

Politecnico di Milano
A.A. 2017-2018
Software Engineering II project
Travlendar+
Design Document
V1

Mirko Mantovani (893784), Matteo Marziali (893904)

November 18, 2017



Contents

1	Introduction	3
1.1	Purpose	3
1.2	Scope	4
1.3	Definition and Acronyms	5
1.3.1	Definitions	5
1.3.2	Acronyms	5
1.4	Revision	5
1.5	References	5
1.6	Document Structure	6
2	Architectural Design	7
2.1	Overview	7
2.2	High-Level components: general architecture identification	8
2.3	Component View	10
2.4	Deployment View	10
2.5	Runtime View	10
2.6	Component Interfaces	10
2.7	Selected architectural styles and patterns	10
2.8	Other design decisions	10
3	Appendix	11
3.1	Used software	11
3.2	Effort spent	11

1 Introduction

1.1 Purpose

TODO

1.2 Scope

TODO

1.3 Definition and Acronyms

1.3.1 Definitions

- **App:** this is the abbreviation for application, in particular this term is used meaning a mobile application.
- **Delay notification function:** this phrase refers to the function which allows to notify the participants of a meeting through an email in case the user is late.
- **Travel:** a travel is any suggested path that goes from the starting point to the meeting location.
- **Route:** this term is used as a synonym of travel.
- **Warning:** warning is the word used to define the conflict between two meetings.
- **Conflict:** a conflict between two or more meetings is what enables the creation of a warning, it means that the set of meetings in conflict are scheduled too close in time in order for the user to be able to attend them all in time.
- **Calendar:** the calendar contains the list of meetings and is grouped by day.
- **Meeting:** is an important keyword of the application, it includes all the informations of an appointment.
- **Reminder:** a reminder is a sort of an alarm triggered at a certain time before an appointment is starting.

1.3.2 Acronyms

- **API:** application programming interface; it is a set of routines, protocols, and tools for building software applications on top of this one.
- **JEE:** Java Enterprise Edition
- **EJB:** Enterprise Java Bean
- **JPA:** Java Persistence API

1.4 Revision

1.5 References

- The document with the assignment for the project
- The RASD document of Travlendar+

1.6 Document Structure

This document is structured in three parts:

- **Introduction:**
- **Architectural Design:**
- **Algorithm Design:**
- **User Interface Design:**
- **Requirements traceability:**
- **Implementation, integration and test plan:**

2 Architectural Design

2.1 Overview

This section of the document gives a detailed view of the physical and logical infrastructure of the system-to-be.

It provides the different types of view over the system as well as the description of the main components and their interactions.

A top down approach will be adopted for the description of the architectural design of the system:

Section 2.2 A description of high-level components and their interactions.

Section 2.3 A detailed insight of the components described in the previous section.

Section 2.4 A set of indications on how to deploy the illustrated components on physical tiers.

Section 2.5 A thorough description of the dynamic behaviour of the software with diagrams for the key-functionalities.

Section 2.6 A description of the different types of interfaces among the various described components.

Section 2.7 A list of the architectural styles, design patterns and paradigms adopted in the design phase.

Section 2.8 A list of all other relevant design decisions that were not mentioned before.

2.2 High-Level components: general architecture identification

The design approach is a JEE Architecture which is based on a client-server 4-tier distributed system.

Here we provide for each tier the definition, choice reasons and used technology:

- **Client Tier:** this tier is responsible of translating user actions and presenting the output of tasks and results into something the user can understand;
- **Web Tier:** it receives the requests from the client tier and forwards the pieces of data collected to the business tier waiting for processed data to be sent to the client tier.
Web Tier is composed by web beans. This tier purpose is the one to interact with the beans in the Business Logic tier and display data according to the user requests.
- **Business Logic Tier:** this tier contains the business logic, it coordinates the application, processes commands, makes logical decisions and evaluations and performs computations.
It is responsible for the communication between the Web Tier and the Persistence Tier. Its components are the EJB Beans.
- **Persistence Tier:** this tier holds the information of the system data model and is in charge of storing and retrieving information from the database.
The persistence tier is composed of the entity beans which represent the entities depicted from our RASD document and then further endorsed in our conceptual design. These entities are fundamental as they represent the connection to our database. Since in JEE we are interested in working in an object oriented environment, they represent a high level object view of the database.
In particular, for Travlendar+ it will be used a relational DBMS: MySQL, and the JPA standards of JEE in order to look and use the database entities in a object oriented way.

