

# Junkyard Crowdsourcing

---

Riccardo Paltrinieri – 10626923

Progetto di *Tecnologie Informatiche per il Web*

2019/2020



## Specifiche di progetto

Il progetto consiste nello sviluppo di un'applicazione web per la convalida d'immagini aeree di località a rischio per la presenza di discariche illegali, con un approccio di crowdsourcing.

Il Manager crea una campagna di crowdsourcing e carica al suo interno un insieme di immagini aeree corredate di varie informazioni. Dopo aver caricato le immagini e i rispettivi dati, il Manager avvia la campagna di crowdsourcing, cui vari lavoratori possono prendere parte per classificare le immagini come rappresentanti vere discariche o come "falsi allarmi".

# Analisi dei dati

L'**applicazione** ha due tipi di utenti:

- Manager: detiene le immagini aeree, crea e pubblica una campagna di crowdsourcing. Ispeziona il risultato della campagna.
- Lavoratore: s'iscrive a una campagna di crowdsourcing e convalida le immagini aeree.

Una **campagna** ha i seguenti dati:

- Nome
- Committente
- Stato

Un **manager** ha i seguenti dati:

- Nome utente
- Parola d'ordine
- Indirizzo email

Un **lavoratore** ha i seguenti dati:

- Nome utente
- Parola d'ordine
- Indirizzo email
- Livello di esperienza
- Foto

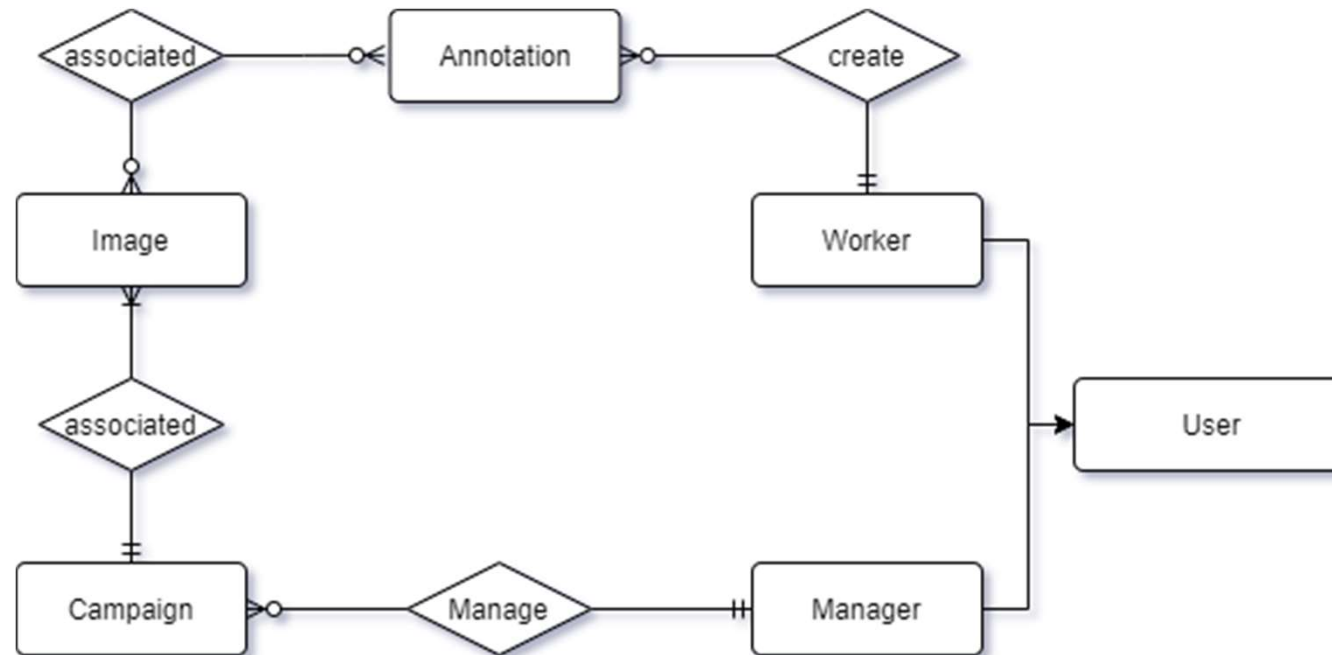
Un'**immagine** di una località a rischio per la presenza di una discarica ha i seguenti dati;

