Raul Bertone
Elis Haruni
Muyassar Kokhkharova
Saidar Ramazanov
Xhoni Robo
Total Team Effort: 23h30m

Project Proposal - Fall-detection based on acceleration & gyroscope data

A) Project Vision (Raul Bertone, Saidar Ramazanov, 2h30m)

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 standalone 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.

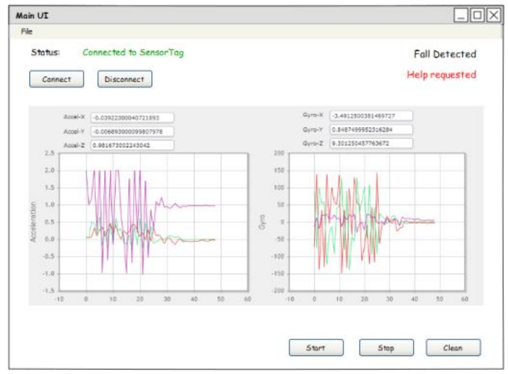
The following elements do not fall within the scope of this project and will not be included in the finished product:

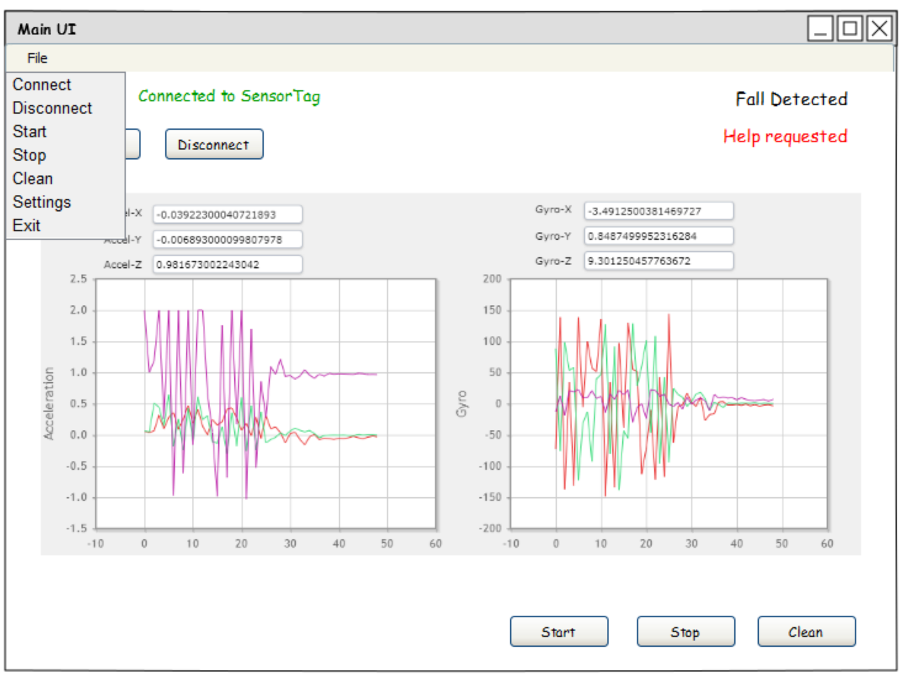
- considerations on the hardware design of the wearable part
- a user manual
- maintenance and support of the product after initial delivery

B) List of functional and non-functional requirements (Muyassar Kokhkharova, 12h)