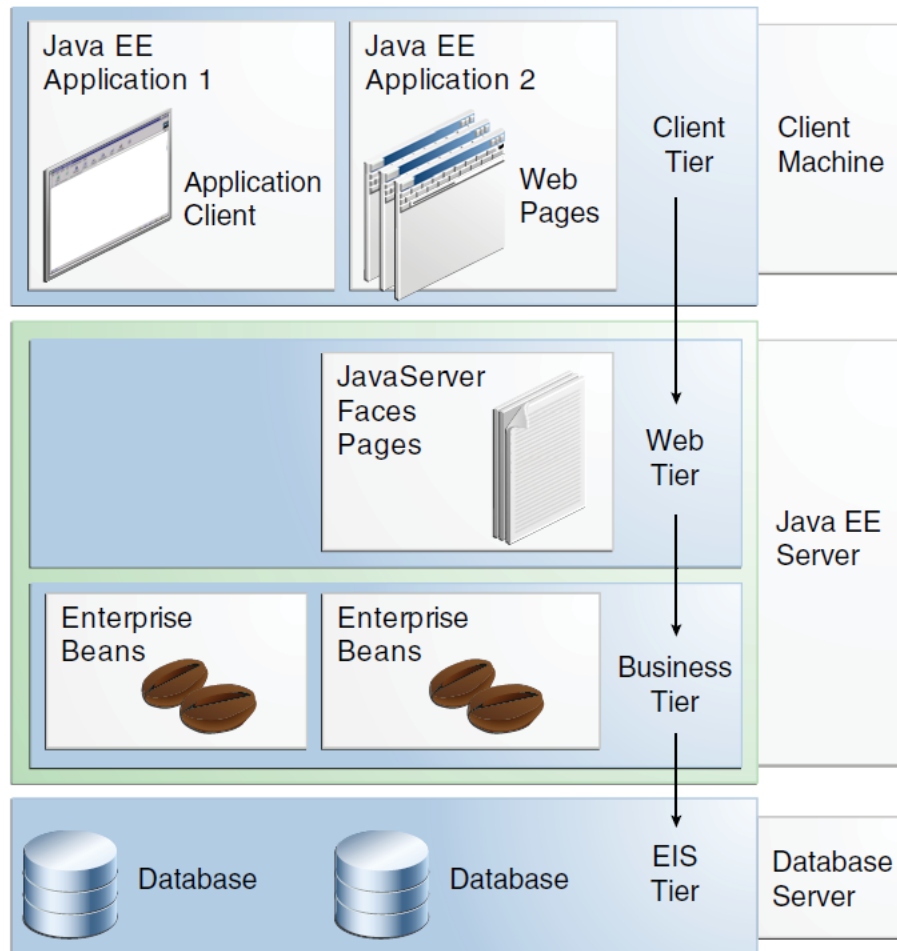


Figure 1: JEE architecture

- 2.3 Component View
- 2.4 Deployment View
- 2.5 Runtime View
- 2.6 Component Interfaces
- 2.7 Selected architectural styles and patterns
- 2.8 Other design decisions

3 Appendix

3.1 Used software

Task	Software
Edit and compile L ^A T _E X code	TeXmaker, TeXstudio
UML modelling	Astah Pro, Signavio
Compile and run Alloy	Alloy Analyzer 4.0
Mockup creation	Balsamiq Mockups 3

3.2 Effort spent

- Matteo Marziali working hours: \approx hours
- Mirko Mantovani working hours: \approx hours