D3.7 Citizen Direct Feedback

Release Notes

Project acronym:	**FLOOD-serv**
Project full title:	**Public FLOOD Emergency and Awareness SERvice**
Grant agreement no.:	**693599**
Responsible:	**ANO**
Contributors:	**Pedro Leite, Sergio Almeida**
Document Reference:	**D3.7**
Dissemination Level:	**PU**
Version:	**1.7**
Date:	**02/12/19**

History

Version	*Date*	*Modification reason*	*Modified by*
0.1	03/01/2017	Initial draft: structure of contents	Pedro Leite
0.2	27/01/2017	Overall content	Pedro Leite
0.3	29/01/2017	Final content and revision	Pedro Leite
Final	29/01/2017	Final content and revision	Pedro Leite
1.1	20/11/2019	Revision after feedback from EC	Pedro Leite
1.2	21/11/2019	Chapters 2, 3, 4 and 5 - reviewed, updated and complemented	Pedro Leite
1.3	22/11/2019	Chapter 6 – reviewed, updated and complemented test cases and results Chapter 7 – updated release notes and links	Pedro Leite
1.4	29/11/2019	Chapter 4 – tech specs added Chapter 9 and appendix I added Total Number of chapters changed Changes in Response to Final Review added	Pedro Leite
1.5	29/11/2019	1.3 Changes in Response to Final Review updated	Pedro Leite
1.6	29/11/2019	1.3 Changes in Response to Final Review updated	Pedro Leite

1.7	02/11/2019	Added references to user guide in Chapter 11 – Appendix II	Pedro Leite

Table of contents

List of figures

List of tables

List of abbreviations

<Abbreviation>	**<Explanation>**
CDF	Citizen Direct Feedback
EMC	Emergency Management Console
SW	Semantic Wiki
TMS	Territory Monitoring System
DM	Decision Maker
F	Facilitator
FE	Flood Emergency Expert
ER	Emergency Responder
SMC	Social Media Component
JSON	JavaScript Object Notation

Naming Conventions and Terminology

Naming Domain	*Standard*	*Examples*	*Link*
Agile Development	SCRUM	Epic, theme, user story	http://www.scrumguides.org/
Requirements	IREB	Use case, non-functional requirements	https://www.ireb.org/content/downloads/1-cpre-glossary/ireb_cpre_glossary_16_en.pdf
QA	ISTQB	System test, unit test, integration test, defect	http://www.istqb.org/downloads/ glossary.html

Table 1 Naming Conventions Industry Standards

Executive summary

D3.7 Citizen Direct Feedback is the seventh deliverable from WP3. This is a software deliverable about development of the CDF tool.

The CDF development is based on the D3.2 document, which deals with the functional specifications through the definition of the user stories for each component composing the FLOOD-serv platform and the description of the technical specification of each component: structure, data model, interfaces, services, data providers and implementation environment.

D3.7 is being developed at the same time as D3.3, D3.4, D3.5, and D3.6, which are reporting the development of the rest of components composing the FLOOD-serv platform, i.e.: the Social media component (SMC), the Emergency Management Console (EMC), the Semantic Wiki (SW), and the Territory Monitoring System (TMS), respectively.

1 Introduction

1.1 Purpose of the Document

The goal of this document is to provide the release notes for the Citizen Direct Feedback. The document is based on D3.1 and D3.2.

1.2 Structure of the Document

The document is organized as in the following:

- Chapter one: Introduction
- Chapter two: Overall approach and methodology
- Chapter three: Overview of user stories implemented
- Chapter four: Technical specifications
- Chapter five: System test case repository
- Chapter six: Test cases overview
- Chapter seven: Test cases and results
- Chapter eight: Release notes
- Chapter nine. Work developed and Conclusions
- Chapter ten: Appendix I – API Doc

For the reference, before the *Changes in Response to Final Review*, the document was organized:

- Chapter one: Introduction
- Chapter two: Overall approach and methodology
- Chapter three: Overview of user stories implemented
- Chapter four: System test case repository
- Chapter five: Test cases overview
- Chapter six: Test cases and results
- Chapter seven: Release notes

1.3 Changes in Response to Final Review

Observations	*Actions and Changes*
It is unclear what exactly was achieved under the WP3 in the areas of the Territory Monitoring System and Citizen Direct Feedback components.	**All the user stories previously identified were implemented with success and the component is up and running and fully integrated in the system**. To underline this, the chapters with the user stories implemented – chapter 3 was updated, Chapter 4 – Tech Specs was added, Chapter 6 and Chapter 7 for the test cases and results was updated, chapter 8 with release notes was updated with the currently working links for all 5 pilots, chapter 9 with work developed and conclusions was added, Appendix I was added.

	The CDF is essential for the objective of the Project to be citizen-centric and to have a two-way communication with citizens. CDF enables the receipt and systematic processing of feedback from citizens (by the use of the CDF Mobile App or the FLOOD-serv Portal), in the CDF back office interface, public administrators, assess information from citizens, send it for further processing for issue resolution and communicate with citizens: ⌡ Filter and process any incoming issue reported via app or portal; ⌡ A Workflow management system for proper address the issues automatically register; o Allowing the public authority to give feedback to the citizen; ⌡ Broadcasting messaging tool with sms integration; ⌡ P2P messaging tool with sms integration; ⌡ Build a database of issues with workflow capabilities and with integration API for the Portal and the EMC; ⌡ Build a database of entities/citizens; ⌡ Build smart forms for dedicated workflows and publish them on the Portal via API; o This allows more formal communications and process flows between citizens and the public authorities. ⌡ Allow the public authority to setup the application and workflows as desired with full customization; The CDF also allows sending mass messages/alerts to citizens via the Mobile App.
more substantial description of the content of listed components should be provided in the Deliverables D3.3, D3.5 - D3.7 accordingly to the remarks of the present report and its Annex 1 - Deliverables due for the Period 2/Final review.	**The whole document was revised and more details was added**: the chapters with the user stories implemented – chapter 3 was updated, Chapter 4 – Tech Specs was added, Chapter 6 and Chapter 7 for the test cases and results was updated, chapter 8 with release notes was updated with the currently working links for all 5 pilots, chapter 9 with work developed and conclusions was added, Appendix I was added.

The document content is not of sufficient quality as provided user cases are very basic, sometimes unrealistic. For example, examples of messages/broadcasts (6.1.16 Results – Send Broadcast emergency) are of little usefulness for the emergency actors as no information on the location and type of flood (river overflow, dike breach, percolation, ...) is provided. Duoro river is 900 km long and at least geographical coordinates of the emitted and validated messages should be an integral part of the broadcast content.	**The whole document was revised and more details was added – more relevant for this point the updated user stories**: the chapters with the user stories implemented – chapter 3 was updated, Chapter 4 – Tech Specs was added, Chapter 6 and Chapter 7 for the test cases and results was updated, chapter 8 with release notes was updated with the currently working links for all 5 pilots, chapter 9 with work developed and conclusions was added, Appendix I was added. **Answering the specific issue raised – messages with location - we don't need to use "location" for filtering because, CDF App is instantiated for the city/town where those people live, not for a whole region. Only people in that city who have installed the App receive messages.** Detailed information about location of the event can be issued in text. The broadcast message intends to be a quick and agile means of alerting citizens of a specific city (instance of the FLOOD-Serv for Genova for example), of an emergency occurrence or any other event the authorities deem fit. So, location does not apply as a mandatory information – nevertheless it can include for another functional context in a future iteration. In contrast, the tickets sent by the citizens via the CDF app or Portal, automatically or manually share the location of the issue, because this is a vital piece of information for the public authority, to further process and cross check.
The document lacks conclusions with respect to compliance of the developed component with the technical specifications (D3.2)	**The whole document was revised and more details was added – more relevant for this point chapter 9 was added,** chapter 3 was updated, as chapter 7 with more details on the test cases. The release notes were also updated.
In addition, the information on accessing the software is erroneous as the access link http://flood-serv.ano.demos.pt/ to this API is not working in August 2019 what does not allow an external potential user for practising with the developed component.	The demo environment, originally referenced in the document was out of commission. The updated links are provided in Chapter 8 – release notes, chapter 4 added and appendix I with the API Docs Updated links: For Decision Makers and/or Operators you can access to the CDF Backoffice directly in the FLOOD-Serv Platform or directly by typing in:

	https://bilbao-floodserv-saas.ano.pt/ https://bratislava-floodserv-saas.ano.pt/ https://genova-floodserv-saas.ano.pt/ https://tulcea-floodserv-saas.ano.pt/ https://vnfamalicao-floodserv-saas.ano.pt/ The credentials to access are: • User: salmeida123 • Password: 123 To access to the CDF mobile app for Citizens: You can download it directly via the FLOOD-Serv Platform or using the direct link: https://tulcea-floodserv-saas.ano.pt/tulcea/images/FLOODserv_1.0.0.4-tulcea.apk https://bilbao-floodserv-saas.ano.pt/bilbao/images/FLOODserv_1.0.0.4-bilbao.apk https://genova-floodserv-saas.ano.pt/genova/images/FLOODserv_1.0.0.4-genova.apk https://bratislava-floodserv-saas.ano.pt/bratislava/images/FLOODserv_1.0.0.4-bratislava.apk https://vnfamalicao-floodserv-saas.ano.pt/vnfamalicao/images/FLOODserv_1.0.0.4-vnfamalicao.apk
D4.4 It is also unclear how CDF, which creates time and space distributed information, can provide a support in the decision-making process, without a layer of data fusion, cross-validation and knowledge data base for a return of experience.	**The whole document was revised and more details was added – more relevant for this point chapter 7 was fully updated and chapter 4 with the tech specs added.** The data collected by the CDF (as TMS and SMC) should be collected to the platform and EMC as these are the system's optimum locations for cross validation and ultimate analysis and decision making. The main goal is to collect issues and return feedback.

2 Overall approach and methodology

The tests described in this document demonstrate that the CDF component (D3.7) has been successfully implemented in all languages and the new features identified in D3.1 and D3.2 have been implemented.

As the next steps integration tests will be implemented as part of WP4

WP5 is about user acceptance tests, which is the last phase of a software testing process. During UAT (User Acceptance Testing), actual software users test the software to make sure that it works in real-world scenarios, according to specifications.

The overall approach applied is SCRUM, consequently the results and documentation the software delivered in D3.7 was following the sprints and user stories implemented.

In SCRUM the tasks are divided into time boxes (small time frames) to deliver specific features in the release so that the working software build can be delivered after each iteration. Builds are incremental in terms of features; the final build of D3.7 has all the features.

Test types and coverage

The tests cover the functionality of the CDF component only.

The integration tests performed was the CDF REST API, through various calls to the API.

Role of the tester

- Ensure End-user satisfaction through delivery of high-quality software.
- Engagement is early during the project from sprint planning.
- Discuss and understand each user story and then decide on acceptance criteria for the same.
- Define activities for themselves to estimate time, updating test cases as and when changes appear, complete testing within the sprint time etc.
- Develop test cases as per the story acceptance criteria and change whenever there is a change in story.
- Deliver high quality software iteratively from a couple of weeks to a couple of months.
- Ensure user stories get clarified where there is insufficient information.
- Break user stories into different testing tasks.
- Decide each story test coverage

3 Overview of user stories implemented

The main goal of the CDF is to provide a two-way communication between citizens and the public authorities currently using the FLOOD-Serv platform. For Citizens a mobile application was created for easy usage and an API that will implement smart forms to be used by the main portal - these are the frontoffice. For the public authorities, a web backoffice was created to:

- Filter and process any incoming issue reported via app or portal;
- A Workflow management system for proper address the issues automatically register;
    - Allowing the public authority to give feedback to the citizen;
- Broadcasting messaging tool with sms integration;
- P2P messaging tool with sms integration;
- Build a database of issues with workflow capabilities and with integration API for the Portal and the EMC;
- Build a database of entities/citizens;
- Build smart forms for dedicated workflows and publish them on the Portal via API;
    - This allows more formal communications and process flows between citizens and the public authorities.
- Allow the public authority to setup the application and workflows as desired with full customization;

The user stories implemented to accomplish with the non-functional, functional and technical requirements of the CDF are collected in the following table. To consider the CDF as totally finished, the 8 user stories collected in this table (from **USCDF1** to **USCDF8**) have also to be implemented.

ID	Summary	Description
USCDF1	Communicate early flood signs	As eCitizen or Certified Citizen/Observer I want to warn the authorities of a potential flood risk. Acceptance criteria: 1) Users can send information about a potential flood risk: Title and Text Description, 2) Users can send automatically geo coordinates of their location or the occurrence (Share current location or specify address) 3) Users can also send photo or video about the occurrence; 4) If defined as a Certified Observe, the issue will can be treated as high risk and thus not pass through the filter stage; 5) Send the alert via the web platform or mobile app.
USCDF2	Filter citizen communications	As a facilitator or flood emergency expert I want to be able to filter occurrences reported by eCitizens and Certified Citizens/Observers. Acceptance criteria: 1) Users can filter citizen communications on a specific area; 2) Can search based on criteria; 3) Can search history of occurrences; 4) Can verify the trustiness of the alerts and define trusted users.

USCDF3	Broadcast emergency information	As a facilitator or flood emergency expert I want to communicate to all registered users on important information about a specific event. Acceptance criteria: 1) Users can send a broadcast message to all the users via portal, mobile application and/or SMS; 2) Filter user base for broadcast; 3) Citizens receive SMS or alert via mobile application.
USCDF4	Follow-up on specific issues	As an eCitizen or Certified Citizen/Observer I want to be able to check the actions taken on a specific matter previously reported on the platform. Acceptance criteria: 1) Users can check the status and actions on a specific matter; 2) Can search based on criteria for issues; 3) Can get details on those actions.

USCDF5	Official Requests for damage control and other more formal communications	As an eCitizen, I want to be able to report damages related to a flood occurrence to the authorities, through the FLOOD-Serv platform. Acceptance criteria: 1) Users can report on damages using a simple form online; 2) they can attach documents and other files, as pre-defined by the organization using the platform; 3) The citizen will receive a receipt as proof of the report; 4) The citizen can check the status of the report; 5) The authorities will be able to process the reports sent by citizens and process them internally; 6) The authorities will be able to report on the status of the issue for the original reporter;

USCDF6	Registration via mobile application	As an eCitizen or Certified Citizen/Observer I want to register in the Flood-Serv platform via the mobile application without accessing the online portal. Acceptance criteria: 1) Users provide the basic user information; 2) They receive a confirmation email of the registration 3) They can then access the portal and mobile application
USCDF7	Certification of Certified Observers	As a facilitator or flood emergency expert I want to certify a registered user as a certified observer. Acceptance criteria: 1) User will check the profile of the citizen 2) Will check the user as certified observer 3) The citizen will then be granted all the functionalities associated with the role Certified Observer.

USCDF8	Definition of Smart Forms for Formal Communications	As a facilitator or flood emergency expert I want to define smart forms to enable citizens to interact with the organization in a more formal manner Acceptance criteria: 1) Users can define the theme associated 2) Users can define the fields need for the form 3) Users can define attachments needed to the form 4) Users can publish the form 5) Users can alter the details of the Form

The CDF backoffice is accessible via web for:

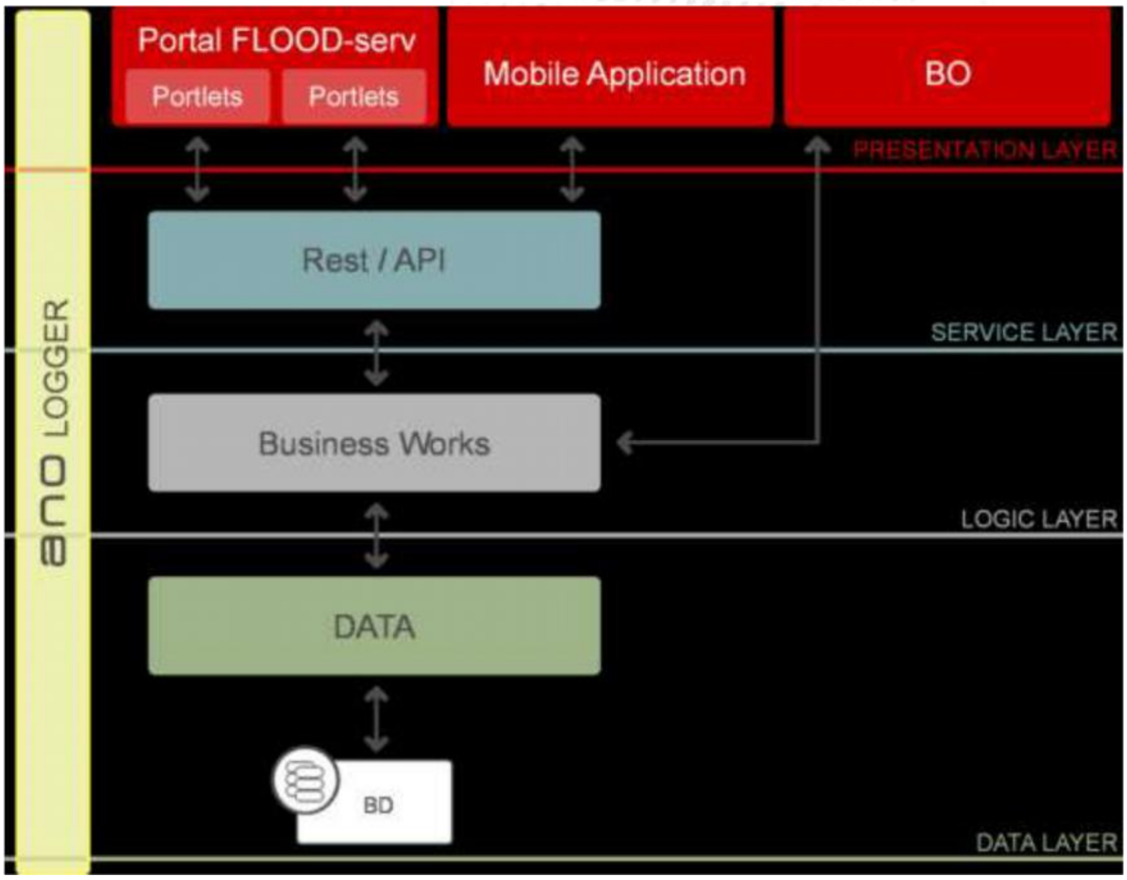
- Decision Makers
- Facilitator/Flood Expert
- Emergency Responders
- System Operator

The CDF frontoffice is accessible via dedicated app or via integration with the portal:

- Citizens

4 Technical Specifications

The CDF component is a service-oriented application with a multi-layer perspective. The architecture of the CDF was designed taking in account the industries best practices allowing scalability, modularity and code efficiency. The various layers allow that data, logic, API and graphical user interfaces are separated and that can be managed separately.



