Lim Sergei

Issuing Innopolis University Name

MySystem (v1.0)

Messir User Manual
- v 1.0.0 -

Based on IEEE Std 1063-2001 [1]

Friday 28th April, 2017 - 00:21

Springer

Contents

List of Figures

Chapter 1
Product information

1.1 Identification

This paper contains user manual for iCrash (v1.0) road crises handling system. The System includes:

- Client application, that provides GUI for actors of the system.
- Server application that is a centralized entity that processes requests from the clients, keeps connection with the database and is responsible for data consistency. The business logic of the whole system is located here.
- Database server that is a persistent storage for storing messages as a data.

1.2 Copyright

1.3 Trademark notices

"iCrash ", "iCrash client", "iCrash server", "iCrash database" and "iCrash common" are trademarks of University of Luxembourg.

1.4 Restrictions

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

1.5 Warranties

There is no warranty at all.
This warranty is not valid if system or hardware requirements are not compatible with this product. This warranty is not valid if any of iCrash parts is manually changed.

1.6 Contractual obligations

The functionalities provided by iCrash are designed only to provide an educational software for the Software Engineering and Development course. This software is not meant to be used for any other purposes.

1.7 Disclaimers

1.8 Contact

university@innopolis.ru

Chapter 2

Introduction

2.1 Scope

This document provides the information that is required to use iCrash the key components

This document does not describe details of implementation

This document does not provide information of all use cases, but provides all necessary information about installation and use the iCrash

This document may be used with other documents of the iCrash (v1.0) system, as well as with the documents provided by third-party companies to have an overall view and correct understanding of the system, how to use and employ it and its benefits.

2.2 Purpose

The purpose of this document is to provide basic examples of using iCrash client module.

This document defines the screenshots of iCrash client module and key terms for understanding iCrash client module interface and functionalities.

This document is meant to be used along with the system at any point of time of its employment.

2.3 Intended audience

Intended audience of the documentation are the system actors who actively exploit main functionalities of the iCrash(v1.0), including Administrator, Coordinator and Editor.

System actors are expected to refer to the documentation for assistance regarding any system usage, error messages resolutions and system operation-related issues.

2.4 *MySystem (v1.0)*

TODO

2.4.1 Actors & Functionalities

Administrator - A person who can add new coordinator in the system and in its environment or delete an existing one.
Editor - A person who works with media and chooses which news should be published.
Coordinator - A person who informs about crises
Witness - human that is a witness of an accident
Victim - human that is a victim of an accident

2.4.2 Operating environment

- Server Application is the main part of the infrastructure, and is responsible for overall business logic of the system and consistency of the data and request handling received from Client Application and Database Server.
- Client Application provides Graphical User Interface for system actors and sends requests to process to Server Application.
- Crash requests are processed by the Server Application and corresponding crash report data is then stored in Database Server.

2.5 Document structure

TODO 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: ProcessMissionOne
Scope: Crisis Management System (*CMS*)
Primary Actor: Coordinator John
Secondary Actor(s): FirstAidWorker Bob,
ExternalResourceSystem ERS
Goal: The intention of the Coordinator is to process mission with ID equal to 1.
Level: User-goal level
Main Success Scenario :

1. *John* instructs the *CMS* to process the mission with ID equal to 12.031005
2. *CMS* selects the internal worker *Bob* to execute the mission 12.031005
3. *CMS* instructs *Bob* to behave as *First Aid Worker (FAW)*
4. *Bob* informs the *CMS* of his arrival
5. *Bob* informs the *CMS* that he starts to execute the mission 12.031005
6. *Bob* informs the *CMS* that the mission 12.031005 outcome is “Mission completed”

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

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 MyMultiProcedure1

...

3.1.2 MyMultiProcedure2

...

3.1.3 MyMultiProcedure3

...

3.2 Mono-procedures

Mono-procedures must be grouped by actors.

3.2.1 MyActor1

3.2.1.1 MyProcedure1MyActor1

...

3.2.1.2 MyProcedure2MyActor1

...

3.2.2 My-Actor2

3.2.2.1 MyProcedure1MyActor2

...

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)