D3.3 ESTABLISH Test plan

Deliverable ID:	D 3.3
Deliverable Title:	Test plan
Revision #:	2.0
Dissemination Level:	Public
Responsible beneficiary:	VTT
Contributing beneficiaries:	All
Contractual date of delivery:	30.11.2017
Actual submission date	27.11.2017

Version history

Date	Contributor	Version
05.09.2017	**SIVECO**	**1.0**
16.10.2017	**SIVECO**	**1.1**
27.11.2017	**SIVECO**	**2.0**
01.12.2017	**VTT – QA review**	**2.0**
07.12.2017	**SIVECO**	**3.0**

Table of Content

1. Introduction

1.1 Scope

This document outlines the approach of system and acceptance testing on ESTABLISH platform and presents the methodology of running the tests.

This Test Plan details the approach of the System and the Acceptance tests. Before running the tests the following should be considered:

- The User Requirements described in the homonymous document are met and delivered;
- The suitability and usability of the system;
- The system is ready to go in production.

1.2 References

The following documents will be referred in this document:

Document	Title	Version
D2.1	State-of-the-art, detailed use case definitions and user requirements	
D2.2	Technical requirements and high level specifications	

1.3 Document overview

In this document we will present the test approach, showing the test phases involved in the system and acceptance phase of the project, the test levels that will be performed along with the associated deliverables.

Furthermore we will present a test execution strategy in terms of defect tracking workflow and defect priority classification. We will present the testing criteria showing the entry and exit criteria, test pass/fail criteria and acceptance criteria.

In the final chapter of the document we will present information on the reports that will follow system and acceptance testing.

2. Test approach

2.1 Approach overview

Before handover to Acceptance Test the development team will have performed unit testing and System testing.

For each development's iteration, system and acceptance testing must be performed:

- Regression tests (for what was developed for previous iterations)
- Non-regression test (for what was developed for the current iteration).

Each test described in "Acceptance Test Specification Document" has a dual risk evaluation ranking. Test cases can be critical (due to high frequency of the operation described and/or its impact on achieving system major objectives) or non-critical (test cases depicting low frequency operations and/or with minor impact on ESTABLISH functionality).

System testing runs before acceptance testing.

2.2 Test phases

According to the project planning, ESTABLISH will be developed in phases. Unit, System and Acceptance Testing must follow the foreseen scheduling.

The foreseen development and testing phases are going to be established in the final version of the test plan (this document).

2.3 Test levels

ESTABLISH will be subject to Unit Testing, System and Acceptance Testing.

Unit testing is the method by which individual units of source code of ESTABLISH sets of one or more application modules, usage procedures, and operating procedures are tested to determine if they are fit for use. Unit testing performs every time when a new function is set in place, allowing issues to be identified and solved in the early phase of the development cycle.

System testing is performed by the testing team before the acceptance testing. Both testing consider project plan development phases.

Both System and Acceptance Testing rely on the requirements defined in the **D2.1 State-of-the-art, detailed use case definitions and user requirements** Document.

Both System and Acceptance Testing aim to check that the created system is correct, completely stable and works as a whole, according to its requirements. System and Acceptance Testing fall within the scope of black box testing; therefore they require no knowledge of the inner design of the code or logic.

When performing System and Acceptance testing solely the test cases presented in Acceptance Test Specification Document will be used.

2.4 Test deliverables

The following deliverables will be presented in relation with ESTABLISH testing tasks:

Deliverable	Description
Test Plan (this document) – preliminary version	Presents test approach, test environments, test execution strategy, testing criteria and test reporting and test cases for System and Acceptance testing.
Test Plan (this document) – final version	Presents the updated version of the preliminary one.

3. Test environments

During the development and testing processes, the Test and Pre-Production environments will be used. Unit testing, System Testing and Acceptance testing will be performed in the Test environment by the testing team.

4. Test execution strategy

The use cases presented in “D2.1 State-of-the-art, detailed use case definitions and user requirements” and ”D2.2 Technical requirements and high level specifications” will be the primary source for the test cases that will cover all areas of in-scope functionality to be tested. All the test cases associated with ESTABLISH will be included in the “Test Plan Document”.
A test case may sub-divide into a more granular series of test conditions where appropriate. Each test condition is a binary statement that will have one possible expected outcome.
The test condition should correspond to a simple statement within which the expected outcome is obvious e.g. “upon pressing login on the home page the login window is displayed”.
The test condition describes ‘what to test’.
There may be multiple test conditions for one business rule or functional requirement, as the conditions must cover all scenarios and possible combinations.

