

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: 0.2

Date: November 30, 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

#### 1.7 Document Structure

Introduction

**Architectural Design** 

**User Interface Design** 

**Requirements Traceability** 

Implementation, Integration and Test plan

**Effort Spent** 

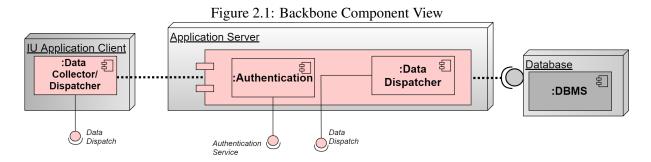
References

# 2. Architectural Design

#### 2.1 Overview

#### 2.2 Component view

#### 2.2.1 Backbone



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.

**Autenthication** Offers services related to User authentication.

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

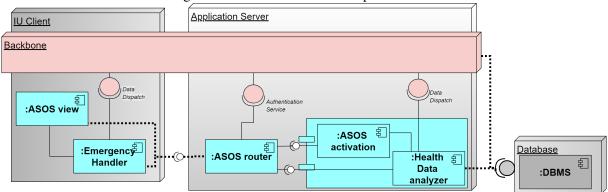
#### 2.2.2 Data4Help

**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 2.3: 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

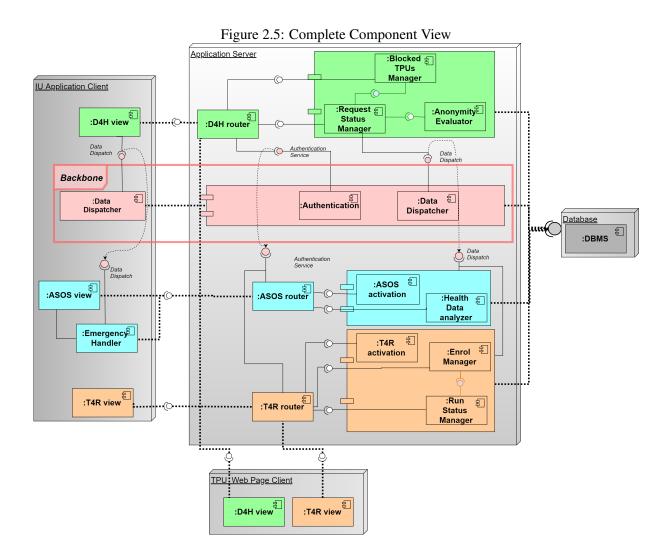
Figure 2.4: Track4Run Component View IU Client <u>Application Server</u> <u>D4H</u> 包 :T4R :Enrol activation Manager <u>Database</u> 包 :DBMS :T4R router :Run 包 Status Manager TPU Web Page Client :T4R view

*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



**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.

- 2.3 Deployment view
- 2.4 Runtime view
- 2.5 Component interfaces
- 2.6 Selected architectural styles and patterns
- 2.7 Other design decisions

