APPUNTI PER STILARE IL RASD

During the

**ACTORS**

* Patient: the person who wears the smartwatch (equipped with a SIM) or, in alternative, a similar device containing the sensor coupled to *(Si usa to o with? Non ho controllato)* a smarthphone through a Bluetooth connection.
* Third party: a company which is interested in monitoring population’s health status (e.g. a health insurance, a pharmaceutical company, the government)
* The ambulance man in charge?????? [Dobbiamo riflettere di più su questa cosa secondo me]

**PHENOMENA**

In this part we expose the phenomena we consider relevant in order to model the part of world of interest.

**WORLD PHENOMENA**

* Diseases /absence of disease
* Emergency situations
* Ambulance moves and arrives
* Ambulance breakdown
* GPS does not work 🡪 assumeremo che funzioni sempre bene
* Cellular is not connected 🡪 segnala che non è connesso

**SHARED PHENOMENA (phenomena that could be either controlled by the world and observed by the application or controlled by the application and observed by the world)**

**[Controlled by the world]**

* Detection of health values
* Request for data from the patient or the third part
* Ambulance ack
* Ambulances informations (position and status)
* Confirmation of good health status by the patient
* Registration to the service by a user or a third party

[I seguenti sono stati aggiunti perché, nel nostro approccio, possono essere in qualche modo rilevati dalla macchina]

* Sensor breakdown
* Application breakdown
* Network breakdown

**[controlled by the machine]**

* Sending data to the third party
* Showing data to the patient
* Sending an alert to an ambulance
* asking the patient to confirm his health status

**MACHINE PHENOMENA**

* Database queries
* Database inserts
* Database creation
* Data analysis and comparation with thresholds
* Shortest path computation
* Communication between the application and the server (SPESSO COMPARE NEI REQUIREMENTS, VA TOLTA): i.e., sending and receiving messages between the app and the server
* Matching ambulances and emergencies
* Data elaboration for showing

**NB:** stiamo assumendo che l’applicazione lavori comunque con un dispositivo indossabile!!

**NB:** sempre mantenere la distinzione tra user e third parties in Data4Help (in SOS le third parties invece non saranno più utenti dell’applicazione)

**Text Assumption:** Il vantaggio dell’utente in data for help sta nella possibilità di poter vedere dati storici relativi ai propri parametri vitali

**Altre ipotesi che potrebbe essere necessario inserire**

* Le persone sono identificate tramite codice fiscale

Parametri vitali rilevabili

* Saturazione del sangue
* Frequenza cardiaco
* Pressione
* Temperatura corporea

Altri parametri rilevabili

* Caduta

**N: B.** Potremmo avere alcuni problemi nella gestione dei casi di falsi positivi

**Schema fisico (DESIGN REQUIREMENT)**

SENSORE ||||| APPLICAZIONE |||||| RETE ||||| SERVER |||| DATABASE

||||| HARDWARE SU CUI GIRA L’APP ||||||

( Nei casi in cui il device ( qui sopra definito come *hardware su cui gira l’applicazione*) non sia uno smartwatch ( che integra il sensore al proprio interno e ha uan scheda sim per comunicare sulla rete), bisogna includere nello schema fisico anche la rete tra sensore ( che si troverà quindi su di un qualsiasi dispositivo indossabile) e il device ( che sarà quindi con ogni probabilità lo smartphone). Tale connessione sarà di tipo Bluetooth (??).

Abbiamo stimato i dati da trasmettere tra sensore e device in circa 1,2 mb al giorno: la ritengo una cifra accettabile per una connessione bluetooth, che mi sembra la più indicata per trasmettere dati a così corto raggio.

**Approccio prudenziale**

Non possiamo ignorare eventuali malfunzionamenti rilevati nel nostro servizio, ma non possiamo nemmeno mandare un’ambulanza a casa delle persone soltanto perché si è spento un componente.

(trade off) -> In caso di malfunzionamenti, chiamiamo un numero di emergenza.

[ENGLISH ONLY]

**GOALS**

In this part we expose functional requirements and domain assumptions associated to the goals of our application that we want to satisfy, distinguing the two actors.

(legenda: (R): functional requirement, (D) domain assumption, (NR) non functional requirement

