POLITECNICO
MILANO 1863

Software Engineering II

Travlendar+

Design

Document

Authors:

Edoardo D'Amico
Gabbolini Giovanni
Parroni Federico

1st November 2017

Indice

Chapter 1

Introduction

1.1 Purpose

1.2 Scope

Travlendar+ is an application that aims to ease the management of the daily appointments of a registered user by providing a desired schedule for those. The application runs within the context of a restricted geographic area and thus should consider only the travel means available in this scope. The application is able to interact with different data sources, in particular Travel Means APIs **FARE DEFINIZIONE** and User device GPS, and to arrange the appointments by exploiting these information. Once the schedule is computed, the application gives to the user the possibility to buy tickets for the involved travel means, then he/she can run it and follow directions provided by the application.

1.3 Definitions, Acronyms, Synonims

1.3.1 Synonims

1.3.2 Definitions

Definition 1.3.1. the user data include:

- schedules;
- appointments;

- user parameters.

Definition 1.3.2. A device is a PC, a Tablet or a Smartphone in which run the last version of his O.S.;

Definition 1.3.3. A Schedule is a set of time-ordered and not overlapping appointments where their starting times are fixed and they're linked to each other by a path covered with a specific transportation mean;

1.3.3 Acronyms

- API: Application Programming Interface;

1.4 Revision history

1.5 Document structure

Chapter 2

Architectural Design

2.1 Overview

Travlendar+ has a multi-tier architecture.

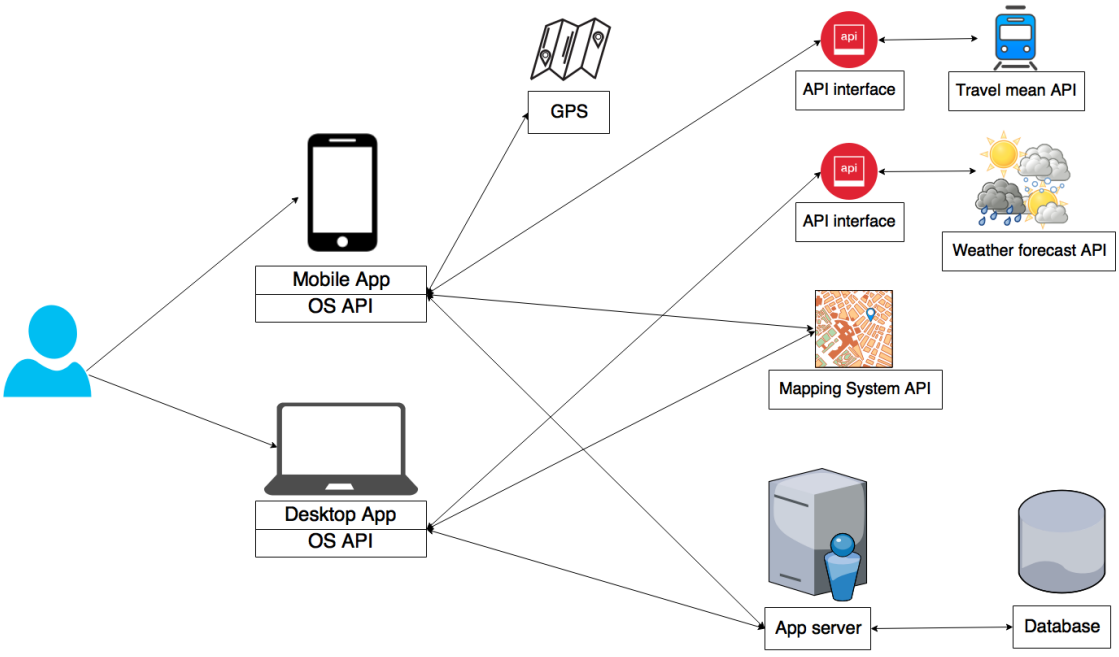

Figura 2.1: High level components

User can interact directly with both version of the application: desktop and mobile, both laying on their respective OS APIs. The schedule computation is performed locally

on the device in which the application runs by querying the GPS of the device and the external APIs:

- Mapping Service API
- Travel Means APIs
- Wheather Forecasts APIs

For the last two ones, the communication happens through different interfaces (wrappers), in order to improve the capability of the system to be expanded. In this way we provide a common pattern for the data acquired from the APIs, so that new ones can be easily included without the need of drastic changes on the central core. In the architecture there are two more components: the Application server, that offers authentication and sinchronization services and the Database used to grant the durability of the data. At the end of the registration phase a new record is allocated in the Database and it will be used for check the identity of a user during the login phase. The Database also stores the user data (1.3.1) so that they can be sinchronized in all the devices in which the user is logged in.

