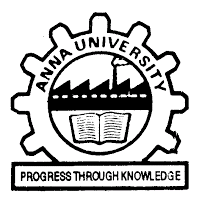
**ANNA UNIVERSITY**

**MADRAS INSTITUTE OF TECHNOLOGY**

**Chromepet, Chennai-600044.**

****

**IT8611-CREATIVE AND INNOVATIVE PROJECT**

**MONITORING SYSTEM FOR DIFFERENTLY ABLED**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

***Submitted by***

Srivatsan Narasimhan

**TABLE OF CONTENETS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **CHAPTER** | **PAGE NO.** |
|  | **ABSTRACT** | **4** |
| **1.** | **INTRODUCTION** | **5** |
| **2.** | **LITERATURE SURVEY** | **6** |
| **3.** | **REQUIREMENTS** | **7** |
| **4.** | **MODELLING** | **8** |
| **5.** | **HARDWARE REQUIREMENTS** | **10** |
|  | 5.1.ARDUINO UNO | **10** |
|  | 5.2.MPU6050 | **15** |
|  | 5.3.GSM MODULE | **17** |
|  | 5.4.BUZZER | **19** |
| **6.** | **SOFTWARE REQUIREMENTS** | **20** |
|  | 6.1.ARDUINO IDE | **20** |
|  | 6.2.ANDROID STUDIO | **21** |
|  | 6.3.IP WEBCAM | **21** |
|  | 6.4.IVIDEON APPLICATION | **22** |
| **7.** | **WORKING OF FALL DETECTOR** | **24** |
| **8.** | **FALL DETECTOR ANDROID APPPLICATION** | **25** |
| **9.** | **SCREENSHOTS** | **26** |
| **10.** | **CONCLUSION** | **28** |
| **11.** | **INFERENCES** | **29** |
| **12.** | **REFERENCES** | **30** |

**ABSTRACT**

It is highly essential to monitor the differently-abled when it comes to the terms of safety. Due to their physical or mental weakness, they have a high risk of resulting in situations where they require assistance. One such instance is an old person falling down from a wheelchair. The system uses accelerometer and gyroscope sensor to detect person movements. This smart device could be mounted on their hand or attached to a wheelchair for detection. A microcontroller is also embedded in this system to constantly transmit the data associated to acceleration. Now the system keeps monitoring for fall detection or any abrupt movement changes in person. A sudden abrupt change would be treated as a fall because of the jerk created. System detects the person has fallen and automatically triggers alert through GSM module to alert the family member of the person about the situation instantly. An SMS will be sent to the family member stating that a fall has occurred. A call alert will also be made by the GSM module. In addition to this an android app has been created for alerting the caretaker by sending an SMS and a video monitoring and recording facility has been provided if in case the family member wishes to monitor directly.

**CHAPTER 1**

**INTRODUCTION**

The fact that when people turn old, they are said to have entered into their second childhood. They need to be taken care like one takes care of a baby just born. They are said to be senile. Unless the person is monitored every moment of the day, one may never know what movement might take place next, and the consequences followed by it is undetermined. One such case is when a person falls onto the ground. It is not possible to monitor their movements without being in the same location. Hence, this project introduces you to the smart technology, where one could monitor even in the absence of him/her in that location. We may require several basic amenities for it, like, Mobile Network, Internet and so on.

When a person is not in the same location as the individual who needs to be taken care of, he/she needs to monitor the movements continuously, maybe using Surveillance Camera. But, practically, he/she may look into the surveillance footage only when the caretaker is idle or has the thought of looking into the footage. He/she needs something that could notify when the individual who is sick, has fallen down.

We have utilized Arduino along with gyroscope sensor and accelerometer that senses the fall made by the individual. Along with sensors used for fall detection, we have bestowed a GSM module, for the device to communicate and let the caretaker know that the fall had been detected. Following this, the caretaker makes use of an android application which has several facilities that could be help him take quick steps for the incident occurred.

**CHAPTER 2**

**LITERATURE SURVEY**

Tutorials Point is an online tutorial website that provides tutorials for android,C,C++,Java,and other languages too. Tutorial point solved our questions related to Android Application Development. Our various doubts such as database connectivity, SMS module implementation and other doubts were cleared by Tutorials point[1].

Stack Overflow is a question and answer site for professional and enthusiast programmers. It is built and run as a part of the Stack Exchange network of Q&A sites. It helped us to solves issues regarding finding the package name of other applications so that we can open other applications from our application[2].

Quora is a [question-and-answer site](https://en.wikipedia.org/wiki/List_of_question-and-answer_websites) where questions are asked, answered, edited and organized by its community of users. Quora helped us decide whether to use Arduino or Raspberry Pi to implement our project[3].

Android Developers which provides classes that describes how to accomplish a specific task which code samples that can be re-used. This was used for doubt clarification [4].

Ivideon is a cloud-based video surveillance solution for business and the home. Ivideon is easy to set up, maintain, and scale, no matter how many locations you have, using this we implemented the live Streaming functionality [5].

