System for Monitoring People with Disabilities in the Event of an Accident using Mobile Terminals

Alexandra Fanca[1], Monica Cujerean[2], Adela Puscasiu[3], Dan-Ioan Gota[4], Honoriu Valean[5]
Department of Automation[1, 3, 4, 5]
Technical University of Cluj-Napoca, Cluj-Napoca, Romania[1, 2, 3, 4, 5]

Abstract—**Being in the speed century, people around the world are busy scheduling every day and therefore, it is impossible to spend enough time with the elderly and people with disabilities or people who have a chronic illness. These persons need a lot more attention and care because they cannot cope with the daily activities like a healthy person would. Daily monitoring and assistance of elderly or disabled people is a very important task, both in the current activity, or, especially when emergencies can occur. Fortunately, we can easily use today's technologies, which are constantly developing to be able to monitor them remotely. This paper tries to find a solution to reduce cases of fatality due to accidents, by using advanced technologies of today, e.g. smartphones, fast communications. The use of these technologies can provide permanent monitoring of the elderly and persons with disabilities, without bounding their mobility and without affecting their quality of life. In this way, if emergency situations arise for the elderly or people with disabilities or chronic diseases, measures can be taken as soon as possible. The development of a mobile application capable to monitor the occurrence of accidents for the above-mentioned persons is obviously a help granted to the doctors involved in ensuring their health. Thus, the main objective of the application is to detect the accidental falls of the persons in the shortest possible time. Another objective is to provide an application that runs in the background of the mobile operating system, using as little as possible the power supply.**

Keywords—Smartphones; built-in smartphone sensors; monitoring people; android applications; accident detecting system

I. Introduction

One of the main problems of the community is those related to people's health. Over time, in finding ways to solve people's medical problems, both the academic community and research laboratories of large companies have been involved. Falls are considered major health hazards for both the elderly and the ill people, especially with chronic and neurodegenerative diseases [1].

People with disabilities or chronic illnesses must carry out their daily activities independently and that is a major challenge for them [2]. Almost half of these cases are not under medical supervision, and in the event of an accident followed by sudden falls or significant injuries, no one will be announced to provide medical care. In this situation, the injured person can be found in severe illness or even deceased.

According to the information provided by the National Statistical Institute, the number of deaths in Romania has remained constant over the last 20 years, ranging from 240,000 to 260,000 deaths annually. The average death rate recorded by Romania is above the mortality rate in the European Union (EU), respectively 13 deaths per thousand inhabitants, compared to the EU average of 10 deaths per thousand inhabitants [3]. This number would drop considerably if intelligent means of patient monitoring were used so that healthcare could intervene at critical times. Thus, in addition, 1 in 3 people could be saved on time. Accidents caused by slips and falls are ranked second place after road accidents and can lead to trauma and even death [4]. Chances of a person to be saved are increasing if assistance is received as soon as the accident has happened [5].

Smartphone has become a good tool used in mobile applications for the medical and safety fields, due to the increasing performance of the sensors and their ability to transmit and collect a large amount of data.

Following the introduction, the rest of the paper is organized as follows: Section II presents the existing main patients monitoring systems. In Section III the proposed system is developed and in Section IV some tests and results are presented. Section V focuses on some discussions, conclusions and future improvements.

II. State of the Art

Over the years the problem of monitoring and detecting human injuries/accidents has been intensively studied. In the literature, several systems based on different techniques for accident detection are reported.

Depending on the techniques used, we can find systems for monitoring people and/or detect falls that use acoustic sensors [6-9], video camera and image processing [10-13] or even sensors attached to the human body [14-16].

In [17] the authors proposed a system that is composed of a wearable accelerometer sensor, magnetic door sensor, Arduino Uno and a camera. For communication, a GSM module and a Wi-Fi module are used. This method is quite complex and also expensive and involves the use of a large number of components. It may not be reliable over time due to hardware degradation. In terms of usability, monitoring can be provided only indoor, where the camera and the magnetic sensor may have a use (there are doors). So, users can't use this technology if they go outside for a walk or shop.

Another fall detection system, proposed in [18], is based on image processing. The system proposed in this paper consists of a computer and a video camera, which can be installed inside a building, on the ceiling or the walls. The system can monitor a room for 24 hours without human intervention. It is based on artificial vision algorithms that monitor the presence of people in a room and detect if a person has fallen. When a fall is detected, an alarm message is sent to the attending physician along with a photograph. If the person recovers, another message is sent.

All these systems require the use of additional components compared to the systems that use sensors built into smartphones. Also, some systems are restricted only to be used inside the buildings.

III. Methods and Material

The purpose of this work is to implement a mobile application in order to remotely monitor people with disabilities or chronic diseases, using built-in sensors of smartphones. This application can be used by anyone with a phone with an Android operating system.