Like the figure shows, each layer is designed to be scalable and with interoperability as a principal mindset. As CDF is at the same time, data silo, web backoffice, web frontoffice via integration with the FlOOD-Serv Portal and mobile application, its API is a core feature and it was developed specifically for the project.

The macro technology stack that support these layers are:

- Logger
- Log4j
- Data
- Java and Hibernate
- Business Works
- Java
- Spring
- Spring Security
- API
- Jersey
- UI
- Backoffice

- JSF
- Primefaces
- Mobile
- Android SDK

In terms of infrastructure, the CDF is supported by a series of virtualized servers (using VMware vSphere Hypervisor (ESXi)):

- Underline OS for each virtual machine: CentOS 6.X
- Database: Oracle DB 11g
- Java Server: Glassfish 5

5 System Test Case Repository

This chapter provides an overview of the current system tests.

The tests described in this document demonstrates that the Citizen Direct Feedback is implemented in all languages and the features identified in D3.1 and D3.2 have been successfully implemented as part of the task T3.9.

5.1 System Tests

The core focus of the system tests is to test the Citizen Direct Feedback component - without integration with the other modules - as a black box as seen by the user. This test level is being performed by dedicated experts (testers, test manager).

WP5 is about user acceptance tests, which is the last phase of a software testing process. During UAT (User Acceptance Testing), actual software users test the software to make sure that it works in real-world scenarios, according to specifications.

5.2 Baseline for System Tests - Preconditions

In order to extract reproduceable and consistent results from executing the system tests they must be performed in a defined environment. Besides system level requirements (database is up and running), there are other internal settings that must be set accordingly. These settings are called preconditions and this chapter lists some of the most relevant ones which are present in most of the verification tests.

Id	Title	Setup
PRE 1	Activated Facilitator	) client exists) user with facilitator role has been added by the FLOOD-serv platform) password for facilitator has been set
PRE 2	Activated Flood Expert	) client exists) user with flood expert role has been added by the FLOOD-serv platform) password for flood expert has been set
PRE 3	Activated Decision Maker	) client exists) user with decision maker role has been added by the FLOOD-serv platform) password for decision maker has been set
PRE 4	Activated Emergency Responder	) client exists) user with emergency responder role has been added by the system operator) password for emergency responder has been set

PRE 5	Activated eCitizen	⌡ client exists ⌡ user with eCitizen role has been registered into the system ⌡ password for eCitizen has been set

6 Test cases – overview

We have grouped the tests carried out into clusters: such related to citizen input, alters and requests treatment and citizen feedback.

	Italian Version	Portuguese Version	Romanian Version	Slovakian Version	Spanish Version
Citizen Input					
Logon to App	X	X	X	X	X
Send alert via mobile app	X	X	X	X	X
Send Damage Report	X	X	X	X	X
Registration via mobile application	X	X	X	X	X
Alerts and requests Treatment					
Filter alerts and communications	X	X	X	X	X
Process alert, communication or damage report	X	X	X	X	X
Certification of citizen	X	X	X	X	X
Define Smart Form	X	X	X	X	X
Citizen Feedback					
Send Broadcast emergency	X	X	X	X	X
Receive Broadcast emergency	X	X	X	X	X
Check alert or communcation status	X	X	X	X	X
WS Rest API	X	X	X	X	X

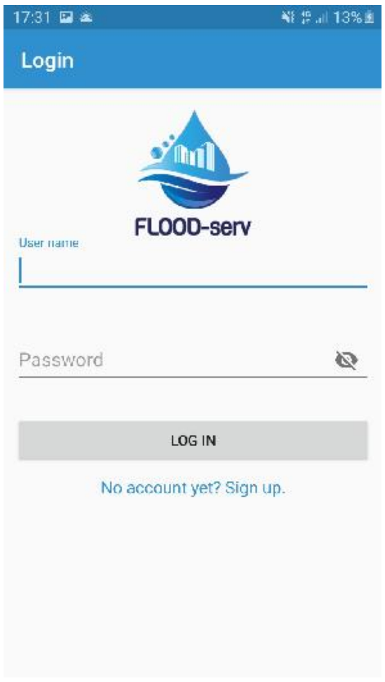
7 Test Cases and Results

7.1 Data Collection – Test Cases and Results

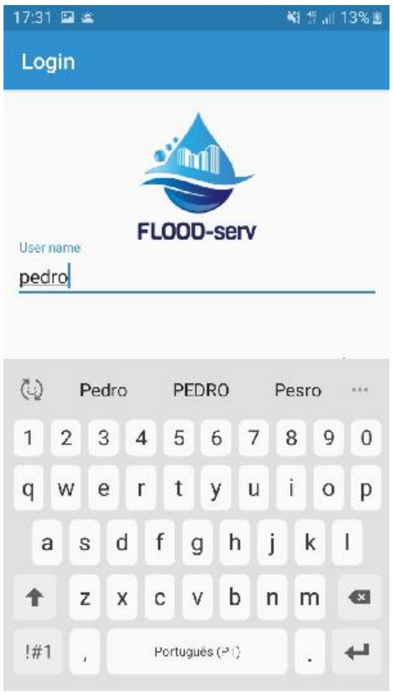
7.1.1 Test Case – Logon to App

Logon to App	
Test Type:	Manual
Status:	Final
Preconditions:	- PRE 5 – Activated eCitizen - eCitizen has mobile application installed and registered
Steps to complete:	1. The eCitizen accesses and logs on to the mobile application
Expected Outcome:	1. The eCitizen is able to log on to the application for further usage

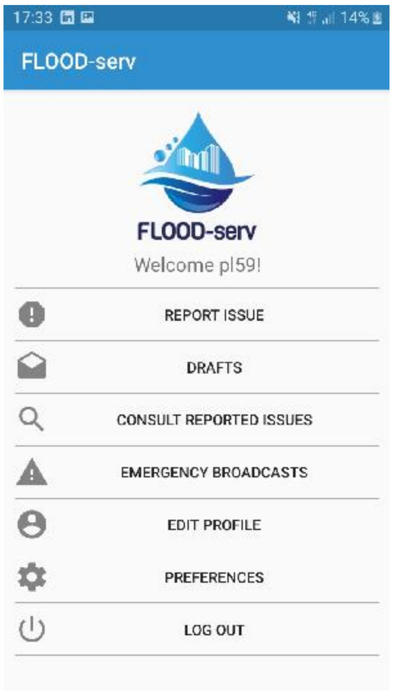
7.1.2 Results – Logon to App



Login screen upon opening the app.



User and password input

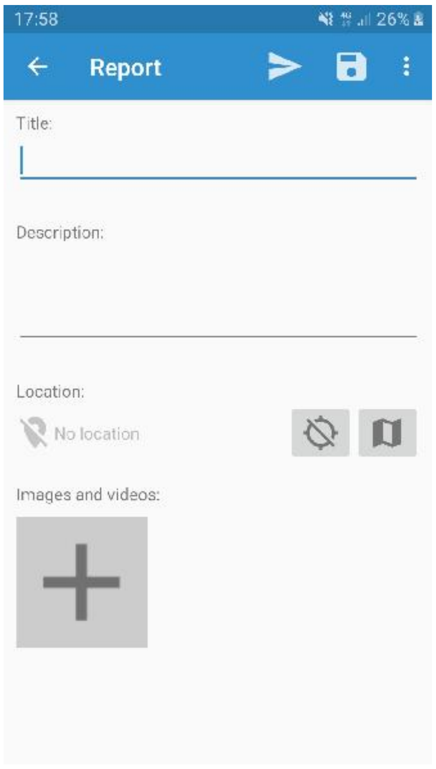


Access to main menu granted

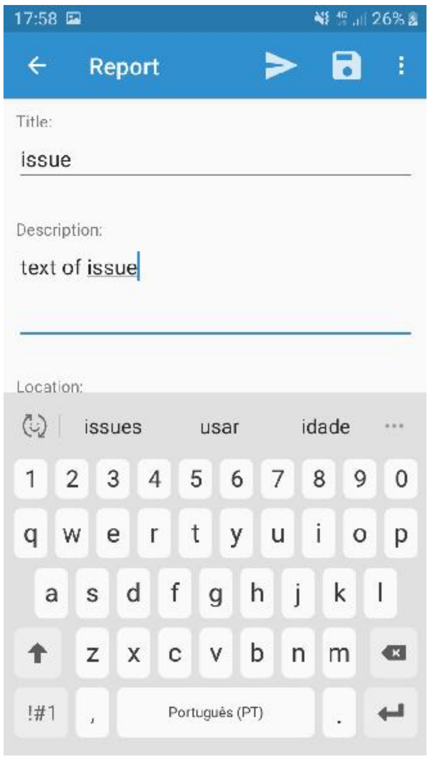
7.1.3 Test Case – Send alert via mobile app

Send alert via mobile app	
Test Type:	Manual
Status:	Final
Preconditions:	• PRE 5 – Activated eCitizen • eCitizen has mobile application installed is registered
Steps to complete:	1. The eCitizen registers information to send 1. Title and Description 2. Can add photo or video 3. Can share location 2. Sends information to the platform
Expected Outcome:	1. The eCitizen is able report an alert or ticked to the platform

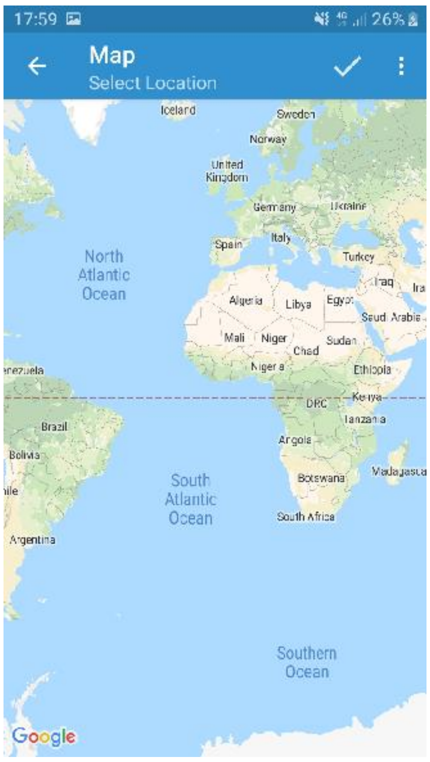
7.1.4 Results – Send alert via mobile app



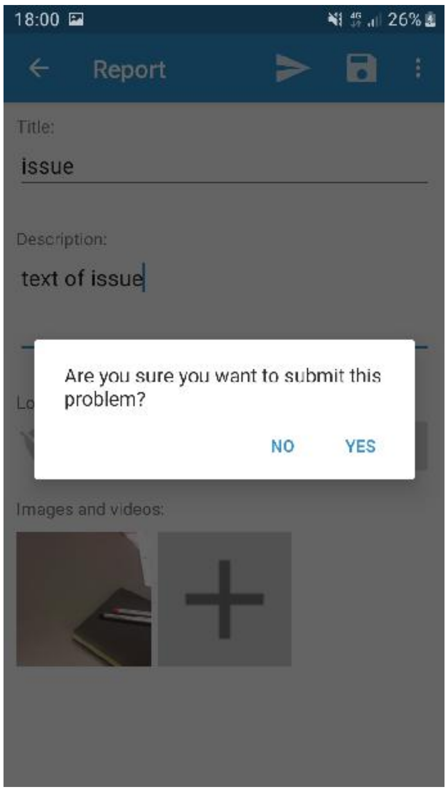
Select "Report issue" on the main menu



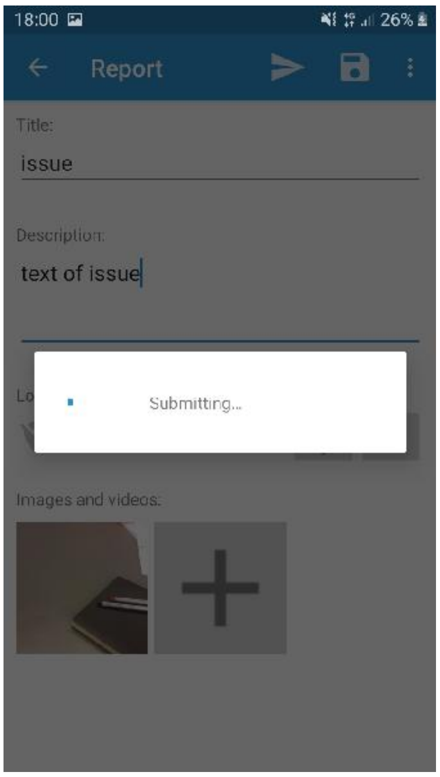
Fill in the request information



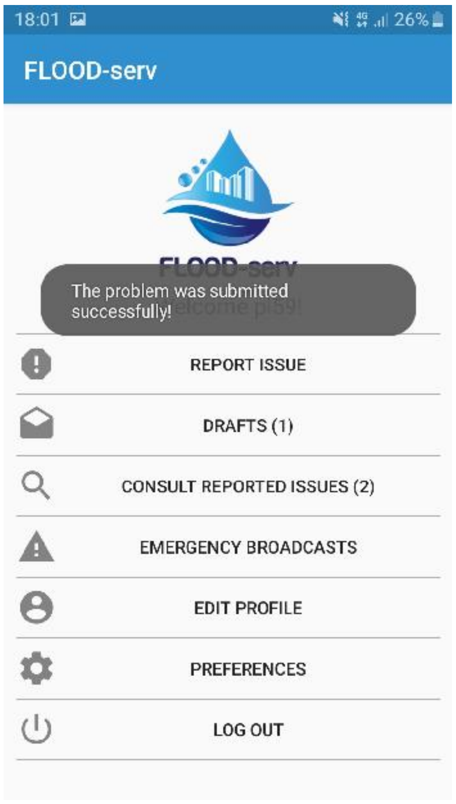
Share automatically the location or select manually



Click "Submit"



Submitting…



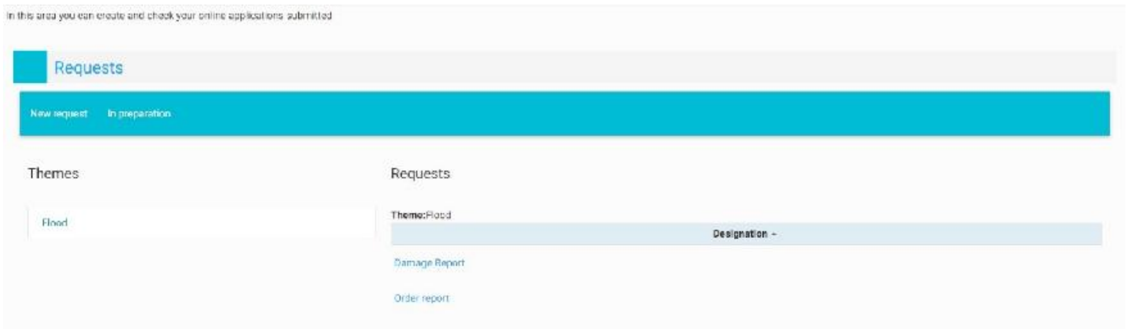
The problem was submitted successfully!