**CHAPTER 3**

**REQUIREMENTS**

**HARDWARE REQUIREMENTS:**

* Arduino Uno
* MPU 6050
* GSM module(SIM900A)
* Buzzer
* Jumpers

**SOFTWARE REQUIREMENTS:**

* Arduino IDE
* Android Studio
* IP Webcam(camera to server)
* Ivideon application(server to Fall Detector app)

**CHAPTER 4**

**MODELLING**

**Data Flow Diagram:**

**DFD0**

GYRO SENSOR

BUZZER

I/P

GSM MODULE

FALL DETECTED

FALL DETECTED

**DFD1:**

MOBILE

CALL

FALL OR JERK

BUZZER

I/P

FALL

FALL MESSAGE

MOBILE

**SEQUENCE DIAGRAM:**

SENSOR

DEVICE

MODULE

INPUT DEVICE

GYRO SENSOR

NPUT CALL

MOBILE

ARDUINO

MOBILE

GSM MODULE

BUZZER

MESSAGE

**CHAPTER 5**

**HARDWARE REQUIREMENTS**

**5.1. ARDUINO UNO:**

Arduino Uno is a microcontroller board based on the ATmega328. The ATmega328  is a single-[chip](https://en.wikipedia.org/wiki/Integrated_circuit) [microcontroller](https://en.wikipedia.org/wiki/Microcontroller) created by [Atmel](https://en.wikipedia.org/wiki/Atmel) in the [megaAVR](https://en.wikipedia.org/wiki/MegaAVR) family. The Atmel [8-bit](https://en.wikipedia.org/wiki/8-bit) [AVR](https://en.wikipedia.org/wiki/Atmel_AVR) [RISC](https://en.wikipedia.org/wiki/Reduced_instruction_set_computing)-based microcontroller combines 32 kB [ISP](https://en.wikipedia.org/wiki/In-system_programming) [flash](https://en.wikipedia.org/wiki/Flash_memory) memory with read-while-write capabilities, 1 kB [EEPROM](https://en.wikipedia.org/wiki/EEPROM), 2 kB [SRAM](https://en.wikipedia.org/wiki/Static_random-access_memory), 23 general purpose I/O lines, 32 general purpose working [registers](https://en.wikipedia.org/wiki/Processor_register), three flexible timer/[counters](https://en.wikipedia.org/wiki/Counter_(digital)) with compare modes, internal and external [interrupts](https://en.wikipedia.org/wiki/Interrupt), serial programmable [USART](https://en.wikipedia.org/wiki/USART), a byte-oriented 2-wire serial interface, [SPI](https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus) serial port, 6-channel 10-bit [A/D converter](https://en.wikipedia.org/wiki/A/D_converter) (8-channels in [TQFP](https://en.wikipedia.org/wiki/Quad_Flat_Package) and [QFN](https://en.wikipedia.org/wiki/Quad_Flat_No-leads_package)/[MLF](https://en.wikipedia.org/wiki/Quad-flat_no-leads_package#Variants) packages), programmable [watchdog timer](https://en.wikipedia.org/wiki/Watchdog_timer) with internal [oscillator](https://en.wikipedia.org/wiki/Electronic_oscillator), and five software selectable power saving modes. The device operates between 1.8-5.5 volts. The device achieves throughput approaching 1 [MIPS](https://en.wikipedia.org/wiki/Million_instructions_per_second#Million_instructions_per_second) per MHz.

**Programming**

The Arduino/Genuino Uno can be programmed with the [Arduino Software](https://www.arduino.cc/en/Main/Software) (IDE). Select "Arduino/Genuino Uno from the Tools > Board menu (according to the microcontroller on your board). The ATmega328 on the Arduino/Genuino Uno comes preprogrammed with a [bootloader](https://www.arduino.cc/en/Hacking/Bootloader?from=Tutorial.Bootloader) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol .

**Power**

The Arduino/Genuino Uno board can be powered via the USB connection or with an external power supply. The power source is selected automatically.External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and Vin pin headers of the POWER connector.The board can operate on an external supply from 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

**Power Pins:**

* **Vin :** The input voltage to the Arduino/Genuino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
* **5V :** This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
* **3.3V :** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
* **GND :** Ground pins.
* **IOREF :** This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.

**Memory**

The ATmega328 has 32 KB (with 0.5 KB occupied by the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM .

**Input and Output:**

.Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode()](https://www.arduino.cc/en/Reference/PinMode), [digitalWrite()](https://www.arduino.cc/en/Reference/DigitalWrite), and [digitalRead()](https://www.arduino.cc/en/Reference/DigitalRead) functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller.

**Special Functions of Pins:**

* **Serial:** 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
* **External Interrupts:** 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
* **PWM:** 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.
* **SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
* **LED:** 13. There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
* **TWI:** A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. There are a couple of other pins on the board.