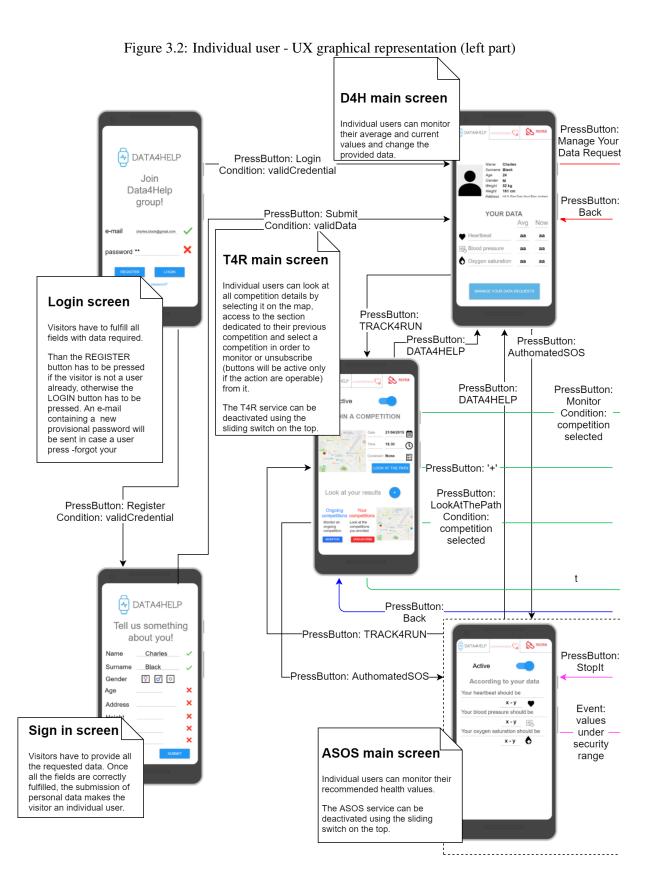
# 3. User Interface Design

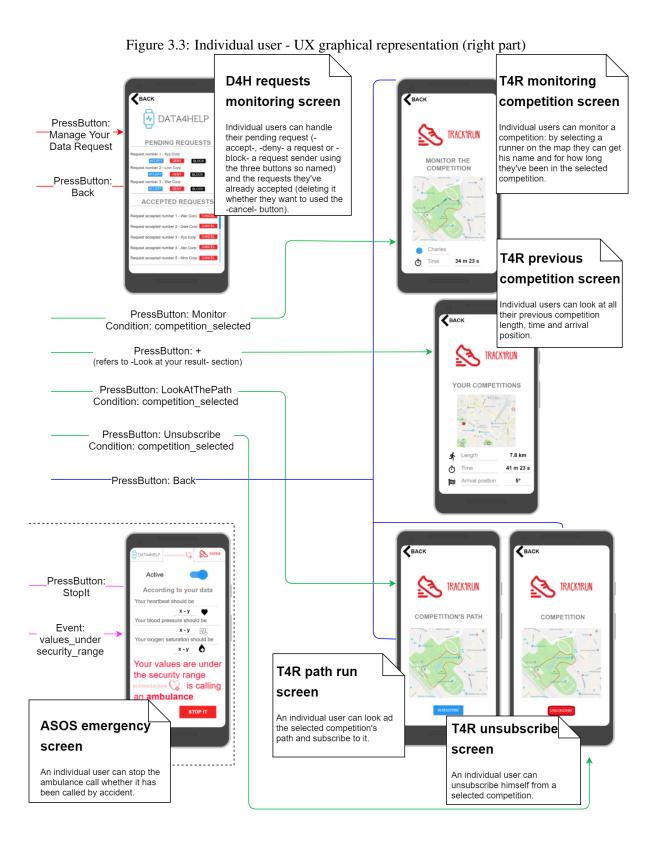
The user interfaces mock-ups are represented in sections 3.1.1, 3.1.2 of RASD. The following UX schemes represents a complete description of the user experience. The screen -T4R unsubscribe screenhas been added and a better description of each mock-up has been provided.

Login window Visitors have to fulfill all fields with data required DATA4HELP Than the REGISTER button has to be pressed if the visitor is not a third-party user already, otherwise the LOGIN button has to be pressed. An e-mail containing a new provisional D4H main window password will be sent in case a user press forgot your password-Sign in window By the -your requests- and -your subscriptions- section third-party users can monitor and handle the Visitors have to provide all the requested data. Once PressButton: Login requests (one time request and subscription respectively) they've PressButton: Login all the fields are correctly fulfilled, the submission of Condition: validCredential ondition: validCredential sent: they can see whether a request has been denied, has not personal data makes the visitor a third-party user. vet an answer or has been DATA4HELP accepted; if so it's possible for the third party user to check (request) or monitor (subscription) data By the -ask for data- section thirdparty users can subscribe or require (-request- button) one time PressButton: Submit data to an individual user (fulfilling Condition: validData the search field with the user id) or to a group; it is also possible to require API key. PressButton: TRACK4RUN -PressButton: DATA4HELP PressButton: Request PressButton: closeWindow (data regarding groups) PressButton: Submit T4R main window By the -your competitions- section third party user can handle the competition they've organized: if a competition has still to take place then an organizer can D4H data request monitor the subscribers list and eventually cancel the competition using the constrain window By the -organize a competition- section a third party user can organize a running By the -insert your constraints- section third parties users can specify the constraint to be applied to By the -monitor a competition- the organizer can monitor the competition he has organized (after he selected it on the map). group data request.

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.





*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

#### **6.2 BATTAGLIA GABRIELE**

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

#### 6.3 CASASOLE BERNARDO

DATE	DESCRIPTION OF THE TASK	HOURS SPENT
27/11/18	group work	3

# 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