POLITECNICO
MILANO 1863

TrackMe project - Argiro' Anna Sofia,
Battaglia Gabriele, Bernardo Casasole

Design Document

Deliverable:	DD
Title:	Design Document
Authors:	Argiro' Anna Sofia, Battaglia Gabriele, Bernardo Casasole
Version:	0.3
Date:	December 2, 2018
Download page:	https://github.com/BernardoCasasole/ArgiroBattagliaCasasole.git

Contents

1. Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions

- *User*: a person, third-party or user, that has registered;
- *Individual User*: every registered person from whom the system collects data;
- *Third-Party User*:every entity registered with the purpose to request data for external use;
- *Live Data*: the data on a IU produced in real time.
- *Stored Data*: the data on a IU collected so far.
- *Data Request*: a request for data made from a TPU.
- *Stored Data Request*: a data request for stored data.
- *Subscription Request*: a request for subscribing to newly generated data.

1.4 Acronyms

- API: Application Programming Interface
- TPU: Third-party User
- D4H: Data4Help
- ASOS: AutomatedSOS
- T4R: Track4Run
- UX: User experience

1.5 Abbreviations

- Ab: abbrevation

1.6 Revision history

- **v0.1 - 27/11/18** Document created
- **v0.2 - 30/11/18** Component view
- **v0.3 - 2/12/18** Model diagrams, User inteface and High level overview

1.7 Document Structure

Introduction

Architectural Design

User Interface Design

Requirements Traceability

Implementation, Integration and Test plan

Effort Spent

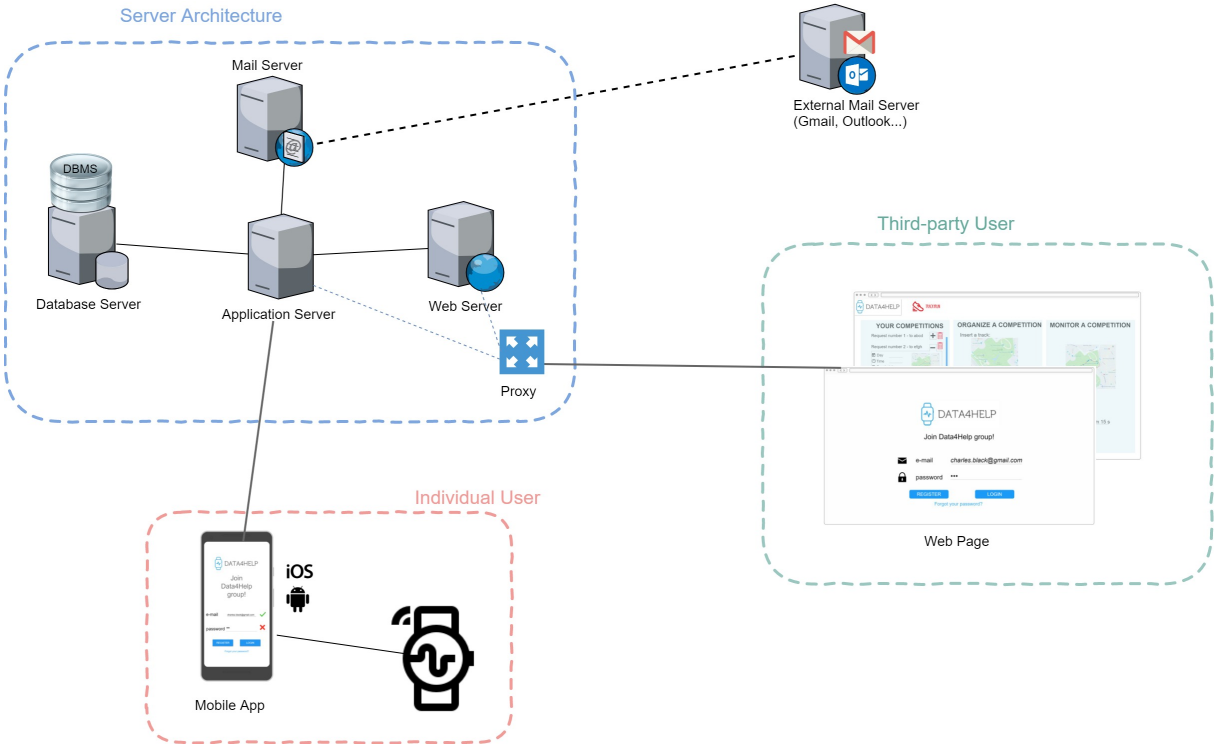
References

2. Architectural Design

2.1 Overview

2.1.1 High level components and basic interactions

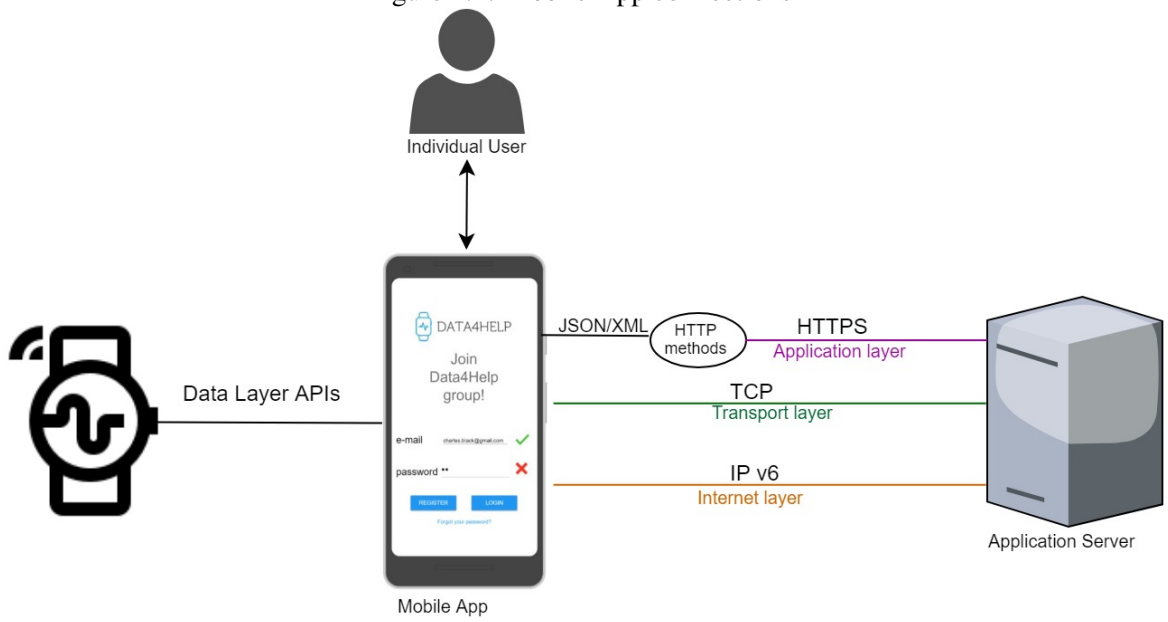
Figure 2.1: High level components' elements