* **AREF**: Reference voltage for the analog inputs. Used with analogReference().
* **Reset:** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

**Communication**

Arduino/Genuino Uno has a number of facilities for communicating with a computer, another Arduino/Genuino board, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, [on Windows, a .inf file is required](https://www.arduino.cc/en/Guide/Windows#toc4). The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](https://www.arduino.cc/en/Reference/SoftwareSerial) allows serial communication on any of the Uno's digital pins.

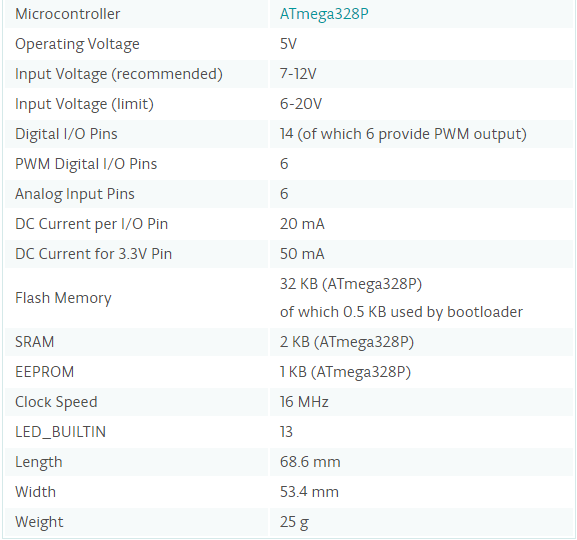
The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino Software (IDE) includes a Wire library to simplify use of the I2C bus; see the [documentation](https://www.arduino.cc/en/Reference/Wire) for details. For SPI communication, use the [SPI library](https://www.arduino.cc/en/Reference/SPI).

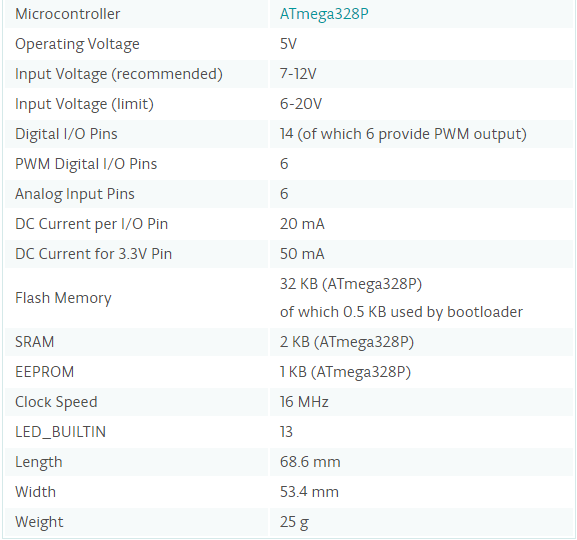
**Automatic (Software) Reset**

Rather than requiring a physical press of the reset button before an upload, the Arduino/Genuino Uno board is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino Software (IDE) uses this capability to allow you to upload code by simply pressing the upload button in the interface toolbar. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

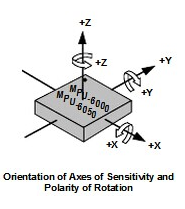
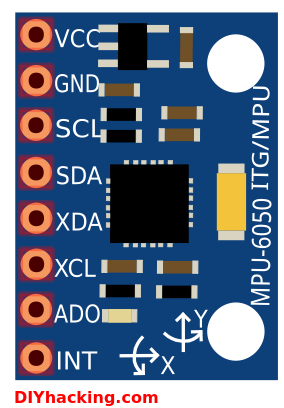
The Uno board contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line.





**5.2. MPU6050:**

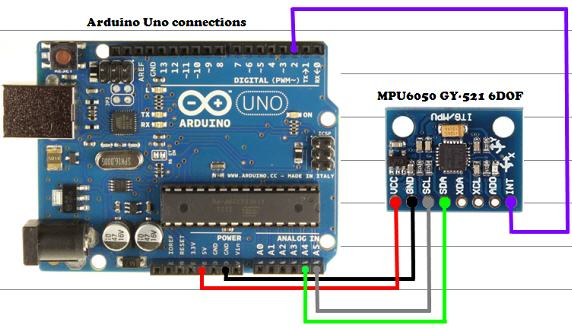
The InvenSense MPU-6050 sensor contains a MEMS accelerometer and a MEMS gyro in a single chip. It is very accurate, as it contains 16-bits analog to digital conversion hardware for each channel. Therefor it captures the x, y, and z channel at the same time. The sensor uses the I2C-bus to interface with the Arduino. The MPU-6050 is not expensive, especially given the fact that it combines both an accelerometer and a gyro. Reading the raw values for the accelerometer and gyro is easy. The sleep mode has to be disabled, and then the registers for the accelerometer and gyro can be read. But the sensor also contains a 1024 byte FIFO buffer. The sensor values can be programmed to be placed in the FIFO buffer. And the buffer can be read by the Arduino. The FIFO buffer is used together with the interrupt signal. If the MPU-6050 places data in the FIFO buffer, it signals the Arduino with the interrupt signal so the Arduino knows that there is data in the FIFO buffer waiting to be read. A little more complicated is the ability to control a second I2C-device. The MPU-6050 always acts as a slave to the Arduino with the SDA and SCL pins connected to the I2C-bus. But beside the normal I2C-bus, it has it's own I2C controller to be a master on a second (sub)-I2C-bus. It uses the pins AUX\_DA and AUX\_CL for that second (sub)-I2C-bus.  
It can control, for example, a magnetometer. The values of the magnetometer can be passed on to the Arduino.