Report submitted

7.1.5 Test Case – Send Damage Report

Send Damage Report	
Test Type:	Manual
Status:	Final
Preconditions:	) PRE 5 – Activated eCitizen eCitizen has mobile application installed is registered
Steps to complete:	1. eCitizen can report on damages using a smart form online; 2. eCitizen can attach documents and other files, as pre-defined by the organization using the platform; 3. eCitizen will receive a receipt as proof of the report; 4. eCitizen can after check the status of the report; 5. The issue will be registered in the Backoffice for further processing
Expected Outcome:	2. The eCitizen is able report a damage that can be processed by the platform operator and receive feedback on it

7.1.6 Results – Send Damage Report


In this area you can create and check your online applications submitted

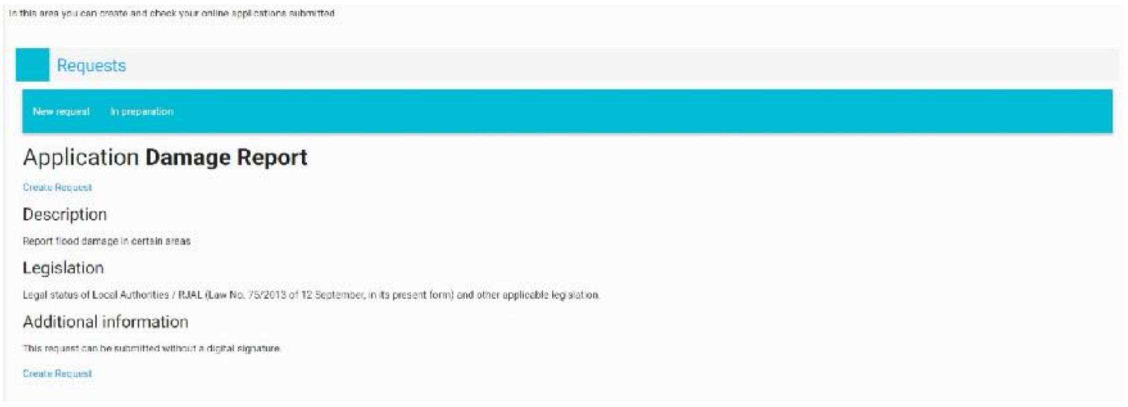
Requests

Themes

Requests

Theme: Flood

Designation ˜
Damage Report
Order report


In this area you can create and check your online applications submitted

Requests

Application **Damage Report**

Description

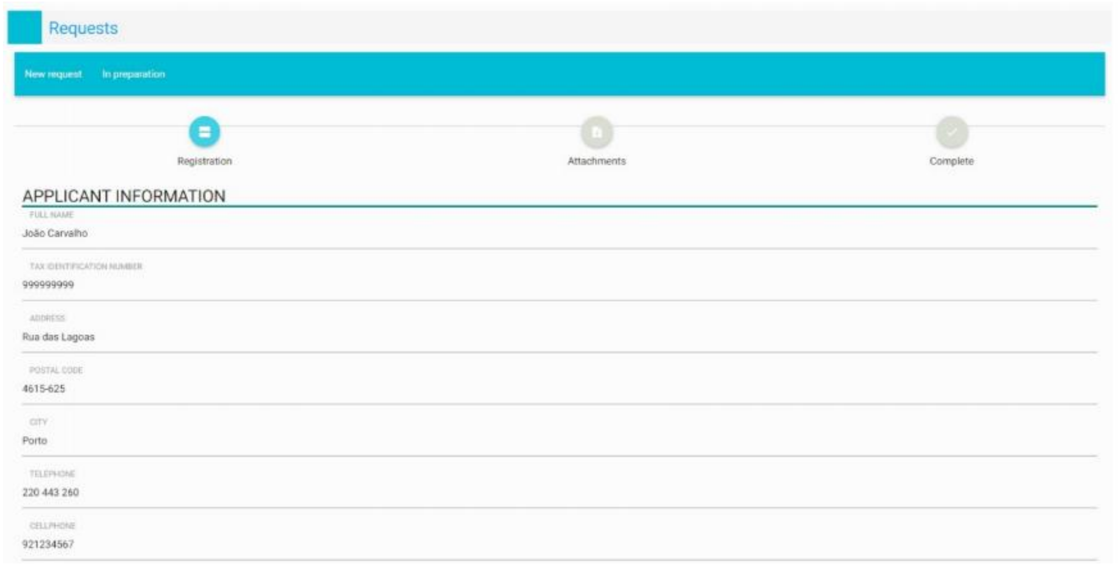
Report flood damage in certain areas

Legislation

Legal status of Local Authorities / RJAL (Law No. 75/2013 of 12 September, in its present form) and other applicable legislation

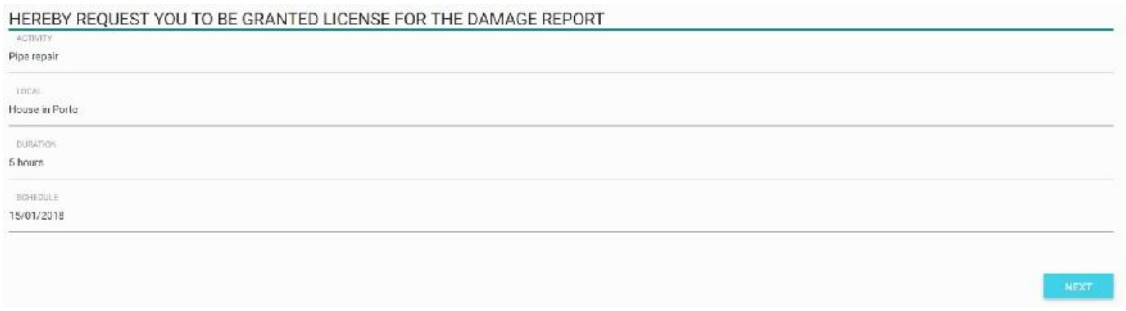
Additional information

This request can be submitted without a digital signature.


Requests

APPLICANT INFORMATION

FULL NAME	João Carvalho
TAX IDENTIFICATION NUMBER	999999999
ADDRESS	Rua das Lagoas
POSTAL CODE	4615-625
CITY	Porto
TELEPHONE	220 443 260
CELLPHONE	921234567



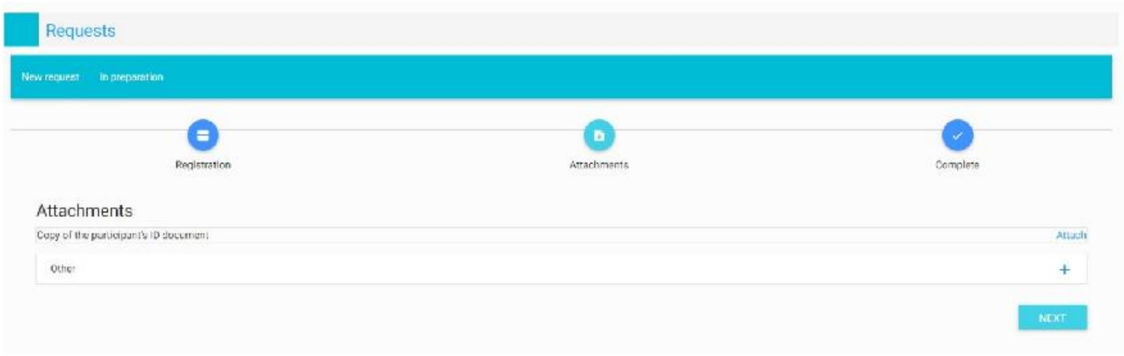
HEREBY REQUEST YOU TO BE GRANTED LICENSE FOR THE DAMAGE REPORT

ACTIVITY
Pipe repair

LOCAL
House in Porto

DURATION
5 hours

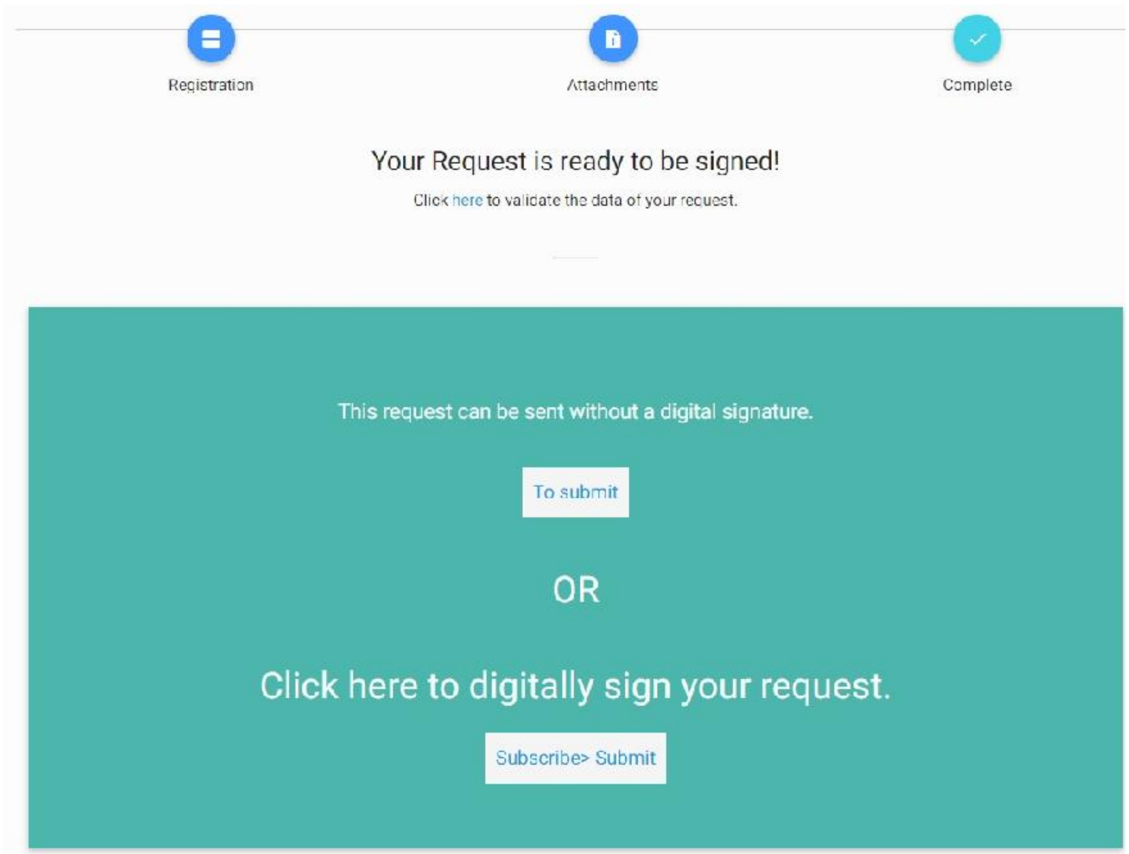
SCHEDULE
15/01/2018



Attachments

Copy of the participant's ID document

Other



Your Request is ready to be signed!

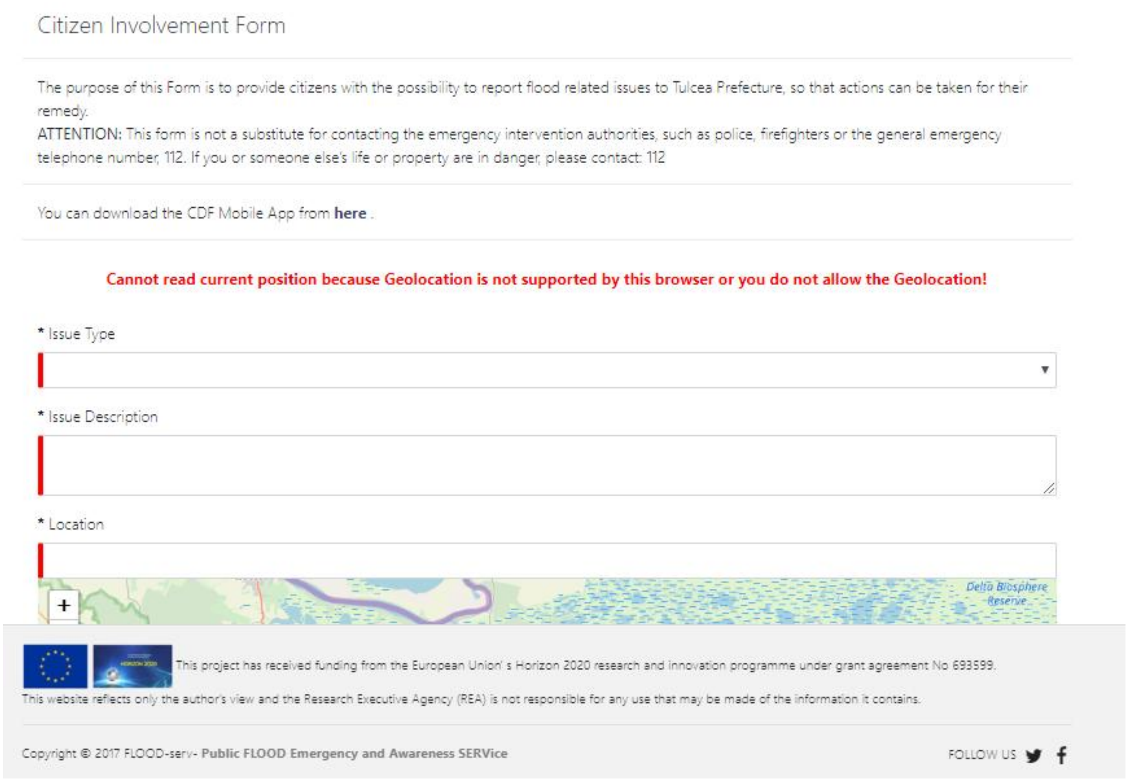
Click here to validate the data of your request.

This request can be sent without a digital signature.

OR

Click here to digitally sign your request.

For this test, a dummy UI was created using angular-js. This proved the corrected of the underline API that will be used by the Portal. The created issue will appear on the backoffice of the CDF for further processing.



Citizen Involvement Form

The purpose of this Form is to provide citizens with the possibility to report flood related issues to Tulcea Prefecture, so that actions can be taken for their remedy.
ATTENTION: This form is not a substitute for contacting the emergency intervention authorities, such as police, firefighters or the general emergency telephone number, 112. If you or someone else's life or property are in danger, please contact: 112

You can download the CDF Mobile App from **here** .

Cannot read current position because Geolocation is not supported by this browser or you do not allow the Geolocation!

* Issue Type

* Issue Description

* Location

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 693599.
This website reflects only the author's view and the Research Executive Agency (REA) is not responsible for any use that may be made of the information it contains.

Copyright © 2017 FLOOD-serv- **Public FLOOD Emergency and Awareness SERVice**

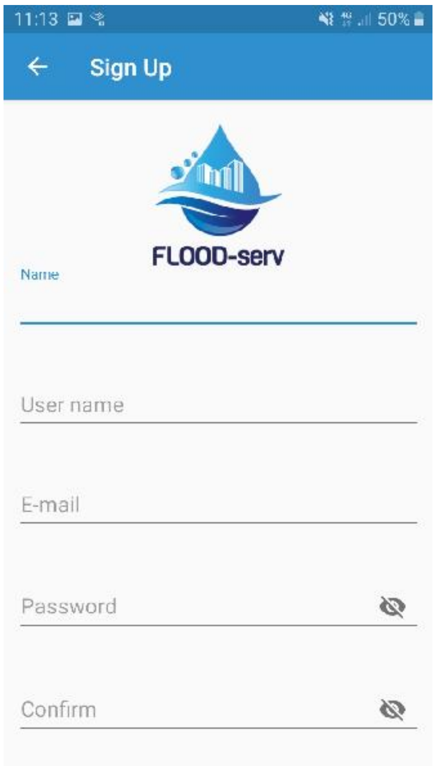
FOLLOW US

This is the implemented version on the portal.

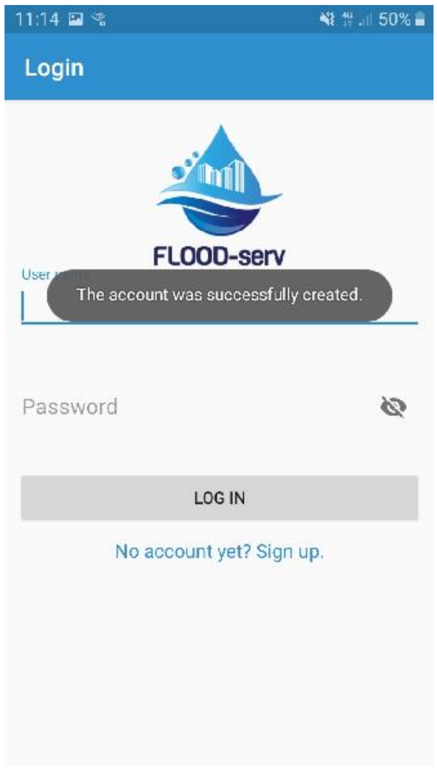
7.1.7 Test Case – Registration via mobile application

Filter alerts and communications	
Test Type:	Manual
Status:	Final
Preconditions:	⟩ Application installed for eCitizen
Steps to complete:	1. eCitizen accesses application 2. Clicks to register 3. Fills in registration form 4. Submits form 5. Receives confirmation
Expected Outcome:	3. eCitizen Registered and able to login

7.1.8 Results – Registration via mobile application



Form to register new user

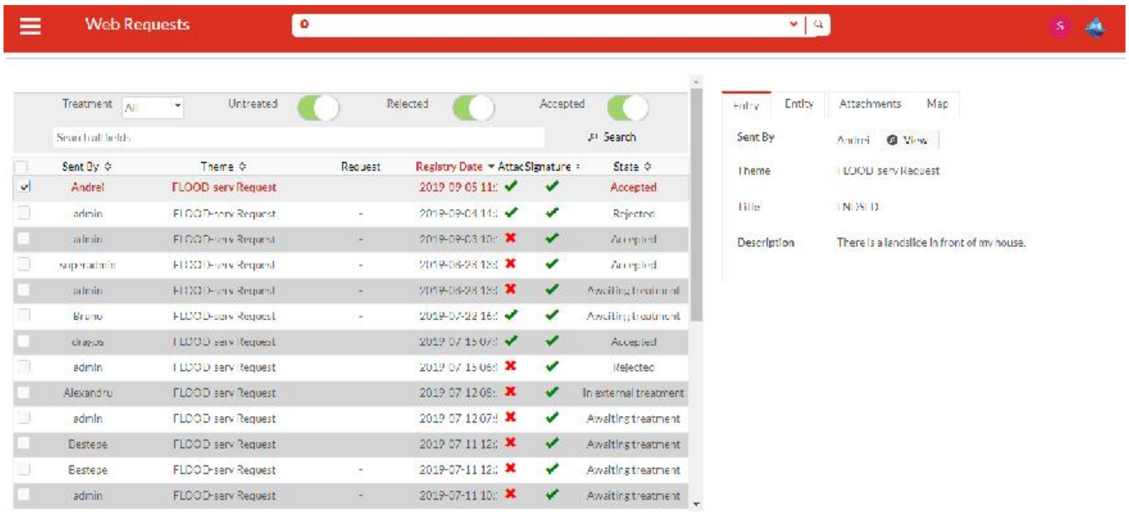


Account created successfully and user is able to login

7.1.9 Test Case – Filter alerts and communications

	Filter alerts and communications
Test Type:	Manual
Status:	Final
Preconditions:	) PRE 1 - Activated Facilitator (Platform operator)
Steps to complete:	6. Platform operator can filter citizen communications on a specific area; 7. Platform operator can search based on criteria; 8. Platform operator can search history of occurrences;
Expected Outcome:	4. The Platform operator is able to filter, accept or reject alerts reported by eCitizens

7.1.10 Results – Filter alerts and communications



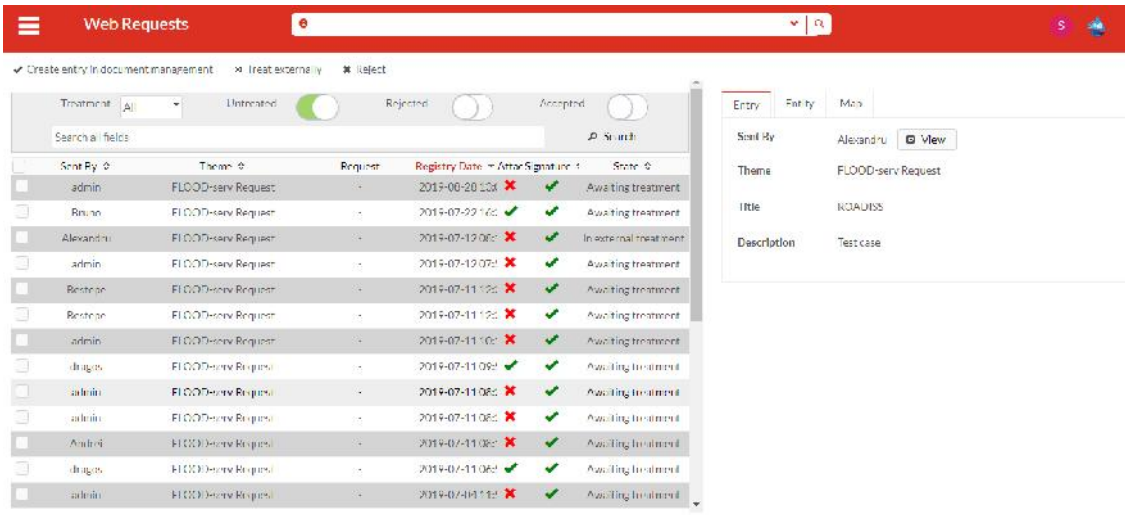
Treatment All | Untreated | Rejected | Accepted

Search all fields | Search

	Sent By	Theme	Request	Registry Date	Attac	Signature	State
✓	Andrei	FLOOD serv Request		2019 09 05 11:	✓	✓	Accepted
	admin	FLOOD-serv Request	-	2019-09-04 11:	✓	✓	Rejected
	admin	FLOOD-serv Request	-	2019-09-03 10:	✗	✓	Accepted
	superadmin	FLOOD-serv Request	-	2019-08-28 15:	✗	✓	Accepted
	admin	FLOOD-serv Request	-	2019-08-28 15:	✗	✓	Awaiting treatment
	Bruno	FLOOD-serv Request	-	2019-07-22 16:	✓	✓	Awaiting treatment
	dragos	FLOOD serv Request		2019 07 15 07:	✓	✓	Accepted
	admin	FLOOD serv Request		2019 07 15 06:	✗	✓	Rejected
	Alexandru	FLOOD serv Request		2019 07 12 08:	✗	✓	In external treatment
	admin	FLOOD serv Request		2019 07 12 07:	✗	✓	Awaiting treatment
	Bestepe	FLOOD serv Request		2019 07 11 12:	✗	✓	Awaiting treatment
	Bestepe	FLOOD-serv Request	-	2019-07-11 12:	✗	✓	Awaiting treatment
	admin	FLOOD-serv Request	-	2019-07-11 10:	✗	✓	Awaiting treatment

Sent By	Andrei
Theme	FLOOD serv Request
Title	INUNDI
Description	There is a landslide in front of my house.