**Clients (*a bit too generic: I propose, instead of this,*** *patients)***:**

1. Provide a form of unique identification (registration/login) of all people using the application
   1. If the user does not insert username or password the application does not let the user access itself(R)
   2. The user has correctly downloaded the application from the online store on his device (smartwatch, smartphone, etc) [i.e. he is not using a crack version] (D)
   3. If the user has not a valid username or password (i.e. it’s the first access), first the application shows him a registration form (R)
   4. If the user does not fill the registration form with his fiscal code and all other personal data, the application refuses the registration (R)
   5. If the username provided in the registration form is already in use, the application refuses the registration (R)
2. Protect the privacy of the patient (*Issue: it could be a bit too generic: what does “protect” in practice mean?? I propose, instead of this,* ***prevent third parties from associating a single user to his data without his permission****, but it could be a bit too specific… che ne pensi zia?****)***
   1. If a third part asks for data of a single user, data are shown if and only if he concedes his permission (R)
   2. if a third part asks for data that involves less than 1000 people, the application refuses (R)
   3. if a third part asks for data that involves more than 1000 people, the application anonymizes data before sending (R)
3. Whenever a user is in danger of life, the application is working and there is internet connection, an ambulance is alerted, guaranteeing a reaction time of less than 5 seconds from the time parameters are below the threshold *(Issue:* ***coherence****; with all this if... if... we should put in the list only the requirements needed for this goal, and avoid the requirements used to handle the application breaks down)*
   1. If the application does not read properly input data every 500 Ms, it informs the user that the sensor is not working as expected, suggests contacting the customer service as soon as possible and asks for confirmation of good health status. (R)
   2. If the user does not respond to confirmation, the application informs the server of that fact (R) [questo va tolto perchè è un machine phenomena]
   3. If input data show a severe disease or the user communicates that an emergency is occurring, the application informs the server that an emergency is occurring, and the server handles this issue (R) [I propose, to avoid including machine phenomena*, If input data show a severe disease or the user communicates that an emergency is occurring,* **the server contact the ambulance that has the shortest path to the location]**
      1. The application knows the threshold for each type of patient (R)
      2. The application compares input data with the threshold (R) [propongo di cancellarle entrambi questi due proprio perchè sono una machine phenomena, cnf. Con le riflessioni in fondo]
   4. The application processes, compares data and send a message to the server in less than 3,5 sec (provided that only three consecutive data below the threshold are a clear signal of severe disease and data are sent every 500 Ms) [non-functional requirement]
   5. GPS always works properly to indicate the patient position (D)
4. If something is not working as expected (the sensor, the application, the network), the patient’s family is alerted within an hour [APPROCCIO PRUDENZIALE]
   1. If the application does not send data for back up purpose every hour, the server sends a message to the emergency number, provided through the registration form (R)
   2. The emergency number is correct (D)
   3. If the user does not fill the registration form with his fiscal code and all other personal data, the application refuses the registration (R) (requirement taken from goal 1)
   4. If the application informs the server that the sensor is not working as expected, the server sends a message t the emergency number (R) [I propose, to avoid including machine phenomena*,* If the application *notices* that the sensor is not working as expected, the server sends a message to the emergency number (R)]
5. Allow the user to see his clinical history (Issue*: a bit too generic; I propose instead of this,* ***to see, under request, reports on his vital parameters that include everything)****;*

[QUESTO VA RIPENSATO PER EVITARE I MACHINE PHENOMENA]

* 1. The application stores data read from the input after comparison with standard (R)
  2. The application sends data to the server every hour (R)
  3. the server stores data received in a database for every registered user (R)
  4. if the user asks for a report clicking on a button, the application submit the request to the server (R)
  5. the user has made sure that there is internet connection when the request is submitted (D)
  6. when requested, the server sends data stored to the application (R)
  7. when receives responses, the application shows data (R)

**[ I PROPOSE TO DELETE THE FOLLOWING GOAL]**

1. Provide the user notifications and advice about his health status
   1. ……

[N: B. Ci eravamo chiesti se dovessimo rendere possibile alle third parties di specificare preferenze o dovessero fare ogni volta una nuova query per chiedere gli stessi dati aggiornati. Rileggendo la traccia ho trovato: “*also, it allows the third party to subscribe to new data and receive them as soon as they are produced”.]*

**Third parties**

1. Allow third parties to access, under request, to anonymized data, or to specific individuals’ data under their permission.
2. Provide a form of unique identification (registration/login)
   1. If the user does not provide his public information (i.e. name
3. Allow third parties to specify constrains in their researches
4. Allow third parties to subscribe to new data

NON FUNCTIONAL REQUIREMENTS

* Availabiliy???

RIFLESSIONI PERSONALI

1. Ho cercato di mettere il più possibile i goal in termini di shared phenomena, cioè di fenomeni che possono essere controllati dal sofware (altrimenti che goals sarebbero se non potessero essere garantiti interamente dall’applicazione?), poi ho letto le slide che fanno:

*Goals are prescriptive assertions formulated in terms of world phenomena (****not necessarily shared)***

**EH MA SE NON SONO SHARED COME FACCIO??**

1. *Requirements are prescriptive assertions formulated in terms of shared phenomena:* forse alle volte sono entrato troppo nel dettaglio e ho espresso machine phenomena? Ho ripensato alcuni requirements che mi sembravano solo machine phenomena, li ho segnati con dei commenti, la cosa importante secondo me è essere coerenti (anche perché quello che succede tra server e applicazione è shared o soltanto machine?)