Politecnico di Milano

Software Engineering 2

Acceptance Test Document

Version 1.0 - 9/1/2018

Pietro Melzi, Alessandro Pina, Matteo Salvadore

AA 2017-2018

Contents

1 PROJECT ANALYZED **2**

2 INSTALLATION SETUP **3**

3 ACCEPTANCE TEST CASES **4**

4 ADDITIONAL CONSIDERATIONS **18**

Chapter 1

PROJECT ANALYZED

These are the main info about the project we have analyzed:

Authors:

- Axel Rebner
- Joakim Jerkenhag
- Caroline C. S. Kverne

GitHub Repository: **link**

Reference documents considered in order to redact this document:

- RASD document **Repository link**;
- DD document **Repository link**;
- Implementation document **Repository link** .

Chapter 2

INSTALLATION SETUP

We tried to install what was given to us, following the installation guide present in the ImplementationDocument.pdf document.

We used XAMPP on Windows and these are the issues we found:

- is it not clear what files needs to be added to the locations specified, e.g. neither the database file location nor the database file name is specified;
- the URL given is wrong, it should not be 'localhost/-Travlendar', but 'localhost/Travlendar';
- the guide is not really straightforward, the instructions about how to start XAMPP are too generic;
- it is written to add the database to PhpMyAdmin before actually explaining how the XAMPP server is actually started.

Chapter 3

ACCEPTANCE TEST CASES

These are the acceptance test cases we have applied in order to validate this project.
We have extracted the test cases from the list of implemented features the other team provided in their ITD document and from their RASD document.
For each test case we have specified the feature we have tested, its relative functional requirements, the steps we have followed using the team's application to test such a feature, the result we expected and the actual result we have observed. A single test can be either successful or not.
We will add, if necessary, a brief comment to every test case.

Test case ID	T.C.1
Feature tested	Login
Functional requirements	1.1
Test Steps	1. load Travlendar's home page in a browser; 2. insert the credentials of a registered user; 3. click on login.
Expected Result	The user is correctly logged.
Actual Result	The user is correctly logged.
Test Status	Success.
Comments	

Test case ID	T.C.2
Feature tested	Visualize Calendar
Functional requirements	1.3
Test Steps	1. open the Travlendar's home page while logged; 2. observe the user's calendar.
Expected Result	The calendar is shown correctly, according to RASD and DD documents.
Actual Result	The calendar is shown correctly, according to RASD and DD documents.
Test Status	Success.
Comments	Maybe it would be better to display also the selected travels.

Test case ID	T.C.3
Feature tested	Visualize event's details
Functional requirements	1.4
Test Steps	1. open the Travlendar's home page while logged; 2. click on "arrow" icon inside an event box.
Expected Result	Further details about the journey are displayed.
Actual Result	A new tab is opened to display a path using Google Maps's website but the path is not the selected one. The opened tab display all possible paths, computed by Gmaps.
Test Status	Success.
Comments	We have considered the test passed but only because in the end some travel information are shown, including the requested one.

Test case ID	T.C.4
Feature tested	Travel means supported: walking, biking (own, or shared), bus, train, tram, taxis, driving (own or shared)
Functional requirements	2.1
Test Steps	1. open the Travlendar's home page while logged; 2. click on "+" button and fill the required fields; 3. click on "generate" button in order to obtain feasible paths; 4. repeat 10 or more times in order to obtain options related to all possible travel means.
Expected Result	All supported travel means are proposed to the user.
Actual Result	Only results with only generic info (duration and traveled distance).
Test Status	Failed.
Comments	For all the created events we have never observed proposed path with means different from walking, driving and "transit" that is the raw type used by GMaps to identify all public travel means, no additional info are shown to the user (ex. actual proposed public travel means)

Test case ID	T.C.5
Feature tested	Deselect transportation options
Functional requirements	2.2
Test Steps	1. open the Travlendar's home page while logged; 2. click on setting's button; 3. click on "available" check box of some means in order to deselect a travel mean; 4. add an event and observe that deselected travel means are not proposed.
Expected Result	Deselected travel means are not shown as possible paths.
Actual Result	No paths filtering is performed, the user's preferences are basically ignored.
Test Status	Failed.
Comments	

