CITY.RISKS

AVOIDING AND MITIGATING SAFETY RISKS IN URBAN ENVIRONMENTS

Deliverable D5.4

Report on System Verification

Editors N. Bakalos (ICCS)

Contributors N. Papadakis (SPH), S. Costicoglou (SPH), A. Litke (INFT) S. Pfennigschmidt (FRAUNHOFER), D. Skoutas (ATH), P. Mehta (FUB)

Version 1.0

Date 15/6/2017

Distribution PUBLIC (PU)

GRANT AGREEMENT NO.653747

Executive Summary

This report presents the efforts towards the verification of the City.Risks Integrated System. The verification took place on a module by module basis, testing the communication and validating the proper exchange of data between each developed service.

An exhaustive set of test cases were performed, testing all aspects of data exchange and communication. The test cases were developed according to the requirements of the City.Risks platform and its individual components as presented in D2.4 “Use Case Requirements and KPIs”. This reports presents these test cases and their outcomes.

Table of Contents

Introduction

The overall architecture of the City.Risks integrated prototype is presented in the figure below.

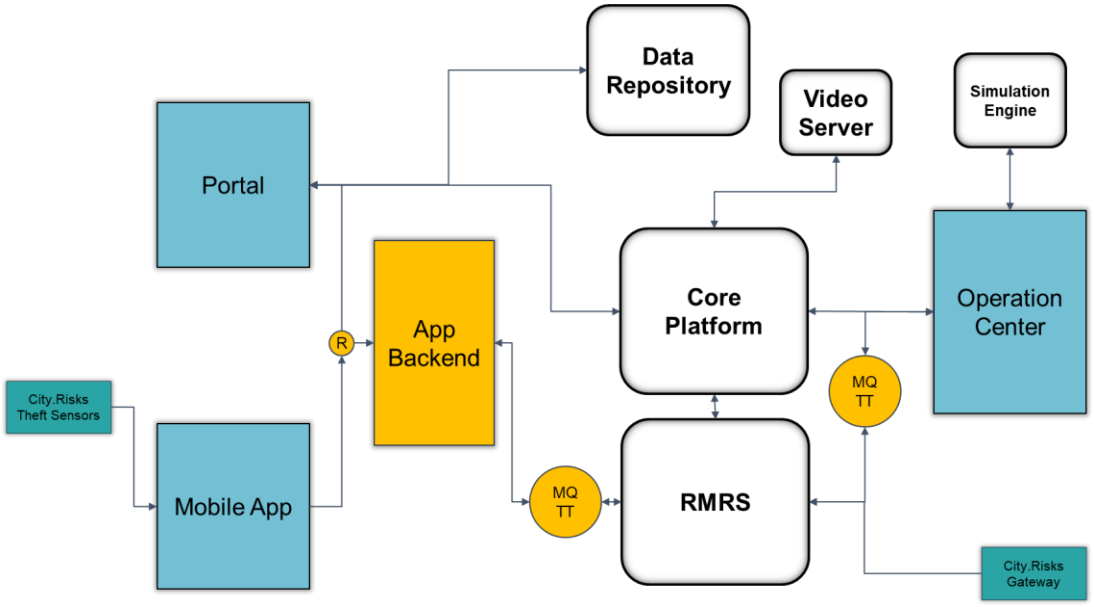

Figure 1: City.Risks Integrated System Architecture

For the verification of this architecture as well as all the systems that compose the City.Risks prototype platform a large number of testcases were designed and executed. The test-cases were designed based on the specifications of the services as derived in D2.5 and also the Use case Scenarios of City.Risks as described in D2.4.

The verification test were designed to assess the City.Risks' system response based on the Requirements and Use Case Scenarios that were set in D2.4. Table 1 reiterates the Use Case scenarios.

