FIGUEIREDO Christophe
DUARTE TOME Stevie
THIELEN Daniel

Issuing GrouPSeveN

Forest Fire(v0.1)

Messir User Manual
- v 1.0.3 -

Based on IEEE Std 1063-2001 [1]

Friday 25th November, 2016 - 20:44

Springer

Contents

List of Figures

Chapter 1
Product information

1.1 Identification

The FORESTFIRE Smartphone Application should work on !!!!every!!! smartphone. The GPS must be enabled inorder to make a accurate mark on the map.
The FORESTFIRE WebApplication needs atleast Firefox 16.0, Google Chrome 5.0.375 and Internet Explorer 10 to work correctly. Javascript should also be enabled.

1.2 Copyright

Copyright ©2016 by GrouPSeveN.
All rights reserved. This user manual or any portion thereof may not be reproduced or used in any manner whatsoever without the express written permission of the publisher.

1.3 Trademark notices

FORESTFIRE is a registered tradmark of GrouPSeveN.

1.4 Restrictions

- The customer may make modifications to the source code, if and only if it is for selfuse. The customer shall not sublicense, sell or otherwise authorize the use of the software, whether in executable form, nor in source code or otherwise, by any third parties.
- In case of any modification to the software made by any party other than GrouPSeveN all the warranties drop.

1.5 Warranties

GrouPSeveN represents and warrants to Customer that it has all necessary rights and authority to execute and deliver this Software License and perform its obligations herunder and to grand the rights granted under this Software License to Customer.
GrouPSeveN futher represents and warrants that, the executable object code of software and the system will perform substantially in accordance with the System Specification and Agreement.

1.6 Contractual obligations

!!!!!!!!!!!!!!!!!!!!

1.7 Disclaimers

This documentation is under active development and as such there may be mistakes and omissions. Please report any errors you find to the developers mailing address: groupseven@g7.com.

1.8 Contact

Information for contacting the issuing organization. Please address all communication concerning FORESTFIRE and this manual to:
E-mail : groupseven@g7.com
Tel : 585858
Website : www.beto.lu

Chapter 2
Introduction

2.1 Scope

This document provides information about how to operate the ***Forest Fire(v0.1)*** . This document also provides information about error messages and problems and how you can fix them.

This document is not intended as a guide to deploy, connect or even configure the ***Forest Fire(v0.1)*** system and how to manage it.

This document may be used with additional information from out website www.beto.lu

2.2 Purpose

The purpose of this document is to make the life of the System Manager easier and to give him a feeling about how ***Forest Fire(v0.1)*** works.

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 *Forest Fire(v0.1)*

The purpose of ***Forest Fire(v0.1)*** is to improve both safety and reaction time to combat against forest fires.

2.4.1 Actors & Functionalities

Here quick overview of the users of the System and their functionalities.

2.4.1.1 Admin

The Admin manages the system. She/He can create new Users, edit existing users as well as deleting them. Of course the Admin can loginto the website.

2.4.1.2 Employee

The Employee is a person who manages with the Fire Departement Officer the alerts. She/He can loginto the website. She/He can also send information to the News as well as sending some Maintenance Team to a defect Device.

2.4.1.3 Witness

The Witness is a person who witnesses a fire and sends an alert to the system via the App.

2.4.1.4 Fire Departement Officer

The Fire Departement Officer has the command off his Fire Departement. She/He can loginto the website and manage the alerts with the Employees.

2.4.1.5 News

The News is a user who can loginto the website and can publish news about the recent confirmed fire alerts.

2.4.1.6 Sensor

The Sensor is a device which can send anomaly to the system which could indicate a forest fire. The System can also force an sensor update if a Witness reported a fire in the area.

2.4.1.7 Drone

The drone is a device which the Fire Departement Officer and the Employees can use to idenify a fire. The drone gets its mission if there is an alert.

2.4.1.8 Maintenance

The Maintenance is a person whos job is to maintenance devices. She/He can loginto the website. If there is a defect device the Maintenance is concerned about this problem. She/He can indicate a status like queued, working or finished.

2.4.2 Police

Police is the person whos job is to handle the evacuation. !!!!!

2.4.3 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

In this section you can find the procedures which the system is using.
Remark : Graphical User Interfaces (GUIs): include GUIs screenshots to show the different stages of the process while its is performed by the actor.

3.1 Actors common procedures

Here you can find procedures which are commen to some actors.

3.1.1 Login Procedure

Use Case: LoginProcedure
Scope: System (*System*)
Primary Actor: User
Secondary Actor:
Intention: The intention of the User is to log into the website.
Level: Sub-functional level
Main Success Scenario :

1. *User* sents his login information to the system.
2. *System* validates information.
3. *System* grands access.

Extensions :

3. *System* denys access.

