Politecnico di Milano

DD

TrackMe Project

2018

Version n°1

Software engineering 2

ABSTRACT:

The design document (DD) provides a functional description of the system

Emma Bortone

Mohamed Gawish

Table of Contents

I. Introduction

I.A. Purpose

The purpose of this document is to define the design document (DD) of the services Data4Help and AutomatedSOS offered by the company TrackMe. This document contains the architecture of the system to be developed and its application domain. It can be used as a baseline for software evaluation and for charge control.

I.A.1. Data4Help

Data4Help is a software-based service allowing third parties to monitor the location and health status of individuals. Data4Help supports the registration of individuals and of third parties.

- Individuals who register to Data4Help allow the company TrackMe to acquire their data.
- Third parties who register to Data4Help can access the data acquired by TrackMe by the mean of requests. They can request to :
    - Access to the data of a specific individual by providing a unique identifier. The request is then transferred to the individual who decide to accept or refuse it.
    - Access to anonymized data of groups of individuals. The request is accepted if the number of individuals satisfying it is higher than 1000.

  When a request is accepted, third parties must be able to access the previously saved data and to subscribe to the request. By subscribing to a request, third parties will receive the new data corresponding to the request as soon as the data is available and without the need to renew the request process each time. Third parties can make an unlimited number of requests and subscriptions

The goals of the Data4Help service are :

- [G 1] : Third parties must be able to request to access to the data of specific individuals or to anonymized groups of individuals.
- [G 2] : At any time, third parties should never have access to data of specific individuals without their agreement.
- [G 3] : Third parties must have the possibility to subscribe to new data if their request is accepted.
- [G 4] : Individuals must be able to consult their data and accept/refuse requests

I.A.2. AutomatedSOS

AutomatedSOS is a service build on top of Data4Help, therefore AutomatedSOS must verify all the requirements of the service Data4Help. In addition, AutomatedSOS offers the possibility to monitor the health of the subscribed individuals and to automatically send an ambulance to the location of the individuals if their health parameter are below certain thresholds.

All the goals of Data4Help are also goals of Automated SOS, but AutomatedSOS have the additional goal :

- [G 5] : An SOS is sent to an ambulance service, specifying the position of the individual, with a reaction time below 5 seconds from the time the individual's health parameters are below threshold.

I.B. Hypotheses on the assignment

To solve the ambiguity and incompleteness of the project, we made some assumptions on the services Data4Help and AutomatedSOS. Those assumptions are listed in the two following paragraphs.

I.B.1. Data4Help

- The data is either collected by a smartphone or by the mean of a smartwatch synchronized to a smartphone application. We made this choice because most of the smartwatches currently on the market are aimed to be linked to a smartphone through an application and we want our software to run on as many platforms as possible (design for portability)
- The subscription to a request on a group of individuals is automatically cancelled if the number of individuals whose data satisfy the request goes bellow 1000 at some point.
- The subscription to a request on a specific individual is automatically cancelled if the individual cancels his agreement.
- The individual can see the data collected by the service Data4Help. Indeed, the individual must earn something in exchange for the data he agrees to share. In this case, the individual gains the ability to control his health data.
- Data4Help must respect the General Data Protection Regulation (GDPR)
- Third parties can be compagnies, organizations or persons who need to acquire data (for example students or independent data scientists).
- The Data4Help service take the form of :
    - **A website and a Web API for data request.**
      We made this choice because using a website is the most convenient solution for a company or an organization who need to punctually acquire data and a web API is the most convenient solution for third parties who need to acquire data regularly (each hour for example)
    - **A smartphone application for data acquisition.**
      We made the choice to consider that the individuals have some piece of Data4Help installed on their personal device which allow Data4Help to acquire their health data. The motivation for the individual to install Data4Help on his smartphone will be to observe his own health data. Other solution would be to gathers data from some other service/system but this would mean to buy the data and the delay might be more important.

I.B.2. AutomatedSOS

- The AutomatedSOS service is not an independent application. The individuals who want to subscribe to AutomatedSOS will first need to download the application Data4Help on their smartphone to register to the service. We made this choice because the AutomatedSOS service is only an extension of the Data4Help service so the requirements of the Data4Help service are also requirements of the AutomatedSOS service.
- AutomatedSOS notifies the ambulance service that an ambulance needs to be sent to a certain location by sending a SMS.

- Individuals who want to subscribe to AutomatedSOS are required to have a smartwatch and to link it to their Data4Help account. Indeed, a smartphone cannot acquire the pulse rate of the individual and this data is significant for detecting emergencies.
- The thresholds are automatically defined by AutomatedSOS depending on the individual's characteristics. We choose to prevent individuals to set themselves their thresholds to avoid unnecessary SOS or an urgent situation not being detected.
- Individuals also have the possibility to download an application on their smartwatch, in addition to the app on their smartphone, although this is not mandatory for the data acquisition. This will allow the individual to have an overview of their health data by simply looking at their watch. We made this choice because the individuals who subscribe to AutomatedSOS are elderly people who might not be comfortable with smartphones.

I.C. Definitions, Acronyms and abbreviations

Third party: company or organization or person who need to access data of individuals

Individual: person willing to share his data with a third party and who want to monitor his health.

Smartwatch: device aimed to be worn on the wrist. A smartwatch can collect data, can be linked to a smartphone application and can support wireless technologies like Bluetooth, Wi-Fi, and GPS.

Subscription: an arrangement for automatically receiving data corresponding to a specific accepted request as soon as the data is available and without the need to renew the request process each time.

Request: the act, for a third party, of asking for data. A request can be individual or anonymized.

Individual request: A request that concerns a unique person known by his fiscal code.

Anonymized request: A request that concerns a group of individuals who fit some criteria. The result of this type of request is anonymized.

Anonymize: to remove any information that shows which particular person a data relates to.

SOS: SMS sent by Data4Help to an ambulance service asking to send an ambulance to a specific location. An SOS is sent when the health parameters of a monitored individual go below threshold.

I.D. Revision history

Version	*Date*	*Description*
V1.1	09/12/2018	First version finished

I.E. Reference documents

[1] SOMMERVILE, Iam. Software engineering 9. International edition.
[2] Assignment Mandatory Project: goal schedule, and rules.
[3] Les numériques, COMPARATIF / Quelle montre connectée choisir ?
https://www.lesnumeriques.com/montre-connectee/comparatif-montres-connectees-a1781.html

I.F. Document structure

This document contains the design description of two services: the Data4Help service and the AutomatedSOS service. This documents contains three main sections :

- Architectural design: This section contains a functional description of the problem provided by component views, deployment views and runtime views. The last part of this section contains a description of the selected architectural styles and patterns.
- Requirement traceability: This section shows how the requirements defined in the RASD map to the design elements.
- Implementation and test plan: This section describes the order in which the subcomponent of the system will the implemented, and describes the test plan of the system.

Finally, this document contains two additional sections which describe the effort spent on the project and the list of reference documents.

II. Architectural design

II.A. Overview

The physical components that are involved in the Data4Help service are:

- Individual's cell phone
- Individual's Smartwatches
- Third party's Computer
- TrackMe system (which contains databases and server)

For the service Automated SOS, the following service must be added :

- Ambulance service

Communications between components:

- Cell phone ↔ Server (The cell phone sends requests to the server and the server answers)
- Computer ↔ Server (The computer sends requests to the server and the server answers)
- Server ↔ Database (The server makes queries and the database answers)
- Smartwatch ↔ Cell phone (The cell phone sends queries to the smartwatch and the smartwatch sends data to the cellphone via Bluetooth)
- Cell phone → Ambulance service (The cell phone sends SMS to an ambulance service)

The server must send data to the different users (individuals or third parties) only when it's necessary. Because the users use the app through devices that can be offline, the server must only send data when asked by the users. Otherwise the user could never receive the data if his device is offline. For those reasons, an event-based system would not be adapted

We need to design a system which involves many stakeholders such as individuals, third parties, ambulance services, track systems. Moreover, in all the interactions the system is providing a service to the users so we decided to use a client-server architectural approach.

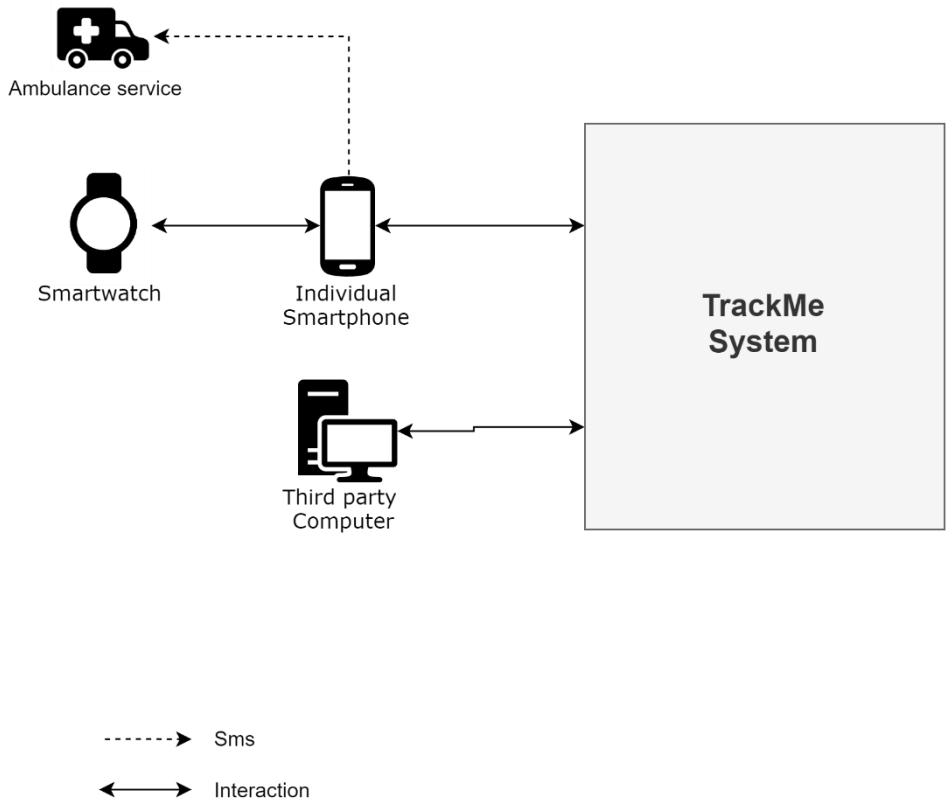


Figure 1 Overview 1

Going deeper in the analysis of the components of the TrackMe system, we are able to identify three different layers :

- **Communication Interfaces :**

  This lawyer contains the interfaces components that allow the system to communicate with external agents (individual's smartphones and third parties' computers). As TrackMe system interacts with different external agents, it needs to have different communication Interfaces :

  - A software module is needed in order to provide functionalities of the system to the individuals on their smartphone. This software module should allow individuals to send requests to the server using an API.
  - A software module is needed to provide the functionalities of the system to the third parties. This module includes a website back -end and an API. The website back-end allows third parties to communicate with the server using a website ; the API allows the parties to download the data requested.

