Fall Detection System Using Sensors Embedded In Smartphones

Vineeth Bhat
Dept.Computer Science and Engineering
Frankfurt University of Applied Sciences
Frankfurt, Germany
vineeth.bhat@stud.fra-uas.de

Jathin Sreenivas
Dept.Computer Science and Engineering
Frankfurt University of Applied Sciences
Frankfurt, Germany
Jathin.Sreenivas@stud.fra-uas.de

Vidya Gopalakrishnarao
Dept.Computer Science and Engineering
Frankfurt University of Applied Sciences
Frankfurt, Germany
vidya.gopalakrishnarao@stud.fra-uas.de

***Abstract*—This document is an introduction to the project Human Activity Recognition(HAR). The scope of the project is to understand the working of the sensors in our smartphones or tablet and access it to track various human activity like walking, jogging, etc and propose a system(application) that detects fall.**

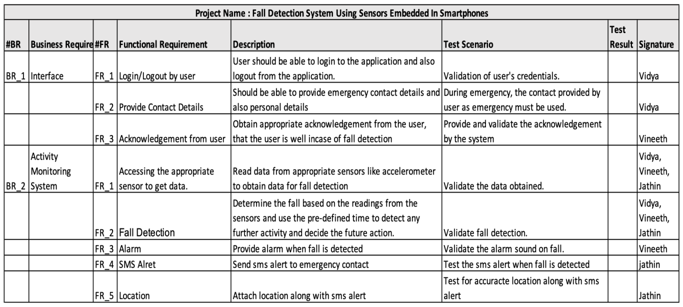
I. Project Vision

Our smartphones are filled with sensors that read various data from the physical world around them. These data is then utilised by various application on the smartphone as per their need. The objective of this project is to provide an application that detects fall, when a person falls while he is walking or jogging.// We are building an application that access the sensors embedded in our smartphones that are necessary to monitor the human activity and collect the data acquired by the sensor, there by utilising it to detect the fall.
The idea behind this project is to monitor the sensors like accelerometer, gyroscope for abnormal or irregular readings while running or walking which follows a particular pattern, when this reading is recorded the system will start a counter for specified time to check if there is any further activity for example back to walking, from the person who has fallen down, if there is no activity from that person then within that predefined time-frame the system will trigger an alert that is a message to a chosen contact as emergency contact, that this person is in danger and in need of help. For the second case if there is an activity for example, just the device has fallen and the person has picked up the device or he has fallen but he is not hurt able to walk back home safely and resumes the normal activity within that predefined time then the system will not trigger an alert message to the emergency contact.

II. Requirements

A. Functional Requirements

- Accessing the sensor using the system's operating system to obtain the data.
- Determine the fall based on the readings from the sensors and use the predefined time to detect any further activity and decide the future action.
- Create an alarm on the device if fall has occurred.
- Message service to the emergency contact provided based on the fall recognition from the user's smartphone.
- Sending the current location of the user along with distress signal.
- Authentication from the user that he is fine, in the form of an alert message asking him if he is fine. If there is no response from the user then it is considered that he is in distress and alert will be triggered.
- Visualisation of data. That is representing the data collected in readable form.



Project Name : Fall Detection System Using Sensors Embedded In Smartphones							
#BR	Business Require	#FR	Functional Requirement	Description	Test Scenario	Test Result	Signature
BR_1	Interface	FR_1	Login/Logout by user	User should be able to login to the application and also logout from the application.	Validation of user's credentials.		Vidya
		FR_2	Provide Contact Details	Should be able to provide emergency contact details and also personal details	During emergency, the contact provided by user as emergency must be used.		Vidya
		FR_3	Acknowledgement from user	Obtain appropriate acknowledgement from the user, that the user is well incase of fall detection	Provide and validate the acknowledgement by the system		Vineeth
BR_2	Activity Monitoring System	FR_1	Accessing the appropriate sensor to get data.	Read data from appropriate sensors like accelerometer to obtain data for fall detection	Validate the data obtained.		Vidya, Vineeth, Jathin
		FR_2	Fall Detection	Determine the fall based on the readings from the sensors and use the pre-defined time to detect any further activity and decide the future action.	Validate fall detection.		Vidya, Vineeth, Jathin
		FR_3	Alarm	Provide alarm when fall is detected	Validate the alarm sound on fall.		Vineeth
		FR_4	SMS Alret	Send sms alert to emergency contact	Test the sms alert when fall is detected		jathin
		FR_5	Location	Attach location along with sms alert	Test for accuracte location along with sms alert		Jathin