Once the application is installed on the phone, the user will be able to create an account by entering the personal data: first name, second name, email, password, personal phone number and two other telephone numbers as the emergency contact persons. Once registered, the user can log in with his credentials (email and password). After logging in, monitoring can begin at the user's request. Monitoring consists of constantly checking the coordinates of the accelerometer to detect a possible accident. The fall will be reported and filtered through the Linux machine's server, which sends an emergency message to the contacts instantly.

All user accident information will be displayed in the graphs. These graphs are implemented in Grafana and the access is granted only to the administrator. The displayed information will target the rate of accidents among the elderly, persons with disabilities or serious chronic diseases.

Another important objective is to use the phone's battery consumption as optimally as possible. For this purpose, a single sensor was chosen to be inspected by the application. This sensor, the accelerometer, is present in both older and state of the art phones.

The server part of the application is implemented as several services and allows performing complex operations.

A. Structural Specifications of the System

The system architecture is based on several independent services that communicate with each other. Thus, each service can be managed independently, without affecting the operation of other services.

This application is designed based on three-tier architecture. The data layer, the business logic layer, and the presentation layer are running on different servers so that it can be independent and can be accessed separately.

The system is composed of the following services:

- The service for users, representing the mobile application with a friendly interface, accessible to people of all ages. The available actions are: register, log in, edit profile, start the application or stop the application;
- The service managed by the admin that starts the Linux virtual machine and controls the filtering of the alerts received from the phone and also manages the graphs that monitor the accident rate on each user.

The connection between the user service and the service managed by the admin is made through a virtual machine API to which HTTP requests are sent.

The architecture of the system is presented in Fig. 1.

B. Functional Specifications of the System

The mobile application has several benefits that make it more flexible, qualitative, with a friendly interface. Regarding the functional specifications, a few examples can be listed:

- Reliability: the application architecture allows long-term operation, which can run on the phone's background for an indefinite period of time, without sending erroneous information to the server or without detecting an unfortunate accident;
- Accessibility: the user-friendly interface of the Android application can be used by anyone, regardless of age or technical knowledge. The users only need to start the application and have the phone permanently with them;
- Security: This targets the Android application as well as the admin's graphical interface (Grafana graphics). When creates an account, the user must fill in all the required fields, otherwise, registration cannot be done. When logs in, the credentials are checked in the database (email and password) and the user can only access the application if he/she enters the data correctly. The admin, in order to be able to access the statistics regarding the number of accidents produced in a period of time, must authenticate with the admin credentials (user and password);
- Database: There are two databases. The first is stored in Cloud Firebase and holds user information. The second database is a local database stored on Linux virtual machine that holds the information displayed by Grafana: user name, two indexes for the occurrence of the accident and the time at which the accident occurred;
- API usage: make possible the connection between the Linux machine and Android. The application sends HTTP requests to the Linux machine, which further filters the data about the user who had the accident and sends messages to the predefined contacts;
- Decomposition in services: the application is composed of several services, these being interconnected but also working independently. Each service is maintained on a different development platform;
- Limitations: The app can work on any phone that has Android operating system independent of the version. The used sensors, accelerometer and GPS are present on most smartphones.

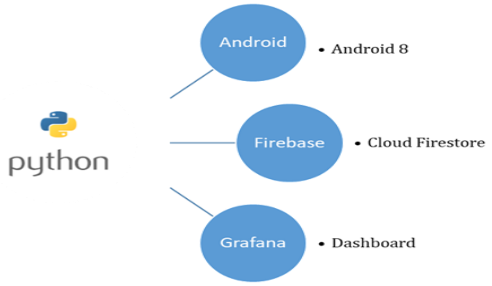


Fig. 1. The Architectural Presentation.

C. The Architecture of the Proposed System

The application architecture consists of several services that work independently. This facilitates the possibility to change the functionality of a service without degrading the rest of the services.

Qualitative requirements, such as functionality, reliability, accessibility, efficiency, security, independence from other platforms, require the design of a software system with a well-defined architecture. This is the reason for designing the service-based architecture. This architecture is relatively complex and is shown in Fig. 2.

As can be seen from the previous figure, the system consists of four important parts:

- Android application.
- Linux server.
- Database.
- Grafana.

The four components of the system are presented below.

1) Android Application: The smartphone allows the use of as many built-in sensors to perform different tasks or to monitor user's actions. For Android, for example, Google provides access to a set of libraries that allow developers to know when the user is in the car or riding a bicycle when entering or leaving a developer-defined perimeter. The proposed application uses the accelerometer to detect any possible impact, fall or other unusual situations that may occur. The Android application is divided into three major components at the logical level: authentication/registration, event monitoring and alerting.

2) Linux server: The role of this machine is to process and filter the information sent by the android application in order to send messages to the contacts in the event of an accident or strange behavior of the phone. It is installed on an AWS (Amazon Web Services) virtual machine under server mode.