- Latitudine, Longitudine
- Comune
- Regione
- Provenienza
- Data di recupero
- Risoluzione

Un'**annotazione** ha i seguenti dati:

- Data di creazione
- Validità
- Fiducia
- Note: un campo di testo libero che registra i commenti del lavoratore

# Database Design



# Database Design

Table: jnk\_users

Columns:

<b>id</b>	int AI PK
<b>username</b>	varchar(45) PK
<b>mail_address</b>	varchar(45)
name	varchar(45)
password	varchar(45)
role	char(7)

Table: jnk\_campaigns

Columns:

<b>id</b>	int AI PK
<b>id_owner</b>	int
<b>name</b>	varchar(45)
customer	varchar(45)
state	char(7)
num_images	int(10) UN zerofill

Table: jnk\_annotations

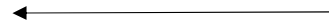
Columns:

<b>id</b>	int AI PK
<b>id_user</b>	int
<b>id_camp</b>	int
<b>id_image</b>	int
validity	int
text	tinytext
date	datetime
trust	char(7)

Table: jnk\_images

Columns:

<b>id</b>	int AI PK
<b>id_campaign</b>	int
name	int
latitude	varchar(45)
longitude	varchar(45)
city	varchar(45)
region	varchar(45)
source	varchar(45)
resolution	char(6)
date	datetime



# Html Components



## **Model Objects (Beans)**

- User
- Campaign
- Image
- Annotation
- CampaignStats



## **Controllers (Servlets)**

- Login
- Signup
- Logout
- EditProfile
- HomePage
- InfoManagerCampaign
- InfoWorkerCampaign
- ManagerEditCampaign
- WorkerEditCampaign
- InfoImage
- Statistics
- GetImage



## **Data Access Objects (DAO)**

- UserDao
- CampaignDAO
- ImageDAO
- AnnotationDAO



## **Filters**

- UserChecker
- ManagerChecker



## **View (Templates)**

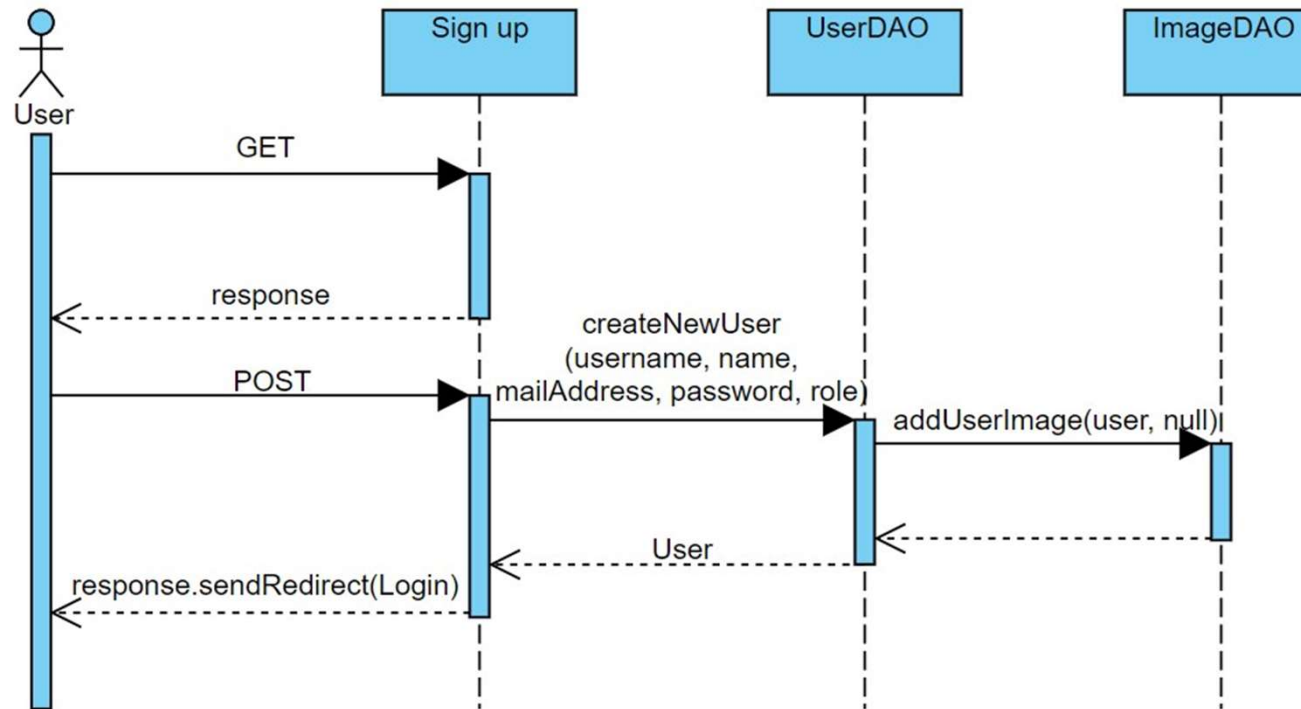
- index
- Home
- WorkerCampaign
- ManagerCampaign
- EditProfile
- CampaignStatistics
- Login
- SignUp