- **Business Logic :**

  This lawyer focuses on the application logic of the TrackMe system. The business logic manages the individuals data, the third parties requests and their subscriptions ; for each of these functions, several software modules are necessary. Those modules will use the communication interfaces between the end user interface and the database. The business logic module receives from the communication interface orders to do specific actions; then the business logic ask to the data base

interface the required data to execute these actions, and finally, returns the result to the communication interface.

- **Data Interface**

This lawyer contains all the modules that allow to store the data produced or retrieved from external resources. These modules allow interaction between the Business Logic modules and the System Databases.

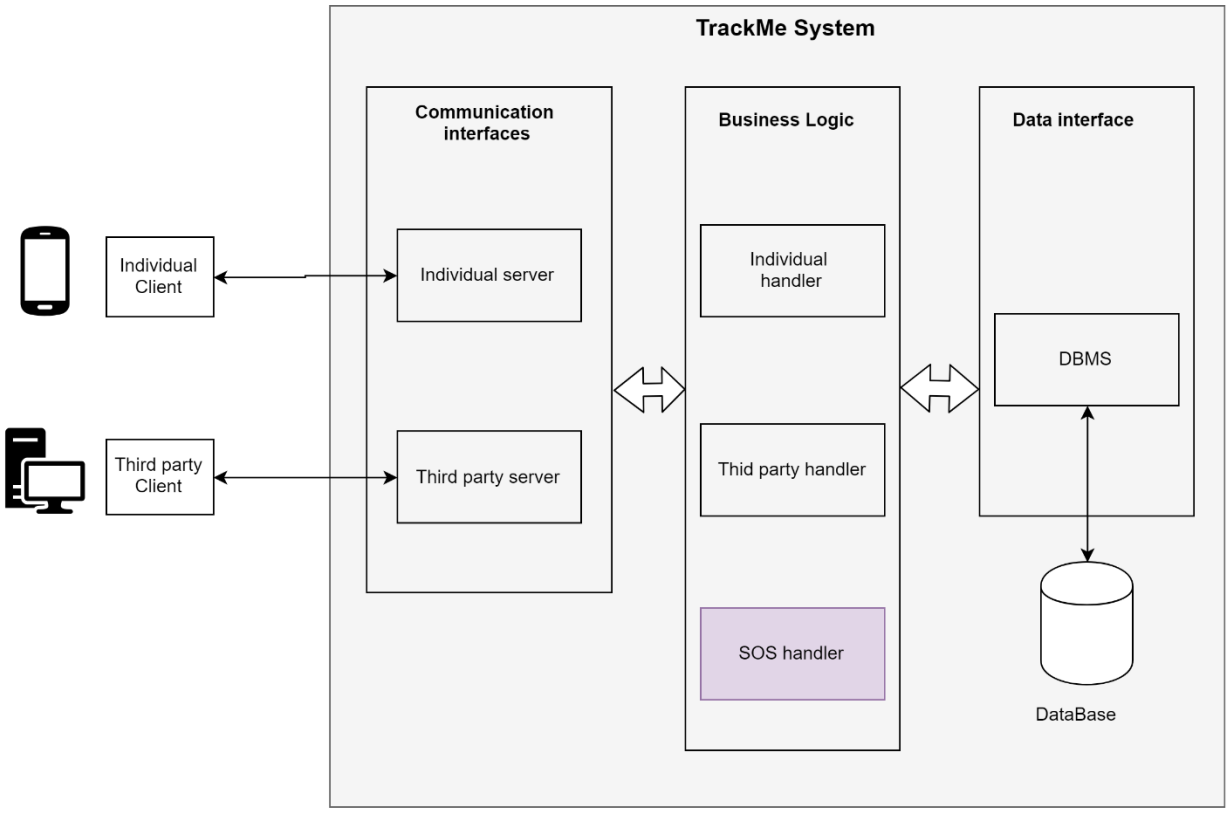


Figure 2 : Overview 2

II.B. Component view

The following diagram, give a high-level representation of the components of the TrackMe system.

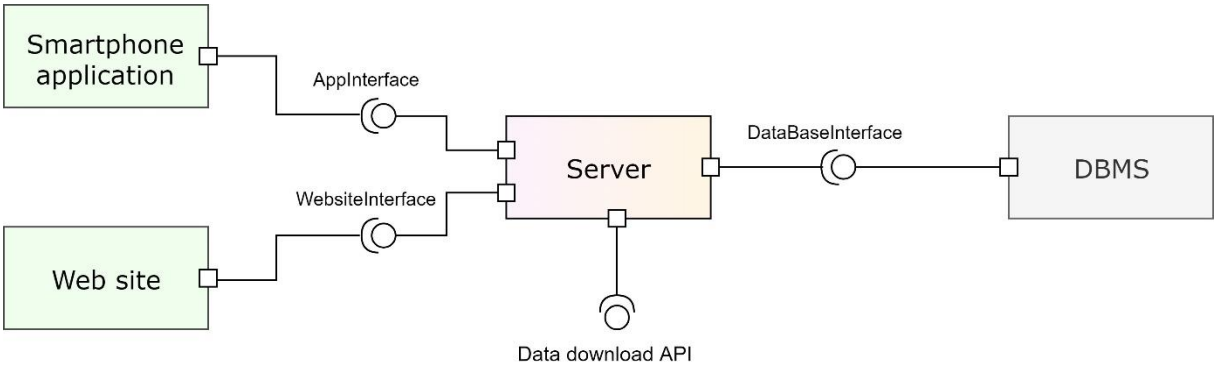


Figure 3: High level representation TrackMe system

As we can see, the system is composed of four main parts : A smartphone application, a Web site, a server and a DBMS. The smartphone application is intended for individuals and the Web site is intended for third parties. The server contains the business logic and make the connection between the others components and the DBMS.

The next component diagram contains a more low-level description of the server components :

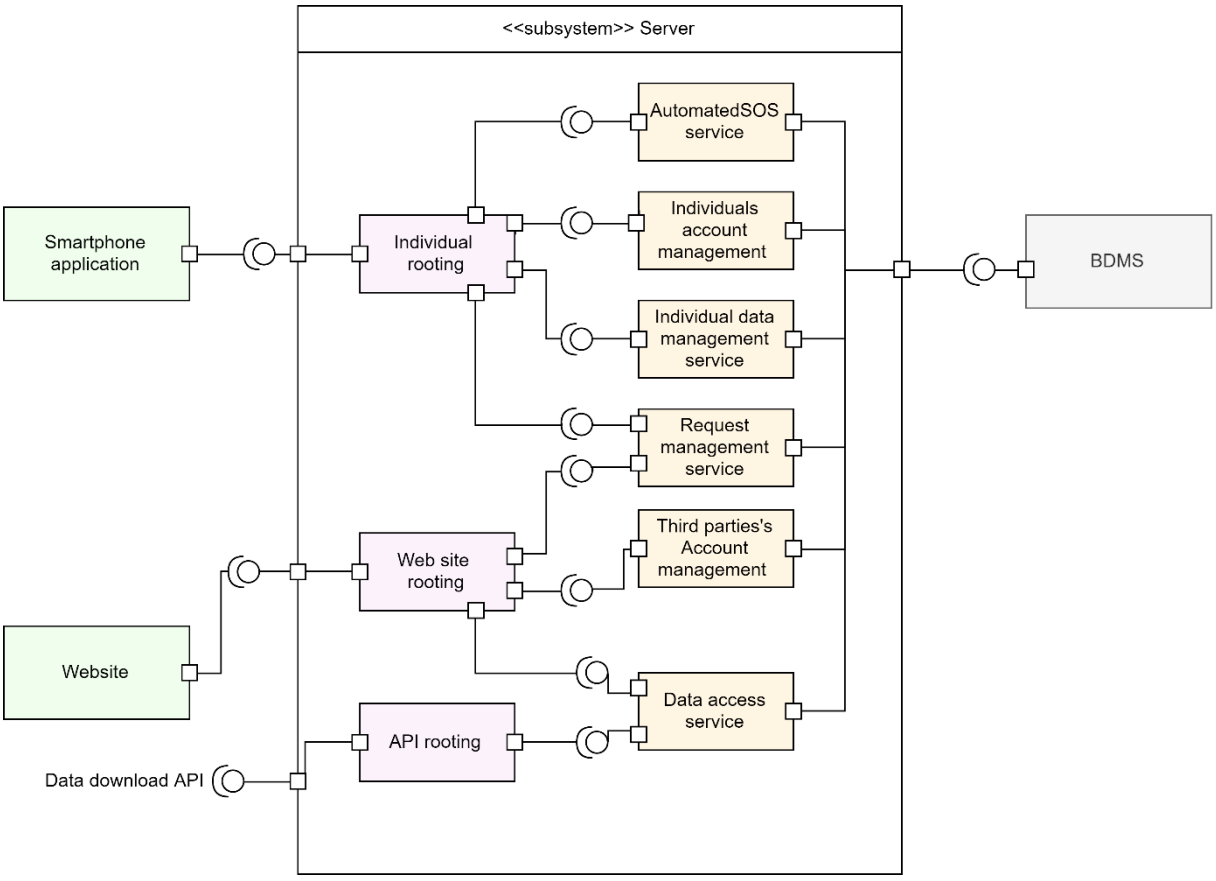


Figure 4: Low level representation trackMe system

As we can see, the server is composed of three main parts : Individual rooting, Web site rooting and API rooting. Those components determine whether a request that is received is valid or not and, if it is, to dispatch it to the relevant service component.

- **Individual rooting.**
  This component manages all the functions the smartphone application has to provide to the individuals.
  Individual rooting redirects the requests from the application to the corresponding component:
  - *AutomatedSOS service* : manages all the functionalities of the AutomatedSOS service
  - *Individuals account management* : manages the functionalities that allow individuals to register, login etc..
  - *Individuals data management service :* manage the logic of the health and position data of individuals
  - *Request management service :* manage the logic of the requests (for individuals : accepting or refusing request)
- **Web site rooting and API rooting.**
  As explained in the RASD, the third parties can access all the functions of the Data4Help system through the Website and have also the possibility to download the data from accepted requests through an API. The API is aimed at accelerating the data acquisition process for third parties who need to acquire data regularly.
  Web site rooting redirect the requests from the website to the corresponding component:
  - *Third parties account management:* manages the functionalities that allow third parties to register, login etc..
  - *Data access service:* manages the logic of formatting data from the Database to the third party.
  - *Request management service:* manage the logic of the requests (for third parties : submitting individual or anonymized requests, subscribing to accepted requests)

The two following diagrams describe more precisely the business logic components, by showing their interactions with the data model. It is assumed that each component is able to invoke operations on the database

