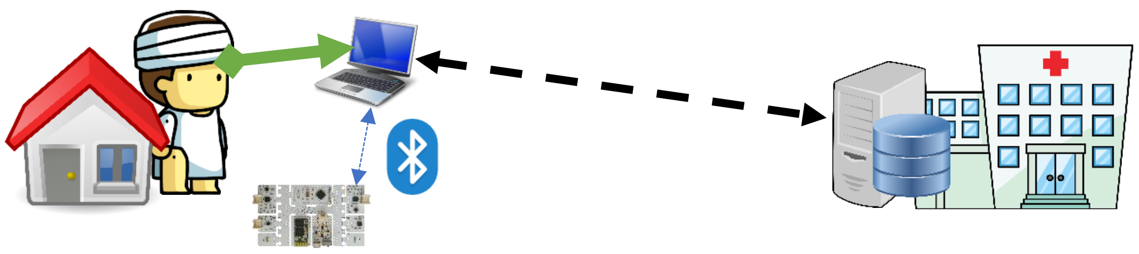
Final Project

Telemedicine Project

You should build a telemedicine application whose purpose is the supervision from the patient's home of a chronic disease. You should choose the chronic disease in question (some examples may be chronic obstructive pulmonary disease, heart diseases, diabetes…).



The application should allow the patient to send to a server located in the hospital signs and symptoms reported by the patient that are relevant for supervising the disease in question (e.g. level of fatigue, dizziness, chest pain, difficulty breathing…) using some (graphic or console based) form. The symptoms and signs, together with the date, must be stored on the server associated with the patient's clinical history. For this purpose, files or a database can be used. The server and the patient's application are **two different applications** that should be clearly differentiated and that can run in different computers.

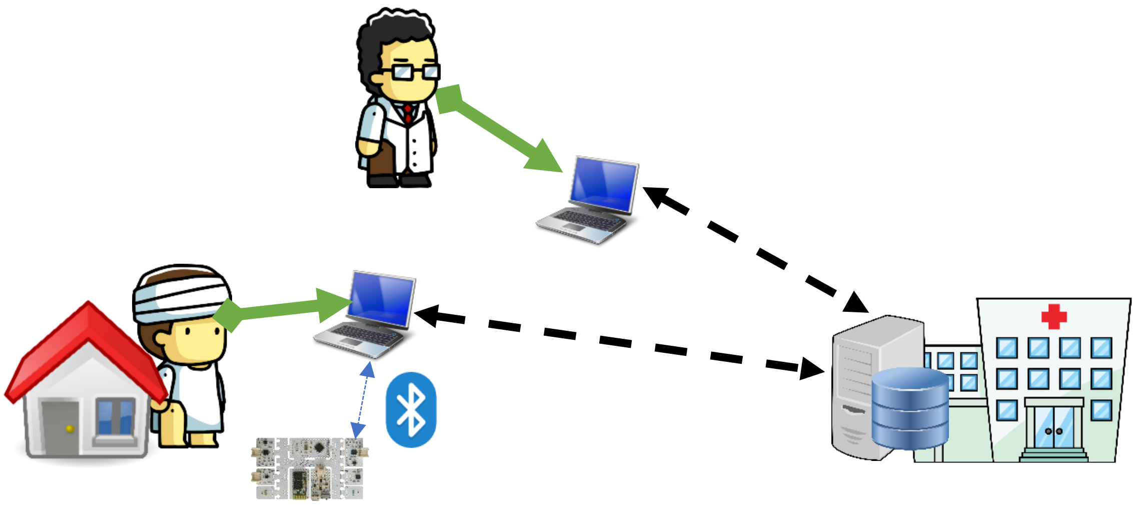
Once the patient has submitted this information, the application will allow the patient to begin registering and sending to the server at least two physiological parameters. The registration of these physiological parameters must be made by a Bitalino device, to which the application will connect using Bluetooth. The physiological parameters received on the server must be stored in a file containing the patient's name and the date and time of the monitoring. For this purpose, a file must be used. The monitoring of physiological parameters may either last a fixed amount of time or be stopped by the patient.