Test Cases
The business use cases will be the primary source of the test cases. When needed, test conditions incorporate into test cases.
A test case is ‘how to test’ a group of at least one test condition.
The test case will contain steps to execute and expected result for each step.
The purpose of the test case is to enable verification of multiple test conditions.
Each test condition may cover more than one test case.
In order to perform more complex testing activities, testers could perform sequences of test cases simulating non-fragmented user behavior, simulating real-life natural usage of the system.

Test Cycles
There will be at least one cycle of all test conditions / test cases executed. As described in chapter “2.1-Approach overview”, both regression tests (for what was developed for previous iterations) and non-regressive tests (for what was developed for current iteration) will be performed for each iteration.

Regardless of the number of cycles, all test conditions/cases that fail will be retested before delivering a fix version.

4.1 Defect tracking workflow

A defect or an incident is defined as any unplanned event that has the potential to significantly affect the operation on ESTABLISH.

4.1.1 Defect reporting

During Acceptance testing all the defect and problems will be uploaded as incidents into the JIRA incident management system, under a separate project.

From JIRA, defects/issues report will be generated and managed so that all the high issues are solved as soon as possible, while the medium and low incidents will be scheduled for a resolution.

During the implementation of ESTABLISH, the teams will follow the standard operating procedure. Incidents can be logged by designated contractor staff members, or members of the SIVECO team (contractor team) using the JIRA system established specifically for this purpose.

The members of the contractor team will identify the causes of incidents, will investigate, manage and resolve them. Where possible lessons learned from the incident will apply to minimize the likelihood of the incidents occurring again.

When adding a new incident into the system or when changing the status of an incident, JIRA users of the ESTABLISH project may be able to receive an email notification from JIRA. This notification will present the actual events that happened on the respective incident.

JIRA administrator may set receiving or not an email notification whenever a ticket is updated.

Within the ESTABLISH project on JIRA an incident may be classified as new feature, bug, improvement, usability problem, technical assistance or task related incident.

Complete details of the incident including a relative priority can be added into the incident management system.

Types of incidents and resolutions that will be used in JIRA

No	Type of incident	Description of type
1	Bug	This type of incident depicts a software code problem; this is usually also named “code error”.
2	Improvement	This type of incident should be used whenever a change is necessary in an application, no matter if it is an aspect of Graphical-User-Interface or of a business or functionality component, or whether it’s a new feature or changing an existing feature.
3	Assistance	This type of incident depicts a need for technical assistance in using the applications.
4	Task	This type of incident depicts a need for a very specific technical task.

No	Type of incident	Description of type
5	New Feature	This type of incident depicts a need for a new development for the existing applications, no matter if there are small, minor changes or new modules.

Roles and responsibilities

The roles involved on a JIRA incident, and their responsibilities, are:

No	Role	Responsibility
1	Reporting user (or tester)	This person may be a member of the testing team. The reporting user is doing: ▪ The initial adding of a new incident into the system ▪ The final validation that the incident has been resolved ▪ If the validation is ok, closing of the incident (the change of status from Resolved to Closed)
2	Software developer or consultant	This person is performing the actual work to solve an incident. The software developer or consultant is doing: ▪ The work necessary to solve an incident ▪ The change of status from Open/Working to Fixed/Resolved.
3	IT/QA manager	The responsible person for this role is doing: ▪ The initial distribution of the JIRA incident to a software developer/consultant ▪ The final validation, before sending the incident back to the reporting user ▪ If the incident is solved, this person will change the status to “Resolved” and assign the incident to the reporting user ▪ If the incident is not solved, this person will change the status to “Reopened” and assign the incident back to the software developer/consultant
4	JIRA administrator	The responsible person for this role is a person from the IT department. The Project Manager is in charge of it.

Mandatory information necessary when adding an incident

When adding a new incident into the JIRA Incident Management System, the user should be as descriptive as possible about the respective issue.