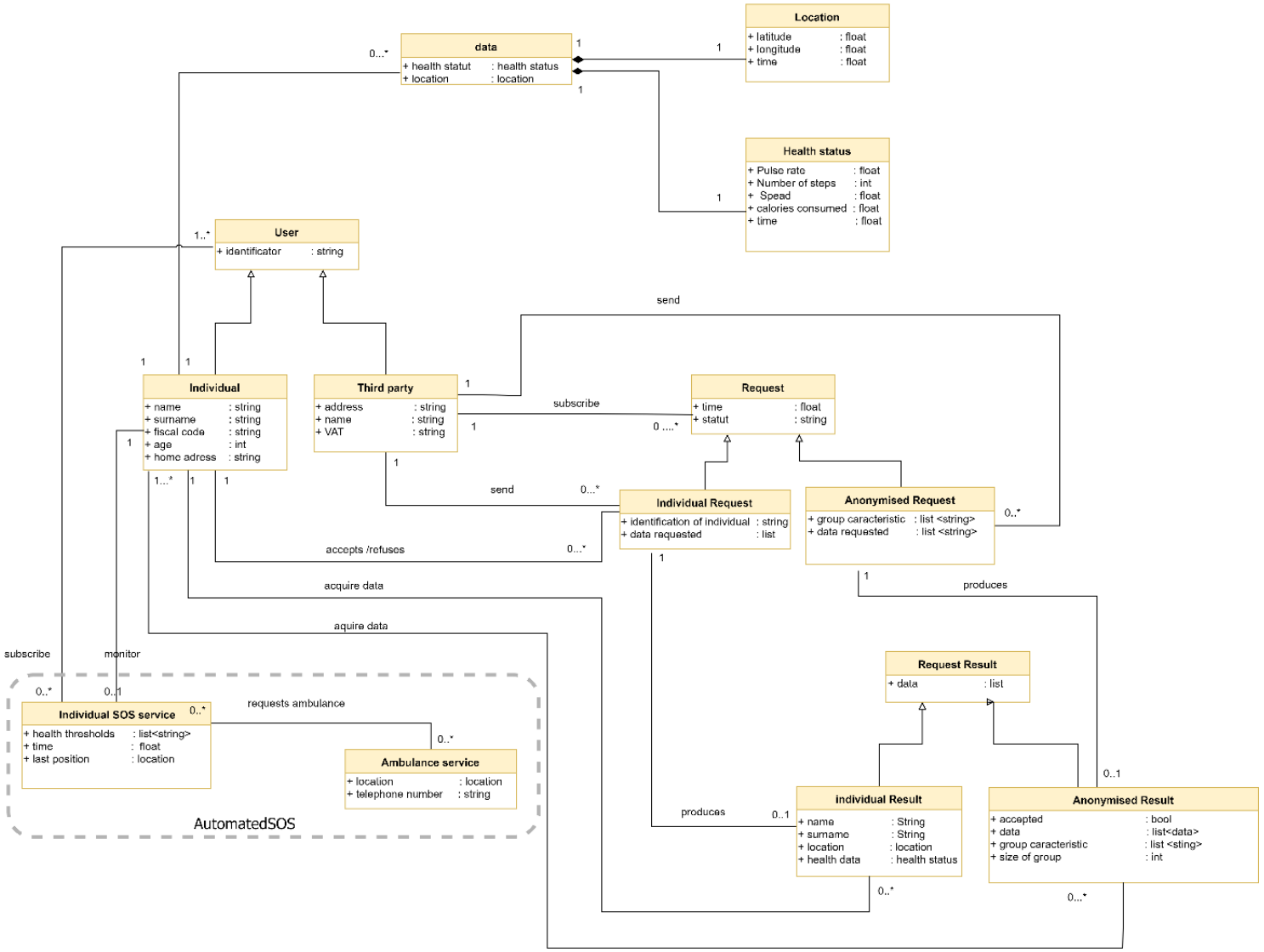


Figure 5: Class diagram of AutomatedSOS system

Some business components contain functions that requires the need to save data in the database. Those functions return a Boolean value : True if the data has been successfully saved, False otherwise. When a function is of this type, the return value is “bool*”. The symbol * make the distinction between functions of this type and other Boolean functions.

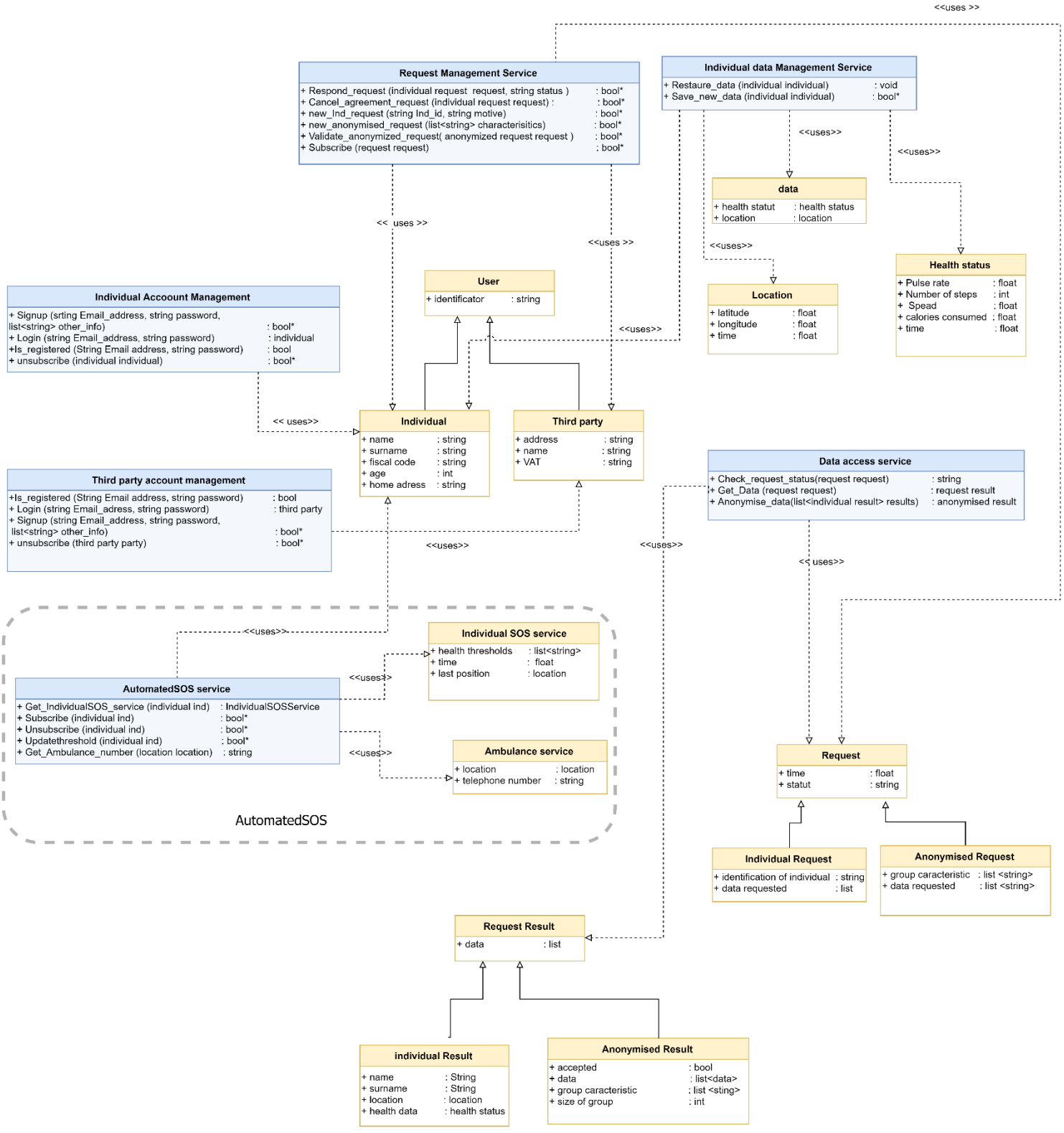


Figure 6: Description of logic components

The following diagram (figure 7) describes in further details the components of the individual's smartphone. A smartphone contains a Database which stores the last health data of individuals. When an internet connection is available, the collected data is send to the TrackMe server.

Moreover, the thresholds and the phone number of the SOS services are also stored in the smartphone database and updated when an internet connection is available. This allows the app AutomatedSOS to send an SOS (using SMS) even without having an internet connection.

