Rafael Valiev
Gregory Yermolaev
Kasatkin Timur

Innopolis University

myCrashVariant

Messir User Manual
- v 1.0 -

Based on IEEE Std 1063-2001 [1]

Saturday 4th February, 2017 - 23:43

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

1.6 Contractual obligations

1.7 Disclaimers

1.8 Contact

Information for contacting the issuing organization.

Chapter 2
Introduction

2.1 Scope

This section has to provide the scope of the user's manual document. In the following some opening statements to use when providing the information corresponding to this section.

This document provides minimum acceptable information for knowing how to use the software system ***myCrashVariant*** .

This document does not provide all details regarding ***myCrashVariant*** .

This document is not intended to provide information about how to connect, deploy, configure, or use any external device or third-party software system that is required for the correct functioning of ***myCrashVariant*** .

This document may be used with other documents provided by third-party companies to have an overall view and correct understanding of the environment and procedures where the software system ***myCrashVariant*** is aimed to be deployed and run.

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.

The purpose of this document is to give some minimum amount of information required for using the system.

This document defines such models as Environment Model, Concept Model, etc. It also defines several views such as Deployment View, Implementation View, etc.

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 *myCrashVariant*

Brief overview of the software application domain and main purpose.

2.4.1 Actors & Functionalities

2.4.1.1 Communication Company

A company that has the capacity to ensure communication of information between its customers and the *myCrashVariant* system.

Functionalities:

- deliver SMS about possible crisis got from witness or victim to the *myCrashVariant* 's phone number in form of alert
- be notified when handling alert by Coordinator of *myCrashVariant* system, sent by the Communication Company previously, is in progress
- transmit SMS messages from company that owns *myCrashVariant* system to any human having an SMS compatible device accessible using a phone number

2.4.1.2 Human

Any person who considers himself related to a car crash either as a witness, a victim or an anonymous person.

Functionalities:

- inform the *myCrashVariant* system about the crisis situation he detected by sending SMS with information about the crisis to some Communication Company which will send an alert to *myCrashVariant* system
- be notified that the ABC company has been informed about the situation
- be informed about the situation of the crisis he/she has related to as a victim or witness
- get QA survey (set of satisfaction single choice text questions with possible mark answer for each from 0 to 5) by SMS message when a crisis, associated with this Human, processed by *myCrashVariant* 's Coordinator. QA survey will be open (*myCrashVariant* system will wait answer from Human) for 5 hours.
- answer obtained QA survey by sending reply in form of corresponding mark-answers for each question separated by spaces

2.4.1.3 Coordinator

An employee of the company, owning *myCrashVariant* system, being responsible of handling one or several crisis.

Functionalities:

- authenticate in *myCrashVariant* system by providing valid (present in the system) login/password and, if login/password pair is valid, pressing on fingerprint scanner with his finger in special mobile app in the following 60 seconds
- observe alerts by their status (pending/valid/invalid)
- observe crises by their status (pending/handled/solved/closed)
- validate/invalidate pending alert (mark alert as real or not)
- handle crisis
- leave free text comment on crisis
- change crisis's status
- close crisis (make crisis's status equal to 'closed')

2.4.1.4 Administrator

An employee of the company, owning *myCrashVariant* system, being responsible of administrating the system.

Functionalities:

- authenticate in *myCrashVariant* system by providing valid (present in the system) login/password and, if login/password pair is valid, pressing on fingerprint scanner with his finger in special mobile app in the following 60 seconds

- add or delete coordinators from the system and its environment
- be notified about intrusions (when some person tried to authenticate as an Coordinator, but provided invalid (not stored in ***myCrashVariant*** system) login/password pair or his fingerprint scan didn't match one associated with corresponding (defined by login/password pair) coordinator)
- visualize satisfaction diagrams in form of eye chart showing relative amounts of each type of mark of for given QA survey question. The administrator can see such chart of question for all crises or for only specified one.

2.4.1.5 Creator

A technician who is installing the ***myCrashVariant*** system on the targeted deployment infrastructure.

Functionalities:

- install the ***myCrashVariant*** system
- define the values for the initial system's state
- define the values for the initial system's environment

2.4.1.6 Activator

A logical representation of the active part the ***myCrashVariant*** system. It represents an implicit stakeholder belonging to the system's environment that interacts with the iCrash system autonomously without the need of a external entity. It is usually used for representing time triggered functionalities.

Functionalities:

- communicate the current time to the system
- notify the administrator that some crisis are still pending for a too long time

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.