**HOOKING THE MPU6050 TO THE ARDUINO:**

The connections are as follows:

|  |  |
| --- | --- |
| **MPU6050** | **ARDUINO UNO** |
| Vcc | 3.3V |
| GND | GND |
| SDA | A4 |
| SCL | A5 |
| INT | D2 |
| AD0 | GND |

****

**5.3. SIM900A:**

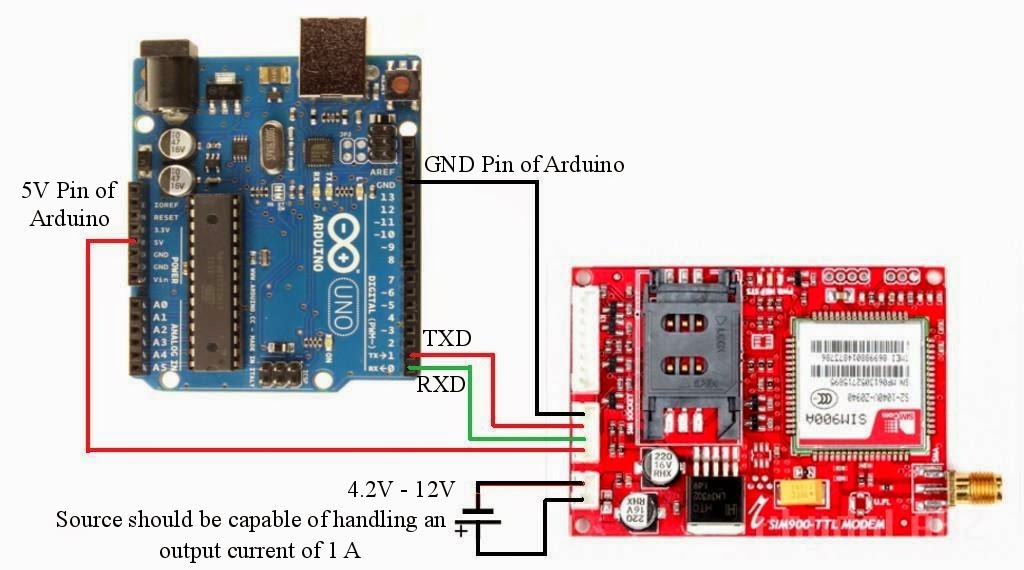
The SIM900A is a complete Dual-band GSM/GPRS solution in a SMT module which can be embedded in the customer applications. Featuring an industry-standard interface, the SIM900A delivers GSM/GPRS 900/1800MHz performance for voice, SMS, Data, and Fax in a small form factor and with low power consumption. With a tiny configuration of 24mmx24mmx3mm, SIM900A can fit in almost all the space requirements in user applications, especially for slim and compact demand of design. The features of SIM900A are listed below.

* Dual-Band **900/ 1800 MHz**
* GPRS multi-slot class 10/8GPRS mobile station class B
* Compliant to GSM phase 2/2+Class 4 (2 W @850/ 900 MHz)
* **Class 1** (1 W @ 1800/1900MHz)
* Control via **AT commands** (GSM 07.07 ,07.05 and SIMCOM enhanced AT Commands)
* Low power consumption: **1.5mA(sleep mode)'**
* Operation temperature:**-40°C to +85 °C**
* **Status indicator(D5)**:It will flashes continuously whenever the call arrives otherwise it is left ON.
* **Network LED(D6)**:This led will blink every second which indicates that the GSM module is not connected to the mobile network. Once the connection is established successfully, the LED will blink continuously every 3 seconds.

