

POLITECNICO DI MILANO



Corso di Laurea Magistrale in Computer Science and Engineering  
Dipartimento di Elettronica e Informazione

---

# Travlendar+

## *Design Document*

---

Reference professor:  
Prof. Elisabetta Di Nitto

Authors:  
Alessandro Aimi Matr. 899642  
Roberto Bigazzi Matr. 899411  
Filippo Collini Matr. 829918

*Version 1.0.0*  
*26/11/2017*  
Academic Year 2017-2018

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.A	Description of the given problem . . . . .	1
1.B	Definitions and Acronyms . . . . .	2
1.B.1	Definitions . . . . .	2
1.B.2	Acronyms . . . . .	2
1.C	Revision History . . . . .	2
1.D	References . . . . .	2
1.E	Document Structure . . . . .	2
1.F	Used tools . . . . .	3
<b>2</b>	<b>Architectural Design</b>	<b>4</b>
2.A	Overview . . . . .	4
2.B	Component View . . . . .	5
2.C	Deployment View . . . . .	6
2.D	Runtime View . . . . .	6
2.E	Component Interfaces . . . . .	11
2.F	Selected Architectural Styles and Patterns . . . . .	11
2.G	Other Design Decisions . . . . .	11
<b>3</b>	<b>Algorithm Design</b>	<b>12</b>
<b>4</b>	<b>User Interface Design</b>	<b>13</b>
4.A	Mockups . . . . .	13
4.B	UX Diagrams . . . . .	22
4.C	BCE Diagrams . . . . .	24
<b>5</b>	<b>Requirements Traceability</b>	<b>25</b>
<b>6</b>	<b>Implementation, Integration and Test Plan</b>	<b>26</b>
<b>7</b>	<b>Effort Spent</b>	<b>27</b>

# Chapter 1

## Introduction

This document is the Design Document (DD) of a mobile application called Travlendar+. It is mainly addressed to the software development team and its purpose is to provide an overall view of the architecture of the new system. The document defines:

- High level architecture;
- Used design patterns;
- Main components of the system;
- Runtime behavior.

### 1.A Description of the given problem

Travlendar+ is a mobile, calendar-based application that helps the user to manage his appointments and to a greater extent set up the trip to his destination, choosing the best means of transport depending on his needs. Travlendar+ will choose the most suitable way to get the user to his destination between a large pool of options, considering public transportation, personal vehicles, locating cars or bikes of sharing services and walking to the destination. It will take account of weather, traffic, possible passengers if any, the user-set break times and the potential will to minimize the carbon footprint of the trip, always focusing on taking him on time to his scheduled appointments.

Eventually the user will be able to purchase the tickets he will use to reach his destination in-app. The great customizability is one of the main strengths of Travlendar+, being able to fully comply with the user needs.

## 1.B Definitions and Acronyms

### 1.B.1 Definitions

### 1.B.2 Acronyms

List of the acronyms used in this paper:

- RASD: Requirements analysis and specification document;
- DD: Design document;
- RMI: Remote Method Invocation;
- API: Application Programming Interface;
- UX: User experience;
- BCE: Boundary Control Entity;
- UI: User interface.

## 1.C Revision History

- 26/11/2017 Version 1.0.0 - First complete drawing up of the DD;

## 1.D References

Documents list:

- Mandatory Project Assignments.pdf

## 1.E Document Structure

The paper is structured as follows:

- Chapter 1: Introduction to the document, including some information about its composition and the description of the problem;
- Chapter 2: This chapter shows the main components of the system and the relationships between them. It also explains the main architectural styles and patterns adopted in the design of the system;

- Chapter 3: This chapter explains how the system will work using algorithms. Java code is used to write down the most significant algorithms for the application;
- Chapter 4: This chapter shows mockups of the application and more details about the User Interface using UX and BCE diagrams;
- Chapter 5: This chapter explains how the decisions taken in the RASD are associated to design decisions;
- Chapter 6: This chapter describes the order of implementation of the subcomponents of the system and the order in which integrate such subcomponents and test the integration;
- Chapter 7: Effort spent by the authors to draw up the document.

## 1.F Used tools

The tools used to create this document are:

- Photoshop for mockups;
- Draw.io for diagrams;
- Atom for algorithms drawing up;
- Github as version controller and to share documents;
- LaTeX for typesetting this document;
- Texmaker as editor;

# Chapter 2

## Architectural Design

### 2.A Overview

Architectural design is of crucial importance in software engineering because it will have to take account of functional and non-functional requirements, to meet the stakeholders needs and requests, and to help not to focus only on standalone elements losing the so called big picture of the system, always adhering to general principles of good quality. An important aspect is in fact to find a good trade-off between the high-level description near to the analysis and the low-level one near to the implementation. Coming up with good quality design and architecture is mostly a matter of experience and in our field, is also known the importance of the reusability of other's people work. So, we tried to build our system with various kind of this patterns and known architectural styles.

