FALL DETECTION RECOGNITION SYSTEM

Safety Critical Systems
Frankfurt University of Applied Sciences
Frankfurt am Main, Germany

Heer Vankawala (Matriculation Number: 1449039)
Ankur Kushwaha (Matriculation Number: 1427420)
Pankaj Patel (Matriculation Number: 1429826)
Pooja Tailor (Matriculation Number: 1449385)
Aerika Modi (Matriculation Number: 1448247)
Prahlad Roy Pareek (Matriculation Number: 1431542)

February 4, 2023

1. Introduction

Smartphones have sensors built in that allow us to extract data and conduct various human behaviour identification tasks. This research makes advantage of that concept to suggest a fall detection system using sensors built into smartphones. Nowadays smartphones are a basic need of society. From Children to elderly people, everyone uses smartphones these days. Smartphones are loaded with sensors that gather information from the outside environment. Then, depending on their needs, different smartphone applications use these data.

According to research, falls in elderly people are the second leading cause of fatalities and bodily harm. If an elderly person falls, the results could be life-threatening, debilitating, or severe organ damage. It is not advantageous that understanding how to utilize advanced detection and analysis tools to observe daily activity and recognize falls in the elderly has become crucial due to the rapid expansion of smartphones.

This project's goal is to develop a tool that can identify falls that occur to a person. This program makes use of the sensors that are built into smartphones to monitor human activity, gather data from the sensor, and use that data to detect falls. In a specific pattern, the accelerometer and gyroscope sensors are being monitored for abnormal or irregular readings; when this reading is taken, the system will start a counter for the allotted period to see if there is any additional activity.

2. Project Scope

In this project we are calculating the fall detection using acceleration and Gyroscope sensors of the mobile phone falling from a waist height of a person, then we send an alert to our prepared android application. In this application, the user has the option to send alerts to a specified emergency contact. If there is a false fall, a buffer time(15 sec) will be provided to the user to cancel the alert message.

3. Requirements

Functional:
- Collection of the sensors' data.
- A time buffer is given to the user for a false fall.
- Alert the emergency contacts provided by the user.
- Sending the current location of the user.

Non-Functional:
- The main requirement is to identify falls in a person using the right smartphone sensors.
- Observing human activity in real-time.
- Accuracy of the sensor's activity measurement.
- The Application must be user-friendly.

4. Process Model

Rational Unified Process Model: A process for developing object-oriented software is called Rational Unified Process (RUP).An Iterative Process for Projects. RUP provides a systematic approach to allocating and managing activities and responsibilities within a development organization by segmenting the project life cycle into several phases. Software developers can deal with the varying needs of clients or the project itself. RUP emphasizes the significance of accurate documentation as well. It also boosts software engineers' productivity.

Steps in our (U) Process Model

Requirement Analysis.

Design and Implementation (Use case diagram).

Coding.

Testing (Integration Testing, System testing).

Deployment.

IEC61508 Safety Standard -- Ensures that our software meets IEC61508   safety standards.

5. Project Estimation

Functional Point Estimation is calculated using two factors
- I.   Unadjusted Functional Points(UFP)
- II.   Complexity Adjustment Factor(CAF)

Calculating the Unadjusted Functional Points (UFP)

Function Type	Simple	Average	Complex	Considered for product	Count	Total
Inputs	3	4	6	3	2	6
Outputs	4	5	7	5	2	10

Enquiries	3	4	6	5	3	15
User File	7	10	15	10	3	30
Interface	5	7	10	7	1	7
						Total = 68

Calculation of Complexity Adjustment Factor (CAF):

Sr. No	Characteristics	(0-5)
1.	Data Communications	3
2.	Distributed Data Processing	0
3.	Performance	3
4.	Heavily Used Configuration	0
5.	Transaction Rate	5
6.	Online data Entry	2
7.	End User Efficiency	4
8.	Online Update	0
9.	Complex Processing	5
10.	Reusability	2
11.	Installation ease	3
12.	Operational ease	3
13.	Facility change	2
14.	Multiple sites	0
		Total =32

Complexity Adjustment Factor (CAF) = 0.65+(0.01*Fi)

CAF = $0.65 + (0.01 * 32) = 0.97$

FPC=UFP∗CAF = 82*0.97 = 65.96 = 66.