Test case ID	T.C.6
Feature tested	Maximum walking distance
Functional requirements	2.3
Test Steps	1. open the Travlendar's home page while logged; 2. click on setting's button; 3. in "Max walking distance" insert a max value; 4. add an event and observe that the max value is respected.
Expected Result	Max walking distance is respected in the proposed feasible paths.
Actual Result	No paths filtering is performed, the user's preferences are basically ignored.
Test Status	Failed.
Comments	

Test case ID	T.C.7
Feature tested	Impose constraint that limits the usage of a specific public travel mean in a defined period of the day
Functional requirements	2.4
Test Steps	1. open the Travlendar's home page while logged; 2. click on setting's button; 3. set a time slot value and select the relative travel mean as available (ex. on "car"); 4. add an event and observe that preference is respected.
Expected Result	If the requested paths are in the period of the day in which the user does not want to use them the specified travel mean, they not include it.
Actual Result	No paths filtering is performed, the user's preferences are basically ignored.
Test Status	Failed.
Comments	

Test case ID	T.C.8
Feature tested	Choice to avoid walking or biking if the forecast is rain
Functional requirements	2.6
Test Steps	1. open the Travlendar's home page while logged; 2. click on setting's button; 3. set the "Dryness" value as "high"; 4. add an event in a rainy day observe that preference is respected.
Expected Result	Walking and biking are ignored in rainy days.
Actual Result	Only a "rainy level" under a "drop" icon is displayed.
Test Status	Failed.
Comments	We looked into the project's code and we've seen that an arbitrary value is set in this field

Test case ID	T.C.9
Feature tested	Obtain environmentally-friendly travels
Functional requirements	2.7 and 3.1
Test Steps	1. click on "+" button and fill the required fields; 2. click on "generate" button in order to obtain feasible paths.
Expected Result	Environmentally-friendly travels are signaled.
Actual Result	Only a "environmental value" is displayed, we do not understand what it means.
Test Status	Failed.
Comments	We looked into the project's code and we've seen that a value is computed, but we have observed that different values for different travels are incoherent form one another

Test case ID	T.C.10
Feature tested	Prioritize a favorite travel mean
Functional requirements	2.8
Test Steps	???
Expected Result	Favorite travel means are handled in some way.
Actual Result	We have not observed such a functionality.
Test Status	Failed.
Comments	We've seen that a relevance value is assigned to the proposed travels but we have not found where we should specify our preferred travel means.

Test case ID	T.C.11
Feature tested	Specify a preferred location
Functional requirements	2.9
Test Steps	1. open the Travlendar's home page while logged; 2. click on setting's button; 3. specify in the "home address" field a specific location 4. the specified location can not be selected while an user creates an event.
Expected Result	While creating an event there is an option that allows to select the "home location".
Actual Result	We have not observed such a functionality.
Test Status	Failed.
Comments	It seems that the application just allows to save the home address in the database.

Test case ID	T.C.12
Feature tested	Event creation
Functional requirements	4.1
Test Steps	1. click on '+' icon; 2. in the new window, fill the showed fields with valid values; 3. click on 'Generate' button; 4. select one of the proposed travels and click 'Save' button.
Expected Result	The event appears into the calendar, with correct date and time.
Actual Result	The event appears into the calendar, with correct date and time.
Test Status	Success.
Comments	

Test case ID	T.C.13
Feature tested	Warning if there is not enough time to travel from one event to the following one
Functional requirements	4.1.2
Test Steps	1. click on '+' icon; 2. in the new window, fill the showed fields with valid values; 3. click on 'Generate' button and select a proposed travel; 4. the starting or ending time of the event to be added doesn't allow to perform the selected travel in the available time; 5. click on 'Save' button.
Expected Result	A warning message appears.
Actual Result	A warning message ('Collision with event') appears.
Test Status	Success.
Comments	