The connection is made using an SSH (Secure Shell) key. The Admin connects using the Putty platform. He can set up a user name and password with which to connect in the future. The main libraries installed for optimum operation are Flask, Firebase-connector, Grafana, and MySQL.

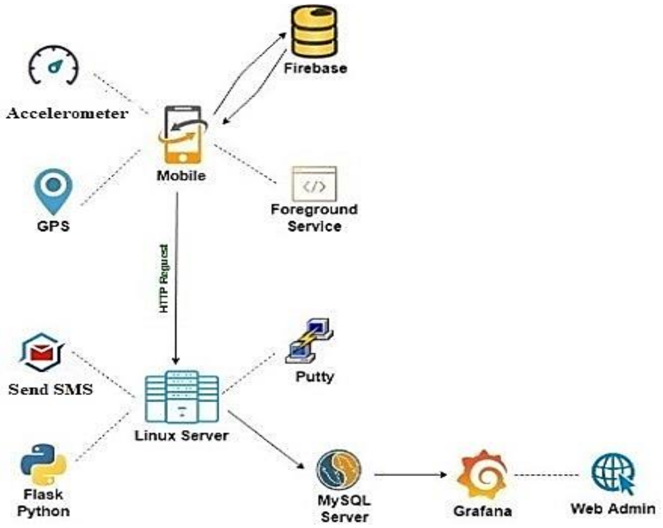


Fig. 2. The Architecture of the Proposed System.

3) Database: Regarding the data level, we can talk about two databases present:

- Firestore Cloud - dedicated database for mobile applications in general, but which can be used for other platforms, for example, node.js, Linux servers, and go java. In this context, the non-relational database is made up of a collection called users, which includes a set of documents, each having several fields, which the user must fill in through the registration process.
- Local Linux machine database - is used for the Grafana Dashboard, because the graphics cannot extract the required data directly from Firebase. Therefore, the database contains a table consisting of username, alert status (1 or 0, depending on the case) and the current date.

4) Grafana: Grafana Dashboard is a graphical interface installed on the Linux machine that can be accessed using the same API as the server. It can only be accessed by admin because authentication is required when accessing it. This includes the accident rate for each individual user, in the form of a graph, which is updated once a minute. Statistics can be maintained from the last 15 minutes to the last year of application usage by a user.

D. Use Case Diagram

One of the essential aspects of modeling a system is the presentation of its dynamic behavior. It is important to highlight the functionalities of the system, both at the client part and at the server part. Depending on its dynamic nature, the diagram highlights certain internal or external factors to make the interaction possible.

The use-case diagram is presented in Fig. 3 and includes the two possible actors of the system: the user of the mobile application and the admin of the server.

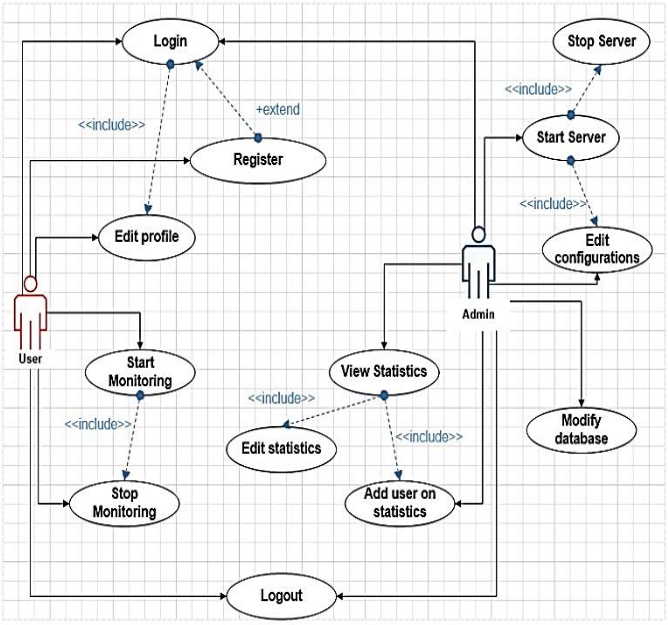


Fig. 3. Use Case Diagram.

As can be seen in the figure above, the roles of the actors are well defined.

Possible actions of the user:

- register to the application.
- log in to the application.
- start monitoring.
- stops monitoring.
- edit his profile with personal data.

The Admin has full rights over all services such as:

- start/stop the server.
- log in to the Grafana interface.
- edit the graphics, modify the logic behind them.
- add, delete users through the Grafana interface.
- add / delete graphs.
- configure the server; access the Firestore Cloud database.

E. Sequential Diagram

The sequence diagram is used to chronologically describe the events that take place within a system. The sequence diagram of the proposed system is shown below (Fig. 4).

As can be seen from the sequential diagram of the system, the sequence of actions is coherent and the steps in which the actions could be performed are finite and well defined.