3.1.2 Alert Send (App)

Use Case: AlertSend(App)
Scope: System (*System*)
Primary Actor: Witness
Secondary Actor: Telecom
Intention: The intention of the Wittness is to send a alert to the system.
Level: Sub-functional level
Main Success Scenario :

1. *Witness* sents fire location to the System an his phone number to the Telecom.
2. *Telecom* verifies recieved number and sends the number also to the System.

3. *System* sends confirmation to the Witness.
4. *System* proceeds with procedure <Validate Alert>.

Extensions :

1. *Witness* calls the Telecom.
2. *Telecom* locates Witness and sends information to the System.

3.2 System procedures

3.2.1 Validate Alert

Use Case: ValidateAlert
Scope: System (*System*)
Primary Actor: System
Secondary Actor: Drones,Sensor,Fire Departement Officer and Employee
Intention: The intention of the System is to validate recieved alert.
Level: Sub-functional level
Main Success Scenario :

1. *System* requests information of the sensor which is the nearest to the fire location.
2. *System* recieves information from the sensor.
3. *System* dispatches Drones.
4. *System* notifies Fire Departement Officer and Employee.
5. *Fire Departement Officer and Employee* proceed with the procedure <NoFire> or <IsFire>.

Extensions :

3.2.2 Sensor Error

Use Case: SensorError
Scope: System (*System*)
Primary Actor: System
Secondary Actor:Sensor
Intention: The intention of the System is to request a update from a Sensor.
Level: Sub-functional level
Main Success Scenario :

1. *System* requests an update from the Sensor.
2. *System* recieves no information from Sensor.
3. *System* proceeds with the procedure <Handel defect equipement>

Extensions :

3.2.3 Drone Error

Use Case: DroneError
Scope: System (*System*)
Primary Actor: System
Secondary Actor: Drone
Intention: The intention of the System is to request a status report of the drone.
Level: Sub-functional level

Main Success Scenario :

1. *System* requests a status report from the Drone.
2. *System* recieves no response from the Drone.
3. *System* proceeds with the procedure <Handle defect equipement>

Extensions :

3.2.4 Handle defect equipement

Use Case: HandleDefectEquipement
Scope: System (*System*)
Primary Actor: System
Secondary Actor: Employee,Maintenance
Intention: The intention of the System is to get the equipement repaired.
Level: Sub-functional level
Main Success Scenario :

1. *Sytsem* sends information of defect equipement to Employee.
2. *System* recieves decision of Employee.
3. *System* sends information of defect equipement to Maintenance.
4. *System* recieves status report of Maintenance.

Extensions :

3.2.5 Fault Alarm

Use Case: FaultAlarm
Scope: System (*System*)
Primary Actor: System
Secondary Actor: Drones,Fire Departement Officer and Employee
Intention: The intention of the System is to handle a fault alarm.
Level: Sub-functional level
Main Success Scenario :

1. *System* sends a "Return to Base" command to the Drones.
2. *System* recieves confirmation form the Drones.
3. *System* sends "relief-mail" to the Fire Departemernt Officer and Employee.

Extensions :

3.2.6 Achivement

Use Case: Achivement
Scope: System (*System*)
Primary Actor: System
Secondary Actor: Storage
Intention: The intention of the System is to achiv old data (Alerts,etc) on a Storage server each few days.
Level: Sub-functional level
Main Success Scenario :

1. *System* sends old data to Storage.
2. *System* recieves status code from storage(ie. success, failed).

Extensions :

3.3 Sensor procedures

3.3.1 Sensor Alert

Use Case: SensorAlert
Scope: System (*System*)
Primary Actor: Sensor
Secondary Actor:
Intention: The intention of the Sensor is to send an anomaly to the system.
Level: Sub-functional level
Main Success Scenario :

1. *Sensor* sents anomaly to System.
2. *System* proceeds with the procedure <Validate Alert>.

Extensions :

3.3.2 Send current status

Use Case: SendCurrentStatus
Scope: System (*System*)
Primary Actor: Sensor
Secondary Actor:
Intention: The intention of the sensor is to send a staus report in a time-interval to the system.
Level: Sub-functional level
Main Success Scenario :

1. *Sensor* performs a status check on itself.
2. *Sensor* sends status report to the System.
3. *System* evaluates status report.

Extensions :

3.4 Admin procedures

3.4.1 Administrative

Use Case: Administrative
Scope: System (*System*)
Primary Actor: Admin
Secondary Actor: User
Intention: The intention of the Admin is to create/edit/delete a User.
Level: Sub-functional level
Main Success Scenario :

1. *Admin* creates/edits/deletes a user using the website.
2. *System* validates entered information.
3. *Admin* recieves status (ie. created, user exits, etc). 4. *User* recieves a Email.

