POLITECNICO
MILANO 1863

TrackMe project Giorgio Cozza and
Barbara Ferretti

Data4Help and AutomatedSOS

Design Document

Deliverable:	DD
Title:	Data4Help and AutomatedSOS - Design Document
Authors:	Giorgio Cozza, Barbara Ferretti
Version:	1.0
Date:	10-December-2018
Download page:	https://github.com/GiorgioCozza/CozzaFerretti.git
Copyright:	

Contents

List of Figures

List of Tables

1 Introduction

1.1 Purpose

As explained in the previous artifact, Data4Help and AutomatedSOS, since handling the same set of data types, but concerning with different purposes, are physically separated applications. The first one represents a privacy-guaranteed service to manage third party requests of access to user-provided health data, the second is a service conceived to ensure first assistance whenever dangerous conditions for registered users are detected. The discussion at this point should be focused specifically on the architecture behind these two services, analyzing the way both the applications will collect health data from the monitoring devices, how TrackMe servers will manage such data and more. To this end, since TrackMe wants to not depend strictly on external cloud infrastructures for commercial and data-security reasons, it decided to host completely its own server farm. For the moment the system will be centralized, but easily **distributable** for future needs, according to the growth factor of the customer number. This, in order to form a private cloud infrastructure built on top of a distributed database system. Unfortunately, the deadline prevents a detailed analysis of this solution for further versions of the project.
This Design Document, however, will provide an overall definition of the main system components and the relative interactions, an in-depth discussion about the chosen architectural patterns and the plan for integration, verification and validation steps. The audience for this artifact is tighter than the one of the RASD due to the more technical language, in particular, it is oriented to all the people involved in engineering the software-to-be. In any case, in order to facilitate the comprehension of this document a previous RASD consultation is strongly suggested.

1.2 Scope

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

1.3.2 Acronyms

ASOS: AutomatedSOS
D4H: Data4Help
API: Application Programming Interface
RASD: Requirement Analysis Specification Document
HW: Hardware

1.3.3 Abbreviations

[Gn]: n-th Goal
[Rn]: n-th Requirement

1.4 Revision History

Up to now, there are no revision of this document as this is the first release.

1.5 Reference Documents

1.6 Document Structure

1.6.1 Chapter 1

In the first chapter is given a general presentation of the aim of this document. Moreover are given other basic information in order to read easily the entire work, like the dictionary for example.

1.6.2 Chapter 2

1.6.3 Chapter 3

1.6.4 Chapter 4

1.6.5 Chapter 5

2 Architectural Design

2.1 Overview

Both Data4Help ad AutomatedSOS are complex services due to the large set of technologies involved in their functionalities. To have a full comprehension of the overall architecture, it is necessary to abstract the entire system, starting the analysis from the customer perspective. Here, in fact, the main challenge is related to device management and data representation, since a large variety of sensors (supported by the service), can be connected and send data simultaneously. Each health indicator sample provided to the app should be processed at first in order to match a common data structure, then sent to the server. Since a D4H user is automatically an ASOS user at the first run of the relative application, it may happen that the same will use both the applications in tandem. This leads to a problem of data redundancy, since the server could receive two samples from the same monitoring device. The problem should easily solved by the server, if it was not computational expensive for growing workloads. This part of the business logic could be easily managed by the apps through a mechanism of message exchanging to synchronize the sensor accesses and preventing the . To this purpose, from an high-level perspective it is useful to rely on the following picture:

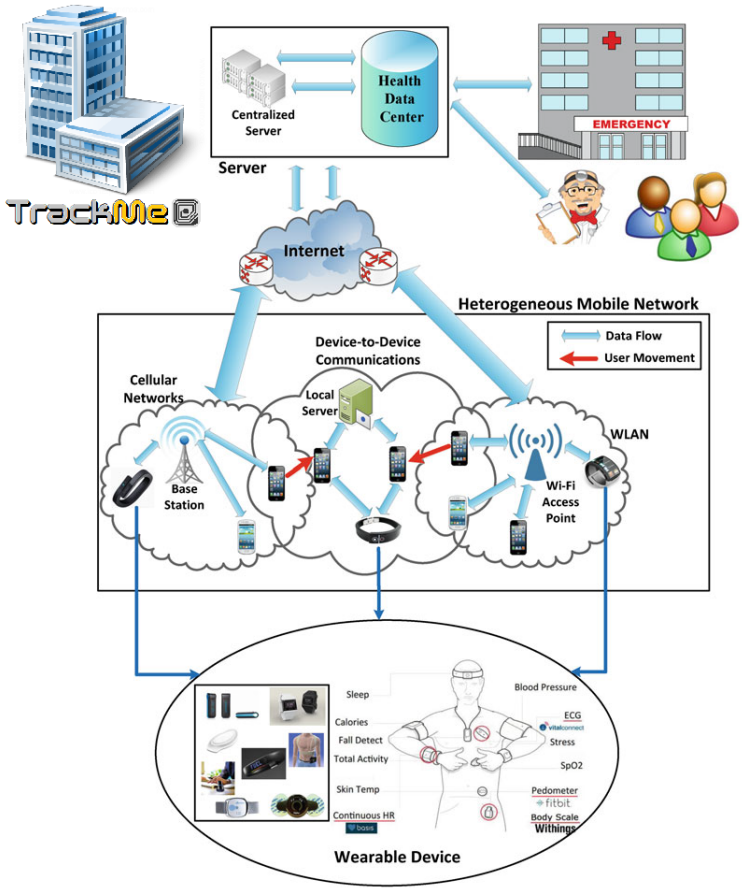

Figure 1

2.2 Component View

T

2.2.1 Client Side

Starting from the interface to the external sensors, the main problem related to the app separation is due to the fact that if the applications are running at a certain moment both of them could theoretically access the same monitoring device, providing to the server an identical sample. As anticipated in the Overview part, both the applications at clien side implement a personal hardware interface to the connected monitoring devices. This decision seems to introduce redundancy in code since a simple third service running in background could manage the connected devices and dispatch data between the two applications, unfortunately communication among running processes follows strict rules in many mobile OSs, someone only allows message exchanging. This makes more difficult to implement a mechanism of access to the monitoring sensors involving at most three different processes. Furthermore, relying on an HW manager service exposes both the apps to possible internal DOS attacks, since another malicious app could flood of messages the background service preventing the main application to allocate accesses to connected sensors.

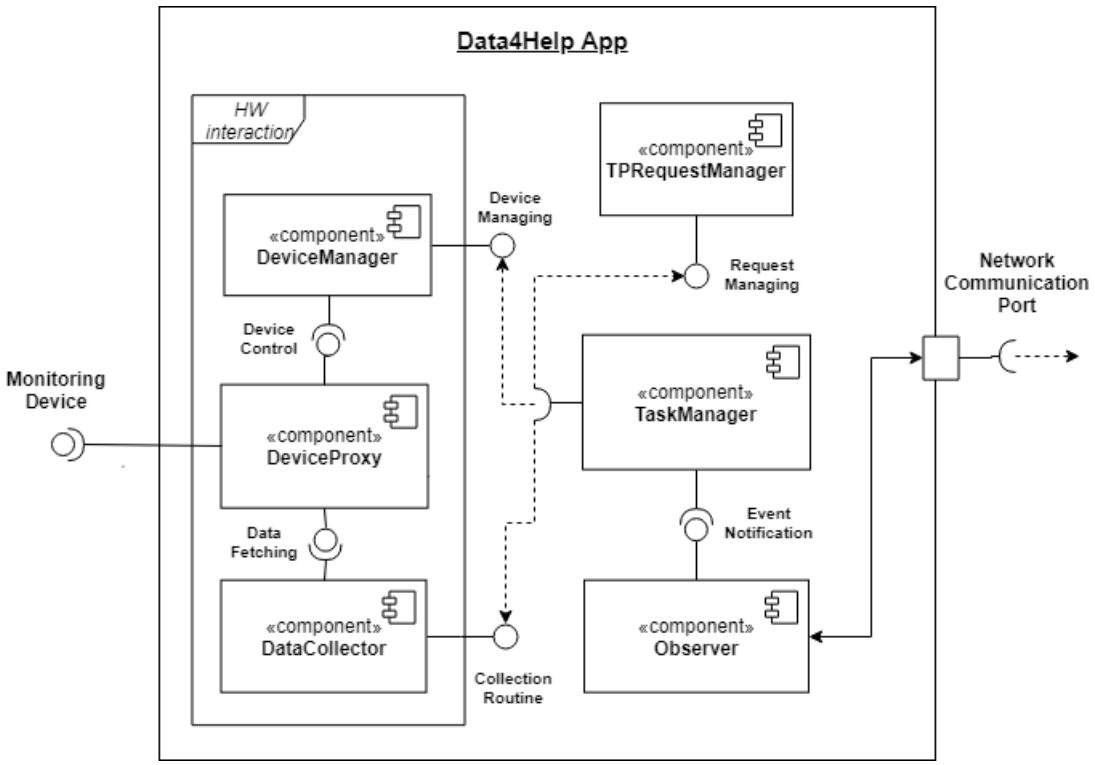

Figure 2

It has been decided to create a communication interface between D4H and ASOS, for synchronizing data collection and preventing both the applications to access the same information. This task should be protected in order to avoid other processes to send false messages to one of them. A security toolkit provides encryption mechanism to this end.