The overall structure, at high level, is made of three main components and their interaction. The red component refers to the tools the individual user needs to interface with Data4Help System, it communicates with the Application Server that is part of the blue component charged with the Server Architecture. This is composed by a Database Server that includes the DBMS, a Mail Server which means to exchange SMTP messages with other Mail Servers (external to the system), an Application Server communicating with any other element in the Server Architecture, a Web Server and a Proxy (meant to dispatch requests to Application and Web Servers). The proxy links the Server Architecture with the green component charged with the interaction with the third party user that takes place through Data4Help Web Page.

2.1.2 Interaction between Server Architecture and Individual User

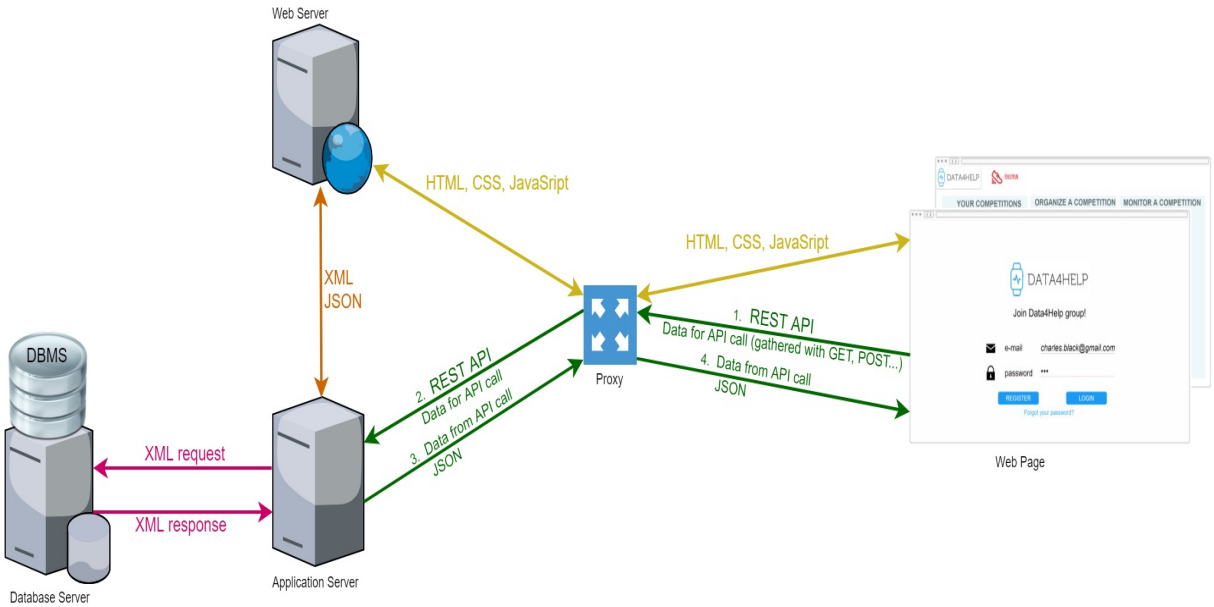
Figure 2.2: Mobile App connections



The Mobile App receives data from the smartwatch, exchanges informations with the Individual User and communicates with the Application Server: at different levels are specified protocols that are supposed to be used.

2.1.3 Interaction between Server Architecture and Third-Party User

Figure 2.3: Connection between Web page and Servers



The browser hosting the Web Page needs to communicate both with the Web Server and the Application Server. The Web Server can easily handle and exchange HTML, CSS and JavaScript files with the client; the Application Server manages methods like GET, POST (that are present in the HTML file) receiving a

REST API call and forwarding data in JSON format. Data to forward are provided by the Database Server which includes the DBMS: a request in XML is sent by the Application Server, the DBMS processes the request and extracts data from the database that are sent back to the Application Server in a XML file. To establish a communication channel between the Application Server and the Web Server is not a necessity, however it provides an alternative to REST API: developers are up to decide to implement them both or to keep the REST API alone.

2.2 Component view

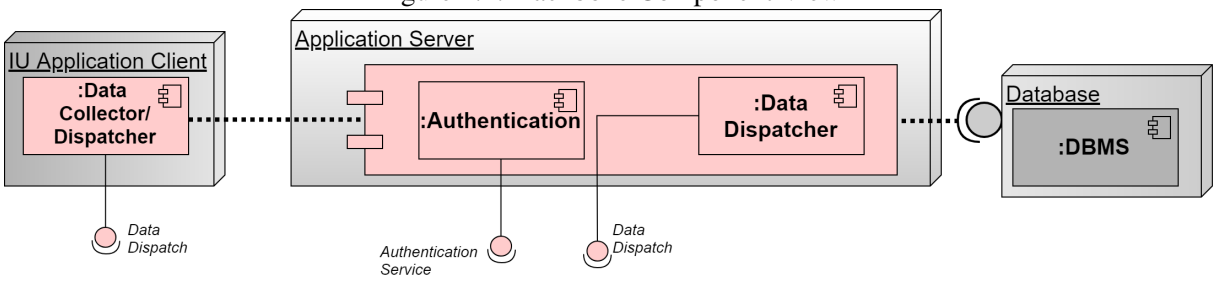
The system is divided in four subsystem:

- **Backbone**: The core of the system whose services are used from all othe subsystems
- **Data4Help**
- **AutomatedSOS**
- **Track4Run**

The last three are divided on the Application server in a router, to handle requests, and a module, connected to the DBMS, containing all other components of the subsystem.