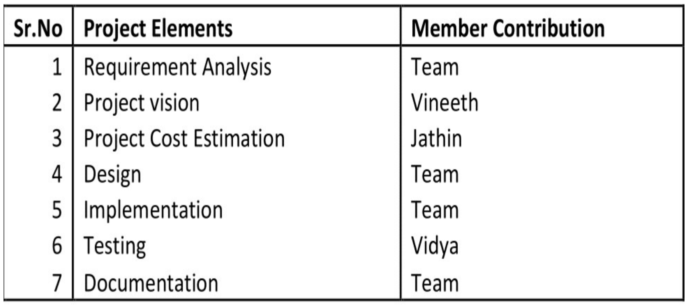
Figure 1. Requirement Tracebility Matrix

B. Non- Functional Requirements

- Primary requirement is to detect fall in a person, using appropriate sensors embedded in the smart phone.
- Real time monitoring of human activity.
- Accurate measurement of the activity by the sensor..
- Appropriate alert message to the emergency contact.
- Accurate tracking of location.
- The application must adapt varying display sizes of mobile screen.
- User friendly UI/UX.

III. Project Estimation

We are using Function Point Analysis and COCOMO model to predict the development time and effort for our project.The actual time required to develop this project will be based on the source lines of code, which is calculated by counting the number of inputs, outputs, inquiries, master files, interfaces required and also the Unadjusted Function Points. Finally, we will get 1609.3 lines of code from the calculation, considering



Sr.No	Project Elements	Member Contribution
1	Requirement Analysis	Team
2	Project vision	Vineeth
3	Project Cost Estimation	Jathin
4	Design	Team
5	Implementation	Team
6	Testing	Vidya
7	Documentation	Team

Figure 2. Resource Utilization

the language factor for JAVA which is assumed to be 38. The development time will be 4.2156 months/person or 3.955 person/month. [4]

- Source Lines of Code: SLOC = 1609.3
- Programmer Productivity:PM = 3.95530 person/month
- Development time: DEV = 4.2156 months

IV. Responsibilities of all team members

Figure. 2 shows the responsibility of all team members.The whole team is involved in major activities such as requirement analysis, design and Implementation.The activities such as project vision, cost estimation and testing is handled individually.

V. Security

- Privacy of user information. Example location, login data.
- Confidentiality of the data collected from the sensor.
- Requesting access permission to monitor the activity and device features.
- User authentication for accessing the app and data.
- Sensor data can not be accessed from outside the Smartphone.

VI. Reliability

- Accurate detection of fall based on sensor readings.
- The data collected from the sensor must be reliable for any further use.
- Activity monitored must be accurate.
- Capability to handle large amount of data and perform as intended.
- State of the application is uninterrupted by external interruption.

VII. Project Design

Considering the above mentioned requirements we have hashed out following design for our projects.

A. Use Case Diagram

Figure. 3 shows the interaction of application with the user. Since the application detects fall based on sensor reading and activity tracking, we have kept the user interaction to a minimal. User logins to the application and provides the emergency details. Apart from this the main feature would be checking the well being of the user in case of fall detected, in the form of an acknowledgement from user.

