

@Sensor - Mobile Application to Monitor a WSN

Nuno Moreira¹ Marco Venda¹ Catarina Silva¹ Luis Marcelino^{1,2} António Pereira^{1,2}

¹Computer Science and Communication Research Centre (CIIC)
School of Technology and Management, Polytechnic Institute of Leiria
Leiria Portugal

²INOV INESC INOVAÇÃO
Leiria, Portugal

{2101311, 2100038}@my.ipleiria.pt {catarina, luis.marcelino, apereira}@ipleiria.pt

Abstract—Due to the growth of potential of Wireless Sensor Networks (WSN), now becomes possible to use this technology in various new generation services such as monitor environment characteristics like temperature, humidity, among others. WSNs are emerging as a new paradigm, gathering information in a collaborative way, where independent sensor nodes collect and share information. In the last few years ubiquitous computing popularity has grown considerably, ambient intelligence and home automation systems are having a good public acceptance. In this context, we developed a mobile application to remotely monitor one WSN, installed at a customer's home. The application's main goal is to interact with a WSN, allowing a user to consult sensor states and receive sensor alerts. Sensor alerts will be received, in case a sensor's threshold value, exceeds the limit. Therefore the user will be notified whenever there are changes in the WSN. Apart the WSN, a base station is also needed for keeping the state of the WSN, store sensor's information and enable communication between the WSN and the developed mobile application via Web Services. This way and having a network connection, the user can check the state of any WSN component, anytime, anywhere. The Android platform was chosen to implement our application, since it is having a great acceptance and continued growth in the mobile market.

Index Terms—Mobile, WSN, Monitor, Android

I. INTRODUCTION

Wireless Sensor Networks (WSN) are emerging as a new paradigm, consisting in the collection of information in a collaborative manner, where independent sensor nodes collect and share information. There are many next generation services that use this technology, since this type of networks has broad applicability. Recently, the quick and increasing development of low power wireless communication devices and microprocessors, in conjunction with the significant progress in the protocols for Ad-Hoc networks and pervasive computing, permitting the application of WSN in various areas, from industrial to home automation [1] [2]. Often, WSNs monitor a particular environmental value, such as the temperature of a given area, or the movement of a given object. We can use these sensors to

constantly monitor certain events, and if some predefined value is reached, an alert is generated. The control of generated events is recorded in a control station that interacts with the WSN. The control of the network has to be done locally on the control station where the data is aggregated. In order to allow the remote interaction with the WSN as well as the reception of generated alerts, we developed both, a mobile and a server application, enabling the interaction to the WSN.

With the continuous growth of the mobile market, in the future, Internet access will be accomplished mainly through mobile devices [3]. Android platform is growing fast and many developers are adopting this platform. In addition, users are also very receptive, making Android an ideal market for our application development.

Throughout this paper we will present the proposed solution and describe the main decisions evolving the project's development. In Section II we cover the related work, followed by the high level architecture of the proposed solution in Section III. The IV Section discusses the implementation of the application and its features, followed by Section V where we present the performed tests. In Section VI we present our conclusion and finally on Section VI we propose some future work.

II. RELATED WORK

To the best of our knowledge, there are no mobile applications with the functionality offered by our proposed solution. Much of the research is still focused on the implementation of the WSN and not in the interaction with such networks via mobile devices. During our research we found only one mobile application related to WSNs, based on a three-dimensional representation of the area covered by the WSN. The main goal of vsiogap3d project is to (and we quote) “*develop an application that presents a three-dimensional model of a building. It should also connect to the data repository and download building sensor information such as temperature and lighting conditions in individual rooms*” [4]. Thus, the proposed approach turns out to be a pioneer in the remote interaction with the WSN via a mobile device.

III. PROPOSAL APPROACH

We developed a mobile application that allows users to query the state of a WSN anytime, anywhere. To this end, there is a need to link some essential components, such as a WSN and a proper interaction application, an Internet connection, a Control Station with the proper Web Services, a relational Data Base and finally a mobile device. This section presents the general architecture of the proposed solution, showing the relationship between the various components, as we can see in figure 1.