## 2.B Component View

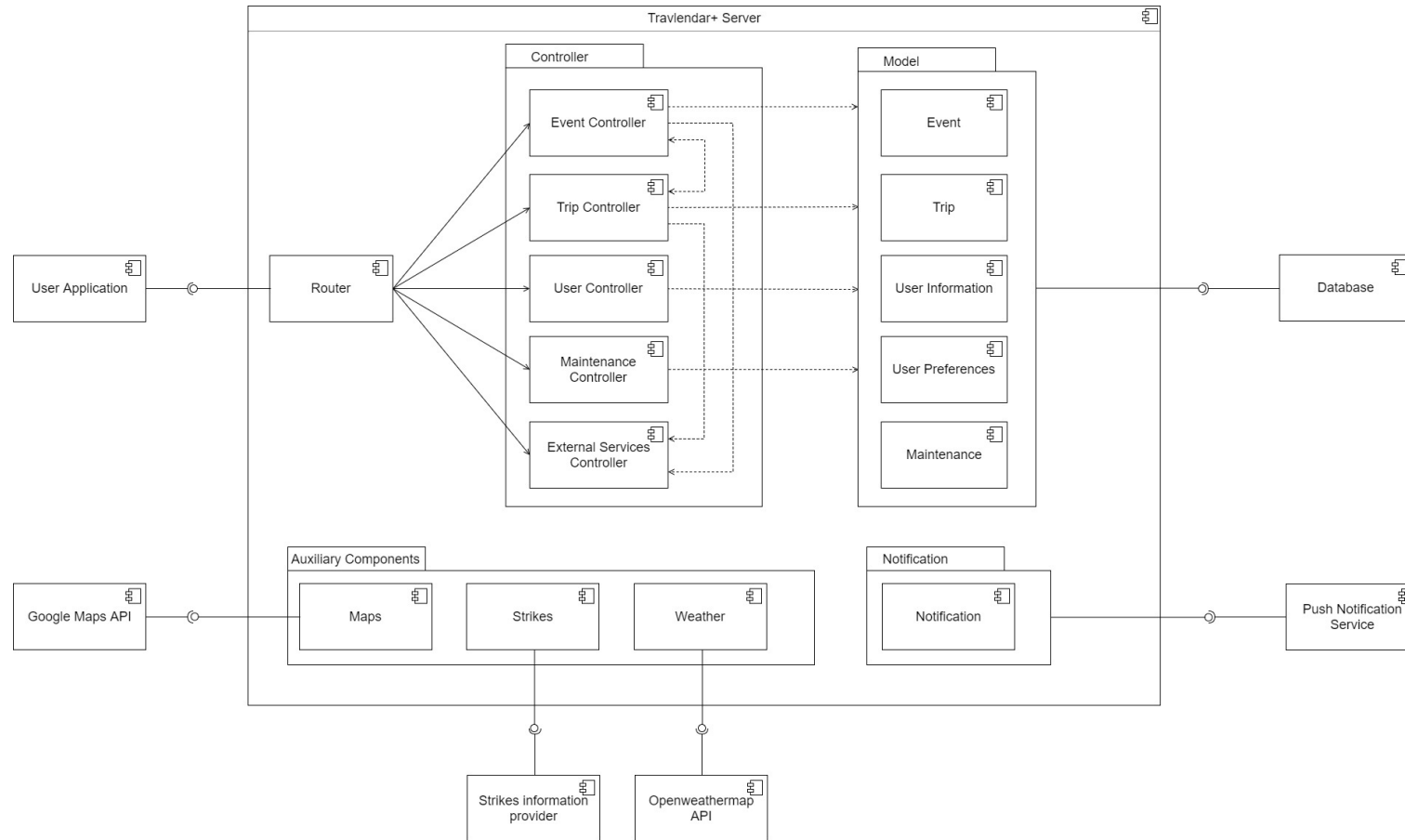


Figure 2.1: Component View of the system

## **2.C Deployment View**

## **2.D Runtime View**

In this section some sequence diagrams will be presented to describe the interactions that happen between the main components of the system when the most common functionalities are used. This is a high-level description of the actual interactions of the system-to-be, so functions and their names may be added, modified or deleted during the development process.