**HOOKING THE SIM900A TO THE ARDUINO:**

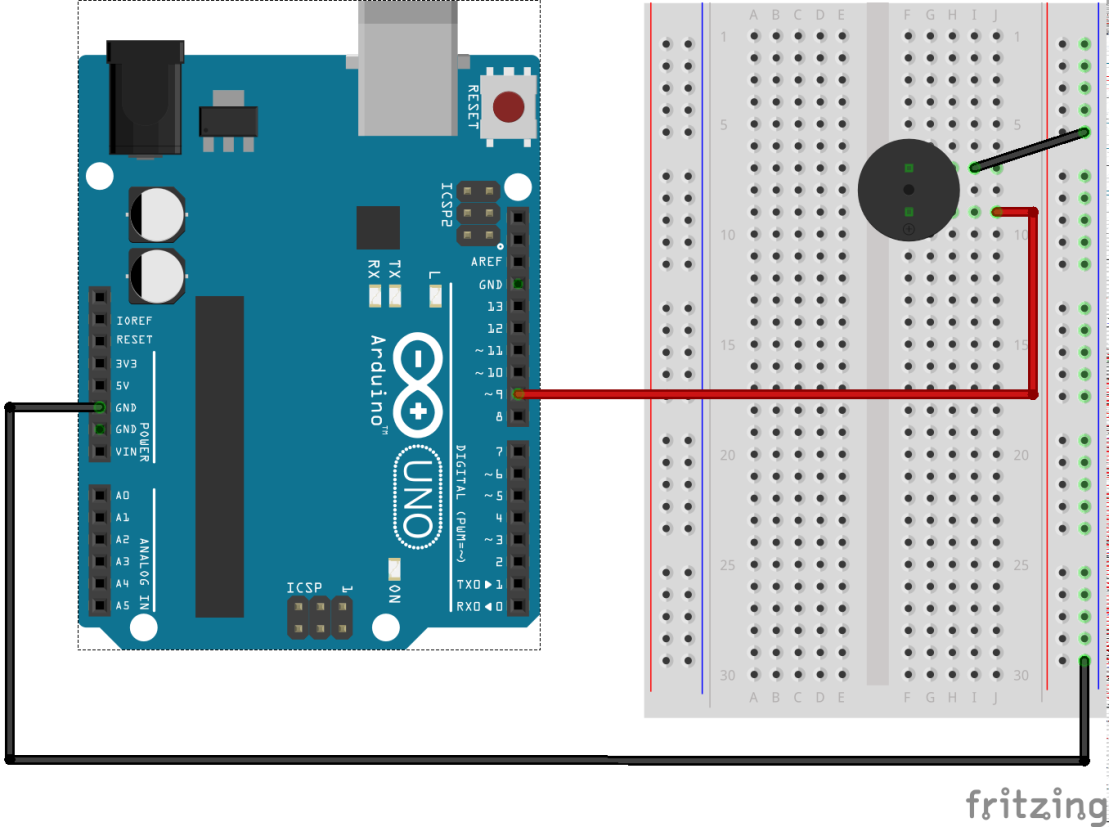
The SIM900A module has 6pins in which two pins for Vcc and Gnd and the rest are 3VR&3VT(3volt Rx & Tx) and 5VR,5VT(5volt Rx & Tx) and the connections are made as follows:

|  |  |
| --- | --- |
| **SIM900A** | **ARDUINO UNO** |
| Vcc | 5V |
| GND | GND |
| 5VR | Pin 9 |
| 5VT | Pin 10 |



**5.4. BUZZER:**

A 5V buzzer has been connected to the output pin of the Arduino to indicate fall or jerk.



**CHAPTER 6**

**SOFTWARE REQUIREMENTS**

**6.1. Arduino IDE:**

A program for Arduino may be written in any [programming language](https://en.wikipedia.org/wiki/Programming_language) for a compiler that produces binary machine code for the target processor. Atmel provides a development environment for their microcontrollers, AVR Studio and the newer Atmel Studio. The Arduino project provides the Arduino [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE), which is a [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) application written in the programming language [Java](https://en.wikipedia.org/wiki/Java_(programming_language)). It originated from the IDE for the languages [Processing](https://en.wikipedia.org/wiki/Processing_(programming_language)) and [Wiring](https://en.wikipedia.org/wiki/Wiring_(development_platform)). It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, [brace matching](https://en.wikipedia.org/wiki/Brace_matching), and [syntax highlighting](https://en.wikipedia.org/wiki/Syntax_highlighting), and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus.

A program written with the IDE for Arduino is called a sketch. Sketches are saved on the development computer as text files with the file extension .ino. Arduino Software (IDE) pre-1.0 saved sketches with the extension .pde.

The Arduino IDE supports the languages [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B) using special rules of code structuring. The Arduino IDE supplies a [software library](https://en.wikipedia.org/wiki/Software_library) from the [Wiring](https://en.wikipedia.org/wiki/Wiring_(development_platform)) project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable [cyclic executive](https://en.wikipedia.org/wiki/Cyclic_executive) program with the [GNU toolchain](https://en.wikipedia.org/wiki/GNU_toolchain), also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

A minimal Arduino C/C++ sketch, as seen by the Arduino IDE programmer, consist of only two functions.

* setup: This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch.
* loop: After setup has been called, function loop is executed repeatedly in the main program. It controls the board until the board is powered off or is reset.

Most Arduino boards contain a [light-emitting diode](https://en.wikipedia.org/wiki/Light-emitting_diode) (LED) and a load resistor connected between pin 13 and ground, which is a convenient feature for many tests and program functions.

**6.2. Android Studio:**

Android Studio is the official [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) for the [Android](https://en.wikipedia.org/wiki/Android_(operating_system)) platform.

