



POLITECNICO
MILANO 1863

Travlendar+ project

Design Document

RICCARDO FACCHINI

ANDREA GUGLIELMETTI

October 29, 2017

Deliverable specific information

Deliverable:	Design Document
Title:	Requirement Analysis and Verification Document
Authors:	Riccardo Facchini - Andrea Guglielmetti
Version:	1.0
Date:	October 29, 2017
Download page:	https://github.com/Riccardo95Facchini/FacchiniGuglielmetti.git
Copyright:	Copyright © 2017, Riccardo Facchini - Andrea Guglielmetti – All rights reserved

Contents

Deliverable specific information	1
Table of Contents	2
List of Figures	3
List of Tables	4
1 Introduction	5
1.1 Purpose	5
1.2 Scope	5
1.3 Definitions, Acronyms, Abbreviation	5
1.4 Revision History	5
1.5 Reference Documents	5
1.6 Document Structure	5
2 Architectural Design	6
2.1 Overview	6
2.2 Component View	8
2.3 Deployment View	9
2.4 Runtime View	10
2.5 Component Interfaces	11
2.6 Selected architectural styles and patterns	12
2.7 Other design decision	13
3 Algorithm Design	14
4 User Interface Design	15
4.1 UserInterfaces	15
5 Requirements Traceability	16
5.1 requirements traceability	16
6 Implementation, Integration and Test Plan	17
6.1 implementation	17
7 Appendix	18
7.1 Effort Spent	18
7.2 References	18

List of Figures

1	Client Server architecture	6
2	Overview of the system architecture	7

List of Tables

1 Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, Abbreviation

1.4 Revision History

1.5 Reference Documents

1.6 Document Structure

2 Architectural Design

2.1 Overview

We need to design a system in which the user asks to the system to store an appointment and calculate the best path from a starting location to the appointment location.

Since this interaction between user and system can be summarize as:

1. User request a service to the system.
2. System responds to the user with the requested service.

Based on this, we decide to use a client-server architectural approach.

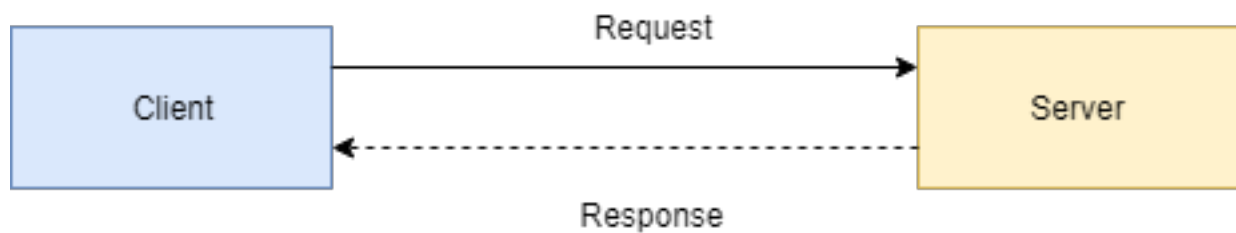


Figure 1: Client Server architecture

Furthermore, the system can be divided into three different subsystems: the presentation layer, the application layer and the data layer as we can see in [Figure 2](#).

- The *Presentation Layer* provides the GUI of the system. This layer contains the mobile application and the web pages.
- The *Application Layer* contains the logic of the application, that receives the requests from the user, computes the best path to reach the appointment, checks the weather and the road conditions and executes the dynamic web pages of the web site.
- The *Data Layer* stores and maintains the data needed from the system to works properly, i.e. user's information and user's appointment information.

