



Plunner
Acceptance test document
Version 1.1

Planner	Version: 1.1
Acceptance test document	Date: 20/01/2016

Revision History

Date	Version	Description	Author
30/12/2015	0.1	Initial Draft	Denis Kuryshov
31/12/2015	1.0	Some corrections and reorganization made	Giorgio Pea
20/01/2016	1.1	Final revision and checking	Giorgio Pea

Planner	Version: 1.1
Acceptance test document	Date: 20/01/2016

Table of Contents

[Revision History](#)

[Table of Contents](#)

[1. Introduction](#)

[1.1 Purpose of the document](#)

[1.2 Intended audience](#)

[1.3 Scope](#)

[1.4 Definitions and acronyms](#)

[1.5 Support documentation](#)

[2. Type of Tests](#)

[2.1 Backend Tests](#)

[2.2 Frontend Tests](#)

[2.3 End to end tests](#)

[3. Testing tools](#)

[3.1 Backend testing tools](#)

[3.2 Front end testing tools](#)

[2. Functionalities to be tested](#)

Plunner	Version: 1.1
Acceptance test document	Date: 20/01/2016

1. Introduction

1.1 Purpose of the document

The purpose of this document is to present and formalize a plan of tests that Plunner has to pass in order to be accepted as a final application.

1.2 Intended audience

The intended audience for this document is:

- **The team behind the project:** since this document describes a plan of tests for Plunner, the team will follow it during the testing process in order to have an acceptable final product or to make appropriate fixings to get an acceptable final product.
- **The supervisors of the project:** the information contained in this document can be used by the supervisors to understand the testing process of Plunner, and so to make suggestions to the team and have an idea of the final quality of the project.
- **The customer of the project:** the information contained in this document can be used by the customer of the project to understand the process the team is taking to guarantee the correctness and quality of the application and may be suggest the necessity of a focus on certain product's quality aspects rather than others.

1.3 Scope

This document aims to give structure and consistency to the tests that the developers of the project will be writing, in order to guarantee the quality of the final result and the achievement of the established requirements (see RDD). This document gives consistency and structure to the testing process of Plunner by listing the type of tests to be used for the application and the features of the applications these tests will cover.

1.4 Definitions and acronyms

Abbreviation, acronym, term	Definition
RDD	Requirements Definition Document for the Project

In addition to the indicated abbreviations and definition see the RDD's definitions and acronyms

1.5 Support documentation

- Requirements Definition Document

Plunner	Version: 1.1
Acceptance test document	Date: 20/01/2016

2. Type of Tests

2.1 Backend Tests

- Unit tests for the different modules of the backend of the application. These tests should cover at least 80% of the codebase.
- Manual tests for the the communication and interaction of the different modules of the backend of the application. These tests have been preferred to automated and more systematic integration tests for reasons of time and complexity.
- Manual tests for the optimization module of the backend of the application.

2.2 Frontend Tests

- Unit tests for the different modules of the frontend of the application. These tests should not be as extensive as the ones for the backend of Plunner, this because unit testing of the frontend is very complex and time expensive.
- Manual tests for the the communication and interaction of the different modules of the frontend of the application. These tests have been preferred to automated and more systematic integration tests for reasons of time and complexity.
- Manual tests to check the usability of the web interface.
- Manual tests to check the responsiveness of the web interface: the web interface should adapt itself to different screen resolution and guarantee by doing so an acceptable usability.
- Manual tests to check the compatibility with different browsers: the web interface should be as much consistent as possible with different browsers.

2.3 End to end tests

For reasons of time and complexity, end-to-end tests should be done manually by one guy from the frontend and another one from the backend to check the correct behavior of the global system

Plunner	Version: 1.1
Acceptance test document	Date: 20/01/2016

3. Testing tools

3.1 Backend testing tools

- PHP Unit (<https://phpunit.de/>)
- Scrutinizer (<https://scrutinizer-ci.com/>)
- Travis (<https://travis-ci.org/>)

3.2 Front end testing tools

- Karma (<https://karma-runner.github.io/0.13/index.html>)
- Jasmine (<http://jasmine.github.io/>)
- Travis (<https://travis-ci.org/>)

Plunner	Version: 1.1
Acceptance test document	Date: 20/01/2016

2. Functionalities to be tested

Here follows a resume of the functionalities of the application that must be tested using the type of tests described in the previous sections. The whole set of functionalities to be tested matches exactly the requirements of the application (see RDD).

2.1 Organization and registered organization

	#	Specific functionality
Registration and login into Plunner	1	An organization can register via email and password
	2	A registered organization can login via email and password
	3	A registered organization can recover its login credentials
	4	Plunner must allow a remember me functionality
	5	A registered organization can see and edit its personal information
	6	A registered organization can change its password

	#	Specific functionality
Registration and managing of Members	1	A registered organization can register to the service, within its context, its members using an email address and a password
	2	A registered organization can edit the data associated with one of their members(name, email, password)
	3	A registered organization can remove one of its members from Plunner

Planner	Version: 1.1
Acceptance test document	Date: 20/01/2016

	#	Specific functionality
Creating and managing teams	1	A registered organization can organize its members in teams
	2	The Service must allow a registered organization to choose for each team a Member to become a planner of the team
	3	A registered organization can change the planner of a team
	4	A registered organization can change the name, the description and the composition of each team
	5	A registered organization can delete a team

2.2 Member

	#	Specific functionality
Login	1	A Member can login within the context of its organization using email and the password
	2	A Member can recover their login credentials
	3	The Service must allow the remember me functionality for Members
	4	A Member can see and edit their personal information
	5	A Member can change its login password

Planner	Version: 1.1
Acceptance test document	Date: 20/01/2016

	#	Specific functionality
Import a schedule from an external service via CalDAV	1	A Member can import a schedule from an external service via the CalDAV protocol
	2	The Service must keep in sync a Member's imported schedules
	3	The Service must keep track of the a Member's imported schedules
	4	A Member can edit and remove its imported schedules
	5	A Member can enable or disable its imported schedules

	#	Specific functionality
Compose a schedule	1	A Member can compose a schedule by specifying a name and a series of time slots of a given period in which they are busy
	2	A Member can edit or delete its composed schedules
	3	The Service must keep track of a Member's composed schedules

	Section	Specific functionality
Receiving the information	1	The Service must be able to send to a Member notifications about the status of a meeting to be planned in which is involved
	2	A Member can see meetings to be planned or planned meetings in which is involved

Planner	Version: 1.1
Acceptance test document	Date: 20/01/2016

2.3 Planner

	#	Specific functionality
Create a meeting	1	A planner of a team g can create a meeting for that team by specifying a name, a description, a duration and possible meeting times
	2	The Service must support meetings to be planned trying to maximize the number of participants

	#	Specific functionality
Managing meetings	1	A planner can cancel or edit a created meeting