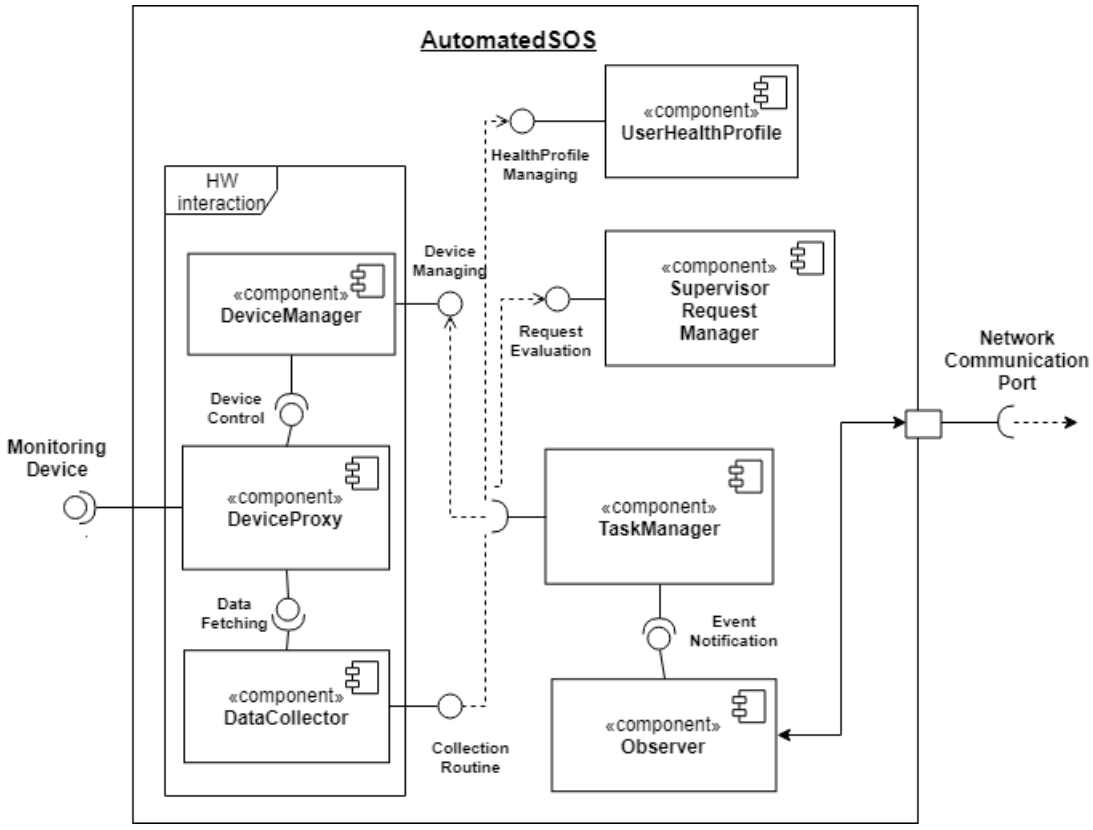

Figure 3

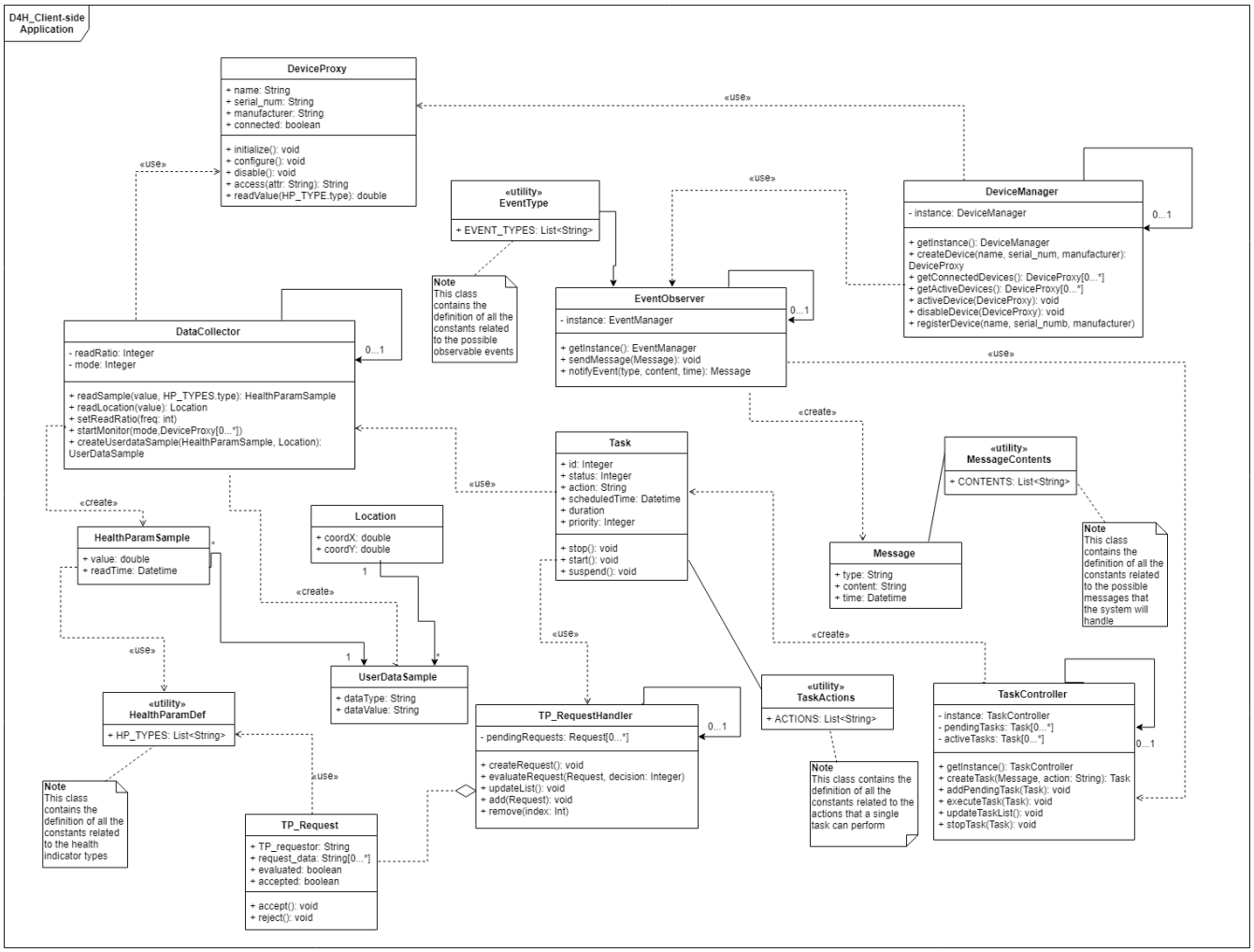

Figure 4

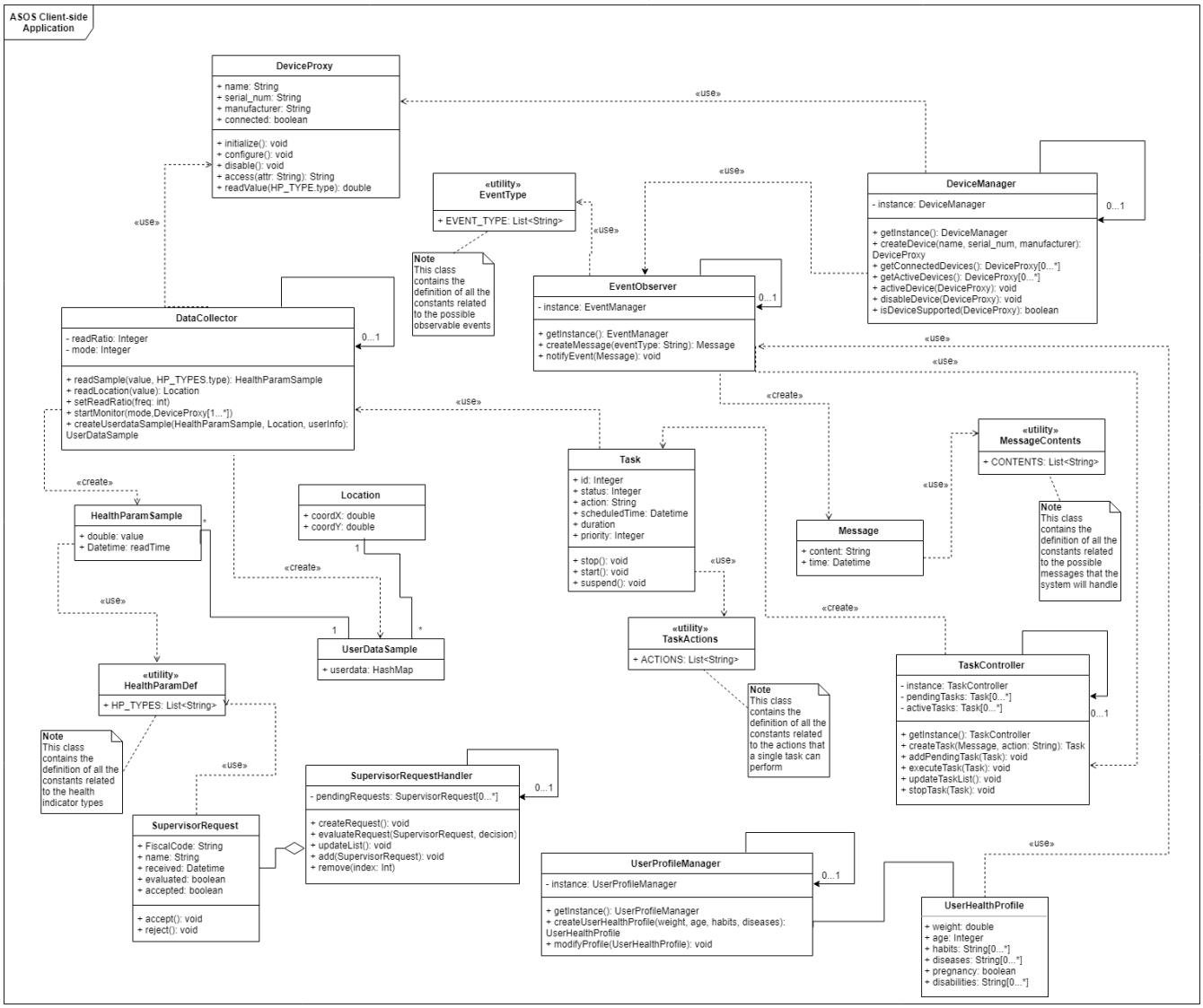

Figure 5

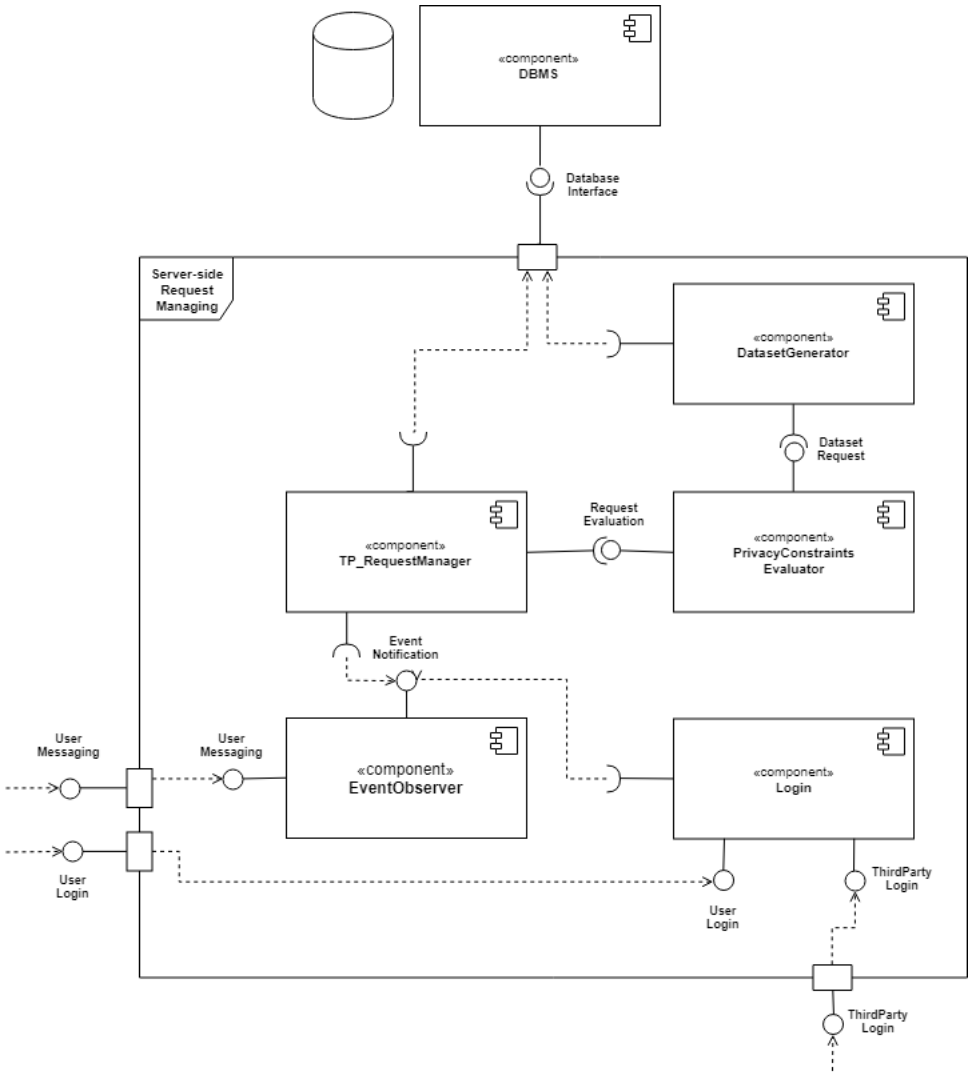

Figure 6

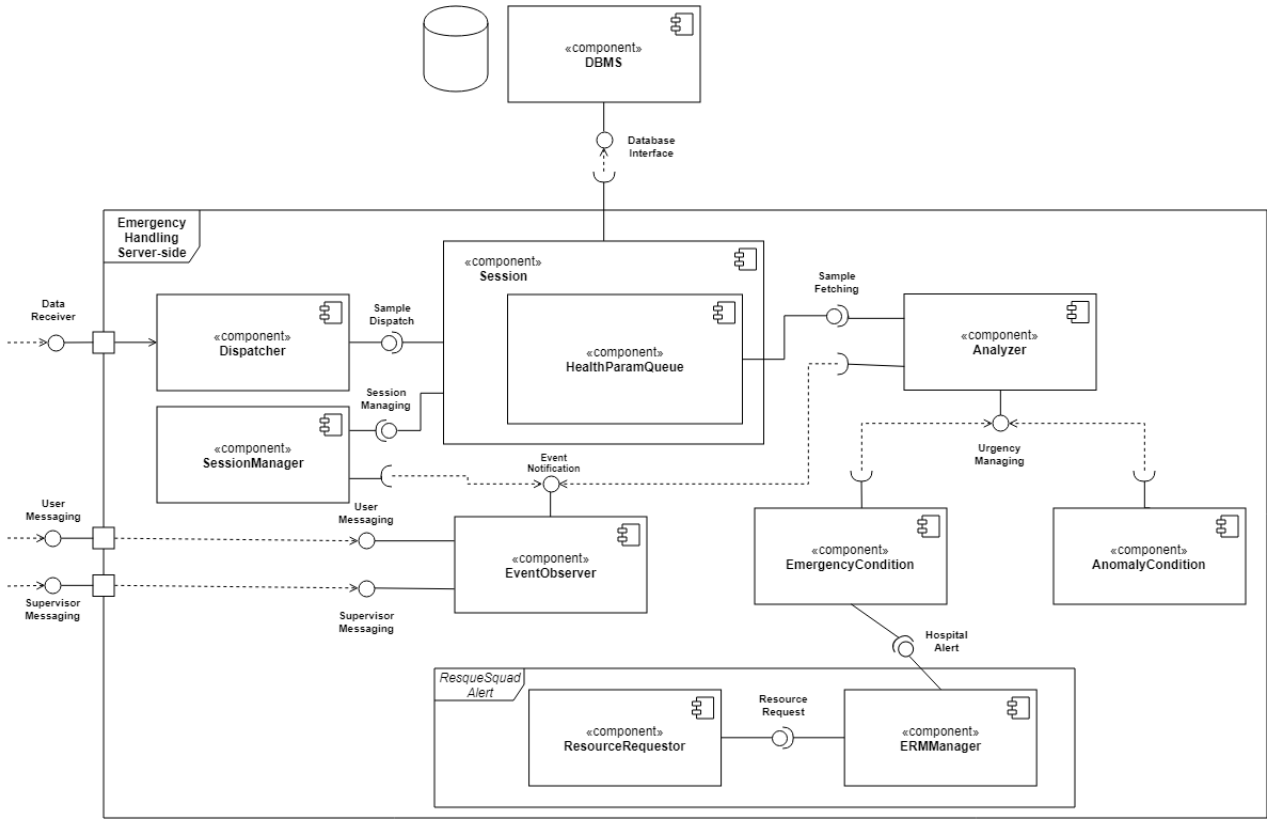

Figure 7

2.3 Deployment View

2.4 Runtime View

2.5 Component Interfaces

2.6 Selected Architectural Styles and Patterns

2.7 Other Design Decisions

3 User Interface Design

3.1 Mock-ups

For what concern the software-to-be mockups, is possible to find them in *Section 3.1* in the *Requirements Analysis and Specification Document*.
As for reminder for the next section, for the Data4Help part, the user section is intended to be a mobile application and the Third Party section a website. On the other hand for AutomatedSOS part the idea is to have only a mobile application.

3.2 UX Diagrams

The following diagrams are intended to give a general idea about the interactions between the various screens and the actions that are permitted. For the sake of clarity here are explained the notation of the interfaces:

- ***«Screen»:*** the screen is what the final user is going to see.
- ***«Input Form»:*** everything that the final user is going to interact with (empty space to fill with text, checkboxes, on/off buttons...).
- ***«Screen Compartment»:*** part of the screen that contains plain text.