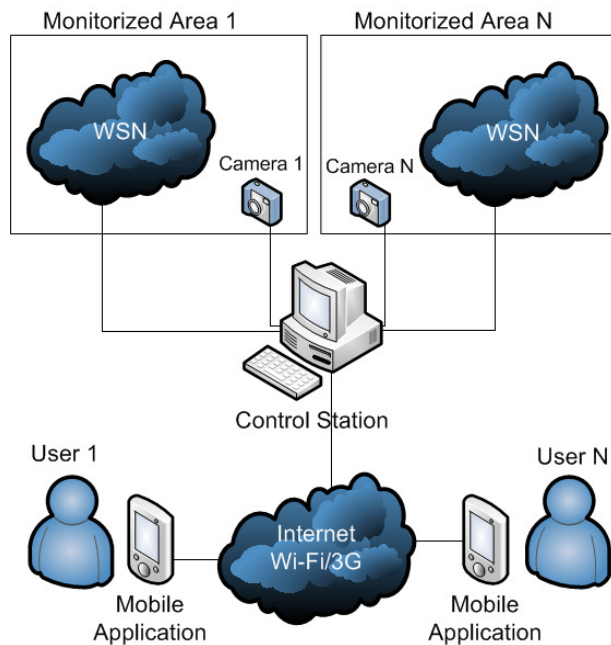


Figure 1 - Proposed Architecture

The proposed architecture consists in several elements that communicate with each other, gathering information about all sensor nodes, in order to be able to remotely monitor and control a WSN. Sensor nodes can be installed in several monitoring areas, which may or may not contain digital cameras. Digital cameras have the functionality of capturing an area's photo, being activated in the moment an alert occurs. This way, beyond having an alert message, users can also receive a related image to verify the reason of the alert's occurrence when requested. On the other hand, with our application, users can request a real time image of any camera installed, to visual monitor any area [5].

Upon detection of an event by a sensor node (when a certain monitored value exceeds a predefined threshold), it will inform the control station of the occurrence. In turn, the control station will register the event and request (when installed) the corresponding area's digital camera to take a snapshot. After this, the control station will browse the list of users associated with the sensor node that detected the event and will provide users with the event's data, via SMS and/or email. In case of an email reception, the user will receive the event's snapshot automatically.

The control station is aware of all monitored areas, sensors and cameras belonging to each area. When installing/configuring a WSN it is necessary to configure the exact location for each sensor and camera in every area. This information is very important for the mobile application, allowing it to draw each sensor node in a virtual plant, according with the given location. Thus, the user can access any sensor through the plant, being easier to identify each sensor. Despite the registration of all WSN components in the control station, it is also necessary to configure the accounts of users with access to the WSN. In addition to the authentication data, the phone number and E-Mail of each user has to be added. This way, the control station will gather all the necessary information about users, to allow the communication to be set. In each request, the user must be identified to access information from the network.

As shown in figure 1, the mobile application uses the Internet to communicate with the control station. An Internet connection is essential for a successful operation, since it aims to allow users to have a remote access to a server. This connection can be made through 3G or Wi-Fi, allowing the user to communicate with the control station anywhere. In turn, the control station will communicate with the WSN/Cameras and forward the requested information to the mobile application. Once the traffic volume is not high, the use of a low-speed 3G connection is possible without significantly degrading the quality of experience of the application.

Since the application is dependent on Web Services communications to perform most of its functionalities (e.g. consul a specific sensor value), a poor connection to the server can occasionally result in a bottleneck to the application's performance. Thus, the application is mostly dependent on the network's speed and the server's performance returning all the requested information, resulting in a key component for a smooth operation.

The mobile application does not have the same capabilities when in online and offline modes. In offline mode, the user is able to receive new alerts and to consult the local registered alerts; see the images previously download to the local storage. On the other hand, besides having the offline mode abilities, in the online mode the user is able to consult sensor values, areas, floors and cameras; take image snapshots; synchronize alert database.

When running the mobile application, the battery lifetime is dependent on many circumstances, like the number of alerts received, the number of consults that the user performs, the need to read and write information on local database and the number of image downloads. Since the most of these actions depend on a network connection (Wi-Fi/3G) usually battery-intensive, a high usage of the application can greatly reduce the mobile's battery life time.

In the next section we present the prototype developed to demonstrate the operation of the proposed concept.

IV. IMPLEMENTATION

This section presents the choices made to implement the desired solution. In order to simplify the presentation of the whole solution, the components will be presented separately.