If we consider that, one function point takes 7 hours of work to develop in JAVA language, 66* 7= 462hrs. The estimate for developing the application would take about 462 hours of work.

For Each person, 462/6 = 77 total hours For Each person per week, we have 3 months = 12 weeks, therefore, ~ 6.5 hours/week.
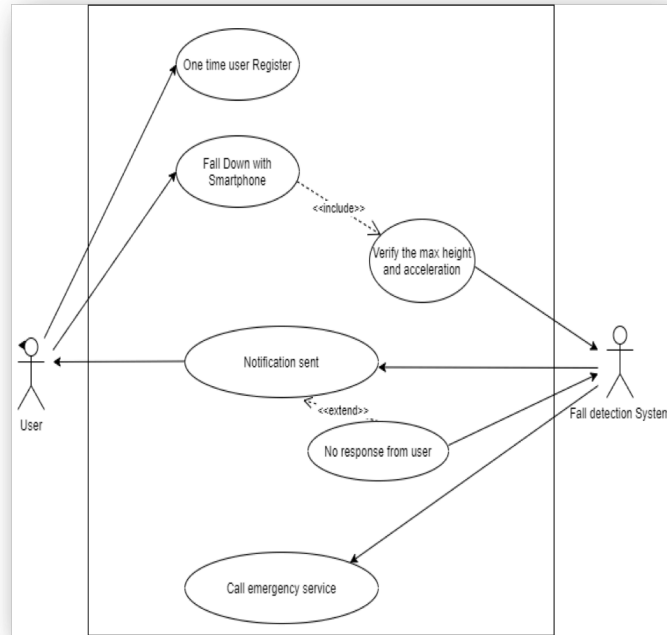
6. Timeline Chart

Activities	Week47	Week48	Week49	Week50	Week51	Week52	Week01	Week02	Week03	Week04
Feasiblity and Planning										
Design and Validation										
Build and Validation										
Testing and Validation										
Customer Acceptance and Validation										
Deployment										

7. Team Structure

Sr. No.	Project Tasks	Task Assignment
1	Requirement Analysis	Ankur , Heer
2	Project Scope	Prahlad , Ankur
3	Project Cost Estimation	Prahlad, Aerika
4	Mathematical Model	Heer, Prahlad, Ankur
5	Safety Plan	Ankur, Prahlad, Heer
6	Coding and Process	Pooja,Aerika ,Heer
7	Designing model	Pooja, Aerika, Heer
8	Hazard Analysis	Pankaj , Ankur ,Heer, Prahlad
9	Documentation	Prahlad , Pankaj

8. Use Cases :

Use Case in Text Format –

| Use Case ID	Fall_Detection_1
Use Case Name	User Authentication Verification
Description	This use case allows user to register themselves to our Fall Detection Android application by submitting personal details such as name, contact details, blood group and contacts to send alert message at the time of emergency.
Primary Actor	End User
Secondary Actor	Android Application
Include Use Cases	
Preconditions	1. Android Application 2. Internet connection
Main Flow	Step 1 – User click on registration page Step 2 – Enter Mandatory Details for registration (Name, contact Details, blood group, additional contact details for alert notification. Step 3 – After Successful registration user must be able to login to the android application.
Alternate Flows	User must be notified if authentication failed.

Use Case ID	Fall_Detection_2
Use Case Name	Transform Acceleration and angular velocity from Android device to MATLAB for analysis
Description	This use case is used for sending data from the MATLAB Mobile application to MATLAB for further analysis. Data like Acceleration and velocity along with respective time will be sent.
Primary Actor	MATLAB mobile app
Secondary Actor	MATLAB
Include Use Cases	Fall_Detection_1
Preconditions	1. Android Application 2. Internet connection 3. MATLAB Mobile
Main Flow	Step 1 – User successfully logged into the app. Step 2 - Details of the user's movement that is acceleration and angular velocity along three axes – x, y and z to MATLAB using MATLAB mobile app.