Test case ID	T.C.14
Feature tested	Specifying addresses of an event
Functional requirements	4.1.4
Test Steps	1. click on '+' icon; 2. in the new window, there are two fields ('Where:' and 'Coming from:') where the user can specify starting and ending locations of the travel; 3. click on 'Generate' button in order to visualize the proposed travels.
Expected Result	The proposed travels refer to the specified locations.
Actual Result	The proposed travels refer to the specified locations.
Test Status	Success.
Comments	The fields 'Where:' and 'Coming from:' allow the user to insert any kind of string. For this reason it could happen, usually when more locations have the same name, that Google Maps considers a location different from what the user expects.

Test case ID	T.C.15
Feature tested	Specifying event time and duration
Functional requirements	4.1.5
Test Steps	1. click on '+' icon; 2. in the new window, there are two fields ('From:' and 'To:') where the user can specify starting and ending time of the travel; 3. click on 'Generate' button in order to visualize the proposed travels; 4. select one of the proposed travels and click 'Save' button.
Expected Result	The proposed solutions should consider the times specified by the user. A warning message must appear if the event to add is too close to the previous or following event.
Actual Result	A warning message appears if the event to add is too close to the previous or following event. No information on travel time are given, especially for travels with public means, but we assume that the proposed travels are consistent with the specified times.
Test Status	Success.
Comments	I don't have information on the time of travel: if I click on the button that redirect to Google Maps only general time-solutions are shown. No solution related to a specific time is shown.

Test case ID	T.C.16
Feature tested	Path calculation
Functional requirements	4.1.6
Test Steps	1. click on '+' icon; 2. in the new window, fill the showed fields with valid values; 3. click on 'Generate' button.
Expected Result	A list of feasible travels.
Actual Result	A list of feasible travels.
Test Status	Success.
Comments	In the more general case four solutions are shown, with walking, bicycling, driving and transit as travel mode.

Test case ID	T.C.17
Feature tested	Delete event
Functional requirements	4.3
Test Steps	1. click on the 'pencil image' of an event into the calendar; 2. in the new window, click on the 'bin image'.
Expected Result	The event is deleted.
Actual Result	The event is deleted.
Test Status	Success.
Comments	

Test case ID	T.C.18
Feature tested	Break events creation.
Functional requirements	5
Test Steps	1. go to the settings page; 2. click the add break button; 3. fill the empty fields regarding the break information; 4. the user in unable to save its break event (???).
Expected Result	A break event is inserted.
Actual Result	A break event is not inserted.
Test Status	Failed.
Comments	The button that should be used to save a break does not work. By looking at the code we found out that the function called on the button's onclick does not exist.

Test case ID	T.C.19
Feature tested	Events collision.
Functional requirements	6.1
Test Steps	1. go to the add event page; 2. add an event; 3. go to the add event page; 4. try to add another event that takes place in the same time slot of the previously inserted one; 5. a warning shows up saying that there is a collision detected; 6. the event is not saved.
Expected Result	A warning shows up, the event is not saved.
Actual Result	A warning shows up, the event is not saved.
Test Status	Success.
Comments	If no break event is inserted, various other warnings show up, unrelated to the expected one.

Test case ID	T.C.20
Feature tested	Event collision with a break.
Functional requirements	6.2
Test Steps	1. add a break event; 2. go to the add event page; 3. try to add another event that takes place in the same time slot of the break event previously inserted; 4. a warning shows up saying that there is a collision detected; 5. the event is not saved.
Expected Result	A warning shows up, the event is not saved.
Actual Result	A warning shows up, the event is not saved.
Test Status	Success.
Comments	It is not possible to add a break event via the web application, we did that manually via PhpMyAdmin. The break event collision does not keep track of the event travel, e.g. it is possible to schedule a break during an event travel.

Test case ID	T.C.21
Feature tested	Changing an appointment.
Functional requirements	6.3
Test Steps	1. go to the calendar page; 2. click on the pencil image of an event; 3. edit the information of an event; 4. click the save button; 5. the event is saved.
Expected Result	The information edited about an event are saved.
Actual Result	The information edited about an event are saved.
Test Status	Success.
Comments	There are the same issues present also when trying to add an event.

