



**POLITECNICO**  
MILANO 1863

# Travlendar+

## Integration Test Plan Document

Bolshakova Liubov  
Campagnoli Chiara  
Lagni Luca

Software Engineering 2  
November 24, 2017

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Objective . . . . .	1
1.2	Meanings . . . . .	1
<b>2</b>	<b>Integration Strategy</b>	<b>1</b>
2.1	Entry Point Classes . . . . .	1
2.2	First Mixture . . . . .	2
2.3	Core Application Classes . . . . .	3
2.4	Second Mixture . . . . .	3
2.5	Fine-Grained Classes . . . . .	4
2.6	Final Mixture . . . . .	5

## 1 Introduction

### 1.1 Objective

The purpose of the Integration Test Plan Document (ITPD) is to schedule and present to the team disposed for testing the application (and its infrastructure) the required and logic sequence of tests to be applied to different components and interfaces that compose our application.

We do this in order to have a confirm that the components and interfaces proposed in the DD behave in a correct and predictable manner and that the interactions between the components act as we expect.

### 1.2 Meanings

## 2 Integration Strategy

### 2.1 Entry Point Classes

This subsection is concerning the entry points of our application or rather, the first components that are involved in the interaction between the external environment and our application.

This is the first thing to do because, if we want to test properly our components and interfaces, we must be sure that the initial data, that will be provided by the user, will be processed correctly and will feed other objects, related to these ones, with correct and appropriate data.

These classes are:

#### 1. Coordinate

- ExternalCompany
- Technician
- User
- UserDevice

We have decided to start from these classes because they're the ones that must exist and that it's not necessary for them to be associated to other classes (except in the case of coordinate).

We want to specify that these classes, only by their own, are useless, but they're necessary for the interaction between the application and the agents.

The Coordinate is the first one to be tested, because it's the only one which is used also by some others.

For the other classes the order is not important.

All the methods of these classes should be tested (also the getters, because they're used also inside other methods of the classes).

### 2.2 First Mixture

At this stage is required to test the Entry classes that are related in order to see if they cooperate correctly.

Caller class	Called class
Coordinate	UserDevice
Coordinate	User
User	UserDevice

## 2.3 Core Application Classes

This subsection is concerning the core of our application.

With that we mean all the classes that, related to the entry point classes , can compose a functional and finite (but too simplified and unprecise ) application.

These classes are essential for our application but they can be obtained only by an interaction between the external environment and our entry points.

These classes are:

1. MeanOfTransport
2. Navigator
3. Trip
4. Meeting
5. Notification
6. NotificationManager
7. TechnicalProblem
8. SystemShared

In this case the order is important because all the classes are related.

## 2.4 Second Mixture

At this stage is required to test how core classes are related to each others and with entry classes in order to test if the basic behaviour of the application is correct.

Caller class	Called class
MeanOfTransport	Coordinate
MeanOfTransport	ExternalCompany
Navigator	MeanOfTransport
Navigator	Trip
Trip	Coordinate
Trip	Meeting
Meeting	Coordinate
Meeting	User
Notification	ExternalCompany
Notification	NotificationManager
NotificationManager	MeanOfTransport
NotificationManager	Navigator
NotificationManager	Trip
User	TechnicalProblem
TechnicalProblem	Technician
SystemShared	User
SystemShared	ExternalCompany
SystemShared	UserDevice
SystemShared	NotificationManager

## 2.5 Fine-Grained Classes

This subsection is concerning classes that are specifications of other classes and are useful to provide a better application from the point of view of features and behaviours

These classes are (most of them) subclasses that adapt a behaviour to a specific situation.

These classes are:

- Visitor
- Client
- UserPreferences
- WindowsPhoneWrapper
- IOSWrapper
- AndroidWrapper
- PathRestriction
- OAMOTRestriction
- OtherTypeOfRestriction
- Break
- StrikeNotification
- WeatherNotification
- OtherNotification
- ExternalMeanOfTransportCompany
- Feet
- Bike
- Boat
- ONAMOT
- Ride
- Car
- Ship
- Tram
- Bus
- Train
- OAMOT
- AdditionalOAMOTData

In this case all the classes can be tested in a random order.

## 2.6 Final Mixture

Caller class	Called class
UserDevice UserDevice UserDevice	WindowsPhoneWrapper IOSWrapper AndroidWrapper
Visitor Visitor Visitor Visitor	TechnicalProblem Meeting UserPreferences UserDevice
Client Client Client Client	TechnicalProblem Meeting UserPreferences UserDevice
OAMOTRestriction OAMOTRestriction	AdditionalOAMOTData Trip
OtherTypeOfRestriction	Trip
Break	Trip
ExternalCompany ExternalCompany	WeatherNotification OtherNotification
ExternalMeanOfTransportCompany ExternalMeanOfTransportCompany ExternalMeanOfTransportCompany ExternalMeanOfTransportCompany ExternalMeanOfTransportCompany ExternalMeanOfTransportCompany ExternalMeanOfTransportCompany ExternalMeanOfTransportCompany ExternalMeanOfTransportCompany ExternalMeanOfTransportCompany ExternalMeanOfTransportCompany	StrikeNotification OtherNotification NotificationManager Bike ONAMOT Boat Car Tram Bus Train Taxi OAMOT
AdditionalOAMOTData AdditionalOAMOTData	Car OAMOT
Feet Feet Feet Feet	Coordinate WeatherNotification OtherNotification NotificationManager
Bike Bike Bike Bike	Coordinate WeatherNotification OtherNotification NotificationManager
Navigator Navigator Navigator Navigator Navigator Navigator Navigator Navigator Navigator Navigator Navigator	Feet Bike Boat ONAMOT Car Tram Ship Bus Train Taxi OAMOT

Caller class	Called class
Navigator Navigator Navigator	WeatherNotification OtherNotification StrikeNotification
Boat Boat Boat Boat Boat Boat	Coordinate WeatherNotification OtherNotification StrikeNotification NotificationManager Ride
ONAMOT ONAMOT ONAMOT ONAMOT ONAMOT ONAMOT	Coordinate WeatherNotification OtherNotification StrikeNotification NotificationManager Ride
Car Car Car Car Car Car Car	Coordinate WeatherNotification OtherNotification StrikeNotification NotificationManager AdditionalAMOTData Ride
Ship Ship Ship Ship Ship Ship Ship	Coordinate WeatherNotification OtherNotification StrikeNotification NotificationManager Ride
Tram Tram Tram Tram Tram Tram	Coordinate OtherNotification StrikeNotification NotificationManager Ride
Train Train Train Train Train Train	Coordinate OtherNotification StrikeNotification NotificationManager Ride
Taxi Taxi Taxi Taxi Taxi Taxi	Coordinate OtherNotification StrikeNotification NotificationManager Ride
Bus Bus Bus Bus Bus Bus	Coordinate OtherNotification StrikeNotification NotificationManager Ride



Caller class	Called class
OAMOT	Coordinate
OAMOT	WeatherNotification
OAMOT	OtherNotification
OAMOT	StrikeNotification
OAMOT	NotificationManager
OAMOT	AdditionalMOTData
OAMOT	Ride
SystemShared	Visitor
SystemShared	Client