| Use Case ID	Fall_Detection_3

Use Case Name	Fall detection calculation
Description	This use case is used for calculating the Magnitude of Acceleration and Angular Velocity. After comparing with the Threshold Values, we detect there is a fall or not.
Primary Actor	MATLAB
Secondary Actor	
Include Use Cases	Fall_Detection_1, Fall_Detection_2
Preconditions	1. Internet connection 2. MATLAB Algorithm
Main Flow	Step 1 – MATLAB Algorithm calculates the linear acceleration using the provided accelerometer and gyroscope data. Step 2 – It compares the calculated Acceleration with threshold limits. Step 3 – After comparing it tells us whether there was a fall or not. Step 4 – It send the data to Android App.

| Use Case ID	Fall_Detection_4
Use Case Name	Send alert notification
Description	These use cases allow the app to send notifications to customers in case of fall detection.
Primary Actor	Android app
Secondary Actor	MATLAB
Include Use Cases	Fall_Detection_1, Fall_Detection_2, Fall_Detection_3
Preconditions	1. Android Application 2. Internet connection 3. Algorithm
Main Flow	Step 1 – Android app consumes data from MATLAB. Step 2 – In case of positive response from MATLAB (Fall Detected = True) then system must send notification to emergency contact details. Step 3 – System must not send emergency alert in all other scenario.