The PO can access the dedicated area to process the received alerts



Treatment All | Untreated | Rejected | Accepted

Search all fields | Search

	Sent By	Theme	Request	Registry Date	Attac	Signature	State
	admin	FLOOD-serv Request	-	2019-08-20 13:	✗	✓	Awaiting treatment
	Bruno	FLOOD-serv Request	-	2019-07-22 16:	✓	✓	Awaiting treatment
	Alexandru	FLOOD-serv Request	-	2019-07-12 08:	✗	✓	In external treatment
	admin	FLOOD-serv Request	-	2019-07-12 07:	✗	✓	Awaiting treatment
	Bestepe	FLOOD-serv Request	-	2019-07-11 12:	✗	✓	Awaiting treatment
	Bestepe	FLOOD-serv Request	-	2019-07-11 12:	✗	✓	Awaiting treatment
	admin	FLOOD-serv Request	-	2019-07-11 10:	✗	✓	Awaiting treatment
	dragos	FLOOD-serv Request	-	2019-07-11 09:	✓	✓	Awaiting treatment
	admin	FLOOD-serv Request	-	2019-07-11 08:	✗	✓	Awaiting treatment
	admin	FLOOD-serv Request	-	2019-07-11 08:	✗	✓	Awaiting treatment
	Andrei	FLOOD-serv Request	-	2019-07-11 08:	✗	✓	Awaiting treatment
	dragos	FLOOD-serv Request	-	2019-07-11 06:	✓	✓	Awaiting treatment
	admin	FLOOD-serv Request	-	2019-07-04 11:	✗	✓	Awaiting treatment

Sent By	Alexandru
Theme	FLOOD-serv Request
Title	ROADISS
Description	Test case

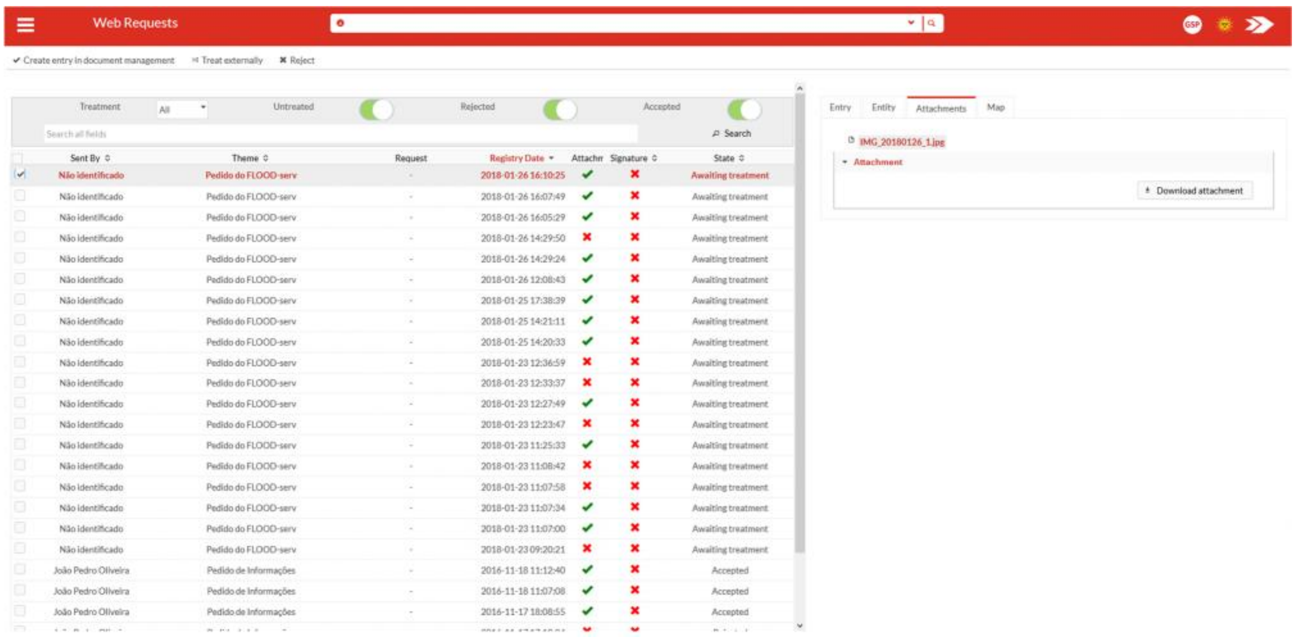
The user can the search fields and filter buttons to search for specific issues

7.1.11 Test Case – Process alert, communication or damage report

Process alert, communication or damage report	
Test Type:	Manual
Status:	Final
Preconditions:	PRE 1 - Activated Facilitator (Platform operator)
Steps to complete:	1. After validating the details, the Platform operator can approve the alert for further approval within the organization – click create entry 2. Will be redirected automatically to the ticket register area and the ticked will be registered automatically

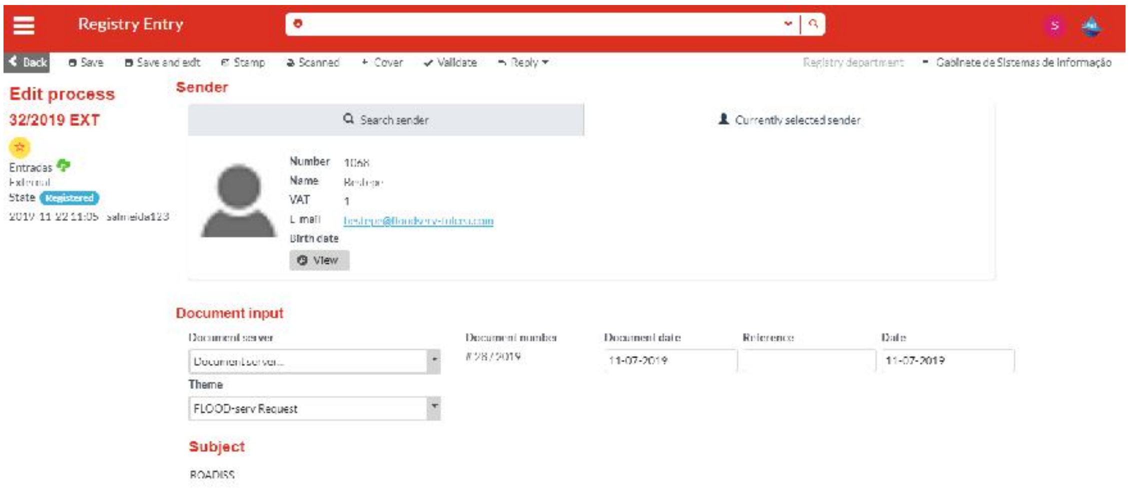
	3. Can provide more information on the ticked 4. Can then forward the ticked to another department/user for further processing 5. It will appear on the todo lisk, to where the ticked was sent 6. Can then be forward, following a free flow workflow 7. More information can be added and more documents attached 8. Platform operator can give feedback or official position on the ticket 9. The ticket can at any point be archived 10. Platform operator can also reject the alert
Expected Outcome:	1. The Platform operator is send an alert for approval respecting and workflow and receive feedback from other departments 2. The citizen

7.1.12 Results – Process alert, communication or damage report



Sent By	Theme	Request	Registry Date	Attachm	Signature	State
Não identificado	Pedido do FLOOD-serv	-	2018-01-26 16:10:25	✓	✗	Awaiting treatment
Não identificado	Pedido do FLOOD-serv	-	2018-01-26 16:07:49	✓	✗	Awaiting treatment
Não identificado	Pedido do FLOOD-serv	-	2018-01-26 16:05:29	✓	✗	Awaiting treatment
Não identificado	Pedido do FLOOD-serv	-	2018-01-26 14:29:50	✗	✗	Awaiting treatment
Não identificado	Pedido do FLOOD-serv	-	2018-01-26 14:29:24	✓	✗	Awaiting treatment
Não identificado	Pedido do FLOOD-serv	-	2018-01-26 12:08:43	✓	✗	Awaiting treatment
Não identificado	Pedido do FLOOD-serv	-	2018-01-25 17:38:39	✓	✗	Awaiting treatment
Não identificado	Pedido do FLOOD-serv	-	2018-01-25 14:21:11	✓	✗	Awaiting treatment
Não identificado	Pedido do FLOOD-serv	-	2018-01-25 14:20:33	✓	✗	Awaiting treatment
Não identificado	Pedido do FLOOD-serv	-	2018-01-23 12:36:59	✗	✗	Awaiting treatment
Não identificado	Pedido do FLOOD-serv	-	2018-01-23 12:33:37	✗	✗	Awaiting treatment
Não identificado	Pedido do FLOOD-serv	-	2018-01-23 12:27:49	✓	✗	Awaiting treatment
Não identificado	Pedido do FLOOD-serv	-	2018-01-23 12:23:47	✗	✗	Awaiting treatment
Não identificado	Pedido do FLOOD-serv	-	2018-01-23 11:25:33	✓	✗	Awaiting treatment
Não identificado	Pedido do FLOOD-serv	-	2018-01-23 11:08:42	✗	✗	Awaiting treatment
Não identificado	Pedido do FLOOD-serv	-	2018-01-23 11:07:58	✗	✗	Awaiting treatment
Não identificado	Pedido do FLOOD-serv	-	2018-01-23 11:07:34	✓	✗	Awaiting treatment
Não identificado	Pedido do FLOOD-serv	-	2018-01-23 11:07:00	✓	✗	Awaiting treatment
Não identificado	Pedido do FLOOD-serv	-	2018-01-23 09:20:21	✗	✗	Awaiting treatment
João Pedro Oliveira	Pedido de Informações	-	2016-11-18 11:12:40	✓	✗	Accepted
João Pedro Oliveira	Pedido de Informações	-	2016-11-18 11:07:08	✓	✗	Accepted
João Pedro Oliveira	Pedido de Informações	-	2016-11-17 18:08:55	✓	✗	Accepted

Click create entry to approve and register the ticket.



Edit process

32/2019 EXT

Entradas
External
State Registered
2019-11-22 11:05 salmeida123

Sender

Number 1068
Name [illegible]
VAT 1
E-mail [illegible]
Birth date

Document input

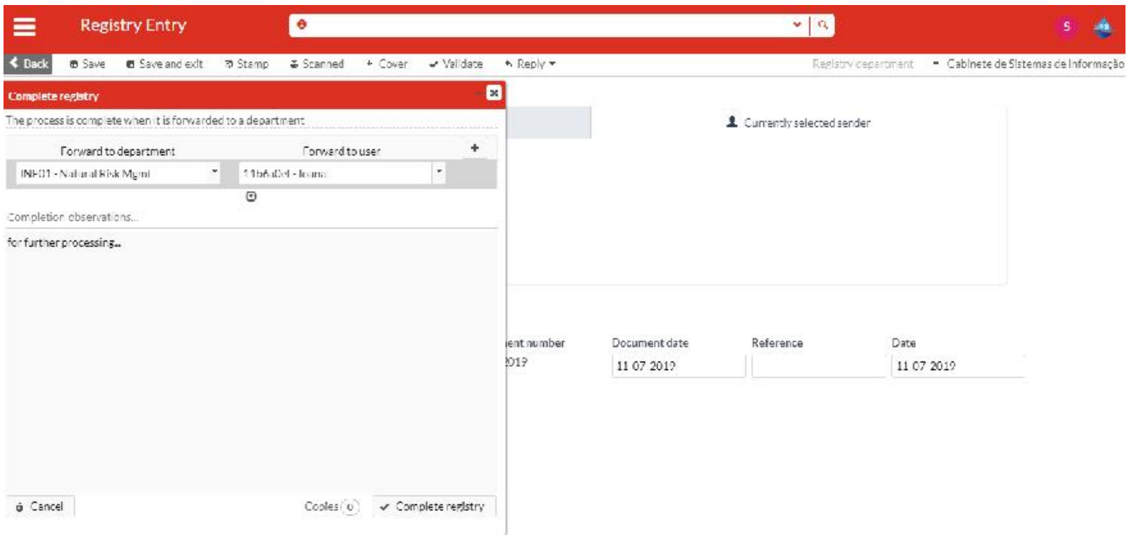
Document server	Document number	Document date	Reference	Date
Document server...	#28/2019	11-07-2019		11-07-2019

Theme: FLOOD-serv Request

Subject

ROADISS

All the information from the issue will automatically flow to the ticket, including the attachments. A unique number is assigned. Further information and attachments can be added.



Complete registry

The process is complete when it is forwarded to a department

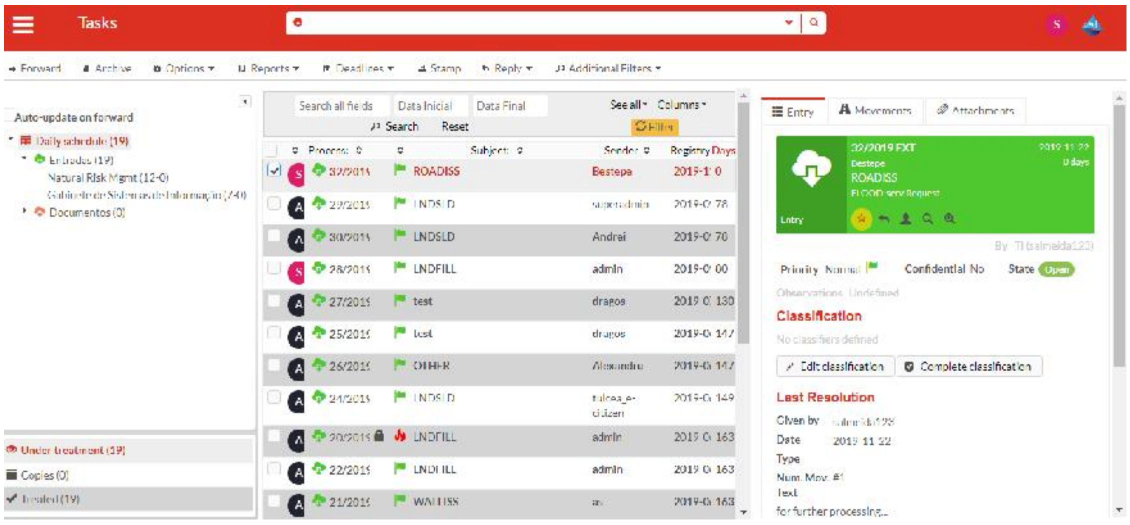
Forward to department	Forward to user
INF01 - Natural Risk Mgmt	116A054 - Ioana

Completion observations...

for further processing...

ent number	Document date	Reference	Date
019	11 07 2019		11 07 2019

After, the ticket can be forwarded to another department/user for further analysis.