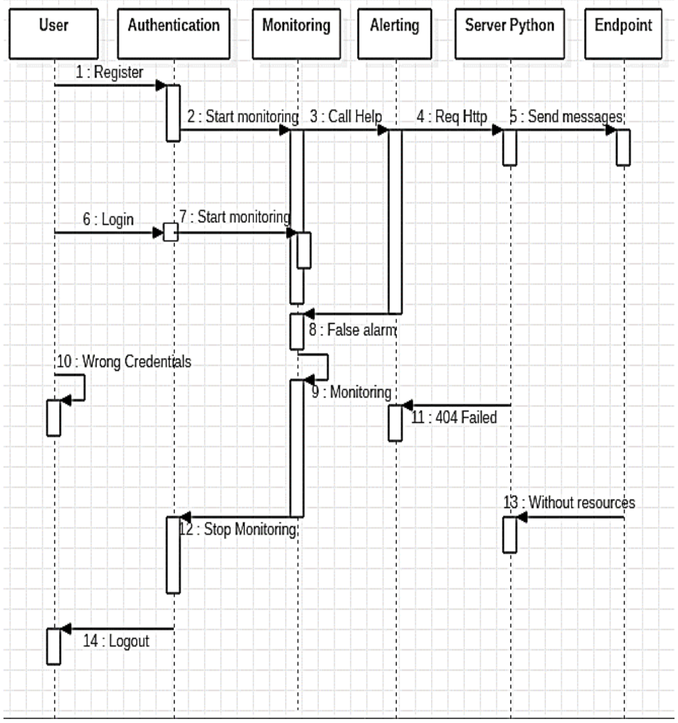


Fig. 4. Sequential Diagram.

In the first half of the diagram, the application implements all the actions that can take place without difficulties: registration, start monitoring, triggering the alert, sending request Http and sending messages. The chain of actions unfolds very quickly, there are no delays.

In the second half of the diagram, there are obstacles that may occur during the actions: logging in with incorrect credentials, false alarm, stopping the monitoring by the user, 404 could not make the call to the server or the server ran out of resources.

F. Implementation of the Android Application

The Android application is divided into three main components at the logical level: authentication/registration, event monitoring and alerting of events that have happened. Below are details on their implementation.

1) Authentication and registration: The login case is presented to the user after installing the application or after it has logged out. From the main activity, the authenticateUser() method is called when starting the application. This method receives the email stored in SharedPreferences as a parameter. The answer is delivered asynchronously via the LoginResponseListener interface back to MainActivity. If the answer is a successful one, the main screen of the application is presented. If the answer is negative, MainActivity presents LoginActivity for authentication. This flow is performed using the startActivityForResult() method, where it is verified that the authentication has been successfully performed. Below the class diagram for authentication is presented (Fig. 5).

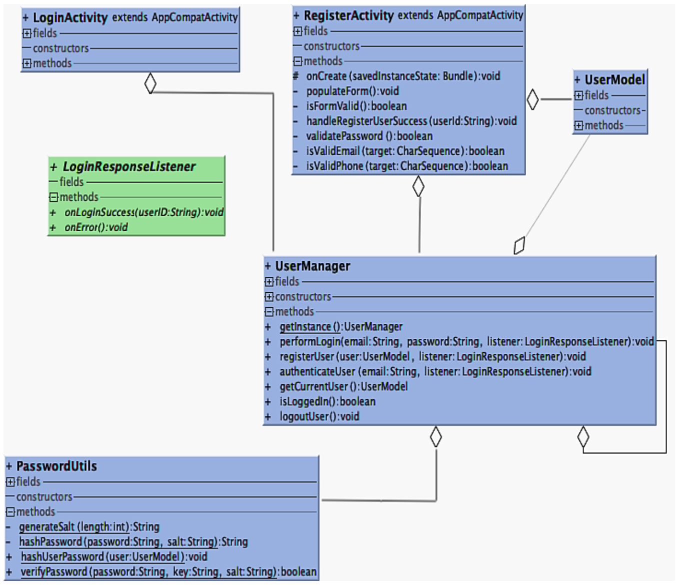

+ LoginActivity extends AppCompatActivity
fields
constructors
methods
+ RegisterActivity extends AppCompatActivity
fields
constructors
methods
# onCreate (savedInstanceState: Bundle):void
- populateForm():void
- isFormValid():boolean
- handleRegisterUserSuccess (userId:String):void
- validatePassword ():boolean
- isValidEmail (target: CharSequence):boolean
- isValidPhone (target: CharSequence):boolean
+ UserModel
fields
constructors
methods
+ LoginResponseListener
fields
methods
+ onLoginSuccess(userID:String):void
+ onError():void
+ UserManager
fields
constructors
methods
+ getInstance ():UserManager
+ performLogin(email:String, password:String, listener:LoginResponseListener):void
+ registerUser (user:UserModel, listener:LoginResponseListener):void
+ authenticateUser (email:String, listener:LoginResponseListener):void
+ getCurrentUser ():UserModel
+ isLoggedIn():boolean
+ logoutUser():void
+ PasswordUtils
fields
constructors
methods
- generateSalt (length:int):String
- hashPassword (password:String, salt:String):String
+ hashUserPassword (user:UserModel):void
+ verifyPassword (password:String, key:String, salt:String):boolean

Fig. 5. Class Diagram for Authentication.