Functional requirements

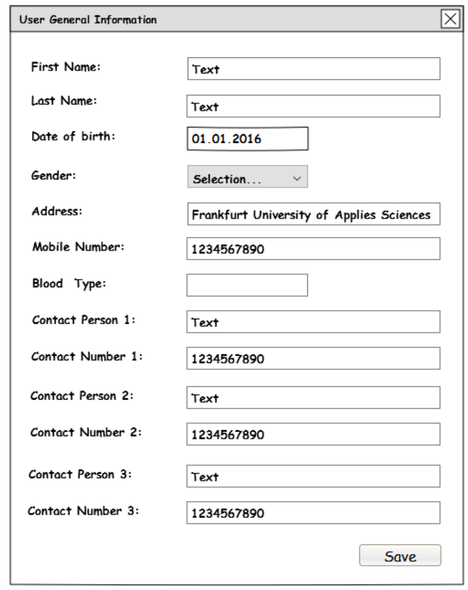
1. Two CC2650 SensorTags are acting as peripherals and sending periodically acceleration and gyroscope data to the PC.
2. A PC application should be able to connect simultaneously to multiple peripherals via BLE.
3. A PC application should be able to receive sensor data (accelerometer & gyroscope) in real time.
4. Data visualization. Line graphs with accelerometer and gyroscope data received from SensorTag.
5. The graph should depict the average value of received sensor data within the offset time.
6. Possibility to enter User‘s general information.
7. Possibility to calibrate system thus differentiate between sudden movements like walking the steps and free fall .
8. Main UI with basic control functions for operator working with a PC application.





Non-functional requirements

1. User general Information is a pop up window and it should contain:



- First name
- Last Name
- Date of Birth
- Gender
- Address
- Mobile number of user
- Blood type
- Contact Person1

  In case of fall this Person will be contacted.
- Contact Number1

  Phone number of a Contact Person.
- Contact Person2

  In case of fall and if Contact Person1 is not replying this Person will be contacted.
- Contact Number2

  Phone number of a Contact Person2.
- Contact Person3*

  In case of fall and if Contact Person1 and Contact Person2 are not replying this Person will be contacted.
- Contact Number3*

  Phone number of a Contact Person3.
- Save button to save changes

2. Application Settings is a pop up window containing following information:

2.1 Offset with default value of 0.25 seconds.

Graphs of gyroscope and accelerometer will be updated every offset time with average values of gyroscope and accelerometer received during offset time.

2.2 Delta for fall detection (gyroscope and accelerometer).

While falling values of accelerometer will gain acceleration and then will be equal to null. If delta of accelerometer received from sensor is bigger than configured delta in settings it will mean that the person is probably fell. The duration of free fall is also important because we have to differentiate between sudden movements like walking the steps and free fall, the duration will be different although delta values might be the same. While falling values of gyroscope will change and after reaching the ground values will be in horizontal position.

2.3 Set Default button

After clicking on it, all the default values will be set and Settings pop up window will remain open.

2.4 Save button

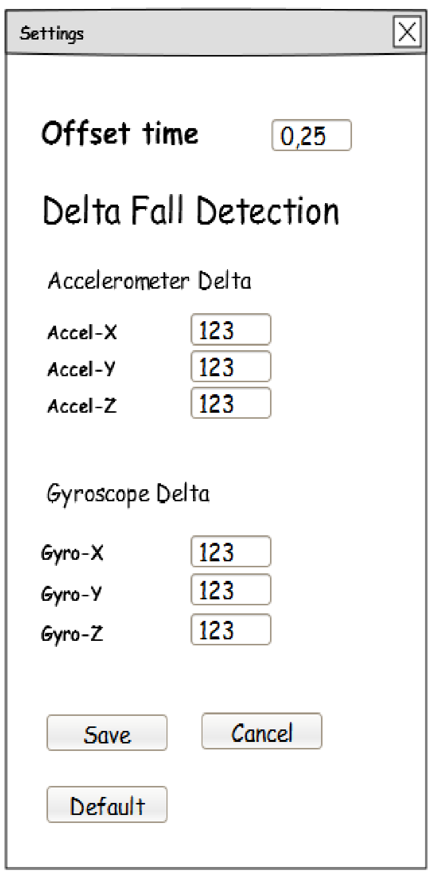
After clicking on save button, all the changes will be saved and Settings pop up window will remain open.

2.5 Cancel button

Cancel changes.

2.6 Close button

Clicking on close button will close Application Settings pop up window.