Table 1: City.Risks Use Cases

Use Case Scenario
UC1: Theft of personal belongings
UC2: Vehicle Theft
UC3: Information Gathering and Dissemination of Ongoing Events
UC4: Tourists' and Women's Safety
UC5 Citizen Engagement
UC6: Neighbourhood Safety

1. Verification Scenario 1: Registering, editing user profile,

In this verification test the Web Application User and Mobile Application User (City.Risks portal and mobile app respectively) can edit their profile data. This verification scenario covers the following City.Risks Requirements;

Code	Description
R1.1.1	An MAU is able to create and edit her/his profile with personal information and preferences, including rules and policies for sharing information.
R1.3.1	The Web Portal subsystem can provide an interface to create and edit user profile with personal information and preferences, including rules and policies for sharing information.

For the Web portal the following steps were executed

Test Case	Individual Steps	Relevant Requirement	Outcome
Edit WAU Profile	User Registration to the Core Platform	R1.3.1	Success
	User Login	R1.3.1	Success
	View User Profile	R1.3.1	Success
	Edit User Profile	R1.3.1	Success

For the mobile application the device stores all the data, protecting the users privacy. The Mobile Application User (MAU) can edit user data and set rules regarding receiving notifications, sharing his/her location and being available for the chat functionality. The following steps were executed

Test Case	Individual Steps	Relevant Use Case Scenario	Outcome
Edit MAU Profile	View User Profile	R1.1.1	Success
	Edit User Profile	R1.1.1	Success
	Change rules regarding notifications, chat availability etc.	R1.1.1	Success

2. Verification Scenario 2: Incident Reporting

In this verification scenario, the Mobile Application can provide reports about incidents. These reports are filtered through RMRS, are received in the City.Risks Operation Center and disseminated in both the web portal and mobile application through various interfaces (maps, augmented reality interfaces).

This verification scenario responds to the following City.Risks requirements:

Group 1: Actor - Mobile Application User (MAU)	
Code	**Description**
R1.1.2	An MAU is able to create and send an incident report to the Operation Center.
R1.1.3	An MAU is able to receive location-based notifications, including warnings and alerts.
R1.1.4	An MAU is able to capture and send multimedia content, such as audio, video and images, via the mobile application to the Operation Center subsystem.
R1.1.5	An MAU is able to view crime statistics by location, time and type on a map.
R1.1.6	An MAU is able to view safety indicators for an area, e.g., safety levels for women, LGBT citizens or ethnic minorities.
R1.1.7	An MAU is able to use augmented reality to view the crime-related information about her/his surroundings.
R1.1.8	An MAU is able to interact with objects presented in augmented reality, i.e., add information to objects, update objects and create new objects.
R1.1.15	An MAU is able to receive personalized and visualized risk information about an area through updates, maps and mAR based on their profile, location, time and preferences.
R1.1.20	An MAU is able to report problems and complaints, such as illegal dropping of waste and illegal business activity, to the Operation Center subsystem.
R1.1.23	An MAU is able to post text messages in the community that she/he is a part of.
R1.1.24	An MAU is able to post multimedia files, such as images, audio and videos, in the community that she/he is a part of.
R1.1.25	An MAU is able to send requests for witnesses to the Operation Center subsystem with the incident location and time.
R1.1.26	An MAU is able to specify a storage limit for her/his history of locations and timestamps on the mobile device.
Group 2: Actor – Operation Center User (OCU)	
R1.2.1	An OCU is able to keep track of ongoing incidents using map and text-based interfaces.
R1.2.2	An OCU is able to filter event information based on event attributes, such as location and time.