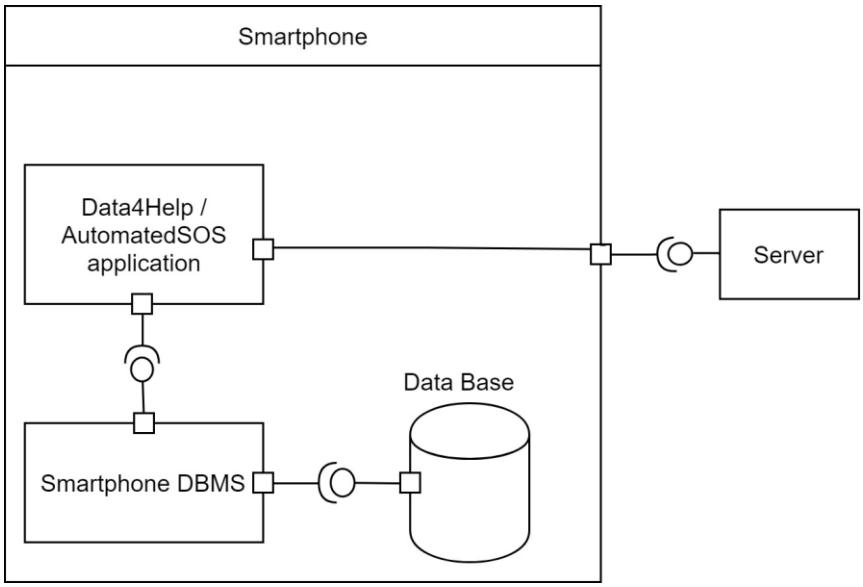


Figure 7: Component view system smartphone

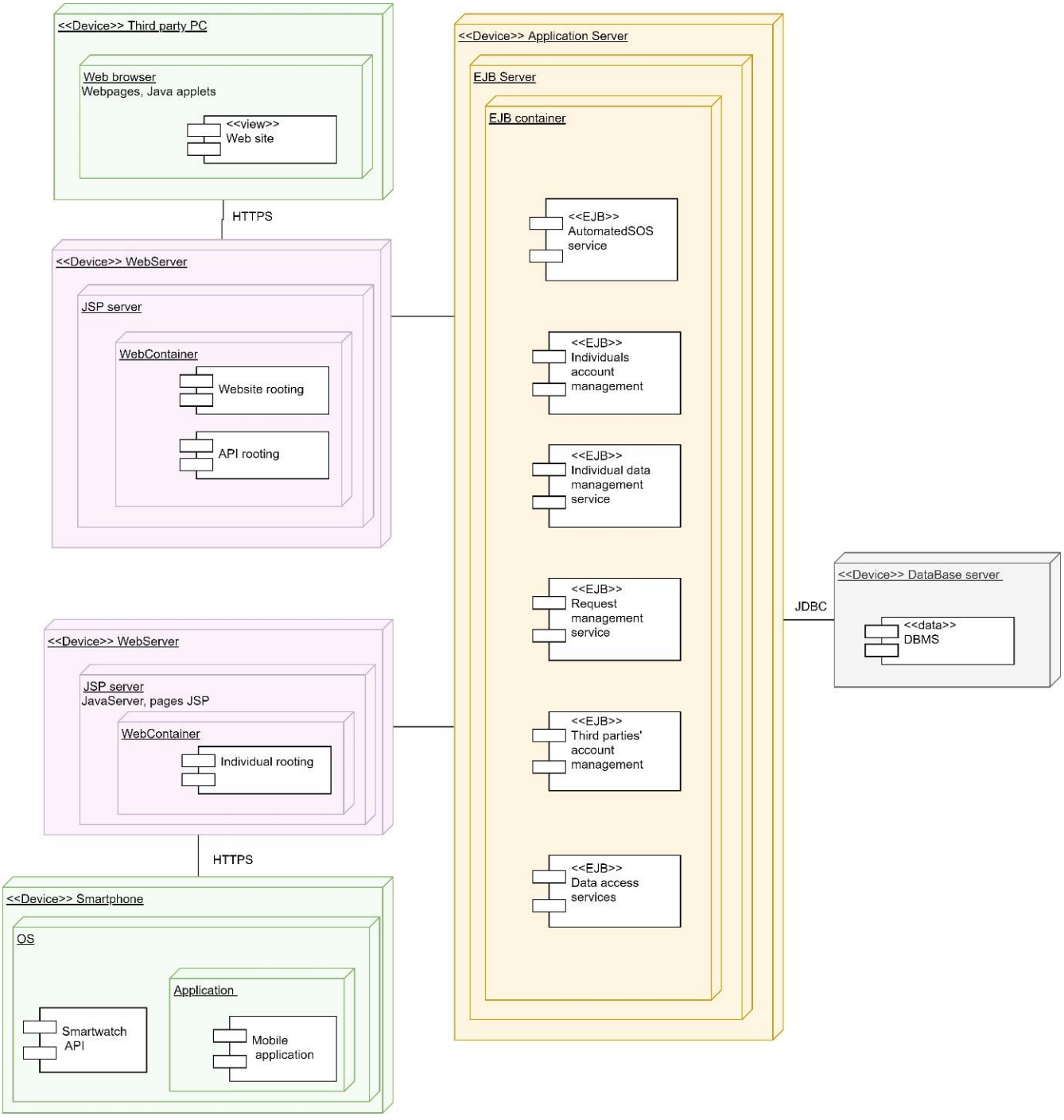
II.C. Deployment view

The following deployment diagram represents the system's use of the physical infrastructure and how the system components are distributed and how they relate to each other.

The application Data4Help and AutomatedSOS require the following devices: A smartphone (for individuals) and a computer (for third parties). Those devices send information to a Webserver. There are two distinct Webservers, one that is in charge of the requests from the smartphone and one that is in charge of the requests from the computer.

The Webserver component communicate with the application server. The application server contains the business logic and communicates with the Webservers and the database server.

The database server stores all the data that is necessary for the Data4Help and AutomatedSOS services. Concerning the data of individuals, the database stores the account information, the history of their health and location data and their thresholds (if they subscribed to AutomatedSOS). Concerning third parties, the database stores their account information, their request and their subscriptions. The database also stores information on ambulance services (phone number and location).


<<Device>> Third party PC
Web browser
Webpages, Java applets
<<view>>
Web site
HTTPS
<<Device>> WebServer
JSP server
WebContainer
Website rooting
API rooting
<<Device>> WebServer
JSP server
JavaServer, pages JSP
WebContainer
Individual rooting
HTTPS
<<Device>> Smartphone
OS
Application
Smartwatch
API
Mobile
application
<<Device>> Application Server
EJB Server
EJB container
<<EJB>>
AutomatedSOS
service
<<EJB>>
Individuals
account
management
<<EJB>>
Individual data
management
service
<<EJB>>
Request
management
service
<<EJB>>
Third parties'
account
management
<<EJB>>
Data access
services
JDBC
<<Device>> DataBase server
<<data>>
DBMS

II.D. Runtime view

In this section we represent some runtime views of the interactions between the components of the Data4Help and AutomatedSOS systems.

II.D.1. A third party makes an anonymized request

Third parties can make individual or anonymized requests. In the case where a third party submit an anonymized request, the validation (or non-validation) of the request can be done immediately as the choice does not involve a response for the individuals concerns.

The third party submits the anonymized request though the website interface. Then, the website rooting service transfers the request to the Request management service that carries out the validation.

The validation function implies an SQL request to the DBMS in order to obtain the number of individuals whose data satisfy the request. Then the status of the requests (validated or refused) is updated in the database and the response is communicated to the third party.

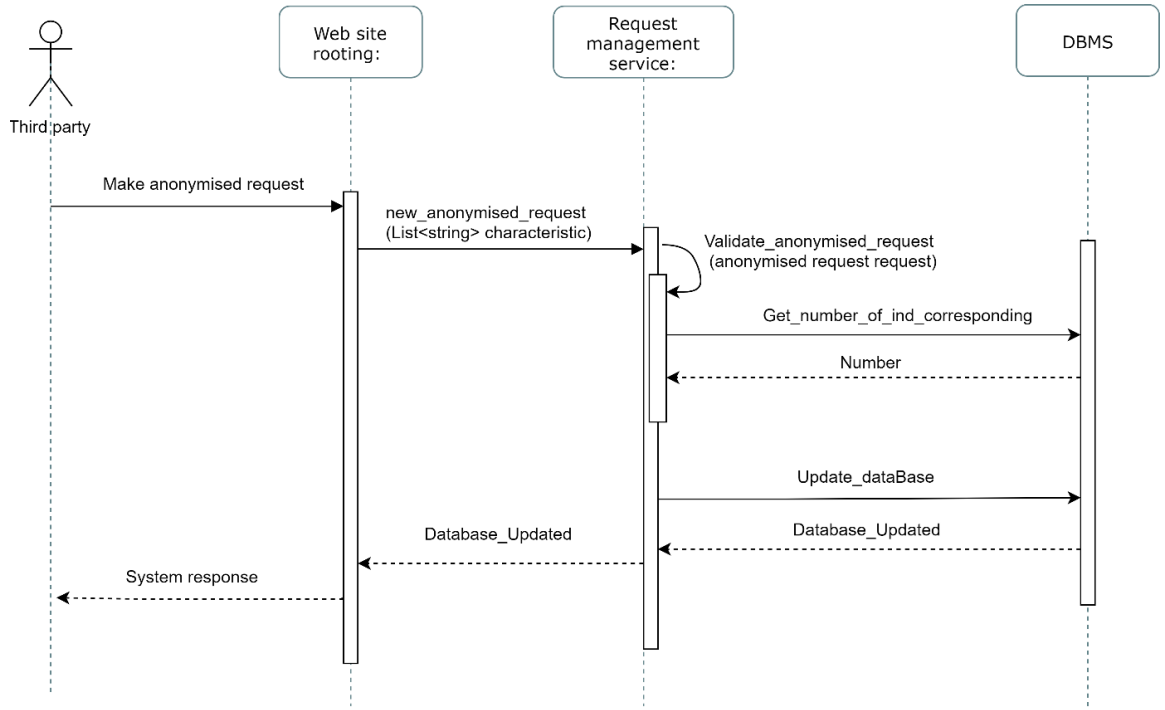


Figure 8: A third party makes an anonymized request

II.D.2. A third party makes an individual request

Third parties can make individual or anonymized requests. In the case of an individual request the validation of the request is based on the response of the individual concerned.
The third party submits the anonymized request though the website interface. Then, the website rooting service transfers the request to the Request management service that saves it in the database with the status “waiting for validation”. Then the third party receives a confirmation message that his request has been saved.

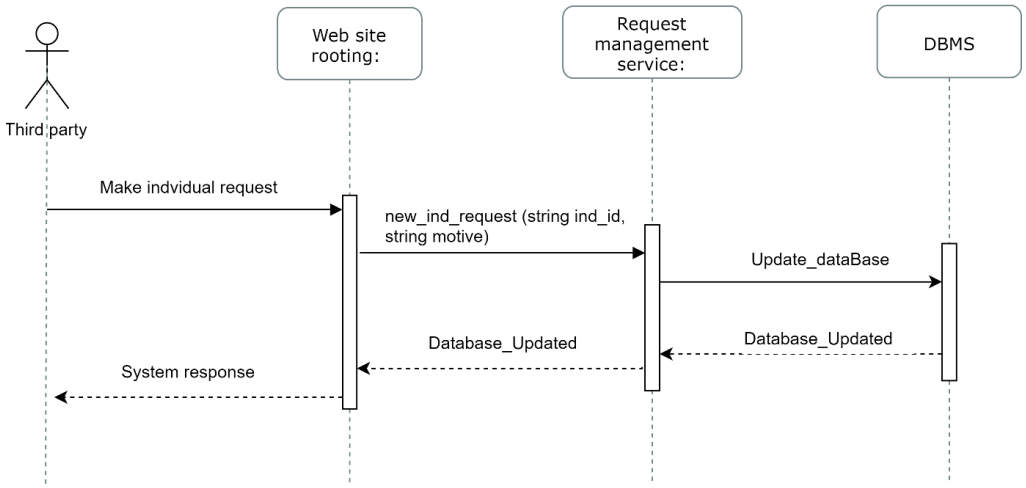


Figure 9: A third party makes an individual request

II.D.3. An individual responds to a request

Individuals may be the subject of requests from third parties. Whenever an individual logs in on the smartphone application, the application rooting service make a request to the DBMS asking for the unanswered requests. Then the application displays on the screen of the user's smartphone all the requests that are waiting for a response.

For each unanswered request, the individual can make the choice between accepting or refusing it by pressing a button. The individual's response is then transferred to the application rooting to the request management service that transfer the modification of the request's status to the database. Then, a confirmation message is printed to the user's smartphone screen if the request's new status is successfully saved in the database