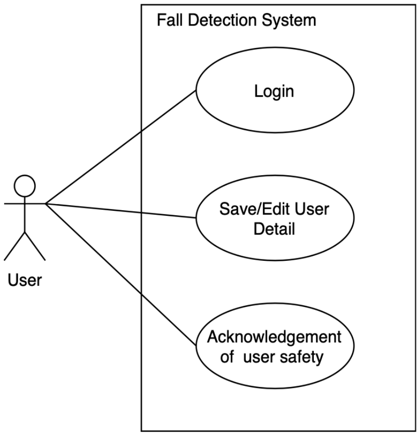


Figure 3. Use case diagram

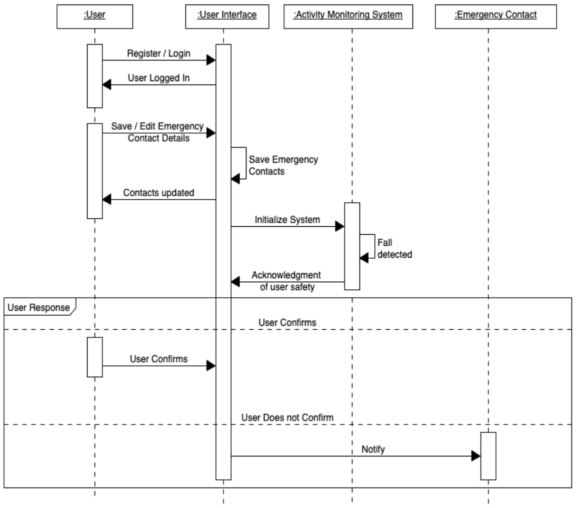


Figure 4. Sequence Diagram

B. Sequence Diagram

To understand the object interaction in a time sequence, we designed the following sequence diagram (Figure. 4) for our project. User, UI, Activity Monitoring System and Emergency contact are involved in the interaction. User initiates the Activity Monitoring System by logging into the application and providing emergency contact details. Once initiated the system tracks the activity and detects fall, in case of any the system requests the user for an acknowledgement that he is able to carry on. Here there are two possible cases. Case 1: If user confirms that he is well and that it is a false alarm then the system will not trigger alert to emergency contact. Case 2: If the user does not acknowledge, then the system considers the user has fallen down and needs assistant and contacts the emergency contact.

C. Class Diagram

Figure. 5 explains the project architecture, explaining the class files and the functions involved in the project.

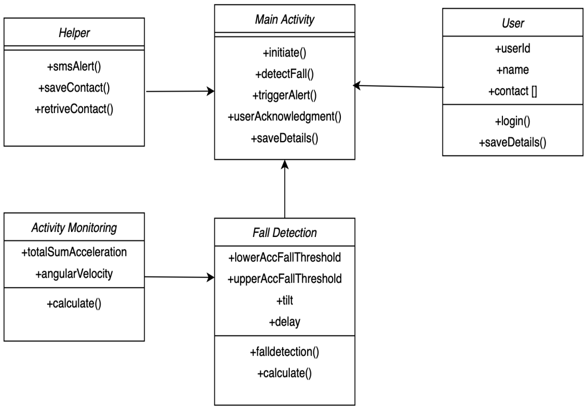


Figure 5. Class Diagram

VIII. Development Environment

As we are using android devices in our team, for this we are considering the android as our development environment. Using Android sensor framework we can access the available sensors and acquire the raw sensor data. Android provides dedicated hardware package for accessing the embedded sensors. It includes the following classes and interface, SensorManager, Sensor, SensorEvent, SensorEventListner.

A. Sensor Framework

The sensor framework provides certain classes and interface to perform variety of sensor related tasks. For example, finding out the available sensors on the device.

The package that includes sensor framework is android.hardware. It provides following classes and interface to access the sensors,

- SensorManager: This class is used to create an instance of the sensor service. It provides methods for accessing, listening and perform various action on the sensors.
- Sensor: This class is used to create an instance of a specific sensor.Sensor's capabilities are determined by this class.
- SensorEvent: Using this class the system creates a sensor event object, which is used to get information about the sensors. It include the following information, raw sensor data, type of the sensor that generated the event, timestamp, accuracy.
- SensorEventListener: Using this interface we can create two callback methods that receive notification when sensor values change or when sensor accuracy changes.