The data entered in the Register Activity is verified and validated by the isFormValid () method as follows: verified it respects the structure and checked and that no field is left blank:

- email - check the address format with a Regex.
- password - it is checked to be no more than 7 characters and coincides with the password confirmation field.
- name - it is only checked to be completed.
- phone numbers - check with a Regex for the formats: 0040000000000 and +40000000000. Verifying phone numbers with 100% accuracy is almost impossible due to the large number of variations of international phone number formats. This application is developed for demonstration and yet only national numbers are valid.

2) Event monitoring: Event monitoring is the main objective of the application. This is achieved by continuous monitoring of the events sent by the phone's accelerometer sensor. The application starts a service that runs all the time in the background. Monitoring Service has access to the accelerometer and registers to receive data from it at a range defined by the SensorManager constant “SENSOR_DELAY_NORMAL”. At each event, the value of the acceleration is checked in m / s2 on the 3 axes. If the value exceeds the preset threshold by the constant “SENSOR_SENSITIVITY_THRESHOLD” with the value of 30, the user will be alerted by a notification, with an expiration time of 20 seconds. The value of 30 m/s2 was chosen following the studies present in the literature. The 20-second alert is accompanied by an alarm sound, and the user has the option to call for help or declare it a false alarm (for example, the phone was accidentally dropped from his hand). If the notification expires or the user confirms it, the phone will call a web service to which the current location and email of the user will be sent as parameters. The current location detection is done using the GPS of the phone in HIGH_ACCURACY mode and starts once the alarm has been activated. This results in lower energy consumption.

For obtaining data about the current acceleration of the device, the accelerometer of the phone has been used. The type of sensor “Sensor.TYPE_LINEAR_ACCELERATION” was chosen so that it returns a three-dimensional vector that indicates the value of the acceleration on the 3 axes in m/s2 and excludes gravitational acceleration. When the device is stationary, the returned values are close to 0.

The Android operating system offers multiple location detection strategies for devices that can be used, depending on the application. GPS_PROVIDER was used for the proposed system, as this method allows the location of the highest accuracy. Initially, we have tried to get the last known location of the operating system through “LocationManager. getLastKnownLocation”. If this cannot be achieved or is less than 1 minute, a new location is required by the “Location Manager.requestLocationUpdates” method.

3) Event alert: NetworkManager is a singleton that has only one method, callHelp (String latitude, String longitude, String userId). This method uses the OkHttp library to make an Http request to the alert server. This is called by MonitoringService.

The purpose of the Linux server is to filter alerts, user information and to then send messages based on information received from the Android application. The figure below shows how the server works (Fig. 6).

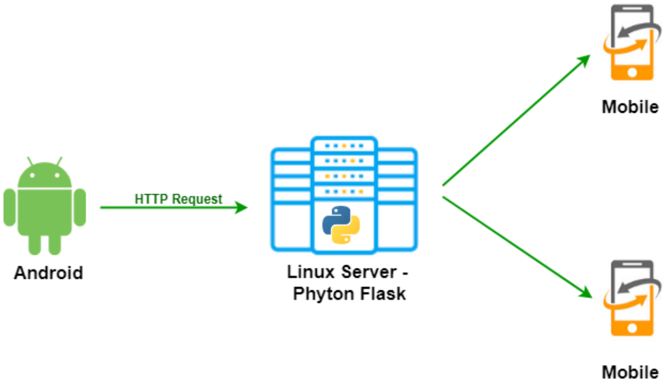


Fig. 6. How the Server Works.

Cento 6.9 was used to create the server (Linux virtual machine). This is an open-source operating system, used by a wide range of developers. Cento was installed on a virtual machine in AWS (Amazon Web Services) under server mode. This means that it has no graphical interface, all system configurations were done by accessing the server through SSH.

