

Reality Sensing, Mining and Augmentation   
for Mobile Citizen–Government Dialogue

FP7-288815

**Test scenario’s &** **results**

**Sensor Collection Service (S1)**

|  |  |
| --- | --- |
| fp7_logo | eu-flag |

co-funded by the European Union

# Template instructions

This template is used for documenting test scenarios and test results. ‘D4.4 – Technical verification and testing strategies’ describes per phase which tests need to be performed and which work package/partner is responsible for setting up and performing these tests.

Along with the software development the test scenarios are constructed based on the requirement as described in ‘D4.1 – System Architecture and Design’ and ‘D5.1 – Detailed Use Case Descriptions’.

These test scenario’s are described and agreed upon before starting the actual tests. This means that all blue sections need to be pre-filled before starting the actual test.

# Test configuration

|  |  |
| --- | --- |
| Software identification | |
| Name | Technical verification and testing strategy |
| Versions | Mobile Sensor Collector (C14)  Server Data Storage Component (C8) |

|  |  |
| --- | --- |
| Test period | |
| Test phase | Service Level Testing |
| Test Types | Functional |
| Test Status | Test Completed |
| Planned test start date | 01.2013 |
| Actual test start date | 05.2013 |
| Test completion date | 11.2013 |
| Partners(s) | UKOB |
| Tester(s) | Heinrich Hartmann, Christoph Schaefer |

|  |  |
| --- | --- |
| Test environment | |
| Test environment | Development |
| Test devices | Mobile Devices:   * LG/Google Nexus 4 * HTC Evo 3D * Galaxy Tab 01 * Motorola Razr XT910 01 |
| Test pc’s | Ubuntu 12.04 Server. x64. Intel Core i5 CPU@2.4Ghz, 4Gb RAM, 100 GB HDD |

|  |  |
| --- | --- |
| References | |
| Reference | Deliverable 1.1 for technical documentation of the sensor collection component  Software Repository: <https://liveandgov.codespaces.com/svn/wp1/Mobile%20Sensor%20Collection%20Component>  Inspection Front End:  http://mobile-sensing.west.uni-koblenz.de:3000/ |

# Test scenarios

The requirements in the table below we refer to the context mining requirements derived in D1.1.

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Requirements | Expected behaviour | [OK/  NOK] |
| 1 | RA.1 On-Spot Location  RA.2 Location Tracking R-SC.1 GPS | The mobile application shall record sensor samples in the appropriate frequencies. Those samples shall be transferred to the server and stored in a database.  The success of the recording can be tested using the Inspection Front End. It allows to visually inspecting the sensor samples with respect to recording frequency and sanity checks. | OK |
| 2 | R-SC.2 Accelerometer, 20Hz | OK |
| 3 | R-SC.3 Gyroscope, 20Hz | Ok |
| 4 | R-SC.4 Magnetometer, 5Hz | OK |
| 5 | R-SC.5 GSM, every 30 sec. | OK |
| 6 | R-SC.6 Wifi, every 30 sec. | OK |
| 7 | R-SC.7 Microphone, 1Hz | NOK |

# Issues raised

|  |  |
| --- | --- |
| Issue No. | SC-1 |
| Scenario ID | ? |
| Severity | Low |
| Type | Change Request |
| Summary | The Microphone Sensor is not recorded |
| Description | The Android API does not support the collection of microphone samples, i.e. noise levels. |
| Workaround | Avoid the use of the microphone sensor. |
| Recommendations | Mining of sensor data is not critical for the project. Therefore it is recommended to postpone the inclusion of this sensor until it becomes relevant. |

|  |  |
| --- | --- |
| Issue No. | SC-2 |
| Severity | High |
| Type | Bug |
| Summary | Irregularities in recorded sensor data |
| Description | In some cases, the recording of sensor data from motions sensors like the accelerometer was very irregular. More precisely, that there were no samples for several seconds and then a whole lot compressed into a small time interval. |
| Workaround | - |
| ~~Recommendations~~  Solution | The issue was caused by too much workload for the CPU of the mobile client. We replaced the DB-driven storage engine of the sensor collector with a flat file. This solved the issue, apparently. |

|  |  |
| --- | --- |
| Issue No. | SC-3 |
| Severity | Medium |
| Type | Bug |
| Summary | Network time request lets application freeze at startup |
| Description | Sometimes, when no reliable internet connection is present at startup, the sensor collection application freezes. |
| Workaround | Connect to the internet at startup of the application. |
| ~~Recommendations~~  Solution | The application was waiting for a network time request to sync the clock of the mobile device. As there are no major implications of the mobile clock being off by a few seconds, we could just remove this network time request. |

|  |  |
| --- | --- |
| Issue No. | SC-4 |
| Severity | High |
| Type | Bug |
| Summary | Loss of data while transferring |
| Description | Sometimes, the recorded samples did not arrive at the server backend, although the mobile device reported them to be transferred. |
| Workaround | Record samples again. |
| ~~Recommendations~~  Solution | The application was using too much memory for the compression of the file before the transfer. We replaced the compression module with a stream based system, which does not require to store all the samples in memory for compression.  Moreover, we implemented an additional flag in the server response, that prevents us from deleting samples on the mobile client before they are stored in the server side database. |

# Issue screenshots

|  |  |
| --- | --- |
| Issue No. | SC-2 |
| F:\Screenshot from 2013-11-11 17:18:46.png  This screenshot shows Gyroscope sensor samples coming in at irregular intervals. In the second 13:17:48, there are only three recorded samples. At 13:17:50 the recording continues at the expected rate. | |