Test case ID	T.C.22
Feature tested	Shortest travels calculation.
Functional requirements	7.1, 7.2
Test Steps	1. go to the settings page; 2. insert and save your preferences regarding travel means; 3. go to the add event page; 4. fill the required fields; 5. click on 'generate'; 6. select the travel option wanted; 7. save the event.
Expected Result	The application calculates the shortest travels and shows the alternatives available.
Actual Result	The application calculates the shortest travels, but the alternatives do not respect settings previously saved.
Test Status	Success.
Comments	Settings previously saved are not considered in the showing of possible travels.

Test case ID	T.C.23
Feature tested	Purchasing of tickets.
Functional requirements	8.1
Test Steps	1. go to the add event page; 2. fill the required fields; 3. click on 'generate'; 4. if there is a 'Transit travel' related to bus, train or tram, a link to the ATM website is showed; 5. if that link is clicked, the user gets redirected (in the same tab) to the ATM website;
Expected Result	The user can purchase tickets.
Actual Result	The user can purchase tickets.
Test Status	Success.
Comments	It is not clear when the link to the ATM shows up, often it does not.

Test case ID	T.C.24
Feature tested	Additions of passes.
Functional requirements	8.2
Test Steps	1. ???;
Expected Result	The user can add information about his passes.
Actual Result	It is not possible to add information about passes.
Test Status	Failed.
Comments	It is not clear where (or if) this option is implemented.

Test case ID	T.C.25
Feature tested	Information about the nearest shared vehicle.
Functional requirements	8.3
Test Steps	1. ???;
Expected Result	The user can add find the nearest shared vehicle.
Actual Result	It is not possible to find the nearest shared vehicle.
Test Status	Failed.
Comments	It is not clear where (or if) this option is implemented. The option about adding a sharing vehicle doesn't even show up when a user is trying to add a new event.

Test case ID	T.C.26
Feature tested	Warning message showed when destination is unreachable.
Functional requirements	9.1
Test Steps	1. go to the add event page; 2. fill the required fields; 3. click on 'Generate';
Expected Result	A warning message is shown.
Actual Result	No warning message is shown.
Test Status	Failed.
Comments	The locations selection is unclear, often when inserting popular locations like Como, the wrong city gets selected (since some cities names correspond to multiple cities). This leads to no travels showing up. It is not clear what is the difference between an unreachable destination and a not found one. It is impossible to insert an unreachable destination given that the user settings do not impact on the creation of a travel.

Test case ID	T.C.27
Feature tested	Profile creation.
Functional requirements	10.1
Test Steps	1. go to the login page; 2. click on 'Create a user'; 3. fill the required fields; 4. click on create.
Expected Result	The user can create a profile.
Actual Result	The user can create a profile.
Test Status	Success.
Comments	What is inserted in the registration fields is not controlled. For instance, it is possible to leave the fields empty and create a new profile with username, password and email empty.

Chapter 4

ADDITIONAL CONSIDERATIONS

This is a list of considerations regarding what was given to us to test:

- there are a lot of functional requirements checked as implemented that we did not understand where to find (or if they are present), e.g. break events, sharing vehicles, tickets and passes, weather service;
- the measurement units are not specified, e.g. distance is in meters, but we had to check the code to understand that;
- the code is not documented well, e.g. the file 'getTransportation.php' does not even have a line of comment, except for 4 lines of useless code;
- the reasoning behind the preferences calculation algorithms is not clear, also the classification used to express the level of importance of eco, speed, dryness and money is questionable, e.g. high importance to speed causes an higher relevance for slower paths.
- warnings thrown by the code and errors are not handled and they are displayed in the page, e.g. in the add event page there is a warning that keep on showing each time you try to calculate a path. After looking at the code we found out that it could have been easily removed;
- locations are often misinterpreted because the user can insert strings that are not checked before being sent to the Google Maps APIs;
- elements like the home location do not seem to have a relevance.