TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS
DEPARTMENT OF
ELECTRONICS AND COMPUTER ENGINEERING

OOSE CASE STUDY REPORT ON
FALL DETECTION SYSTEM

A COURSE CASE STUDY REPORT SUBMITTED TO THE DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING IN FULFILLMENT OF THE REQUIREMENTS FOR THE PRACTICAL COURSE ON OBJECT ORIENTED SOFTWARE ENGINEERING [CT 657]

Submitted by

Akanksha Giri (076BEI004)
Devraj Parajuli (076BEI013)
Prayag Man Mane (076BEI027)
Sanim Kumar Khatri (076BEI037)

Submitted to

Asst. Prof. Santosh Giri
Department of Electronics and Computer Engineering
IOE, Pulchowk Campus

20th Falgun, 2079

TABLE OF CONTENT

LIST OF FIGURES

1. FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

1.1 Functional requirements

R.1: Detect Fall

Description: The model is used to predict fall situations based on the data processed by Pi Pico from the raw data obtained from the MPU6050 sensor.

R.1.1: Read acceleration and angular velocity data

- ➢ Input: Raw change in capacitance data from MPU6050
- ➢ Processing: Convert the raw data into relevant processable data using sensitivity and other constraints.
- ➢ Output: Processed 6-axis acceleration and gyroscope value

R.1.2: Predict Fall

- ➢ Input: Processed Acceleration and angular velocity data from MPU6050 sensor.
- ➢ Processing: Feed the input data into the ML model to get Fall/Not fall
- ➢ Output: Fall or Not Fall

R.2: Alert

Description: The RF modules establish the communication between transmitter and Receiver. The Receiver decodes the encoded data received from the transmitter and rings the buzzer for one minute if the data it received is fall-detected data.

R.2.1: Alert

- ➢ Input: Fall Detected
- ➢ Output: Ring the buzzer

Traceability Matrix

Req ID	1.1	1.2	2.1
1.1			
1.2	U		
2.1	R	U	

1.2 Non-functional requirements

R.1: Performance requirements

R.1.1: The raw data shall be processed in the model within 3 seconds.

R.1.2: After a fall is detected, alerting shall take no longer than 2 seconds.

R.2: Design Constraints

R.2.1: The System shall be developed using Python.

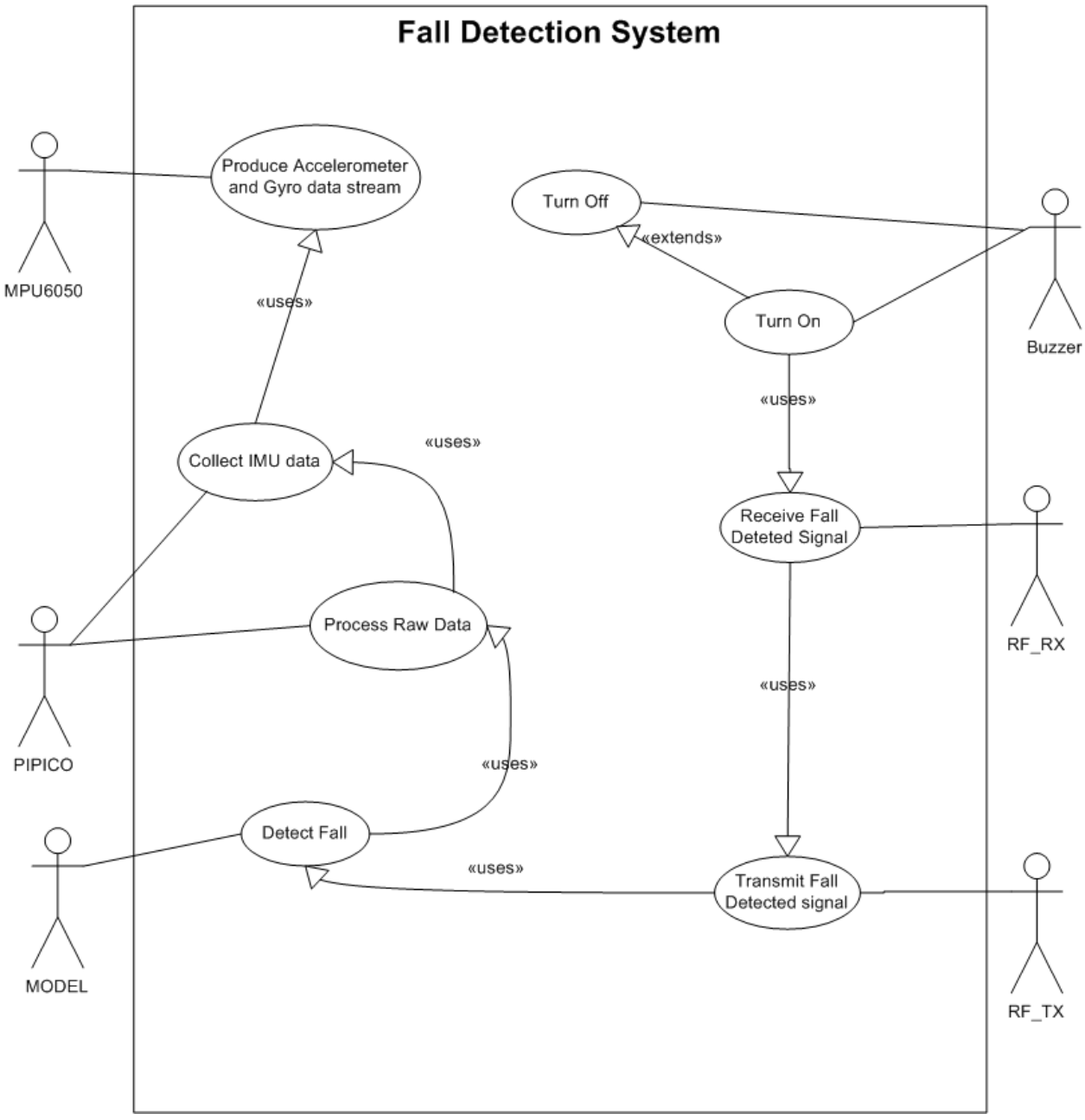
R.2.2: The collected data shall be logged in a CSV file.