3. In Main UI:

3.1 Graphs with Accelerometer and Gyroscope data.

3.2 Accelerometer and Gyroscope data in the graphs will be updated at the same offset time.

3.3 Buttons connect/disconnect(to establish the bluetooth connection).

3.4 Buttons start/stop receiving gyroscope and accelerometer data.

3.5 Button Clear graph.

3.6 Label for Sensor Status (Connected to the sensor or No connection).

3.7 Label for Fall Detection.

If fall is detected inscription about falling will be displayed.

If not inscription «Fall was not detected» will be displayed.

3.8 Label for «help requested».

If fall was detected and user did not pressed the button on SensorTag to report about False alarm inscription on requesting help will be displayed.

3.9 When the Sensor is not connected to the PC Application(e.g. when Data is not being received), than Buttons Start, Stop and Clear should be deactivated*

3.10 Stop button is Activated when Start is pressed.

4. „False alarm“. When device senses a fall it beeps first and if button on SensorTag is not pressed quickly it calls for help

C) Safety, security and reliability requirements (Xhoni Robo, 2h)

For an application whose sole purpose is the detection of a person falling, it is important to ensure that, at the very least, do what is required of it. However, it becomes of critical importance when paired with the fact that the user may be in danger following this fall. If this were a project made to be used for actual health benefits, failure to correctly assess when a person is in danger may even be fatal. Safety in this project translates to the protection of the hardware as well as software mechanisms that allow the fall to be detected. Preventing physical damage of the sensors is pretty self explanatory, and can be done by simply changing the design so that the fall itself would not be enough to break the sensors. As far as software is concerned, we need to ensure that the final application not only receives the data and correctly uses it, but also ensure that there are no interferences by other devices. In the case of low connectivity, the application should immediately notify the user. Lastly, the application should also ensure that it picks only the data received from the sensor tags. That way, there will be no issues with interference. Should anything not work as intended, the user should be notified immediately.

Once safety and security is ensured, the final application needs to also be reliable. The most basic reliability requirement is to prevent the application of notifying us of events that are similar to a fall, but that provide no danger to the user. This includes physical activities such as walking, running and even jumping. Below is the full list of safety, security and reliability requirements:

Safety:

- Software correctly notifies when a person has fallen
- Software correctly notifies user when one or both sensor tags do not work
- Software or SensorTags correctly notify user when connectivity is low
- If transmission stops abruptly during an activity similar to a fall, count that as a fall

Security:

- Data is collected from the associated SensorTags
- Other devices cannot send data to the application
- If any interferences are detected, notify the user

Reliability:

- Software differentiates between falls and other similar activities
- Software does not crash during long sessions where a lot of data is streamed
- Software notifies user when the SensorTags are low on battery
- SensorTags only send the necessary data. Other sensors should be disabled.

D) Project Plan (Raul Bertone, 6h30m)

i) Project 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) Project Scheduling (Raul Bertone, Saidar Ramazanov, 2h)

Sprint 1	Sprint 2	Sprint 3
Focus: design	Focus: prototype implementation	Focus: prototype implementation
Tasks	**Tasks**	**Tasks**
Literature research: fall detection	UI Design	Sensortag: sensor data gathering
Mathematical model	Sensortag: basic sensor data gathering	Communication: Bluetooth connection
Definition of use cases	Communication: basic Bluetooth connection	PC application: data visualization
Component design	PC application: basic data visualization	Integration testing
Individual weekly reports	Individual weekly reports	Individual weekly reports
May 26, May 27, May 28, May 29, May 30, May 31, Jun 01	Jun 02, Jun 03, Jun 04, Jun 05, Jun 06, Jun 07, Jun 08	Jun 09, Jun 10, Jun 11, Jun 12, Jun 13, Jun 14, Jun 15