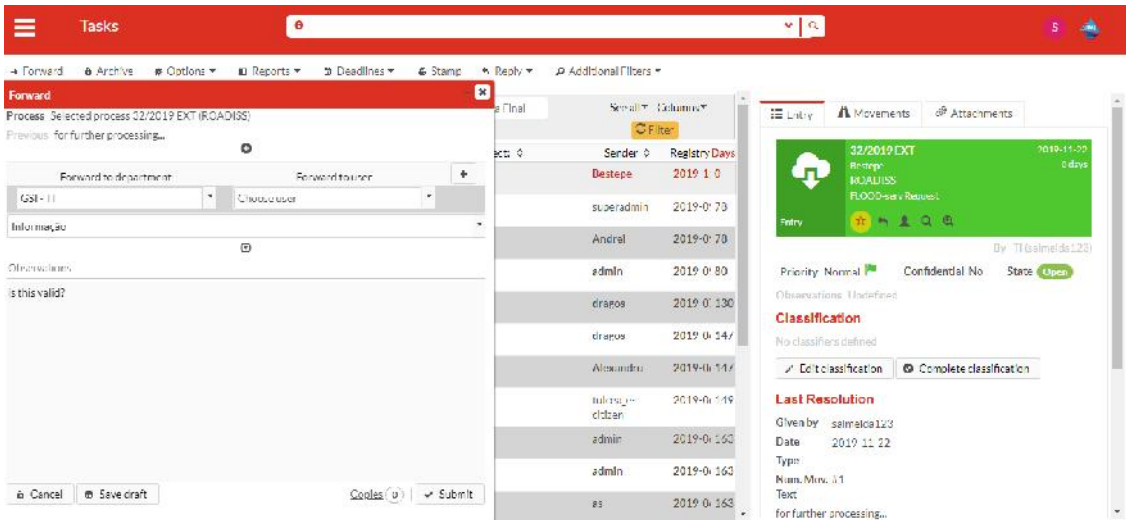
Auto-update on forward
- Daily schedule (19)
  - Entradas (19)
    - Natural Risk Mgmt (12-0)
    - Gabinete de Sistemas de Informação (7-0)
  - Documentos (0)

Under treatment (19)
Copies (0)
Treated (19)

Process		Subject	Sender	Registry Days
32/2019		ROADISS	Bestepe	2019-1 0
29/2019		LNDSLD	superadmin	2019-0 78
30/2019		LNDSLD	Andrei	2019-0 70
28/2019		LNDFILL	admin	2019-0 80
27/2019		test	dragos	2019-0 130
25/2019		test	dragos	2019-0 147
26/2019		OTHER	Alexandru	2019-0 147
24/2019		LNDSLD	tulcea_e-citizen	2019-0 149
20/2019		LNDFILL	admin	2019-0 163
22/2019		LNDFILL	admin	2019-0 163
21/2019		WATERS	as	2019-0 163

32/2019 EXT
Bestepe
ROADISS
FLOOD-serv.Request
2019-11-22
0 days
Entry
By TI (salmeida123)
Priority Normal
Confidential No
State Open
Observations Undefined
Classification
No classifiers defined
Last Resolution
Given by salmeida123
Date 2019-11-22
Type
Num. Mov. #1
Text
for further processing...

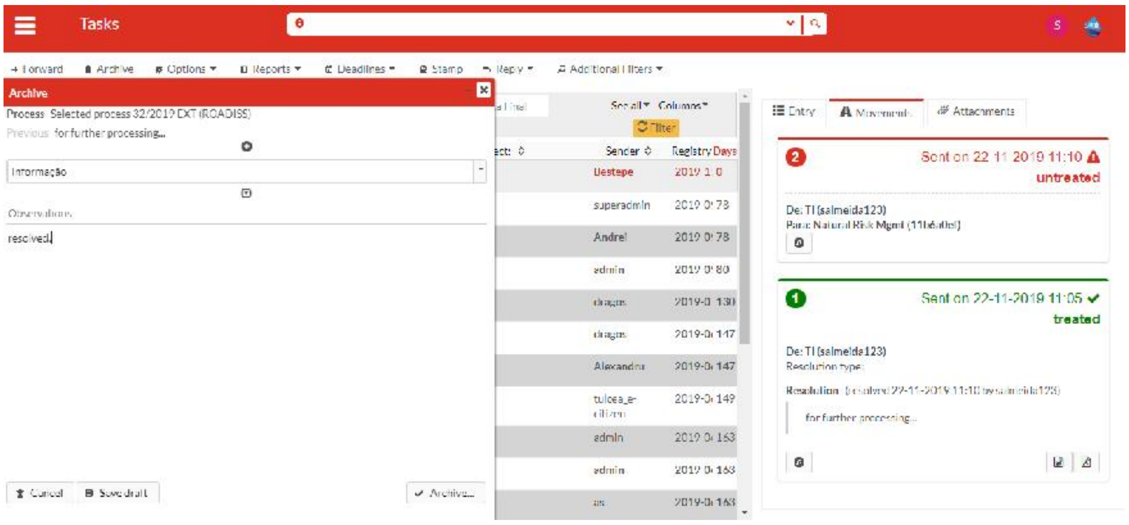
It will appear on the todo lisk, to where the ticked was sent



Forward
Process Selected process 32/2019 EXT (ROADISS)
Previous for further processing...
Forward to department: GSI - TI
Forward to user: Choose user
Informação
Observations
is this valid?

32/2019 EXT
Bestepe
ROADISS
FLOOD-serv.Request
2019-11-22
0 days
Entry
By TI (salmeida123)
Priority Normal
Confidential No
State Open
Observations Undefined
Classification
No classifiers defined
Last Resolution
Given by salmeida123
Date 2019-11-22
Type
Num. Mov. #1
Text
for further processing...

Can then be forward, following a free flow workflow. More information can be added and more documents attached. Platform operator can give feedback or official position on the ticket



Archive

Process: Selected process 32/2019 EXT (ROADISS)

Previous: for further processing...

Informação

Observations

resolved.

Sender	Registry	Days
Destepe	2019 1	0
superadmin	2019 0	78
Andrei	2019 0	78
admin	2019 0	80
dragos	2019-0	130
dragos	2019-0	147
Alexandru	2019-0	147
tulcea_e-citizen	2019-0	149
admin	2019 0	163
admin	2019 0	163
as	2019-0	163

2 Sent on 22 11 2019 11:10 untreated

De: TI (salmeida123)
Para: Natural Risk Mgmt (11b6a0el)

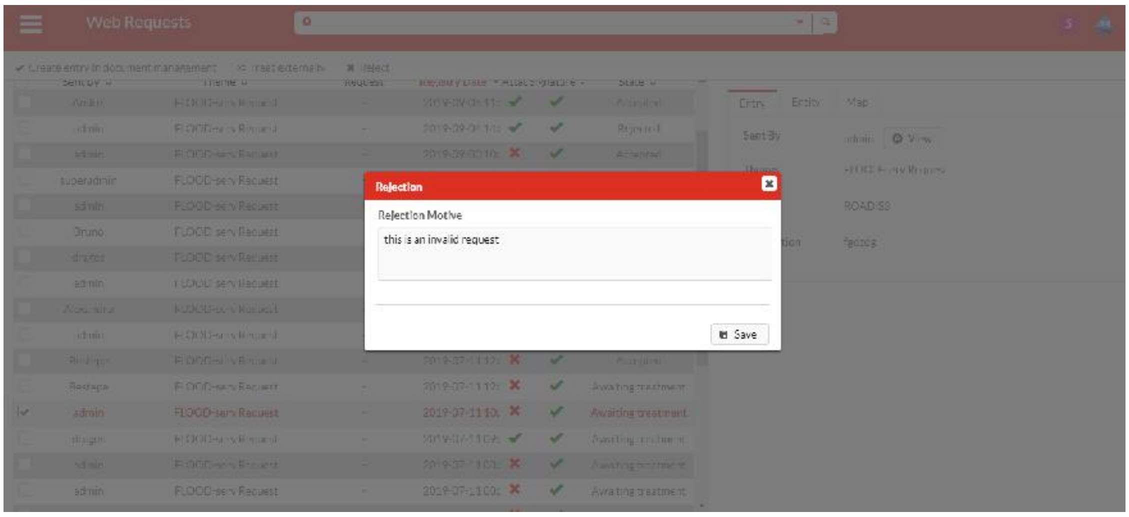
1 Sent on 22-11-2019 11:05 treated

De: TI (salmeida123)
Resolution type:

Resolution (resolved 22-11-2019 11:10 by salmeida123)

for further processing...

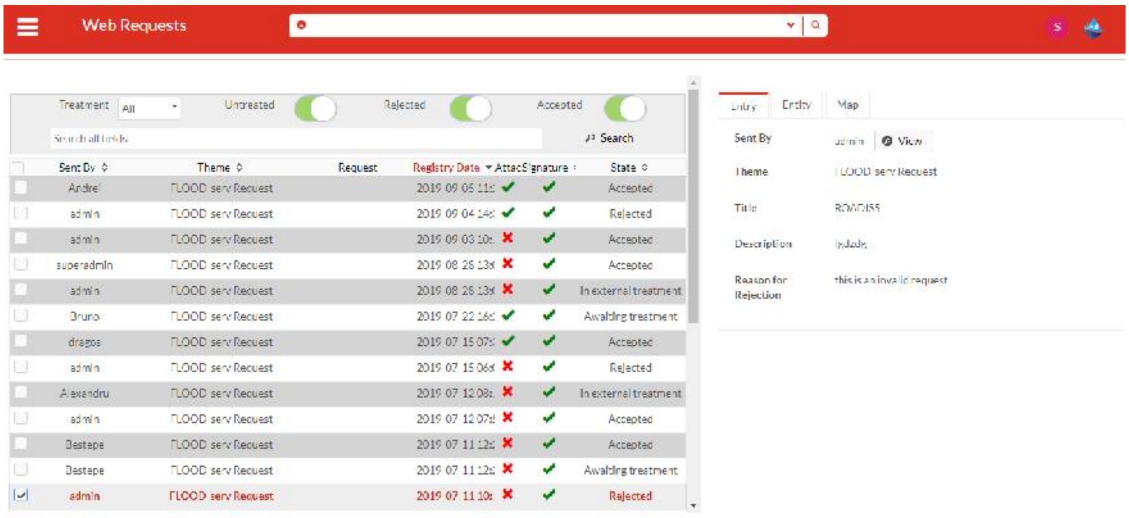
The ticket can at any point be archived.



Rejection

Rejection Motive

this is an invalid request



Treatment: All | Untreated | Rejected | Accepted

Search all fields

Sent By	Theme	Request	Registry Date	Attac	Signature	State
Andrei	FLOOD serv Request		2019 09 05 11:	✓	✓	Accepted
admin	FLOOD serv Request		2019 09 04 14:	✓	✓	Rejected
admin	FLOOD serv Request		2019 09 03 10:	✗	✓	Accepted
superadmin	FLOOD serv Request		2019 08 28 13:	✗	✓	Accepted
admin	FLOOD serv Request		2019 08 28 13:	✗	✓	In external treatment
Bruno	FLOOD serv Request		2019 07 22 16:	✓	✓	Awaiting treatment
dragos	FLOOD serv Request		2019 07 15 07:	✓	✓	Accepted
admin	FLOOD serv Request		2019 07 15 06:	✗	✓	Rejected
Alexandru	FLOOD serv Request		2019 07 12 08:	✗	✓	In external treatment
admin	FLOOD serv Request		2019 07 12 07:	✗	✓	Accepted
Destepe	FLOOD serv Request		2019 07 11 12:	✗	✓	Accepted
Destepe	FLOOD serv Request		2019 07 11 12:	✗	✓	Awaiting treatment
admin	FLOOD serv Request		2019 07 11 10:	✗	✓	Rejected

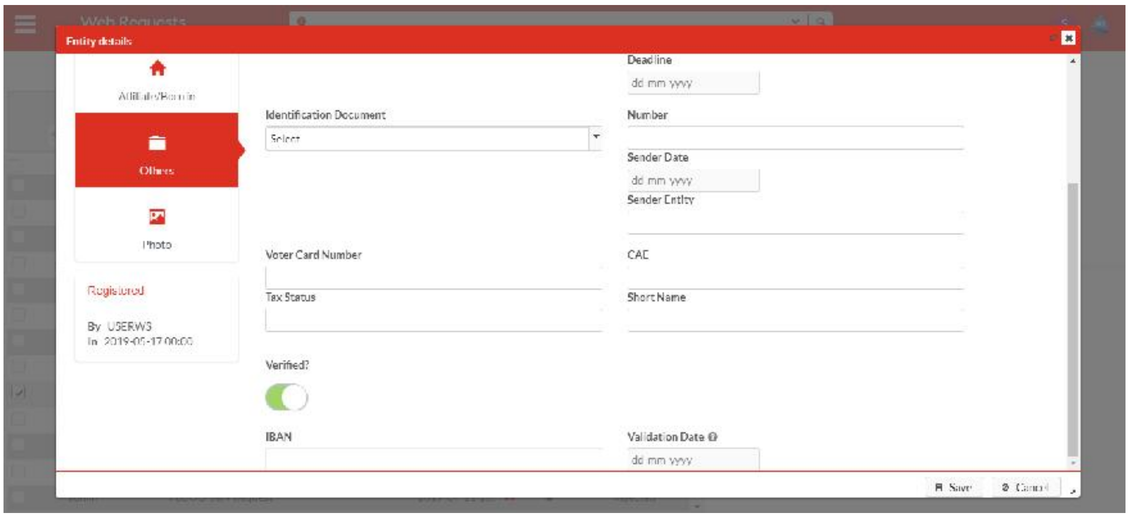
Sent By	admin
Theme	FLOOD serv Request
Title	ROADISS
Description	lydola
Reason for Rejection	this is an invalid request

Platform operator can also reject the alert

7.1.13 Test Case – Certification of citizen

Certification of citizen	
Test Type:	Manual
Status:	Final
Preconditions:	• PRE 1 - Activated Facilitator (Platform operator) • PRE 5 – Activated eCitizen • eCitizen has mobile application installed
Steps to complete:	1. Platform operator will check the profile of the eCitizen – (via Base Data one the main menu or on the entity tab within the web requests screen – where the alerts are processed) 2. Will check the user as certified observer 3. The citizen will then be granted all the privileges associated with the role Certified Observer.
Expected Outcome:	1. The Platform operator is able to check and certify eCitizen registered user

7.1.14 Results – Certification of citizen



Registered

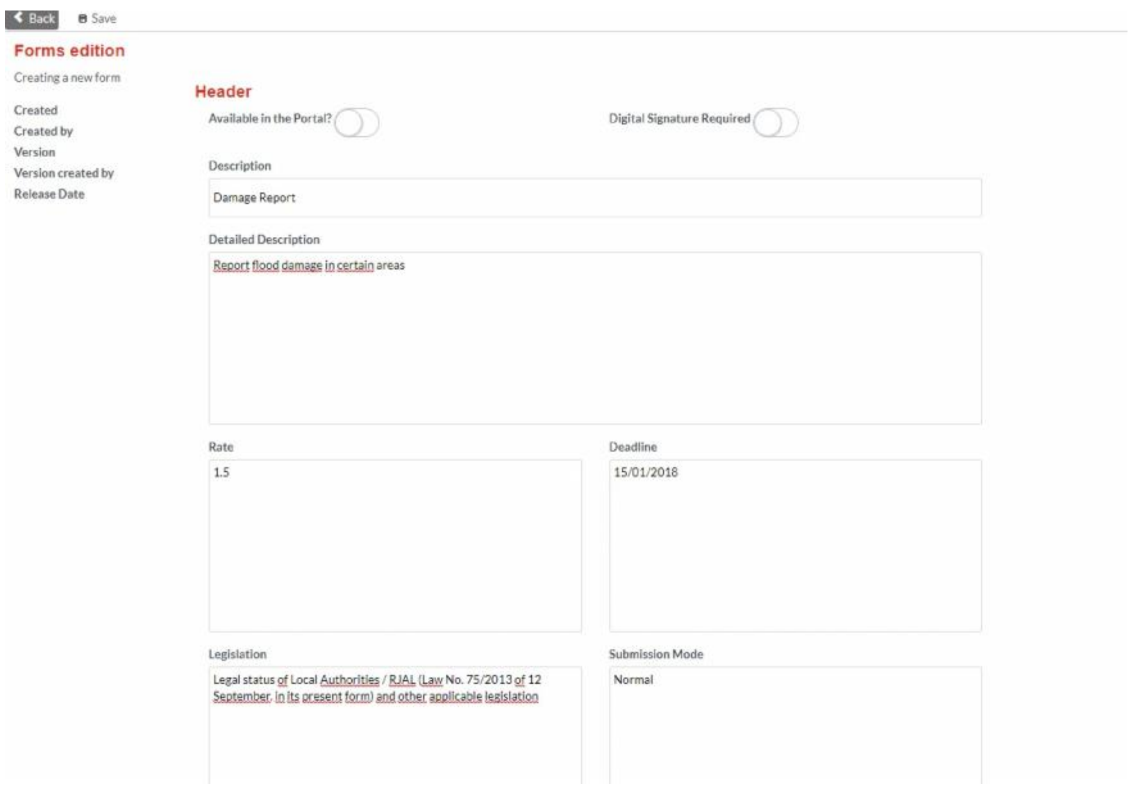
By USERW3
In 2019-05-17 00:00

	Deadline: dd mm yyyy
Identification Document: Select	Number
	Sender Date: dd mm yyyy
	Sender Entity
Voter Card Number	CAE
Tax Status	Short Name
Verified?	
IBAN	Validation Date: dd mm yyyy

7.1.15 Test Case – Define Smart Form

Define Smart Form	
Test Type:	Manual
Status:	Final
Preconditions:	PRE 1 - Activated Facilitator (Platform operator)
Steps to complete:	1. Platform Operator can define the theme associated – Access via main menu "online Service" » "Base Data" » "Forms" 2. Platform Operator can define the basic info, sections and fields needed for the form 3. Platform Operator can define attachments needed to the form 4. Platform Operator can publish the form 5. Platform Operator can alter the details of the Form
Expected Outcome:	1. The Platform operator is able to define, alter, delete and publish smart form

7.1.16 Results – Define Smart Form



Available in the Portal?