The following features are provided in the current stable version:

* [Gradle](https://en.wikipedia.org/wiki/Gradle)-based build support
* Android-specific [refactoring](https://en.wikipedia.org/wiki/Code_refactoring) and quick fixes
* [Lint](https://en.wikipedia.org/wiki/Lint_(software)) tools to catch performance, usability, version compatibility and other problems
* [ProGuard](https://en.wikipedia.org/wiki/ProGuard_(software)) integration and app-signing capabilities
* Template-based wizards to create common Android designs and components
* A rich [layout editor](https://en.wikipedia.org/wiki/Graphical_user_interface_builder) that allows users to drag-and-drop UI components, option to [preview layouts](https://en.wikipedia.org/wiki/WYSIWYG) on multiple screen configurations
* Support for building [Android Wear](https://en.wikipedia.org/wiki/Android_Wear) apps
* Built-in support for Google Cloud Platform, enabling integration with Google Cloud Messaging and App Engine.
* An Android Virtual Device that is used to run and debug app

**6.3. IP Webcam:**

IP Webcam turns your phone into a network camera with multiple viewing options. Optional Ivideon cloud broadcasting is supported for instant global access.  
Two-way audio supported in tinyCam Monitor on another android device.  
Use IP Webcam with third-party MJPG software, including video surveillance software, security monitors and most audio players. Features include

* Video upload to Dropbox, SFTP, FTP and Email using Filoader plugin
* Several web renderers to choose from: Flash, Javascript or built-in
* Video recording in WebM, MOV, MKV or MPEG4 (on Android 4.1+)
* Audio streaming in wav, opus and AAC (AAC requires Android 4.1+)
* Motion detection with sound trigger, Tasker integration.
* Date, time and battery level video overlay.
* Sensor data acquisition with online web graphing.
* Videochat support (video stream only for Windows and Linux via an universal MJPEG video streaming driver)
* Cloud push notifications on motion and sound, cloud recording for motion-triggered records, online video broadcasting powered by Ivideon.
* Cloud push notifications on motion and sound, cloud recording for motion-triggered records, online video broadcasting powered by Ivideon.

**6.4. Ivideon Application:**

Ivideon is one of the best app for video surveillance, remote video monitoring and video recording for security cameras, DVRs and NVRs.   
It will allow you to keep tabs on things while you’re away with event notifications and recordings stored either locally or in the Ivideon cloud.  
Ivideon easily supports both single and multiple locations and tracks what matters to you no matter where you are, keeping you updated on important events in real time and storing those events in the cloud for easy access.  
  
Ivideon is scalable, simple and reliable. From a webcam to monitor a home to an IP-camera network serving an international business, Ivideon can handle it.  
  
Every day, more than 2,000,000 individuals and businesses use Ivideon to:

* access high-quality video online (with sound) via any standard connected device including Android
* record and store video in the cloud with guaranteed reliability and data security
* receive automatic push- and e-mail notifications of suspicious movements or sounds
* quickly search and view recorded surveillance video
* access quality video over any bandwidth down to 3G
* easily expand to any number of cameras
* share camera access with PTZ, Archive, and Live options
* transfer and store data securely
* embed surveillance feeds on websites or blogs
* share camera links over social networks
* search video events linked to POS, Access Control and Home Automation systems

Ivideon is safe. Ivideon is simple to install and maintain. And Ivideon is accessible from anywhere. But with all the functionality of more expensive and complex solutions.

**CHAPTER 7**

**WORKING OF FALL DETECTOR**

**Step 1:**

The device including Gyroscope, Accelerometer and GSM module is supposed to be mounted to the chair.

**Step 2:**

The accelerometer and gyroscope combined, function to detect the fall made by the person to be taken care of, which sends the data to the Arduino board.

**Step 3:**

The Arduino board senses the fall, which has been coded using constraints, resulting which it sends signal to buzzer. Hence, the buzzer starts buzzing.

**Step 4:**

Simultaneously, while the buzzer is fizzling, AT commands are sent to GSM module which includes two functionalities.

**Step 5:**

First, the GSM module is set to Text Mode. Using command, a Short Message Service is sent to the registered mobile number. Following which the call is made to the same number.

**Step 6:**

The caretaker gets notified by SMS. He/She must use the “Fall Detector Application”.

**Step 7:**

The caretaker must register on the application as shown on the opening Interface.

**Step 8:**

After completing registration, he/she must next login to access other functionalities.

**Step 9:**

The caretaker could Live Stream the situation and later send message to the neighbors if not accessible to help.

**CHAPTER 8**

**FALL DETECTOR ANDROID APPLICATION**

**Registration Interface:**

The registration page in the Fall Detector application allows user to register into the Fall Detector application. It fetches basic details of users such as

* Name
* Email id
* Phone number
* Care takers contact number
* Age
* Username
* Password

The above details are stored in the Userdetails database under users table.

**Login Interface:**

The login page in the Fall Detector application allows users to login into the application. The user credentials are validated and then the user is allowed to enter into the application.