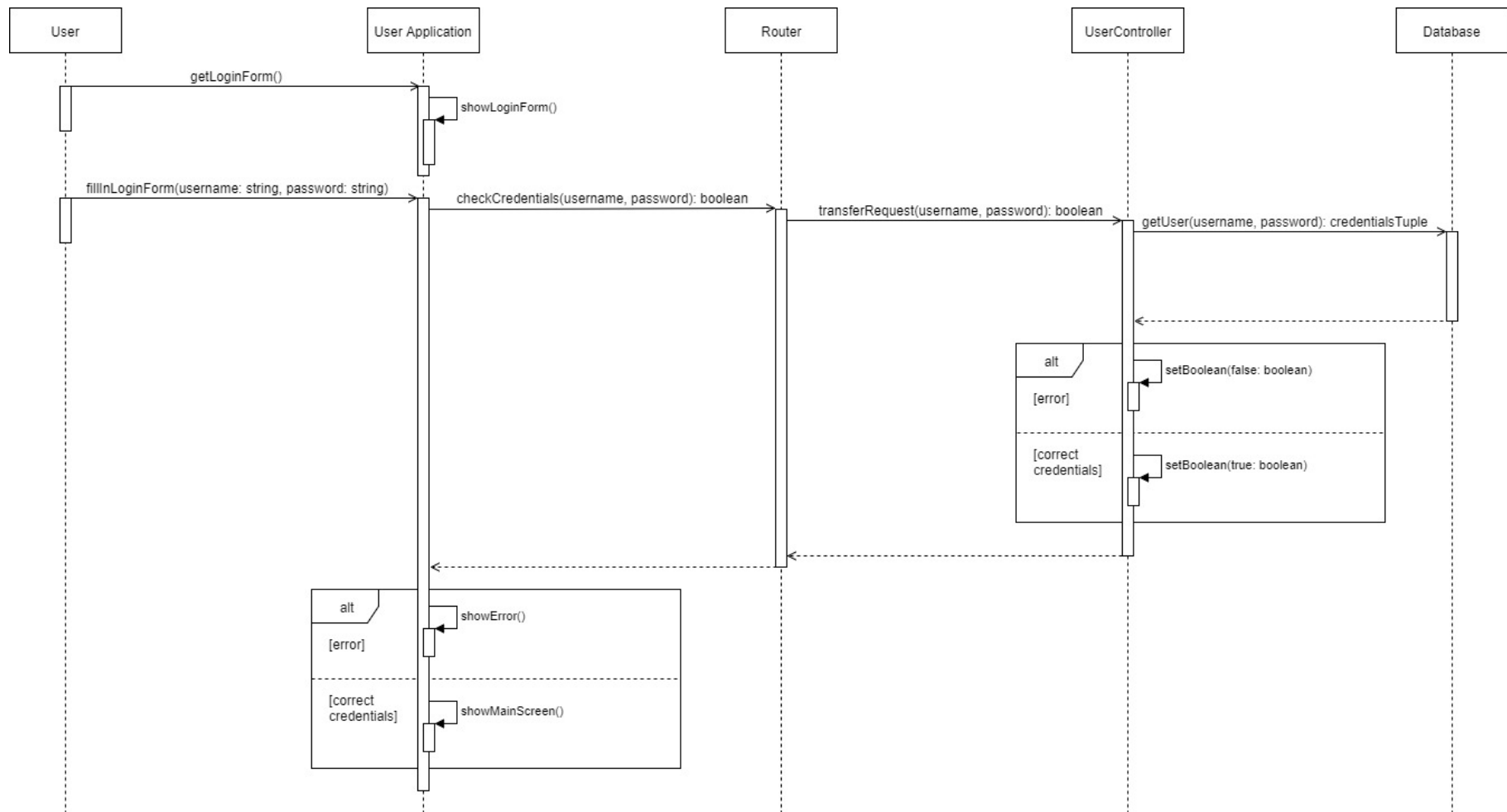


Figure 2.2: User Login runtime view

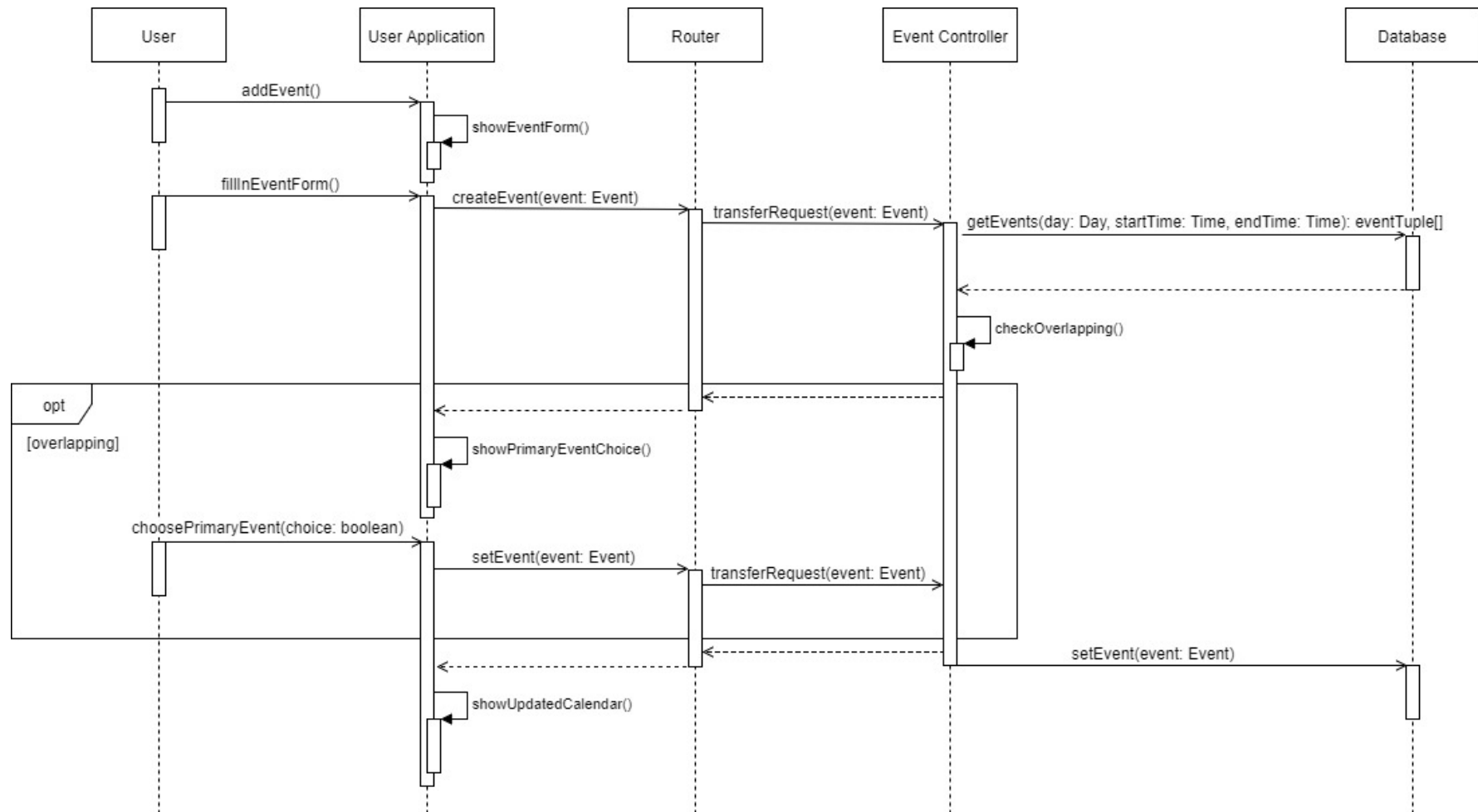


Figure 2.3: Adding of an event in the calendar runtime view

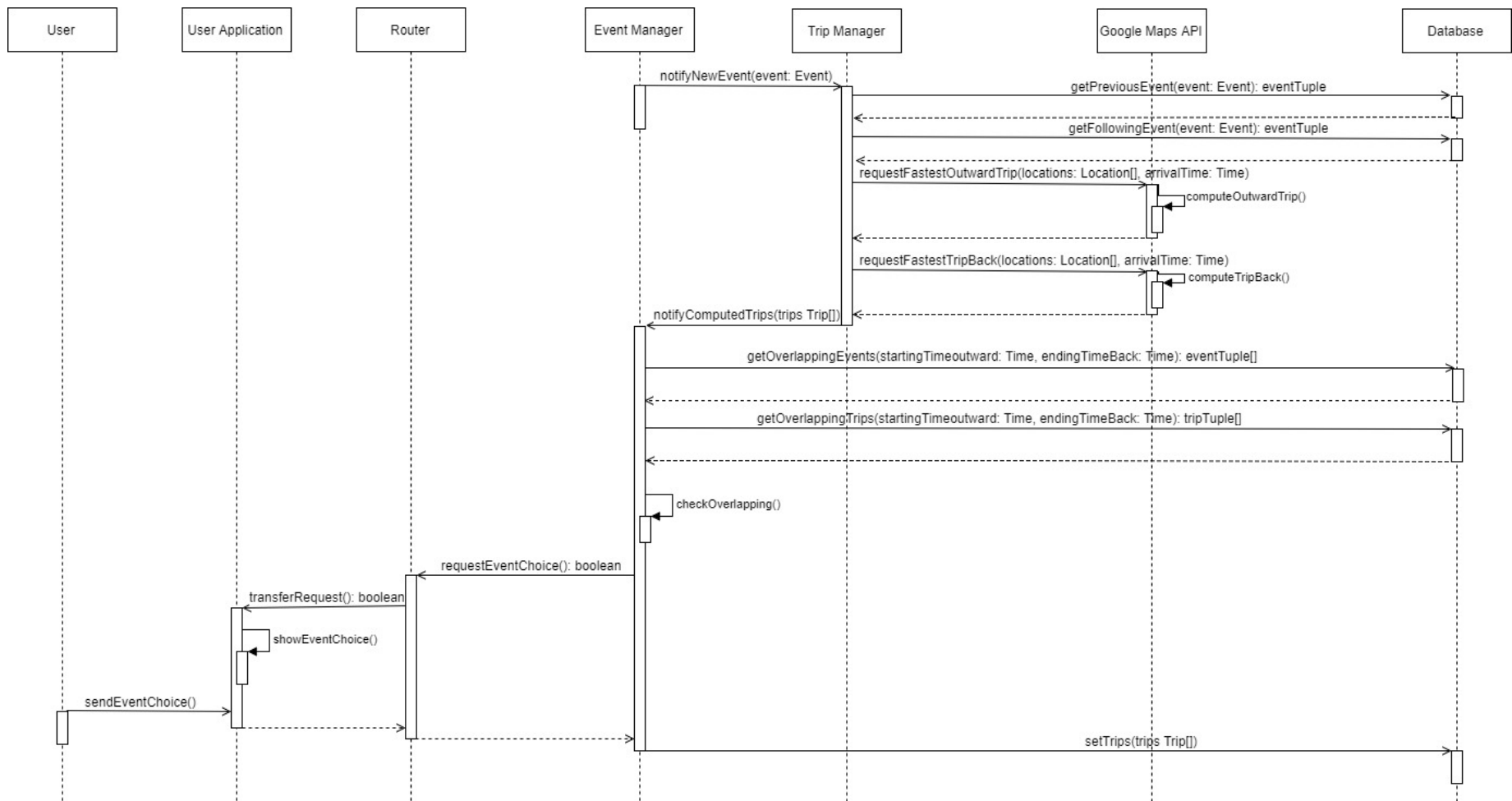


Figure 2.4: Trip planning runtime view

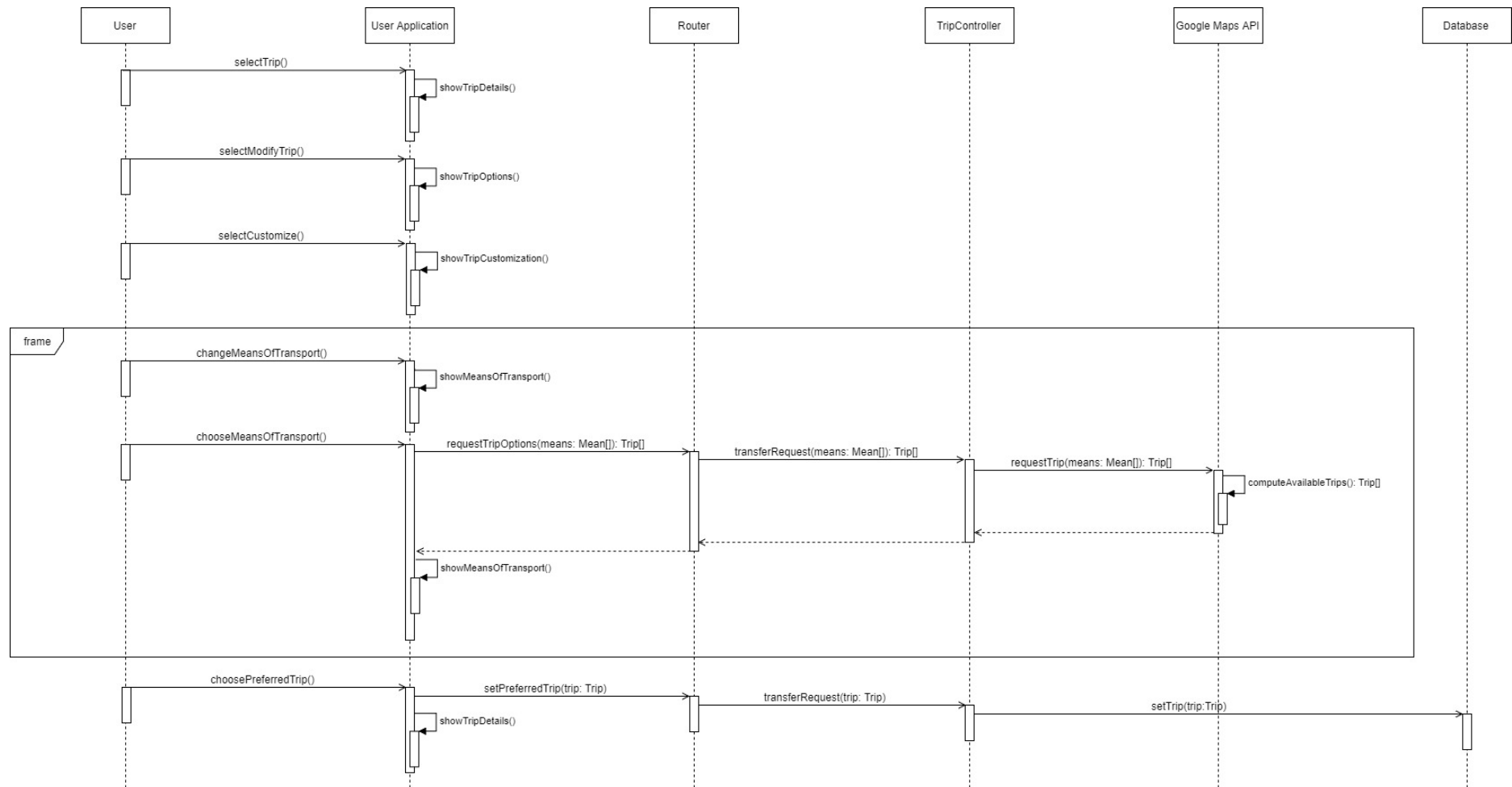


Figure 2.5: Trip customization runtime view

**2.E Component Interfaces**

**2.F Selected Architectural Styles and Patterns**

**2.G Other Design Decisions**

# Chapter 3

## Algorithm Design

# Chapter 4

## User Interface Design

This section is a recapitulation of the section 3.B.1 (User Interfaces) of the Requirements Analysis and Specification Document and a deepening of design aspects of the user interface. The application will be developed as a mobile application for the main mobile operating systems (iOS and Android). As the system will appear the same for all the users, it will provide all the functionalities described in the RASD, in a unique user interface.

### 4.A Mockups

Following some mockups will provide an idea of the user interface while the user interacts with it, making use of the functionalities of the application. It includes most of the main screen that the user will face.