3.1 Multi-procedures

3.1.1 DeployAndRun

Procedure: DeployAndRun
Scope: Crisis Management System (*CMS*)
Primary Actor: Administrator Bill
Secondary Actor(s): Creator theCreator,
Coordinator Steve,
Activator theClock,
Communication Company Tango
Goal: The goal is to install the ***myCrashVariant*** system on its infrastructure and to exploit its capacities related to the secure administration and efficient handling of car crash situations depending on alerts received.
Level: Summary level
Main Success Scenario :

1. *theCreator* instructs *CMS* to create the system and environment indicating that the number of communication company actor instances for system's environment is 4 (one of them is identified here by *Tango*)
2. *theClock* sets *CMS*'s clock to '2017:11:24 - 03:20:00'
3. *Bill* follows identification procedure to be allowed to add or delete the necessary crisis coordinators (one of them is identified here by *Steve*) by first providing login 'icrashadmin' and password '7WXC1359' to *CMS*
4. *CMS* approves login/password pair provided by *Bill* and asks him to provide fingerprint scan of his finger using special mobile app in the following 60 seconds
5. *Bill* opens special mobile app, scans his finger with the special finger scanning device and app sends data about *Bill*'s fingerprint to the *CMS*
6. *CMS* approves obtained fingerprint scan and shows message "You are logged! Welcome . . . " to *Bill*
7. *Bill* instructs *CMS* to add coordinator with id "1", login "steve" and password "pwdMessirExcalibur2017"
8. If *Steve* has not provided his fingerprint scan yet to the *CMS*, *Bill* asks *Steve* to provide his fingerprint scan information to save it in *CMS* for future authorization
9. *CMS* shows authorization window for coordinator *Steve*
10. *Bill* informs *CMS* that he wants to logout
11. *CMS* informs *Bill* that he has logouted by showing message "You are logged out! Good Bye!" to him

12. *theClock* sets *CMS*'s clock to '2017:11:26 - 10:15:00'
13. *Tango* informs *CMS* about alert with human kind "witness", date "2017:11:26", time "10:10:16", phone number "+3524666445252", latitude "49.627675", longtitude "6.159590" and comment "3 cars involved in an accident"
14. *CMS* saves obtained alert with ID 1 and sends SMS message "+3524666445252, Your alert has been registered. We will handle it and keep you informed" to *Tango*, which dispatches it to it's customer with corresponding phone number
15. *theClock* sets *CMS*'s clock to '2017:11:26 - 10:30:00'
16. *theClock* instructs *CMS* to sollicitate crisis handling
17. *CMS* notifies *Bill* about pending alert by showing message "There are alerts pending since more than defined delay. Please REACT!" to him
18. *CMS* notifies *Steve* about pending alert by showing message "There are alerts pending since more than defined delay. Please REACT!" to him
19. *Steve* follows identification procedure by first providing login "steve" and password "pwdMessirExcalibur2017" to *CMS*
20. *CMS* approves obtained login/password pair and asks *Steve* to provide fingerprint scan of his finger using special mobile app in the following 60 seconds
21. *Steve* opens special mobile app, scans his finger with the special finger scanning device and app sends data about *Steve*'s fingerprint to the *CMS*
22. *CMS* approves obtained fingerprint scan and shows message "You are logged! Welcome . . . " to *Steve*
23. *Steve* asks *CMS* to list for him crises this status "pending"
24. *CMS* sends a crisis with ID 1 details to *Steve*
25. *Steve* instructs *CMS* to declare himself as the handler of a crisis having ID 1
26. *CMS* sends SMS message "+3524666445252, The handling of your alert by our services is in progress" to *Tango*, which dispatches it to it's customer with corresponding phone number
27. *CMS* shows message "You are now considered as handling the crisis!" to *Steve*
28. *theClock* sets *CMS*'s clock to "2017:11:26 - 10:45:00"
29. *Steve* instructs *CMS* to set status of alert with ID 1 as a "valid"
30. *CMS* sets status of alert with ID 1 to "valid" and shows message "The Alert is now declared as valid!" to *Steve*
31. *Tango* informs *CMS* about alert with human kind "witness", date "2017:11:26", time "10:20:00", phone number "+3524666445000", latitude "49.627095", longtitude "6.160251" and comment "A car crash just happened."
32. *CMS* sends SMS message "+3524666445000, Your alert has been registered. We will handle it and keep you informed" to *Tango*, which dispatches it to it's customer with corresponding phone number
33. *theClock* sets *CMS*'s clock to "2017:11:26 - 12:45:00"
34. *Steve* instructs *CMS* to set status of crisis with ID 1 to "solved"
35. *CMS* changes status of crisis with ID 1 and shows message "The crisis status has been updated!" to *Steve*
36. *Steve* instructs *CMS* to update textual information (comment) available to crisis with ID 1 to "3 victims sent to hospital, 2 cars evacuated and 4 rescure unit mobilized"
37. *CMS* updates comment on crisis with ID 1 and shows message "The crisis comment has been updated!" to *Steve*
38. *Steve* instructs *CMS* to close (change status to "closed") crisis with ID 1
39. *CMS* closes crisis with ID 1 and shows message "The crisis is now closed!" to *Steve*
40. *CMS* opens QA survey for crisis with ID 1 for 5 hours by sending SMS message "+3524666445252, your alert has been processed. To let us know about your satisfaction about handling your alert, please answer the two following questions: 1) How quick your alert was handled? 2) Did actions performed by our coordinator help in resolving the situation? We expect you to answer these questions in 5 hours by sending us mark answers (numbers from 0 to 5 each) to each question, separated by spaces (e.g. '4 5')." to *Tango* which dispatches it to customer with corresponding number.
41. *Tango* informs *CMS* about answer to QA survey with sender phone number "+3524666445252", reply "5 4", date '2017:11:26' and time '13:00:00'
42. *CMS* saves information of obtained reply to QA survey from person with phone number "+3524666445252"

