

# Progetto: “GeoPost”

## Corso di Mobile Computing

### Versione 2

Le differenze rispetto alla versione precedente sono riportate in giallo

## Introduzione

Scopo del progetto è la realizzazione di un prototipo di client per un social network geo-referenziato. In sintesi, ogni utente ha uno o più amici che segue, in modo analogo a quanto avviene in altri social network (es: twitter). Un utente può pubblicare un messaggio di stato (un breve testo) associato alla posizione in cui si trova; tutti gli utenti che lo seguono potranno vedere la posizione e il messaggio.

Si osservi che il sistema è un prototipo e dunque molti aspetti non strettamente necessari sono ignorati. Ad esempio: le tecniche per garantire la sicurezza e la privacy e alcune funzionalità (come, ad esempio, la registrazione, la cancellazione delle amicizie, il recupero della password, etc...)

## Descrizione del sistema

Il sistema è composto da un server (che sarà fornito dal docente), e da un client, che dovrà essere realizzato dagli studenti (in android e/o phonegap).

Il sistema permette ad ogni utente di registrarsi tramite una procedura web che sarà fornita. Tutte le altre funzionalità, descritte nel seguito, dovranno essere accessibili via applicazione mobile.

- **Login:** ogni utente fornisce il suo nome utente e la sua password

- **Amici seguiti.** L'utente può vedere l'elenco degli amici seguiti:

- Su una mappa, sulla quale viene mostrato un segnaposto per ogni amico seguito. Toccando il segnaposto viene mostrato un “balloon” con il nome e lo stato dell'amico.

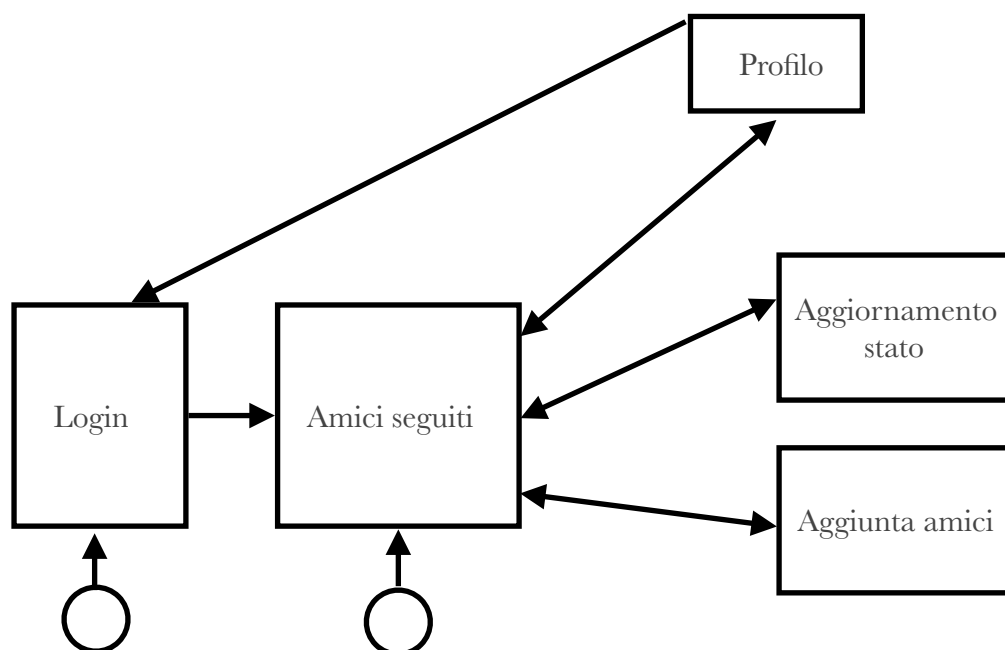
- In una lista, vengono mostrati gli amici seguiti, ordinati per distanza. Per ogni amico sono mostrati: il nome utente, il messaggio di stato e la distanza.

- **Aggiornamento stato:** ogni utente può aggiornare il proprio stato indicando un messaggio e la posizione in cui si trova, che deve essere mostrata sulla mappa.

- **Aggiunta amici.** Un utente può ricercare nuovi amici indicandone il nome utente. Una funzionalità di auto-completamento deve supportare questa operazione.
- **Profilo.** Ogni utente può vedere i propri dati: nome utente, ultimo messaggio e ultima posizione (mostrata sulla mappa). L'utente può effettuare il logout.

## Schema di navigazione

Gli studenti devono implementare lo schema di navigazione fornito. Ogni rettangolo nello schema seguente indica una schermata, la cui funzionalità è definita nella sezione precedente.



La progettazione interna delle singole schermate è a discrezione degli studenti. Si richiede tuttavia quanto segue: la schermata di profilo deve presentare due interfacce diverse per schermi stretti e per schermi larghi (vedi sezione “testing” in fondo). Su schermi larghi devono essere mostrate sulla sinistra le informazioni testuali (nome utente e messaggio di stato), mentre sulla parte destra deve essere mostrata la mappa. Su schermi stretti le informazioni sono organizzate verticalmente.

Le altre schermate devono presentare una visualizzazione accettabile sia su schermi larghi che su schermi stretti.

## Comunicazione con il server

Le API rese disponibili dal server sono documentate nel seguito.