Digital Signature Required

Description: Damage Report

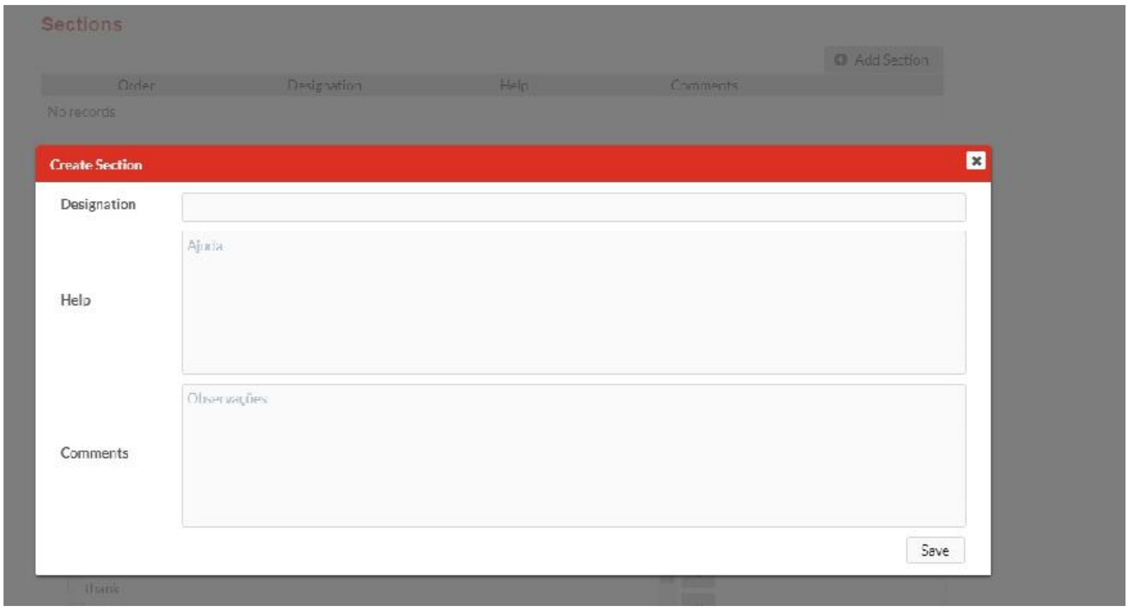
Detailed Description: Report flood damage in certain areas

Rate: 1.5

Deadline: 15/01/2018

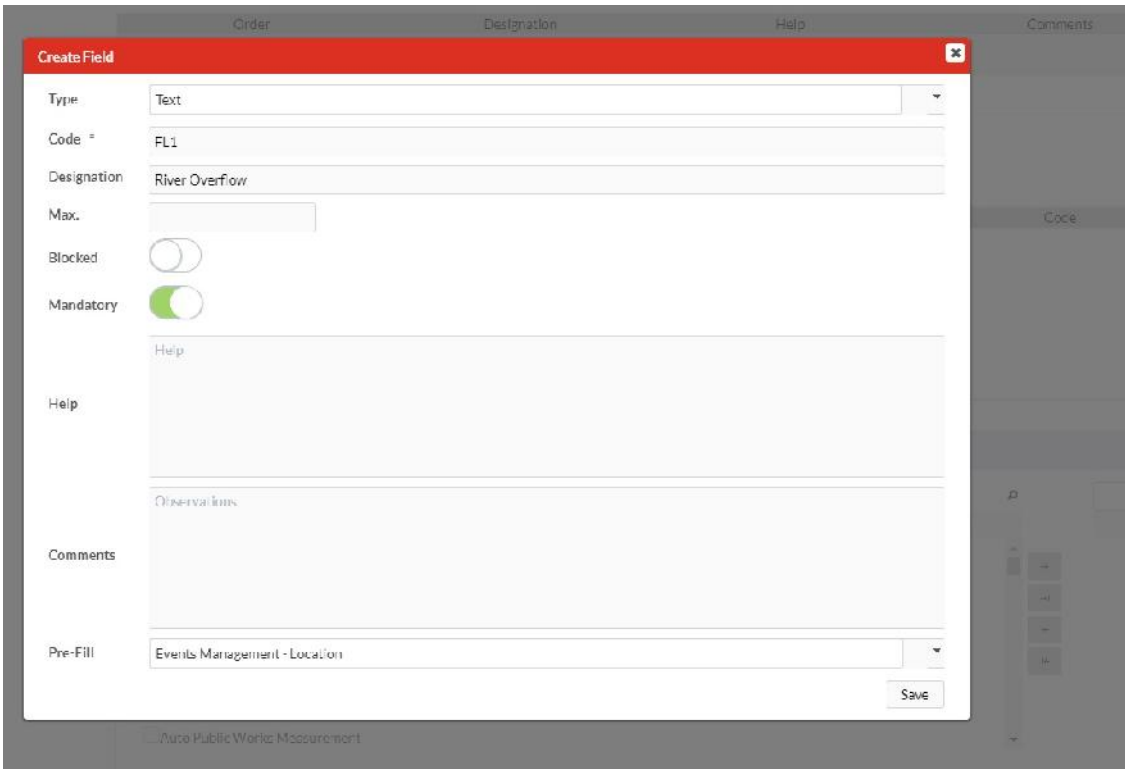
Legislation: Legal status of Local Authorities / RJAL (Law No. 75/2013 of 12 September, in its present form) and other applicable legislation

Submission Mode: Normal



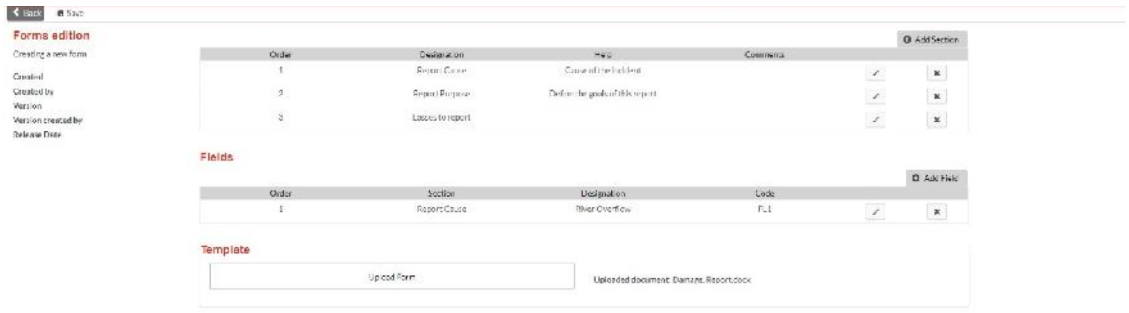
Create Section

Designation	
Help	Ajuda
Comments	Observações



Create Field

Type	Text
Code *	FL1
Designation	River Overflow
Max.	
Blocked	Off
Mandatory	On
Help	Help
Comments	Observations
Pre-Fill	Events Management - Location



Forms edition

Creating a new form

Created
Created by
Version
Version created by
Release Date

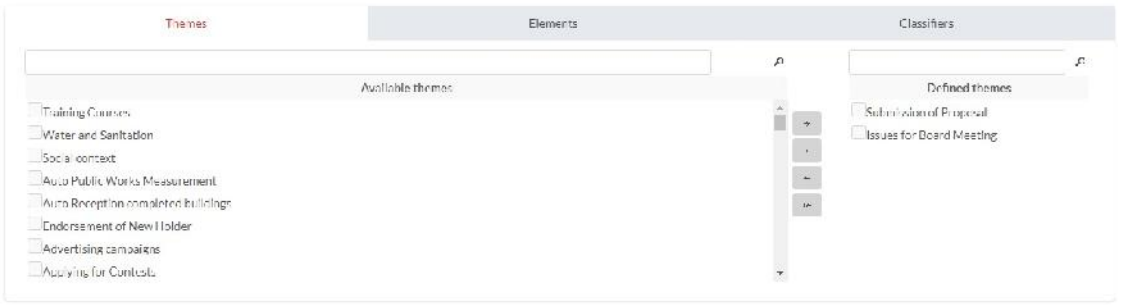
Order	Designation	Help	Comments
1	Report Cause	Cause of the incident	
2	Report Purpose	Define the goals of this report	
3	Losses to report		

Fields

Order	Section	Designation	Code
1	Report Cause	River Overflow	FL1

Template

Uploaded document: Damage_Report.docx

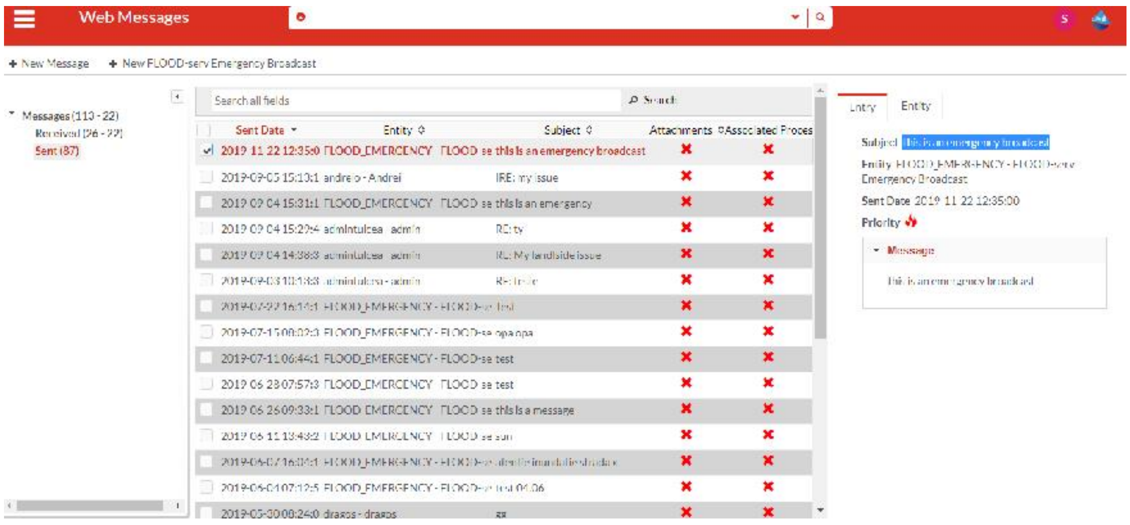


Creating and publishing a smart, will allow to it be accessible via API, and thus can be shown and used in the portal. This flexibility allows the public authorities to implement more agile or formal processes within the platform.

7.1.17 Test Case – Send Broadcast emergency

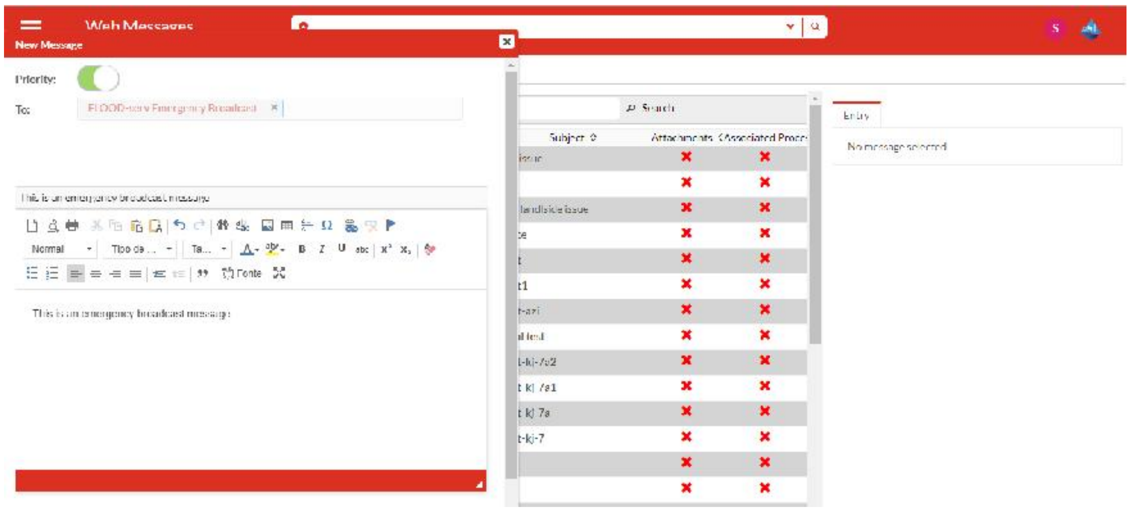
Send Broadcast emergency	
Test Type:	Manual
Status:	Final
Preconditions:	- PRE 1 - Activated Facilitator (Platform operator) - PRE 5 – Activated eCitizen - eCitizen has mobile application installed
Steps to complete:	1. Platform Operator can define a new broadcast (sms or message) message. Main Menu: “Online Service” » “Web Messages” 2. Can send a broadcast message to eCitizens and emergency responders 3. eCitizens will receive a push notification and a message in the FLOOD-Serv app. 4. As an alternative, the message can be sent via SMS. 5. The PO can also send a message to a particular eCitizen. This
Expected Outcome:	1. The Platform operator is able send messages and broadcast messages 2. The eCitizens receive the message via SMS, portal or app

7.1.18 Results – Send Broadcast emergency



	Sent Date	Entity	Subject	Attachments	Associated Proces
✓	2019 11 22 12:35:0	FLOOD_EMERGENCY	FLOOD se this is an emergency broadcast	✗	✗
	2019-09-05 15:13:1	andreio - Andrei	RE: my issue	✗	✗
	2019 09 04 15:31:1	FLOOD_EMERGENCY	FLOOD se this is an emergency	✗	✗
	2019 09 04 15:29:4	adminulcea admin	RE: ty	✗	✗
	2019 09 04 14:38:3	adminulcea admin	RE: My landslide issue	✗	✗
	2019-09-03 10:18:3	adminulcea - admin	RE: [illegible]	✗	✗
	2019-07-22 16:14:1	FLOOD_EMERGENCY - FLOOD-se test		✗	✗
	2019-07-15 08:02:3	FLOOD_EMERGENCY - FLOOD-se opa opa		✗	✗
	2019-07-11 06:44:1	FLOOD_EMERGENCY - FLOOD-se test		✗	✗
	2019 06 28 07:57:3	FLOOD_EMERGENCY	FLOOD se test	✗	✗
	2019 06 26 09:33:1	FLOOD EMERGENCY	FLOOD se this is a message	✗	✗
	2019 06 11 13:43:2	FLOOD EMERGENCY	FLOOD se [illegible]	✗	✗
	2019-06-07 16:04:1	FLOOD_EMERGENCY - FLOOD-se [illegible]		✗	✗
	2019-06-04 07:12:5	FLOOD_EMERGENCY - FLOOD-se test 04.06		✗	✗
	2019-05-30 08:24:0	dragos - dragos	zz	✗	✗

Subject: this is an emergency broadcast
Entity: FLOOD_EMERGENCY - FLOOD-serv Emergency Broadcast
Sent Date: 2019 11 22 12:35:00
Priority
Message
This is an emergency broadcast

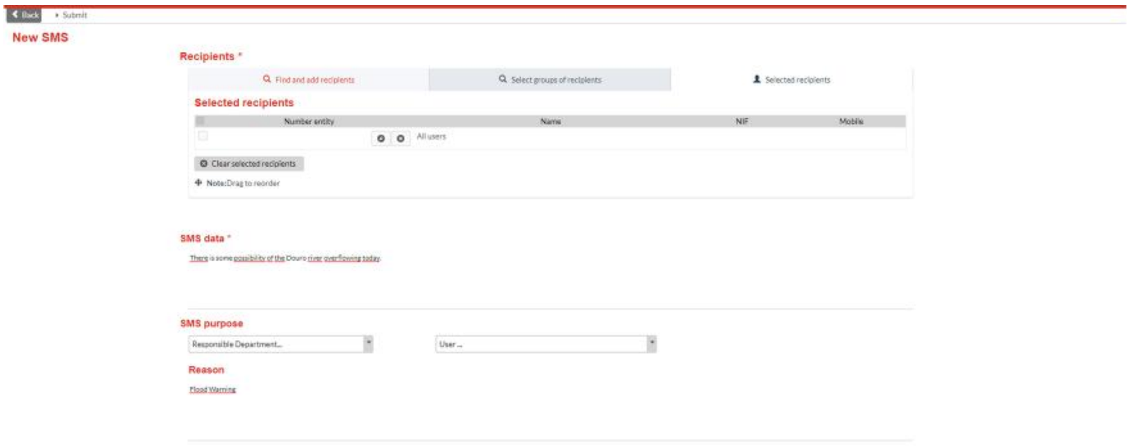


Priority:
To: FLOOD-serv Emergency Broadcast
This is an emergency broadcast message
This is an emergency broadcast message

Subject	Attachments	Associated Proces
issue	✗	✗
	✗	✗
landslide issue	✗	✗
te	✗	✗
t	✗	✗
t1	✗	✗
t-azi	✗	✗
al test	✗	✗
t-kj-7a2	✗	✗
t kj 7a1	✗	✗
t kj 7a	✗	✗
t-kj-7	✗	✗
	✗	✗
	✗	✗

No message selected

The purpose of this feature is to have a simple and agile broadcast message sent to every citizen



New SMS

Recipients *

Selected recipients

	Number entity	Name	NIF	Mobile
		All users		

Note: Drag to reorder

SMS data *

There is some possibility of the Douro river overflowing today.

SMS purpose

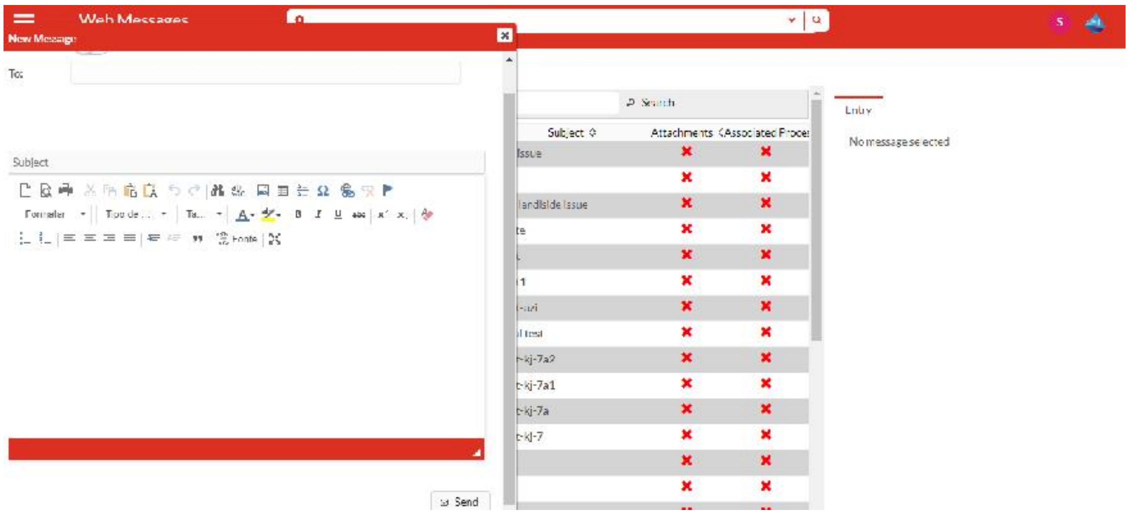
Responsible Department...