The server located in the hospital must allow the connection of **multiple patients at the same time**. The server should have some kind of user interface that allows the administrator to stop it, ideally after checking that there aren't any clients connected. To carry out this task a password must be provided. This user interface could be through the network, or not.

Up to this point the project (including the documentation) can get a **maximum** grade of 6 if it is based on the console or 7.5 if it is done using a graphical user interface.

The **robustness** of the application to network failures, clients' disconnections, patients entering incorrect data, connection failures to Bitalino and in general any failure that might occur will be also considered in the grade.

Doctor's connection (Up to 2 points)



The project can be extended to allow the doctors connection to the server. The doctors should be able to see all the information that the patients had send to the hospital, and where appropriate modify or add more information. This should be a new application capable of running in a different computer.

Additional functionality for the project

Possible additional functionality for the project:

- Allow user identification by using usernames and passwords, the password must not be stored in plain text. You should allow to create new users dynamically. (up to 1 extra points).
- Use public-key encryption, avoiding that somebody could listen to the stream without the key. (up to 1 extra points).
- Allow user or administrator connection using Telegram, smartphone application, web, etc. (up to 1.5 extra points).
- Allow the doctor to view in the server in real time the physiological parameters that are been registered form the patient/patients in a graphical interface (up to 1 extra point).
- Other ideas about additional functionality for extra points are also possible and should be discussed with the professor.

The robustness of the application to network failures, clients' disconnections, patients entering incorrect data, connection failures to Bitalino and in general any failure that might occur will be also considered in the grade.

Deliverables

Along with the project’s source code should be delivered a **user manual of the application**. In this manual you should describe the **condition** for which the application was designed and why the symptoms and signs sent are relevant for supervision and how do you use the patient client if proceed explain also de doctor’s client. You should also deliver an **admin manual** that explains step-by-step what you need to do to run the server, to execute the application, etc. you should also explain the **protocol** that the application uses to communicate between the client and the server.

- Source code
- User Manual of the application
- Admin manual

The code should work “**out of the box**” without needing to modify or include anything in the code.

Delivery Date:

- **Last day ->12/12/2019**
  - Defense before the 19/12/2019

The project **must** be defended individually by each member of the development team. The **maximum** grade a student may get in the defense is the note assigned to the project, although depending on the student's defense the grade can be lower. All team members must know the detailed operation of all parts of the application. If it is considered that the defense is not passed the grade of the project and therefore of the subject will be a **failing grade**.

Planification (per weeks)

0. Create groups
1. Choose a disease, study the disease and choose the signals and parameters to process
2. Design and code the interface with the Bitalino
3. Store the information in files
4. Complete the client
5. Design the communication protocol between client and server
6. Implement the server (One Client)
7. Implement the server (Several Clients)
8. Interface of the Server and mechanism to close the server
9. Tests of the application and documentation
10. (Optional Elements) Implement the doctor client
11. (Optional Elements)
12. Complete tests of all the application (with different computers and the Bitalino). Everything is working and we have **completed all the requirements**. Read this document again before delivering the project.
13. Reserve
14. Reserve

Resources and links

Bitalino SDK Java in Github

https://github.com/BITalinoWorld/java-sdk

Bitalino Documentation

http://bitalino.com/en/learn/documentation

Java tutorial about how to use MD5 for storing passwords

https://howtodoinjava.com/security/how-to-generate-secure-password-hash-md5-sha-pbkdf2-bcrypt-examples/

Telegram API for bots

https://core.telegram.org/

Java library to create Bots and examples

https://github.com/rubenlagus/TelegramBots

https://github.com/rubenlagus/TelegramBotsExample

https://monsterdeveloper.gitbooks.io/writing-telegram-bots-on-java/content/chapter1.html

Android API for reading from the Bitalino

https://github.com/BITalinoWorld/revolution-android-api

Example project communication with Bitalino in Java

(Student Portal)