3.2.1 Data4Help

User

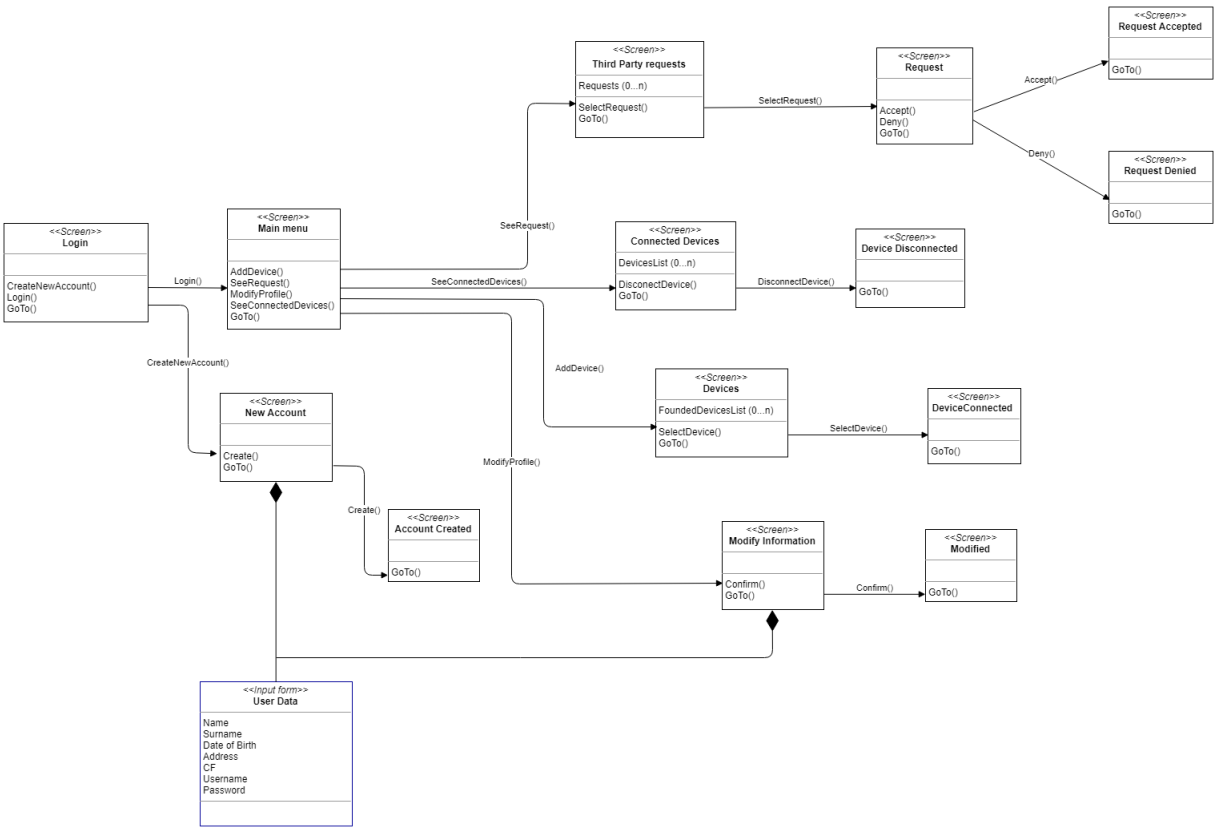

Figure 8: Data4Help User UX diagram

Third Party

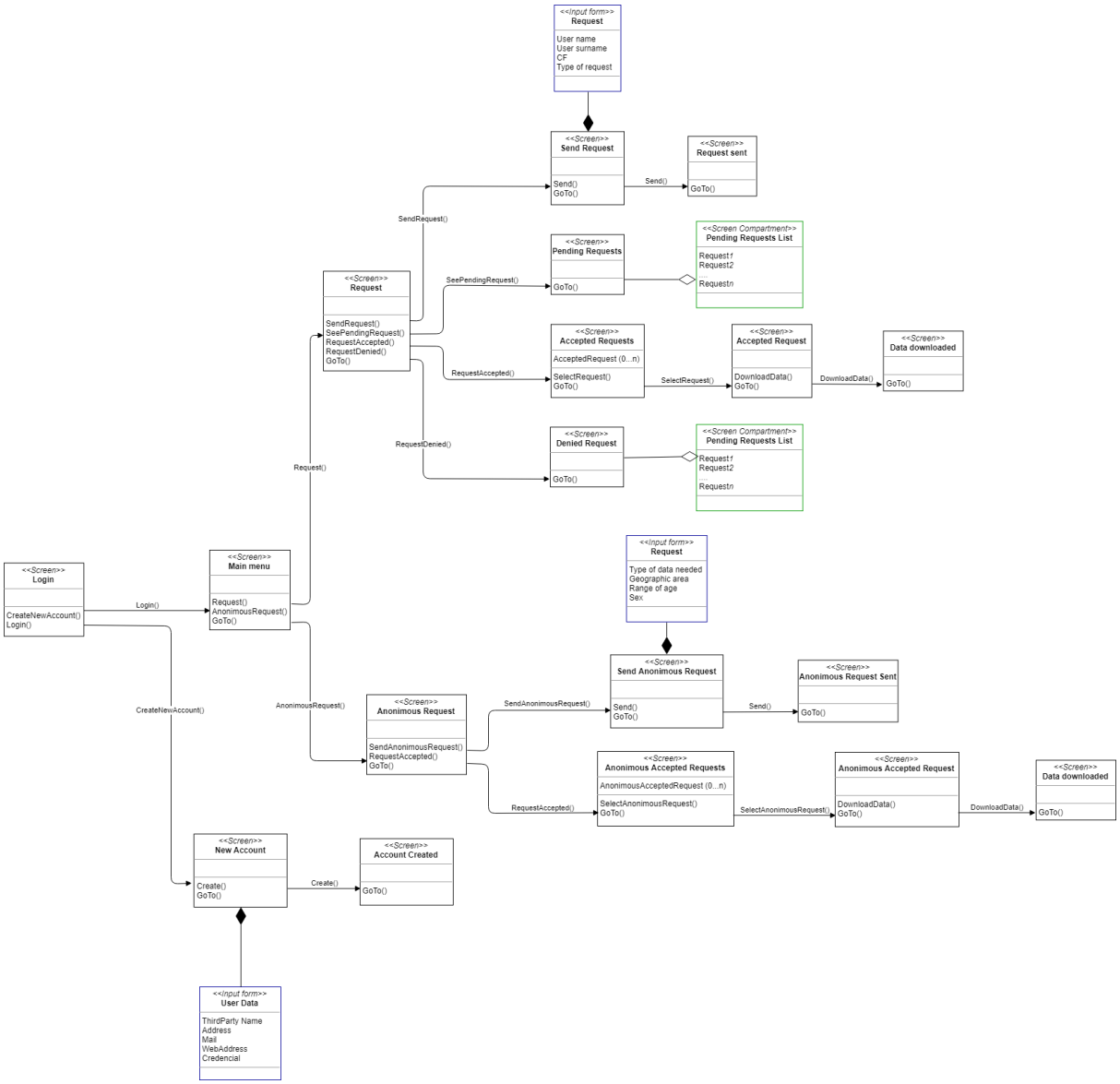

Figure 9: Data4Help Third Party UX diagram

3.2.2 AutomatedSOS

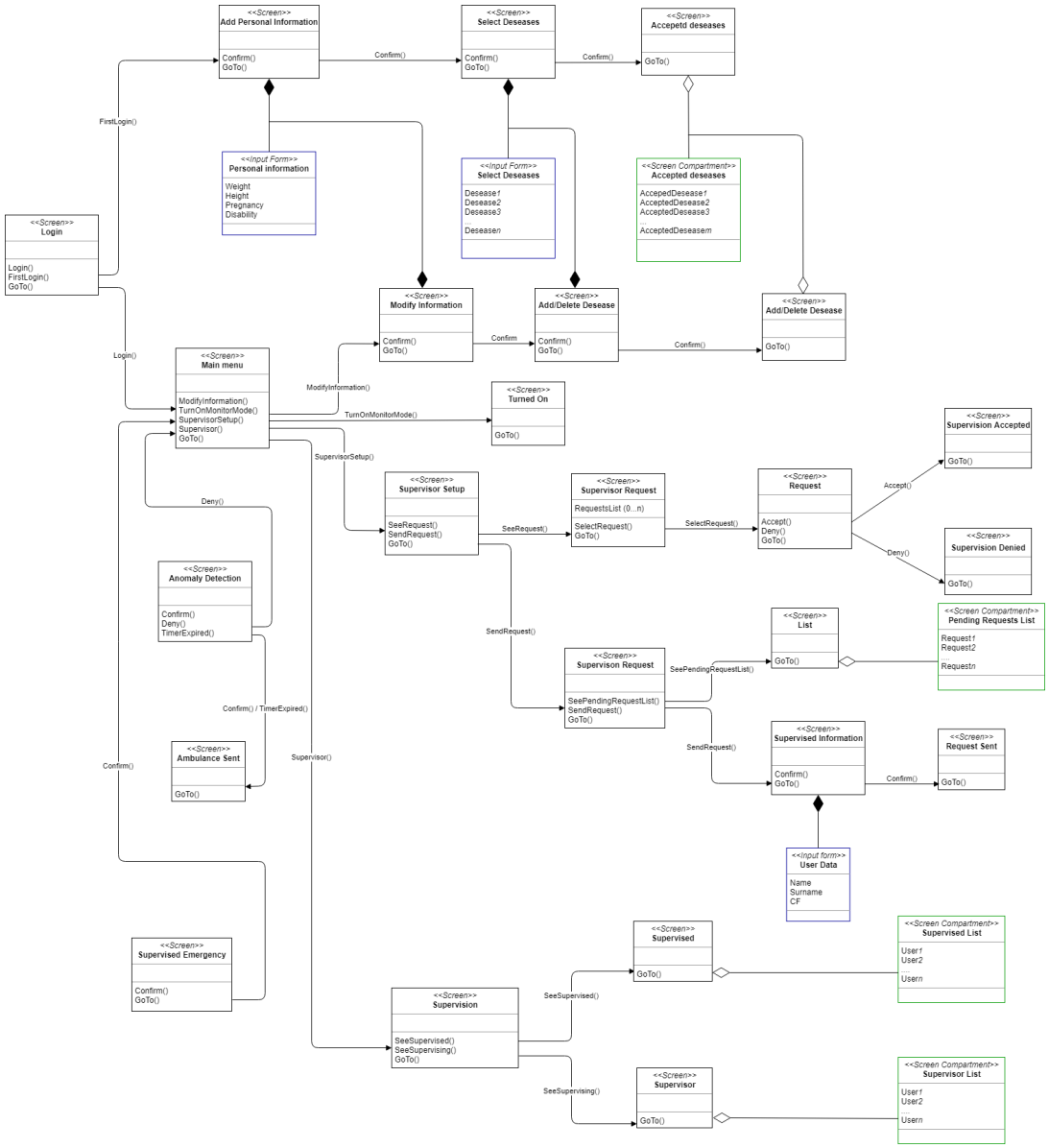

Figure 10: AutomatedSOS user UX diagram

4 Requirements Traceability

4.1 Data4Help

[G1]: A user can register personal health monitoring devices in the system

- [R1]: The system must allow a registered user to associate a sensor device to the service.

  –

[G2]: TrackMe acquires periodically health parameters specifically related to a user

- [R2]: Each user is uniquely identified by the system.

  –

- [R3]: The system can acquire health data by specific sensors connected to the user's main device.

  –

- [R4]: The system must provide the possibility to users to insert specific health information that cannot be acquired through other monitoring devices.

  –

[G3]: Third parties can access health data of specific users, if expressively authorized by the them

- [R5]: The system must be able to identify and certificate the reliability of each organization that wants to request user data.

  –

- [R6]: Each registered organization that wants to access health data of specific users must be able to formulate a request providing information related to the purpose of the request.