User ...

Reason

Flood Warning

As an alternative, the message can be sent via SMS.



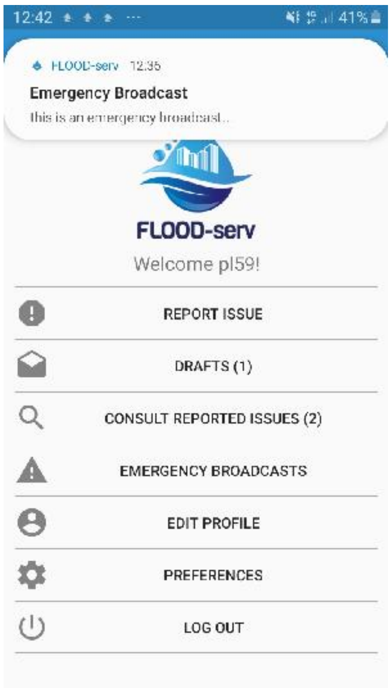
Subject	Attachments	Associated Proce
Issue	✗	✗
	✗	✗
landslide Issue	✗	✗
te	✗	✗
l	✗	✗
1	✗	✗
-uzi	✗	✗
l test	✗	✗
-kj-7a2	✗	✗
-kj-7a1	✗	✗
-kj-7a	✗	✗
-kj-7	✗	✗
	✗	✗
	✗	✗

The purpose of this feature is to send a message to a particular citizen, thus allowing another channel for a two-way communication. In the portal the citizen can send messages to the public authority.

7.1.19 Test Case – Receive Broadcast emergency

Receive Broadcast emergency	
Test Type:	Manual
Status:	Final
Preconditions:	- PRE 5 – Activated eCitizen - eCitizen has mobile application installed
Steps to complete:	1. PO Sends broadcast message
Expected Outcome:	1. eCitizen receives message on mobile app and is notified via push notification

7.1.20 Receive Broadcast emergency

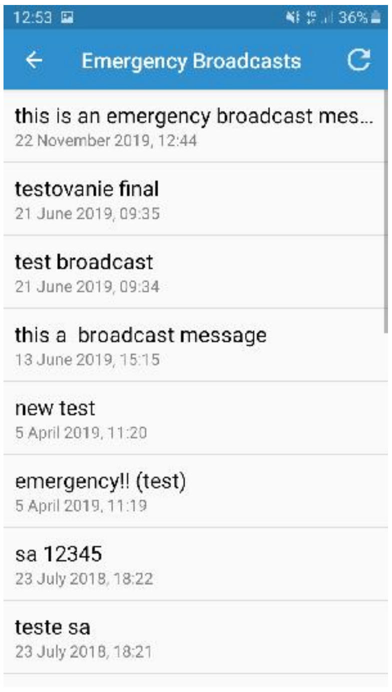

12:42 41%

FLOOD-serv 12:35

Emergency Broadcast

this is an emergency broadcast...

The Citizen receives a push notification



this is an emergency broadcast mes...
22 November 2019, 12:44

testovanie final
21 June 2019, 09:35

test broadcast
21 June 2019, 09:34

this a broadcast message
13 June 2019, 15:15

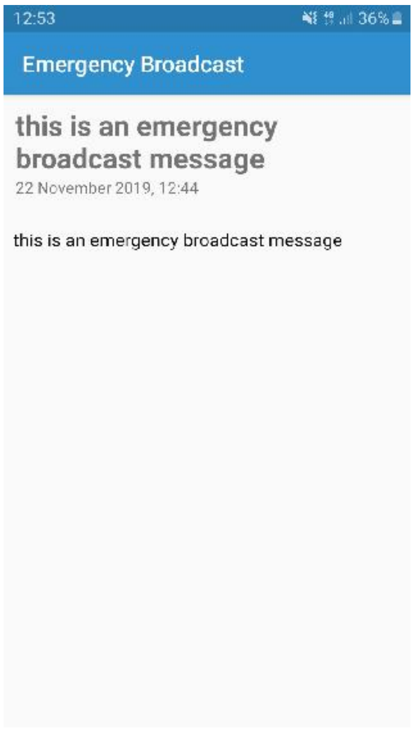
new test
5 April 2019, 11:20

emergency!! (test)
5 April 2019, 11:19

sa 12345
23 July 2018, 18:22

teste sa
23 July 2018, 18:21

Can check the history of messages


12:53 36%

this is an emergency broadcast message
22 November 2019, 12:44

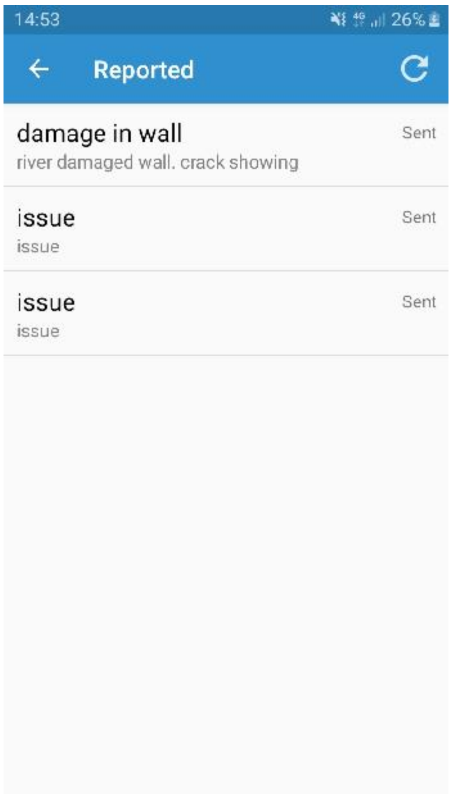
this is an emergency broadcast message

Clicking in the notification or via the broadcast messages area in the app, we can consult the details of the message.

7.1.21 Test Case – Check alert or communcation status

Check alert or communcation status	
Test Type:	Manual
Status:	Final
Preconditions:	- PRE 5 – Activated eCitizen - eCitizen has mobile application installed
Steps to complete:	1. eCitizen can check the status and actions on a specific matter; 2. Can search based on criteria for issues, in the portal; 3. Can get details on those actions.
Expected Outcome:	1. eCitizen can check status of reported issue or report

7.1.22 Results – Check alert or communcation status



14:53 26%

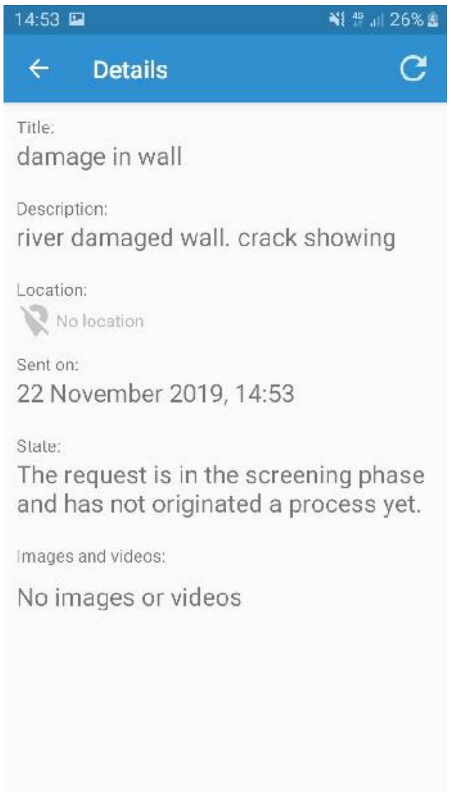
Reported

damage in wall
river damaged wall. crack showing
Sent

issue
issue
Sent

issue
issue
Sent

List of reported issues in the app



14:53 26%

Details

Title:
damage in wall

Description:
river damaged wall. crack showing

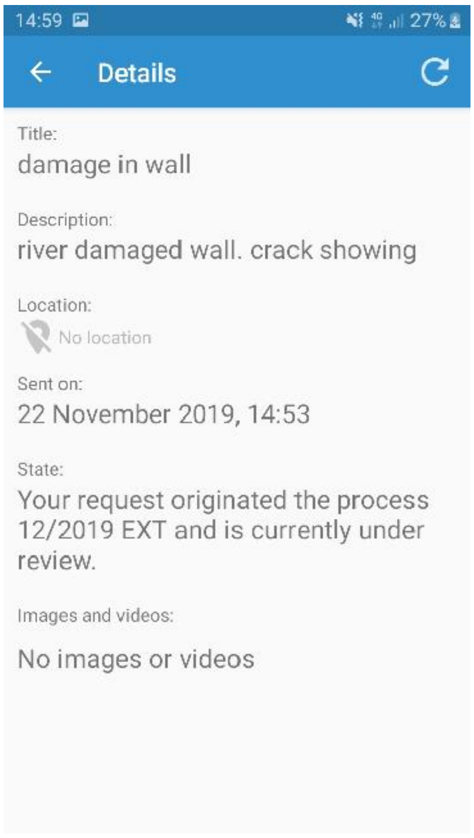
Location:
No location

Sent on:
22 November 2019, 14:53

State:
The request is in the screening phase and has not originated a process yet.

Images and videos:
No images or videos

Detail of reported issue with no feedback



Title:
damage in wall

Description:
river damaged wall. crack showing

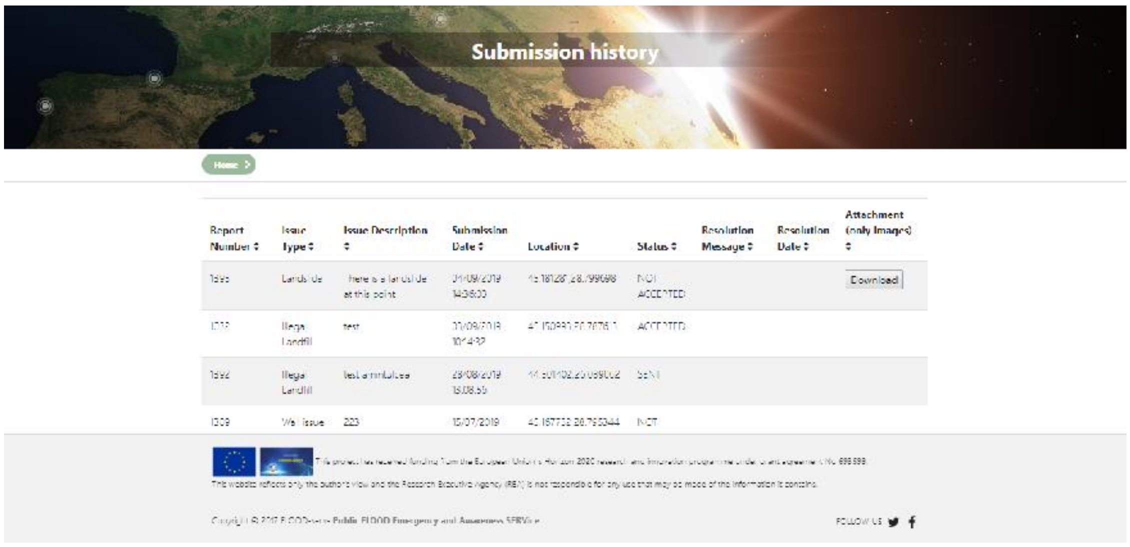
Location:
No location

Sent on:
22 November 2019, 14:53

State:
Your request originated the process 12/2019 EXT and is currently under review.

Images and videos:
No images or videos

Detail of reported issue with approved



Report Number ⇕	Issue Type ⇕	Issue Description ⇕	Submission Date ⇕	Location ⇕	Status ⇕	Resolution Message ⇕	Resolution Date ⇕	Attachment (only Images) ⇕
1395	Landslide	There is a landslide at this point	01/09/2019 14:36:00	45.18128,28.799698	NOT ACCEPTED			Download
1332	Illegal Landfill	test	30/09/2019 10:14:32	41.150993,20.787611	ACCEPTED			
1392	Illegal Landfill	test ambulance	28/08/2019 15:08:55	44.501402,26.089002	SENT			
1309	Wall issue	223	15/07/2019	40.167732,26.795344	NOT			

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant agreement No 699.598.
This website reflects only the author's view and the Research Executive Agency (REA) is not responsible for any use that may be made of the information it contains.

Copyright © 2017 FLOODserv - Public FLOOD Emergency and Awareness SERVice

FOLLOW US

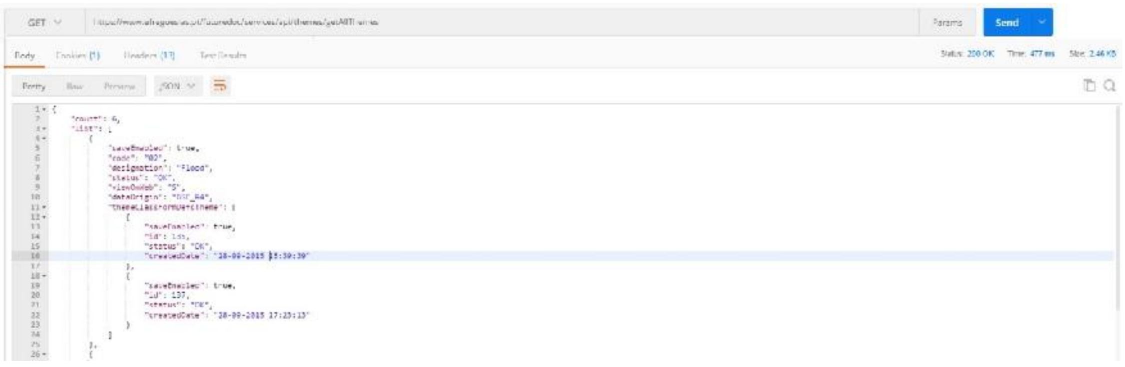
List of reported issues in the portal

7.1.23 Test Case – Call WS Rest API

	Call WS Rest API
Test Type:	Automated/Manual
Status:	Template
Preconditions:	- PRE 2 - Activated Facilitator - API Available
Steps to complete:	1. Open Browser 2. Call WS URL
Expected outcome:	1. JSON file with response

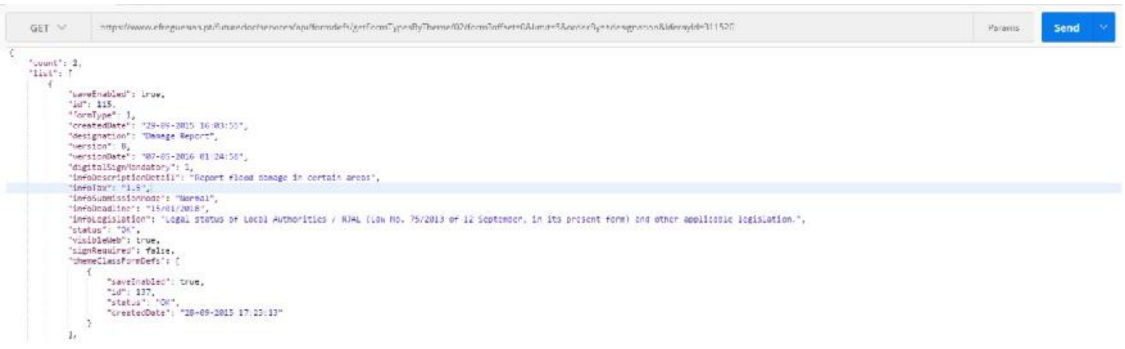
7.1.24 Results – Call WS Rest API

Get All Themes



```
{
    "count": 6,
    "list": [
        {
            "saveEnabled": true,
            "code": "02",
            "designation": "Flood",
            "status": "OK",
            "viewOnWeb": "S",
            "dataOrigin": "DST_R4",
            "themeClassFormDefsTheme": [
                {
                    "saveEnabled": true,
                    "id": 135,
                    "status": "OK",
                    "createdDate": "28-09-2015 15:30:30"
                },
                {
                    "saveEnabled": true,
                    "id": 137,
                    "status": "OK",
                    "createdDate": "28-09-2015 17:23:13"
                }
            ]
        },
        {
```

Get Forms by Theme



```
{
    "count": 2,
    "list": [
        {
            "saveEnabled": true,
            "id": 115,
            "formType": 1,
            "createdDate": "29-09-2015 16:03:55",
            "designation": "Damage Report",
            "version": 0,
            "versionDate": "07-05-2016 01:24:58",
            "digitalSignMandatory": 1,
            "infoDescriptionDetail": "Report flood damage in certain areas",
            "infoTax": "1.5",
            "infoSubmissionMode": "Normal",
            "infoDeadline": "15/01/2018",
            "infoLegislation": "Legal status of Local Authorities / RJAL (Law No. 75/2013 of 12 September, in its present form) and other applicable legislation.",
            "status": "OK",
            "visibleWeb": true,
            "signRequired": false,
            "themeClassFormDefs": [
                {
                    "saveEnabled": true,
                    "id": 137,
                    "status": "OK",
                    "createdDate": "28-09-2015 17:23:13"
                }
            ],
```

8 CDF Release Notes

This section contains the CDF release notes.

8.1 System access requirements

CDF can be accessible through any web browser in any device, doing it a multiplatform tool available everywhere. The browser compatibility list is the next:

- Chrome 63+
- Firefox 57.0.4+
- Internet Explorer 10+

The mobile application is compatible with:

- Android 4.1+

8.2 Features

The CDF component a multi-platform system with these main functionalities:

- Filter and process any incoming issue reported via app or portal;
- A Workflow management system for proper address the issues automatically registered;
    - Allowing the public authority to give feedback to the citizen;
    - Allowing the citizen to give continuous feedback;
- Broadcasting messaging tool with sms integration;
- P2P messaging tool with sms integration;
- Build a database of issues with workflow capabilities and with integration API for the Portal and the EMC;
- Build a database of entities/citizens;
- Build smart forms for dedicated workflows and publish them on the Portal via API;
    - This allows more formal communications and process flows between citizens and the public authorities.
- Allow the public authority to setup the application and workflows as desired with full customization;

8.3 Installation guide

For Decision Makers and/or Operators you can access to the CDF Backoffice directly in the FLOOD-Serv Platform or directly by typing in:

https://bilbao-floodserv-saas.ano.pt/

https://bratislava-floodserv-saas.ano.pt/

https://genova-floodserv-saas.ano.pt/

https://tulcea-floodserv-saas.ano.pt/

https://vnfamalicao-floodserv-saas.ano.pt/

The credentials to access are:

- User: salmeida123
- Password: 123

Note: Since the PA can delete users, this user can at any point be deleted by the PA. If so, please refer directly to the FLOOD-Serv platform to gain access.