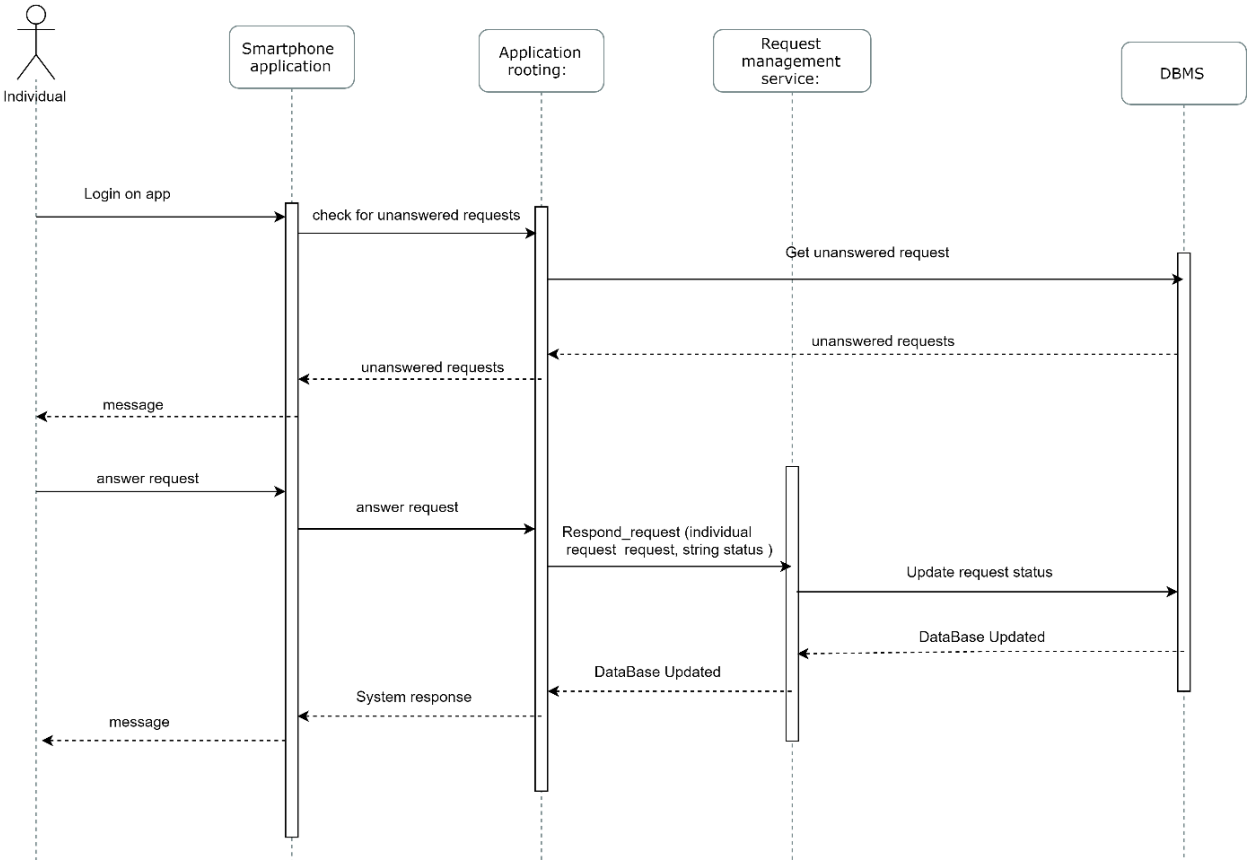


Figure 10: An individual responds to a request

II.D.4. A third party subscribes to a request

Third parties have the possibility to subscribe to their accepted requests. By subscribing to a request, a third party has the possibility to download the newly available data corresponding to the request without going again through the new-request process. When a third party makes a request and when that request is accepted, the third party has access to the data until the date on which the request was made. When a third party subscribe to a request, he has access to the up-to-date data as long as his request stays valid.

When a third party logs in the Data4Help website, if he has an accepted request, he has the possibility to subscribes to it (if a request is not accepted, the "subscribe button" does not appear). When the third party clicks on subscribe, the website rooting transfers the request to the Request management service which transfers it to the data access service that asks to the DBMS the status of the requests.
If the status is accepted, the status of the request is updated in the database and the third party receives a confirmation message.
This double check increases the security and avoid third party being able to access data from unwilling individuals.

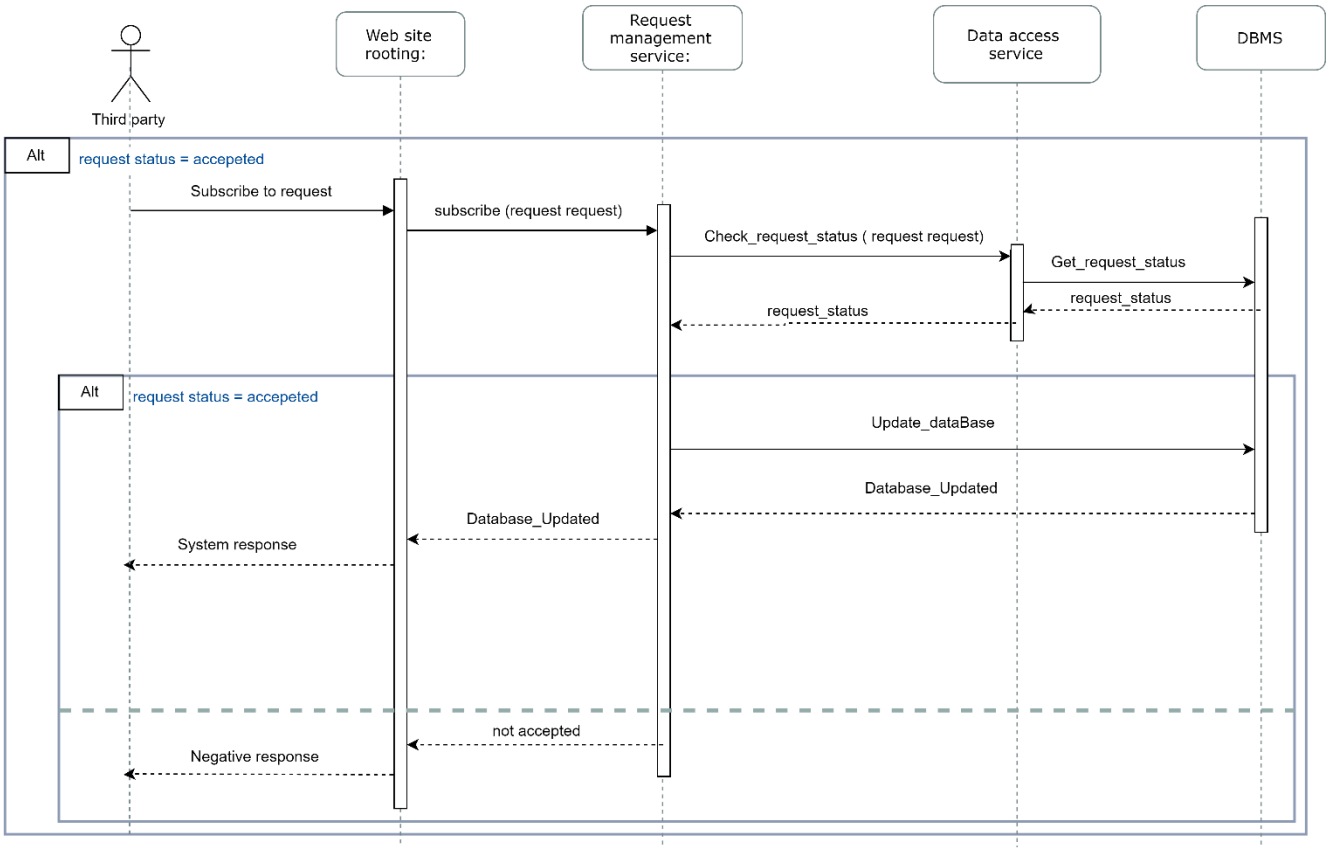


Figure 11: A third party subscribes to a request

II.D.5. A third party download data from an anonymized request

When a third party logs in the Data4Help website, a request is made to the Website rooting service in order to know what needs to be displayed on the third party's computer screen. If the third party has some accepted requests, the download button will appear on his screen.

When the third party clicks on the download button, the Website rooting service requests to the Request management service to validate the anonymized request. This requires calling the DBMS to count the number of individuals concerned.

Without this double check, if a third party never refreshed the webpage since the moment his request was accepted (so since the moment the download data button was available) and if during this time the number of individuals concerned by the request drop under 1000, a third party would be able to acquire unauthorized data.

If the request is anonymizable, the Website rooting service asks the DBMS to get the needed data. Then the data access service anonymizes the data and sends the result to the third party.

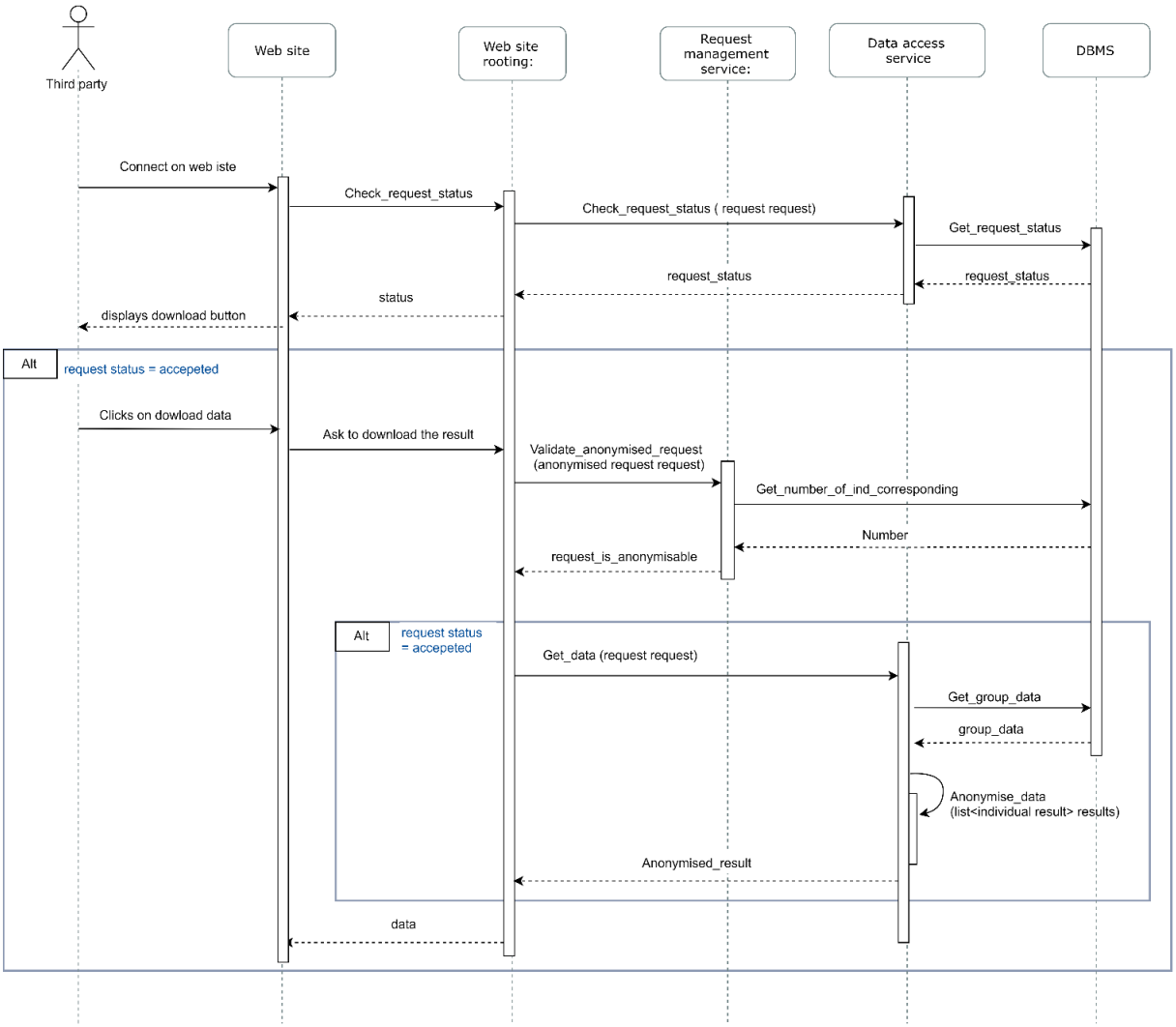


Figure 12: A third party download data from an anonymized request

II.D.6. A third party download data from an individual request

When a third party logs in the Data4Help website, a request is made to the Website rooting service in order to know what needs to be displayed on the third party's computer screen. If the third has some accepted requests, the download button will appear on his screen.