# Casi d'uso

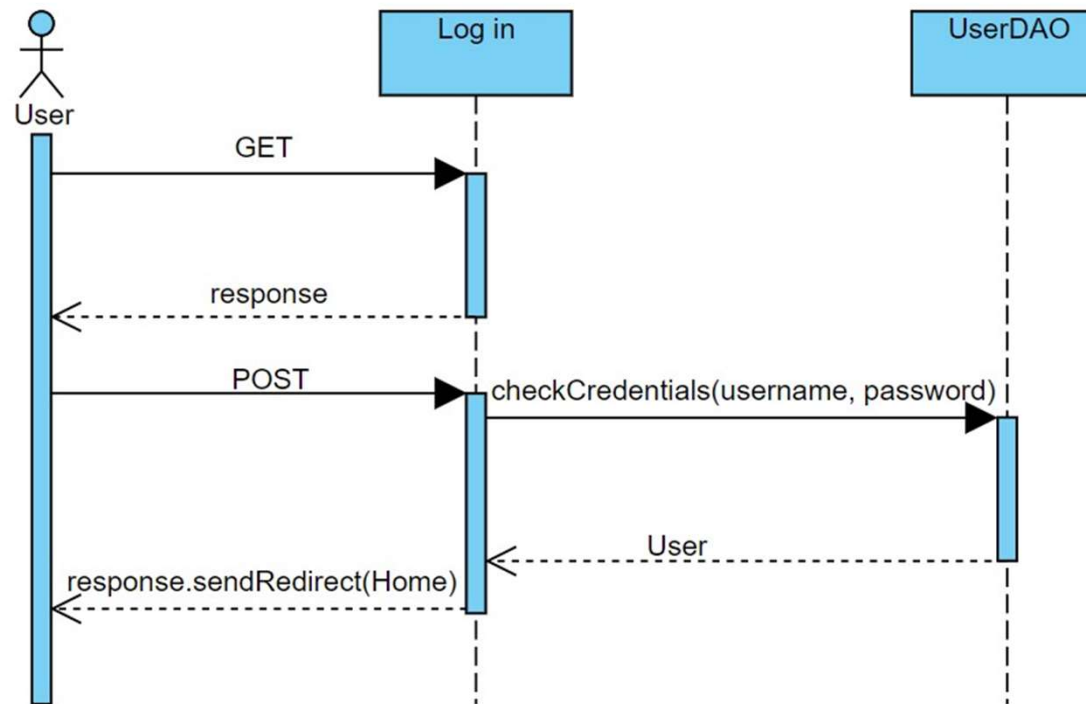
1. Utente si registra.
2. Utente esegue l'accesso.
3. Manager crea una campagna.
4. Manager carica le immagini e i dati di una campagna.
5. Manager avvia/chiude la campagna.
6. Manager controlla il contenuto di una campagna.
7. Manager controlla le statistiche di una campagna.
8. Lavoratore aggiunge un'annotazione.
9. L'utente modifica le sue informazioni personali.
10. Logout