INSERIRE FOTO STACK PARRO

The figure above represents the stack architecture of our application and shows the hierarchy of the layers. Each level can communicate with the upper and lower layer.

2.2 Component View

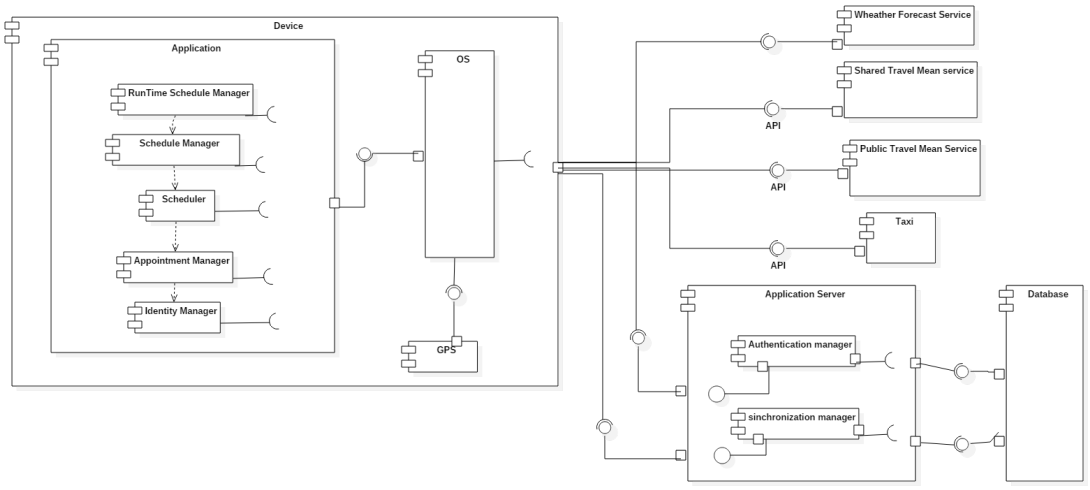

Figura 2.2: Component diagram

The application [1] includes several components:

- Identity Manager: it requests a service to the authentication provider either for the login or the registration of a user.
- Appointment Manager: handles all the operations affecting appointment like modification, deletion and creation.
- Scheduler: it computes and saves the schedules of the user.
- Schedule Manager: handles all the operations affecting the schedule like showing and selecting the schedule to be run, notifying the user if a better schedule involving shared means has been found, buying the ticket for the means within the schedule.
- Runtime Schedule Manager: Shows the directions and the progress of the running schedule.
- Synchronization Manager: it requests a synchronization to the Synchronization provider when needed.

[1] in the image is reported only the mobile version of the application since the desktop version is identical to it, with the only difference that it can't exploit GPS services, so for mantaining the graphic clear only one of the two is reported.

The application Server is composed by the following two components:

- Authentication Provider: it handles the registration and the login phase of the user.
- Synchronization Provider: it has the purpose of synchronizing the user data (1.3.1) in the database when something is changed.

The four components representing a service (wheather forecast, shared travel mean, public travel mean and the taxi service) aren't included in the system because they are external services but they have been represented since our system is using the interfaces exposed from them.