2.2.1 Backbone

Figure 2.4: Backbone Component View



This is the backbone of the system: collects the data on the device, keep it syncronized though the system and provide functionality to receive Live Data and to access to Stored Data; furthermore provide functionality concernig authentication.

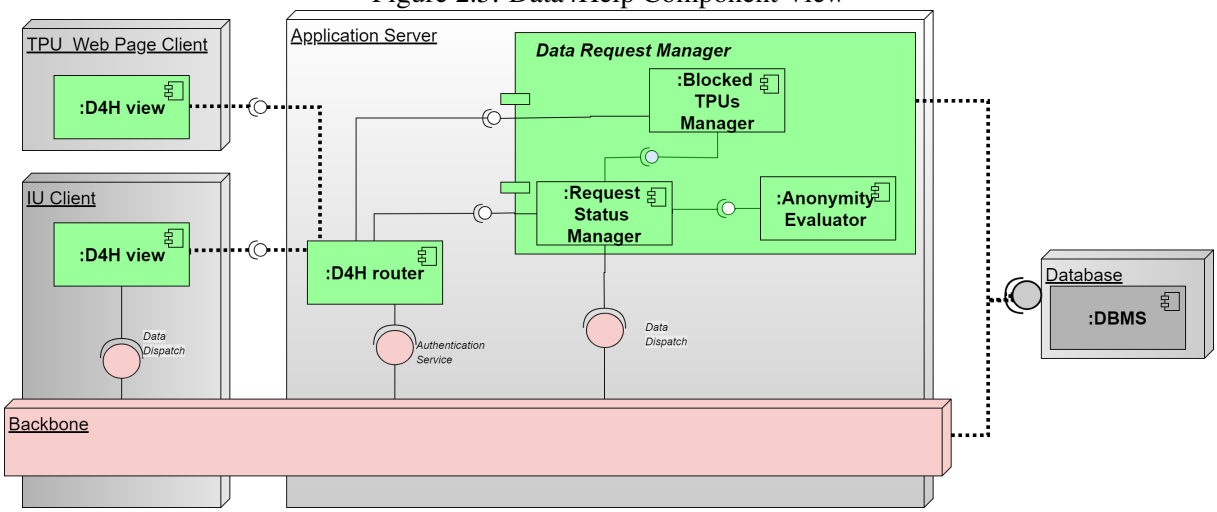
Data collector/dispatcher Allow subscribtion and publishes/dispatches the collected Live Data of the Individual User logged in from the device.

Autenthication Offers services related to User authentication.

Data Dispatcher Allow subscribtion and publishes/dispatches the collected Live Data of all Users. Offers the functionality to access Stored Data of all Users.

2.2.2 Data4Help

Figure 2.5: Data4Help Component View

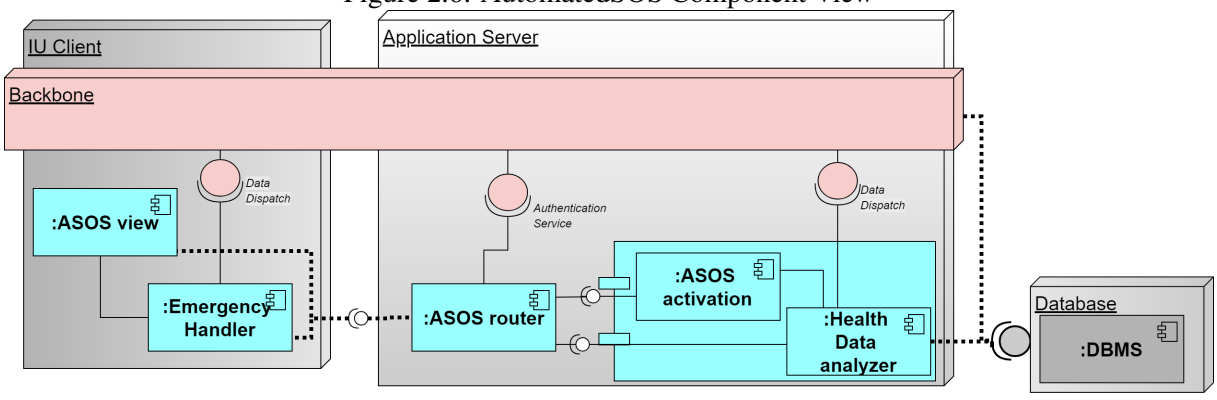


D4H router Validate the requests received from the client and dispatch them to the corresponding module or component.

Data Request Manager Provides functionality to create, approve, deny requests, block users and provide the relative data; Anonymity Evaluator is responsible to check anonymity constraints.

2.2.3 AutomatedSOS

Figure 2.6: AutomatedSOS Component View



ASOS router Validate the requests received from the client and dispatch them to the corresponding module or component.

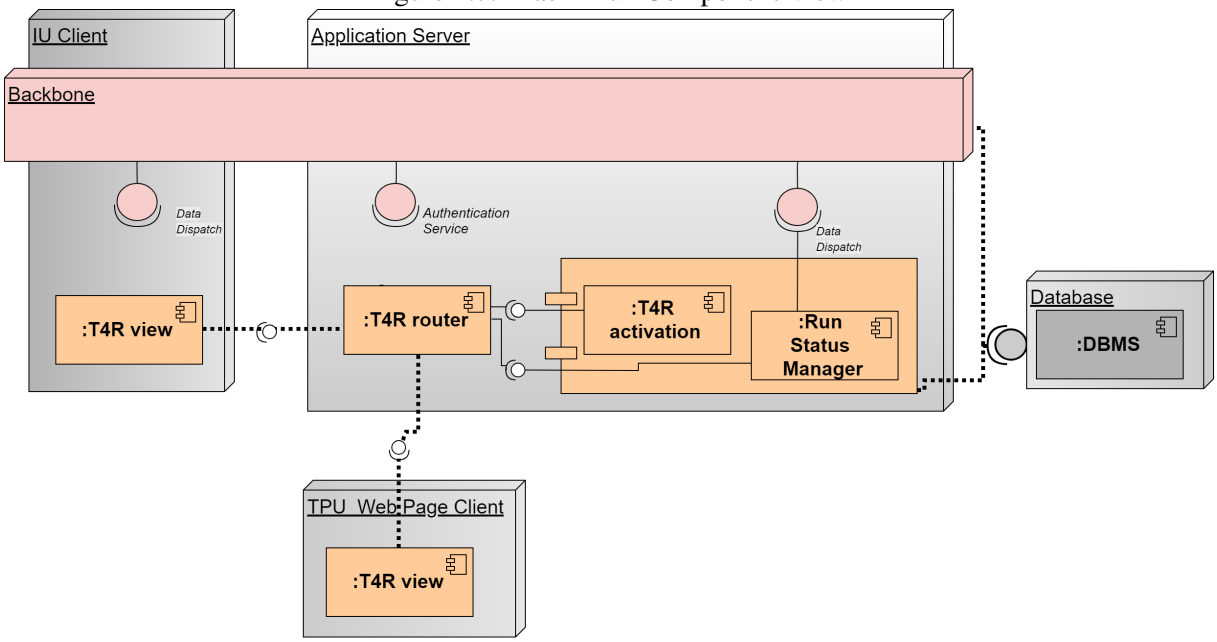
ASOS Activation Offers the functionality for the activation and deactivation of the ASOS service.

