Name and Surname author 1
Name and Surname author 2
Name and Surname author 3

Issuing Organisation Name

*MySystem (v1.0)*

# Messir User Manual
# - v 1.0.3 -

**Based on IEEE Std 1063-2001 [1]**

Thursday 2nd February, 2017 - 19:20

Springer

# Contents

# List of Figures

# Chapter 1
# Product information

## 1.1 Identification

Include precise information of the software product like identification name (that you can include in the Glossary), list of parts that compose it (indicating identification numbers for each part). Specify the applicable operating environment(s), including version(s) of hardware, communications, and operating system(s).

## 1.2 Copyright

## 1.3 Trademark notices

## 1.4 Restrictions

Restrictions on copying or distributing the software and its associated documentation.

## 1.5 Warranties

- The system has an ability to handle a certain volume of requests
- The system is designed such that has an ability to handle higher volumes, the system provides scalability
- The performance of the system will be provided by
- The system's performance, operations, functionality and maintenance conforms to the specification
- The version of the system is the most current
- The systems does not contain any disabling features, viruses, unidentified content, instructions to permit unauthorized access to the system
- The supplier provides a customer with information regarding any failure of security measures and executes regular audits
- The supplier has the right to grant the customer the rights of use of the software
- The technology is a original work of iCrash LLC and does not violate the rights of any third party

## 1.6 Contractual obligations

## 1.7 Disclaimers

iCrash LLC assumes no responsibility for the use of the software, conveys no license under any patent, copyright, or mask work right to the product. Use of the software by a user is at the user's risk. iCrash LLC reserves the right to make changes in the software without notification. iCrash LLC makes warranty that such application will be suitable for the specified use without further testing or modification.

## 1.8 Contact

Phone numbers:

General inquiries
+7 999 555 55 55

Technical Support
+7 999 999 50 00

Mailing address:

info@icrash.com

iCrash LLC
1 Universitetskaya
Innopolis, Russia 420500

# Chapter 2
# Introduction

## 2.1 Scope

Introduction scope goes right here. An example of using glossay might be.Zzzz wooZ
This document KuzmaTermName provides ...
This document does not ...
This document is not ...
This document may be used with ...

## 2.2 Purpose

In this section you explain the purpose (i.e. aim, objectives) of the user's manual. In the following some examples of opening statements to be used in this section. W
The purpose of this document is ...
This document defines ...
This document is meant to ...

## 2.3 Intended audience

Description of the categories of persons targeted by this document together with the description of how they are expected to exploit the content of the document.

## 2.4 *MySystem (v1.0)*

Here test test

### *2.4.1 Actors & Functionalities*

Overview of all the ***actors*** interacting with the software being them either humans (called end-users in the standard [1]) or not. For each actor, describe the main software functions that are offered to him. Structure of this sub-section MUST be by actor/functionalities.

- *actComCompany*: Communication Company
  - Delivering any SMS sent by any human to the iCrash's phone number.

  - Transmit SMS messages from the ABC company that owns the iCrash system to any human having an SMS compatible device accessible using aphone number.

- *actAdministrator*: Administrator
  - Adding or deleting coordinator actors from the system and its environment.
- *actCoordinator*: Coordinator
  - Monitoring the existing alerts and crisis.
  - Managing alerts and crisis until their termination.
- *actActivator*: Activator
  - Communicate the current time to the system.
  - Notify the administrator that some crisis are still pending for a too long time.
- *actMsrCreator*: Creator
  - Installing the iCrash system.
  - Defining the values for the initial system's state.
  - Defining the values for the initial system's environment.
  - Ensuring the integration of the iCrash system with its initial environment.

### *2.4.2 Operating environment*

Brief overview of the infrastructure on which the software is deployed and used.

## 2.5 Document structure

Information on how this document is organised and it is expected to be used. Recommendations on which members of the audience should consult which sections of the document, and explanations about the used notation (i.e. description of formats and conventions) must also be provided.

# Chapter 3
# Usage Guide

This section is aimed at describing the general use of the software, since it is **deployed, configured** and **run**.