A. WSN

In order to implement the proposed solution a WSN is required. To this end, we used a prototype currently being developed at CIIC – Computer Science and Communication Research Center¹. The WSN is based on the TinyOS operating system and uses the sensor nodes iMote2. The application layer used to interact with the WSN was developed by the elements within the research project where the development of WSN is included. We developed the Web Services and Database functionalities.

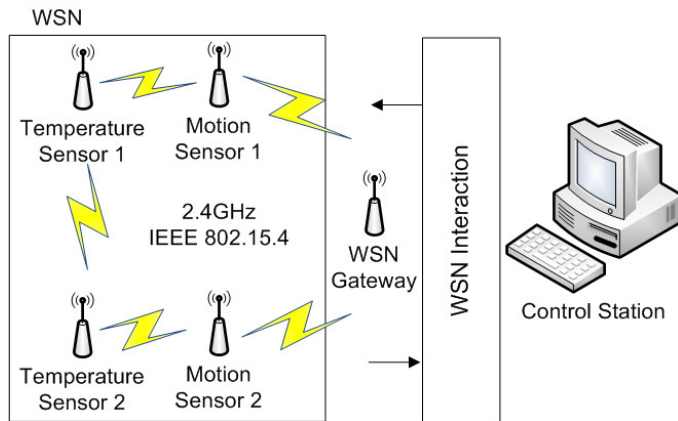


Figure 2 - Wireless Sensor Network Component

After the proper integration, the control station can interact with the WSN, recording the events occurred, being capable to respond to mobile application's requests. On Figure 2 we can see the WSN component.

B. Control Station

As mentioned earlier, the control station is a very important component in the proposed solution, representing an interface between the mobile application and the WSN. In this subsection we present the components related to the development of our application (Web Services and Database) shown in figure 3, the representation of the components used on control station.

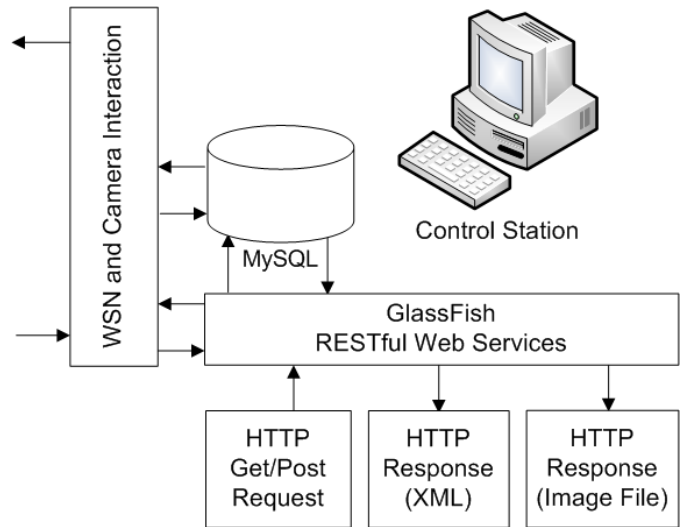


Figure 3 - Control Station Component

Initially, the mobile application sends a request to the control station and in response; the control station sends back a XML or Image file containing the requested information. All communications between the control station and mobile devices are done via Web Services. Resorting to this type of technology, the solution's growth is guaranteed, especially for the use of new content types. The use of other platforms is also possible without the need to restructure the applications architecture.

After studying the possible web service solutions, SOAP and RESTful, the choice fell on RESTful as it presents a lesser burden on communication and requires no additional toolkits on the Android platform, being based on the most common communication architecture of the Internet (HTTP). We used the Glassfish application server to host the required web services. Since all the code of control station is based on Java language, Glassfish seems to be the most appropriated choice, since it is free and has good support for RESTful web services. The use of such language allows portability across multiple platforms, making the control station adaptive to various development environments. Regarding the database used, the choice fell on the MySQL Relational Database Management System (RDMS). As Glassfish, MySQL is free and has the ability to run on multiple platforms.