# 1. Utente si registra

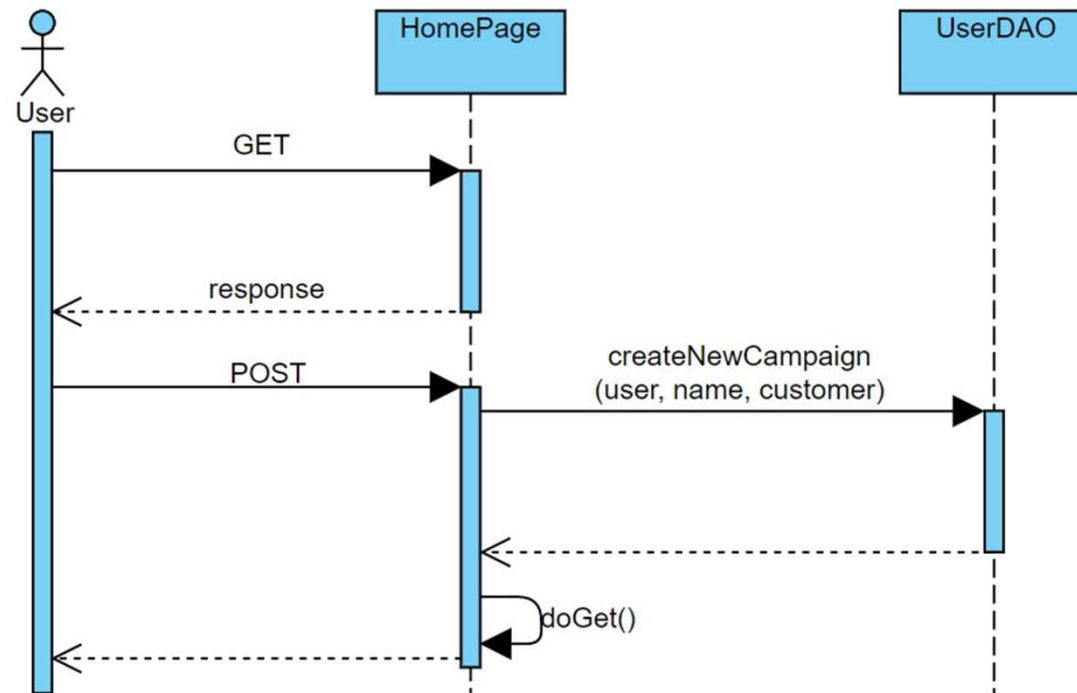




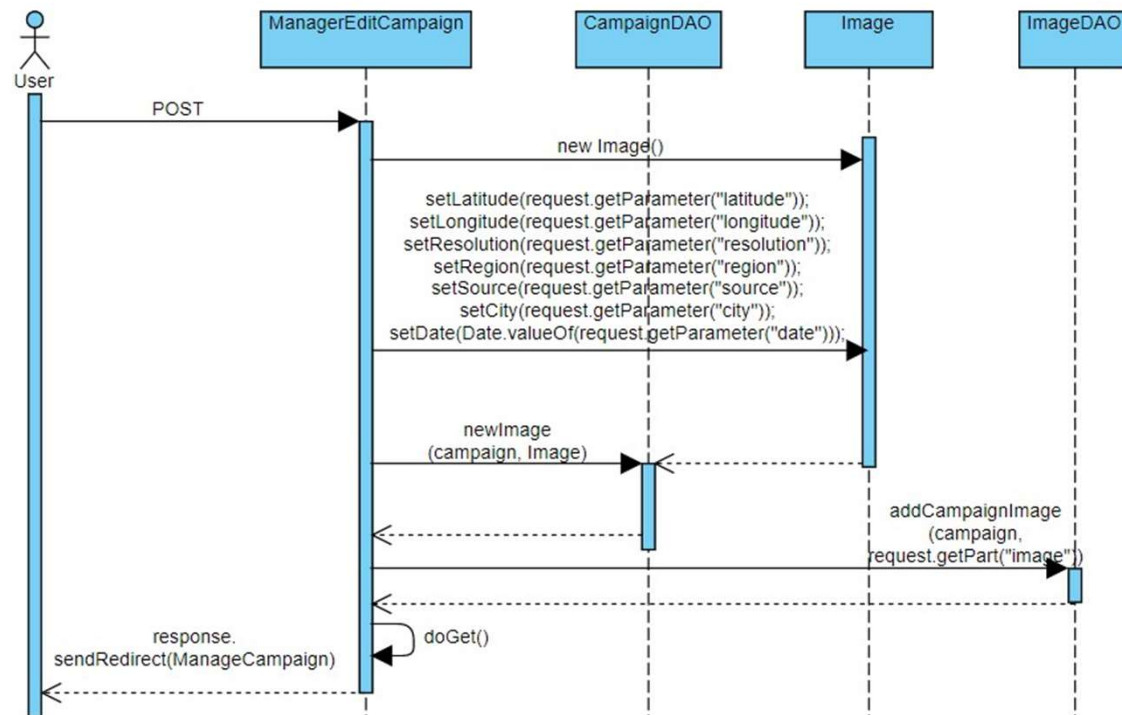
