

Nikita Kurylenko  
Kuzma Leshakov  
Maria Kocherga

Innopolis University

iCrashVariantG06:

Crisis Management System  
Messip User Manual  
- v 1.0.3 -

Based on IEEE Std 1063-2001 [\[1\]](#)

Friday 3<sup>rd</sup> February, 2017 - 20:01

# Contents

<b>1</b>	<b>Product information</b>	<b>5</b>
1.1	Identification	5
1.2	Copyright	5
1.3	Trademark notices	5
1.4	Restrictions	5
1.5	Warranties	5
1.6	Contractual obligations	6
1.7	Disclaimers	6
1.8	Contact	6
<b>2</b>	<b>Introduction</b>	<b>7</b>
2.1	Scope	7
2.2	Purpose	7
2.3	Intended audience	8
2.4	InnoSystem v 1.0	8
2.4.1	Actors & Functionalities	8
2.4.2	Operating environment	8
2.5	Document structure	9
<b>3</b>	<b>Usage Guide</b>	<b>11</b>
3.1	Multi-procedures	11
3.1.1	Deploy And Run	11
3.2	Mono-procedures	12
3.2.1	Coordinator	12
3.2.2	Administrator	13
<b>4</b>	<b>Software operations</b>	<b>15</b>
4.1	MyOperation	15
4.1.1	MyExample1	15
<b>5</b>	<b>Error messages and problem resolutions</b>	<b>17</b>
5.1	Error message 1	17
5.1.1	Problem identification	17
5.1.2	Corrective actions	17
5.2	Error message 2	17
5.2.1	Problem identification	17
5.2.2	Probable cause	17
5.2.3	Corrective actions	18
5.3	Error message 3	18
5.3.1	Problem identification	18
5.3.2	Probable cause	19
5.3.3	Corrective actions	19
5.4	Error message 4	19
5.4.1	Problem identification	19
5.4.2	Probable cause	19
5.4.3	Corrective actions	20

CONTENTS	3
5.5 Error message 5	20
5.5.1 Problem identification	20
5.5.2 Probable cause	20
5.5.3 Corrective actions	20
<b>A Title of the appendix 1</b>	<b>21</b>
A.1 My Section	21
A.1.1 My subSection	21
<b>Glossary</b>	<b>23</b>
References	25
References	25

# List of Figures

5.1	Error message - problems with log-in . . . . .	18
5.2	Error message - problems with GUI . . . . .	18
5.3	Error message - solution to GUI . . . . .	19

# 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

Copyright iCrash LLC 2017. All rights reserved.

iCrash is protected by copyright law and international treaties. The software is licensed (not sold), and its use is subject to a License Agreement. Unauthorized use, copying or distribution of the program or any of its components may result in severe criminal or civil penalties, and will be prosecuted to the maximum extent under the law. All trademarks and registered trademarks used in this documentation are property of their owners:

Java SE Development Kit (JDK), MySQL Server, MySQL Workbench are trademarks or registered trademarks of Oracle Corporation.

TeX Live are trademarks or registered trademarks of TeX user groups.

xindy is a trademark or registered trademark of Joachim Schrod.

ps2pdf is a trademark or registered trademark of Artifex Software, Inc.

Scene Builder is trademark or registered trademark of Gluon Labs.

StarUML is a trademark or registered trademark of MKLab, Co.

Vagrant is a trademark or registered trademark of HashiCorp.

### 1.3 Trademark notices

iCrash and all iCrash taglines are trademarks or registered trademarks of iCrash LLC in the R.F. and/or other countries. All other third-party product and company names may be trademarks of their respective owners

### 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

The intention is that this document will provide a comprehensive guide, enabling users to use iCrash to its full extent. It is in three parts.

Usage Guide which is aimed at describing the general use of the software since it is deployed, configured and run. Special attention is paid to actors which rely on the software to perform a set of business activities (procedures) aimed at reaching a particular goal. Procedures are split into two groups: multi-procedures and mono-procedures.

Software Operations which explain allowed software operations using parameters, pre- and post-conditions, output messages, triggering.

Error messages and problem resolutions which lists and explains in detail all problems that might have arisen while using the software.

This document does not provide details about relationships with software stakeholders as it is described in additional software licence agreement. It also contains limited information of copyright and trademark notices.

This document is not intended to be final as updates and corrections are planned to be added later.