After the virtual instance was created in AWS, the connectivity part was set. The "brain" of the server application is implemented in the api.py file. The main function of the API is composed of several steps:

- It connects to Firebase (the online DB used by the Android application) and extracts all users with the related information, as those inserted into a dictionary variable. This is a feature of Python mode called Firebase;

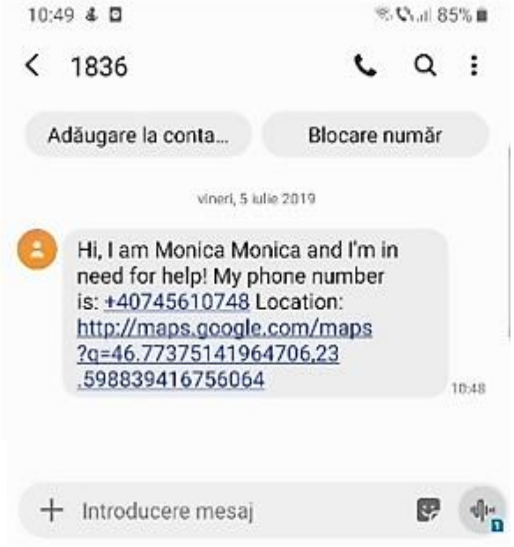
- Parse the dictionary created above and verify that the email address sent by the Android application to this API is valid and belongs to a Firebase user;

- If the above condition is valid, some variables such as First Name, First Name, Telephone Number, etc. are extracted from the entries in Firebase of the user concerned;

- The SMS sending function is executed. This feature is a query of another API provided by AWS through which we can send SMS. This option was chosen because it is a secure solution that does not involve high costs and is available 100% without downtime.

- The last step of the function is to enter information into the local MySQL database about the user who queried this API. The person's name, date and value "1" are added. This helps us create a graph and see how many times a person has used this API using Grafana.

The messages are sent on two different phone numbers of the contact persons and contain information about: the name and surname of the person in danger, his telephone number and the location where he can be found. The message format is shown below (Fig. 7).



10:49 85%

1836

vineri, 5 iulie 2019

Hi, I am Monica Monica and I'm in need for help! My phone number is: +40745610748 Location: http://maps.google.com/maps?q=46.77375141964706,23.598839416756064

10:48

Fig. 7. Format Alert Message.

IV. Results

After the implementation of the system, it was required to test and validate it. Several tests were performed, such as:

- testing the functionality of the Android application;

- server security testing;

- optimization of battery consumption;

- production of statistics using Grafana.

As for testing the functionality of the mobile application, several aspects have been studied and tested, such as the login part, the register, the sending of Http requests in the most correct way.

Login is done by filling in the email and password fields (Fig. 8(a)). Failure to complete them or adding incorrect credentials, which are not found in the database, will trigger the following error: "wrong credentials". If the credentials match the values stored in the database, the user can open the main page of the application. There, it will be able to start monitoring, stop monitoring, but it also has a menu with two buttons: "My profile" and "Logout".

For the register part, the presence of certain rules established in the analysis and implementation part of the system was considered (Fig. 8(b)). The rules established for creating a user account include:

- filling in all the fields;

- the email must have an appropriate structure: zzzyy@gmail.com;

- the telephone numbers must have the prefix of Romania and 10 digits;

- the password must be of maximum 7 characters;

- "password" and "confirm password" must match.

"My profile" is the button that allows the user to edit their profile. It can change: name, surname, email, personal phone number, and phone numbers of the contact persons and also can reset his password. The editing part of the profile has the same rules as the registration ones.

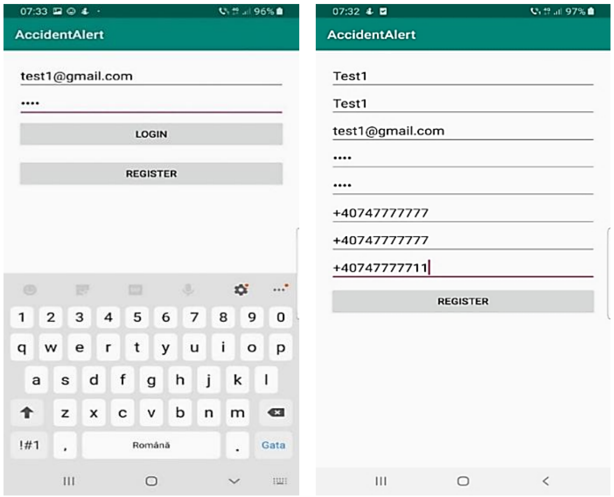
Another important aspect of the Android part of the application is location permissions. Once the application is installed, if the location is not activated, the user receives a message which requests permission to the location of the smartphone. The application will not work without the enabled location.

For security purposes, the server can only be accessed by admin. Also, the Android application contains the exact path of the Http request to the server. Since the server is built on an AWS virtual machine, many hackers may try to access the server, but this is not possible due to server security. Any other application that tries to access the API will receive error 404. In conclusion, as far as server security is concerned, no other foreign application can access the Linux machine.

Another important objective of the paper was to achieve the lowest energy consumption. A battery consumption statistic was obtained through experiments (the application runs all the time in the background). The results are presented in Table I.

As can be seen from the performed tests, the proposed system requires a maximum consumption of 15.2% of the battery, without it being included in the list of applications for battery optimization.

Except for hardware malfunctions that may affect the sensors of the phone, the application has no issues in terms of life, because it works in optimal parameters.



test1@gmail.com

....