3.2 Mono-procedures

Mono-procedures must be grouped by actors.

3.2.1 Coordinator

3.2.1.1 Monitor

Procedure: Monitor
Scope: Crisis Management System (*CMS*)
Primary Actor: Coordinator Steve
Goal: Steve's goal is to get the detailed list of pending alerts and crises to decide on next actions to undertake.
Level: User-goal level

Main Success Scenario :

1. *Steve* instructs the *CMS* to give him detailed list of pending alerts
2. *CMS* sends *Steve* alerts with status 'pending'
3. *Steve* instructs the *CMS* to give him detailed list of pending crises
4. *CMS* sends *Steve* crises with status 'pending'

3.2.1.2 ManageCrisis

Procedure: ManageCrisis
Scope: Crisis Management System (*CMS*)
Primary Actor: Coordinator Steve
Goal: The goal is to do an action that makes the handling of alert with ID 1 and crisis with ID 1 progress, and mark alert with ID 2 as invalid.
Level: User-goal level
Main Success Scenario :

1. *Steve* instructs the *CMS* to mark alert with ID 1 as 'valid'
2. *CMS* informs *Steve* that it has marked alert with ID 1 as 'valid' by showing message “The alert is now declared as valid" to him
3. *Steve* instructs *CMS* to declare himself as the handler of a crisis having ID 1
4. *CMS* shows message “You are now considered as handling the crisis!” to *Steve*
5. to deal with the crisis, *Steve* calls ambulance, tow truck and rescue units to crisis' location (these actions performed outside the *CMS*)
6. *Steve* instructs the *CMS* to mark crisis with ID 1 as 'solved'
7. *CMS* updates status of crisis with ID 1 and notifies *Steve* about it by showing message “The crisis status has been updated!" to him
8. *Steve* instructs *CMS* to update textual information (comment) available to crisis with ID 1 to “3 victims sent to hospital, 2 cars evacuated and 4 rescue unit mobilized"
9. *CMS* updates comment on crisis with ID 1 and shows message “The crisis comment has been updated!” to *Steve*
10. *Steve* instructs *CMS* to close (change status to “closed”) crisis with ID 1
11. *CMS* closes crisis with ID 1 and shows message “The crisis is now closed!” to *Steve*
12. *Steve* instructs the *CMS* to mark alert with ID 2 as 'invalid'
13. *CMS* informs *Steve* that it has set status of alert with ID 2 to 'invalid' by showing message “The alert is now declared as invalid!" to him

---

3.2.1.3 GlobalCrisisHandling

Procedure: GlobalCrisisHandling
Scope: Crisis Management System (*CMS*)
Primary Actor: Coordinator Steve
Goal: Steve's goal is to authenticate in system and 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. *Steve* follows identification procedure by first providing login “steve” and password “pwdMessirExcalibur2017” to *CMS*
2. *CMS* approves obtained login/password pair and asks *Steve* to provide fingerprint scan of his finger using special mobile app in the following 60 seconds
3. *Steve* opens special mobile app, scans his finger with the special finger scanning device and app sends data about *Steve*'s fingerprint to the *CMS*
4. *CMS* approves obtained fingerprint scan and shows message “You are logged! Welcome . . . ” to *Steve*
5. *Steve* instructs the *CMS* to give him detailed list of pending alerts
6. *CMS* sends *Steve* alerts with status 'pending'
7. *Steve* instructs the *CMS* to give him detailed list of pending crises
8. *CMS* sends *Steve* crises with status 'pending'
9. *Steve* instructs the *CMS* to mark alert with ID 1 as 'valid'
10. *CMS* informs *Steve* that it has marked alert with ID 1 as 'valid' by showing message “The alert is now declared as valid" to him
11. *Steve* instructs *CMS* to declare himself as the handler of a crisis having ID 1
12. *CMS* shows message “You are now considered as handling the crisis!” to *Steve*
13. to deal with the crisis, *Steve* calls ambulance, tow truck and rescue units to crisis' location (these actions performed outside the *CMS*)
14. *Steve* instructs the *CMS* to mark crisis with ID 1 as 'solved'
15. *CMS* updates status of crisis with ID 1 and notifies *Steve* about it by showing message “The crisis status has been