C. Mobile Application

The developed application allows users to consult sensors and cameras, being also possible to analyze the registered alerts. The application is divided in many sub-sections to grant a good data presentation. This way, is possible to consult sensors by a house plan, by floor or by type. The house plan option, will lead to a 2D image that represents a visual area distribution, where the related sensors will be shown. This way, interacting directly with the plan, users can access any sensor. When consulting sensors, the user will be informed about their name, current value and threshold, being also informed about the sensor's location. When consulting the registered alerts, user will be able to filter the consult by area,

¹ Computer Science and Communication Research Center, Polytechnic Institute of Leiria. [Online]. HYPERLINK "<http://ciic.ipleiria.pt/>"

sensor, all alerts or consult just the alerts from the sensors that are being monitored by the user. Users will be informed about the alert's value, its threshold and date, and it will be presented some additional information about the sensor that triggered the alert.

To receive alerts via SMS and/or Email, users must initially select the sensors they wish to monitor. When selecting the desired sensors, users can customize alerts reception, by defining two different values, the user's personal threshold (minimum value for each sensor to be reached, so that the user should be notified) and the number of occurrences in a time range to be notified (e.g. only be notified if 3 alerts occur in 120 minutes). Therefore, users will receive alerts according to the different settings previously conducted. After defining these values, users can change them any time. When these values are not defined, the application will assume the default values.

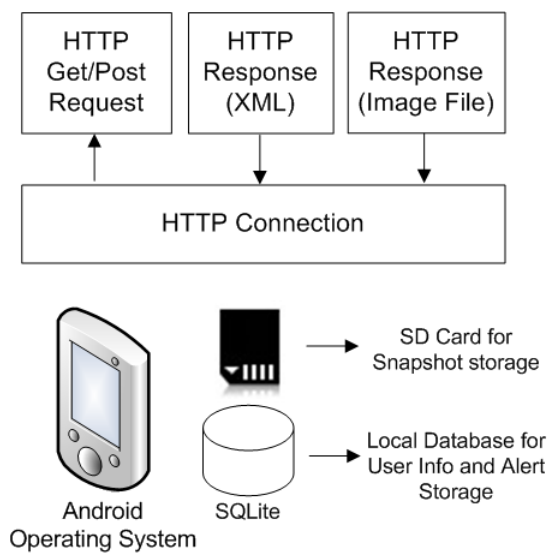


Figure 4 - Mobile Application Component

As shown in Figure 4, we used a SQLite database on the mobile device. This database is used to store the user's authentication information and the alerts from the sensors to which the user is associated. When connected to the control station, the user can synchronize the alerts from the local database with remote database, allowing the consult of alerts in offline mode.

As noted above, the mobile application allows the user to capture an image through the cameras installed in various areas of monitoring. To this end, the user can choose the desired camera that will perform the capture and the resulted image will be downloaded to the mobile device immediately. Besides capturing snapshots whenever user wishes, it is also possible to download and check images captured at the occurrence of an alert. The user can view a list of alerts and download the image associated with the respective alert. With the images stored locally, the user is able to see the images whenever he wants. With the continuous growth of data, there is the possibility of exhausting the device's memory space. Image files can flood device's built in memory, so the

application is prepared to store all image files in the mobile's SD Card. Thus, users can download a great number of images without compromising the internal storage. The implemented gallery, organizes the images by "alert images" and "user images", where the "alert images" are the images associated with alerts and "user images" are the images captured by the user.

As the mobile application mainly depends on the interaction with web services. Since we decided to use RESTful web services, there was no need to install or develop an API to handle SOAP web services. In order to invoke the web services provided by the control station, HTTP connections are used, as we can see on Figure 4.

Using the Internet, the control station can take some time to send a response, depending on the network's quality. To avoid blocking the application UI, we resorted to the use of AsyncTask class, "An asynchronous task is defined by a computation that runs on a background thread and whose result is published on the UI thread" [6].

Depending on the request made by the user, the corresponding web service will be invoked, for example, if the user wishes to know the current value of a sensor, the HTTP GET request is sent with the identification of the desired sensor. On the server side, upon receiving the request, the control station will communicate with the sensor node and receive its current value. After receiving the sensor's response, the control station will create the proper XML and forward the response to the mobile application. This communication process is illustrated on Figure 5.