R1.2.3	An OCU is able to aggregate event information from different sources.
R1.2.4	An OCU is able to visualize event information over a map.
R1.2.5	An OCU is able to send a common notification, such as an alert or a warning, to a specific subset of recipients based on their location.
R1.2.6	An OCU is able to view reports of an incident over a map.
R1.2.7	An OCU is able to locate MAU within a specific range of an incident occurrence.
R1.2.8	An OCU is able to send advisories and instructions to MAU based on their location.
R1.2.9	An OCU is able to simulate different scenarios for planning and preparedness, e.g., using agent-based simulations.
Group 3: Access Layer – Web Portal Subsystem	
R1.3.2	The Web Portal subsystem can display crime statistics by location, time and type on a map.
Group 6: Application Layer – Risks Management Subsystem	
R1.6.1	The Risks Management subsystem is able to track the position of an MAU.
R1.6.2	The Risks Management subsystem is able to monitor MAU near a specific MAU.
R1.6.3	The Risks Management subsystem is able to monitor static (maps, crime statistics) and real-time (current events and conditions) data about a location.
R1.6.4	The Risks Management subsystem is able to supply an MAU at risk with situation-aware personalized information and advisories on how to respond.
R1.6.5	The Risks Management subsystem is able to send warnings and alerts to a certain MAU or a group of MAU based on location.
R1.6.6	The Risks Management subsystem is able to aggregate and visualize event-related information.
Group 9: Remote Layer – Mobile Application	
R1.9.4	The mobile application provides an interface for rendering, generating and modifying content based on the mAR data description language.
R1.9.5	The mobile application is able to interact with the mAR WPS for generating and displaying dynamic mAR representations in real-time.
R1.9.6	The mobile application is able to display safety and risk information in the mAR view of the mobile device.
R1.9.13	The mobile application is able to evaluate based on location history if it was in the spatial and temporal range of an incident.

2.1. City.Risks Portal

For the City.Risks portal the following steps were executed

Test Case Type	Test Case	Individual Steps	Relevant Requirements	Outcome
Receiving Reports & Alerts	Receive Incidents	Portal Queries RMRS to receive active incidents	R1.6.6	Success
		RMRS sends active incident list	R1.6.3, R1.6.4, R1.6.5	Success
		Web Portal Receives the list of active incidents	R1.1.3, R1.1.4, R1.3.2	Success
		Web Portal displays incidents on the map	R1.3.2	Success
	Receive Incident Reports	Web Portal queries RMRS to receive reports associated with an active incident	R1.6.6	Success
		RMRS sends reports connected to the incident	R1.6.6	Success
		Web portal receive the incident reports	R1.1.3, R1.1.4, R1.3.2	Success
		Web Portal displays the incident report	R1.3.2	Success

2.2. City.Risks Mobile App

For the City.Risks Mobile App the following test cases were executed

Test Case Type	Test Case	Individual Steps	Relevant Requirements	Outcome
Sending Reports	App sends a report	RMRS creates a new incident	R1.1.2, R1.1.4, R1.1.20, R1.1.23,	Success

			R1.1.24, R1.1.25, R1.1.26, R1.6.6	
		OC receives and displays the incident	R1.2.1, R1.2.2, R1.2.3, R1.2.4, R1.2.5, R1.2.6	Success
		App shows the incident	R1.2.7, R1.2.8, R1.6.5, R1.9.4, R1.9.5, R1.9.6	Success
		Web portal shows the incident	R1.3.2	Success
	App sends a report for an existing incident	RMRS attaches the report to the incident	R1.1.2, R1.1.4, R1.1.20, R1.1.23, R1.1.24, R1.1.25, R1.1.26, R1.6.6	Success
		OC receives the update and displays the incident	R1.2.1, R1.2.2, R1.2.3, R1.2.4, R1.2.5, R1.2.6	Success
		App shows the report in the incident	R1.2.7, R1.2.8, R1.6.5, R1.9.4, R1.9.5, R1.9.6	Success
		Web portal shows the report in the incident	R1.3.2	Success
	App send a first FLAG report for an incident	RMRS increments the flag property of the incident	R1.1.2, R1.1.4, R1.6.6	Success
		OC receives the update and displays the updated flag count	R1.2.1, R1.2.2, R1.2.3, R1.2.4, R1.2.5, R1.2.6	Success
		App shows the incident has been flagged	R1.2.7, R1.2.8, R1.6.5, R1.9.4, R1.9.5, R1.9.6	Success

		Web Portal shows the incident has been flagged	R1.3.2	Success
	App sends a second FLAG report for an incident	RMRS increments the flag property of the incident	R1.1.2, R1.1.4, R1.6.6	Success
		OC receives the update and displays the updated flag count	R1.2.1, R1.2.2, R1.2.3, R1.2.4, R1.2.5, R1.2.6	Success
		App shows the incident has been flagged	R1.2.7, R1.2.8, R1.6.5, R1.9.4, R1.9.5, R1.9.6	Success
		Web Portal shows the incident has been flagged	R1.3.2	Success