This software is used by actors. These actors rely on the software to perform a set of business activities (called here procedures) aimed at reaching a particular goal.

These prodedures are splet in two groups:

- **Multi-procedures:** which are procedures at **summary** or **user-goal** level involving several active or pro-active actors. Each of these procedures aims at illustrating intertwined business activities required to be performed by the involved actors to reach the expected goal. Each business activity between the system and an actor must correspond to a **system operation** instance given with actual parameter values.
- **Mono-procedures:** which are procedures at **summary** or **user-goal** level involving only one active or pro-active actor. Each of these procedures aims at illustrating the required business activities an actor has to perform to reach the expected goal. Each business activity between the system and the actor must correspond to a **system operation** instance given with actual parameter values.

Each process has to be documented using the following textual description template [2] **BUT its content must be as low level as possible with actual values**:

---

**Procedure:** UgSecurelyUseSystem
**Scope:** System
**Primary Actor**: Authenticated
**Goal:** the Administrator's goal is to follow an identification procedure to be allowed to add or delete thenecessary crisis coordinators that will be granted the responsibility to handle alerts and crisis.
**Level**: User-Goal level
**Main Success Scenario** :

1. *Authenticated* executes the oeLogin use case
2. *Authenticated* executes the oeLogout use case

---

**Remark-**Processes presentation: processes should be introduced to the reader in a pedagogical manner. Thus, simple and common processes should be presented before than more complex and less utilised ones.
**Remark-**Graphical User Interfaces (GUIs): include GUIs screenshots to show the different stages of the process while its is performed by the actor(s).

## 3.1 Multi-procedures

### *3.1.1 Deploy And Run*

---

**Procedure:** SuDeployAndRun
**Scope:** System
**Primary Actor**: Administrator

**Secondary Actor(s)**: MrsCreator,
Coordinator,
Activator,
ComCompany
**Goal:** The goal is to install the iCrash system on its infrastructure and to exploit its capacities related to thesecure administration and efficient handling of car crash situations depending on alerts received.
**Level**: Summary level
**Main Success Scenario** :
1. *MrsCreator* executes the oeCreateSystemAndEnvironment use case
2. *Administrator* executes the ugAdministrateTheSystem use case
3. *ComCompany* executes the oeAlert use case
4. *Activator* executes the oeSetClock use case
5. *Activator* executes the oeSollicitateCrisisHandling use case
6. *Coordinator* executes the oeGlobalCrisisHandling use case

**Extensions** :
2.a None internal worker can execute the mission
2.a.1 *CMS* sends a request for an external resource to the *ERS* actor instance
2.a.2 *ERS* informs *CMS* that the request can be processed
2.a.3 *ERS* informs *CMS* that *Bob* can now be selected as first aid worker
**procedure continues at step 3**

---

## 3.2 Mono-procedures

Mono-procedures must be grouped by actors.

### *3.2.1 Coordinator*

#### 3.2.1.1 Global crisis handling

---

**Procedure:** SuGlobalCrisisHandling
**Scope:** System
**Primary Actor**: Coordinator
**Goal:** The Coordinator's goal is to monitor the alerts receivedand the corresponding crisis in order to act as necessary to handle the crisis.
**Level**: Summary level
**Main Success Scenario** :
1. *Coordinator* executes the ugSecurelyUseSystem use case
2. *Coordinator* executes the ugMonitor use case
3. *Coordinator* executes the ugManageCrisis use case

---

#### 3.2.1.2 Manage Crisis

---

**Procedure:** UgManageCrisis
**Scope:** System
**Primary Actor**: Coordinator
**Goal:** The goal is to do an action that makes the handling of a crisis or an alert progress.
**Level**: User-Goal level
**Main Success Scenario** :
1. *Coordinator* executes the oeValidateAlert use case
2. *Coordinator* executes the oeSetCrisisStatus use case
3. *Coordinator* executes the oeSetCrisisHandler use case

4. *Coordinator* executes the oeReportOnCrisis use case
5. *Coordinator* executes the oeCloseCrisis use case
6. *Coordinator* executes the oeInvalidateAlert use case