When the third party clicks on the download button, the Website rooting service requests to the Data access service to get the required data. To do so, the data access service checks the request status and if the request is accepted, ask the needed data to the DBMS. Then the data is sent to the third party.

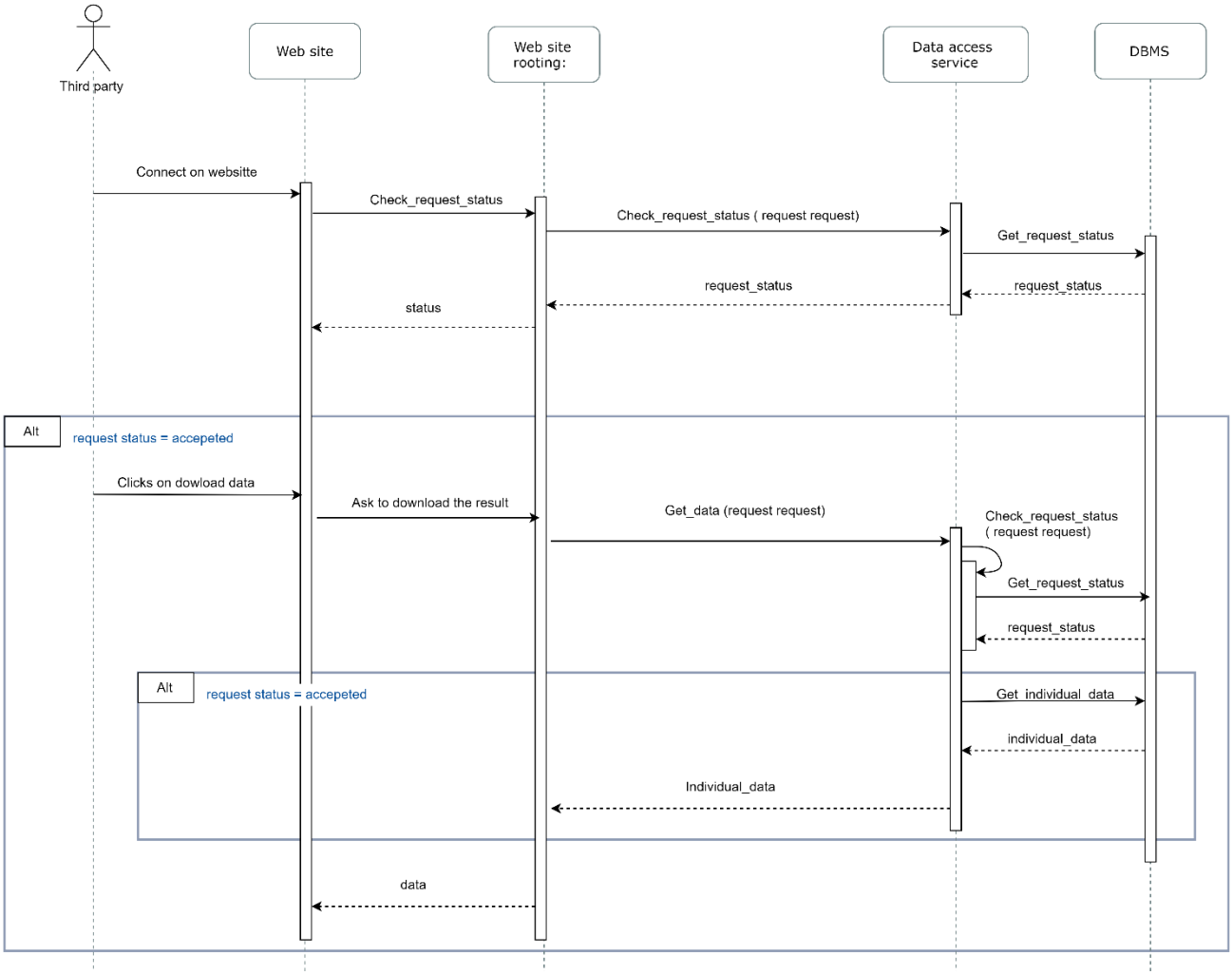


Figure 13: A third party download data from an individual request

II.D.7. AutomatedSOS

The service AutomatedSOS must send an SOS to an ambulance service when the health parameters of the user are below certain thresholds. The service must therefore be functional even without an internet connection.

To do so, the emergency data (the thresholds and the phone number of the closest emergency service) must be regularly updated (every five minutes) and stored in the smartphone internal database when an internet connection is available. This is what is shown on the first sequence diagram (figure 14).

Therefore, the AutomatedSOS service will be able to detect when the individual is bellow threshold and to send the emergency SMS using the data collected when an internet connection was available.

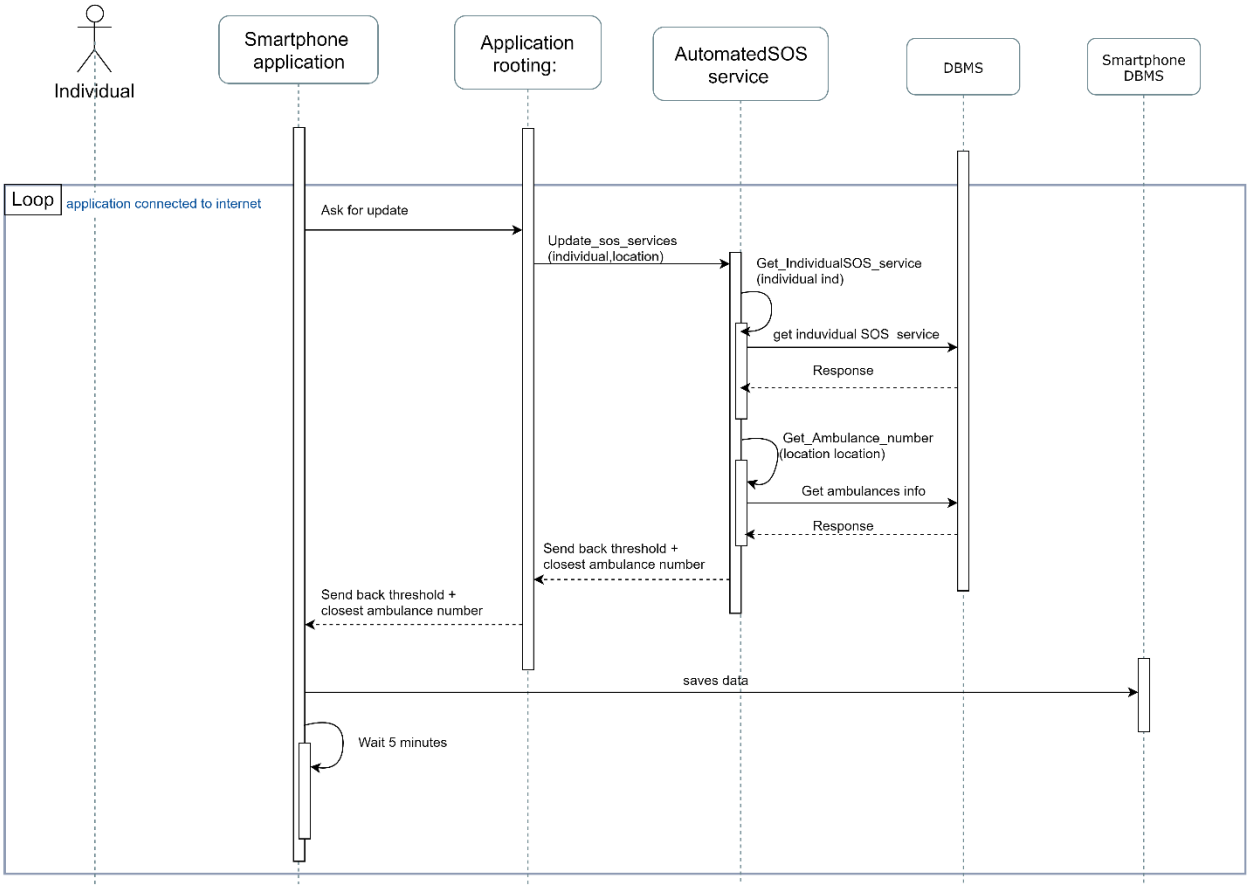


Figure 14: Updating emergency information when online

The following diagram shows the process of sending an emergency SMS. This process does not need any internet connection has it uses only the data stores in the smartphone database.

The smartphone application acquires new data from the users, saves the data in the internal database and checks if the thresholds are exceeded. If they are, the smartphone sends an SMS to the closest emergency service. The process from the acquisition of new data to the sent of the SMS must last less than 5 seconds.

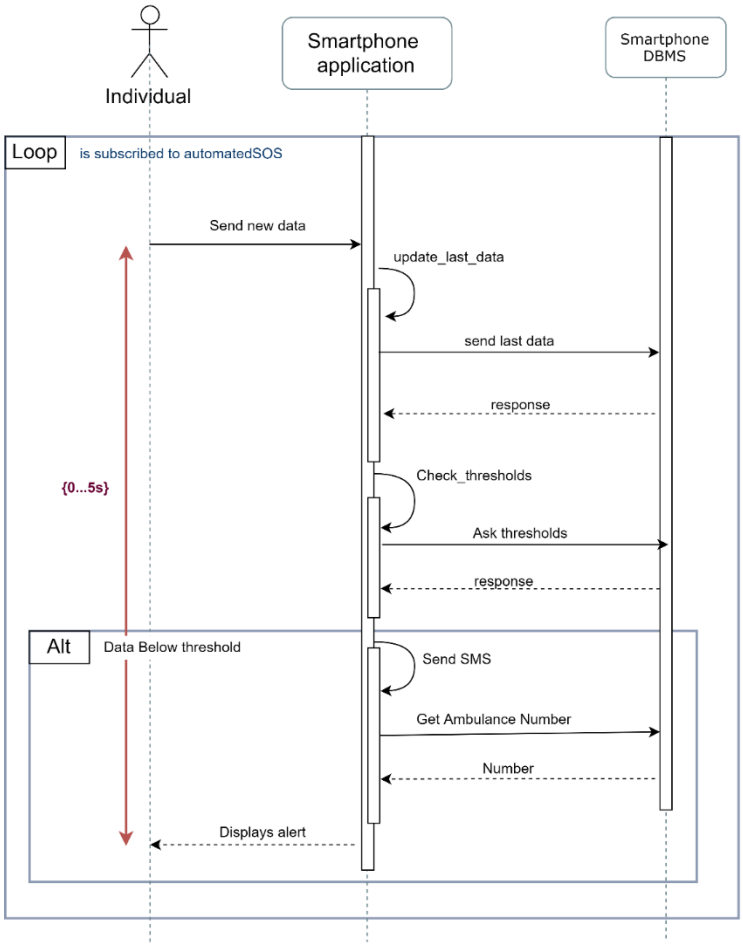


Figure 15: Sending SOS

II.D.8. The application sends new data to the Data4Help server

The Data4Help application sends the health and location data of the individual from the smartphone data base to the Data4Help server.

The data is retrieved in the smartphone database and is sent by the Application rooting service to the Individual data management service that send the new data to the Data4Help DBMS.

