



## Relazione finale

Gruppo VarTmp7 - Progetto Stalker vartmp7@gmail.com

### Informazioni sul documento

<b>Versione</b>	1.0.0-R1
<b>Approvatore</b>	Riccardo Tassetto
<b>Redattori</b>	Xiaowei Wen Claudia Zattara
<b>Verificatori</b>	Marco Ferrati Lorenzo Taschin
<b>Uso</b>	Esterno
<b>Distribuzione</b>	Prof. Vardanega Tullio Prof. Cardin Riccardo <i>Imola Informatica S.p.A</i> VarTmp7

### Descrizione

Questo documento contiene la relazione finale richiesta dal proponente e redatta dal gruppo VarTmp7

## Indice

<b>1</b>	<b>Introduzione</b>	<b>4</b>
1.1	Scopo del documento	4
1.2	Struttura documento	4
<b>2</b>	<b>Applicazione Android</b>	<b>5</b>
2.1	Scelte implementative	5
2.1.1	Scelta version SDK	5
2.1.2	Scelta tecnologie per il tracciamento	5
2.2	Report test	5
2.2.1	Test di precisione GPS	5
2.2.2	Test unitari	5
2.3	Problemi aperti	6
<b>3</b>	<b>Backend</b>	<b>7</b>
3.1	Scelte implementative	7
3.1.1	Estensioni utilizzate	7
3.1.2	Gerarchia degli amministratori	7
3.1.3	Autenticazione	7
3.2	Report test	8
3.2.1	Test di carico	8
3.2.1.1	Numero di utente	8
3.2.1.2	Richieste al secondo	8
3.2.1.3	Tempo di risposta	8
3.3	Problemi aperti	9
<b>4</b>	<b>Webapp</b>	<b>10</b>
4.1	Scelte implementative	10
4.2	Report test	10
4.2.1	Test unitari	10
4.3	Problemi aperti	10

## Elenco delle figure

1	Numero di utenti. . . . .	8
2	Richieste per secondo. . . . .	8
3	Tempo di risposta espresso in ms. . . . .	9

## Registro delle modifiche

Versione	Data	Descrizione	Nominativo	Ruolo
1.0.0-R1	2020-05-14	Versione 1.0.0-R1 approvata	Riccardo Tassetto	<i>Responsabile</i>
1.0.0-INC-15	2020-05-11	Verificata §3. Documento pronto per l'approvazione.	Xiaowei Wen	<i>Verificatore</i>
0.2.0-INC-15	2020-05-11	Stesura §3	Marco Ferrati	<i>Redattore</i>
0.2.0-INC-14	2020-05-06	Verifica §4	Riccardo Tassetto	<i>Verificatore</i>
0.1.0-INC-14	2020-05-06	Stesura §4	Claudia Zattara	<i>Redattore</i>
0.1.0-INC-14	2020-05-05	Verifica §1 e §2	Lorenzo Taschin	<i>Verificatore</i>
0.0.0-INC-14	2020-05-05	Stesura §1 e §2	Xiaowei Wen	<i>Redattore</i>
0.0.0	2019-11-24	Creazione scheletro del documento.	Claudia Zattara	<i>Redattore</i>

## 1 Introduzione

### 1.1 Scopo del documento

In questo documento, richiesto dal proponente, vengono esplicate le scelte implementative adottate dal gruppo **VarTmp7** per sviluppare il capitolato denominato **Stalker**.

### 1.2 Struttura documento

Il documento è suddiviso in tre sezioni, una per ogni componente del sistema Stalker. I componenti sono:

- Applicazione Android;
- backend;
- WebApplication.

Ognuna delle tre sezioni è a sua volta suddivisa in:

- scelte implementative;
- report test;
- problemi aperti sviluppi futuri.

## 2 Applicazione Android

### 2.1 Scelte implementative

Per motivazione delle scelte di architetture, tecnologie e design pattern consultare il *Manuale sviluppatore\_v.2.0.0-R1*.

#### 2.1.1 Scelta version SDK

Il gruppo dopo un attento studio e ricerca ha deciso di utilizzare la versione **Oreo**. Sono state scelte le seguenti versioni di Sdk:

- compileSdkVersion: 26;
- minSdkVersion: 26;
- targetSdkVersion: 26.

È stata scelta questa versione di sdk poiché offre le API LocationServices che permette di delegare la gestione a basso livello del tracciamento. Per la gestione a basso livello si intende l'utilizzo dei diversi provider e diversi metodi per determinare la coordinate. Inoltre, la versione **Oreo** è supportato dal 72.26% dei dispositivi Android secondo il portale *StatCounter.com*.

#### 2.1.2 Scelta tecnologie per il tracciamento

Abbiamo scelto di utilizzare le API di *Fused Location Provider* per rilevare la posizione del dispositivo dell'utente, queste API permettono di effettuare le richieste di aggiornamento in pochi semplici chiamate ai metodi, e inoltre permette di fornire una serie di parametri che definiscono la precisione, l'intervallo minimo, l'intervallo massimo, la distanza minima che deve avvenire tra un aggiornamento e l'altro.