R.3: Availability

R.3.1: The system shall be operated 24/7.

2. UML USE CASE DIAGRAM

2.1 Diagram



2.2 Description

Use case: Produce accelerometer and gyroscope data
Actors: MPU6050
Pre condition: The sensor should be powered
Post Condition: The sensor has produced accelerometer and gyroscope data
Description:

The sensor is activated and starts reading accelerometer and gyroscope data, which is measured by the physical movement and orientation of the user. The sensor sends it to the Raspberry-PiPico connected to it.

Use Case: Collect IMU data
Actors: PIPICO
Pre Condition: The board should be connected to the IMU via I2C interface
Description:

The board activates the I2C interface and initiates communication with the sensor. The board sends commands to the sensor and requests data. The board reads raw data sent by the sensor.

Use Case: Process Raw Data
Actors: PIPICO
Post Condition: The raw data has been processed to extract useful features for machine learning.
Description:

The raw data is processed to perform filtration, feature extraction, and calibration. The biases and offsets are corrected from the raw data. The data is transformed in a common range.

Use Case: Detect Fall
Actors: Model
Pre Condition: The sensor data is being collected and processed by the system
Post Condition: The system has detected a fall event and sent the signal to the RF module.
Description:

The system continuously monitors system data for significant change. The preprocessed data is passed to the machine learning model for analysis. The model analyzes the data to determine whether a fall event has occurred.

Use Case: Transmit detected signal
Actors: RF_TX
Pre Condition: The RF module is properly connected to the system and configured for communication.
Description:

The RF module encodes fall detected status from the model and transmits it.

Use Case: Receive detected signal
Actors: RF_RX
Description:

The RF module decodes the received signal into the relevant data.

Use Case: Turn On
Actors: Buzzer
Pre Condition: RF_RX received a fall detected signal.
Description:

The buzzer is powered on at a certain frequency for one minute.

Use Case: Turn Off
Actors: Buzzer
Pre Condition: Buzzer is turned on.
Description:

The buzzer is turned off after one minute of being powered on.

Overall overview of the use case diagram:

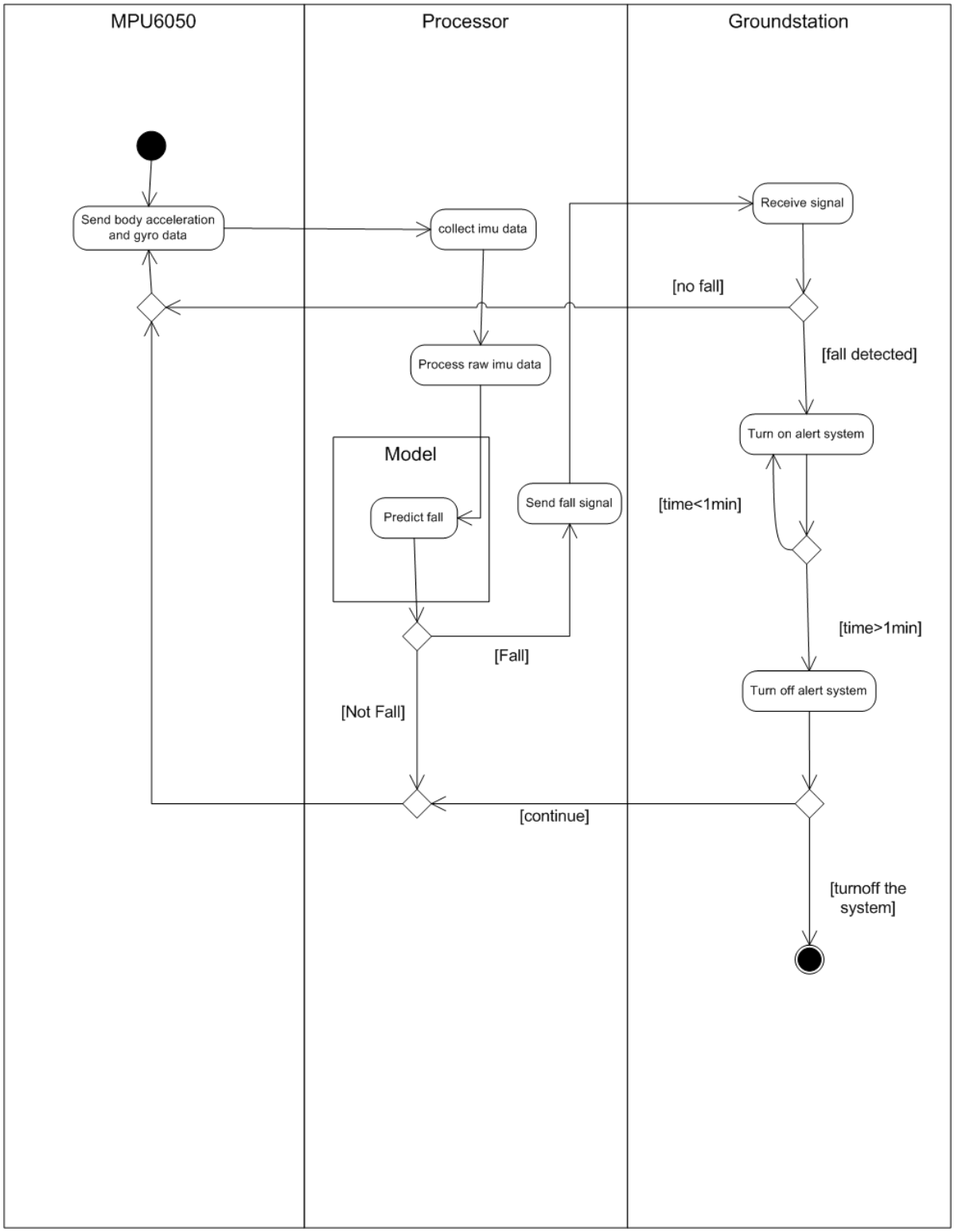
MPU6050 produces the accelerator and gyro data from its 3-axis capacitance values. Then the PIPICO collects the produced data from the MPU6050. After collecting the data, it processes the data into relevant values using sensitivity values and other constraints.

The model detects/predicts falls on the basis of the processed data from PIPICO, it is necessary to process the data to predict the fall.

The rf_tx transmits a fall detected signal if a fall is detected by the model. The rf_rx receives the fall detected signal if it is transmitted. Only when the signal is received the buzzer is turned on. Buzzer can be turned off if it has been turned on.