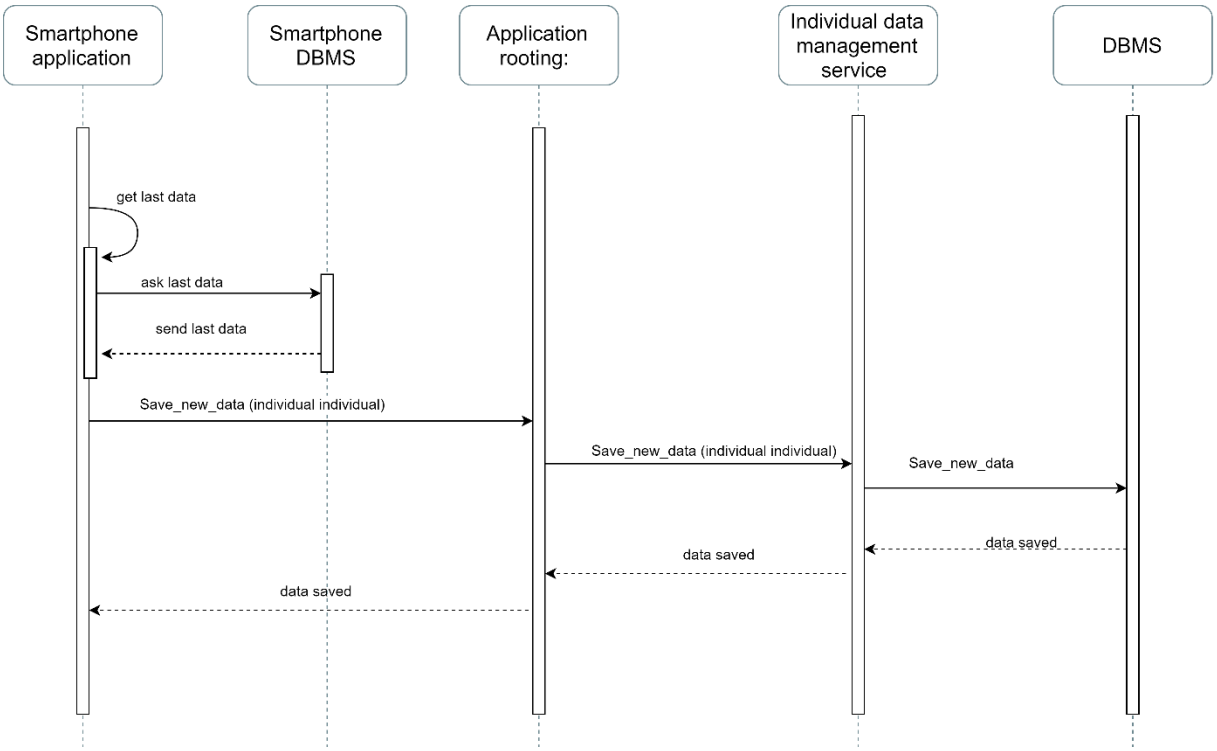


Figure 16: The application sends new data to the Data4Help server

II.E. Component interfaces

The component interfaces where already described in the RASD document and in the component view of this document.

II.F. Selected architectural styles and patterns

II.F.1. Architectural design

Architectural design is concerned with understanding how a system should be organized and designing the overall structure of that system. In the model of the software development process, architectural design is the first stage in the software design process. It is the critical link between design and requirements engineering, as it identifies the main structural components in a system and the relationships between them. The output of the architectural design process is an architectural model that describes how the system is organized as a set of communicating components.

We choose agile processes, so we can empower the process of designing and developing Data4Help and AutomatedSOS , our target it was to apply the four important values of the agile manifesto that are :

1- Focus should be more on individuals and interactions instead of processes and tools.

2-Working software is more important that comprehensive documentation.

3-Customer collaboration is more vital than contract negotiation.

4-The process should respond to change rather than follow a plan.

Not only but we took in consideration also the 12 principles of agile software development:-

1-Deliver customer satisfaction by delivering valuable software continuously.

2-Always accept change of requirements matter how early or late in the project.

3-Deliver software that works within a shorter timescale.

4-Both developers and business professionals must work closely together daily throughout the duration of the project.

5-Information is best transferred between parties in face-to-face conversations.

6-Motivate people to build a project by creating an environment of appreciation, trust, and empowerment.

7-Working software is the key measure of progress.

8-The agile process promotes sustainable development.

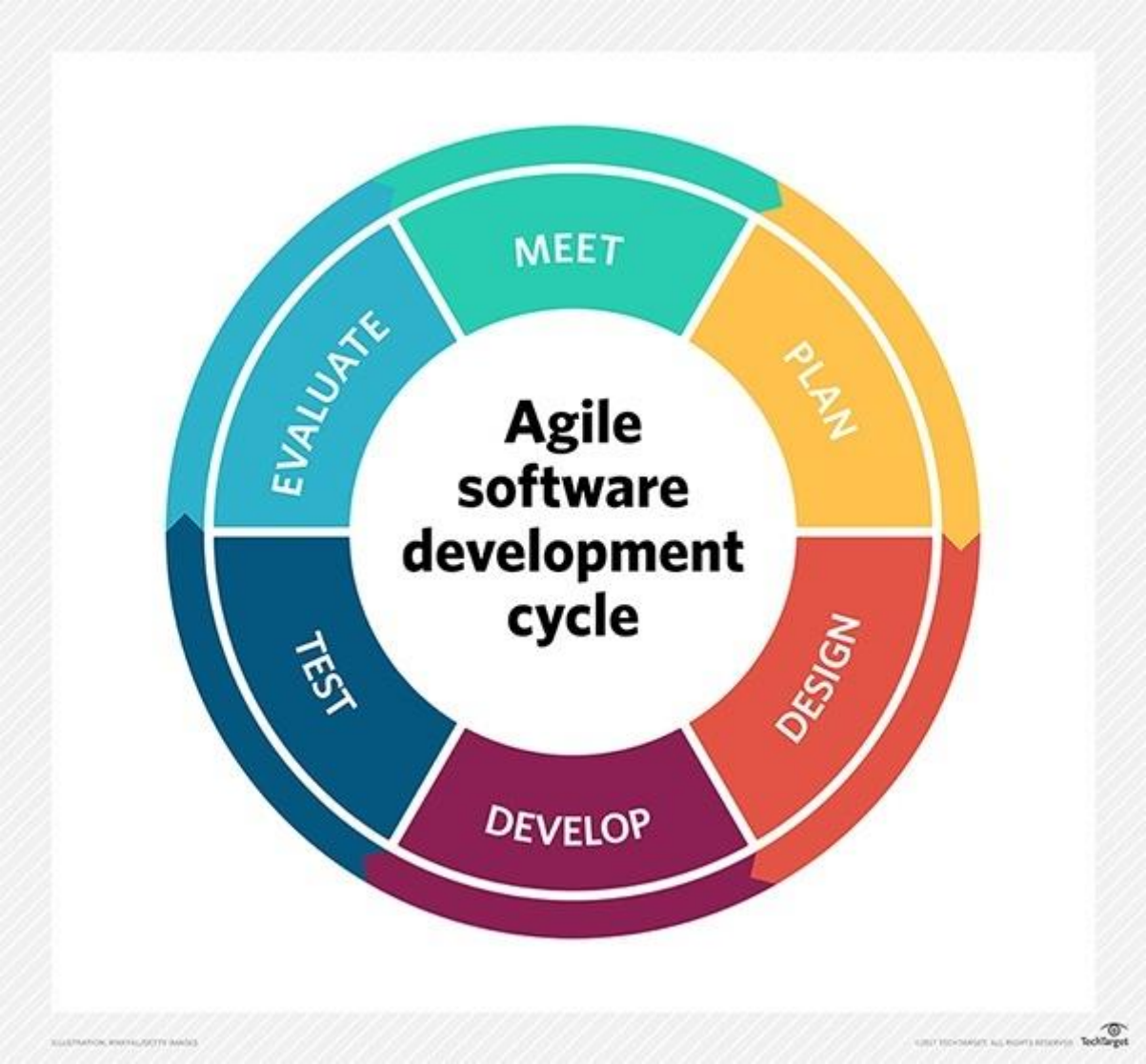
9-Continuous attention to excellence and quality in technical development and design boosts the agility.

10- Simplicity is a vital part of effective agile management.

11-Self-organized teams produce the best architecture, requirements, and design.

12-Teams should reflect through inspection and adaption to be more effective.

Also this figure describe the agile process: -



II.F.2. Design Pattern

-Repository pattern

We choose Repository pattern because large amounts of data will be shared with TrackMe that will have a large repository that third party companies will access these data, we saw that repository pattern is the best choice for this project.

This table describes well the reason why we choose the repository pattern

Name	Repository
Description	All data in a system is managed in a central repository that is accessible to all system components. Components do not interact directly, only through the repository.
Why	We use this pattern because the system have large volumes of information that need to be stored for a long time.
Advantages	Components can be independent—they do not need to know of the existence of other components. Changes made by one component can be propagated to all components. All data can be managed consistently (e.g., backups done at the same time) as it is all in one place.
Disadvantages	The repository is a single point of failure so problems in the repository affect the whole system. May be inefficiencies in organizing all communication through the repository. Distributing the repository across several computers may be difficult.

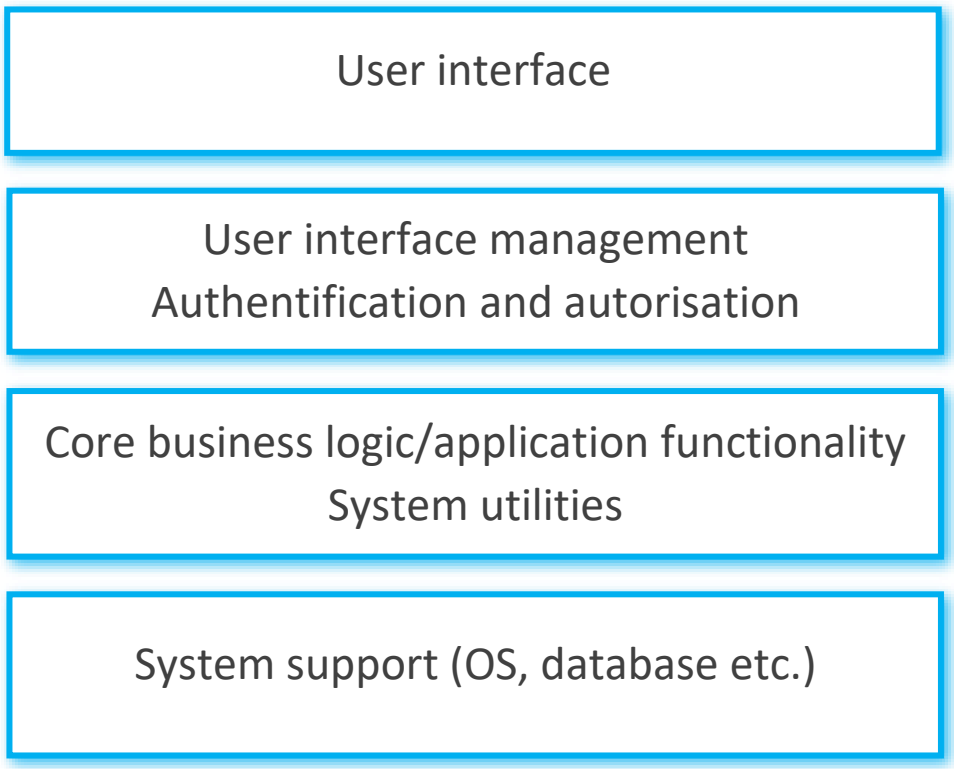
-Layered pattern

In addition to the repository pattern we also decided to use the layered architecture pattern to model the interfacing of sub-systems, the layered pattern also is useful in organizing the system into a set of layers (or abstract machines) each of which provide a set of services, and last but not least supports the incremental development of sub-systems in different layers. When a layer interface changes, only the adjacent layer is affected.