La precisione del tracciamento è garantita dalle API di Fused Location Service ed è impostabile a livello di codice sorgente, il gruppo ha scelto di utilizzare diverse precisioni in base alla distanza del dispositivo dal luogo delle organizzazioni che sta tracciando più vicino, nel senso che, in base a questa distanza verranno modificati i parametri per ricevere l'aggiornamento delle posizioni. In questo modo, si modifica la frequenza con il quale attivare i sensori GPS, evitando il consumo eccessivo di batteria.

### 2.2 Report test

#### 2.2.1 Test di precisione GPS

La precisione del tracciamento è garantito dalle API di Fused Location Service ed è impostabile a livello di codice sorgente. Il gruppo ha scelto di utilizzare diverse precisioni in base alla distanza del dispositivo. La precisione massima possibile può arrivare a due metri, ma per non consumare eccessiva batteria è stato scelto di non utilizzare questa precisione.

#### 2.2.2 Test unitari

Sono stati eseguiti tutti i test unitari, con il code coverage del **100%** per la parte del test non instrumented. Mentre per i test delle classi che richiedono l'instrumentazione non è stato possibile coprire la totalità del codice, tuttavia è stata raggiunta una code coverage pari al **85%**, che è un valore accettabile secondo gli obiettivi di qualità stabilite per il progetto.

### 2.3 Problemi aperti

Attualmente non sono stati rilevati problemi o malfunzionamenti nell'applicazione se non il fatto della grafica migliorabile. Mentre come sviluppi futuri si potrebbero applicare le seguenti migliorie:

- nella schermata del tracciamento visualizzare una mappa con al centro la posizione attuale, e i luoghi delle organizzazioni con il tracciamento attivo;
- si potrebbero creare dei "widget" ovvero una miniatura dell'applicazione che si possono posizionare nella Home screen;
- aumentare le modalità per filtrare la cronologia dei tracciamenti qualora ci siano;
- salvare in locale una copia della cronologia dei tracciamenti in modo da evitare di effettuare le richieste al server;

## 3 Backend

### 3.1 Scelte implementative

Per realizzare il backend è stato scelto di utilizzare il linguaggio Python 3.7.7 in particolare il web-framework Flask 1.1.1.

Basandoci su Flask abbiamo utilizzato diverse estensioni che hanno permesso di facilitarci il lavoro potendo così concentrarci su altri aspetti del sistema come per esempio la configurazione per il funzionamento su Kubernetes/Google Kubernetes Engine.

#### 3.1.1 Estensioni utilizzate

- Flask-CORS 3.0.8;
- Flask-JWT-Extended 3.24.1;
- Flask-Mail 0.9.1;
- Flask-RESTful 0.3.8;
- Flask-SQLAlchemy 2.4.1.

#### 3.1.2 Gerarchia degli amministratori

Il backend si occupa di gestire gli amministratori che sono di quattro tipi:

- System Administrator (SA);
- Owner Administrator (OA);
- Manager Administrator (MA);
- Watcher Administrator (WA);

Un utente amministratore può essere SA, OA o nessuno dei due; la possibilità di operare su un'organizzazione è data dalla relazione (è tale relazione che va a specificare il tipo e quindi MA o WA) tra amministratore e organizzazione; in questo modo per un utente amministratore è possibile operare su più organizzazioni.

I SA vengono inseriti nel sistema tramite linea di comando e hanno il compito di creare gli OA e approvare i luoghi delle organizzazioni.

Gli OA hanno il potere di creare organizzazioni e luoghi per le organizzazioni che gestiscono e tutti i permessi che hanno MA e WA. Inoltre possono creare nuovi amministratori di tipo MA o WA e associarli alle organizzazioni che loro gestiscono.

I MA possono operare sulle organizzazioni alle quali sono stati associati potendo modificare i dati dell'organizzazione, creare e modificare i luoghi e visualizzare i tracciamenti e i vari report. I WA possono operare sulle organizzazioni alle quali sono stati associati potendo visualizzare i tracciamenti e i vari report.

#### 3.1.3 Autenticazione

Il backend espone delle REST API usate sia dall'applicazione mobile sia dalla web application utilizzata dagli amministratori.

Per gestire l'accesso alle API vengono usati due metodi di autenticazioni in base all'API che si vuole usare.



L'applicazione mobile utilizza l'autenticazione tramite API key. Ogni organizzazione ha associato un token che può essere usato per interfacciarsi con le API. In questo modo tutte le organizzazioni possono implementare il sistema Stalker in una loro applicazione.

La web application per poter interfacciarsi con il backend deve inviare, oltre al token di un'organizzazione, anche il JWT associato all'amministratore che sta operando tramite la web application.

## 3.2 Report test

### 3.2.1 Test di carico

Per eseguire i test di carico è stata usata la suite Locust 0.14.6. Di seguito sono riportati i risultati ottenuti eseguendo il backend con un carico massimo di 300 utenti e una crescita di 20 utenti al secondo.