In a basic application we use these classes and interface to perform following tasks, identifying sensor and sensor capabilities and monitor sensor events.

IX. Hardware and Software

Following list specifies the hardware and software used in our project.

- Smartphone: Google Pixel
- Operating System: Android
- IDE: Android Studio
- Version Control: GitHub

A. Sensor

There are 3 categories of sensors,

- Motion Sensors: These sensors are responsible for monitoring the acceleration force and rotational forces along the three axes. This category includes,gravity sensors, accelerometers, gyroscopes and rotational vector sensors.
- Position Sensors: Physical positioning of the device is measured b these sensors. These sensors include orientation sensors and magnetometer.
- Environmental Sensors: Environmental parameters such as temperature, pressure, illumination and humidity are measured using these sensors. This includes, barometer, photometer and thermometer.

B. Sensors Used

There are various number of sensors that the android platform provides, for our project we'll be using the following sensors,
Accelerometer: Acceleration is the measurement of the rate of change in velocity, the accelerometer is an electromechanical device that is used for measuring acceleration forces. In the case of a sensor present in a mobile device, dynamic to sense movement or vibration. The accelerometer can operate in two ways the piezoelectric effect and the capacitance sensor. The Piezoelectric accelerometer uses microscopic crystal structures that become "squeezed" due to forces. These crystals create a voltage when squeezed and accelerometer interprets this voltage to determine the acceleration. Whereas the capacitance accelerometer changes the capacitance between microstructures located next to the device. If an accelerative force moves any of these structures the capacitance changes and accelerometer will read the capacitance and translates to a voltage for interpretation. In the android device the accelerometer measures the acceleration that is applied to a device on all three axes (x, y, and z), including the force of gravity. We are using this sensor to track the movement of the phone, and also to track the orientation and acceleration force of the device. In our application the acceleration along all the 3 axes in m2/s and the corresponding graph will be displayed, so if the phone is moved in any direction, the data and graph will change accordingly in real-time.
Gyroscope: The gyroscope sensor is a device that can measure and maintain the orientation and angular velocity of an object. They are more advanced than accelerometers. Accelerometers can measure only linear motion, whereas a gyroscope can measure tilt and lateral orientation. The Coriolis force concept is used in the Gyroscope sensors. To measure the angular rate, the rotation rate of the sensor is converted into an electrical

signal. Measures a device's rate of rotation around each of the three physical axes (x, y, and z). A gyroscope can be used for navigation and measurement of angular velocity. In our application we will show the orientation along the three 3 axes in rad/s along with the corresponding graph.

Magnetometer: The Magnetometer sensor reads the strength of the magnetic field around an Android device.The sensors measure the physical position of a device which reports the ambient magnetic field, sensor produces voltage which is proportional to the strength and polarity of the magnetic field along the axis each sensor is directed. The sensed voltage is converted to digital signal representing the magnetic field intensity. Magnetic field measured along the three sensor axes x, y, and z fields of "sensors_even_t.magnetic" and all values are in micro-Tesla(T).

X. Fall Detection Algorithm

The application uses a threshold based algorithm to detect falls. Threshold based algorithms are preferred as these algorithms require low computational cost and lower complexity. This fall detection algorithm as depicted in (Figure. 6), defines several parameters depending on the accelerometer and gyroscope outputs, and a decision is made using the threshold values for these parameters. The parameters used in the algorithm are The total sum acceleration vector SV and The angular velocity θ. The total sum acceleration vector is calculated by

$$SV = \sqrt{(A_x)^2 + (A_y)^2 + (A_z)^2} \qquad (1)$$

where A_x, A_y and A_z are the reading from the accelerometer along the x,y and z axes. The angular velocity θ is calculated by

$$\theta = \sqrt{(\omega_x)^2 + (\omega_y)^2 + (\omega_z)^2} \qquad (2)$$

where ω_x, ω_y and ω_z are readings from the gyroscope along the x, y and z axes.