Sprint 4	Sprint 5	Sprint 6
Focus: features implementation	Focus: features implementation	Focus: testing
Tasks	**Tasks**	**Tasks**
Additional features implementation	Additional features implementation	Final report
Performance testing	Individual weekly reports	Test statistics
Individual weekly reports		Reliability testing
		Individual weekly reports
Jun 16, Jun 17, Jun 18, Jun 19, Jun 20, Jun 21, Jun 22		

Sprint 7
Focus: final report
Tasks
Final report

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

v) Project risk analysis (Elis Haruni, 6h)

Risk Management Matrix

NAME	Elis Haruni	OBJECTIVE	Minimizing the Project Risk

REF/ID	PRE-MITIGATION: RISK	RISK SEVERITY	RISK LIKELIHOOD	RISK LEVEL	DEPARTMENT / LOCATION	MITIGATIONS / WARNINGS / REMEDIES	POST-MITIGATION: RISK SEVERITY	RISK LIKELIHOOD	RISK LEVEL	ACCEPTABLE TO PROCEED?
	Gold plating inflates scope	UNDESIRABLE	PROBABLE	HIGH	ENGINEERS	Everyone should discuss with team and can add something only after a team …	TOLERABLE	POSSIBLE	LOW	YES
SCOP	Scope creep inflates scope	TOLERABLE	POSSIBLE	MEDIUM		Mace sure that we don't go out of scope in our favorte parts, also is good to go out …	TOLERABLE	POSSIBLE	MEDIUM	YES
SCOP	Estimates are inaccurate	UNDESIRABLE	PROBABLE	HIGH	MANAGEMENT	Use trusted and good mathematical estimation methods	TOLERABLE	POSSIBLE	MEDIUM	YES
SCOP	Activities are missing from scope	INTOLERABLE	PROBABLE	HIGH	MANAGEMENT	Discuss with proffeors/Tutors and also check to not forget anytihg unassigned	UNDESIRABLE	IMPROBABLE	MEDIUM	YES
COST	Cost forecasts are inaccurate	ACCEPTABLE	IMPROBABLE	LOW						YES
CHMGM	Change management overload	UNDESIRABLE	POSSIBLE	MEDIUM		Discuss and ignore changes if they will complicate the work more than needed	UNDESIRABLE	IMPROBABLE	MEDIUM	YES
CHMGM	Lack of a change management system	TOLERABLE	POSSIBLE	MEDIUM						YES
CHMGM	Inaccurate change priorities	INTOLERABLE	PROBABLE	EXTREME	MANAGEMENT	Be carful to check Changes before decidint them	TOLERABLE	POSSIBLE	EXTREME	YES
CHMGM	Low quality of change requests	UNDESIRABLE	IMPROBABLE	EXTREME			TOLERABLE	IMPROBABLE	HIGH	YES
CHMGM	Change request conflicts with requirements	UNDESIRABLE	IMPROBABLE	EXTREME			TOLERABLE	IMPROBABLE	HIGH	YES
STAK	Stakeholders become disengaged	INTOLERABLE	IMPROBABLE	LOW						
STAK	Stakeholders fail to support project	UNDESIRABLE	IMPROBABLE	MEDIUM						
STAK	Stakeholder conflict	UNDESIRABLE	POSSIBLE	HIGH	MANAGEMENT	Discuss everything in details with tutors	TOLERABLE	POSSIBLE	MEDIUM	YES
STAK	Process inputs are low quality	INTOLERABLE	IMPROBABLE	EXTREME						
COM	Project team misunderstand requirements	INTOLERABLE	PROBABLE	EXTREME	MANAGEMENT	Be sure everyone understand his part and the requerements include Text Use Cases	UNDESIRABLE	POSSIBLE	MEDIUM	YES
COM	Communication overhead	UNDESIRABLE	POSSIBLE	MEDIUM	MANAGEMENT	Start working as soon as possible.	TOLERABLE	POSSIBLE	MEDIUM	YES
COM	Under communication	UNDESIRABLE	PROBABLE	HIGH	MANAGEMENT	make sure everyone report weekly this will at least be a form of communication.	TOLERABLE	IMPROBABLE	HIGH	YES
COM	Users have inaccurate expectations	UNDESIRABLE	POSSIBLE	EXTREME	MANAGEMENT	Discuss again with proffesor and tutors what we are building.	TOLERABLE	IMPROBABLE	EXTREME	YES