2.3. City.Risks Operation Center

For the City.Risks Operation Center the following test cases were executed

Test Case Type	Test Case	Individual Steps	Relevant Requirements	Outcome
sending alerts	OC sends new alert without existing incident	RMRS creates a new incident	R1.6.1, R1.6.3	Success
		OC receives and displays the incident	R1.2.1, R1.2.2, R1.2.3, R1.2.4, R1.2.5, R1.2.6,	Success
		App shows the incident with the alert	R1.2.7, R1.2.8, R1.6.5, R1.9.4, R1.9.5, R1.9.6	Success
		Web Portal shows the incident with the alert	R1.3.2	Success

	OC sends a new alert for an existing incident	RMRS attaches the alert to the incident	R1.6.1, R1.6.2, R1.6.3, R1.6.4, R1.6.6	Success
		OC receives and displays the updated incident	R1.2.1, R1.2.2, R1.2.3, R1.2.4, R1.2.5, R1.2.6,	Success
		App shows the incident with the alert	R1.2.7, R1.2.8, R1.6.5, R1.9.4, R1.9.5, R1.9.6	Success
		Web Portal shows the incident with the alert	R1.3.2	Success
sending operations	OC sends a close incident operation to RMRS	RMRS closes the incident	R1.6.1, R1.6.2, R1.6.3, R1.6.4, R1.6.6	Success
		OC receives the incident	R1.2.1, R1.2.2, R1.2.3, R1.2.4, R1.2.5, R1.2.6	Success
		App will no longer show the incident	R1.2.7, R1.2.8, R1.6.5, R1.9.4, R1.9.5, R1.9.6	Success
		Web-Portal will no longer show the incident	R1.3.2	Success
	OC sends a remove report operation	RMRS removes the report from the incident	R1.6.1, R1.6.2, R1.6.3, R1.6.4, R1.6.6	Success
		OC receives and displays the updated incident	R1.2.1, R1.2.2, R1.2.3, R1.2.4, R1.2.5, R1.2.6	Success
		App no longer shows the report for the incident	R1.2.7, R1.2.8, R1.6.5, R1.9.4, R1.9.5, R1.9.6	Success
		Web-Portal no longer shows the report for the incident	R1.3.2	Success
	OC sends re-classification result for an incident	RMRS sets the incident type	R1.6.1, R1.6.2, R1.6.3, R1.6.4, R1.6.6	Success

		OC receives and displays the updated incident	R1.2.1, R1.2.2, R1.2.3, R1.2.4, R1.2.5, R1.2.6	Success
		App shows the new type	R1.2.7, R1.2.8, R1.6.5, R1.9.4, R1.9.5, R1.9.6	Success
		Web portal shows the new type	R1.3.2	Success

3. Verification Scenario 3: Theft Detection

In this verification scenario, the mobile application, operation center and City.Risks BLE theft sensor were verified. The scenario tests the activation of a sensor, ability of the mobile app to "sight" a stolen sensor and report the sighting back to the operation center.

This verification scenario covers the following City.Risks requirements.

Group 1: Actor - Mobile Application User (MAU)	
Code	**Description**
R1.1.9	An MAU is able to register a theft detection sensor with the Operation Center.
R1.1.10	An MAU is able to report a theft to the Operation Center via the mobile application.
Group 5: Application Layer – Operation Center Subsystem	
R1.5.3	The Operation Center subsystem is able to activate a theft detection sensor remotely by generating and sending a wake-up signal.
R1.5.7	The Operation Center subsystem is able to re-estimate the reach area around the stolen area with time.
Group 9: Remote Layer – Mobile Application	
R1.9.2	The mobile application is able to identify a specific theft detection sensor or IoT device, i.e., a sensor or IoT device attached to an item/vehicle that was stolen
R1.9.7	The mobile application is able to communicate with theft detection sensors via a proximity-based mechanism.
R1.9.8	The mobile application is able to receive a short-range signal from a theft detection sensor.
R1.9.9	The mobile application is able to send a notification with the position, time and ID of a nearby theft detection sensor to the Operation Center.
Group 10: Remote Layer – Theft Detection Subsystem	
R1.10.3	The theft detection sensor is able to communicate with the mobile application using Bluetooth Low Energy radio.
R1.10.4	The theft detection sensor is able to broadcast a signal periodically to mobile devices in proximity.
R1.10.6	The theft detection sensor is able to get activated by a remote signal from the Operation Center subsystem.