Test1

Test1

test1@gmail.com

....

....

+40747777777

+40747777777

+40747777711

(a) (b)

Fig. 8. (a) Login Activity. (b) Register Activity.

TABLE. I. BATTERY CONSUMPTION

Time period	Battery percentage
One day	10,6 %
A day that sends at least 6 alerts, maximum 10 alerts	15,2 %

A. *Making Statistics using Grafana Application*

The proposed system offers, besides the mobile monitoring and a warning application in case of accident, the possibility of making statistics regarding the number of detected accidents for each user. This can be done by accessing the Grafana application which is also installed on the Linux virtual machine.

Logging into the Grafana application can only be done by the admin, using the username and password as credentials. Once logged in, the admin can view user accident statistics for a certain period of time. The server has a script set in the Linux scheduler to run every minute. This script is connected to Firebase and extracts the name of each user and inserts into the local MySQL database the user name, date and number '0'. In this way, we will have a table in the local database where there will be multiple lines with the names of the people and the digits 0 or 1. The value 1 is inserted only when the API is called by the Android application.

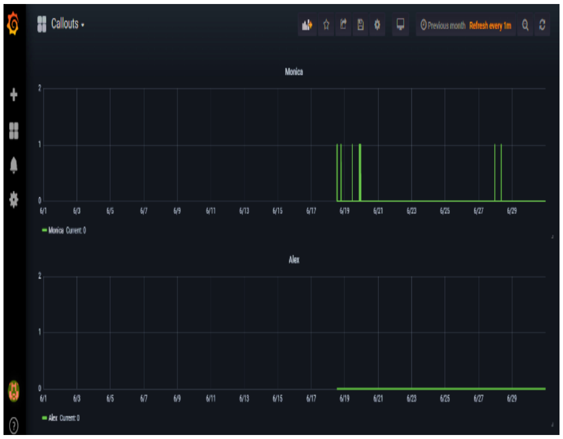


Fig. 9. Accident Statistics – June 2019.

The accident statistics for some users for June 2019 are presented (Fig. 9).

V. DISCUSSION AND CONCLUSIONS

One of the biggest challenges was to keep a service permanently running without affecting the rest of the applications. This is necessary to be able to monitor the users 24/24h. In the latest versions of Android, different process prioritization mechanisms have been implemented in order to save the battery and the processes are often stopped. Also, different phone manufacturers can change these mechanisms. For example, on Huawei, events sent by the accelerometer stop working after a few minutes. This issue can be solved by excluding the application from battery optimization in the phone settings. Also, a wake lock was used to increase service priority.

Another challenge is to detect the fall event and to remove false alerts. Determining the optimum acceleration threshold contributes to solve the problem only partially.

By combining all the technologies presented above, we have succeeded in implementing a modern solution, which adapts to the size of each mobile device and which is built on the principle of simplicity in use.

Looking at the context from the beginning of the paper, this application is very useful for helping people with disabilities, or the elderly, because it significantly reduces the risk of health hazard after an accident or even the risk of death. It can bring a huge contribution to the society, by allowing medical staff to do other works instead of monitoring people 24 hours a day at the hospital or at home.

A solution has been found to reduce the death of accidents due to the use of today's advanced technologies, for example, smartphones, sensors incorporated in smartphones, fast communication services. This will help the monitoring of helpless people even remotely. In this way, the elderly, people with disabilities or chronic diseases can be saved in time in the event of an unforeseen accident and also the monitoring of accidents is provided without affecting the mobility or quality of life.

In conclusion, the application complies with all the constraints set from the beginning, as can be seen from the results of testing and validation. The proposed application is not similar to other and works better by optimizing power consumption, displaying a user-friendly interface, using a database that updates very quickly and using a server that can send alert messages within three seconds of receiving alerts on smartphones.

As future improvement of the proposed solution, the use of machine learning technologies is desired. Firebase ML kit offers the possibility to train a sensor flow model for categorizing events and using artificial intelligence to decide whether it is a false alarm or not. It is possible to record the last sequence of values sent by the accelerometer and through the input of the user who decides whether it is a false alarm or not, to train the AI part of the application.

The statistics regarding the number of accidents are very important. These statistics can provide information on the number of falls produced in a given time, as well as information on which category of people are most prone to falls. So, another improvement may be the update of the dashboard with an automatically created graph when the application detects a new registered user. Now, it is up to the admin.

As a further development, the use of ML for falls detection and for minimizing false results can be a good solution. In this way, using the dataset obtained from the sensor, the application can learn the behavior of the user and can predict depending on the current movement if a fall is imminent or not.

References

[1] Mohammad Ashfak Habib, Mas S. Mohktar, Shahrul Bahyah Kamaruzzaman,Kheng Seang Lim, Tan Maw Pin, and Fatimah Ibrahim. “Smartphone-Based Solutions for Fall Detection and Prevention: Challenges and Open Issues”, in Sensors (Basel) 2014 Apr.

