Requirements

3.3 Non Functional Requirements