APPUNTI PER STILARE IL RASD

During the developing of this project we followed an evolutionary and agile model (to forecast, for example, a service to track movement in general.

**DEFINITIONS**

* “Data about health status”: when in the following parts we state “data”, “health status” and “data about health status”, we are meaning the following values
  + Blood saturation: it’s an indicator of the status of lungs and of respiratory tract in general, of the cardio-vascular system (e.g. to detect suffocation)
  + Hearth rate: it’s an indicator of hearth diseases (to detect arrhythmias)
  + Blood pression: it hardly ever helps to detect an emergency, but it’s useful for third parties and statistics (blood pressure out of range can indicate/cause a huge number of chronical diseases)
  + Body temperature: it’s an indicator of fever
  + Patient’s falling: if the patient has suddenly fallen there could be various causes and effects that, though all the other values are not able to detect them, put in serious risk the patient’s life.
* Data Anonymization (when possible): deleting the fiscal code associated to every data tuple obtained by the query. If the total number of fiscal codes is less than 1000, refusing to return data.

[è NECESSARIO DARE QUESTA DEFINIZIONE???]

**ABBREVIATIONS**

* [Gn]: n-goal
* [Dn]: n-domain assumption
* [R-n]: n-functional requirement

**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, an hospital)
* The ambulance man in charge?????? [Dobbiamo riflettere di più su questa cosa secondo me]

**PHENOMENA**

**WORLD PHENOMENA**

**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) [definition from slides]**

**MACHINE PHENOMENA**

**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
* Abbiamo deciso di considerare anche le situazioni in cui cade la connessione internet etc (che invece potevamo evitarci facendo delle ipotesi) [ho scritto questo a partire da una domanda che ho trovato su beep]

Parametri vitali rilevabili [ho aggiunto questa cosa nelle definizioni]

* 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

1. GOAL G3

, *(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 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 machine contacts 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]
  2. 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]
  3. GPS always works properly indicating the patient’s position (D)

1. [APPROCCIO PRUDENZIALE]
2. 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 communicated by the sensor)****;*

[Scrivendolo in grassetto siamo furbi perché teniamo in considerazione il fatto che i dati potrebbero essere incompleti e non riguardare alcuni lassi di tempi

[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)

1. **[come ho ripensato il goal 5]** Allow the user to see, under request, reports on his vital parameters that include everything communicated by the sensor
   1. the user has made sure that there is internet connection when the request is submitted (D)
   2. If the user asks for a report, the machine shows data stored in the database (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. Provide a form of unique identification (registration/login)
   1. If the third party does not insert username and password the machine does not let the user access ant functionality [login] (R)
   2. If the third party declares that it has not a valid username or password (i.e. it’s the first access), first the machine shows him a registration form (R)
   3. If the third party does not fill the registration form with his official e-mail and all other public data, the machine refuses the registration (R)
   4. If the username provided in the registration form is already in use, the application refuses the registration (R)

[DEVO ANCORA RIGUARDARE QUEST’ULTIMA PARTE CHE POTREBBE ESSERE IMPRECISA]

1. Allow third parties to access to data if and only if they could be anonymized, or to specific individuals’ data under their permission.
   1. the third party has made sure that there is internet connection when the request is submitted (D)
   2. If the third party asks for data and the number of people involved is greater than 1000, the machine shows data stored in the database after having anonymized them (R)
   3. if the third part asks for data and the number of people involved is less than 1000, the machine refuses (R)
   4. if and only if the third party asks for data of a specific individual and he acceps, the machine shows data stored in the database (R)

1. Allow third parties to specify constrains in their researches
2. Allow third parties to subscribe to new data
   1. The user
   2. When more than 15% of data in a “favourite query” specified by the user changes, the machine sends a notification to the user via e-mail which includes new data(R)

NON FUNCTIONAL REQUIREMENTS

* **Availability and reliability**
  + Concerning the server: We need a server available 24/7 to handle emergency messages as fast as possible because, among other reasons, there is no way for the app the handle an emergency without the server;
  + Concerning the app: we also need availability for the app, but not (with so much relevance) reliability , due to the fact that it takes time to detect a disease ( the sensor does not send data in real time ecc..), in which there could be an app break down with no significative consequences for the service ( provided that recovery time is under 500 ms).
* **Security:** thinking of a market such as the American one, where health care is subject to negotiation between patients and companies, security of all sensitive information which could advantage malicious companies, is a very important concern for our application. Due to this reason, data encryption should be implemented in communications between the app and the server and the web apps.
* **Performance:** of course, we need a fast reaction to emergency. Concretely, we state that the machine must guarantee a reaction time of less than 5 seconds from the time the parameters are below the thresholds. In addiction to this, 4G connection is required to ensure immediate communication with the server.
  + Concerning the server, we opt for an event-based architecture
  + [QUI NON STO ESAGERANDO??????….] SLIDE 5D
  + [FORSE CI TORNERà UTILE PER IL DD]
* **Accuracy:** this is the non-functional requirement we consider the most relevant. While assuming (here and in all other parts of RASD, cfr, *domain assumptions*) that GPS works properly, our biggest concern is about the sensor which detects values: there is a concrete risk of frequent false positives, caused by some little offsets. Such cases are particularly annoying because we can’t ignore the alarms but at the same time, they make us waste time, money and resources, threatening the correct working of our machine. We pointed out that there is a trade-off between this issue (which would request to wait for a lot of consecutive signals of emergency to be sure) and the issue of performance (which would request to react immediately). We decided to wait for 1,5 sec before alerting, which means, values below thresholds for three times in a row.

**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?)
2. Alle volte ho paura di stare entrando troppo nell’implementazione, che invece andrebbe evitata in un RASD.

**OPEN QUESTIONS**

* What happens if the number of people satisfying the request is bigger than 1000 when the request is accepted, but then decreases under this threshold (due to death, aging etc.)?

[qua dobbiamo metterci d’accordo]

*La mia proposta: se il numero di persone che corrisponde a quella query scende sotto i mille, i dati non vengono più resi disponibili alla third party (lo trovo più semplice da modellare con i goals)*

* How to distinguish between third parties in order to provide unique identification?

*La mia proposta: indirizzo e-mail ufficiale della società.*

(per le persone usiamo il fiscal code)

* + Che cosa dobbiamo modellizzare con Alloy? Brain storming: data, third party, user, application, queries, requests…..

**HARDWARE COMPONENTS**

**[l’ho raffinato in hardware limitations, che comparirà nel RASD]**

* A server running databases
* A smartwatch with internet connection and GPS and memory with little capability

OR

OR

* A physical device with a sensor and Bluetooth connection AND a smartphone with internet connection and GPS and a memory with little capability
* A physical device with a sensor and Bluetooth connection AND a smartphone with internet connection and GPS and a memory with little capability

**HARDWARE LIMITATIONS**

* iOS or Android smartphone with 2G/3G/4G connection and Bluetooth connection
* 4G connection
* GPS connection
* Wearable device with Bluetooth connection

In alternative

* iOS or Android smartwatch
* 4G connection
* GPS connection

For visualizing data,

* Modern browser able to render graphs and statistical models