9. **Data Flow Diagram**
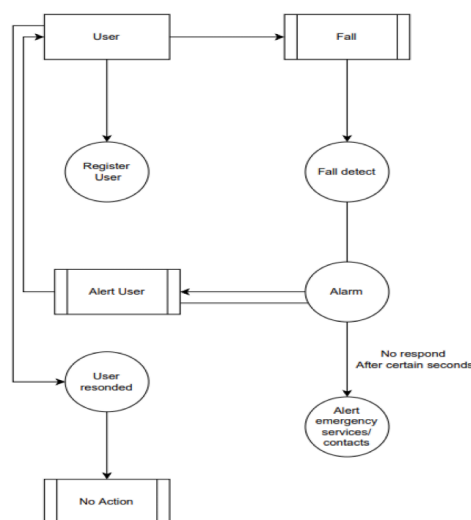
10. **Sequence Diagram**
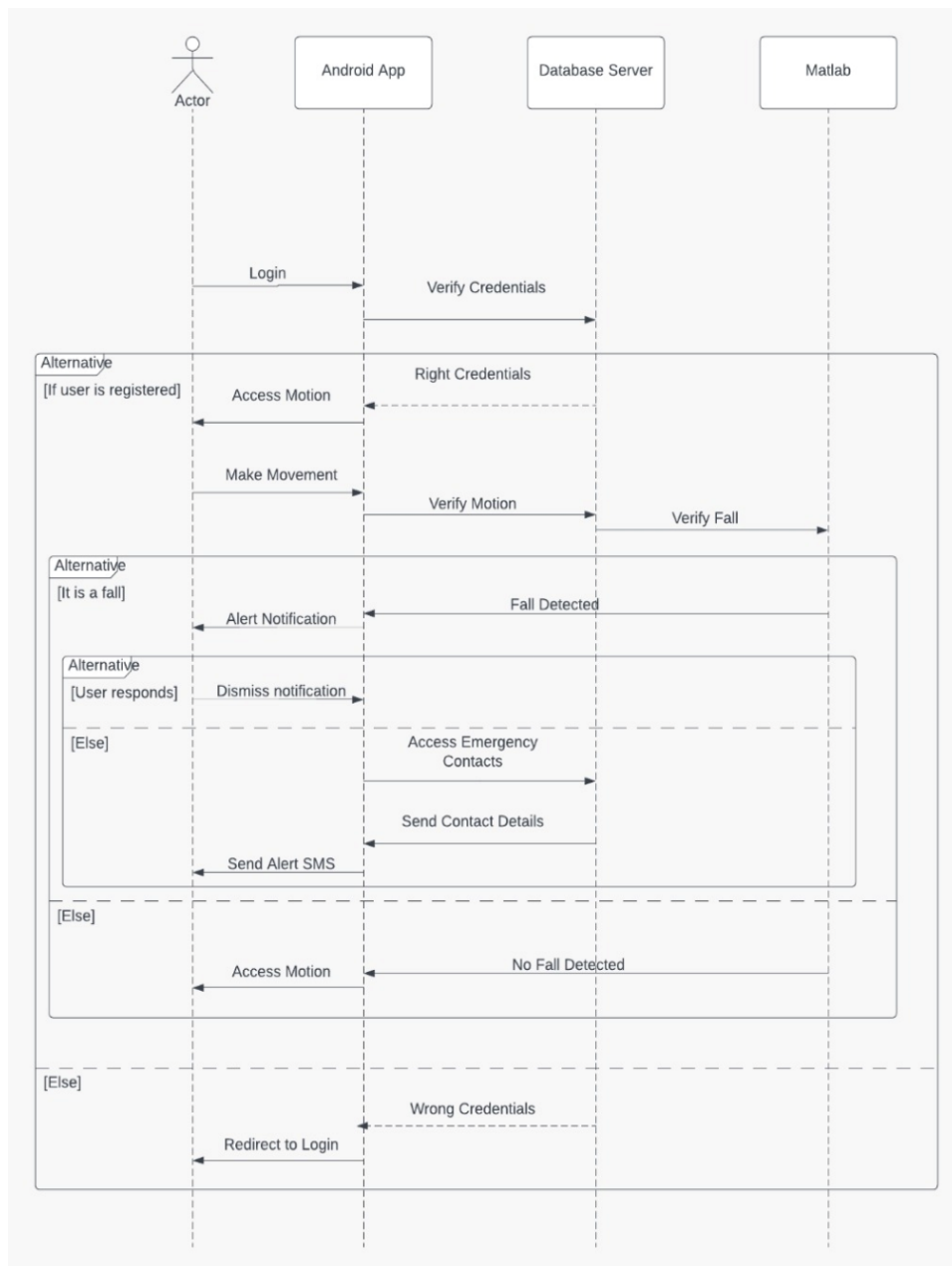
11. Class Diagram
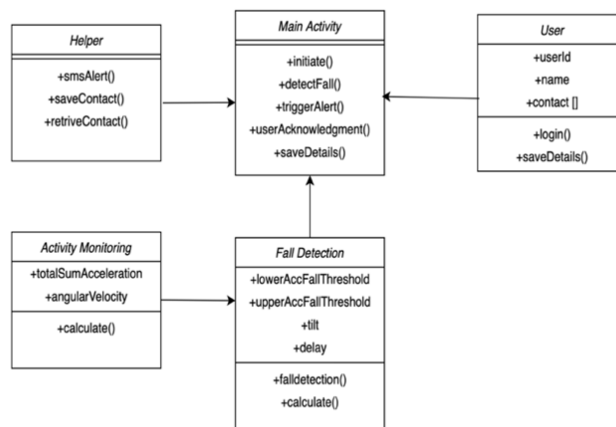
12. Fault Tree Analysis Diagram
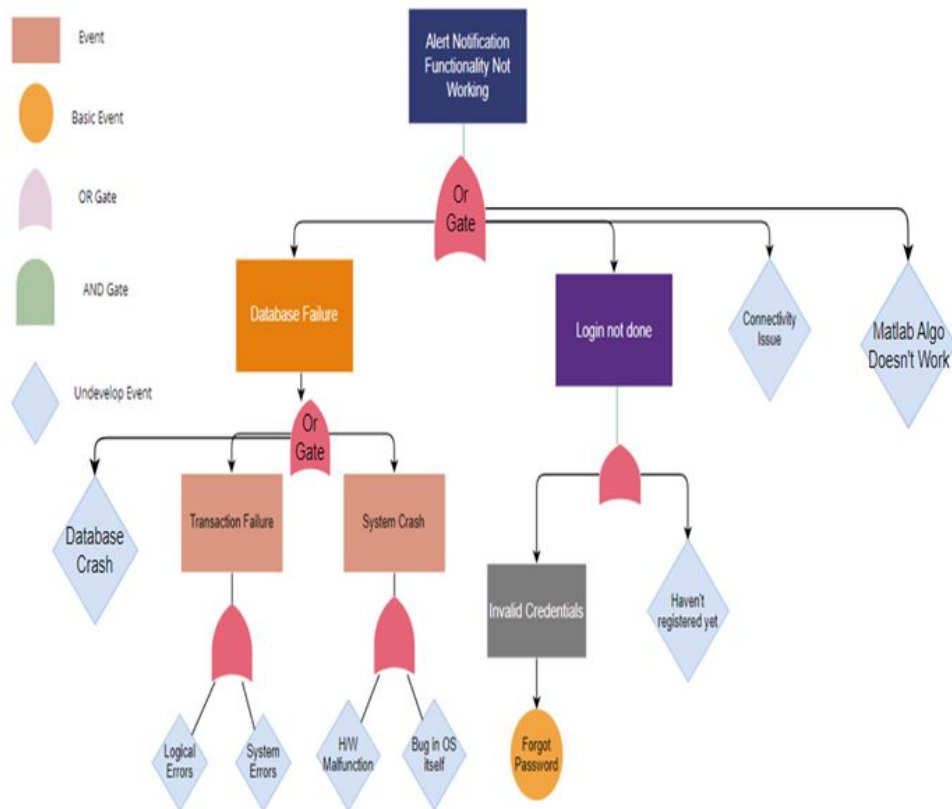
13. STAMP Analysis