COM	Impacted individuals aren't kept informed	INTOLERABLE	POSSIBLE	EXTREME	MANAGEMENT	Take resposibility to upload and merge the weekly report for tutors to check.	UNDESIRABLE	IMPROBABLE	MEDIUM	YES
R&T	Resource shortfalls	INTOLERABLE	POSSIBLE	HIGH	MANAGEMENT	Before assigning the tasks to every team member ask for their capabilities	UNDESIRABLE	POSSIBLE	HIGH	YES
R&T	Learning curves lead to delays and cost overrun	UNDESIRABLE	PROBABLE	LOW		This should go to unprobable after discussing with proffesors.	TOLERABLE	IMPROBABLE	LOW	YES
R&T	Resources are inexperienced	TOLERABLE	PROBABLE	LOW		Try to divide the tasks based on team members experience	TOLERABLE	PROBABLE	LOW	YES
R&T	Resource performance issues	UNDESIRABLE	PROBABLE	MEDIUM			UNDESIRABLE	PROBABLE	MEDIUM	YES
R&T	eam members with negative attitudes towards the …	UNDESIRABLE	POSSIBLE	LOW			UNDESIRABLE	POSSIBLE	LOW	YES
R&T	Low team motivation	TOLERABLE	POSSIBLE	LOW						YES
ARCH	Architecture lacks flexibility	UNDESIRABLE	POSSIBLE	MEDIUM		Agile and SCRUM development based archiitecture	TOLERABLE	IMPROBABLE	MEDIUM	YES
ARCH	Architecture is not fit for purpose	UNDESIRABLE	POSSIBLE	HIGH		choose some architecturial tools that are confirmed for this type of project.	TOLERABLE	IMPROBABLE	HIGH	YES
ARCH	Architecture is infeasible	UNDESIRABLE	POSSIBLE	MEDIUM	MANAGEMENT	Preform estimation methods for our architecture or choose another archiceture	UNDESIRABLE	POSSIBLE	MEDIUM	YES
DES	Design is infeasible	TOLERABLE	IMPROBABLE	HIGH			TOLERABLE	IMPROBABLE	HIGH	YES
DES	Design lacks flexibility	UNDESIRABLE	POSSIBLE	EXTREME	MANAGEMENT	Make sure the desig is flexible by performing over it estimation methods for flexibility	TOLERABLE	IMPROBABLE	EXTREME	YES
DES	Design is not fit for purpose	INTOLERABLE	IMPROBABLE	EXTREME			INTOLERABLE	IMPROBABLE	EXTREME	YES
TECH	Technology components aren't fit for purpose	INTOLERABLE	IMPROBABLE	EXTREME						YES
TECH	Technology components aren't scalable	UNDESIRABLE	POSSIBLE	HIGH	ENGINEERS	Read documentation carefully before chhsing components, consider team members experince on similar projects	TOLERABLE	POSSIBLE	MEDIUM	YES
TECH	Technology components aren't interoperable	UNDESIRABLE	PROBABLE	EXTREME	ENGINEERS	Again this can be avoided if we read the full information for components	TOLERABLE	POSSIBLE	HIGH	YES
TECH	Technology components aren't compliant with sta…	INTOLERABLE	PROBABLE	EXTREME	ENGINEERS	Be sre componets fullfill the IEEE and other Internatioal Standars	INTOLERABLE	IMPROBABLE	EXTREME	YES
TECH	Technology components have security vulnerabil…	INTOLERABLE	POSSIBLE	EXTREME	ENGINEERS	Update Security Protocols	ACCEPTABLE	POSSIBLE	MEDIUM	YES
TECH	Technology components are over-engineered	ACCEPTABLE	PROBABLE	LOW						YES
TECH	Technology components lack stability	UNDESIRABLE	POSSIBLE	MEDIUM	ENGINEERS	This can be accepted as a risk if we prepare a good specification of the system	ACCEPTABLE	POSSIBLE	MEDIUM	YES
TECH	Technology components aren't extensible	INTOLERABLE	IMPROBABLE	HIGH						YES