To access to the CDF mobile app for Citizens:
You can download it directly via the FLOOD-Serv Platform or using the direct link:

https://tulcea-floodserv-saas.ano.pt/tulcea/images/FLOODserv_1.0.0.4-tulcea.apk

https://bilbao-floodserv-saas.ano.pt/bilbao/images/FLOODserv_1.0.0.4-bilbao.apk

https://genova-floodserv-saas.ano.pt/genova/images/FLOODserv_1.0.0.4-genova.apk

https://bratislava-floodserv-saas.ano.pt/bratislava/images/FLOODserv_1.0.0.4-bratislava.apk

https://vnfamalicao-floodserv-saas.ano.pt/vnfamalicao/images/FLOODserv_1.0.0.4-vnfamalicao.apk

Any verification tests described in this document can be repeated using such links, with the possibility to change the language to verify that the application is running for the five languages of the pilot cities and in English as well.

9 Work Developed and Conclusions

Regarding the CDF component, the 8 user stories were implemented, with all tests proving its compliance with the original requirements. For this, the development consisted on (macro level):

- Implementing the mobile application;
- Implementing the API to integrate with the platform – check Appendix I for more detail;
- Developing the Business Works layer to implement the new underline logic;
- Developing the Data layer to accommodate the new data structures.

The previous technology stack deemed fit for the changes and new modules incorporated.

Under the work developed in WP3, the CDF is ready to be integrated with the FLOOD-Serv system. CDF is also currently being deployed separately for a different project.

10 APPENDIX I: API Documentation

10.1.1 Introduction

The following chapters identify the methods present in the three main areas of the CDF API. For the URL, each pilot has its own CDF instance:

https://bilbao-floodserv-saas.ano.pt/

https://bratislava-floodserv-saas.ano.pt/

https://genova-floodserv-saas.ano.pt/

https://tulcea-floodserv-saas.ano.pt/

https://vnfamalicao-floodserv-saas.ano.pt/

For the API link, they obey the same logic:

https://{pilot_instance_name}/{pilot_contextroot_name}/services/api/records/

Pilot	{pilot_instance_name}	{pilot_contextroot_name}
Bilbao	bilbao	bilbao
Bratislava	bratislava	bratislava
Genova	genova	genova
Tulcea	tulcea	tulcea
Vila Nova de Famalicão	vnfamalicao	vnfamalicao

For example, for the **GetProcessesByDate** of the **STATES** API, for Genova the link is:

https//genova-floodserv-saas.ano.pt/genova/services/api/records/getProcessesByDate/

10.1.2 SYNC Users

10.1.2.1 Sync Users

Link	https://{pilot_instance_name}/{pilot_contextroot_name}/services/api/floodserv/
Path	sync
Method	POST
Produces	text/plain

Parameters from headers	"Authorization":"Basic Auth" “Username”: “USERWS” “Password:” “floodserv123”
Return	“OK”

10.1.3 STATES

10.1.3.1 Get a list of processes filtered by date of creation

Link	https:// {pilot_instance_name}-floodserv-saas.ano.pt/{pilot_contextroot_name}/services/api/records/
Path	getProcessesByDate/{date}
Method	GET
Produces	application/json
Parameters from headers	Authorization => “Basic VVNFUldTOmZsb29kc2VydjEyMw==”
Parameters from path	date => date in milliseconds

Return	[
	  {
	    "id": <process identifier>,
	    "number": <process number>,
	    "year": <process year>,
	    "description": "<process description>",
	    "theme": "<process theme>",
	    "creationDate": "<date created in milliseconds>",
	    "processedBy": "<username>",
	    "status": "<process state>",
	    "entityRequester": "<entity name>"
	  },
	  …
	]

10.1.3.2 Get the list of attachments of a specific process

Link	https:// {pilot_instance_name}-floodserv-saas.ano.pt /{pilot_contextroot_name}/services/api/attachments/
Path	getAttachments/{id}

Method	GET
Produces	application/json
Parameters from headers	Authorization => "Basic VVNFUldTOmZsb29kc2VydjEyMw=="
Parameters from path	id => process identifier

Return	[
	  {
	    "id": <attachment identifier>,
	    "creationDate": "<date created in milliseconds>",
	    "name": "<file name>"
	  }
	]

10.1.3.3 Get the last version of file of a specific attachment

Link	https://{pilot_instance_name}-floodserv-saas.ano.pt /{pilot_contextroot_name}/services/api/attachments/
Path	getFile/{id}
Method	GET
Produces	application/octet-stream
Parameters from headers	Authorization => "Basic VVNFUldTOmZsb29kc2VydjEyMw=="
Parameters from path	id => attachment identifier
Return	The file

10.1.3.4 Get the last version of file of a specific attachment (in base64)

Link	https:// {pilot_instance_name}-floodserv-saas.ano.pt /{pilot_contextroot_name}/services/api/attachments/
Path	getFileBase64/{id}
Method	GET
Produces	application/json

Parameters from headers	Authorization => “Basic VVNFUldTOmZsb29kc2VydjEyMw==”
Parameters from path	id => attachment identifier

Return	{
	“data”: “<base64 encoded file content>”
	}

10.1.3.5 Get list of movements of a process

Link	https://{pilot_instance_name}-floodserv-saas.ano.pt /{pilot_contextroot_name}/services/api/movements/
Path	getMovementsByProcess/{id}
Method	GET
Produces	application/json
Parameters from headers	Authorization => “Basic VVNFUldTOmZsb29kc2VydjEyMw==”
Parameters from path	id => process identifier
Return	[{ "id": <movement identifier>, "number": <movement number, "creationDate": "<date created in milliseconds>", "userOrigin": "<origin user>", "userDestination": "<destination user>", "departmentOrigin": "<origin department>", "departmentDestination": "<destination department>", “resolutionDate”: “<resolution date in milliseconds>”, “resolutionDescription”: “<resolution description>” }, ...]

10.1.3.6 Update the status of a specific process

Link	https:// {pilot_instance_name}-floodserv-saas.ano.pt /{pilot_contextroot_name}/services/api/records/
Path	updateState/{id}/{status}
Method	POST
Produces	application/json
Parameters from headers	Authorization => “Basic VVNFUldTOmZsb29kc2VydjEyMw==”
Parameters from path	id => process identifier status => the new process state
Return	A boolean indicating success or failure

10.1.3.7 Get a list of entities

Link	https:// {pilot_instance_name}-floodserv-saas.ano.pt /{pilot_contextroot_name}/services/api/persons/
Path	getEntities
Method	GET
Produces	application/json
Parameters from headers	Authorization => “Basic VVNFUldTOmZsb29kc2VydjEyMw==”

Return	
	```
	[
	  {
	    "id": <entity identifier>,
	    "name": "<entity name>",
	    “number”: “<entity number>”
	  },
	  …
	]
	```

10.1.3.8 Get full details on a specific entity

Link	https:// {pilot_instance_name}-floodserv-saas.ano.pt /{pilot_contextroot_name}/services/api/persons/
Path	getCompleteEntity/{id}
Method	GET
Produces	application/json
Parameters from headers	Authorization => "Basic VVNFUldTOmZsb29kc2VydjEyMw=="
Parameters from path	id => entity identifier

Return	{
	 "id": "<entity identifier>",
	 "name": "<entity name>",
	 "number": "<entity number>",
	 "email": "<entity e-mail>",
	 "phoneNumber": "<entity phone number>",
	 "birthday": "<entity birth date in milliseconds>",
	 "address": {
	 "id": "<address identifier>",
	 "street": "<street>",
	 "district": "<district>",
	 "county": "<county>",
	 "town": "<town>",
	 "postalCode": "<postalCode>"
	 }
	 }

10.1.3.9 Base Data

10.1.3.9.1 Available process states

- R - Registry
- P - Pending
- A - Archived
- UA - Unarchived
- S - In follow-up
- DR- Draft

10.1.4 WEB REQUESTS

10.1.4.1 Login User

Link	http://195.82.131.198/oauth2_server/public/index.php
Path	api/login
Tip	POST
Parameter from body	email* password*
Return	"token_type" "expires_in " “access_token

10.1.4.2 Report Issue

Link	https:// {pilot_instance_name}-floodserv-saas.ano.pt /{pilot_contextroot_name}/services/api/floodserv/
Path	report
Method	POST
Consumes	multipart/form-data
Produces	application/json

Parameters from headers	Authorization => “Basic VVNFUldTOmZ1dHVyZWRvYw==”
	token => the Oauth2 access token

Parameters from body	title => title of the issue description => description of the issue latitude => latitude (location) longitude => longitude (location) attachments => the images and videos, as a list of multipart attachments named “attachment1”, “attachment2”, etc.

Return	{
	“newId”: “<internal ID of the created issue/request>”,
	“state”: “SENT”
	}

10.1.4.3 Get Issue State

Link	https:// {pilot_instance_name}-floodserv-saas.ano.pt /{pilot_contextroot_name}/services/api/floodserv/
Path	getState/{id}
Method	GET
Produces	application/json

Parameters from headers	Authorization => “Basic VVNFUldTOmZ1dHVyZWRvYw==”
	
	token => the Oauth2 access token

Parameters from query	Id => internal ID of the issue/request

Return	{
	“state”: “<state of the request>”,
	“processNumberDisplay”: “<created process number display>”,
	“stateMessageId”: “<string ID of the state of the request for Android>”,
	“resolutionMessageId”: “<string iD of the resolution for Android>”,
	“resolutionDate”: “<resolution date (number of milliseconds since January 1, 1970, 00:00:00)>”
	}

10.1.4.4 Get Reported Issues

Link	https:// {pilot_instance_name}-floodserv-saas.ano.pt /{pilot_contextroot_name}/services/api/floodserv/
Path	getIssues
Method	GET
Produces	application/json

Parameters from headers	Authorization => “Basic VVNFUldTOmZ1dHVyZWRvYw==”
	
	token => the Oauth2 access token

Return	{
	“issues”: [
	{
	"id": "<internal ID of the issue/request>",
	"general": {
	"title": "<issue title>",
	"description": "<issue description>",

	"dateSent": "<send date (number of milliseconds since January 1, 1970, 00:00:00)>"
	},
	"state": {
	“state”: “<state of the request>”,
	“processNumberDisplay”: “<created process number display>”,
	“stateMessageId”: “<string ID of the state of the request for Android>”,
	“resolutionMessageId”: “<string iD of the resolution for Android>”,
	“resolutionDate”: “<resolution date (number of milliseconds since January 1, 1970, 00:00:00)>”
	},
	"location": {
	"latitude": "<latitude>",
	"longitude": "<longitude>"
	},
	"attachments": {
	"count": "<number of attachments>",
	"attachments": [
	{
	"id": "<internal ID of the attachment>",
	"originalName": "<original file name>",
	"size": "<file size (bytes)>",
	"dateSent": "<send date (number of milliseconds since January 1, 1970, 00:00:00)>"
	},
	...
	]
	}
	},
	...
	]
	}

10.1.4.5 Download Attachment

Link	https://{pilot_instance_name}-floodserv-saas.ano.pt /{pilot_contextroot_name}/services/api/floodserv/
Path	getAttachment/{id}
Method	GET
Produces	application/octet-stream
Parameters from headers	Authorization => "Basic VVNFUldTOmZ1dHVyZWRvYw==" token => the Oauth2 access token
Parameters from query	Id => internal ID of the attachment
Return	The file

10.1.4.6 Base Data

10.1.4.6.1 Possible Values for "state"

- DRAFT
- SENT
- PREPARATION
- ACCEPTED
- ACCEPTED ARCHIVED
- ACCEPTED DEFERRED
- ACCEPTED REJECTED
- ACCEPTED DEFERRED ARCHIVED
- ACCEPTED REJECTED ARCHIVED
- ERROR

10.1.4.6.2 Possible values for "stateMessageId"

- issue_state_full_DRAFT
- issue_state_full_SENT
- issue_state_full_PREPARATION
- issue_state_full_NOTACCEPTED
- issue_state_full_ACCEPTED
- issue_state_full_ACCEPTED_ARCHIVED
- issue_state_full_ACCEPTED_DEFERRED
- issue_state_full_ACCEPTED_REJECTED
- issue_state_full_ACCEPTED_DEFERRED_ARCHIVED
- issue_state_full_ACCEPTED_REJECTED_ARCHIVED
- issue_state_full_ERROR

10.1.4.6.3 Possible values for “resolutionMessageId”

- issue_resolution_d
- issue_resolution_r

10.1.5 WEB MESSAGES

10.1.5.1 Messages Received

Link	https://{pilot_instance_name}-floodserv-saas.ano.pt /{pilot_contextroot_name}/services/api/floodserv/getSentWebMessagesByUser
Path	api/floodserv/getSentWebMessagesByUser
Type	GET
Parameters from headers	token

Return	JSON (application/json)
	{
	  "count": 2,
	  "list": [
	    {
	      "saveEnabled": true,
	      "id": 1447,
	      "entryDate": "11-06-2019 14:17:12",
	      "type": "NOR",
	      "wmePrioridade": 0,
	      "title": "test",
	      "details": "test_details",
	      "from": "USER",
	      "to": "APP",
	      "haveAttachs": "NO",
	      "dataOrigin": "GSE_R4",
	      "read": false
	    },
	    {
	      "saveEnabled": true,
	      "id": 1446,
	      "entryDate": "07-06-2019 17:03:29",

	"type": "NOR", "wmePrioridade": 0, "title": "OK", "details": "OK", "from": "USER", "to": "APP", "viewDate": "07-06-2019 17:04:05", "haveAttachs": "NO", "read": true }] }

I/O:

```
@GET
@Path("/getSentWebMessagesByUser")
@Produces(MediaType.APPLICATION_JSON)
PaginationModel<WebMessage> getSentMessages(
        @HeaderParam("token") String token,
        @DefaultValue("0") @QueryParam("offset") Integer offset,
        @DefaultValue("10") @QueryParam("limit") Integer limit,
        @DefaultValue("-entryDate") @QueryParam("orderBy") String orderBy,
        @QueryParam("filter") String filter
);
```

10.1.5.2 Messages Sent

Link	https://{pilot_instance_name}-floodserv-saas.ano.pt /{pilot_contextroot_name}/services/api/floodserv/getSentWebMessagesByUser
Path	api/floodserv/getSentWebMessagesByUser
Type	GET
Parameters from headers	token
Return	JSON (application/json) {

```
  "count": 2,
  "list": [
    {
      "saveEnabled": true,
      "id": 1447,
      "entryDate": "11-06-2019 14:17:12",
      "type": "NOR",
      "wmePrioridade": 0,
      "title": "test",
      "details": "test_details",
      "from": "USER",
      "to": "APP",
      "haveAttachs": "NO",
      "dataOrigin": "GSE_R4",
      "read": false
    },
    {
      "saveEnabled": true,
      "id": 1446,
      "entryDate": "07-06-2019 17:03:29",
      "type": "NOR",
      "wmePrioridade": 0,
      "title": "OK",
      "details": "OK",
      "from": "USER",
      "to": "APP",
      "viewDate": "07-06-2019 17:04:05",
      "haveAttachs": "NO",
      "read": true
    }
  ]
}
```

Note: The list can be filter as the following example: https://{pilot_instance_name}-floodserv-saas.ano.pt /{pilot_contextroot_name}/services/api/floodserv/getSentWebMessagesByUser?offset=0&limit=1.

I/O:

```
@GET
@Path("/getSentWebMessagesByUser")
@Produces(MediaType.APPLICATION_JSON)
PaginationModel<WebMessage> getSentMessages(
        @HeaderParam("token") String token,
        @DefaultValue("0") @QueryParam("offset") Integer offset,
        @DefaultValue("10") @QueryParam("limit") Integer limit,
        @DefaultValue("-entryDate") @QueryParam("orderBy") String orderBy,
        @QueryParam("filter") String filter
);
```

10.1.5.3 Create New Message

Link	https://{pilot_instance_name}-floodserv-saas.ano.pt /{pilot_contextroot_name}/services/api/floodserv/createNewWebMessage
Path	api/floodserv/createNewWebMessage
Type	POST
Parameters from headers	token

Parameters from body	JSON (application/json)
	{
	"title" : "test",
	"details" : "test_details"
	}

Return	JSON (application/json) { "saveEnabled": true, "id": 1447, "entryDate": "11-06-2019 14:17:12", "type": "NOR", "wmePrioridade": 0, "title": "test", "details": "test_details", "from": "USER",

	"to": "APP", "haveAttachs": "NO", "dataOrigin": "GSE_R4", "read": false }

I/O:

```
@POST
@Path("/createNewWebMessage")
@Consumes(MediaType.APPLICATION_JSON)
@Produces(MediaType.APPLICATION_JSON)
WebMessage createNewWebMessage(
        @HeaderParam("token") String token,
        WebMessage wmsg);
```

11 APPENDIX I: User Guide

The user guide, in Powerpoint and Video format are available in each instance of the CDF of each pilot in the following links:

- https://bilbao-floodserv-saas.ano.pt/bilbao/images/CDF_Quick_guide.pptx
- https://bratislava-floodserv-saas.ano.pt/bratislava/images/CDF_Quick_guide.pptx
- https://genova-floodserv-saas.ano.pt/genova/images/CDF_Quick_guide.pptx
- https://tulcea-floodserv-saas.ano.pt/tulcea/images/CDF_Quick_guide.pptx
- https://vnfamalicao-floodserv-saas.ano.pt/vnfamalicao/images/CDF_Quick_guide.pptx