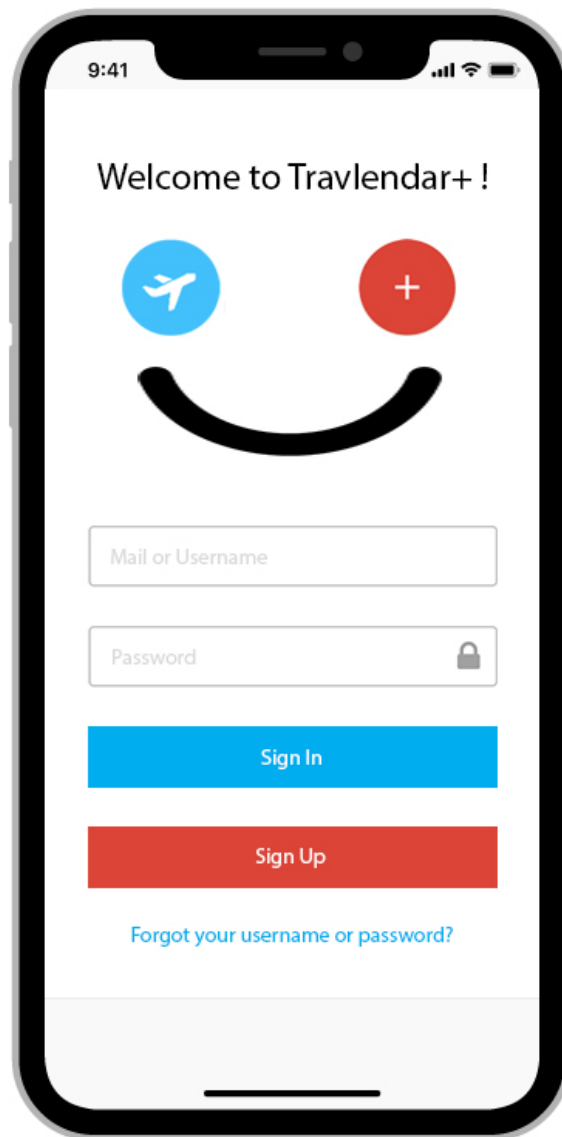
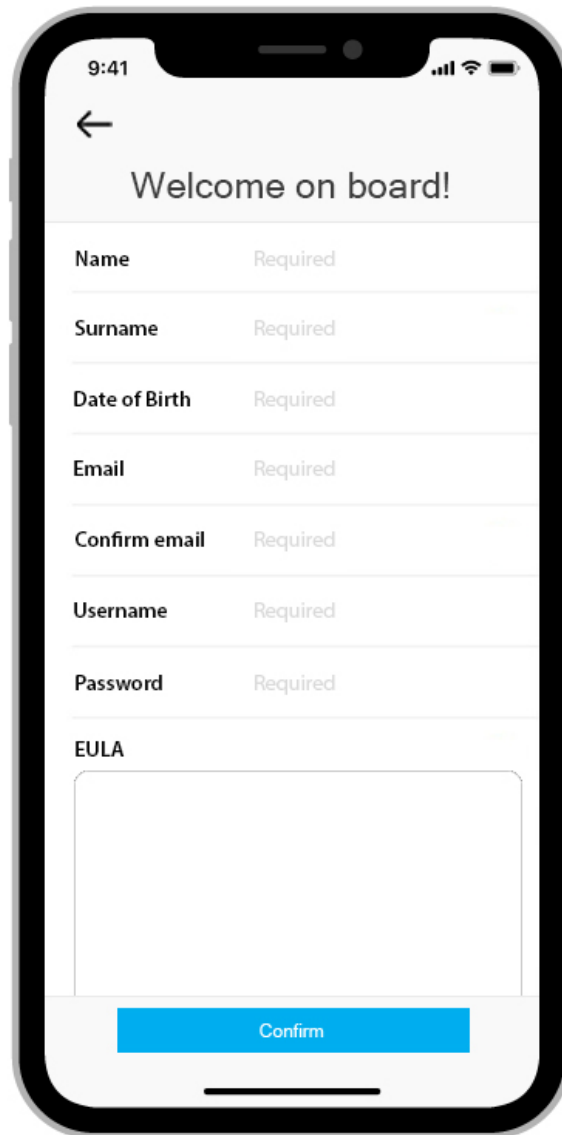


Figure 4.1: Mockup of the login screen





The mockup shows a mobile application registration screen. At the top, the status bar displays the time 9:41, signal strength, Wi-Fi, and battery icons. Below the status bar is a navigation bar with a back arrow icon. The main heading is "Welcome on board!". The form consists of several input fields, each with a label and a "Required" status: Name, Surname, Date of Birth, Email, Confirm email, Username, and Password. Each field has a yellow "Required" label to its right. Below these fields is a section labeled "EULA" with a large, empty rectangular box for the user to accept the terms. At the bottom of the screen is a blue "Confirm" button.

Field	Required
Name	Required
Surname	Required
Date of Birth	Required
Email	Required
Confirm email	Required
Username	Required
Password	Required