**Update Profile Interface:**

The update profile interface allows user to update the details of care takers. The phone numbers of the care takers are fetched and stored in the database.

**Stream Video Interface:**

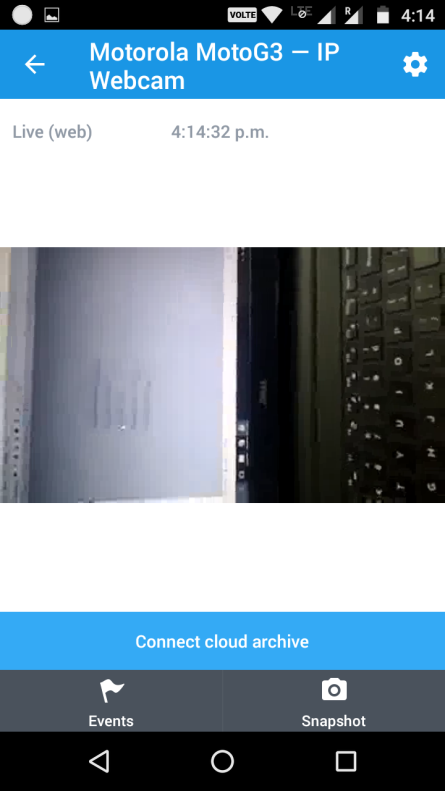
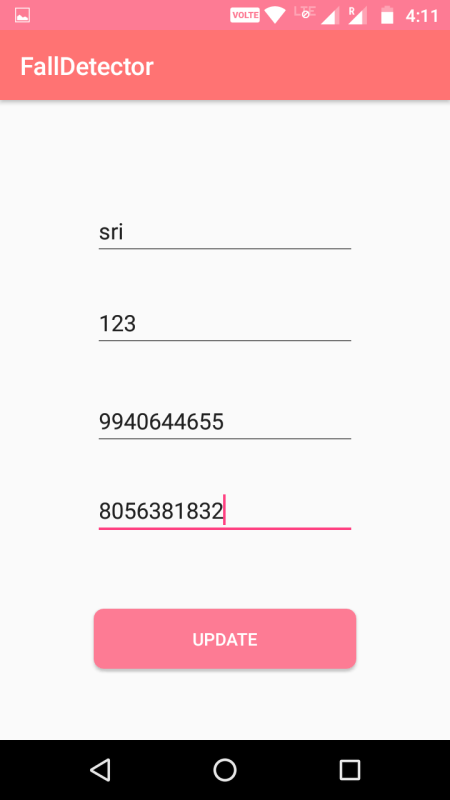
This interface has been utilized to live stream the fall that has occurred in the Fall Detector application.

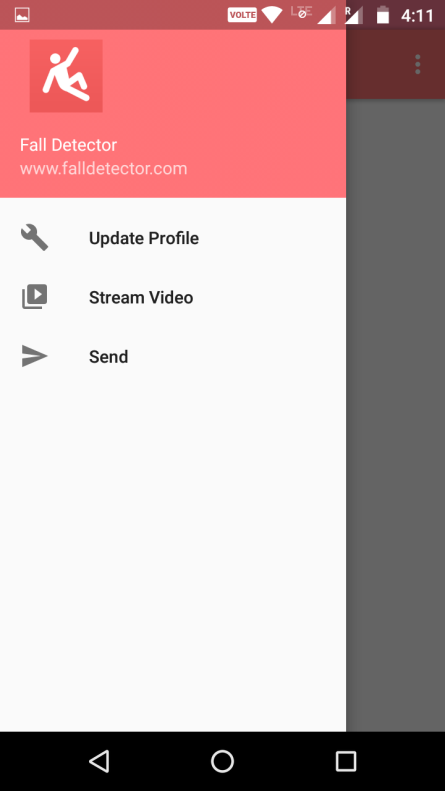
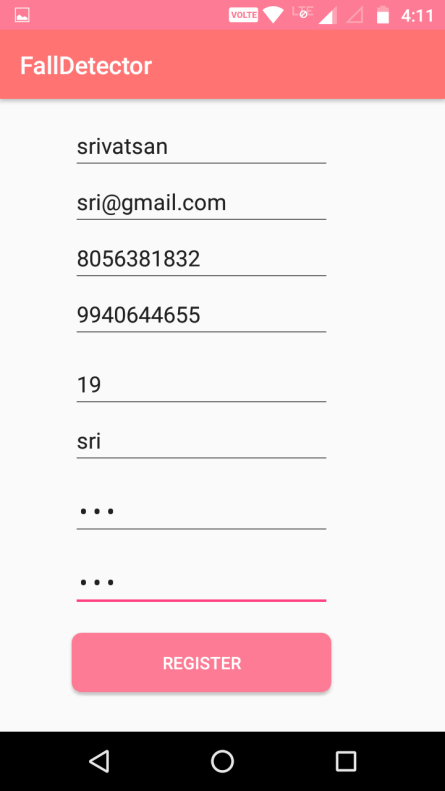
**SMS Interface:**