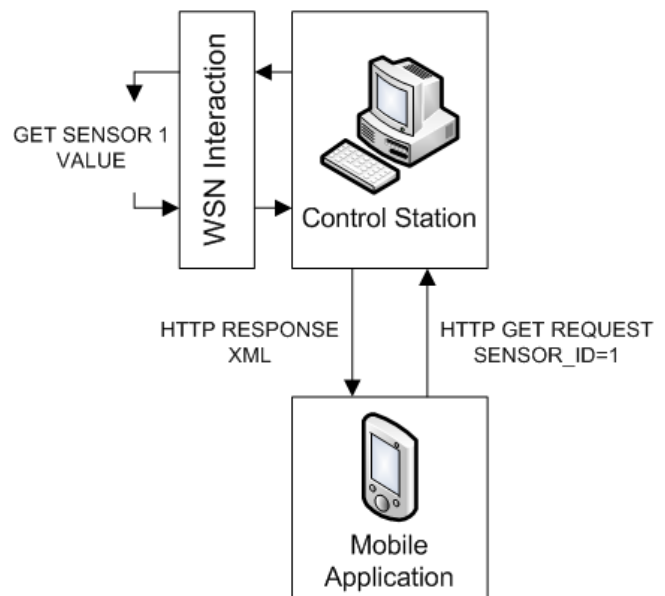


Figure 5 - Components Interaction

With the exception of image files, all content sent by the control station is in XML format. The use of XML enables the application scalability to new types of content and other platforms. When making a particular request, the mobile application knows what XML format is expected to be

received in response. After receiving the content, the mobile application will parse the received XML.

The application's user interface was carefully thought, to avoid users losing too much time looking for, how to interact with the application. Thus, we preferred to use more screens with organized information than using fewer screens with too much information. All the menus are followed by a proper icon or a description to give the user a good perception of what is intended in each menu row. The main screens were designed to gather and present information in a very simple way, avoiding visual confusion. The application is able to run in mobile devices containing the last generation screens, being capable to adjust all the contents to different resolutions. The entire application is designed to provide a good interaction by the user, covering some important aspects, like:

- **Visual Difficulties:** The displayed screens use shades of black and gray, so that people with colorblindness can easily distinguish and read the screens content.
- **Handicap Difficulties:** The application requires a reduced number of user actions to achieve an expected result. All the displayed items have a considerable size in order to help handicapped people to select the desired component, avoiding selection failures.
- **Context Awareness:** The screen title is used to contextualize users, in order to provide additional information about what is intended in each screen. Every screen has its corresponding title.

On the next section we present the performed tests and the conclusions taken after its execution.

V. EVALUATION

During the evaluation process of the mobile application, we adopted the 10-cent qualitative test model [7]. Five participants with ages understood between 21 and 28 years were used to perform the tests, characterized by having a college degree and some experience with the use of smartphones (OS: Android).[8]

The main goal of the performed tests was to verify the interface perception and the performance achieved by each participant, when using the developed application.

A supervisor was present during each test, monitoring the total time that each participant took to accomplish each task. He was also responsible to take some important notes, related with all the participants' errors and difficulties during the interaction with the application. This way, important data was gathered so the development team could interpret it and carry out possible adjustments to the application. The supervisor only guided the participants throughout the testing phase and did not help them perform any task.

Since the application can be used everywhere (that allows Internet connection), regardless of external variables and noise, the testing location was not relevant. To perform the tests, two smartphones were used, a HTC Magic and a HTC

Dream (G1) with the Android operating 2.2 and Android operating 2.1, respectively. The WSN's testing prototype was consisted in tree sensors, two moving sensors and one temperature sensor, followed by a webcam, to take some area snapshots.

The conducted tests consisted in two testing phases that will subsequently be presented:

A. First Testing Phase

These tests were performed after the conclusion of the initial development phase. A questionnaire containing six different tasks to accomplish was delivered to each participant before beginning the tests, in order to assist us collecting data, throughout the conducted interviews.

The tasks presented on the participants questionnaire where:

1. Change the application settings to remove the need to authenticate the user, when starting the application;
2. Consult the current value and threshold from sensor "MOV1", located in "ROOM1". After this, take a snapshot of the area and open the same snapshot through the image gallery;
3. Using the house plan, select the sensor located in the "ROOM 2", choose to monitor it and customize its data with the following values: Threshold: 20, Alerts Number: 5, Interval: 3;
4. After receiving an alert via email, the participant should access the camera from the corresponding area and take a snapshot;
5. List all the received alerts, from the sensor of temperature named "TEMP1" located in "KITCHEN".
6. Access to the gallery and view the user's personal snapshots. Make "Zoom" the first image displayed.