EULA

Confirm

Figure 4.2: Mockup of the registration screen

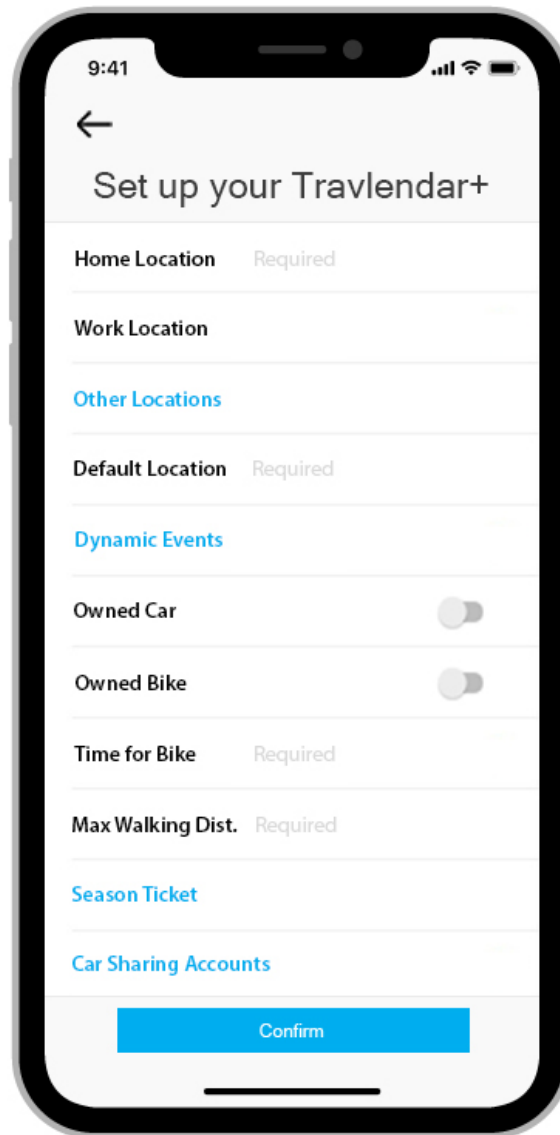


Figure 4.3: Mockup of the screen where the user can set his preferences

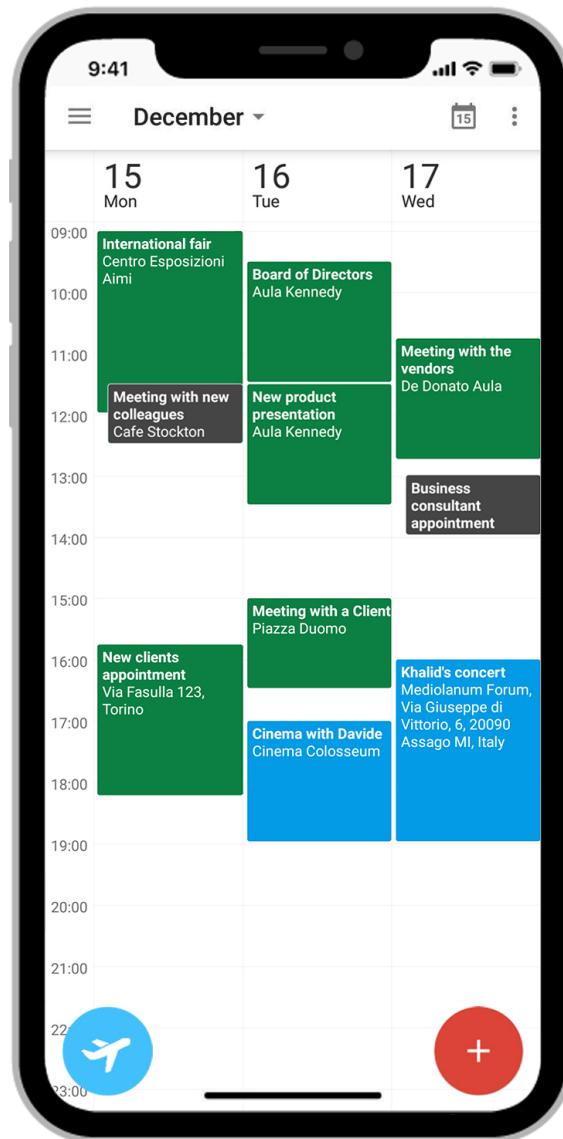


Figure 4.4: Mockup of the calendar screen with primary and secondary events

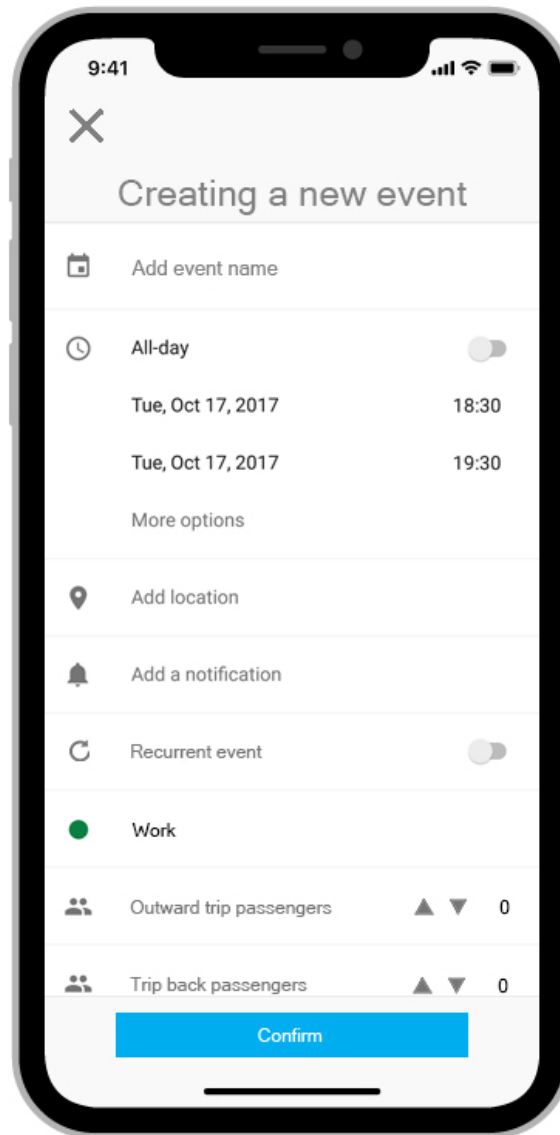


Figure 4.5: Mockup of the screen that allows to add an event



Figure 4.6: Mockup of the screen with the trips shown

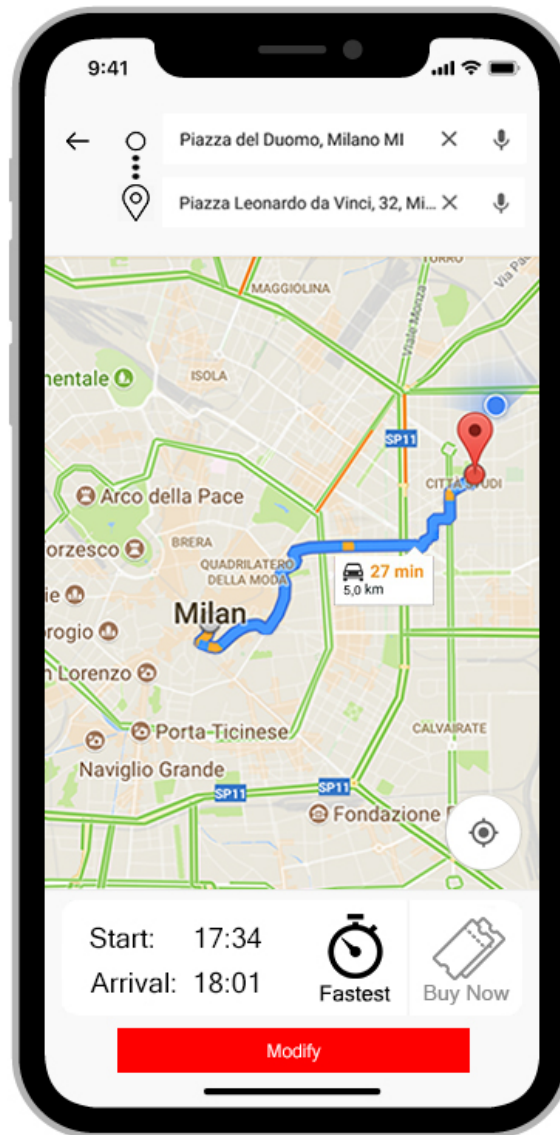


Figure 4.7: Mockup of the trip details screen

