POLITECNICO
MILANO 1863

Acceptance Test Document

Travlendar+

Federico Betti - Tommaso Bianchi

Deliverable:	ATD
Title:	Acceptance Test Document
Authors:	Federico Betti - 899914 Tommaso Bianchi - 894183
Version:	1.0
Date:	January 12, 2018
Download page:	Github
Copyright:	Copyright © 2018, Federico Betti - Tommaso Bianchi All rights reserved

Contents

List of Tables

1 Introduction

1.1 Purpose

This document is an Acceptance Test Document (ATD). Its main purpose is to control and test the project presented in Tested Project section. We proceeded following the installation instructions reported on the ITD document of the tested project, passing through system and model testing, code checking and user experience evaluation. All the work we made on the test project is presented in this document, divided in proper sections.

1.2 Tested Project

We are going to test the Travlendar+ project made by Ennio Nasca and Pierluca D'Oro that you can find here. We have considered their Implementation and Testing Document (ITD) for the installation and evaluation process and their Requirement Analysis and Specification Document (RASD) to compare their result with the planned functionalities, requirements and goals.

1.3 Document Structure

The DD is organized into 6 main sections:

1. **Introduction**: this section contains an overview on the purpose of this document, toghether with references to the tested project and the related documents.
2. **Installation Process**: in this section there is the review we made on the installation process following the guide that the ITD of the tested project gives.
3. **User Experience**: in this section there is the review we made on the User Experience of the project. We have tried the system extensively navigating all its pages from the point of view of a user.
4. **Testing**: this section describes which types of tests have been made on the application.
5. **Conclusions**: this section contains a brief summary of our thoughs about the project we tested.

2 Installation Process

To install and configure the project we have closely followed the instructions given in Chapter 4 of their ITD. To have a running version of python 3.6, with pip and all needed dependencies, on a fresh Ubuntu 16.04 Xenial machine, we needed some additional work not reported on the document, so we think that the Prerequisite section of the ITD should have been expanded more.
In particular, to install a working version of Django we had to invoke the following console commands:

```
$> sudo apt-get update
$> sudo apt-get upgrade python3
$> sudo apt-get install python3-pip
$> sudo pip3 install --upgrade setuptools
$> cd path/to/Implementation
$> sudo pip3 install -r requirements.txt
```

After that, we had problems in running the Django server from the "I&T/Implementation" folder, while everything worked perfectly in the folder extracted from the "implementation2017_01_07.rar" archive; for the rest of the tests, we kept using the code from the archive.

We also successfully installed the project on a Windows 10 machine using the python3 installer found here, and then typing

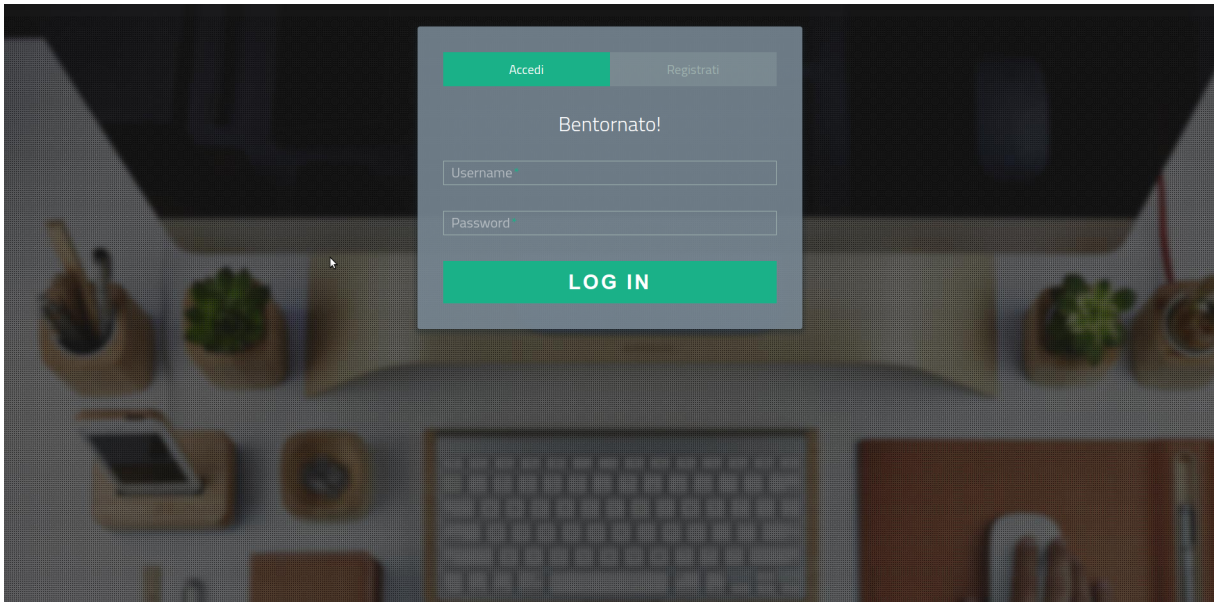
```
$> pip install -r requirements.txt
```