2.3 Deployment View

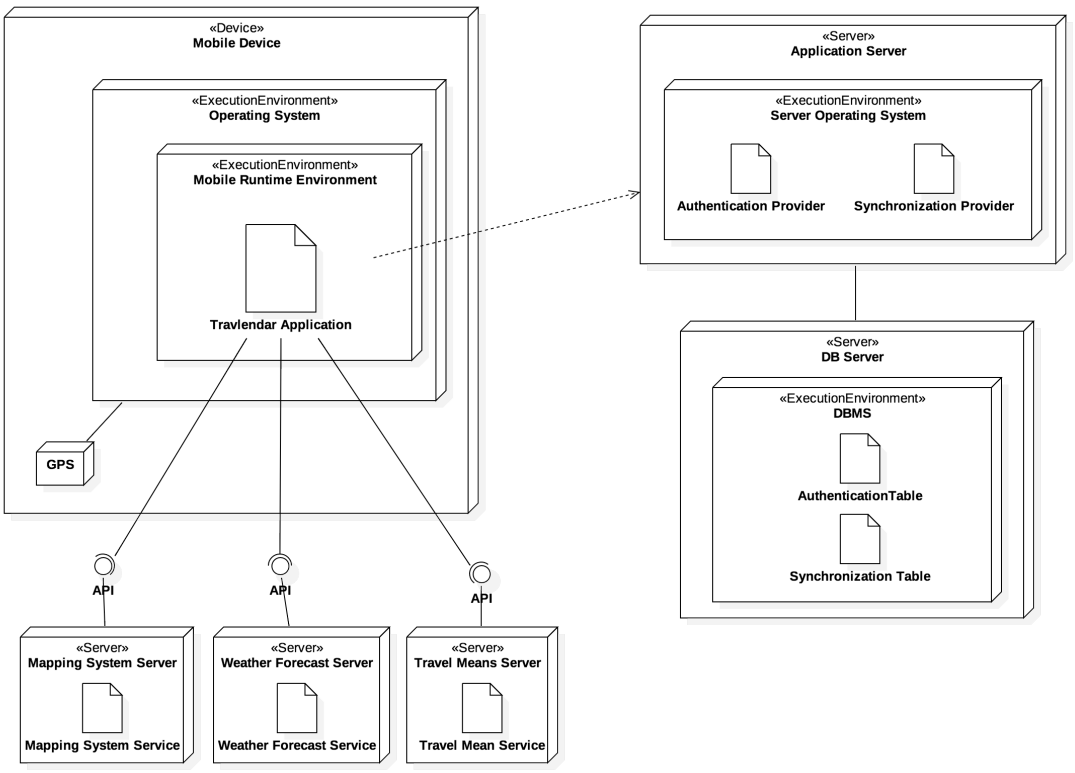

Figura 2.3: Deployment Diagram

2.4 Runtime View

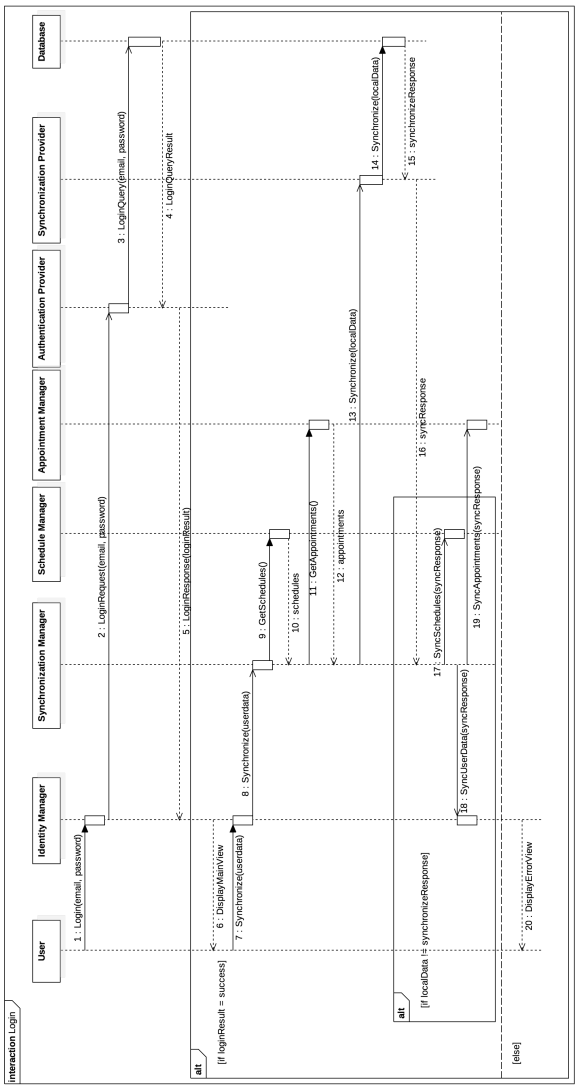

Figura 2.4: Login runtime view

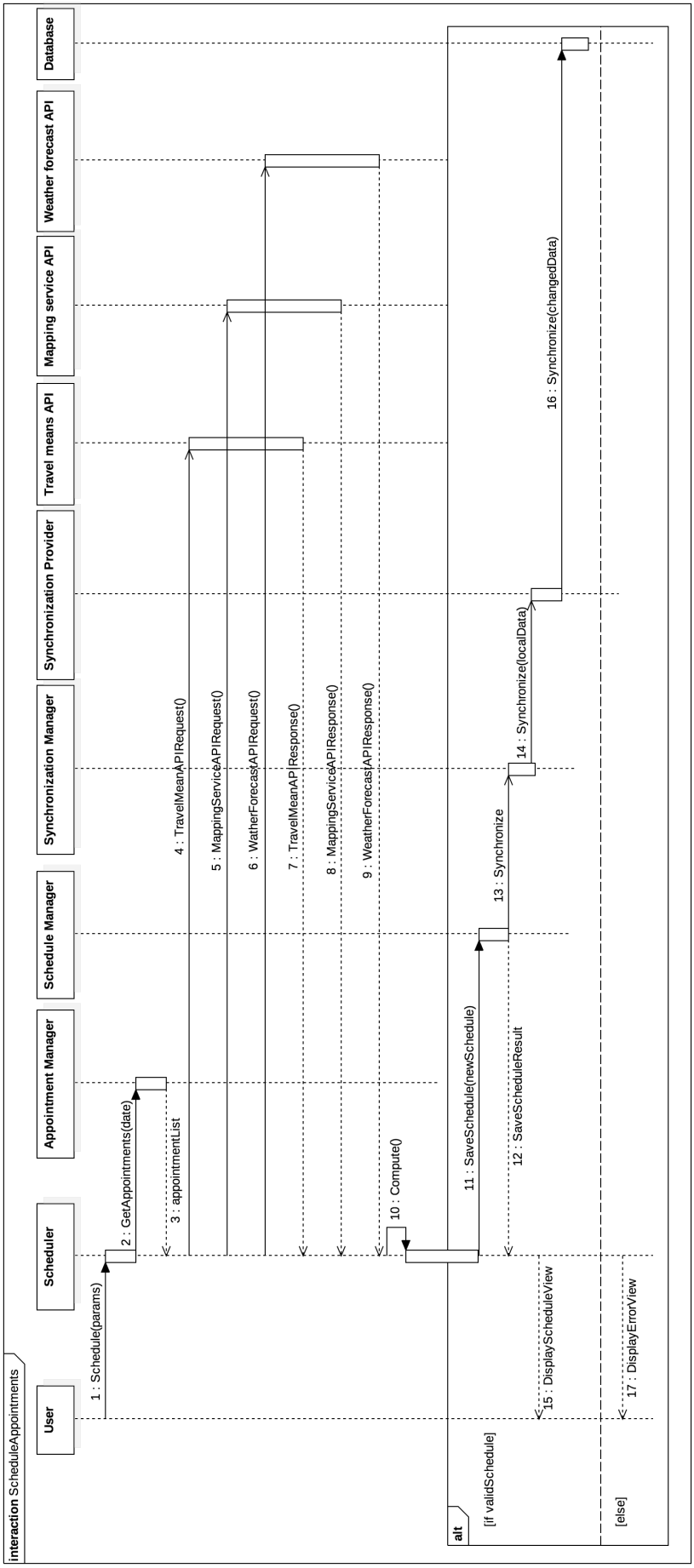