updated!" to him
16. *Steve* instructs *CMS* to update textual information (comment) available to crisis with ID 1 to “3 victims sent to hospital, 2 cars evacuated and 4 rescure unit mobilized"
17. *CMS* updates comment on crisis with ID 1 and shows message “The crisis comment has been updated!” to *Steve*
18. *Steve* instructs *CMS* to close (change status to “closed”) crisis with ID 1
19. *CMS* closes crisis with ID 1 and shows message “The crisis is now closed!” to *Steve*
20. *Steve* informs the *CMS* that he wants to logout
21. *CMS* informs *Steve* that he has logouted by showing message “You are logged out! Good Bye!” to him

---

3.2.2 Administrator

3.2.2.1 SecurelyUseSystem

Procedure: SecurelyUseSystem
Scope: Crisis Management System (*CMS*)
Primary Actor: Administrator Bill
Goal: Bill’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. *Bill* informs *CMS* that he wants to authenticate using login ’icrashadmin’ and password ’7WXC1359’
2. *CMS* approves login/password pair provided by *Bill* and asks him to provide fingerprint scan of his finger using special mobile app in the following 60 seconds
3. *Bill* opens special mobile app, scans his finger with the special finger scanning device and app sends data about *Bill*’s fingerprint to the *CMS*
4. *CMS* approves obtained fingerprint scan and shows message “You are logged! Welcome . . . ” to *Bill*
5. *Bill* informs the *CMS* that he wants to logout
6. *CMS* informs *Bill* that he has logouted by showing message “You are logged out! Good Bye!” to him

---

3.2.2.2 AdministrateTheSystem

Procedure: AdministrateTheSystem
Scope: Crisis Management System (*CMS*)
Primary Actor: Administrator Bill
Goal: Bill’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. *Bill* follows identification procedure to be allowed to add or delete the necessary crisis coordinators (one of them is identified here by *Steve*) by first providing login ’icrashadmin’ and password ’7WXC1359’ to *CMS*
2. *CMS* approves login/password pair provided by *Bill* and asks him to provide fingerprint scan of his finger using special mobile app in the following 60 seconds
3. *Bill* opens special mobile app, scans his finger with the special finger scanning device and app sends data about *Bill*’s fingerprint to the *CMS*
4. *CMS* approves obtained fingerprint scan and shows message “You are logged! Welcome . . . ” to *Bill*
5. *Bill* instructs *CMS* to add coordinator with id “1”, login “steve” and password “pwdMessirExcalibur2017”
6. *CMS* creates new coordinator with id “1”, login “steve” and password “pwdMessirExcalibur2017”
7. *Bill* instructs *CMS* to delete coordinator with id “1”
8. *CMS* deletes coordinator with id “1”

3.2.2.3 MonitorQualityInsurance

Procedure: MonitorQualityAssurance
Scope: Crisis Management System (*CMS*)
Primary Actor: Administrator Bill
Goal: The intention of Administrator Bill is to observe relative amounts of each type of mark-answer (number from 0 to 5) for QA survey question “How quick your alert was handled?” in context of all crises and in context of crisis with ID 1
Level: User-goal level
Main Success Scenario :

1. *Bill* instructs the *CMS* to show satisfaction diagram for question “How quick your alert was handled?" without providing

ID of specific crisis
2. *CMS* shows eye chart diagram showing relative amounts of each type of mark for given question regarding all crises to *Bill*
3. *Bill* instructs the *CMS* to show satisfaction diagram for question “How quick your alert was handled?" and crisis with ID 1
4. *CMS* shows eye chart diagram showing relative amounts of each type of mark for given question regarding crisis with ID 1 to *Bill*

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-subsections, 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

Concept Model the Model that describes the different types required to specify the software system. . . . 7

Deployment View The physical view depicts the system from a system engineer's point-of-view. It is concerned with the topology of software components on the physical layer, as well as the physical connections between these components. For example, how many nodes are used and what is deployed on what node. A Deployment View is modelled as a UML Deployment Diagram. 7

Environment Model the Model that describes the different actors supposed to interact with the software system. 7

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: Societics. 5

Implementation View This view describes the software system components. It focuses on software modules and subsystems. It describes the hierarchies or layers for components. This view is modelled as a UML Component Diagram. 7

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