The following details could be relevant, when talking about an error/usability issue:

- The operating system name and version (e.g. Windows XP SP2)
- The browser name and version (e.g. Microsoft Windows Internet Explorer v8.0)
- The resolution of the screen (e.g. 1024x768 pixels, 32-bit color depth)
- The zone of ESTABLISH where the issue has occurred (also providing the link to that particular page)
- The exact steps necessary to reproduce the problem (e.g. access “X” functionality, perform ”Y” action, view the result, click on the result etc,)
- Description of the exact problem (e.g. a pop-up will appear for a very short period of time and the user is not able to actually see the message inside it). Screenshot attachment

4.1.2 Defect solving

The following is an overview of the proposed defect tracking and resolution process: all incidents created are assigned to a member of the team. The incident will then be assigned for investigation and diagnosis.

After the correction has been applied and initial testing of the fix has taken place, the status of the incident will be changed to “Fixed”. In this instance, ”Resolved” means that the developer is satisfied with the fix that has been applied to the incident and is awaiting final approval of these changes.

When working on an issue, the types of Resolutions which will be used by the user who will resolve the incident are:

Type of resolution	Description of the resolution
Rejected	Request is refused
Fixed	A fix for this issue has been done and tested
Won’t fix	The problem described is an issue which will never be fixed
Duplicate	The problem is a duplicate of an existing issue
Incomplete	The problem is not completely described
Cannot reproduce	All attempts to reproduce this issue failed or not enough information was available to reproduce this issue. If more information appears later, then the issue should be reopened
Suspended	The issue is being suspended, until more details will be available or until a decision will be made regarding some aspects
Function as designed	Feature is working as designed, it is not a bug
Implemented	Request was implemented
Unresolved/ Reopened	Even if the software developer or IT responsible considered the incident as solved, the reporting user or an other user involved in ESTABLISH project consider that the incident is not solved and they reopen it

Type of resolution	Description of the resolution
Administration	The issue is in administration status, which means the IT management is investigating the status of this issue in order to decide the future steps which may be necessary in order to close the current issue.
Cancelled	The incident is neither solved, nor closed, but the IT management/reporting user consider that this incident is no longer up-to-date because some other changes occurred in the meantime (e.g. a change in the interface GUI) and the incident may no longer appear on the system because the feature has changed.

4.1.3 Defect closure

All resolved incidents will be subject to a final quality assurance before being closed off and the status of the incident changed to “Closed”.

According to the incident lifecycle, a JIRA item can have one of the following statuses:

- Open issue/Creation
  - The issue created by a member of the ESTABLISH team.
  - The person who created the incident is named “Reporting user” (or tester)
- In progress/Working
  - The issue has been assigned to a software developer or consultant and the activities performed are progressing
- Reopened
  - The issue considered as Fixed by a software developer or consultant, but the reporter or another person involved in ESTABLISH project checked the solution and this considered it not fixed. Consequently, the issue is reopened. If this is the case, the person that performed the check will assign the incident to the software developer/consultant who performed the correction.
- Resolved/Waiting
  - After checking it, the issue has been considered as Resolved by a contractor representative, depending on the type of issue
  - The incident is assigned to the reporting user
- Closed
  - The reporting user has checked the solution for the respective incident considered it acceptable, so that the status changes to “Closed”.

Below may be seen the JIRA workflow diagram with statuses and resolutions:

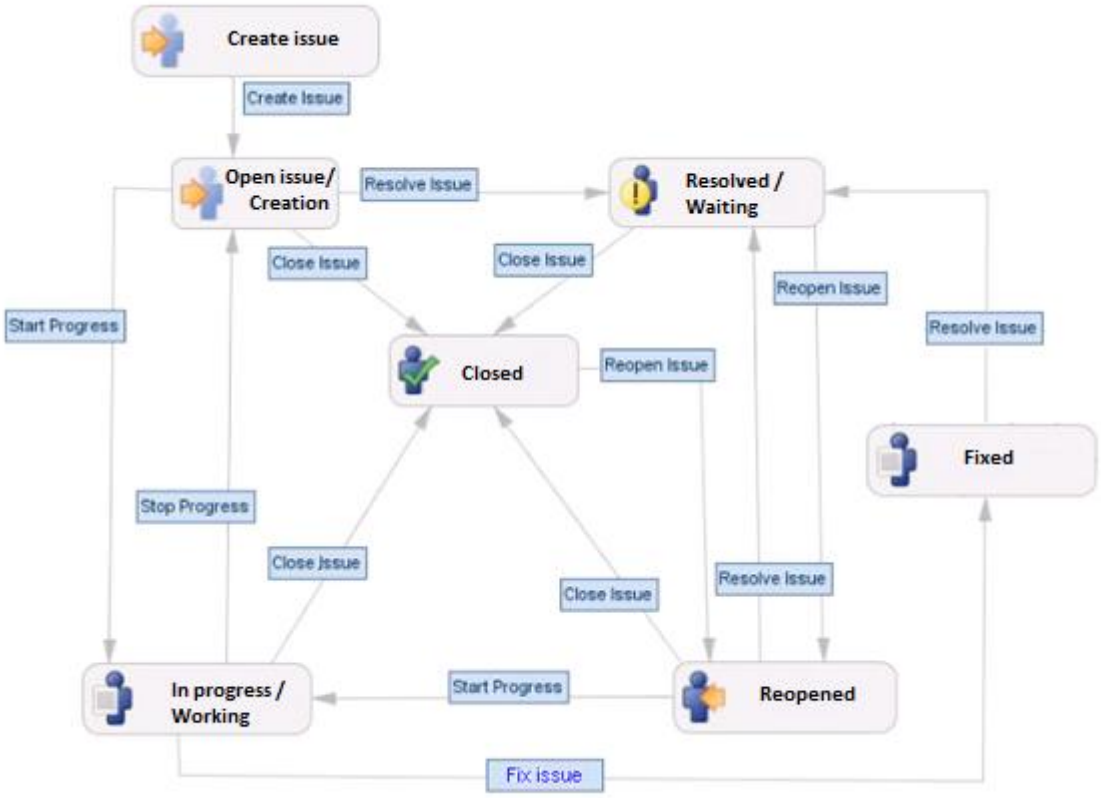

Figure 1. Statuses and resolutions on the JIRA workflow

The workload associated to a JIRA incident is considered finished only after its reporter has checked the solution, is satisfied by it and marks the issue as closed.

4.2 Defect priority classification

When reporting ESTABLISH issues, the following defect priority classification will be used:

Issue priority	Description
High	Serious systemic problems (e.g. complete or partial system unavailability, instability, or non-usability, including serious performance degradations). Loss of critical business functionalities with no reasonable workarounds. Problems with the website that limit core user functionalities; or seriously degrade user experience; or result in display of incorrect data; or negatively affect the public image of the ESTABLISH project.
Medium	Non critical systemic problems that do not cause serious loss of

Issue priority	Description
	usability. Loss of non-critical business functionalities. Loss of business functionalities but with reasonable workarounds present.
Low	Minor inconveniences with business functionalities. Trivial website issues that cause little or no inconvenience or degradation of user experience. Website issues that affect less than 1% of public users.

5. Testing criteria

5.1 Entry/Exist criteria

Before either system or acceptance execution can commence, it is a necessary condition that the entry criteria have been satisfied to ensure that testing progresses as smoothly as possible.

The Testing phase entry criteria are:

- The Test Plan (this document) accepted.
- The Acceptance Test Specification Document accepted for the ongoing testing phase.
- A stability test has been satisfactory: all major functionality for the ongoing testing phase is accessible and present and tests as well as pre-prod environments are ready and stable.
    - Test environment ready for use.
    - Pre-production environment ready for use.
    - Website data sources initialized with data and all necessary migrations performed from the development environments.
    - All technical requirements for the establishment of the environment and the website are ready.
    - Testing personnel are familiar with the system functionality.
    - Formal defect tracking mechanism/process is established.

The Testing phase exit criteria are:

- All required test cases fully executed for the appropriate test/development phase.
- Agreement on the test results by all the users involved in testing process.
- Partial acceptance of the test results from ESTABLISH team for intermediate stages of development.
- Final acceptance of the test results from ESTABLISH team.

5.2 Test pass/fail criteria

All test conditions that satisfy their expected results passed.

All Test Cases that satisfy all of their expected results passed.

Test Conditions and Cases that do not satisfy their expected results failed and have a defect of appropriate priority raised go to retesting and then a fixed version shall be released.

5.3 Test suspension criteria

A test phase may be suspended and handed back to development team if any of the following apply:

- The number of defects is such that a majority of test cases were not successfully completed.
- Major in scope functionality not delivered.

5.4 Acceptance criteria

Partial and final acceptance of ESTABLISH testing process achieved after meeting the following:

- The Test Plan (this document) is accepted.
- The Acceptance Test Specification Document is accepted for the ongoing testing phase (iteration).
- During Acceptance Testing for the ongoing testing phases (iterations) all the non-regression and regression tests developed for the ongoing iteration were performed.
- “Factory acceptance test” document created on the basis of ESTABLISH JIRA project show all “High” reported incidents as “Closed”.