## 2. Utente esegue l'accesso



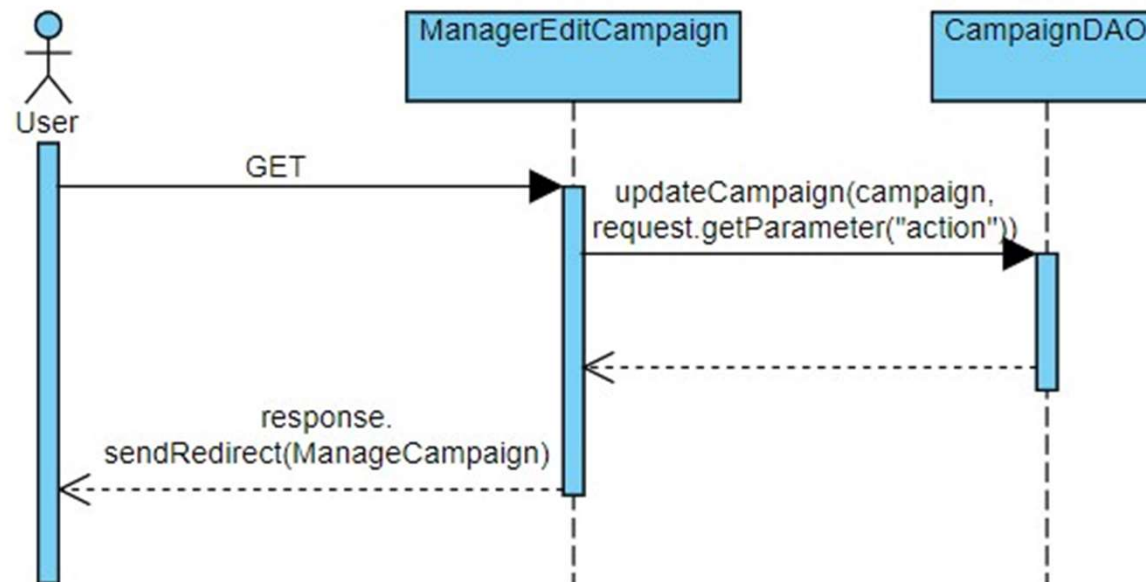
### 3. Manager crea una campagna