This document may be used with iCrash Design Document, deployment iCrash version, development iCrash version.

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 InnoSystem v 1.0

The iCrash system belongs to the Crisis Management Systems Domain. It is a system dedicated to crisis professional and non professional end users. It has to be considered as an autonomous and external service for the society. It is not an institutional system certified and guaranteed by any governmental entity and thus, must be used with caution.

### 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 a phone 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

The iCrash application is supposed to be deployed over at least 3 different computers: 1 database server, 1 application server, and at least 1 client.

However in order to ease deployment process deployment via Vagrant is possible. It requires single PC to deploy and run iCrash.



Vagrant (author rights described in Product Information - Copyright section) is an open-source software product for building and maintaining portable virtual development environments. It is used to demonstrate the application working using users machine, servers application and database machine.

Another application required for successful deployment of iCrash is VirtualBox (ver. 5.1.x).

## 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 procedures are split 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 the necessary 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 these secure 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 ugAdministratedTheSystem use case
3. *ComCompany* executes the oeAlert use case
4. *Activator* executes the oeSetClock use case
5. *Activator* executes the oeSolicitCrisisHandling 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 received and 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 actions to undertake

**Level:** User-Goal level

**Main Success Scenario :**

1. *Coordinator* executes the oeGetAlertsSet 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 the necessary 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

### 5.1 Error message 1

#### *5.1.1 Problem identification*

When I try to log in through the iCrash admin system, it doesn't accept that which is written in the readme file.

#### *5.1.2 Corrective actions*

1. The simplest way to know if all the VM are up and running is looking at the messages that appear after you type `vagrant up`
2. Other thing you could try is to type in your console: `vagrant box list` .It shows the different vagrant boxes already downloaded in your machine.
3. Also in case of using Vagrant project you need to check if all virtual machines were started correctly. In particular you should login into the server VM and see if the server has been started: you can do it by typing: `ps -ax | grep java` which displays all Java processes running on the machine.

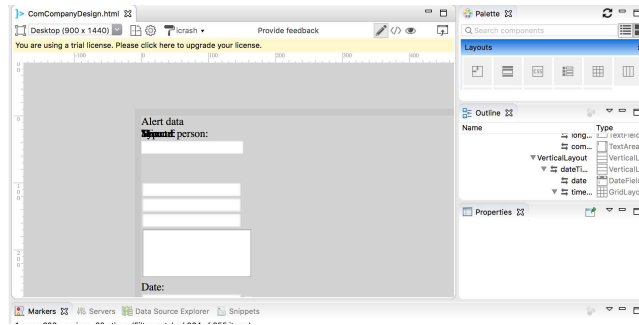
### 5.2 Error message 2

#### *5.2.1 Problem identification*

I made iCrash project changes in order to implement the variants. Before changing I could start the project but after modifications, an error is displayed. I think maybe this is an error in the server.

#### *5.2.2 Probable cause*

This problem may happen because you made a copy of the iCrash project into the same workspace (let's say you decided to called iCrash2), but you did not change the Web Project settings configuration



**Fig. 5.1** Error message - problems with log-in

### 5.2.3 Corrective actions

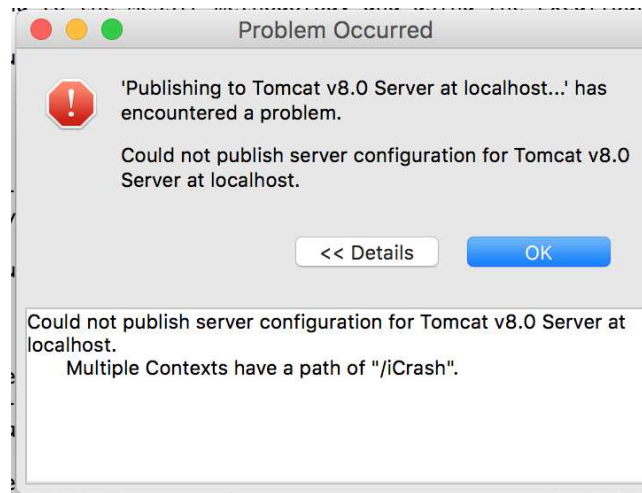
Right-click on the project->properties, then select Web Project Settings. You need to change the context root to a name other than iCrash (maybe iCrash2).