3 User Experience

In this chapter we present our impressions on the User Experience of the tested project. These are opinions from a user point of view that we had during test and usage of the system.

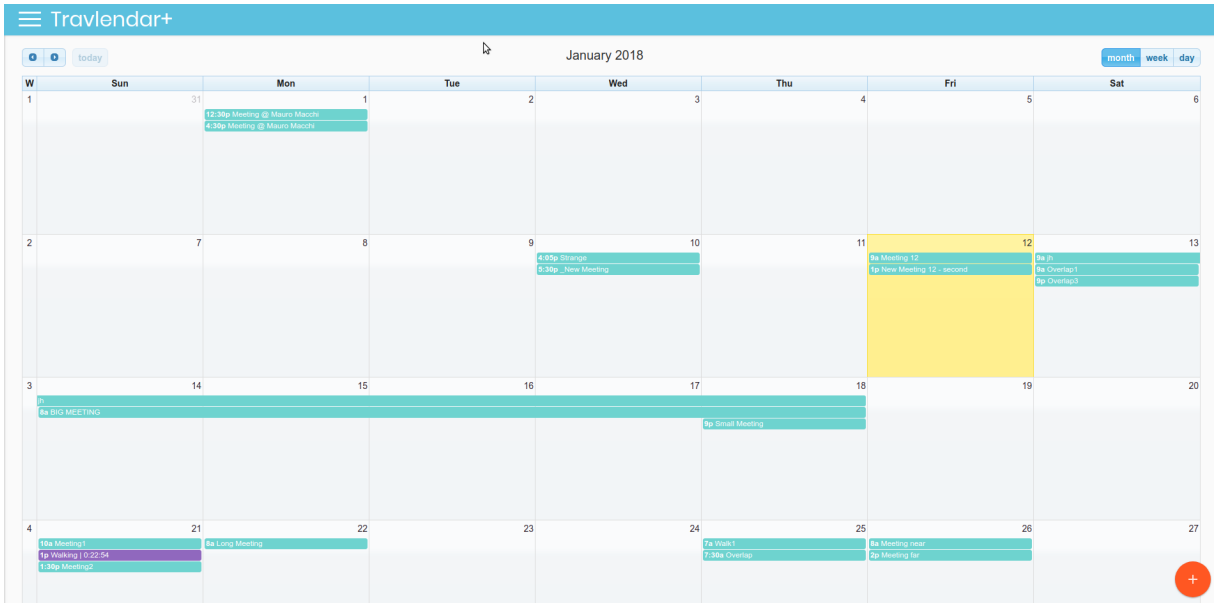
3.1 Login Page

In our opinion the login page, strongly empowered by the large usage of JavaScript, is good looking even if it does not follow the colour scheme and the general look of the calendar page found once logged into the system.
We found some problems in case of invalid signup: if you insert some invalid data, such as a password too short or too similar to the username, the system redirects you to the login form instead of the signup one, which is slightly annoying. On the bright side, if you go back to the registration page you can still find the data you entered as well as some hint on the errors you did.



3.2 Calendar Main Page

The calendar page of the system, that goes through daily, weekly and monthly views, is really immediate and provides everything you need, in spite of being a bit similar to Google Calendar's page.
For what concerns the form that allows a user to create an appointment, we found that the procedure used to insert dates and the location is not very user-friendly: locations are not suggested, for example using the Google APIs, although these APIs are used later to retrieve longitude and latitude from the name of the location inserted; to insert a starting and ending date for an event, you have to provide always a full date format (e.g. '2018-01-10 10:30') and after a few uses it becomes very uncomfortable, especially being used to select dates from datepickers.