BaseURL: <https://ewserver.di.unimi.it/mobicomp/geopost>

Tutte le chiamate restituiscono:

- Il codice di stato HTTP 200 in caso di richiesta corretta;
- Il codice di stato HTTP 400 nel caso in cui non siano forniti i parametri indicati nel seguito;
- Il codice di stato HTTP 401 nel caso in cui sia fornito un numero di sessione non valido

### REGISTER

Funzionalità di aggiunta utente (via web): <https://ewserver.di.unimi.it/mobicomp/geopost/register>

### LOGIN

Effettua il login. Verifica le credenziali e se corrette ritorna un session\_id. Se errate viene ritornato il codice di stato HTTP 400.

<https://ewserver.di.unimi.it/mobicomp/geopost/login>

Input (via post): “username” e “password”.

Output: session\_id. Il numero di session\_id rimane valido fino a quando non viene chiamata la funzione logout. Esempio

5ajy6

### FOLLOWED

Torna l'elenco dei contatti che un utente segue, con i loro dati.

<https://ewserver.di.unimi.it/mobicomp/geopost/followed>

Input (via get): session id. Esempio:

[https://ewserver.di.unimi.it/mobicomp/geopost/followed?  
session\\_id=5ajy6](https://ewserver.di.unimi.it/mobicomp/geopost/followed?session_id=5ajy6)

Output (in JSON): una lista di contatti, per ciascuno è indicato il nome utente, il messaggio di stato, latitudine e longitudine. Esempio

```
{“followed”:[  
  {“username” = “Anna”, “msg”=“Ciao”, “lat”:10.5, “lon”:15.7},  
  {“username” = “Luca”, “msg”=“Hola”, “lat”:12.5, “lon”:13.7},
```

```
]
}
```

## STATUS\_UPDATE

Permette di aggiornare lo stato (messaggio e posizione) di un utente.

`https://ewserver.di.unimi.it/mobicomp/geopost/status_update`

Input (via get): session\_id, message, lat, lon. Esempio:

`https://ewserver.di.unimi.it/mobicomp/geopost/status_update?  
session_id=5ajy6&message=hallo&lat=15.7&lon=12.4`

Output vuoto

## LOGOUT

Permette di effettuare il logout, cancellando dunque il session\_id

`https://ewserver.di.unimi.it/mobicomp/geopost/logout`

Input (via get): session\_id. Dopo questa chiamata il session\_id non è più valido e deve essere ricreato con una nuova chiamata login. Esempio:

`https://ewserver.di.unimi.it/mobicomp/geopost/logout?  
session_id=5ajy6`

Output vuoto.

## USERS

Ritorna l'elenco dei nomi utente che iniziano con un certo prefisso. Se il prefisso è vuoto, ritorna tutto l'elenco dei nomi utenti.

`https://ewserver.di.unimi.it/mobicomp/geopost/users`

Input (via get): session\_id, usernamestart (le lettere con le quali inizia il nome utente) e limit (il numero massimo di risultati da ritornare). Esempio:

`https://ewserver.di.unimi.it/mobicomp/geopost/users?  
session_id=5ajy6&usernamestart=lu&limit=10`

Output (in JSON): la lista di nomi utenti che iniziano con la stringa indicata. L'elenco ha al più "limit" elementi. Esempio

```
{"usernames" : ["luca", "luca123", "lucia"]}
```

## FOLLOW

Permette all'utente di iniziare a seguire un'altro utente, specificato dal suo nome utente.

`https://ewserver.di.unimi.it/mobicomp/geopost/follow`

Input (via get): session\_id, username (del nuovo utente da seguire). Esempio:

`https://ewserver.di.unimi.it/mobicomp/geopost/follow?`

`session_id= 5ajy6&username=luca`

Output: "OK" se l'operazione va a buon fine. In alternativa codice di stato HTTP 400 con i seguenti messaggi di errori:

CANNOT FOLLOW YOURSELF

ALREADY FOLLOWING USER

USERNAME NOT FOUND

## PROFILE

Restituisce le informazioni relative all'utente.

`https://ewserver.di.unimi.it/mobicomp/geopost/profile`

Input (via get): session\_id. Esempio:

`https://ewserver.di.unimi.it/mobicomp/geopost/profile?`

`session_id= 5ajy6`

Output (in JSON): la username dell'utente, l'ultimo messaggio di stato inviato, l'ultima posizione (lat e lon). Esempio:

```
{ "username": "Sergio", "msg": "ciao", "lat": 42.2, "lon": 30 }
```

## Testing

Durante la prova in laboratorio gli studenti dovranno utilizzare il simulatore del device Nexus S con risoluzione di schermo 480x800 e versione del sistema operativo Android 7.0 (API level 24). Si intende con "schermo stretto" lo schermo del device tenuto in posizione verticale (portrait), mentre si intende con "schermo largo" lo schermo del device tenuto in posizione orizzontale (landscape).

Durante le prove di esame sarà richiesto di aver creato 4 utenti di prova. Come nome utente usare un prefisso scelto da voi seguito da "-test1", "-test2", "test-3" e "-test4". Le password di questi utenti dovranno essere fornite durante l'esame (usate una sola password uguale per i 4 utenti, semplice da trascrivere). Creare le seguenti amicizie:

- L'utente test1 deve seguire tutti gli altri.
- L'utente test2 segue test3 e test4.

- L'utente test3 segue test4.
- L'utente test4 segue tutti gli altri.

Ogni utente deve aver inserito lo stato, con messaggio “ciao da [nome-utente]” in una posizione a vostra scelta a Milano.