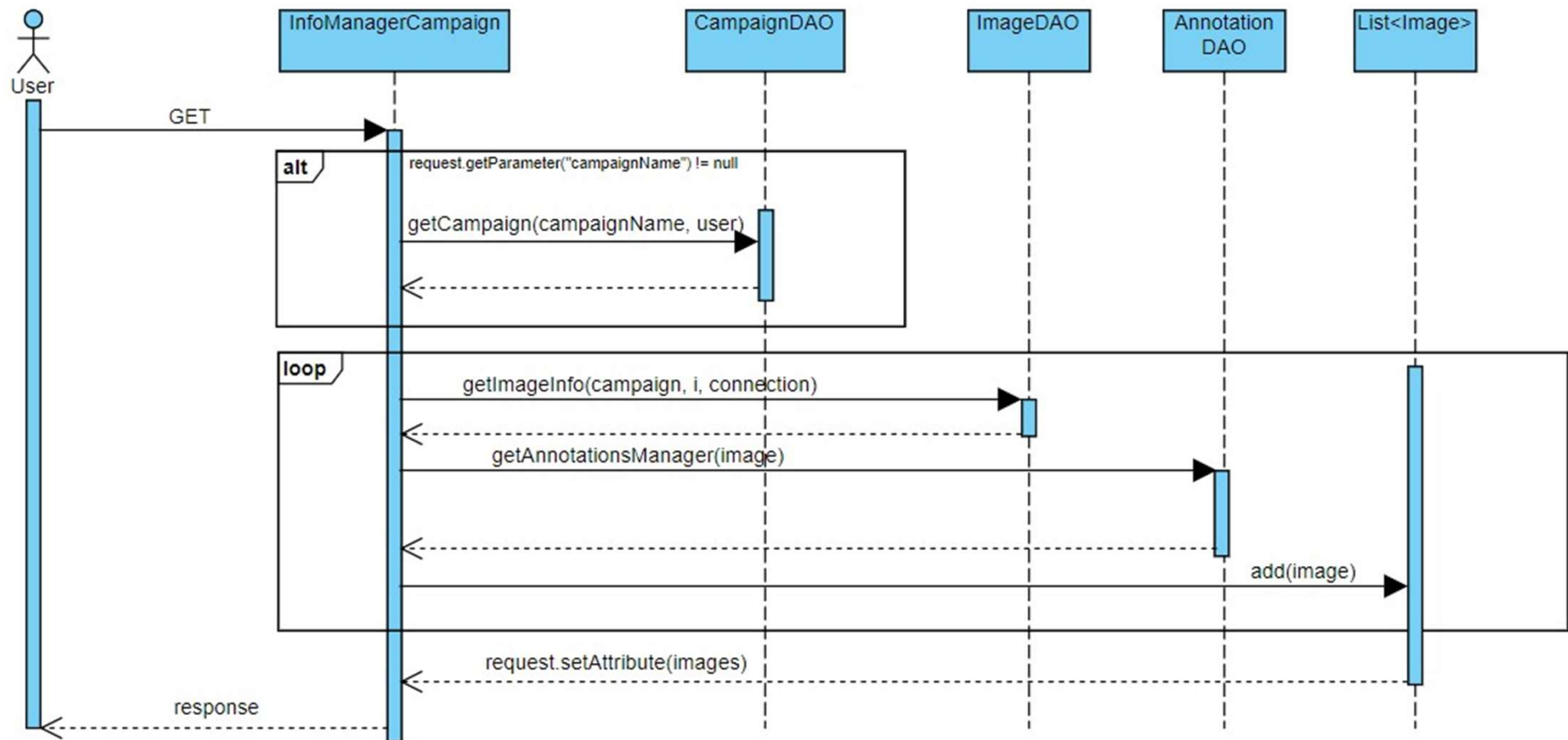
## 4. Manager carica le immagini e i dati di una campagna.