The following steps were executed to address this verification scenario:

Test Case	Individual Steps	Relevant Requirements	Outcome
App sends a stolen item report for a registered tag	RMRS creates a new Incident	R1.1.10	Success
	OC receives the incident and displays it	R1.5.3, R1.5.7	Success
RMRS sends theft notification to App	App receives it asking for user consent	R1.5.3, R1.5.7, R1.9.7, R1.9.8, R1.9.9R.1.9.9	Success
	OC receives and displays the update	R1.5.3, R1.5.7,	Success
App sends sighting to RMRS	RMRS attaches the sighting to the incident	R1.9.2, R1.9.9, R1.10.3, R1.10.4, 1.10.6 R1.9.7, R1.9.8, R1.9.9	Success
	OC receives and displays the update	R1.5.3, R1.5.7	Success
App registers a tag	App displays the tag information	R1.1.9, R1.9.7, R1.9.8, R1.9.9, R1.10.3, R1.10.4	Success

4. Verification Scenario 4: Safe Routing

In this verification scenario, the safe routing services are tested. The relevant City.Risks requirements are:

Code	Description
R1.1.16	An MAU is able to search for routes between two locations on a map.
R1.1.17	An MAU is able to view safety related information on routes between two locations on a map.

For the City.Risks portal the following test cases were executed

Test Case	Individual Steps	Relevant Requirements	Outcome
Receive Safe Route	Web portal request a safe route from the data repository given a starting and end point	R1.1.16	Success
	Data Repository returns the route	R1.1.16	Success
	Web portal receives the route and displays It on a map	R1.1.16	Success

While the web portal visualises a more static safe routing services, that is based only on historical data, the mobile app is a more live service, taking into account active incidents and rerouting in the case of an incident report intersecting with the suggested route. For the mobile app the following test cases were executed.

Test Case	Individual Steps	Relevant Requirements	Outcome
Receive Safe Route in the App	App sends routing request	R1.1.16, R1.1.17	Success
	App receives the route and displays it	R1.1.16, R1.1.17	Success
	The app reroutes when an incident report intersects with the route	R1.1.16, R1.1.17	Success

5. Verification Scenario 5: Request for Witness

In this verification scenario, the Mobile app is sending a request for witness to RMRS. RMRS filters the user data and correlates a MAU's position with the report's location during a specific timeslot. A group chat is created and MAU's can enter this chat after receiving a notification and provide data to the MAU that requested the witness.

The following requirements are covered:

Group 1: Actor - Mobile Application User (MAU)	
Code	**Description**
R1.1.27	An MAU is able to receive requests for witnesses by the Operation Center subsystem.
R1.1.28	An MAU is able to confirm or reject requests for witnesses from the Operation Center subsystem.

To verify this the following steps were executed:

Test Case	Individual Steps	Relevant Requirements	Outcome
MAU creates a new request for witness	App sends a new request for witness	R1.1.27, R1.1.28	Success
	RMRS invites relevant users	R1.1.27, R1.1.28	Success
	App displays the request	R1.1.27, R1.1.28	Success
	A group chat is initialised	R1.1.27, R1.1.28	Success
	MAU's join or ignore the notifications	R1.1.27, R1.1.28	Success
App removes a request for witness	App no longer displays the request	R1.1.27, R1.1.28	Success

6. Verification Scenario 6: Crime Related Data and Safe Points

In this verification scenario, the Resource and Data Layer (Data Repository) and the Web Portal can provide services relevant with historical crime data and city data. The following requirements are addressed in the scenario