3.3 General Considerations

- Breaks are fixed from 12.00 to 14.00 and when one or more appointments overlap with them and make them not doable, it is signalled using a window alert. This procedure is done correctly by the system. However, in spite this is written as a requirement in the RASD document of the project, we believe that not to give the possibility to create customizable breaks is limiting for a complete and satisfying usage of Travlendar+. In addition, the system signals you that a break is not doable only when it actually becomes not doable and you can not check it anymore afterwards. It is also not possible to see the computed starting times for breaks that are actually doable.

- When two appointments overlap, the system signals it with a warning using a window alert and deletes the appointment that has just generated the warning. In our opinion not letting you to create this event is a strong choice, e.g. the second appointment is much more important so you want to sacrifice the former, or you want to take them both. Travlendar should have taken it into account and at least give you the possibility to choose which one to delete or create a separate page where all these warnings, maybe together with not doable breaks, are listed.

- While testing the application we have also found a little bug: if you try to create two different appointments in two locations whose names are not the same, but that get geocoded to the same latitude-longitude pair (for example "Duomo Milano" and "Duomo di Milano"), than the second appointment is not correctly created and the application's execution flow stops. This is probably because of a unique index constraint on the latitude-longitude pair firing at the database level.

4 Testing

Test Name	Valid Signup
Event Flow	1. Go to the login page. 2. Click on the registration button. 3. Insert a non-used nickname, a non-used email and two equal passwords in password and password confirmation fields. 4. Click on the confirmation button.
Expected Output	The system redirects the user to an empty calendar page.
Actual Output	The system redirects the user to an empty calendar page.
Notes	

Table 1: Test - Valid Signup

Test Name	Valid Login
Event Flow	1. Go to the login page. 2. Insert a correct username with its password. 3. Click on the login button.
Expected Output	The system shows you your calendar page.
Actual Output	The system shows you your calendar page.
Notes	

Table 2: Test - Valid Login

Test Name	Invalid Signup
Event Flow	1. Go to the login page. 2. Click on the registration button. 3. Fill the form with invalid data. 3.1. Fill with an already used username, a valid email and a valid password. 3.2. Fill with a new username, an already used email and a valid password. 3.3. Fill with a new username, a valid email and two different passwords in password and password confirmation fields. 4. Click on the confirmation button.
Expected Output	The system does not let you to signup for invalid credentials, staying in the same page and showing errors messages.
Actual Output	1. The system does not let the user to signup but redirects the user to the login page without signalling errors. 2. The system lets the user to signup and shows the calendar page. 3. The system does not let the user to signup but redirects the user to the login page without signalling errors.
Notes	If you click on the registration button after having been redirected to the login page, you can find error messages about your invalid data.

Table 3: Test - Invalid Signup

Test Name	Invalid Login
Event Flow	1. Go to the login page. 2. Insert invalid data, such as non-existent username or incorrect password. 3. Click on the login button.
Expected Output	The system prevents you from logging in.
Actual Output	The system prevents you from logging in.
Notes	

Table 4: Test - Invalid Login

Test Name	Unauthorized access
Event Flow	1. Try to load a page inside the application without being logged in.
Expected Output	The system prevents you from doing that.
Actual Output	The system prevents you from doing that.
Notes	The system redirects you to the login page.

Table 5: Test - Unauthorized access

Test Name	Valid Appointment Creation
Event Flow	1. Login. 2. Click on the button to create a new appointment. 3. Insert a title and two subsequent dates. 4. Click on the save button.
Expected Output	The system redirects you to the calendar page where the new appointment is placed.
Actual Output	The system redirects you to the calendar page where the new appointment is placed.
Notes	A user can create an appointment without inserting a location. However this is in contradiction with requirements number 8 of the RASD document.

Table 6: Test - Valid Appointment Creation

Test Name	Invalid Appointment Creation
Event Flow	1. Login. 2. Click on the button to create a new appointment. 3. Insert a title, a start date, an end date that is before it and a location. 4. Click on the save button.
Expected Output	The system does not allow you to create an appointment because of invalid dates.
Actual Output	The system redirects you to the calendar page where the new appointment is placed without an ending date and with a default duration of 2 hours.
Notes	This is in contradiction with requirements number 10 of the RASD document that prevents an appointment to have such invalid dates.