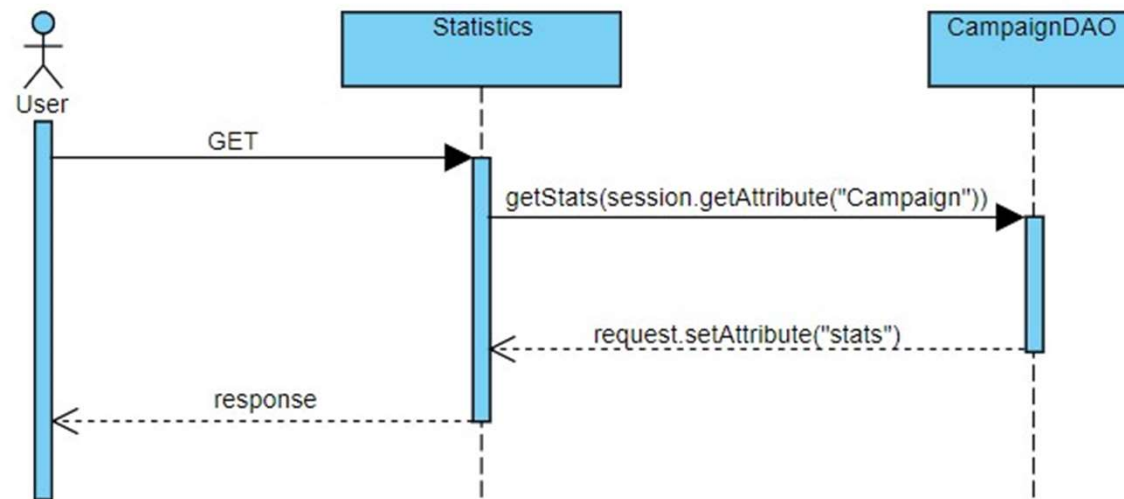
## 5. Manager avvia/chiude la campagna.



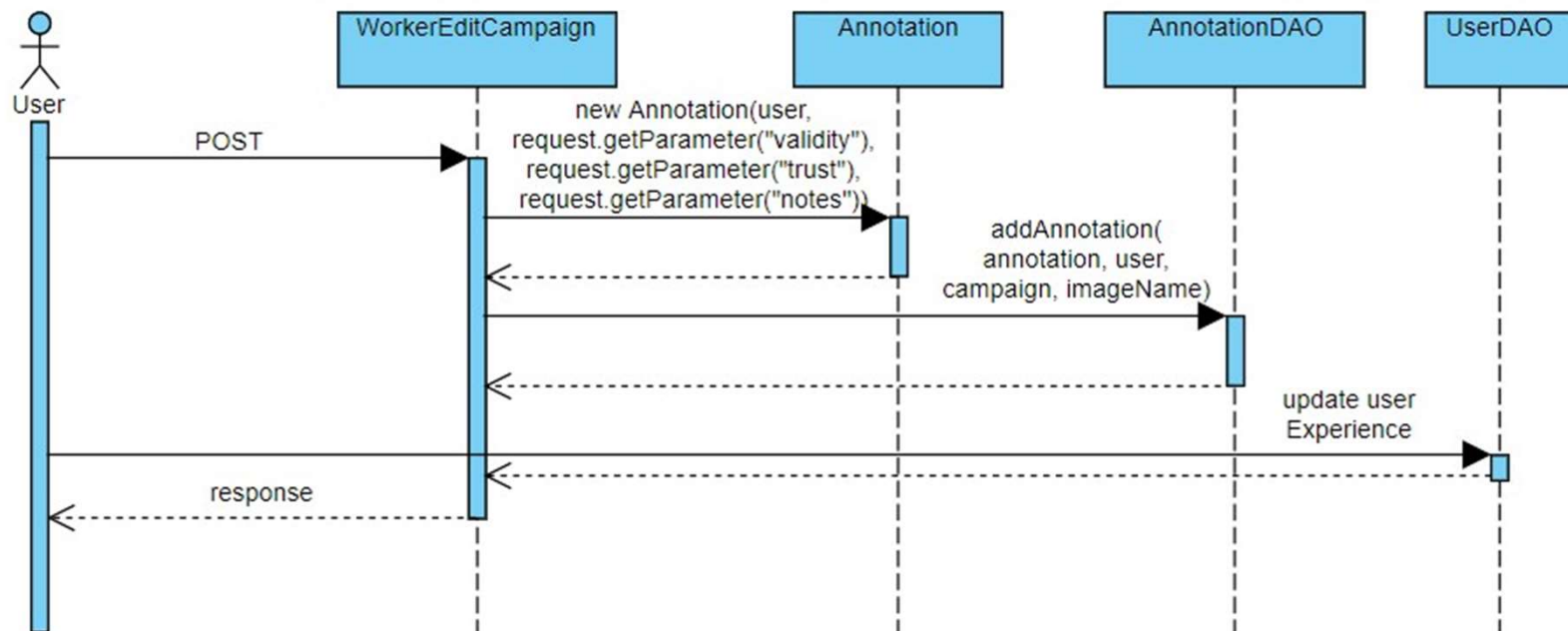
## 6. Manager controlla il contenuto di una campagna.



