

DD

Design Document

Data4Help, AutomatedSOS and Track4Run

Irene Nizzoli, Isabella Piacentini, Elio Salvini

10483798 10508831 10490058

POLITECNICO DI MILANO

Sommario

[1 Introduction 3](#_Toc531553852)

[1.1 Purpose 3](#_Toc531553853)

[1.2 Scope 3](#_Toc531553854)

[1.3 Acronyms, Abbreviations 3](#_Toc531553855)

[1.3.1 Acronyms 3](#_Toc531553856)

[1.3.2 Abbreviations 4](#_Toc531553857)

[1.4 Document Structure 4](#_Toc531553858)

[2. Architectural Design 5](#_Toc531553859)

[2.1 Overview 5](#_Toc531553860)

[2.3 Deployment view 6](#_Toc531553861)

# Introduction

## Purpose

The purpose of this document is to give a more detailed description of the architecture of Data4Help system. It will include the illustration of specific components and design choices that will guide the developers during implementation, integration and testing.

Overall this document outlines these elements:

* The high-level architecture
* The main components and their respective interfaces
* The runtime behaviour
* The design patterns
* The algorithm design of the most critical parts of the application
* Implementation plan
* Integration plan
* Testing plan

## Scope

The aim of TrackMe is to provide a service to either companies in need of data for business researches or individuals for more personal reasons. The main functions of Data4Help are managing requests from different users and saving and protecting a great quantity of data. A registration will be needed to provide clients a personalized experience, both by giving them the results of their requests and by showing monitored user their private health status data. The project is extended by AutomatedSOS that monitors the data of subscribed users and contacts medical services in case of need. The target of this system are elderly people who lives alone or are simply worried about their health conditions. This will need a 24/7 reliability of the application. Finally, Track4Run allows run organizers to create new races, tracks runners and show their position on the map to all possible spectators.

## 1.3 Acronyms, Abbreviations

### 1.3.1 Acronyms

* RASD: Requirement Analysis and Specification Document
* API: Application Programming Interface
* GPS: Global Positioning System
* DAD: Data Acquisition Device
* CF: “Codice Fiscale”
* SSN: Social Security Number
* DD: Design Document
* MVC: Model View Controller
* GUI: Graphical User Interface
* DB: Database
* DBMS: Database Management System

### 1.3.2 Abbreviations

* [Gn]: nth goal
* [Rn]: nth functional requirement

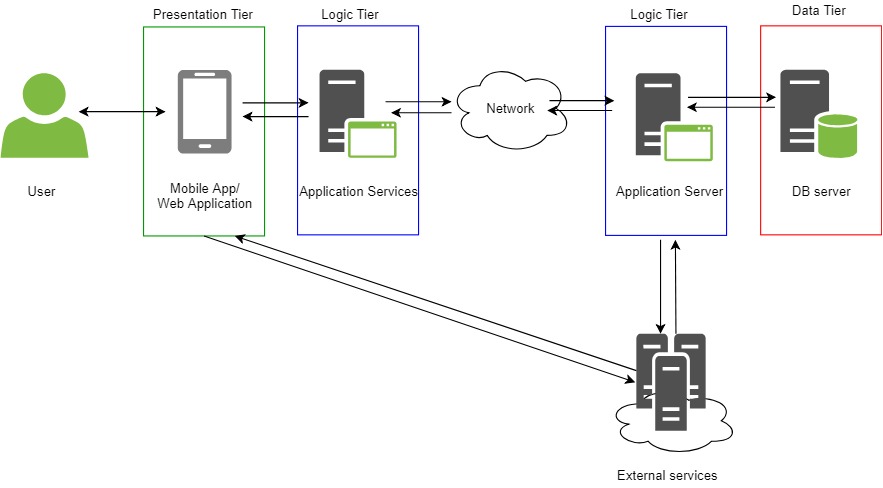
## 1.4 Document Structure

1. **Introduction:** this chapter contains the purpose and the scope of the design document. There’s also a list of the acronyms and the abbreviation that will be used in the document in order to make it more comprehensible.
2. **Architectural Design:** this sectiongives a general idea of the architecture including the three most important views: component, deployment and runtime. The interaction of the component interfaces and some architectural styles and patterns are also contained here.
3. **User Interface Design:** this chapter presents a reference to the mock-ups previously presented in the RASD document.
4. **Requirements traceability:** clarifies how the requirements that have been defined in the RASD map to the design elements that are defined in this document.
5. **Implementation, integration and test plan:** reveal the order in which it is intended to implement the subcomponents of the system and the order in which it is planned to integrate such subcomponents and test the integration.
6. **Effort spent:** shows the number of hours each member of the group spent for every chapter of the document.
7. **References:** presents the external documents used in the construction of the DD document.