Health Data analyzer Offers functionality to extrapolate the critical health parameters for every Individual User;

Emergency Handler Responsible to handle critical health conditions based on the data published by the *Data collector/dispatcher*

2.2.4 Track4Run

Figure 2.7: Track4Run Component View



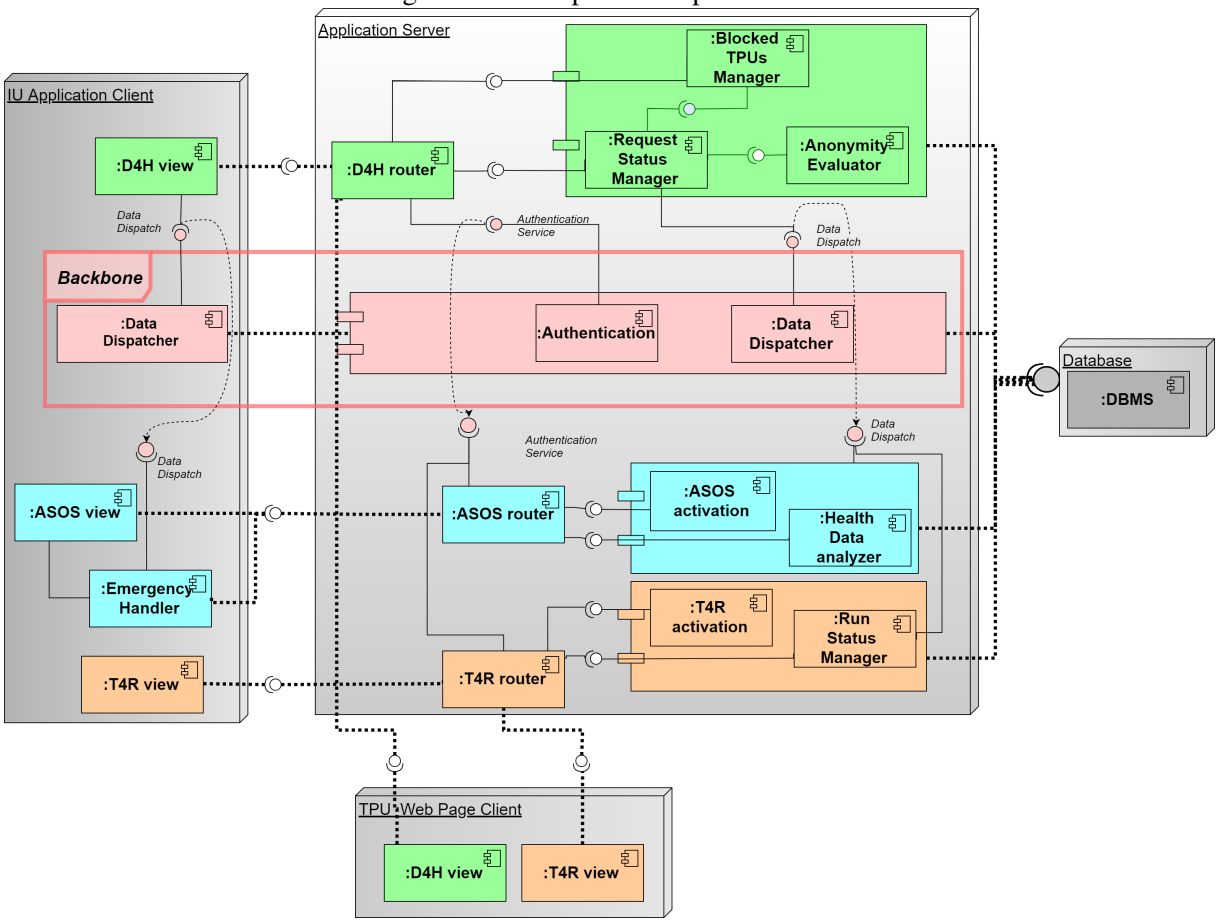
T4R router Validate the requests received from the client and dispatch them to the corresponding module or component.

T4R Activation Offers the functionality for the activation and deactivation of the T4R service.

Run Manager Provides functionality to create, cancel, enrol in and spectate runs;.

