



# POLITECNICO MILANO 1863

TrackMe  
Software Engineering 2 Project  
*ITD Document*

Stefano Martina, Alessandro Nichelini, Francesco Peressini

A.Y. 2018/2019  
Version 1.0.0

December 31, 2018

# Contents

<b>1</b>	<b>Introduction and scope</b>	<b>3</b>
1.1	Introduction . . . . .	3
1.2	scope . . . . .	3
<b>2</b>	<b>Functionality implemented</b>	<b>3</b>
<b>3</b>	<b>Adopted frameworks</b>	<b>3</b>
3.1	Adopted programming language . . . . .	3
3.2	Middleware adopted . . . . .	3
3.2.1	Frameworks . . . . .	3
3.3	Libraries . . . . .	3
3.4	API used . . . . .	4
<b>4</b>	<b>Source code structure</b>	<b>4</b>
<b>5</b>	<b>Testing</b>	<b>4</b>
<b>6</b>	<b>Installation instructions</b>	<b>4</b>
6.1	iOS mobile app installation instructions . . . . .	4
6.2	Python backend installation instructions . . . . .	4

# 1 Introduction and scope

## 1.1 Introduction

## 1.2 scope

# 2 Functionality implemented

# 3 Adopted frameworks

## 3.1 Adopted programming language

The project has been developed using two programming language.

- iOS application has been developed in the "new" programming language by Apple: Swift (target version: Swift 4.2).
- Backend software has been developed using Python (target version: Python 3.6)

The team has chosen Swift over Objective-C because of the effort Apple is putting in its development. Moreover Swift has been thought to work with iOS in a more specific way than Objective-C.

Python has been chosen as the programming language for the backend because most of the team members has previous proficiency in using it.

## 3.2 Middleware adopted

### 3.2.1 Frameworks

For the backend: Flask framework has been adopted. Flask is a web development framework that let the building of easy and light web environments. It has been used to both build the API infrastructure and the third parties' web interface.

## 3.3 Libraries

For Apple Swift in iOS the following external libraries has been adopted:

- Alamofire by Alamofire (<https://github.com/Alamofire/Alamofire>) to better handle web request client side.
- SwiftyJSON by SwiftyJSON (<https://github.com/SwiftyJSON/SwiftyJSON>) to better handle JSON data in Swift.

For the backend, the following Python libraries has been adopted:

- Flask: the main library of the previously described framework Flask.

- Flask HTTP Auth: an extension of Flask framework to implemented basic HTTP auth directly in the same environments.
- MySQLConnector: this library is necessary to handle communication with a SQL database.
- The following standard libraries were also used:
  - Python-Sys
  - Python-Secret
  - Python-PPrint
  - Python-Collections
  - Python-Json

### 3.4 API used

The project has not used external API than the ones directly developed in-house.

## 4 Source code structure

## 5 Testing

## 6 Installation instructions

### 6.1 iOS mobile app installation instructions

### 6.2 Python backend installation instructions

Python backend code should run on each Python 3.x distribution. More in details the target version in which the whole project has been devolped is Python 3.6.

Once installed Python, it's recommended to build a Python virtual environment to run the code (you can check the official Python documentation here: <https://packaging.python.org/guides/installing-using-pip-and-virtualenv/>). The following libraries have to be installed (Python-pip is the preferred method to install these packages):

- Flask (target version: 1.0.2)
- Flask-HTTPAuth (target version: 3.2.4)
- mysql-connector (target version: 2.1.6)
- mysql-connector-python-rf (target version: 2.2.2)

Once the virtual environment has been built and activated, and the repository has been closed, it's enough to call the python intrepeter from a console to run the backend server: the main file is: *Api.py* located in *Path/to/repository/Server/*