## 2. Architectural Design

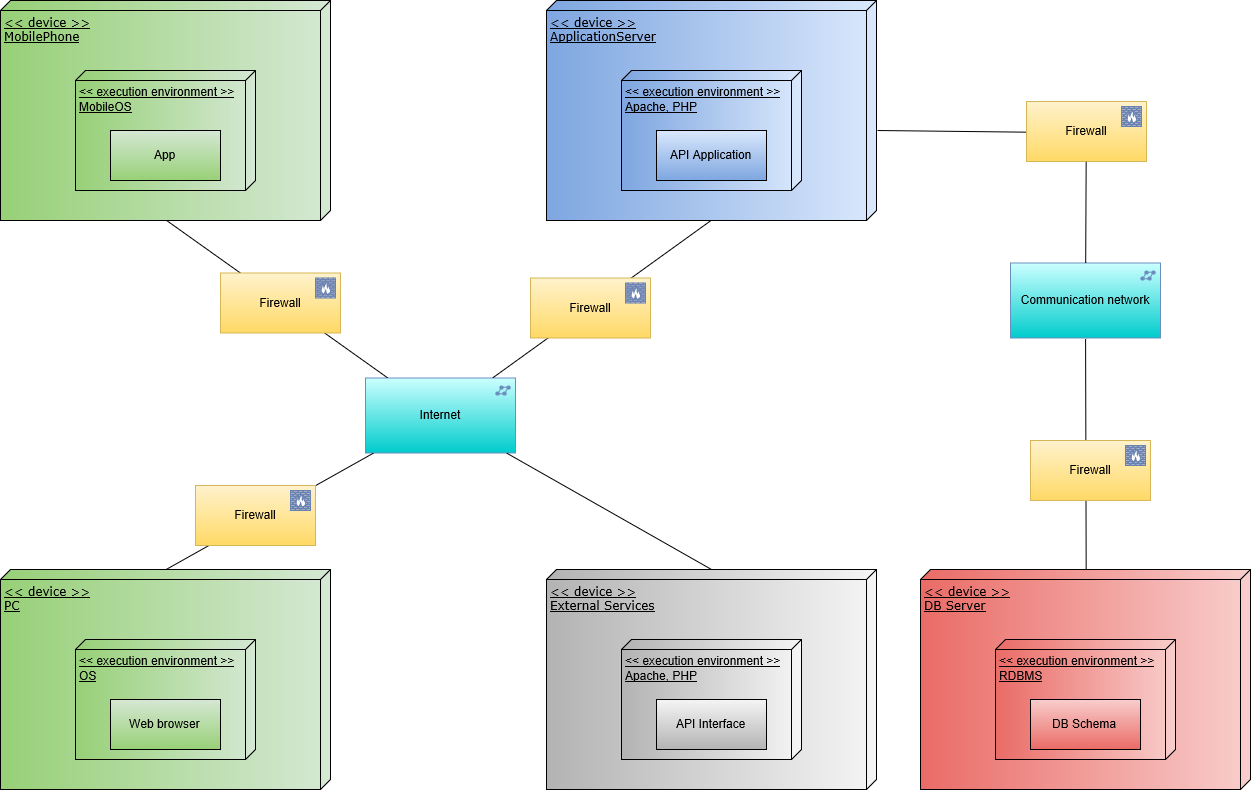
### 2.1 Overview

There are three separated layers in the flow in which the process of the application architecture design is executed:



* *Presentation tier*: This is the top-most level of the application. It consists in a user interface (view) and communicates with the application services and the external services. The connection with the external services is useful because the map (for Track4Run) can be directly obtained by the mobile app/web application.
* *Logic tier*: This layer controls the functionality of the application. The logic of the application is mostly stored on the server side, only a small part, such as the logic to detect if the individual’s data are below the threshold (just for the AutomatedSOS service), is located on the client side. The choice distributed logic architecture allows better performances in contacting the external medical services in case of emergency.
* *Data tier*: The data tier comprises of the database/data storage system and data access layer. Here the information is stored persistently and can be retrieved at any time.

### 2.3 Deployment view



This deployment diagram shows the architecture of the system from a physical view point. Also, the distribution of software among the different hardware nodes is presented. The main nodes involved in the system are:

* *Client nodes*:
  + *Mobile phone*, users can access to Data4Help services through a mobile application that communicates with TrackMe’s server or directly with other external services (such as the street map service for Track4Run).
  + *PC* or other devices able to access to web services, Data4Help services are also available through its website, accessible via web browser.
* *Application server*, most of application logic belongs to this node. The application server communication with clients is based on the client-server pattern. This node also communicates with the system’s DB that stores all users’ data. The application server exploits external services API to provide a complete service to its clients.
* *DB server*, this node’s aim is to store all users’ data, from general account information to health status and location data. This node is directly accessed only by the application server.
* *External services*, client nodes and application server take advantage of this services to provide all AutomatedSOS and Track4Run functionalities.