6. Test reporting

Viewing the list of issues recorded in the project directly on JIRA

A summary of the status of all incidents relating to the ESTABLISH is available by selecting the “Issues” tab from the JIRA system.

A complete list of all the incidents for ESTABLISH is available by selecting the “Summary” tab on the left of the Dashboard and then the “Filters” tab.

The available filters are to view: all the incidents, resolved recently, outstanding, added recently, unscheduled, updated recently, assigned to the current user, most important, reported by the current user.

In order to view all the incidents from the project, a JIRA user will select the filter “All” and a page with the most important details of the issues will be displayed.

For each displayed incident, some important details will be shown, such as (in display order):

- Type of incident
- Number of the incident
- A description of the incident
- Assignee and reporter
- Priority

- Status
- Resolution
- Date when the incident was created
- Date when the incident was updated
- Due date.

Testing report

Based on both Acceptance Test Specification Document and ESTABLISH JIRA project, at the end of each testing phase the contractor will create a testing report - Factory Acceptance Test Document showing the following:

- Test cases status. This will show the list of test cases presented in the "Test Plan Document" along with their status. Test cases status will be presented using the following template:

Short description	Run ID	Version	Date	Tester	Not run	Passed	Failed	Observation
TC.001.BN								
Short description	RUN_1	1.0	…	…	X			…
	RUN_1	1.0	…	…		X		…

- Test executive summary. This synopsis will present the number of incidents created in JIRA for each priority level, showing how many incidents have been created, how many have been closed and how many are still in progress of solving.

Priority level	Open	In progress	Reopened	Fixed	Resolved	Closed	TOTAL
Low							
Medium							
High							
TOTAL							

7. Test scenarios and test cases

7.1 Optimized City and Mobility Planning (Prodevelop)

7.1.1 Test Scenarios

This chapter contains a non-exhaustive list of identified Test Scenarios for the Spanish pilot. This list will be completed in the next phase, after all details of the technical solution will be known. As is very early to produce complete Test Case definitions, only a sample is provided.

In the use case on *Optimized City and Mobility Planning*, the Spanish consortium will build an advanced web application for providing planning services and mobility information both for citizens and for city authorities considering relevant information such as contamination or traffic conditions. This pilot will also enable gamification activities to motivate people, to improve efficiency of the transport system and promote sustainable habits in the context of transport mobility.

The use case includes the development of a **dashboard** that displays the status of metrics and key performance indicators (KPIs) related with it. The essential features of the dashboard include a customizable interface and the ability to pull real-time data. The dashboard will have two perspectives, the first for the authorities and the second for the citizens. The functions reserved to City Authority users are:

- Generate predictions about pollution and traffic load;
- Make simulations of pollution protocols;
- Create new pollution protocols;
- Monitor pollution values;
- Receive recommendations/alerts about the pollution values;

Citizens will have access to:

- Route planning;
- Receive recommendations/alerts about the pollution values;
- Recommendations about mobility when pollution protocols are activated (public transport, parking);

Another system component is the **Mobile App** for citizens. The application will be installed on smartphones and will have the same functions as the web applications for citizens.

