**Politecnico di Milano**

A.A. 2015-2016

Software Engineering 2

Requirements Analysis and Specifications Document

Alessandro Macchi

Caterina Finetti

Simone Manzoli

**The problem: myTaxiService**

**Part I**

The government of a large city aims at optimizing its taxi service. In particular, it wants to: i) simplify the access of passengers to the service, and ii) guarantee a fair management of taxi queues.

Passengers can request a taxi either through a web application or a mobile app. The system answers to the request by informing the passenger about the code of the incoming taxi and the waiting time.

Taxi drivers use a mobile application to inform the system about their availability and to confirm that they are going to take care of a certain call.

The system guarantees a fair management of taxi queues. In particular, the city is divided in taxi zones (approximately 2 km2 each). Each zone is associated to a queue of taxis. The system automatically computes the distribution of taxis in the various zones based on the GPS information it receives from each taxi. When a taxi is available, its identifier is stored in the queue of taxis in the corresponding zone.

When a request arrives from a certain zone, the system forwards it to the first taxi queuing in that zone. If the taxi confirms, then the system will send a confirmation to the passenger. If not, then the system will forward the request to the second in the queue and will, at the same time, move the first taxi in the last position in the queue.

Besides the specific user interfaces for passengers and taxi drivers, the system offers also programmatic interfaces to enable the development of additional services (e.g., taxi sharing) on top of the basic one.

**Part II**

A user can reserve a taxi by specifying the origin and the destination of the ride. The reservation has to occur at least two hours before the ride. In this case, the system confirms the reservation to the user and allocates a taxi to the request 10 minutes before the meeting time with the user.

**Part III**

A user can enable the taxi sharing option. This means that he/she is ready to share a taxi with others if possible, thus sharing the cost of the ride. In this case the user is required to specify the destination of all rides which he/she wants to

share with others. If others are willing to start a shared ride from the same zone going in the same direction, then the system arranges the route for the taxi driver, defines the fee for all persons sharing the taxi and informs the passengers and the taxi driver.

**1 Introduction**

# Purpose

This document represent the Requirement Analysis and Specification Document (RASD). The main goal of this document is to completely describe the system in terms of functional and non-functional requirements, analyse the real need of the customer to modelling the system, show the constraints and the limit of the software and simulate the typical use cases that will occur after the development. This document is intended to all developer and programmer who have to implement the requirements, to system analyst who want to integrate other sy[[1]](#footnote-1)stem with this one, and could be used as a contractual basis between the customer and the developer.

# Actual System

The software house wants to offer a new weather based online calendar. We suppose that until now nothing has been created and we have to create the entair application without using or modify a previous system.

# 1.3 Scope

The aim of the project is to create a brand new weather based online calendar for helping people scheduling their personal events considering weather forecast. Calendar may be set as public or private. Public calendars can be seen by all registered users, instead private ones can be seen only by owners. An event contains information about when and where the event will take place. The user can check the weather forecast and decide the best schedule for the specific event. User, once registered, should be able to create, delete and update events. User, can also invite other users to partecipate to a specific event. Invited users can only accept or decline the invitation. User can choose if the specific event will be public or private, determine the visibility of the event towards other registered user. In case of bad weather conditions for outdoor events, the system should propose to its creator, three days before the event take place, the closest sunny day. The application should also notify the bad weather to all event participants one day before the event. The user will see the notifications when they log in to the system.

# Actors

* + - Visitor: all visitor users can only see the login page and complete the regis- tration form to be able to access to all the functionality of the application as registered user.
    - Registered user: this type of user, after successful login, is the only that can see the calendar interface and create, delete and update events. They can also invite other user to a specific event during the creation process

the weather forecas of that event to the user.

# Goals

List of the goals of MeteoCal application:

* + - [G1] Allow a visitor to became a registered user and choose the public or private nature of his/her calendar.
    - [G2] Allow user to log in to application.
    - [G3] Allow user to create a new event in the calendar and choose the public or private nature.
    - [G4] Allow user to modify an existing event of his/her calendar.
    - [G5] Allow user to delete an existing event of his/her calendar.
    - [G6] Allow user to invite/delete other user to a specific event of his/her calendar.
    - [G7] Allow user to see the weather forecast of a specific event of his/her calendar.
    - [G8] Allow user to see the public event of other registered user.
    - [G9] Allow user to see event to which has been invited.
    - [G10] After login, application will notify only the creator user three days before an event takes place if the weather is not good.
    - [G11] After login, application will notify invited user one days before an event takes place if the weather is not good.

# Definitions, Acronyms, Abbreviations

* + - **Definitions**
      * Creator user: is a registered user that creates an event and becomes the owner of the event itself.
      * Calendar owner: all registered user that administrate his/her calendar.
      * Event owner: user that has created an event.
      * Invited user: is a registered user who has been invited to an event from other registered user and he/she have accepted.
      * Event: anything that takes place or happens in the real world. In our case an event is the only entity that user can create, delete or modify.

1. Visibility: relation between two entities (calendar, event and user) that imply property of been visible. If A is visible from B, A will show in- formations to B. If A is not visible from B, A will hide informations to B.
2. Closest sunny day: is the first sunny day encountered during the analysis of a specific period of time.
3. Bad weather: climate conditions such as rain, storm, snow.

# Acronyms

* + - * RASD: Requirements Analysis and Specification Document.
      * DB: DataBase.
      * DBMS: DataBase management system.
      * API: Application Programming Interface.
      * Database Management System (DBMS).
      * OS: Operating System.
      * JVM: Java Virtual Machine.
      * JEE: Java Enterprise Edition.

# Abbreviations

* + - * [Gn]: n-goal.
      * [Rn]: n-functional requirement.
      * [Dn]: n-domain assumption.
      * [Mnn]: n-mock up.
  1. **Reference Documents**
     + Specification Document: MeteoCal Project AA 2014-2015.pdf.
     + IEEE Std 830-1998 IEEE Recommended Practice for Software Require- ments Specifications.
     + IEEE Std 1016tm-2009 Standard for Information Tecnology-System Design- Software Design Descriptions.

8

1. [↑](#footnote-ref-1)