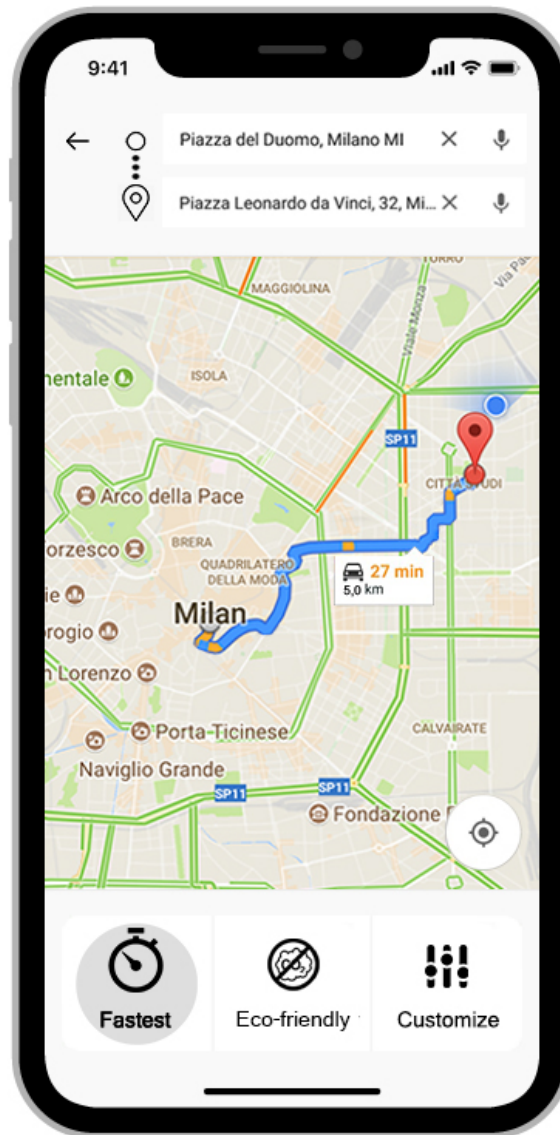


Figure 4.8: Mockup of the screen that allows the user to modify a trip

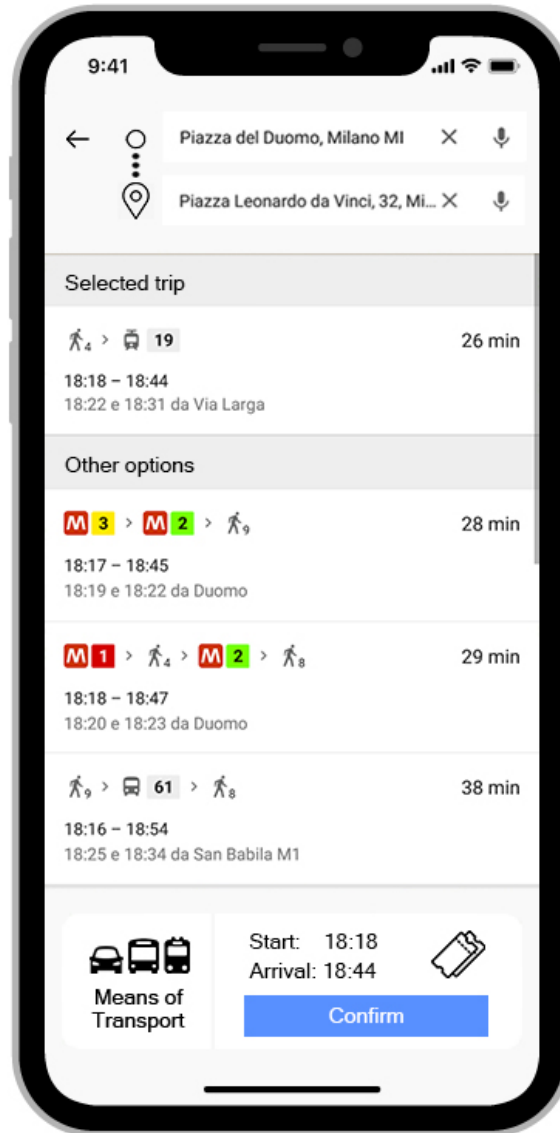


Figure 4.9: Mockup of the screen with the customization of a trip

## 4.B UX Diagrams

UX diagrams provide information about the user interface of the system and how the user interacts with it. For the diagram comprehension purposes, additional screens used to add specific information (other locations, dynamic



events, season tickets, etc...) in the preferences setup are not considered.