  –

- [R7]: The system must notify each user of a third party request as soon as it is formulated, and allow him/her to accept or reject the request.

  –

- [R8]: For each third party request the system should provide to the user information related to the requestor and the purpose of the request.

  –

- [R9]: Once a third party request is accepted by the user, the third party must have full access to the entire collection of data of the user.

  –

[G4]: Third-parties can request anonymous information about groups of users

- [R10]: Third parties must be able to request health data of groups of anonymous users, according to several criteria without being expressively authorized.

  –

- [R11]: The system must prevent third parties to trace back to specific user information through anonymous requests.

  –

4.2 AutomatedSOS

[G5]: The user has the possibility to specify the diseases he/she has, so the system can evaluate which health parameters to monitor

- [R12]: The system must give the possibility to the user to specify which diseases he/she has.

  –

- [R13]: The system must prevent the possibility to use the service, if the minimum required sensors are missing.

  –

[G6]: An ambulance may be sent within 5 seconds, if an emergency or anomaly condition is detected

- [R14]: The system must allow the user to turn on and off the service each time he/she wants.

  –

- [R15]: The system must stop monitoring when minimum required sensors are missing.

  –

- [R16]: The system must be able to distinguish and detect emergency or anomaly conditions occurring to a specific user.

  –

- [R17]: When an anomaly condition is detected, the system must send a notification to the user, asking if it is an emergency condition. If the user does not answer the notification within 30 seconds, the anomaly must become an emergency

  –

- [R18]: When an emergency condition is detected, the nearest ambulance must be alerted, providing all the information about the situation.

  –

[G7]: A user designated as Supervisor of another one, is notified of both emergency and anomaly events occurring to the supervised user

- [R19]: A user must have the possibility to request to become Supervisor of another one.

  –

- [R20]: Each supervised user must be able to accept or reject the possibility to have a Supervisor.

  –

- [R21]: The Supervisor must be notified by the system of all emergency and anomaly conditions occurring to the supervised user.

  –

5 Implementation, Integration and Test Plan

6 Effort Spent

Here is the effort spent by each group member in working at this document.

Barbara Ferretti	Hours
User Interfaces	5

Giorgio Cozza	Hours

References