2.2.5 Full system

Figure 2.8: Complete Component View

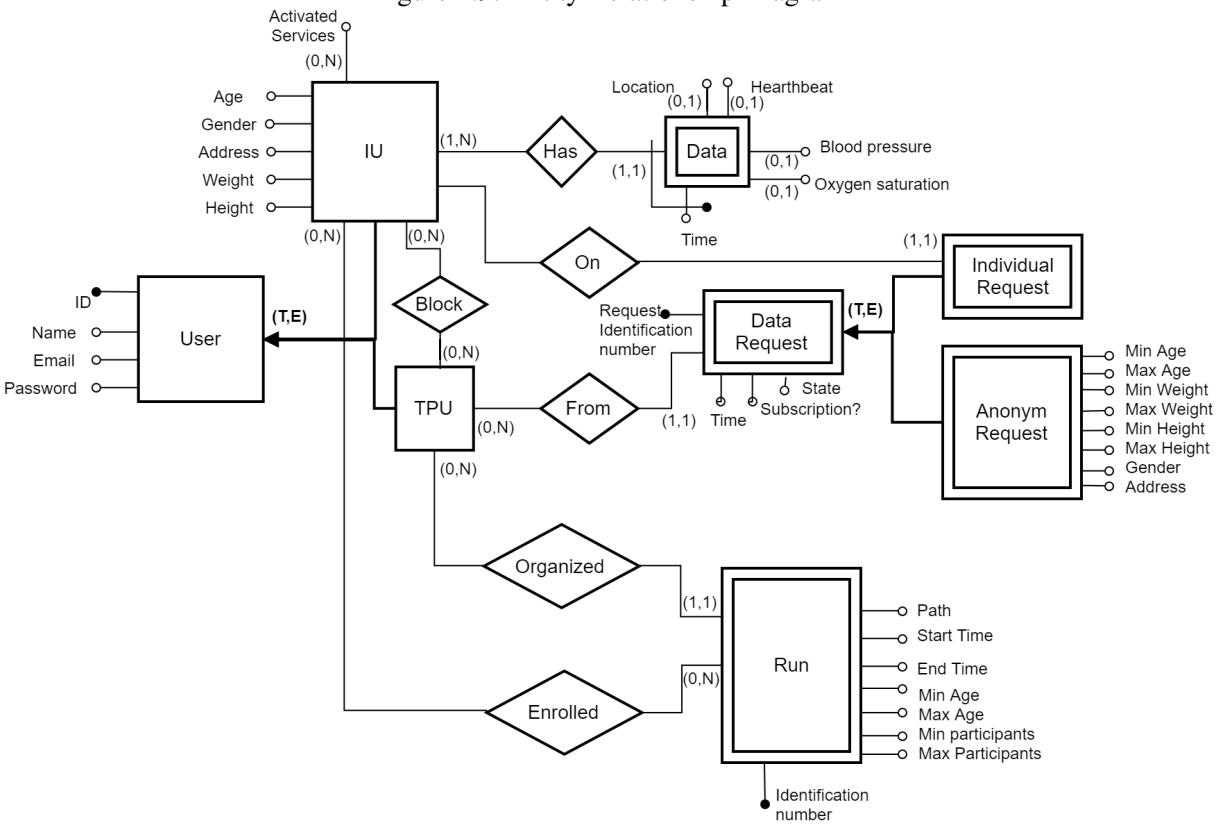


Data Managing From a more high level point of view, the backbone provides services to retrive the Individual Users data, stored or live.
This makes the red components and modules of the architecture the backbone, collecting and dispatching data, while the other subsystems can handle their unique authorization condition: D4H authorizing data dispatching based on approved requests, ASOS on the activation of the service and T4R on the enrollement in competitions.
This way all subsystem will work independently from each other.

2.2.6 Entity Relationship Diagram

The following section provides a conceptual representation of the model.

Figure 2.9: Entity Relationship Diagram



Tables

- ***User***(ID, Name, Email, Password)
- ***TPU***(ID, Name, Email, Password)
- ***IU***(ID, Name, Email, Password, Age, Gender, Address, Weight, Height)
- ***Data***(IU, Time, Location, Heartbeat, Blood pressure, Oxygen saturation)

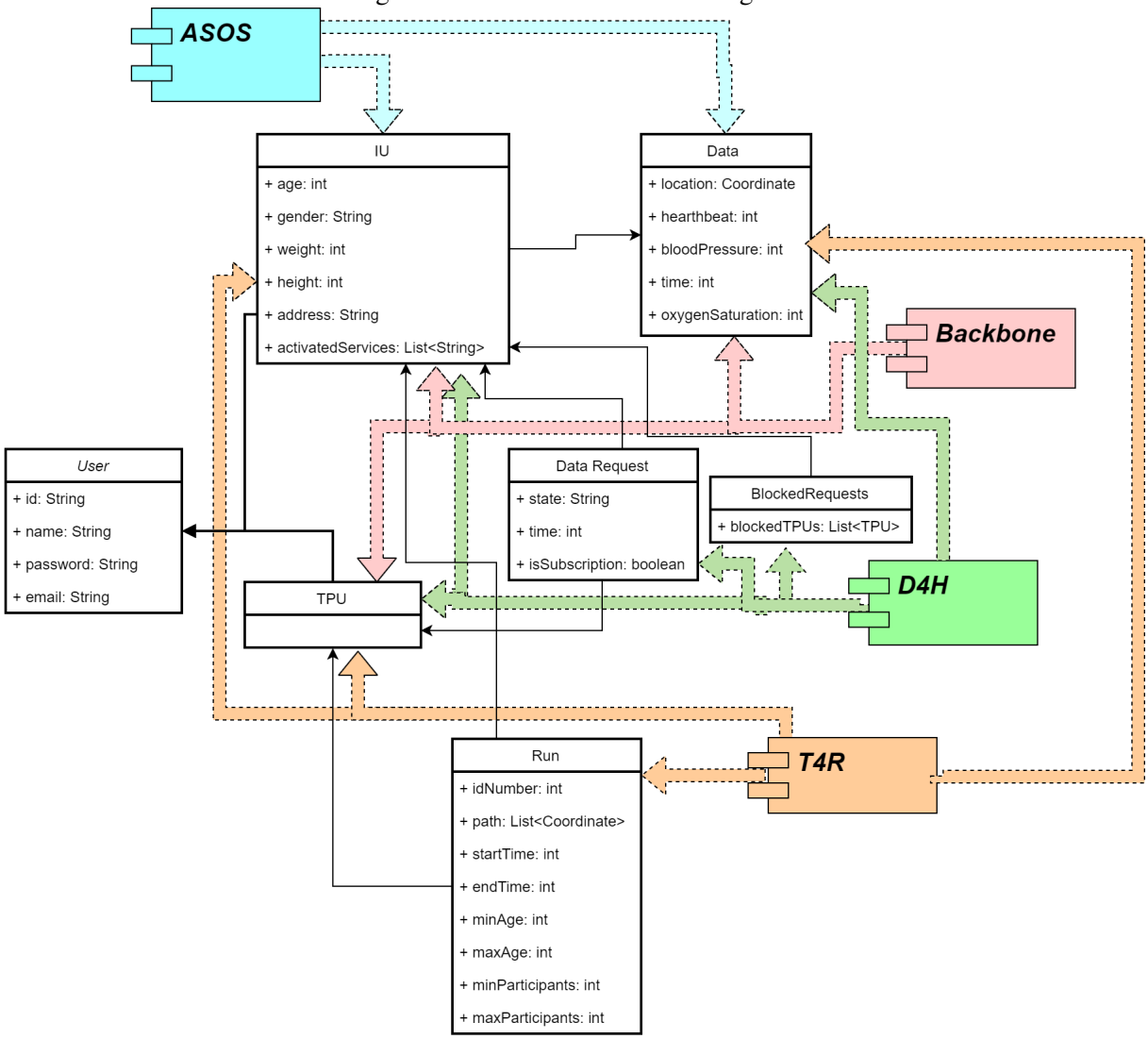
- ***Individual Request***(Request Identification Number, IU, TPU, Time, State, Subscription?)
- ***Anonym Request***(Request Identification Number, TPU, Time, State, Subscription?, Min Age, Max Age, Min Weight, Max Weight, Min Height, Max Height, Gender, Address)