Table 7: Test - Invalid Appointment Creation

Test Name	Appointment Update
Event Flow	1. Login. 2. Open the page of an appointment already in your calendar. 3. Click on the update button. 4. Change the start time, end time and location. 5. Click on the save button.
Expected Output	The system shows the calendar with the updated version of the appointment.
Actual Output	The system shows the calendar with the updated version of the appointment.
Notes	

Table 8: Test - Appointment Update

Test Name	Appointment Delete
Event Flow	1. Login. 2. Click on an appointment. 3. Click on the delete button and confirm.
Expected Output	The system redirects you to the calendar page where the deleted appointment does not exist anymore.
Actual Output	The system redirects you to the calendar page where the deleted appointment does not exist anymore.
Notes	

Table 9: Test - Appointment Delete

Test Name	Two appointments in the same day generate a travel
Event Flow	1. Login. 2. Create an appointment in the morning of some day. 3. Create an appointment in the afternoon of the same day.
Expected Output	The system shows the calendar with the two new appointments and the travel between them.
Actual Output	The system shows the calendar with the two new appointments and the travel between them.
Notes	Make sure the two appointments have reasonably near locations, otherwise no travel can be computed.

Table 10: Test - Two appointments in the same day generate a travel

Test Name	Create an appointment that overlaps with another already present and generate a warning
Event Flow	1. Login. 2. Click on the button to create a new appointment. 3. Insert a title, a start date that is in the middle of an other appointment, a later end date and a location. 4. Click on the save button.
Expected Output	The system prevents the user creating a new appointment because it overlaps with an existing one and signals it to the user.
Actual Output	The system prevents the user creating a new appointment because it overlaps with an existing one and signals it to the user.
Notes	If you do not provide a location, the system actually lets you save the meeting and places it on the calendar, overlapping also graphically with the conflicting one. In this case no warning is generated.

Table 11: Test - Create an appointment that overlaps with another already present and generate a warning

Test Name	Two appointments in the same day generate a warning if the travel is too long
Event Flow	1. Login. 2. Create an appointment in the morning of some day. 3. Create an appointment in the afternoon of the same day, with a location so distant from the one of the other appointment that a travel would take more than the available time.
Expected Output	The system shows the user a warning.
Actual Output	The system shows the user a warning.
Notes	Appointments are kept in the calendar, but no travel is created.

Table 12: Test - Two appointments in the same day generate a warning if the travel is too long

Test Name	Create an appointment that covers the entire break time slot and signal it to the user
Event Flow	1. Login. 2. Click on the button to create a new appointment. 3. Insert a title, a start date before 12.00, an end date after 14.00 and a location. 4. Click on the save button.
Expected Output	The system creates the appointment and signals to the user that the break of that day is not doable anymore.
Actual Output	The system creates the appointment.
Notes	In case the break is inhibited due to an appointment that starts before 12.00 and ends after 14.00, the warning is not signalled. When the break is inhibited by a travel that the system should plan between two subsequent appointments, a warning is generated but it is not shown to the user because it is covered by the warning that signals that the appointment has been created.

Table 13: Test - Create an appointment that covers the entire break time slot and signal it to the user

Test Name	Two appointments in the same day generate a travel with a specific travel mean if that's the only available one
Event Flow	1. Login. 2. Open the preferences and disable all travel means but one. 3. Create an appointment in the morning of some day. 4. Create an appointment in the afternoon of the same day.
Expected Output	The system shows the calendar with the two new appointments and the travel between them, computed with the only available travel mean.
Actual Output	The system shows the calendar with the two new appointments and the travel between them, computed with the only available travel mean.
Notes	

Table 14: Test - Two appointments in the same day generate a travel with a specific travel mean if that's the only available one

5 Conclusions

We think that the project we tested had a pretty limited scope, still providing all the core functionalities of a Travlendar+ application. Most of the planned goals of the RASD have been accomplished and almost all of the requirements have been satisfied, so we evaluate it positively.
The python code is properly structured and well commented, and this makes it fairly readable even without any prior knowledge on the system nor on the framework. The couple of bugs found were not really relevant issues, and we think they can be easily fixed once the development team has been notified of their existence.
Overall, the application is easy to use and immediate to understand for any user. Nevertheless, we think that the project needs major improvements and extended functionalities to be properly released on the market.