In case you still have problems to run the web application, have a look into the file server.xml, placed inside the Servers project, into the folder Tomcat v8.0 Server. Search for the entries "Context" and ensure that the property path points to the right context for each entry appearing into the file.

## 5.3 Error message 3

### 5.3.1 Problem identification

I would like to modify a class in iCrash H5, whose GUI was created with Vaadin Designer, but it looks broken and is unmodifiable. What could I do?



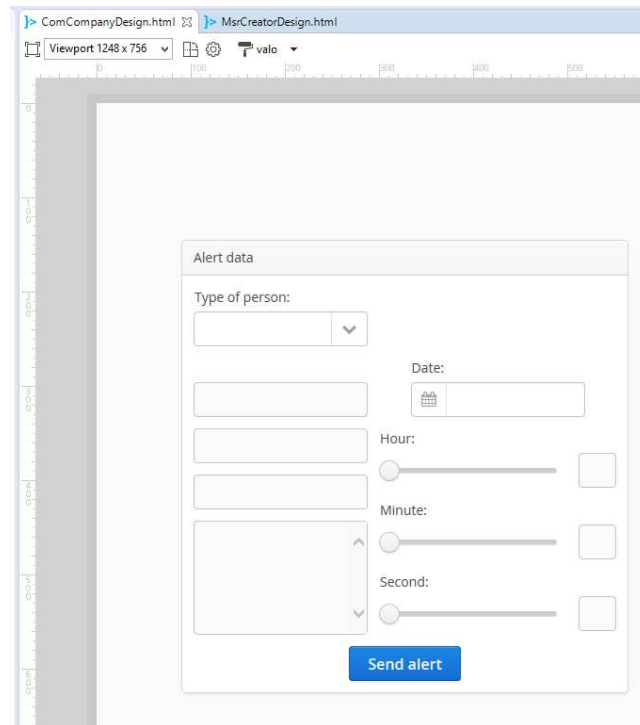
**Fig. 5.2** Error message - problems with GUI

### 5.3.2 Probable cause

GUI might not be supported by your drivers.

### 5.3.3 Corrective actions

Method 1: You need to change ComCompanyDesign's theme from icrash to valo. It then solves the problem:



**Fig. 5.3** Error message - solution to GUI

Method 2:

Unzip 1.zip to project's WebContent VAADIN-themes-icrash

Clean the project (build automatically should be on) and reopen ComCompanyDesign. You will then get the same result with icrash theme - the design will become editable.

## 5.4 Error message 4

### 5.4.1 Problem identification

When compiling a message indicates that there is a file missing.

### 5.4.2 Probable cause

Issues with correct PATH.

### ***5.4.3 Corrective actions***

You need to ensure that:

- 1) you are using Excalibur v1.5.1
- 2) you have the latest version of the Excalibur Standard Libraries setup in your environment, If not sure, then remove the projects from the workspace, and then add them again.
- 3) you have the latest released iCrash Specification in your workspace. If not sure, then remove the project from the workspace and add it again.

Once you have ensured the previous steps, check if the problem remains.

## **5.5 Error message 5**

### ***5.5.1 Problem identification***

Is it normal that all the views (even the ones that come with the icrash specification project) and their documentation are missing? Is there a way to recover the views that come with the project and their captions?

### ***5.5.2 Probable cause***

Problems with displaying data.

### ***5.5.3 Corrective actions***

No it is normal that all the original views of the icrash specification project are missing and their documentation are missing.

1. Close Eclipse
2. You can copy/paste the following files from the original icrash project to your icrash variant project.
  - 2.a the representation.aird file
  - 2.b all msrd files in the folder lu.uni.lassy.excalibur.examples.icrash/views"
- 3 Launch Eclipse

# 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



# Glossary

Actor	An actor is a person, organization, or external system that plays a role in one or more interactions with the system. . . . .	8
Glossary	the description of terms that are likely unfamiliar to the audience. The glossary shall include an alphabetical list of terms and definitions. Documentation using abbreviations and acronyms unfamiliar to the audience shall include a list with definitions, which may be integrated with the glossary. Terms included in the glossary should also be defined on their first appearance in printed documentation. Here there is an example of how to include an expression into the glossary: <a href="#">Societics</a> . . . . .	5
Societics	Represents the fields of hardware/software systems used for the society extension. . . . .	23





## 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)