13.1 Hazards

H1 - Accelerometer and Gyroscope data not sent to MATLAB

H2 - Wrong algorithm calculation.

H3 – Alert notification received late

H4 – Alert notification not received

H5 – False alarm raised

H6 - The application does not provide the correct historic value

H7- Alert not sent to the user

13.2 Safety Requirements and Safety Constraints

Hazards	Safety Requirements	Safety Constraints
H1	The correct sensor data should be given to MATLAB	All the accelerometer and Gyroscope data should be sent to MATLAB using MATLAB mobile at a fixed frequency.
H2	Correct MATLAB calculation should be given	MATLAB application should calculate the corrective values of magnitude of acceleration and angular velocity and compare it to the threshold values
H3	Alert Notification should receive on time	The application should process the output received from MATLAB and should give the alarm within a time frame as per Service Level Agreement(SLA). (0 – 15 seconds)
H4	Alert notification must be received	Alert notifications should be received in the application so that user could cancel the false fall detection alerts.
H5	Emergency Contact should not receive the false alerts	The application should provide the functionality to the user to cancel the false alerts of falls.

| H6	The application should provide the correct historical data	The application should be using the Database to record the past alerts which were sent to contactable person in case of fall.
H7	Alert must be sent to emergency Contact	The application should send the alert to the emergency contact.

14. Safety Plan

The safety plan was created following a thorough examination of the system's hazard analysis results when it is subjected to specific environmental and technical conditions. We incorporated several safety mechanisms that were considered safety constraints within the system if the system failed because a hazard describes a system's state that should not occur for safety reasons. We conducted a preliminary hazard analysis to identify probable hazards and their contributing flaws in a systematic manner. We then chose which flaws to fix and how to mitigate them after discussing them with each other. The mitigations were outlined as mandatory safety measures for the system.