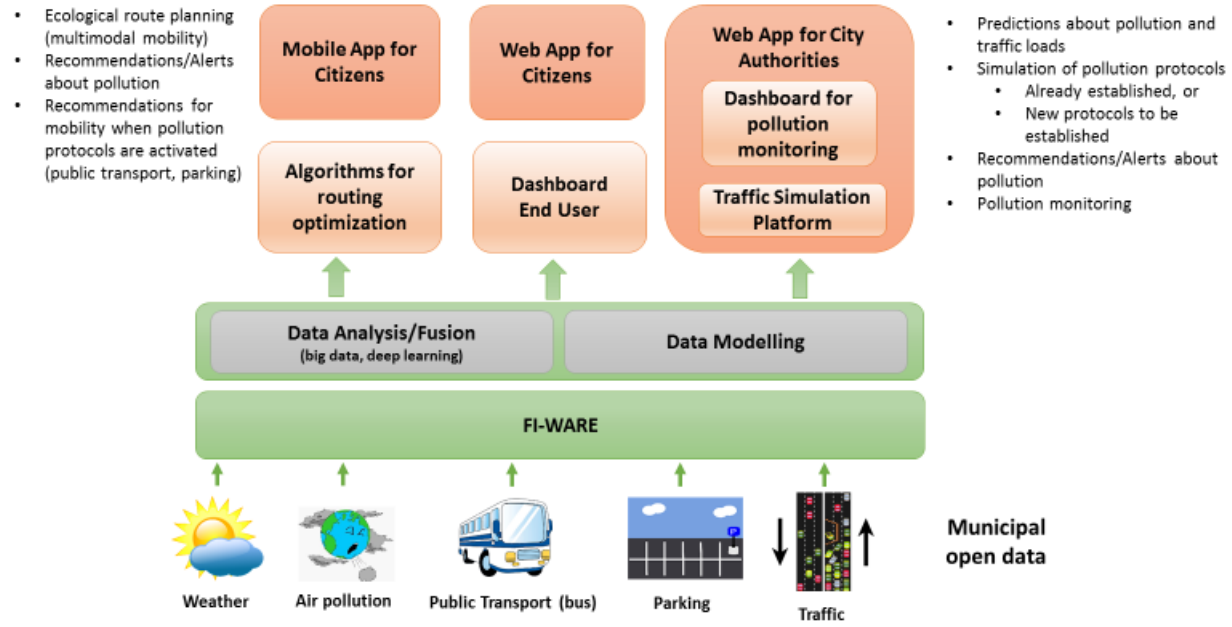

Figure 2. First approach for the implementation of Optimized City and Mobility Planning use case.

In parallel with the detailed technical specifications and sometimes even in parallel with the development, the table below will be completed with all functionalities and associate test scenarios.

Component	Subcomponent	Function	Test Scenario	Code TS
Web Application	City Authority Dashboard	Generate predictions about pollution and traffic load	Identified Test Scenario 1	TS.DASH.01
			Identified Test Scenario 2	TS.DASH.02
…	…	…	…	…
Web Application	City Authority Dashboard	Make simulations of pollution protocols;	…	…
Web Application	City Authority Dashboard	Make simulations of pollution protocols;	…	…
Web Application	City Authority Dashboard	Create new pollution protocols;	…	…
Web Application	City Authority Dashboard	Monitor pollution values;	…	…
Web Application	City Authority Dashboard	Receive recommendations/alerts about the pollution values	…	…
…	…	…	…	…

Web Application	City Authority Dashboard	User Management	Create new City Authority users	TS.USER.01
			Modify City Authority users information	TS.USER.02
			Delete a City Authority user	TS.USER.03
			Associate permissions	TS.USER.04
…	…	…	…	…

7.1.2 Test Cases

For each Test Scenario in the table, a set of Test Cases will be identified and written in the template below. The template contains also sample data about test cases of the test scenario TS.USER.01 – Create new City Authority Users.

TS Code	**TS.USER.01**
TS Name	Create new City Authority users
TC Code	TS.USER.01-TC.01
TC Version	1.0 – initial test
TC Name	**Create new City Authority User – SUCCESS / NO DUPLICATE / NO SET PERMISSIONS**
Component	Web Application
Sub-Component	City Authority Dashboard
Function	User Management
Actor	City Authority User having Admin profile
Special Requirements	Test case needs preparation of a test data set including several real or fictive persons with: first name, last name, valid email address (business address inside the organization), wanted profile, permissions.
Pre-Condition	User is logged in the Web applications and has Admin profile. He clicked on menu Administration and then on sub-menu User management. A list of users is shown with appropriate column headers and filter fields and criteria. Above the list a button “Add new user” is displayed and enabled.
Post-Condition	When transaction is successful, a new user will be created in the system and will be displayed in the user list, having the correct personal information, email address, profile and permissions.
Date	
Tester	

Step	**Actions and Data input**	**Expected Result**	**Obtained Result**	**PASSED/ FAILED**

1	User clicks on button “Add new user”.	An input/update data form is displayed. All fields are empty. Some fields are mandatory. Fields First Name, Last Name, email address are free text fields Field User profile is a dropdown box filled with values: GUEST, TYPE1, TYPE2, … Button Save is displayed and active Button Cancel is displayed and active List of special permissions is displayed but empty.		
2	User enters data in all mandatory and non-mandatory fields.	After each input, cursor is navigating to next field. At the end, the cursor will be placed on button “Save”.		
3	User clicks on “Save” button.	Data is saved, the form is closing and entered data for the new user is correctly and completely displayed in user list.		