Extensions :

3.5 Employee

3.5.1 Send Info to News

Use Case: SendInfoToNews
Scope: System (*System*)
Primary Actor: Employee
Secondary Actor: News
Intention: The intention of the Employee is to inform the News about the fire.
Level: Sub-functional level
Main Success Scenario :

1. *Employee* sends a confirmation to the system to inform the News.
2. *System* sends the fire informations to News.
3. *Employee* recieves confirmation from the System.

Extensions :

3.6 Fire Departement and Employee procedures

3.6.1 Is Fire

Use Case: IsFire
Scope: System (*System*)
Primary Actor: Fire Departement and Employee
Secondary Actor:
Intention: The intention the Fire Departement and Employee is to confirm that there is a fire.
Level: Sub-functional level
Main Success Scenario :

1. *Fire Departement and Employee* confirm that there is a fire.
2. *Fire Departement and Employee* recieve an confirmation of the System.
3. *Fire Departement* takes action.

Extensions :

3.6.2 No Fire

Use Case: NoFire
Scope: System (*System*)
Primary Actor: Fire Departement and Employee
Secondary Actor:
Intention: The intention the Fire Departement and Employee is to confirm that there is NO fire.
Level: Sub-functional level
Main Success Scenario :

1. *Fire Departement and Employee* indicate that the alert was faulty.

2. *Fire Departement and Employee* recieve an confirmation of the System.
3. *System* proceeds with the procedure <Fault Alert>.

Extensions :

3.7 Drone

3.7.1 Send Current Location

Use Case: sendCurrentLocation
Scope: System (*System*)
Primary Actor: Drone
Secondary Actor:
Intention: The intention is that the Drone wants to send it's status report in a time interval to the system.
Level: Sub-functional level
Main Success Scenario :

1. *Drone* runs a status check of itself.
2. *Drone* sends it's status report to the System.
3. *System* evaluates the drone's status report.

Extensions :

3.7.2 Send Location

Use Case: sendLocation
Scope: System (*System*)
Primary Actor: Drone
Secondary Actor:
Intention: The intention is that the Drone wants to send it's location in a time-interval to the System
Level: Sub-functional level
Main Success Scenario :

1. *Drone* gets it's location.
2. *Drone* sends it's loction to the system.
3. *System* evaluates the information.

Extensions :

3.7.3 Routine Flight

Use Case: routineFlight
Scope: System (*System*)
Primary Actor: Drone
Secondary Actor:
Intention: The intention is that the Drone wants to execute a everyday at a specific time a routine flight.
Level: Sub-functional level
Main Success Scenario :

1. *Drone* sends a notification of departure to the system and to the Fire Station Officer.
2. *Drone* takes of and flies the configured track.

3. *Drone* activates it's cameras.
4. *Drone* returns to base.

Extensions :

3.7.4 Misson Flight

Use Case: missionFight
Scope: System (*System*)
Primary Actor: Drone
Secondary Actor:
Intention: The intention is that the drone needs to perform a misson flight.
Level: Sub-functional level
Main Success Scenario :

1. *Drone* recieves misson information
2. *Drone* send confirmation to system and to the Fire Station Officer.
3. *Drone* flies to recieved location.
4. *Drone* activates camera.
5. *Drone* cycles around the location.
6. *Drone* recieves Return to Base message.
7. *Drone* returns to base.

Extensions :

3.8 Police

3.8.1 Handle Evacuation Order

Use Case: handleEvacuationOrder
Scope: System (*System*)
Primary Actor: Police
Secondary Actor:
Intention: The intention is that Police needs to handle a Evacuation Order.
Level: Sub-functional level
Main Success Scenario :

1. *Police* recieves evacuation order and information.
2. *Police* does his job to evacuate the location.
3. *Police* informs the system that he evacuation is done.

Extensions :

3.8.2 Handle Safe Zone

Use Case: handleSafeZone
Scope: System (*System*)
Primary Actor: Police
Secondary Actor:
Intention: The intention is that Police needs to secure the area of operation.
Level: Sub-functional level

Main Success Scenario :

1. *Police* recieves misson information.
2. *Police* notifies the System that he's on a mission.
3. *Police* does his job to secure the area of operation.
4. *Police* informs system.

Extensions :

3.9 Media

3.9.1 Make News

Use Case: makeNews
Scope: System (*System*)
Primary Actor: Media
Secondary Actor: Employee
Intention: The intention is Media creates news for the website.
Level: Sub-functional level
Main Success Scenario :

1. *Media* recieves fire information from the Employee.
2. *Media* sends confirmation to the Employee.
3. *Media* creates the news based upon the fire information.
4. *Media* send News to the System.
5. *Employee* accepts the news.

Extensions :

3.9.2 Use news

Use Case: useNews
Scope: System (*System*)
Primary Actor: Media
Secondary Actor: Employee
Intention: The intention is that Media wants to use the fire information for it's own.
Level: Sub-functional level
Main Success Scenario :

1. *Media* recieves fire information from the Employee.
2. *Media* uses the information to warn people in their field (Radio, TV, etc.