[2] World Health Organization, Disability and health, 16 Junuary 2018, Article available online: .https://www.who.int/news-room/fact-sheets/detail/disability-and-health.

[3] National Institute of Statistics: General mortality rate, Available online: https://www.mediafax.ro/social/institutul-national-de-statistica-rata-mortalitatii-generale-dar-si-a-celei-infantile-mai-mare-in-romania-decat-media-ue-foto-14915507.

[4] World Health Organization, Falls, 16 Junuary 2018, Article available online: https://www.who.int/news-room/fact-sheets/detail/falls

[5] S. Rauscher, G. Messner, P. Baur, J. Augenstein, K. Digges, E. Perdeck, G. Bahouth, and O. Pieske. “Enhanced Automatic Collision Notification System- Improved Rescue Care Due To Injury Prediction- First Field Experiencee”, 2009.

[6] Y. Li, K. C. Ho and M. Popescu, "A Microphone Array System for Automatic Fall Detection," in IEEE Transactions on Biomedical Engineering, vol. 59, no. 5, pp. 1291-1301, May 2012. doi: 10.1109/TBME.2012.2186449.

[7] A. Irtaza, S. M. Adnan, S. Aziz, A. Javed, M. O. Ullah and M. T. Mahmood, "A framework for fall detection of elderly people by analyzing environmental sounds through acoustic local ternary patterns," 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, 2017, pp. 1558-1563. doi: 10.1109/SMC.2017.8122836

[8] Y. Li, K. C. Ho and M. Popescu, "Efficient Source Separation Algorithms for Acoustic Fall Detection Using a Microsoft Kinect," in IEEE Transactions on Biomedical Engineering, vol. 61, no. 3, pp. 745-755, March 2014. doi: 10.1109/TBME.2013.2288783.

[9] Y. Zigel, D. Litvak and I. Gannot, "A Method for Automatic Fall Detection of Elderly People Using Floor Vibrations and Sound—Proof of Concept on Human Mimicking Doll Falls," in IEEE Transactions on Biomedical Engineering, vol. 56, no. 12, pp. 2858-2867, Dec. 2009. doi: 10.1109/TBME.2009.2030171.

[10] S. C. Agrawal, R. K. Tripathi and A. S. Jalal, "Human-fall detection from an indoor video surveillance," 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Delhi, 2017, pp. 1-5. doi: 10.1109/ICCCNT.2017.8203923.

[11] Z. Huang, Y. Liu, Y. Fang and B. K. P. Horn, "Video-based Fall Detection for Seniors with Human Pose Estimation," 2018 4th International Conference on Universal Village (UV), Boston, MA, USA, 2018, pp. 1-4. doi: 10.1109/UV.2018.8642130.

[12] A. Y. Alaoui, A. E. Hassouny, R. O. H. Thami and H. Tairi, "Video based human fall detection using von Mises distribution of motion vectors," 2017 Intelligent Systems and Computer Vision (ISCV), Fez, 2017, pp. 1-5. doi: 10.1109/ISACV.2017.8054942.

[13] G. M. Basavaraj and A. Kusagur, "Vision based surveillance system for detection of human fall," 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, 2017, pp. 1516-1520. doi: 10.1109/RTEICT.2017.8256851.

[14] J. Limpanadusadee, P. Kesawattana, T. Wongsawat and D. Wongsawang, "EldTec: Improvement on Wearable Sensor for Elderly Fall Detection," 2018 Seventh ICT International Student Project Conference (ICT-ISPC), Nakhonpathom, 2018, pp. 1-6. doi: 10.1109/ICT-ISPC.2018.8523991.

[15] P. Pierleoni, A. Belli, L. Palma, M. Pellegrini, L. Pernini and S. Valenti, "A High Reliability Wearable Device for Elderly Fall Detection," in IEEE Sensors Journal, vol. 15, no. 8, pp. 4544-4553, Aug. 2015. doi: 10.1109/JSEN.2015.2423562.

[16] T. Tamura, T. Yoshimura, M. Sekine, M. Uchida and O. Tanaka, "A Wearable Airbag to Prevent Fall Injuries," in IEEE Transactions on Information Technology in Biomedicine, vol. 13, no. 6, pp. 910-914, Nov. 2009. doi: 10.1109/TITB.2009.2033673.

[17] K. Suganya, Dr. M. Saravanan, “A Survey on Monitoring and Fall Detection System for Old Aged Person”, in International Journal for Research in Applied Science & Engineering Technology (IJRASET), vol. 6, Nov. 2018.

[18] Miguel K, Brunete A, Hernando M, Gambao E. “Home Camera-Based Fall Detection System for the Elderly”. In Sensors (Basel). 2017;17(12):2864. Published 2017 Dec 9. doi: 10.3390/s17122864.