When the user falls, the SV calculated using (Equation. 1) changes rapidly and the θ calculated using (Equation. 2) also increases along the fall direction. This application uses 4 thresholds vales to determine a fall.

- FT1 (lower acceleration threshold): When the user starts to fall down, the acceleration starts decreasing. There is a significant change in SV.
- FT2 (time-delay): When the user falls down, the application checks if user is down for FT2 seconds, to avoid false detection like if the user is bending down or if user wakes up after fall.
- FT3 (upper acceleration threshold): This corresponds to value after user hits the floor, which is a large pike on how hard the user has fallen down to the ground. FT3 is set to value more than 1.8g which is the impact of the user on the ground.
- FT4 (tilt): Corresponds to the tilt when the user falls on the floor. FT4 is set to a value greater than 70deg which proves that user has actually is down on the ground and avoids false detection such as if user sits on the sofa. If the calculated θ exceeds FT4 then it can be a a possible fall.

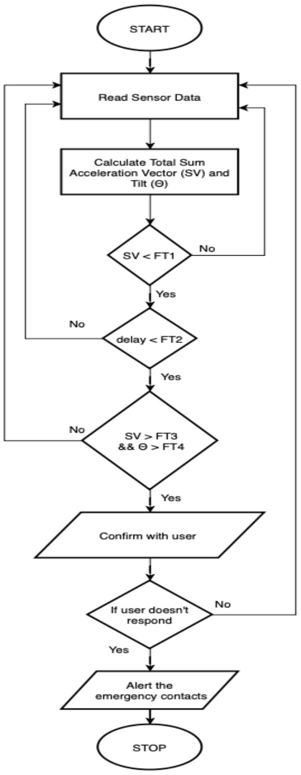


Figure 6. Fall Detection Algorithm Flowchart

At the end if the application has detected a fall and but the user was not effected because of the fall, to avoid a alarm in these cases the application brings up a notification for the user to acknowledge to confirm if the user has not fallen. If the user acknowledges that he has not fallen, the emergency contact is not notified, else a SMS sent to the emergency contact which comprises of the address of the user's location.

References

[1] Sensors: https://developer.android.com/guide/topics/sensors Accessed: 09.06.2020

[2] Gyroscope: https://ef.engr.utk.edu/hyperphysics/hbase/gyr.html Accessed: 09.06.2020

[3] Live Science: https://www.livescience.com/40102-accelerometers.html Accessed: 09.06.2020

[4] Cost Estimation, Padmaja, http://people.cs.ksu.edu/ padmaja/Project/CostEstimate.html, Accessed: 09.06.2020

[5] E. Thammasat and J. Chaicharn, "A simply fall-detection algorithm using accelerometers on a smartphone," The 5th 2012 Biomedical Engineering International Conference, Ubon Ratchathani, 2012, pp. 1-4, doi: 10.1109/BMEiCon.2012.6465471.

[6] F. Sposaro and G. Tyson, "iFall: An android application for fall monitoring and response," 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Minneapolis, MN, 2009, pp. 6119-6122, doi: 10.1109/IEMBS.2009.5334912.

[7] S. Abdelhedi, R. Bourguiba, J. Mouine and M. Baklouti, ”Development of a two-threshold-based fall detection algorithm for elderly health monitoring,” 2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS), Grenoble, 2016, pp. 1-5, doi: 10.1109/RCIS.2016.7549315.

[8] Bourke, Alan ÓLaighin, Gearóid. (2008). A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. Medical engineering physics. 30. 84-90. 10.1016/j.medengphy.2006.12.001.

[9] H. W. Guo, Y. T. Hsieh, Y. S. Huang, J. C. Chien, K. Haraikawa and J. S. Shieh, ”A threshold-based algorithm of fall detection using a wearable device with tri-axial accelerometer and gyroscope,” 2015 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), Okinawa, 2015, pp. 54-57, doi: 10.1109/ICIIBMS.2015.7439470.