Il sistema sul quale è stato eseguito è un MacBook Pro 2017, i7-7820HQ 2,9GHz, 16GB di RAM e MacOS 10.15.4.

**3.2.1.1 Numero di utente** Di seguito viene riportato il grafico rappresentante la crescita degli utenti nel tempo. Una volta raggiunto il cap di 300 il numero rimane costante.

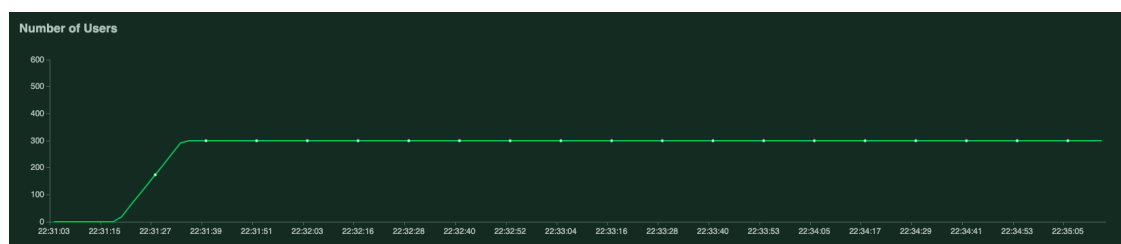


Figura 1: Numero di utenti.

**3.2.1.2 Richieste al secondo** Di seguito viene riportato il grafico rappresentante il numero di richieste effettuate al backend. In verde quelle andate a buon fine (la quasi totalità) mentre in rosso quelle fallite.

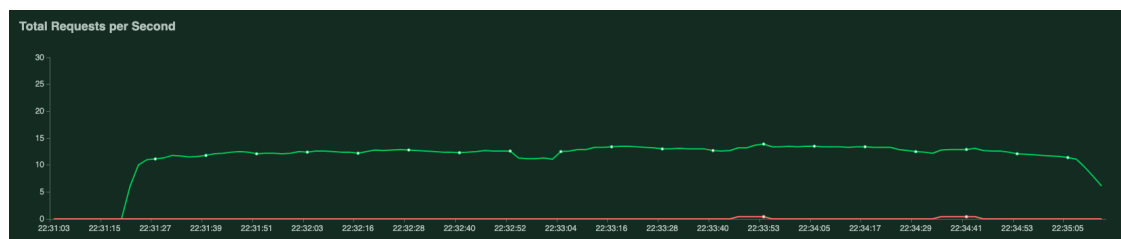


Figura 2: Richieste per secondo.

**3.2.1.3 Tempo di risposta** Di seguito viene riportato il tempo di risposta. Questo dato è il meno interessante in quanto all'applicazione non interessa dopo quanto tempo il backend abbia risposto all'aggiunta di un nuovo tracciamento (a meno di errori che comunque sono estremamente rari come si è visto dal grafico precedente).

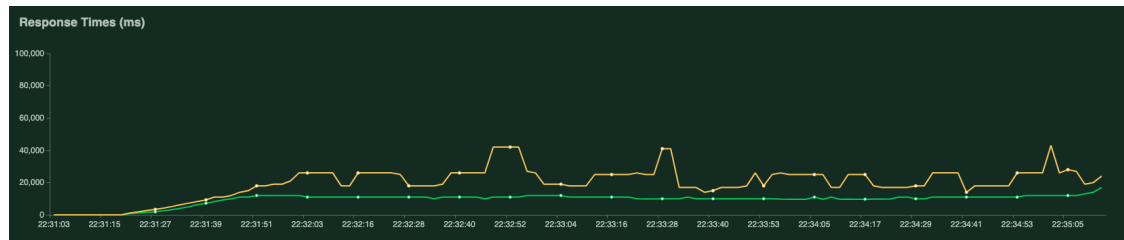


Figura 3: Tempo di risposta espresso in ms.

### 3.3 Problemi aperti

Attualmente non sono stati rilevati problemi o malfunzionamenti nel backend. Come sviluppi futuri si potrebbero implementare:

- nuovi metodi di autenticazione oltre a LDAP;

## 4 Webapp

### 4.1 Scelte implementative

Per realizzare la webapp è stato scelto di utilizzare il linguaggio Angular 9.0.7 con le librerie:

- Material 9.2.0 per lo styling grafico;
- Leaflet 1.6.0 per la visualizzazione delle mappe interattive;
- Leaflet-Geoman, plugin per la modifica delle mappe interattive.

Per la motivazione delle scelte di architetture, tecnologie e design pattern si rimanda al *Manuale sviluppatore\_v.2.0.0-R1*.

### 4.2 Report test

#### 4.2.1 Test unitari

Sono stati eseguiti i test unitari di tutti i servizi scritti per la realizzazione della webapp.

È stata raggiunta, una code coverage pari al **87%**, che è un valore accettabile secondo le metriche di qualità stabilite per il progetto.

### 4.3 Problemi aperti

Come sviluppi futuri si potrebbero implementare:

- ulteriori filtri per la visualizzazione dei report.