3. UML ACTIVITY DIAGRAM

3.1 Diagram



3.2 Description

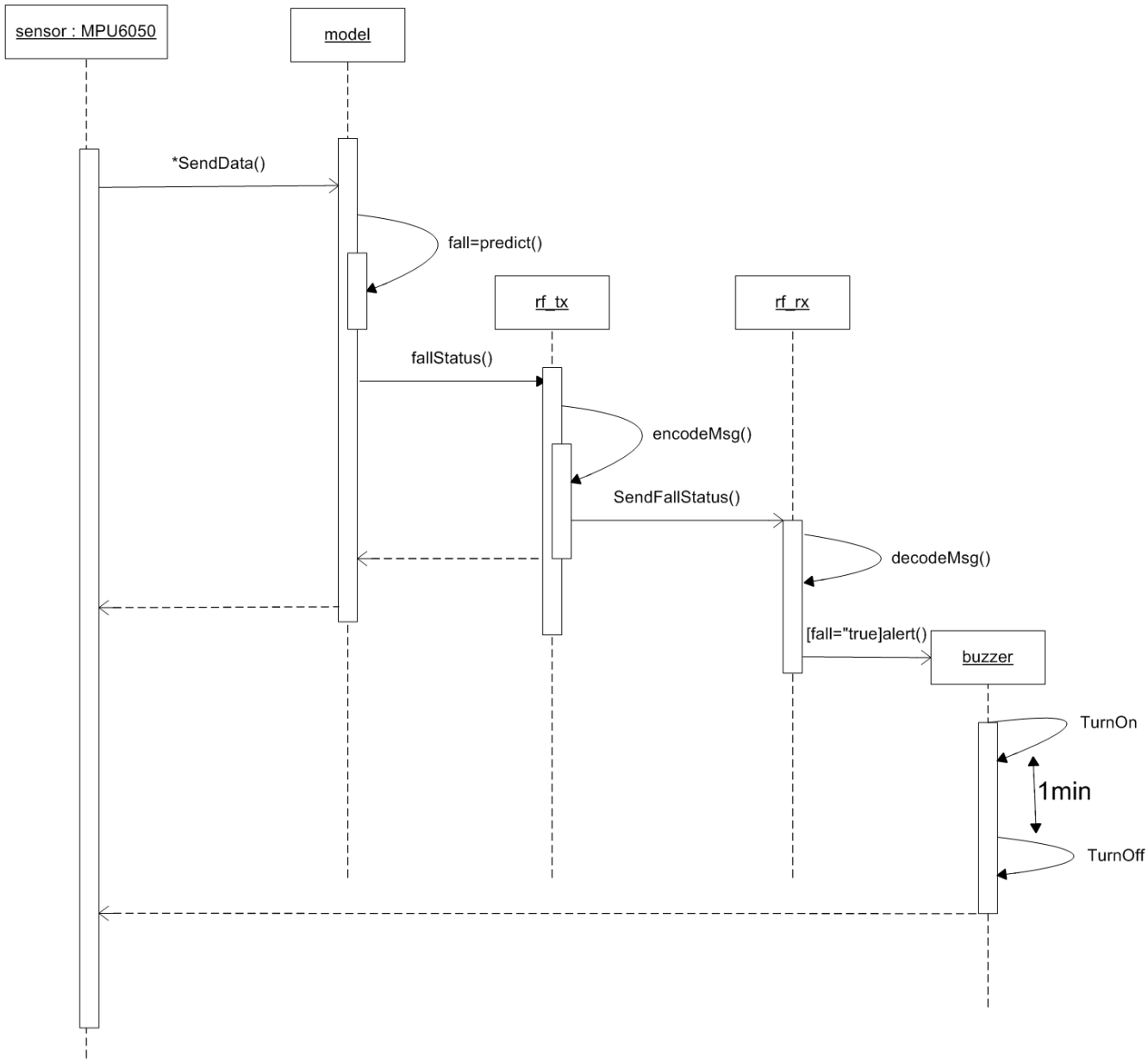
The activity diagram begins with the MPU6050 sensor gathering body acceleration and gyroscope data. This raw data is then collected and processed by the Raspberry Pi Pico processor. Within the processor, a model generates a 'fall signal' based on the processed data to predict if a fall is imminent. This 'fall signal' is then continuously transmitted to the ground station for further analysis.

If the ground station detects a fall based on the 'fall signal', it activates the alert system. The alert signal remains active for one minute before turning off automatically. However, if the ground station does not detect a fall, no action is taken.

The product continuously monitors the body it is attached to, detecting potential falls until it is turned off. Overall, this system utilizes sensor data, processing power, and transmission capabilities to predict and detect falls, with a built-in alert system to notify others if a fall occurs.

4. UML SEQUENCE DIAGRAM

4.1 Diagram



4.2 Description

The 6-axis accelerometer and gyroscope sends the data as a change in capacitance which is equivalent to some voltage level and the processor processes the raw data into relevant acceleration and gyro data which are the attributes of the object ‘sensor’ of class ‘MPU6050’. The ‘sensor’ sends the message SendData() to the model iteratively. In each iteration, the model predicts if the data signifies there is a fall or not.

The model then sends the status of the fall to the rf_tx and returns a message to the sensor so the sensor continues sending data iteratively. Then the rf_tx encodes the data itself and sends the sendfallstatus() message to rf_rx. The rf_rx sends a decode() message to itself. If the condition fall=’true’ is met, then the module creates a buzzer object.

The buzzer is turned on till 1 minute, when the time is exceeded the buzzer object is destroyed and the message is returned to the sensor object.

5. UML CLASS DIAGRAM

5.1 Diagram