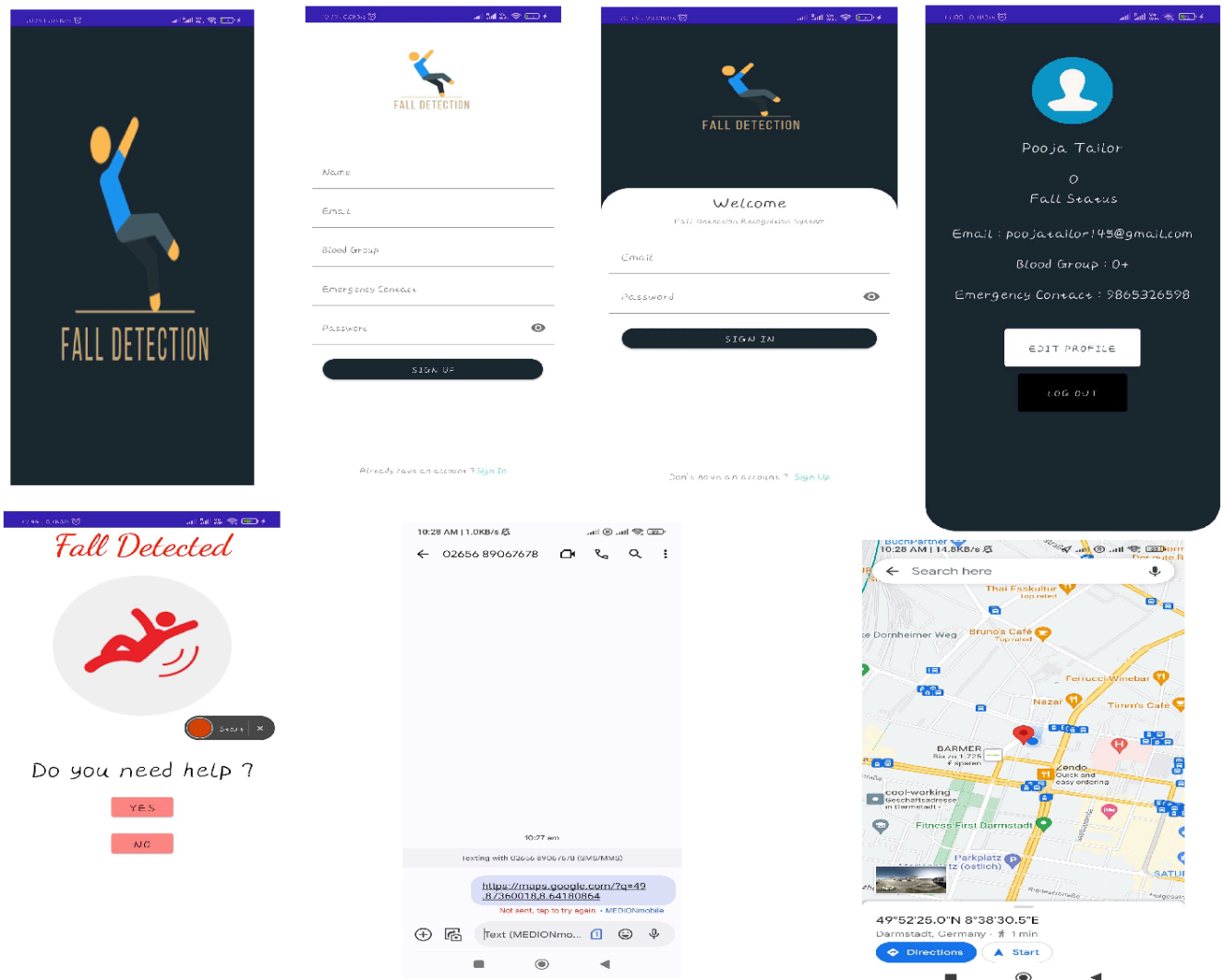| Safety Requirement ID:	H1,H2
Hazard Category:	Incorrect Magnitudes of acceleration and angular velocity calculation.
Hazard:	Wrong data was fetched to update the results.
Root Cause:	▪ Improper calculation of the results.
	▪ Improper data from mobile sensors.
Risks:	Inaccurate fall detects predictions due to incorrectly calculated values.
Safety Mechanism:	Ensure data archival is proper from smartphone sensors and the Mathematical Calculation is proper.

Safety Requirement ID:	H3,H4,H5
Hazard Category:	Coding or Programming bugs.
Hazard:	Unhandled Exceptions
Root Cause:	Code smells.
Risks:	Application crashes.
Safety Mechanism:	Ensure proper coding guidelines are followed during development and testing.

Safety Requirement ID:	H6
Hazard Category:	Database errors
Hazard:	1. Configuration file issues, 2. Poor performance due to low memory, 3. Connection failure
Root Cause:	1. Improper database deployment, 2. Hardware issues or deadlocks arise due to improper user management.
Risks:	The unavailability of the database will result in an error in user registration. Ensure
Safety Mechanism:	the database is deployed properly, 2. Ensure orderly execution of optimized queries to avoid deadlocks and scheduled maintenance checks should be carried out to ensure the proper functioning of the database.