This is a table that describes well why we have chosen this pattern: -

Name	Layered architecture
Description	Organizes the system into layers with related functionality associated with each layer. A layer provides services to the layer above it so the lowest-level layers represent core services that are likely to be used throughout the system.
Why	We decided to use it when building new facilities on top of existing systems, to be more specific building AutomatedSOS on top of Data4Help, also it is useful when the development is spread across several teams with each team responsibility for a layer of functionality.
Advantages	Allows replacement of entire layers so long as the interface is maintained. Redundant facilities (e.g., authentication) can be provided in each layer to increase the dependability of the system.
Disadvantages	In practice, providing a clean separation between layers is often difficult and a high-level layer may have to interact directly with lower-level layers rather than through the layer immediately below it. Performance can be a problem because of multiple levels of interpretation of a service request as it is processed at each layer.

This is a picture that describes a generic layered pattern :-



Explaining this generic layered pattern and linking it with the software :-

Layer 1 (User interface) :- we have 3 user interfaces (TrackMe UI , Third party UI and User UI).

Layer 2 (User interface management authentication and authorization) :- In this layer each created account will have different authorities as described before in the RASD.

Layer 3 (Core business logic/application functionality system utilities) :- For example AutomatedSOS will calculate some data to send a request to the ambulance .

Layer 4 (System support(OS, database etc.)) :- We also mention before in the RASD that the mobile application will work on android platform and there will be a web application for both TrackMe and Third parties .

III. User interface design

The two end users are third parties and individuals. The user interfaces are presented for both types of users in the RASD document.

IV. Requirements traceability

In the following table presents a mapping correspondence between the requirements defined in the RASD related to each goal and the components identified in the server component diagram.

Goal	Requirement	Component
[G1] : Third parties must be able to request to access to the data of specific individuals or to anonymized groups of individuals.	R.8 ; R.9	DBMS, request management service, Data access service, Individual data management service
[G2] : At any time, third party should never have access to data of specific individuals without their agreement.	R.10 ; R.11	DBMS, Data access management system
[G3] : Third parties must have the possibility to subscribe to new data if their request is accepted.	R.12	DBMS, request management system
[G 4] : Individuals must be able to consult their data and accept/refuse requests	R.13	DBMS, request management system, Individual data management service
[G1], [G2], [G3] & [G4]	R.1 ; R.2 ; R.3 ; R.4 ; R.5 ; R.6 ; R.7	DBMS, Individual account management, Third partie's account management
[G 5] : An ambulance is requested to the location of the individual with a reaction time below 5 seconds from the time the parameters are below threshold	R.14 ; R.15 ; R.15 ; R.17	DBMS, AutomatedSOS service, Individual account management

V. Implementation and test plan

V.A. Implementation

For the implementation of Data4Help and AutomatedSOS as we mentioned before we will use agile methodology, we took in consideration that AutomatedSOS work on top of Data4Help , so we have first to implement and finish Data4Help completely and test all its functionalities then start implementing AutomatedSOS .

1. Data4Help
    a. Mobile application with all its functionalities.
    b. TrackMe interface with all its functionalities.
    c. Third party interface with all its functionalities.

We planned that a , b and c can be implemented simultaneously then each part tested alone as we will specify later in the testing section then test all together to make sure that Data4Help work correctly and start building AutomatedSOS on top of it so we can minimize the number of errors also we will be able to track any error in AutomatedSOS easily if we finished Data4Help completely well .

Also we are taking in consideration that implementation is not only programming, we will try to avoid some problems and issues like :-

- **Reuse**:- We believe that modern software is constructed by reusing existing components or systems. Implementing the software we will try to make as much use as possible of existing code.

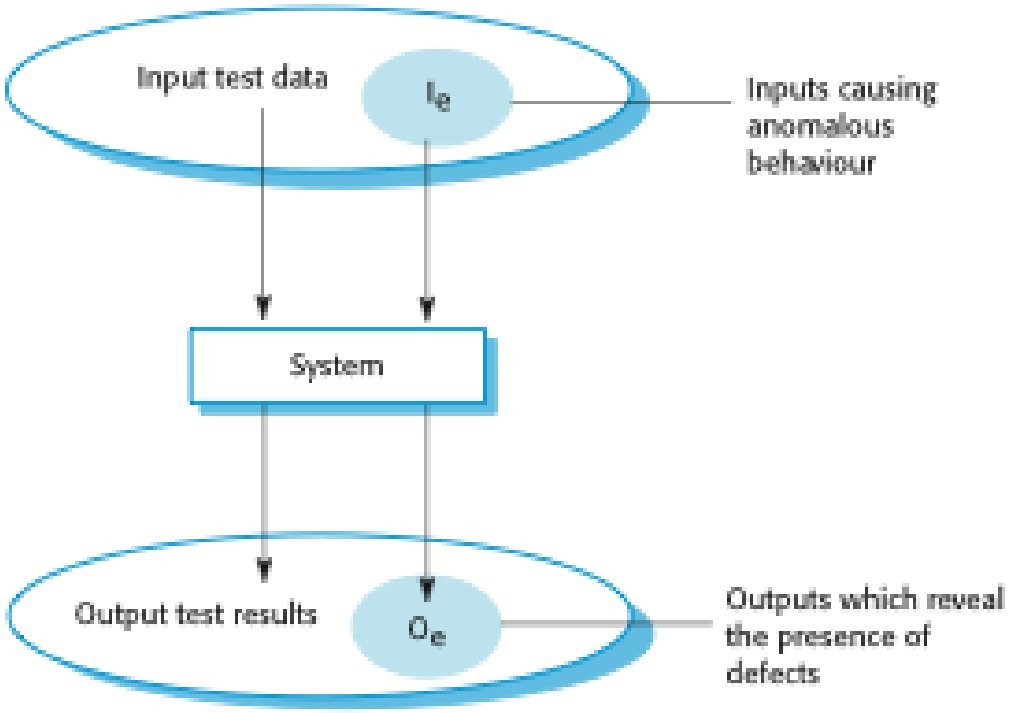
- **Configuration management :-** During the implementation we intend to release more than a version each one with modified characteristics and improvements , we will save all the previous versions in case of need we can find what we want.

- **Host-target development** :- We plan to develop the software on a specific computer then we will test it on different computers to make sure that it will not make any problems.
-

V.B.Testing

-Test Methodology

Several aspects must be tested in order to conduct the progress of the system. The main target of testing the system is to make sure that all the functions works correctly, and the response time is accurate. In this section we will discuss the software and user test.



-Stages of testing

We intend to follow the three stages of testing to make sure that every part in the software works correctly, we will explain in this section the three stages that we are planning to do :-

- Development testing, where the system is tested during development to discover bugs and defects.
- Release testing, where a separate testing team test a complete version of the system before it is released to users.
- User testing, where users or potential users of a system test the system in their own environment.

Every stage of this stages will be explained now.

-Development testing

Development testing includes all testing activities that are carried out by the team developing the system.

- Unit testing, where individual program units or object classes are tested. Unit testing should focus on testing the functionality of objects or methods.
  Units may be:
  - Individual functions or methods within an object
  - Object classes with several attributes and methods

- Component testing, where several individual units are integrated to create composite components. Component testing should focus on testing component interfaces.
- System testing, where some or all of the components in a system are integrated and the system is tested as a whole. System testing should focus on testing component interactions, in our case tasting if AutomatedSOS works correctly with Data4Help.

-User Test

Another aspect that we have aimed for is to provide the end-user a simple user interface with usability in mind. When designing the user interface, we have tried to make the components of the GUI self explanatorily and resemble common icons that users are familiar with. To test the usability of the system, several users will have to test the software. The computer experience of these testers is inexperienced in using computer. Prior the testing the goal and basic functionality of the system are told to the user. During the test we will ask the user about events that may have occurred during the testing of the system such as graphical warnings and sound events. Finally, at the end of each test we will ask the user if there is any ambiguity while testing the system and what recommendation they have.

Also we intend to follow the three user test phases that are :-

- Alpha testing

Users of the software work with the development team to test the software at the developer's site.

- Beta testing

A release of the software is made available to users to allow them to experiment and to raise problems that they discover with the system developers.

- Acceptance testing

Customers test a system to decide whether or not it is ready to be accepted from the system developers and deployed in the customer environment. Primarily for custom systems.

-Dataset for testing

As we mentioned before Data4Help and AutomatedSOS will contain a lot of information about registered people and testing this before launching the software will be a big challenge so we intend to use a ready dataset with big number of data about people ,normal people and people with special health conditions , using a dataset will facilitate the process of testing also we will be able to check if our results are completely correct or not because we will have all the references that we need from this dataset , we didn't decided yet what database we will use , but we are sure that datasets for this specific kind of software are available on the web.

-Automated testing

As we mentioned before that our software will need to manage a big amount of data that's make testing all the components in the software very complicated , in this case we intend to use an automated testing tool that is (JUnit).

VI. Effort spent

VI.A. Mohamed

Date	*Work*	*Hours*
20/11/2018	Reading and understanding the DD	2
24/11/2018	Reading and deciding architectural style	2
26/11/2018	Writing the agile methodology	1
30/11/2018	Choosing the patterns	2
1/12/2018	Testing	3
3/12/2018	Changing patterns and adding new pattern	2
5/12/218	Correcting some concepts in agile methodology	1
7/12/2018	Completing testing part and Implementation	2
10/12/2018	Reading all the document and correcting further errors	2

VI.B. Emma

Date	*Work*	*Hours*
18/11/2018	Writing the main components in draft form	1
25/11/2018	Component view diagrams	1
27/11/2018	Component view + deployment view	2
1/12/2018	Component view + deployment view	2
2/12/2018	Deployment view diagrams	1
3/12/218	Added explanation text on Component view	1
5/12/2018	Added explanation text on Deployment view + Runtime view diagrams	4
6/12/2018	Runtime view diagrams	3
7/12/2018	Runtime view diagrams + Explanation text	2
8/12/2018	Update on runtime view + Requirements traceability	3
9/12/2018	Reading everything again + correcting some mistakes + putting everything together	2

VII. References

[1] SOMMERVILE, Iam. *Software engineering 9*. International edition.

[2] Assignment *Mandatory Project: goal schedule, and rules*.

[3] Les numériques, *COMPARATIF / Quelle montre connectée choisir ?* https://www.lesnumeriques.com/montre-connectee/comparatif-montres-connectees-a1781.html