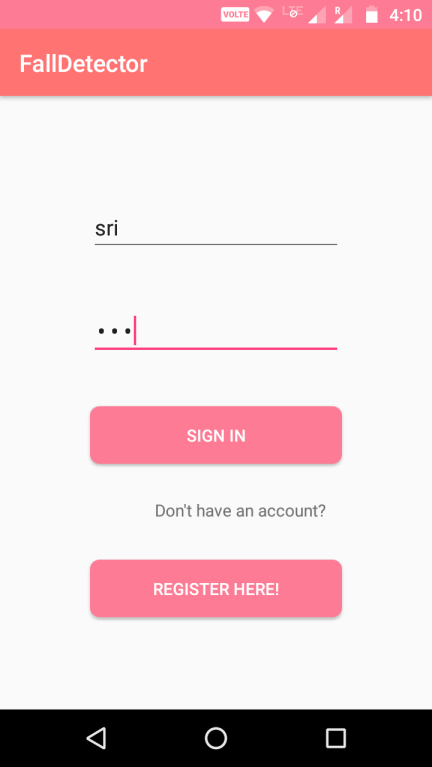
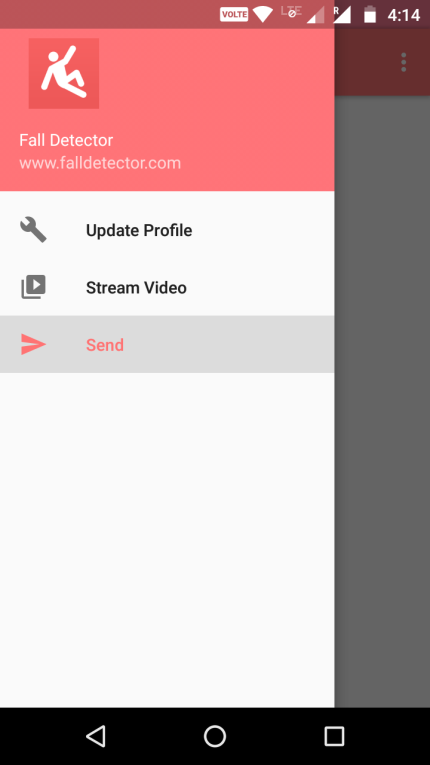
On clicking the Send option in navigation bar an SMS is automatically sent to the neighbor.

**CHAPTER 9**

**SCREENSHOTS**

** **

** **

** **

**CHAPTER 10**

**CONCLUSION**

Thus, we have now found a system that could help an individual (caretaker) know that the person has fallen down. We have fulfilled all the requirements in the system, like, firstly, the fall could be detected and secondly, the fall has been notified to the caretaker. Also, this system has provided a facility for the caretaker to respond to the fall, like, livestreaming video and if he/she is not nearby, could send a notification to the persons whom the caretaker thinks would be available nearby.

**CHAPTER 11**

**INFERENCE**

Although, we have accomplished every requirement in this project, we have had a small bug that was not tackled with successfully. The project promised for a switch to be attached along with the device which could notify in case of false alarms, but this task has not been accomplished.

**CHAPTER 12**

**REFERENCES**

https://www.arduino.cc/en/main/arduinoBoardUno

http://playground.arduino.cc/Main/MPU-6050

https://www.instructables.com/id/Interfacing-SIM900A-GSM-Modem-with-Arduino/?ALLSTEPS

https://en.wikipedia.org/wiki/Arduino

https://en.wikipedia.org/wiki/Android\_Studio

https://play.google.com/store/apps/details?id=com.pas.webcam&hl=en

https://play.google.com/store/apps/details?id=com.ivideon.client&hl=en

http://www.learnerswings.com/2014/05/how-to-connect-sim900-ttl-gsmgprs-modem.html

[www.42bots.com/tutorials/arduino-uno-and-the-invensense-mpu-6050-6dof-imu/](http://www.42bots.com/tutorials/arduino-uno-and-the-invensense-mpu-6050-6dof-imu/)

[www.geeetech.com](http://www.geeetech.com/)

[www.playground.arduino.cc/Main/MPU-6050](http://www.playground.arduino.cc/Main/MPU-6050)

[www.en.wikipedia.org/wiki/I%C2%B2C](http://www.en.wikipedia.org/wiki/I%C2%B2C)

[www.tutorialspoint.com/android/](http://www.tutorialspoint.com/android/)

www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Register-Map1.pdf

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | | |

<https://www.tutorialspoint.com/android/android_sqlite_database.htm>

<http://www.androidbegin.com/tutorial/android-video-streaming-videoview-tutorial/>

<https://www.tutorialspoint.com/android/android_sending_sms.htm>

<http://stackoverflow.com/questions/3872063/launch-an-application-from-another-application-on-android>

[https://www.ivideon.com/my/service/logi](https://www.ivideon.com/my/service/login)n