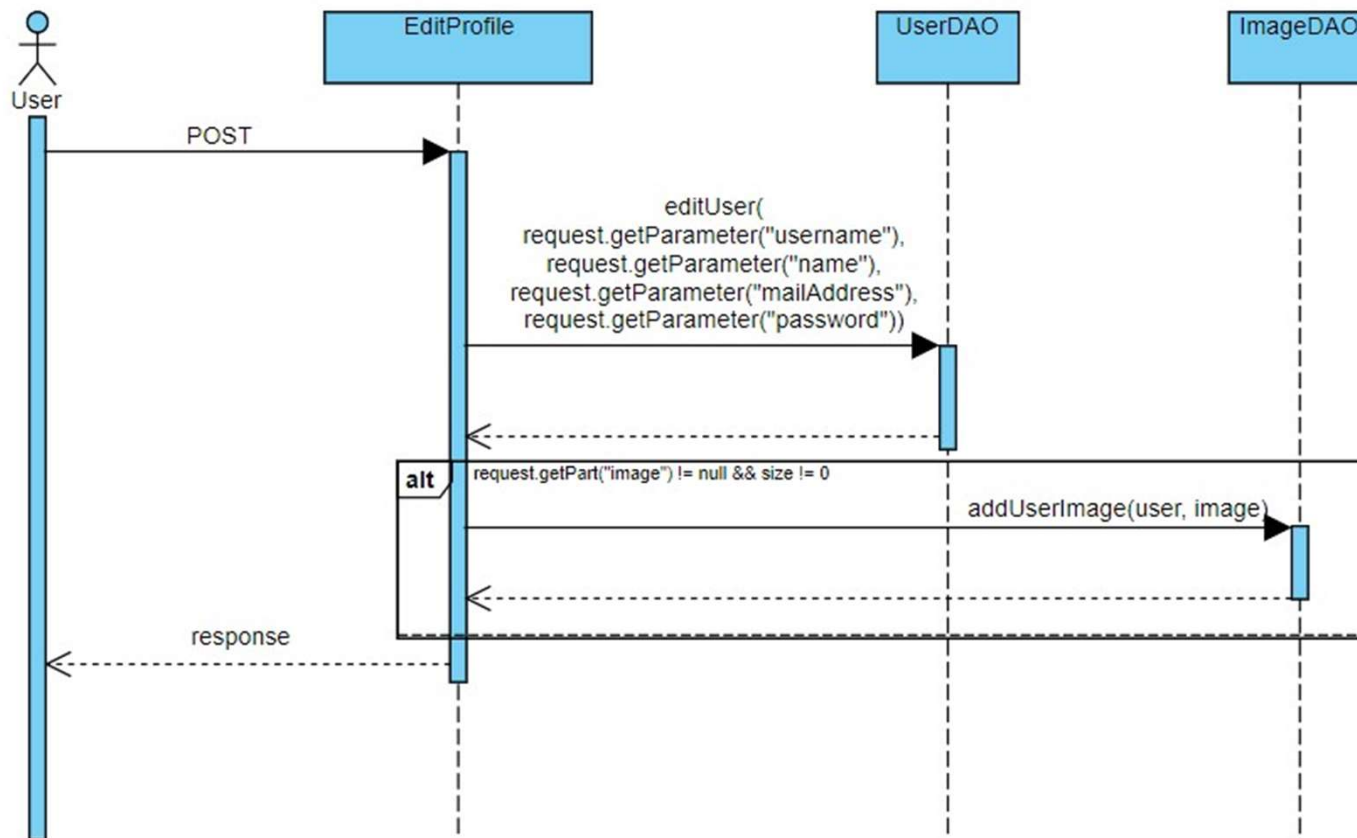
## 7. Manager controlla le statistiche di una campagna.



## 8. Lavoratore aggiunge un'annotazione.



## 9. L'utente modifica le sue informazioni personali.





## 10. L'utente esegue il Logout