TECH	Technology components aren't reliable	UNDESIRABLE	POSSIBLE	LOW	ENGINEERS	Components are verified to be relaible	ACCEPTABLE	POSSIBLE	LOW	YES
TECH	Information security incidents	ACCEPTABLE	PROBABLE	LOW						YES
TECH	System outages	TOLERABLE	IMPROBABLE	HIGH						YES
TECH	Legacy components lack documentation	UNDESIRABLE	PROBABLE	HIGH	MANAGEMENT	Trusted seller	ACCEPTABLE	IMPROBABLE	LOW	YES
TECH	Legacy components are out of support	TOLERABLE	IMPROBABLE	LOW						YES
TECH	Components or products aren't maintainable	ACCEPTABLE	IMPROBABLE	LOW						YES
TECH	Project management tool problems & issues	TOLERABLE	POSSIBLE	LOW		We are a small team that will and can meet every time	ACCEPTABLE	POSSIBLE	LOW	YES
INTG	Failure to integrate with systems	UNDESIRABLE	IMPROBABLE	HIGH						YES
INTG	Integration testing environments aren't available	TOLERABLE	POSSIBLE	LOW						YES
INTG	Failure to integrate components	UNDESIRABLE	PROBABLE	HIGH	MANAGEMENT	As a small team we can control compontents one after other te bo sure	ACCEPTABLE	POSSIBLE	HIGH	YES
INTG	Project disrupts operations	UNDESIRABLE	IMPROBABLE	MEDIUM						YES
REQ	Requirements fail to align with strategy	UNDESIRABLE	POSSIBLE	MEDIUM						
REQ	Requirements fail to align with systems	UNDESIRABLE	POSSIBLE	LOW						
REQ	Requirements have compliance issues	UNDESIRABLE	IMPROBABLE	LOW						YES
REQ	Requirements are ambiguous	UNDESIRABLE	PROBABLE	HIGH						
REQ	Requirements are low quality	INTOLERABLE	POSSIBLE	HIGH		Check requirements before starting the project	UNDESIRABLE	IMPROBABLE	HIGH	YES
REQ	Requirements are incomplete	UNDESIRABLE	POSSIBLE	LOW		We can improve requirements during the process	ACCEPTABLE	PROBABLE	LOW	YES
DIS	Decisions are low quality	UNDESIRABLE	PROBABLE	MEDIUM	MANAGEMENT	We are small team and make decisions together	TOLERABLE	PROBABLE	MEDIUM	YES
DIS	Decisions are incomplete	INTOLERABLE	PROBABLE	HIGH	MANAGEMENT	Meetings and clear decisions documented weekly report will also help to minimize this risk	UNDESIRABLE	POSSIBLE	MEDIUM	NO
PROCU	Loss of intellectual property	ACCEPTABLE	IMPROBABLE	LOW						YES
AUTH	Project team lack authority to complete work	UNDESIRABLE	POSSIBLE	MEDIUM	MANAGEMENT	Team meeting and remove some requirements if it is out of our scope	UNDESIRABLE	IMPROBABLE	MEDIUM	YES
AUTH	Authority is unclear	INTOLERABLE	PROBABLE	EXTREME	MANAGEMENT	Team meeting and discuss and assign roles to everyone	TOLERABLE	IMPROBABLE	EXTREME	YES
PM	Failure to follow methodology	UNDESIRABLE	PROBABLE	HIGH	MANAGEMENT	Present the methotolgy to all team members and check peridicaly if they are following	TOLERABLE	POSSIBLE	MEDIUM	YES
PM	Errors in key project management processes	TOLERABLE	PROBABLE	MEDIUM	MANAGEMENT	Use of scientific and good estimation methods	TOLERABLE	POSSIBLE	MEDIUM	YES
UA	User interface is low quality	UNDESIRABLE	PROBABLE	HIGH	ENGINEERS	We alredy have a user interface, also make sure everything works on the interface	TOLERABLE	IMPROBABLE	HIGH	YES
UA	User interface isn't accessible	INTOLERABLE	POSSIBLE	EXTREME	ENGINEERS	UI will be on Desktop	INTOLERABLE	IMPROBABLE	EXTREME	YES
UA	Project reduces innovation	INTOLERABLE	IMPROBABLE	EXTREME						YES
UA	Users reject the product	TOLERABLE	IMPROBABLE	LOW						YES