- ***Run***(Identification number, TPU, IU, Path, Start Time, End Time, Min Age, Max Age, Min participants, Max Participants)

2.2.7 Model Interaction Diagram

The following diagram show a different representation of the model to better highlight its interaction with the application server. For each subsystem module that was connected to the DBMS in 2.2.5 is shown its relationship with the module.

Figure 2.10: Model Interaction Diagram

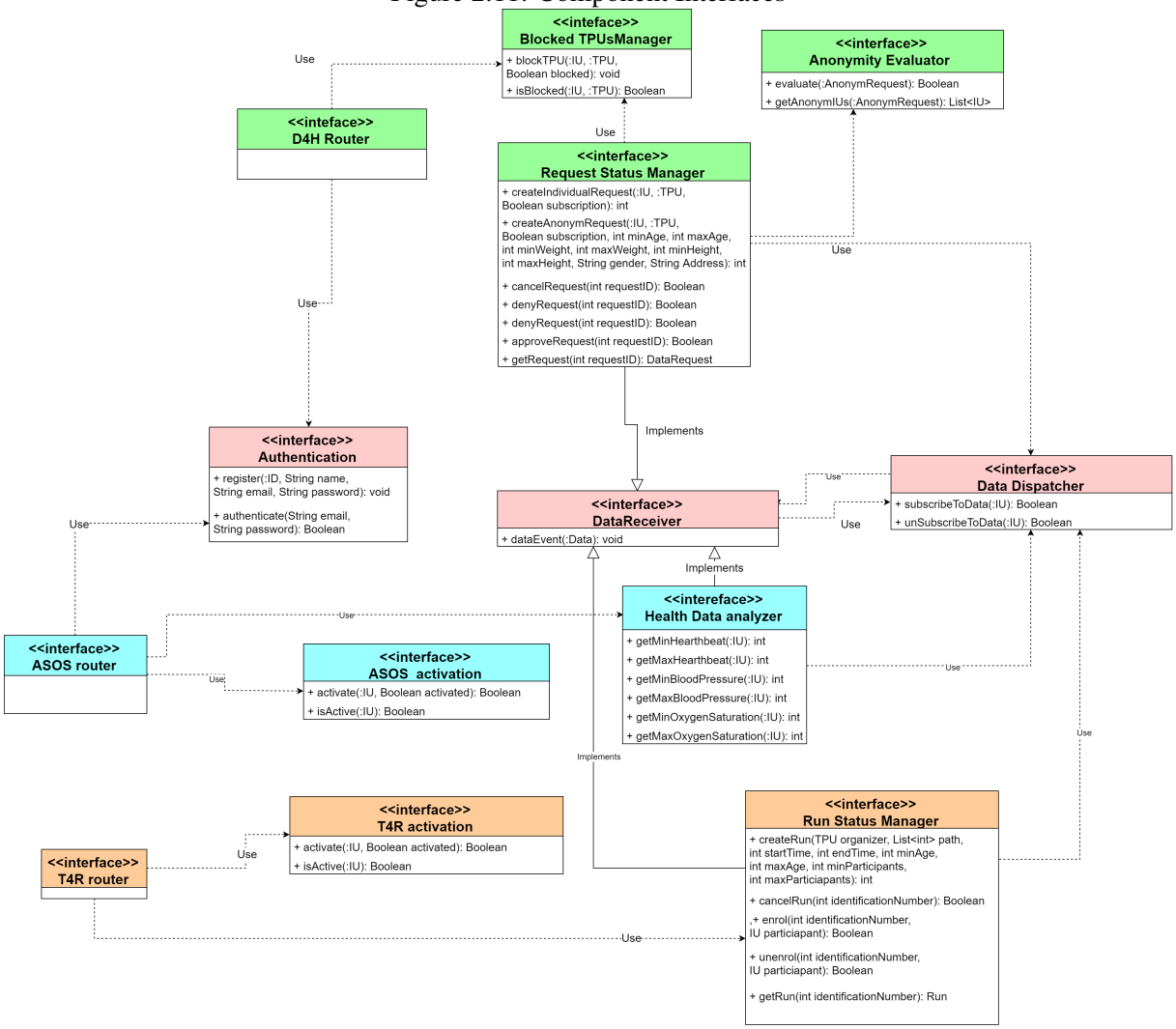


2.3 Deployment view

2.4 Runtime view

2.5 Component interfaces

Figure 2.11: Component Interfaces



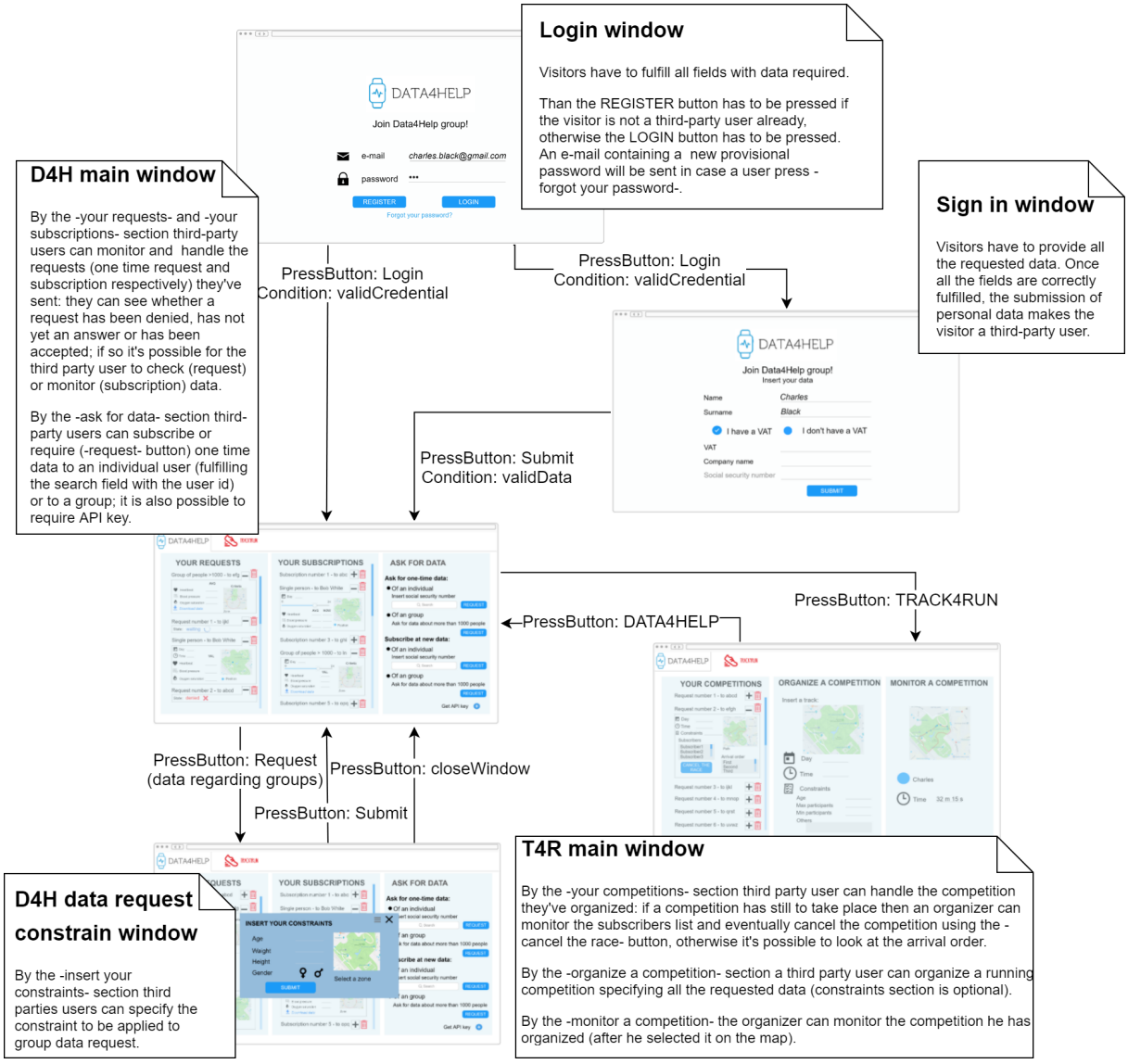
2.6 Selected architectural styles and patterns

2.7 Other design decisions

3. User Interface Design

The user interfaces mock-ups are represented in sections 3.1.1, 3.1.2 of RASD. The following UX schemes represents a complete description of the user experience. The screen -T4R unsubscribe screen- has been added and a better description of each mock-up has been provided.

Figure 3.1: Third-party user - UX graphical representation



Third-party user The scheme above represents the main desktop screens and the way -condition and action needed- how the third-party user can move trough them.

Figure 3.2: Individual user - UX graphical representation (left part)

