# Trackme DD

## DESIGN DOCUMENT (DD)

## Release date: 10/12/18

Version 0.1

Table of content

[Trackme DD 1](#_Toc531013661)

[DESIGN DOCUMENT (DD) 1](#_Toc531013662)

[1. Introduction 2](#_Toc531013663)

[1.1. Purpose 2](#_Toc531013664)

[1.2. Scope 2](#_Toc531013665)

[1.3. Definitions, Acronyms, Abbrevations 2](#_Toc531013666)

[1.4. Revision History 2](#_Toc531013667)

[1.5. Reference Documents 2](#_Toc531013668)

[1.6. Document Structure 2](#_Toc531013669)

[2. Architectural Design 2](#_Toc531013670)

[2.1. Overview 2](#_Toc531013671)

[2.2. Component view 2](#_Toc531013672)

[2.3. Deployment view 2](#_Toc531013673)

[2.4. Runtime view 2](#_Toc531013674)

[2.5. Component interfaces 2](#_Toc531013675)

[2.6. Selected architectural styles and patterns 2](#_Toc531013676)

[2.7. Other design decisions 2](#_Toc531013677)

[3. User Interface Design 2](#_Toc531013678)

[4. Requirements Traceability 2](#_Toc531013679)

[5. Implementation, Integration and Test Plan 2](#_Toc531013680)

[6. Effort Spent 2](#_Toc531013681)

# Introduction

## Purpose

This is the Design Document of the TrackMe project. Its purpose is to present more technical details about the system to be implemented. It’s valid both for *Data4Help* and *AutomatedSOS:* when in some specific parts and diagram some differentiations must be done, it’s always explicitly indicated. Programmers and developers are supposed to read carefully this document and respect its principles.

## Scope

## Definitions, Acronyms, Abbrevations

* SOA: *Service Oriented Architecture*
* GUI: *Graphical User Interface*
* API: *Application Program Interface*

## Revision History

This is the first release of this document.

## Reference Documents

## Document Structure

### Introduction

In the first chapter there are some useful guidelines for the reader to understand properly the overall project and the document itself (i.e. its purpose, the notations used hereunder and so on).

### Architectural Design

Chapter 2 contains all the needed models, views and diagrams about the architecture of the application, and the explanations of all choices that have been made

### User Interface Design

Chapter 3 presents some interesting mock-ups, integrating the interfaces already presented in the RASD.

### Requirements Traceability

Chapter 4 puts in correlation all the components defined in this document with the requirements established in the RASD.

### Implementation, Integration and Test Plan

Chapter 5 points out all the details about how to implement and test all the elements, and how to integrate them to build the entire system and make sure everything works correctly.

### Effort Spent

In chapter 6 there are some data about the effort spent by each member of the group in elaborating this document.

# Architectural Design

## Overview

All the system is based on the SOA (*Service Oriented Architecture)* paradigm: every functionality to provide is associated to a service, and a service is implemented in a module. All the modules interact each other within a defined communication system, using the interfaces that every service exposes to the others. Because of that, there is huge decoupling between all the services, so that the ones that regard *AutomatedSOS* can be easily added to the system.

The architectural design of our application contemplates three logical layers:

* The **presentation laye**r handles the communication with the final users of the application, so it’s basically the GUI.
* The **application layer** contains all the services that we have already mentioned and is accessible from the presentation layer via some API.
* The **data layer** is concerned in storing data.

There is no strict correspondence between the logical layers and the physical tiers where the application is deployed, since some services are located on the app running on the user’s device, which therefore contains a part of the application layer in addition to the presentation layer. In details,

* The **device tier** could be represented by a smartphone connected to a wearable device or, in alternative, by a

## Component view

## Deployment view

## Runtime view

## Component interfaces

## Selected architectural styles and patterns

## Other design decisions

# User Interface Design

# Requirements Traceability

# Implementation, Integration and Test Plan

# Effort Spent