



POLITECNICO
MILANO 1863

TRAVLENDAR+

DD

Design Document

Kostandin Caushi 898749

Marcello Bertolini 827436

Raffaele Bongo 900090

Date 26/11/2017

Version 1

Table of Contents

1	Introduction	3
1.1	Purpose	3
1.2	Scope	3
1.3	Definitions	3
1.4	Acronyms	3
1.5	Abbreviations	3
1.6	Reference Documents	3
1.7	Document Structure	3
2	Architectural Design	4
2.1	Overview	4
2.2	High Level Components and Their Interactions	4
2.3	Component View	5
2.4	Deployment View	5
2.5	Runtime View	5
2.6	Selected Architectural Styles and Patterns	5
2.7	Other Design Decisions	6

List of Figures

1	Proposed Architecture	4
2	Components High Level	4
3	Component View	5

1 Introduction

1.1 Purpose

This document has the aim of entering into the detail of Travlendar+ system. We will show the software architecture that we have designed for our system with different levels of abstraction, analyzing deeply the main components. Additionally we will exhibit a runtime view of the system, showing as the various components will interact between themselves specifying architectural and design patterns used. Finally we will present the critical algorithm implemented, a requirement - software components correspondency and an high level plan about implementing and integrating the various components.

1.2 Scope

Travlendar+ is a calendar-based application that has the aim of managing the many meetings, events and appointments that a user has to deal with every day.

The system will let the user create events in his personal calendar, checking if he is able to reach them on time and supporting his choices about the way of reaching the location.

Travlendar+ will also give the user the possibility to buy tickets of a town's public and private means of transport and it will also allow him to manage his travels to reach other cities, creating specific travel events in the calendar section. The system will offer other additional features :

- The possibility to register the season ticket for the public transport. Travlendar+ will notify the user when the expiry date is near.
- The possibility to set the starting time, ending time and the preferred duration of every day lunch. The system will guarantee to reserve at least 30 minutes for this purpose.
- In case of outdoor trips, the user will be able to insert the period he will spend out of town and the system will suggest him the most convenient transport tickets available, keeping in mind the information given.
- The possibility of setting the anticipation time for reaching the various events. The system will warn the user when he needs to leave in order to arrive on time.

1.3 Definitions

1.4 Acronyms

1.5 Abbreviations

1.6 Reference Documents

1.7 Document Structure

2 Architectural Design

2.1 Overview

In this chapter we will analyze the proposed architecture and components of the Travlendar+ system.

The proposed architecture has three tiers :

- *Presentation Tier* : represented by Browser and Mobile App. It's how the system shows himself to the user.
- *Web and Business Tier* : represented by Web Server, which contains javascript and html code in order to create dynamic pages, and Application Server, which contains all the system's logic (the so called Business Logic).
- *Database Tier* : represented by DB Server, that contains and manages persistent data in an efficient way.

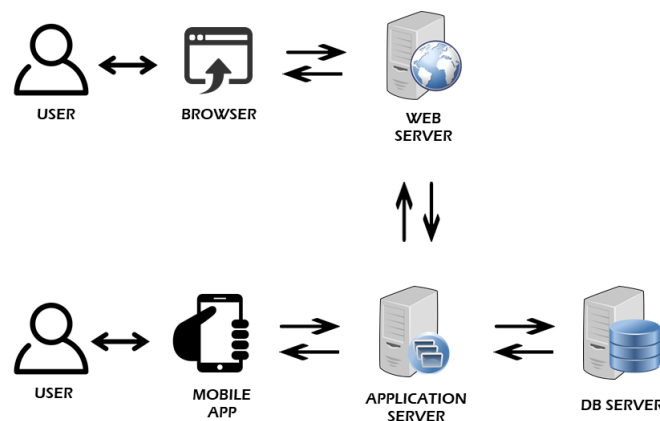


Figure 1: Proposed Architecture

2.2 High Level Components and Their Interactions

Here we have a proposed High Level Component Diagram

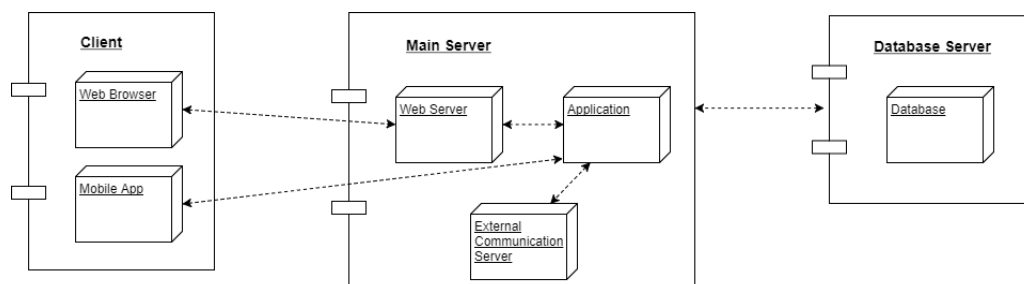


Figure 2: Components High Level

Analyzing this Diagram we can see :