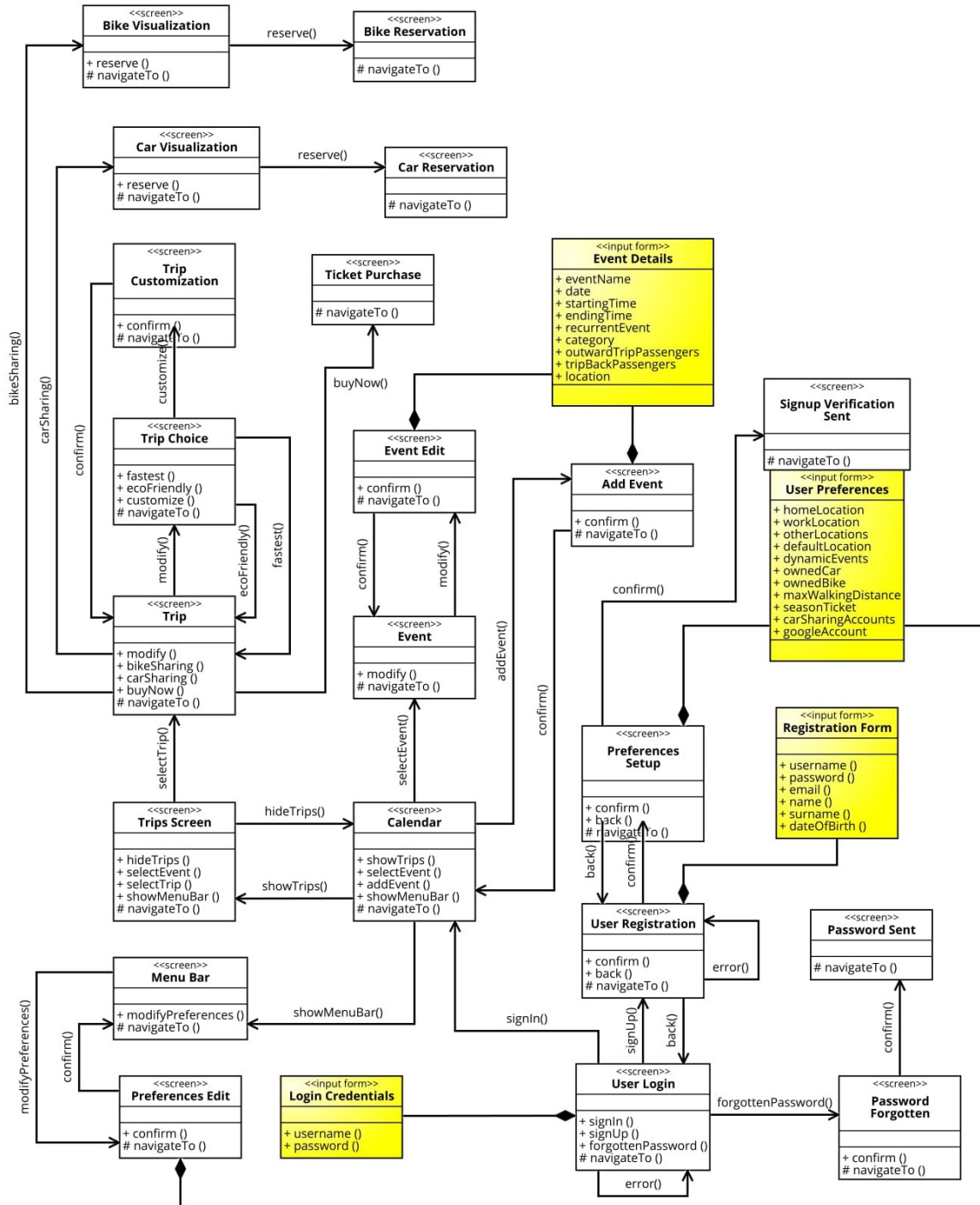


Figure 4.10: UX diagram of the application

## 4.C BCE Diagrams

For the implementation of the system, a Model-View-Controller design pattern is adopted and BCE diagrams are useful to show how user interactions are managed internally by the system. Boundaries are objects that interface with the users of the application; Entities object model the access to data; Controls object manage the communication between boundaries and entities.

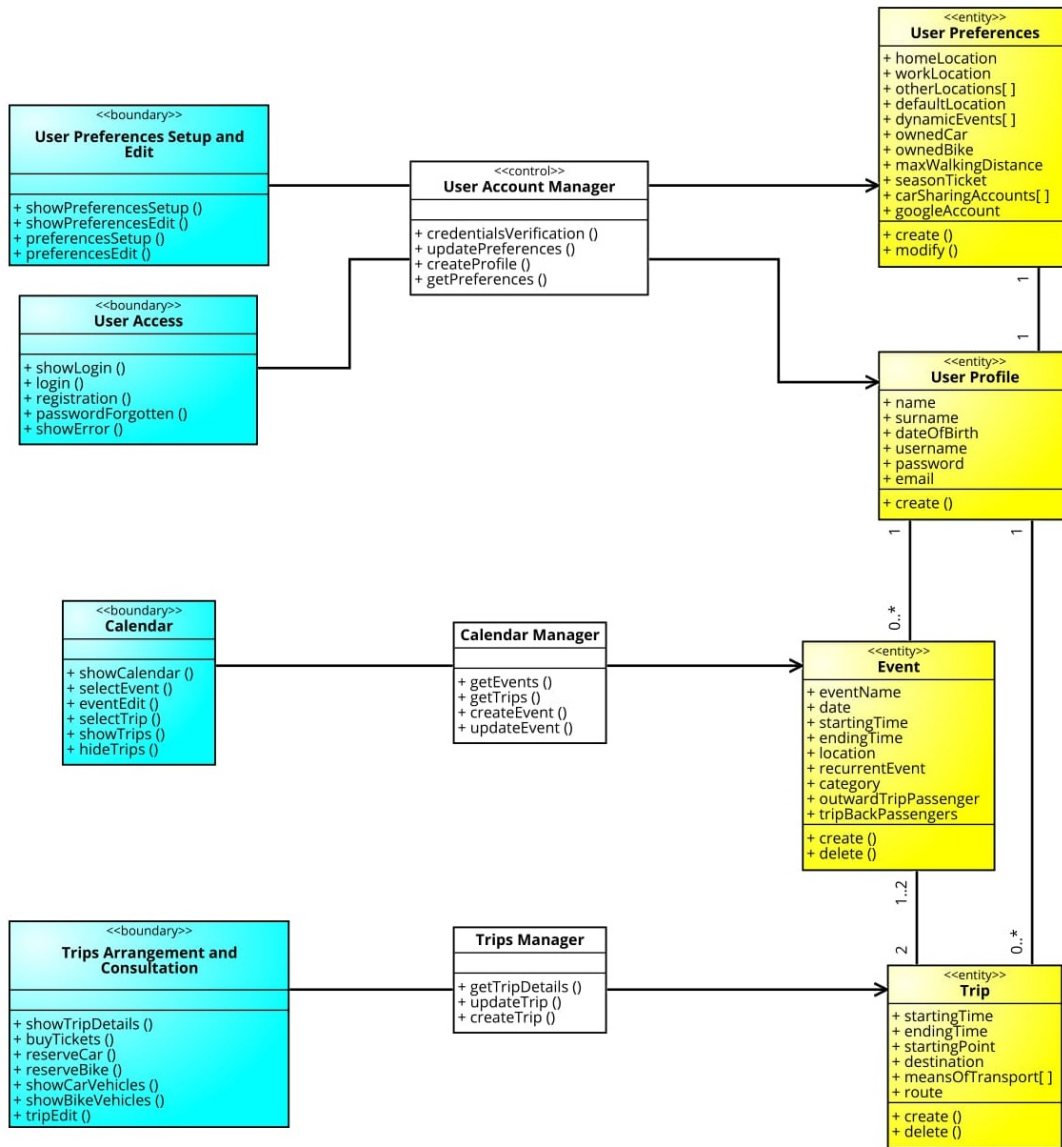


Figure 4.11: BCE diagram of the application

## Chapter 5

# Requirements Traceability

## Chapter 6

# Implementation, Integration and Test Plan

## Chapter 7

### Effort Spent