15. STAMP Diagram

8
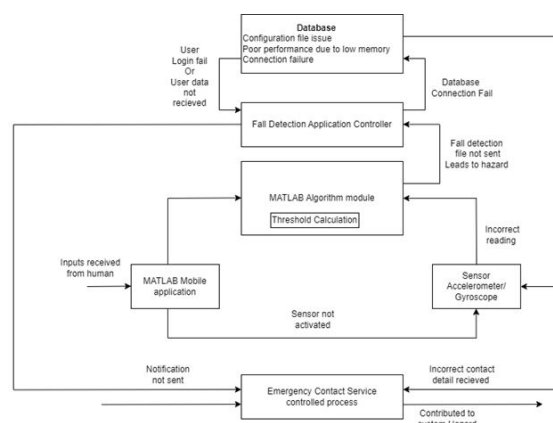
16. HMI Model

17. GitHub Link

https://github.com/adaptablePrahlad/FallDetectionUsingMatlab

18. References

1. R. May, "Safety standards including IEC 61508," Open Control Systems - The Importance of Industrial Standards, 2004, pp. 6/1-615, Doi: 10.1049/ic:20040130.
2. https://www.softwaretestingclass.com/software-estimation-techniques/
3. https://www.geeksforgeeks.org/software-engineering-cocomo-model/
4. https://www.61508.org/knowledge/what-is-iec-61508.php
5. Leveson, Nancy (April 2004), "A New Accident Model for Engineering Safer Systems".
6. J.D. Andrews, S.J. Dunnett Event-tree analysis using binary decision diagrams, 2000.
7. Balgos, V.H., Signature, V.H.B., January, M., Eikema, Q.D.V., Thesis, H., Systems, E., Accepted, D., Hale, P., Program, M.: A Systems Theoretic Application to Design for the Safety of MedicalDiagnostic Devices (2012), http://sunnyday.mit.edu/safer- world/Balgos-thesis.pdf
8. What is Fault Tree Analysis, https://sixsigmastudyguide.com/fault-tree-analysis/
9. Reykjavik University: What is STAMP/STPA, https://en.ru.is/stamp/what-is-stamp/#::text=STPA (Systems-Theoretic Process Analysis, used across diverse industry sectors.

10. Stoop, J., Benner, L.: What do STAMP-based Analysts Expect from Safety Investigations? Procedia Eng. 128, 93–102 (2015), http://dx.doi.org/10.1016/j.proeng.2015.11.508

11. A. S. Kapse, S. Shoba, R. Tamuli, A. Sinha and S. Samantara, "Android Based Fall Detection and Tracking App for Aged People," 2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS), Coimbatore, India, 2022, pp. 1113-1116, doi: 10.1109/ICAIS53314.2022.9743024.

12. J. Liu and T. E. Lockhart, "Development and Evaluation of a Prior-to-Impact Fall Event Detection Algorithm," in IEEE Transactions on Biomedical Engineering, vol. 61, no. 7, pp. 2135-2140, July 2014, doi: 10.1109/TBME.2014.2315784.

13. Cao, Yabo & Yang, Yujiu & Liu, WenHuang. (2012). E-FallD: A fall detection system using android-based smartphone. Proceedings - 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2012. 10.1109/FSKD.2012.6234271.

14. Nancy G. Leveson, "Engineering and Operating Safer Systems Using STAMP," in Engineering a Safer World: Systems Thinking Applied to Safety , MIT Press, 2012, pp.171-180.

15. Nancy G. Leveson, "Using STAMP," in Engineering a Safer World: Systems Thinking Applied to Safety , MIT Press, 2012, pp.169-169.

16. S. S. Brilliant, J. C. Knight and N. G. Leveson, "Analysis of faults in an N-version software experiment," in IEEE Transactions on Software Engineering, vol. 16, no. 2, pp. 238-247, Feb. 1990, doi: 10.1109/32.44387.