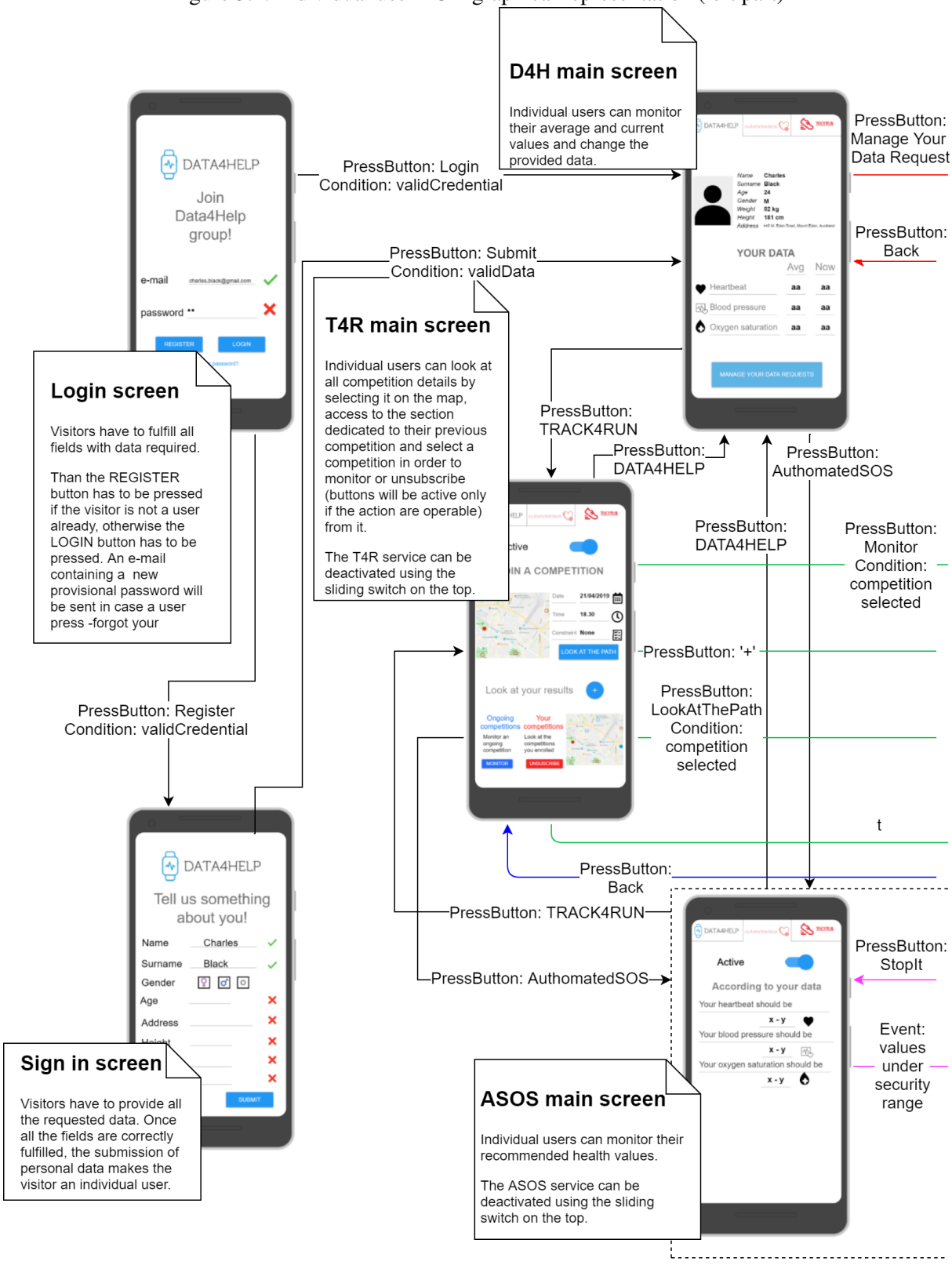
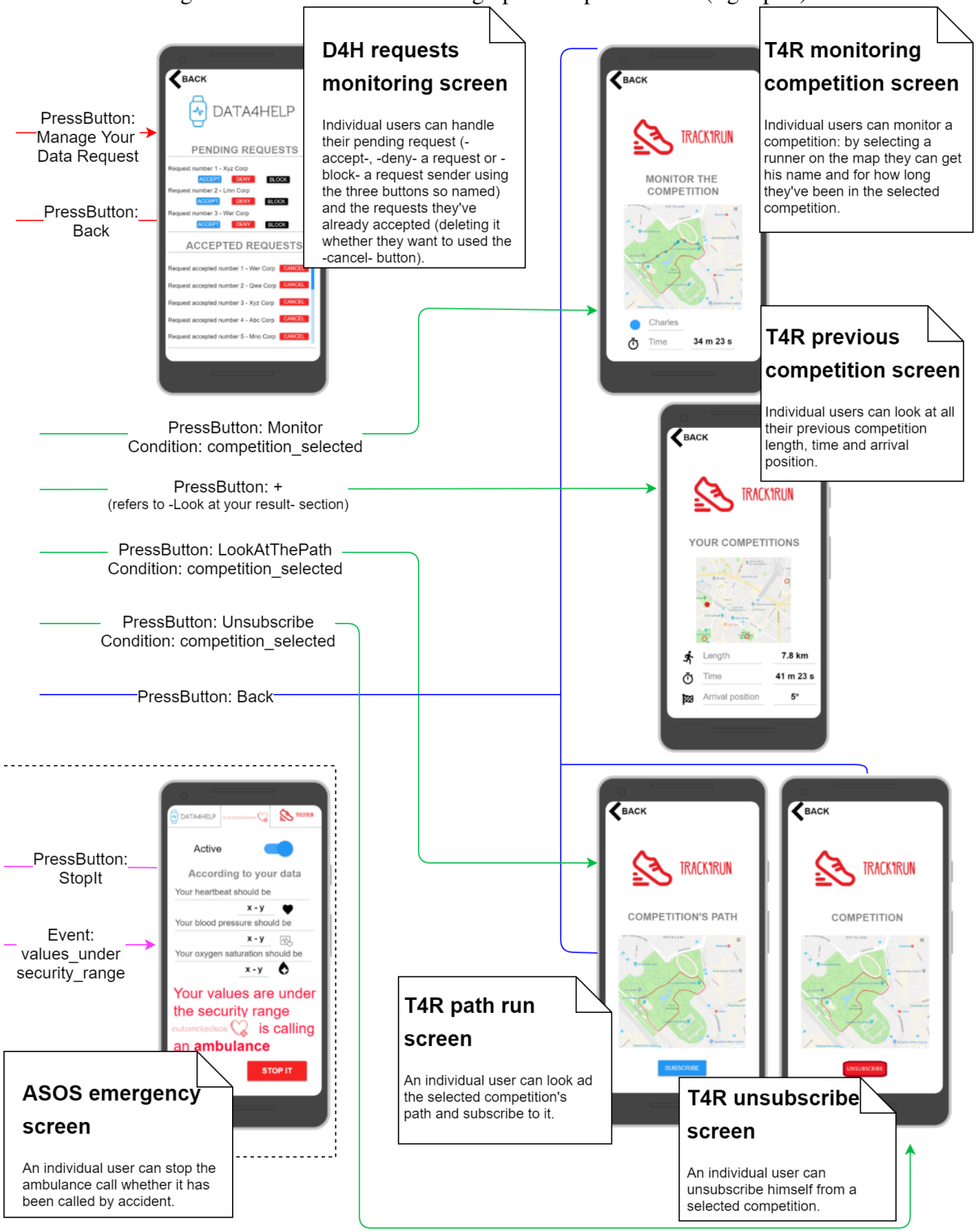


Figure 3.3: Individual user - UX graphical representation (right part)



Individual user The two schemes above represents the main mobile screens and the way -condition and action needed- how an individual user can move trough them. The scheme has been divided in two parts in order to provide a better readability.

4. Requirements Traceability

5. Implementation, Integration and Test plan

6. Effort Spent

6.1 ARGIRO' ANNA SOFIA

DATE	DESCRIPTION OF THE TASK	HOURS SPENT
27/11/18	group work	3
2/12/18	high level overview	4

6.2 BATTAGLIA GABRIELE

DATE	DESCRIPTION OF THE TASK	HOURS SPENT
27/11/18	group work	3
30/11/18	component view	4
2/12/18	model diagrams	4

6.3 CASASOLE BERNARDO

DATE	DESCRIPTION OF THE TASK	HOURS SPENT
27/11/18	group work	3
2/12/18	User interface design	4

7. References

7.1 Reference Documents

7.2 Software

- TeXWorks v0.6.2
- Umlet v14.2
- Draw.io v9.4.1
- proto.io v6.3.2.3