Extensions :

Chapter 4
Software operations

Here you can find allowed software operations and examples on how you can use them.

4.1 Log into Website

A User of the system wants to login to the website

Parameters: username, password
Precondition: The User is not logged in.
Post-condition: The User is logged in.
Output messages: Welcome <username>.
Triggering:

1. The User presses the <Login> link in order to get redirected to the login page.
2. The User enters his username into the field <Username> and his password into the field <Password>.
3. The User presses the button <Login>.

4.2 Logout from the Website

A User from the system wants to logout from the website.

Parameters:
Precondition: The User is logged into the website.
Post-condition: The User is logged out from the website.
Output messages: Logged out... until next time.
Triggering:

1. The user presses the link <Logout>.

4.3 Create User

The admin creates a new user to the system.

Parameters: firstName, lastName, username, password, email
Precondition: The admin is logged into the website.
Post-condition: A new user has been created on system.
Output messages: User <username> has been created.
Triggering:

1. From within user management page on the website fill out the required user information related to the new user like username,status,. . .
2. Click on the Submit button to add the new user to the database.

4.4 Delete User

The admin deletes an existing user from the system.

Parameters: username
Precondition: The admin is logged into the website.
Post-condition: A user has been deleted from the system.
Output messages: User <username> has been deleted.
Triggering:

1. From within the usermanagement page on the website select the user related to the user information to delete him.
2. Click on the Delete button to remove the user from the database.

4.5 Edit User

The admin edits an existing user on the system.

Parameters: username
Precondition: The admin is logged into website.
Post-condition: A user has been edited on the system.
Output messages: User <username> has been edited.
Triggering:

1. From within the usermanagement page on the website select the user related to the user information to update him.
2. Using the user information fill out the form.
3. Click on the “Update User“ button to update the user on the database.

4.6 Edit User

The admin edits an existing user on the system.

Parameters: username
Precondition: The admin is logged into the website.
Post-condition: A user has been edited on the system.
Output messages: User <username> has been edited.
Triggering:

1. From within the usermanagement page on the website select the user related to the user information to update him.
2. Using the user information fill out the form.
3. Click on the “Update User“ button to update the user on the database.

4.7 Alert Send

A witness wants to send a firealert using the app.

Parameters: coordinates, phonenumber
Precondition: Alert not yet sended
Post-condition: The alert has been sended
Output messages: Alert recieved.
Triggering:

1. The witness calls up the <FireAlert> application.
2. The witness selects the button !!!!!send message!!! or the button <Call>.
3. The witness select the <send location> button or she/he passes the information through the Telecom operator.

4.8 View all alerts

A ???user??? wants to see all the alerts on the website.

Parameters:
Precondition: User is logged into the website.
Post-condition: User is logged into the website.
Output messages:
Triggering:

1. The user presses the link <Alerts>. ?????

Chapter 5
Error messages and problem resolutions

Here is a short list which problems the customer might encounter during the use of the application. A solution to most of the problems is also there.

5.1 Error message 1

5.1.1 Problem identification

Screen form the WebApplication shows error code: 404 page not found

5.1.2 Probable cause

- It could be that the error is your mistake because you typed the URL incorrectly.
- Another possibility is that the website has been moved and there is no redirect from the old URL to the new URL.

5.1.3 Corrective actions

- Please check and correct the URL.
- Please conntact administrator inorder to get the new URL.

5.2 Error message 2

5.2.1 Problem identification

Cannot login : user does not exist.

5.2.2 Probable cause

- The username you entered is yet not registed.
- You may have made a mistake in your username.

5.2.3 Corrective actions

- Please first register inorder to login to the page.
- Please check your username for any errors and correct them and try again.

5.3 Error message 3

5.3.1 Problem identification

The Website shows the error: Please activate Javascript.

5.3.2 Probable cause

- The WebApplication uses JavaScript, you may have JavaScript disabled or you have a addon which disables it.

5.3.3 Corrective actions

Please enable Javascript:
Google Chrome

- Chrome menu icon on the browser toolbar.
- Settings.
- Show advanced settings.
- In "Privacy" section, click Content settings.
- In "JavaScript" section, select allow all sites to run JavaScript.
- Done.

Internet Explorer

- Click Tools on the web browser menu.
- Internet options.
- Security tab.
- Custom level.
- Enable Active Scripting in the Scripting section.
- Done.

Mozilla Firefox

- Enter as address: about:config.
- Search for "javascript.enabled".
- Change the value to "true".

5.4 Error message 4

5.4.1 Problem identification

Cannot find gps coordinates.

5.4.2 Probable cause

- The area you are might not be covered by any satelite.

5.4.3 Corrective actions

- You can try to move a bit away from the current position and try it again.

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