The completion of the tests mentioned above, allowed us to detect possible improvements to the application, relevant to its performance and perception. After collecting and analyzing the testing data, the application suffered some modifications in order to become more efficient. Some of the most relevant improvements made, are described below:

- In order to prevent mistakes during the application's configuration process, an extra field related with the password confirmation, was created.
- The process of activating a sensor for monitoring proved to be quite messy and time consuming, due to the requirement of following a few steps. A significant improvement to the interface was made, in order to gather all the necessary data to accomplish this task.

Thus, the process of activating a sensor to monitor and its data set, became more intuitive to the end user.

- The Email alert notification, revealed to have a very confusing information disposal. Users revealed difficulties in interpreting/finding the information they pretended in the email. To resolve this state, all the email info organization has been redrafted to allow a more effective consult.
- While consulting photos, in general, users revealed difficulties in realizing the meaning of each sub-menu displayed. In order to make this task easier, each sub-menu description has been improved, making it clearer to understand.

After the improvements mentioned above (among others), a second test phase was performed and it is described below.

B. Second Testing Phase

The second testing phase was performed, in order to verify if the changes early made, meet the user's needs. To perform this test, we used the same participants and questionnaires as the first testing phase, so conclusions could be made.

We concluded with these last tests, that there was a great acceptance of the changes by the participants and their performance increased significantly. In general, the test results were positive, revealing extremely valuable information that we used to improve the application.

VI. CONCLUSION AND FUTURE WORK

The market for mobile devices is growing and in the future, all mobile devices will have Internet access. This way it is possible to develop mobile applications that can take advantage of the Internet, and improve the quality of future applications.

The main contributions of this project are the concept of interacting with a WSN and remotely monitor the state of all sensor nodes, through a mobile application.

Tests were performed with the use of a prototype and revealed that the developed solution successfully interacted with the WSN and the evolved components. It proved that is possible to remotely interact and monitor a WSN through a mobile application. Using the application a user is notified if an event is detected in WSN, being also possible to check the status of the WSN from any location, requiring only an Internet connection.

Some of the future work will be to improve the application's user interface, especially the graphical layout related with the house's plan. In this case, functionalities like pinch and zoom could be added in order to improve user's interaction. Since the communication between the mobile device and the control station involves confidential data, it will be important to add a security layer (SSL) to the communication, ensuring that potential attackers do not

intersect data. There is also the need to conduct more tests with users, in order to improve the overall solution.

ACKNOWLEDGMENT

We want to thank CIIC members for their support in developing our solution. The provided WSN allowed us to test the solution as a whole, testing the interaction between all components of the implemented solution.

REFERENCES

- [1] I. F. Akyildiz, "Wireless Sensor Networks: a survey," *Computer Networks*, vol. 38, pp. 393-422, March 2002.
- [2] H. Karl and A. Willig, "A short survey of wireless sensor networks," *Telecommunication Networks Group*, October 2003.
- [3] (2010, October) Wireless Sensor Networks and Home Control. [Online]. HYPERLINK "<http://www.abiresearch.com/research/1005845-Wireless+Sensor+Networks+and+Home+Control+>" "<http://www.abiresearch.com/research/1005845-Wireless+Sensor+Networks+and+Home+Control+>"
- [4] (2011, January) vsiogap3d - 3D Visualisation of Sensor Information on Google Android Platform. [Online]. HYPERLINK "<http://code.google.com/p/vsiogap3d/>"
- [5] Nuno Moreira, Filipe Felisberto, Isabel Marcelino António Pereira, "A scalable architecture for Body Area Networks," *EURASIP*, To be approved.
- [6] (2011) AsyncTask - Android Developer. [Online]. HYPERLINK "<http://developer.android.com/reference/android/os/AsyncTask.html>"
- [7] Elder Care's Fall Detection System, "Filipe Felisberto, Nuno Moreira, Isabel Marcelino, Florentino Fdez-Riverola and António Pereira," 2011.
- [8] Jakob Nielsen, *Quantitative Studies: How Many Users to Test?*, 2006.