---

#### 3.2.1.3 Monitor

---

**Procedure:** UgMonitor
**Scope:** System
**Primary Actor**: Coordinator
**Goal:** The Coordinator's goal is to get the detailed list of existing crisis or alerts to decide on next actionsto undertake
**Level**: User-Goal level
**Main Success Scenario** :

1. *Coordinator* executes the oeGetAlertsSett use case
2. *Coordinator* executes the oeGetCrisisSet use case

---

### *3.2.2 Administrator*

#### 3.2.2.1 Administrate The System

---

**Procedure:** UgAdministrateTheSystem
**Scope:** System
**Primary Actor**: Administrator
**Goal:** Administrator's goal is to follow an identification procedure to be allowed to add or delete thenecessary crisis coordinators that will be granted the responsibility to handle alerts and crisis.
**Level**: User-Goal level
**Main Success Scenario** :

1. *Administrator* executes the ugSecurelyUseSystem use case
2. *Administrator* executes the oeAddCoordinator use case
3. *Administrator* executes the oeDeleteCoordinator use case

---

#### 3.2.2.2 MyProcedure2MyActor2

. . .

# Chapter 4
# Software operations

Explain each allowed software operations (i.e. an atomic unit of treatment, a service, a functionality) including a brief description of the operation, required parameters, optional parameters, default options, required steps to trigger the operation, assumptions upon request of the operation and expected results of executing such operation. Describe how to recognise that the operation has successfully been executed or abnormally terminated. The template given below (i.e. section 4.1 has to be used).

Group the operations devoted to the needs of specific actors. Common operations to several actors may be grouped and presented once to avoid redundancy.

## 4.1 MyOperation

The system operator creates and adds a new crisis to the system after being informed by a third party (citizen, organization) and selects a crisis handler for the crisis.

**Parameters:** Reporter Personal Information, Crisis Information, Crisis Handler
**Precondition:** The system operator is logged in and has received information from a reporter.
**Post-condition:** A new crisis has been added to the system and the new crisis has been assigned to a crisis handler, the Handler has received an automatic notification from the system.
**Output messages:** The selected Crisis Handler will be notified automatically once the crisis has been created.
**Triggering:**

1. From within the crisis management window fill out the required entries related to the personal information of the reporter such as name and phone number.
2. Fill out the entries related to the crisis type, impacted area, priority, description, GPS coordinates, address and finally choose a Crisis Handler from the combo box.
3. Click on the "Submit" button in and add the entry to the database.

### *4.1.1 MyExample1*

Examples should illustrate the use of **complex operations**.

Each example must show how the actor uses the software operation under description to achieve (at least one of) its expected outcome.

It might be required to include GUI screenshots to illustrate the example.

# Chapter 5
# Error messages and problem resolutions

All known problems in using the software should be listed and explained in details using the structure presented below.

Contact information for reporting any problems (either with the software or this document) should be clearly indicated

## 5.1 Error message 1

### *5.1.1 Problem identification*

A description explaining the meaning of the faced problem.

### *5.1.2 Probable cause*

A description explaining the reasons why such a problem has been raised.

### *5.1.3 Corrective actions*

Describe the required steps the actor should take to recover from such situation.

# Appendix A
# Title of the appendix 1

Here you write the context of the appendix, structuring such content in sections, sub-sections and sub-sub-sections, if needed.

An example of appendix is the flat presentation of all the graphical user interface screens. Each screen can be presented (identification symbol and description) and screens transition graph can be given.

## A.1 My Section

Description of the section.

### *A.1.1 My subSection*

#### A.1.1.1 My subSubSection

## References

1. IEEE: IEEE Standard for Software User Documentation. IEEE Std 1063-2001 (Dec 2001) 1–24
2. Armour, F., Miller, G.: Advanced Use Case Modeling: Software Systems. Addison-Wesley (2001)

## References

1. IEEE: IEEE Standard for Software User Documentation. IEEE Std 1063-2001 (Dec 2001) 1–24
2. Armour, F., Miller, G.: Advanced Use Case Modeling: Software Systems. Addison-Wesley (2001)