**a) Project vision**

This project is realized as part of the course in Smart Sensor Network Systems for the summer semester 2018. It consists in the design and implementation of a fall-detection system based on acceleration and gyroscopic data. Mr La Blunda will act as project owner. [1]

The set-up has two main parts: the first consists of two Sensortags, which have to be worn around the waist of the test subject, that will gather the sensor data and send it over a Bluetooth connection to the Base Station for elaboration; the second part is the Base Station, a Bluetooth equipped PC which will run the application that will elaborate the sensor data, try to identify falls, and if necessary request help.

The project span is 7 weeks, the latest possible delivery date being June 22nd.

This is a stand alone project, with no interaction with other groups or organizations, and no dependencies on other projects by this or other teams.

As the project is intended to develop the understanding of smart sensor networks and the technical understanding of their development process, the software itself is not the sole product. Every relevant document produced during the development, including but not limited to, this document, weekly individual reports by the team members, and a final report and presentation, will be part of the delivered artifacts.

The following elements do not fall within the scope of this project and will not be included in the finished product:

* considerations on the hardware design of the wearable part
* a user manual
* maintenance and support of the product after initial delivery

**d) Project plan**

**i) Estimation**

For the estimation of effort, the COCOMO II model was used [3], which was based on the value of Function Points [2].

**i-1) Function points**

The Function Points calculation process was conducted only until the Unadjusted Function Points values where obtained, because it is these values which are employed by the COCOMO II model.

In identifying the Application Boundary, we considered the two Sensortag devices and the PC application not as standalone systems, but as two of three modules that make up the complete application. As a consequence, the internal communication between the modules does not constitute a transaction; also, the complete system results stand-alone, and does not therefore possess External Interface Files.

**i-1.1) Transactions**

In the following table Transaction (External Input, External Output, External Inquiry) are listed.

They are subdivided according to the Actor that is responsible for them.

|  |  |  |  |
| --- | --- | --- | --- |
|  | External Input | External Output | External Inquiry |
| User | Insert system calibration (3FP)  Insert user general information (3FP)  Insert helper contact information (3FP)  Load defaults (3FP) | Accelerometer graph (4FP)  Gyroscope graph (4FP)  Label “Fall detection” (4FP)  Label “Help requested” (4FP) | Open application settings (3FP)  Start/Stop (3FP)  Connect /Disconnect (3FP)  Close button (3FP)  Clear graph button (3FP) |
| Helper |  | Send email (4FP) |  |
| Sensors | Gyroscope (3FP)  Accelerometer (3FP)  Snooze Alarm (Button 2) (3FP)  Bluetooth connection PC (4FP) Bluetooth connection Sensortags (4FP) |  |  |
| Actuators |  | Buzzer “false alarm” (4FP) |  |

**i-1.2) Internal Logical Files**

In the following table, ILFs are listed. They are subdivided according to the software module they

belong to.

|  |  |
| --- | --- |
| Module | Internal Logical Files |
| Sensortags | None |
| PC Application | User Information (7FP)  System calibration values (7FP)  Helper contact data (7FP) |

Total (unadjusted) Function Points: 89

**i-2) Estimation of Effort**

The estimation of effort was conducted with COCOMO II. Considered the early stage of

development of the project, the Early Design Model was selected.

**Scaling Drivers**

|  |  |
| --- | --- |
| *Driver* | *Value* |
| Precedentedness | High |
| Development Flexibility | Nominal |
| Risk Resolution | Nominal |
| Team Cohesion | High |
| Process Maturity | Very Low |

**Cost Drivers**

|  |  |
| --- | --- |
| *Driver* | *Value* |
| Facilities | Nominal |
| Personnel Experience | Nominal |
| Personnel Capability | High |
| Required Reusability | Low |
| Platform Difficulty | Nominal |
| Product Reliability and Complexity | Low |
| Required Development Schedule | Nominal |

**Results**

|  |  |
| --- | --- |
|  | *Value* |
| Person-Months | 9.7 |
| Schedule Months | 1.9 |
| SLOC | 4717 |

**i-2.1)Notes**

The result of 1.9 months seems at first encouraging. However, our analysis must also consider that, on one hand, the team is working only part-time on the project, and on the other, that the final product will be a prototype, not a production ready system. In light of these facts, we believe the estimation to be generally accurate.

**ii) @Johnny this part you find in the excel file “schedule.xlsx”**

**iii) Project Organization**

In consideration of the short time span of this project, and of the prototype nature of the final system, the team decided to employ agile development techniques, specifically Scrum. Sprints will have a duration of one week. Two scrum meeting will be held each week, tentatively on Wednesdays at 16:00 and Fridays at 15:00.

Official communication will be organized through two channels:

* for short or urgent messages and general coordination, the Slack “SSNS” group chat;
* for communication with the project owner, the forum “Group A” on Moodle, or per email

All common artifacts (source code, documentation, reference sources, etc.) are to be uploaded on the team’s GitHub repository.

**iv) Responsibilities of all team members**

All team members will assume several roles during the project. However, each person has been assigned a main role, making him or her the coordinator of all the individual efforts for a specific subject.

|  |  |
| --- | --- |
| Name | Main Role |
| Raul Bertone | Project Manager, Scrum Master |
| Elis Haruni | Lead Java Developer |
| Muyassar Kokhkharova | Statistics, UI Designer |
| Saidar Ramazanov | Mathematical Model |
| Xhoni Robo | Lead C Developer |

**f) Setup and description of the development environments**

*SensorTags*

The development for the SensorTags (and if necessary for the Launchpad) will be in C. For it we will use Texas Instrument’s own Code Composer Studio as well as the toolchain provided by Texas Instruments.

*PC application*

The PC application will be implemented in Java. For its development we will use Eclipse as an IDE, Junit for unit testing.

All mentioned software are available for both Linux and Windows, so we will leave the choice of an OS open for each team member (none makes use of MacOS). Additionally, the choice of Java as the implementation language allows the deployment and testing of the PC application on all OSs.

As a version control system software we selected *git,* and, based on this, GitHub as an online shared repository. All common artifacts (source code, documentation, etc.) will be uploaded there. In the root folder a readme file will describe the intended use of the different folders. The master branch is protected from accidental modifications by requiring the use of pull-requests for its update.

The repository can be found at the following address: <https://github.com/raulbertone/SSNS>

We integrated GitHub into our Slack group to be timely informed about commits performed by other team members.

[1] M.Sc. L. La Blunda, "Project Proposal," [Online]. Available: https://moodle.frankfurt-university.de/pluginfile.php/512521/mod\_forum/attachment/94672/Project1.pdf

[2] A. J. Albrecht, "Measuring Application Development Productivity," in Proceedings of the Joint

SHARE, GUIDE, and IBM Application Development Symposium, Monterey, California, 1979.

[3] B. Boehm, "COCOMO II Model Definition Manual," [Online]. Available:

http://csse.usc.edu/csse/research/COCOMOII/cocomo2000.0/CII\_modelman2000.0.pdf.