Figura 2.5: Schedule appointments runtime view

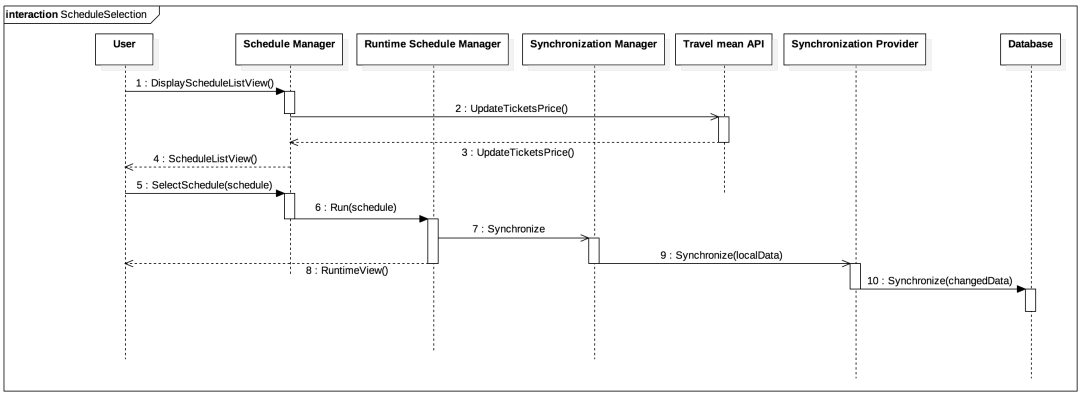

Figura 2.6: Schedule selection runtime view

2.5 Component Interfaces

2.6 Selected architectural styles and patterns

2.7 Other Design Decisions