RISK RATING KEY	LOW	MEDIUM	HIGH	EXTREME
	0 – ACCEPTABLE	1 – ALARP (as low as reasonably practicable)	2 – GENERALLY UNACCEPTABLE	3 – INTOLERABLE
	OK TO PROCEED	TAKE MITIGATION EFFORTS	SEEK SUPPORT	PLACE EVENT ON HOLD

		SEVERITY			
		ACCEPTABLE LITTLE TO NO EFFECT ON EVENT	TOLERABLE EFFECTS ARE FELT, BUT NOT CRITICAL TO OUTCOME	UNDESIRABLE SERIOUS IMPACT TO THE COURSE OF ACTION AND OUTCOME	INTOLERABLE COULD RESULT IN DISASTER
LIKELIHOOD	IMPROBABLE RISK IS UNLIKELY TO OCCUR	LOW – 1 –	MEDIUM – 4 –	MEDIUM – 6 –	HIGH – 10 –
	POSSIBLE RISK WILL LIKELY OCCUR	LOW – 2 –	MEDIUM – 5 –	HIGH – 8 –	EXTREME – 11 –
	PROBABLE RISK WILL OCCUR	MEDIUM – 3 –	HIGH – 7 –	HIGH – 9 –	EXTREME – 12 –

RISK SEVERITY KEY	RISK LIKELIHOOD KEY	RISK LEVEL KEY	ACCEPTABLE TO PROCEED? KEY
ACCEPTABLE	IMPROBABLE	LOW	YES
TOLERABLE	POSSIBLE	MEDIUM	NO
UNDESIRABLE	PROBABLE	HIGH	
INTOLERABLE		EXTREME	

E) Safety and Security Plan (Xhoni Robo, 1h)

Many of the requirements for safety and security can be considered by most to be common sense. However, they will be implemented in the later parts of the development process. This is due to the fact that a somewhat functional application can be made without these requirements in mind. However, the final application should include all the requirements. Most of them are easy to implement and should not take too much time. Even with setbacks, all the requirements should be implemented before the deadline. However, in case of some unexpected events, we need to prioritize them. The priority takes into account not only a requirements importance, but also the time it takes to implement it. This is done with an arbitrary scale in mind, and is so far only an estimate. Below is a list of the safety, security, and reliability requirements, ordered in the way that we initially plan to implement them:

1. Data is collected from the associated SensorTags
2. SensorTags only send the necessary data. Other sensors should be disabled.
3. Software correctly notifies when a person has fallen
4. Software correctly notifies user when one or both sensor tags do not work
5. Software or SensorTags correctly notify user when connectivity is low
6. Software notifies user when the SensorTags are low on battery
7. Software differentiates between falls and other similar activities
8. Software does not crash during long sessions where a lot of data is streamed

9. If transmission stops abruptly during an activity similar to a fall, count that as a fall
10. Other devices cannot send data to the application
11. If any interferences are detected, notify the user

F) Setup and description of 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.

G) Literature References

[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.