- *On the Client Side* : The Mobile App of Travlendar+ for all users that uses a smartphone and has already downloaded it or the Web Browser for all the others.
- *On the Server Side* : The Web Server that, as told before, creates dynamic html and js pages, for the Web Browser, using data elaborated by the server's logic and the Application that is actually the server's logic. There is also a third component, the External Communication Server, that manages the communication of our server with the external ones, such as Google servers or Transport Service servers, in order to send and recieve informations and data from them.
- *On the Database Side* : The Database used to contain and manage all data and informations that our system needs to handle.

2.3 Component View

TEXT HERE

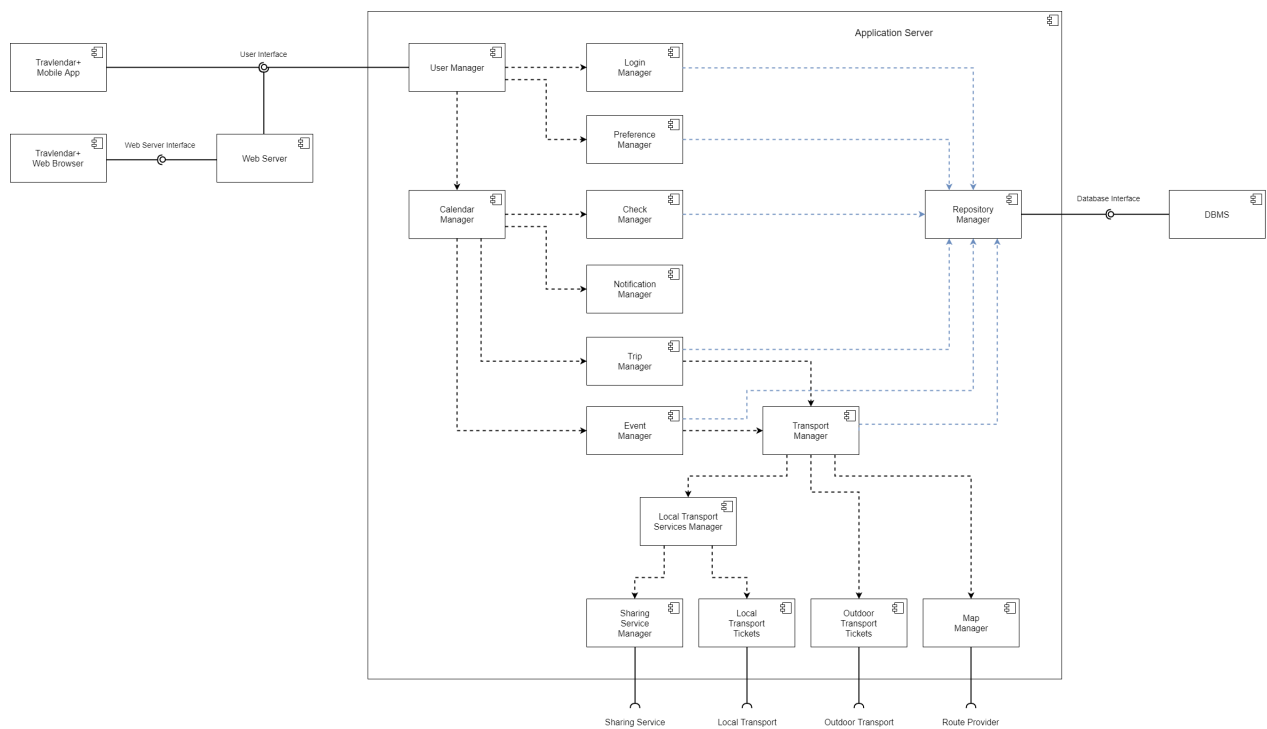


Figure 3: Component View

2.4 Deployment View

TEXT HERE

2.5 Runtime View

TEXT HERE

2.6 Selected Architectural Styles and Patterns

TEXT HERE

2.7 Other Design Decisions

TEXT HERE