Test Plan for Intelligent Surveillance system for smart cities

Nour Ahmed, Nour El Hoda Hisham, Mariam Hesham, Samiha Hesham, Sandra Fares
Supervised by: Eng Lobna Mostafa, Dr.Islam Tharwat

March 30, 2021

Table 1: Document version history

Version	**Date**	**Reason for Change**
1.0	27-Mar-2021	Test Plan First version is defined.
1.1	29-Mar-2021	Non-functional Requirements testing is added

GitHub: https://github.com/SandraFW/intelligent-Surveillance-for-smart-cities

Contents

1 Introduction 3

1.1 Purpose . . . 3

1.2 Scope . . . 3

2 Test Scenario: check crime detection's functionality 3

2.1 Test Cases . . . 3

3 Test Scenario: Check for camera failure 4

3.1 Test Cases . . . 4

4 Test Scenario: Check for video summarization functionality 4

4.1 Test Cases . . . 4

5 Test Scenario: Check for face blurring functionality 5

5.1 Test Cases . . . 5

6 Test Scenario: Check for Deep learning detection functionality 5

6.1 Test Cases . . . 5

7 Test Scenario: Check for face detection's functionality 6

7.1 Test Cases . . . 6

8 Test Scenario: application's login functionality 6

8.1 Test Cases . . . 6

9 Test Scenario: creating a new account 7

9.1 Test Cases . . . 7

10 Test Scenario: Non-functional Requirements testing 7

10.1 Scalability testing . . . 7

10.2 Security testing . . . 7

10.3 Availability Testing . . . 7

1 Introduction

1.1 Purpose

This document aims to describe the approaches and methodologies that will be used to test the system functions performance and whether it meets our requirements mentioned in the Software requirements specification paper or not.

1.2 Scope

The scope of this test plan is to make sure that all the requirements proposed in the software specification requirements are met and developed correctly. Any Test plan changes will be done, they will be documented in this document by changing the version numbers of the document.

2 Test Scenario: check crime detection's functionality

The crime detection function FR03 , It depends on the camera captured the video successfully then it must detect either the video streams has any abnormal behaviour or normal in order to send the video stream to the following function.

2.1 Test Cases

Table 2: Test Cases for Scenario 1

Test Case ID	Test Case Desc	Functional Req Code	Test Data	Expected Result
TC01	an incident got detected	FR03	video frames	the incident is annotated with a boundary box and the video is sent to be summarized in the next layer
TC02	no incident is detected	FR03	video frames	video is sent to the archive in the cloud layer

3 Test Scenario: Check for camera failure

As our system depends heavily on data captured by the sensors (cameras), it is essential to check the availability of sensors and make sure that their functionality is not interrupted by any external source.

3.1 Test Cases

Table 3: Test Cases for Scenario 1

Test Case ID	Test Case Desc	Functional Req Code	Test Data	Expected Result
TC03	a black screen appeared while capturing data	FR01	video	an alert is triggered
TC04	a green screen appeared while capturing data	FR01	video	an alert is triggered
TC05	an unfocused camera view	FR01	video	quality enhanced

4 Test Scenario: Check for video summarization functionality

It is beneficial to extract the needed information instead of processing irrelevant data, especially when our goal is to decrease computational power. That is why video summarization plays a huge goal in this system. It's important to make sure that it successfully does its role.

4.1 Test Cases

Table 4: Test Cases for Scenario 1

Test Case ID	Test Case Desc	Functional Req Code	Test Data	Expected Result
TC06	a video is received from the edge layer	FR04	video frames	keyframes extracted and video is summarized

5 Test Scenario: Check for face blurring functionality

After applying video summarization, video streams are passed to a face blurring algorithm to protect the privacy of citizens.

5.1 Test Cases

Table 5: Test Cases for Scenario 1

Test Case ID	Test Case Desc	Functional Req Code	Test Data	Expected Result
TC07	a video is received from the summarization function	FR05	video	blurred faces

6 Test Scenario: Check for Deep learning detection functionality

After applying video summarization and face blurring, video streams are passed to the detection model, placed in the fog layer, so that it could detect and classify the crime use case that took place.

6.1 Test Cases

Table 6: Test Cases for Scenario 1

Test Case ID	Test Case Desc	Functional Req Code	Test Data	Expected Result
TC08	an anomaly got detected	FR06	video frames	video is extracted to apply deblurring
TC09	anomaly is not detected	FR06	video frames	video is sent to the archive in the cloud layer

7 Test Scenario: Check for face detection's functionality

If an anomaly got detected, the faces get deblurred and a face detection algorithm gets applied.

7.1 Test Cases

Table 7: Test Cases for Scenario

Test Case ID	Test Case Desc	Functional Req Code	Test Data	Expected Result
TC10	face detected	FR07	video frames	trigger an alarm

8 Test Scenario: application's login functionality

8.1 Test Cases

Table 8: Test Cases for Scenario

Test Case ID	Test Case Desc	Functional Req Code	Test Data	Expected Result
TC11	user entered an incorrect email or password	FR09	string	user is not allowed to proceed with the application
TC12	user entered a correct email and password	FR09	string	user can proceed with the application and view alerts

9 Test Scenario: creating a new account

9.1 Test Cases

Table 9: Test Cases for Scenario

Test Case ID	Test Case Desc	Functional Req Code	Test Data	Expected Result
TC13	Admin creates, for a user, an account that already exists	FR10	string	new account is not added and an alert message appears
TC14	Admin enters wrong data format	FR10	string	validation messages appear
TC15	Admin enters correctly formatted data	FR10	string	a new account got created

10 Test Scenario: Non-functional Requirements testing

After deploying all the functional requirements, the non functional requirements mentioned in software specification document will be tested.And then it will be compared with other systems introduced to check if our work accomplished its target.

10.1 Scalability testing

the system will undergo some scalability tests using ifogsims simulator to check Scalability attributes as:

- Response Time
- CPU Usage
- Network Usage
- Performance under load

and then it will be compared with other similar systems performance.

10.2 Security testing

We will apply a security test as vulnerability scanning, security scanning,etc to identify the threats in the system and measure its potential vulnerabilities, so the threats can be encountered.

10.3 Availability Testing

One of the most non functional testing as the system will work continuously for long period of times.We must ensure that the system components do not crash.