Code	Description
Group 3: Access Layer – Web Portal Subsystem	
R1.3.2	The Web Portal subsystem can display crime statistics by location, time and type on a map. (in connection with the Resource and Data Layer Green)
Group 8: Resource and Data Layer	
R1.8.1	The Resource and Data Layer is able to add, update and provide data through an API.
R1.8.2	The Resource and Data layer is able to access and update different types of content from different data sources. This includes: ▪ crime reports and statistics ▪ geospatial data (maps) ▪ demographic data ▪ 3D environment data (if available) ▪ multimedia data (audio, video, images) ▪ theft detection sensor registration data (sensor registry) ▪ applications and services registration data (tools registry) ▪ user IDs, user profiles, user groups and roles (user registry)
R1.8.3	The Resource and Data layer is able to support different types of attribute-based queries.
R1.8.4	The Resource and Data Layer is able to support different types of analysis (i.e., offline, real-time and continuous).

The following test cases were executed:

Test Case Type	Test Case	Individual Steps	Relevant Requirements	Outcome
Receive Data from the Data repository	Receive crime related data	Web portal queries the Data Repository giving the area of interest, crime type and data (month, year)	R.1.3.2, R1.8.2	Success
		Data Repository sends crime data related with the query parameters	R.1.8.1, R1.8.2, R.1.8.3, R.1.8.4	Success
		Web Portal receives the parameters and	R.1.3.2, R1.8.2	Success

		visualizes them on the map		
	Receive Points of interest	Web Portal queries the repository for point's of interest near one specific area	R.1.3.2, R1.8.2	Success
		Data repository returns POI data in accordance with the query parameters	R.1.8.1, R1.8.2, R.1.8.3, R.1.8.4	Success
		Web portal receives the data from the Data repository and displays it.	R.1.3.2, R1.8.2	Success

7. Other Verification Tests

The following requirements were covered in the development of the individual City.Risks components as presented in D3.1 “City.Risks Core Platform”, D3.2 “City.Risks Data Repository, D3.5 “Theft Detection Sensor Validation” and D3.6 “City.Risks SDK”, D4.1 “Risk Management and Response Engine”, D4.2 “City.Risks Web Portal and API”, D4.3 “Geo-based augmented reality mobile application”, D4.4 “Theft detection mobile application” and, D5.1 “City.,Risks Operation Center”.

Code	Description
Group 4: Access Layer – Interoperability Subsystem	
R1.4.1	The Interoperability subsystem can enable program-to-program interoperability using an API and open web standards-based protocols (e.g. HTTP, SOAP, XML, etc.).
R1.4.2	The Interoperability subsystem can allow a third-party application to access and query data to provide additional features on top of the City.Risks platform.
Group 5: Application Layer – Operation Center Subsystem	
R1.5.1	The Operation Center subsystem is able to provide specific crime-related structured data to the security services via a REST API based on parameters like time period, area or type.
R1.5.2	The Operation Center subsystem is able to extract contextual data about an incident from web sources.
Group 7: Application Layer – Core Platform Subsystem	
R1.7.1	The Core Platform subsystem can enable a third-party to build and integrate a custom application into the City.Risks platform using the SDK.
R1.7.2	The Core Platform can provide low level system monitoring and event logging.
R1.7.3	The Core Platform subsystem is able to provide an infrastructure for managing user communities, e.g., creating communities and updating community memberships.
R1.7.4	The Core Platform subsystem is able to provide an infrastructure for managing community policies, e.g., roles and access.
Group 9: Remote Layer – Mobile Application	
R1.9.1	The mobile application is able to communicate with the Operation Center subsystem over wireless interfaces.
Group 10: Remote Layer – Theft Detection Subsystem	
R1.10.1	The theft detection sensor is able to function for a long time period without an external power source.
R1.10.2	The theft detection sensor can be attached to a personal item, such as handbag, luggage or bicycle.
R1.10.5	The theft detection sensor is able to go into hibernation mode.