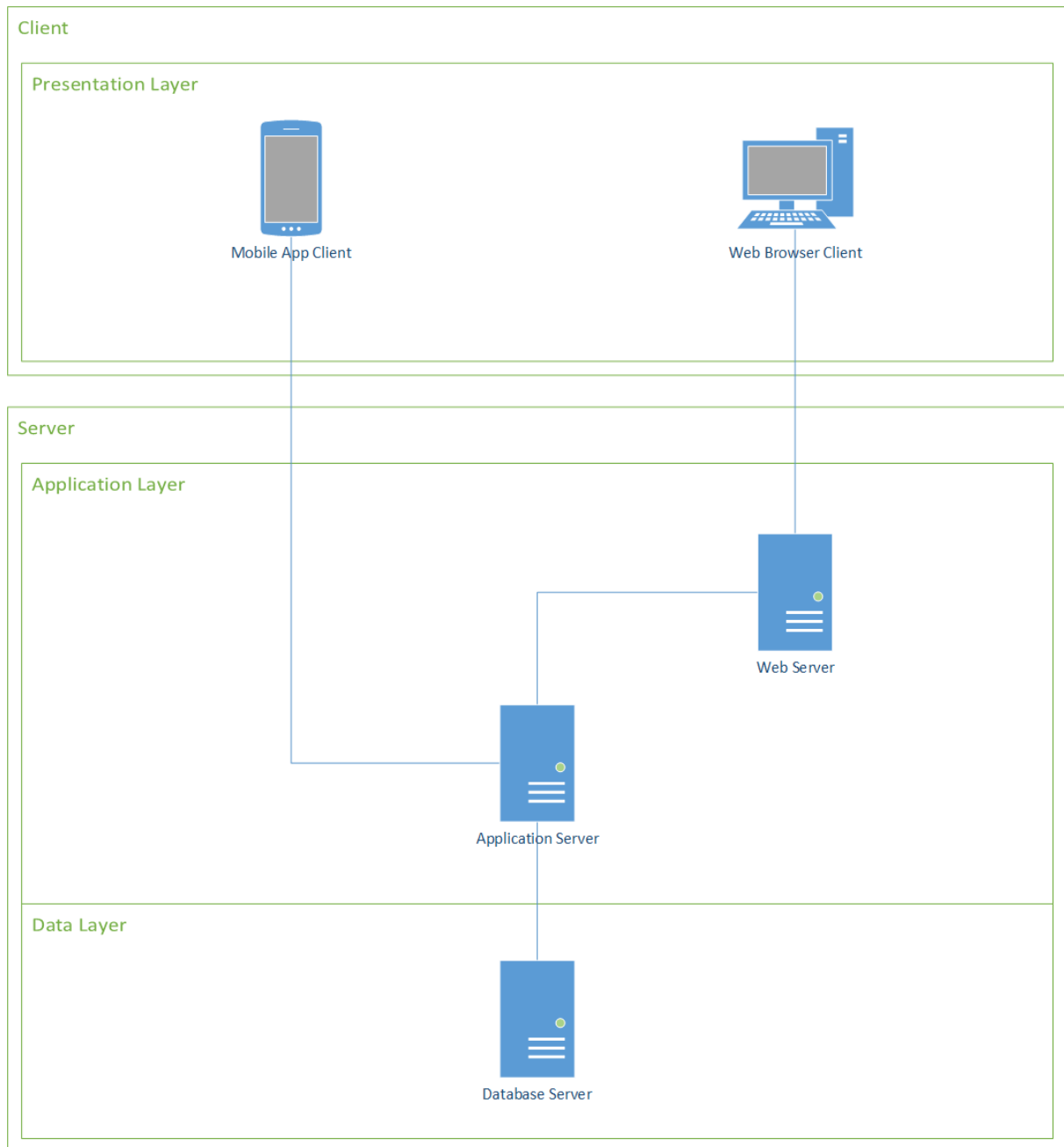


Figure 2: Overview of the system architecture

2.2 Component View

2.3 Deployment View

2.4 Runtime View

2.5 Component Interfaces

2.6 Selected architectural styles and patterns

2.7 Other design decision

3 Algorithm Design

Algorithm

4 User Interface Design

4.1 UserInterfaces

5 Requirements Traceability

5.1 requirements traceability

6 Implementation, Integration and Test Plan

6.1 implementation

7 Appendix

7.1 Effort Spent

7.2 References