

## JungleFlat

### ABSTRACT

JungleFlat è un'applicazione che cerca di organizzare in modo efficace e semplice la convivenza fra inquilini (in particolare modo studenti fuori sede) che hanno abitudini e stili di vita diversi.

L'obiettivo è quello di risolvere le problematiche più comuni riscontrate fra inquilini, ovvero: gestione delle spese condivise, scadenze delle bollette, turni delle pulizie, etc.

JungleFlat cerca di risolvere queste problematiche in modo semplice, creando dei gruppi e dando l'accesso alle seguenti funzionalità: **Post-it** (per lasciare degli avvisi agli inquilini), **Lista Spesa** condivisa, **Calendario** (per visualizzare in modo rapido le scadenze dell'appartamento), gestione **Pulizie** (possibilità di assegnare dei compiti e di lasciare dei consigli utili) e gestione delle **Spese condivise** (condividere e tenere traccia delle spese in comune). Le funzionalità sono strutturate in modo da poter essere utilizzate in maniera facile e veloce, senza che l'utente ci debba spendere troppo tempo.

### RELATED WORK

L'applicazione prende spunto da google keep( liste e postit condivisi), google calendar e da diverse app per la gestione delle finanze (FlatMate). Quello che non ho trovato è un'applicazione che avesse tutte queste funzionalità messe insieme e che avesse solo quelle che servono ad uno studente fuori sede, senza complicarne troppo l'utilizzo.

### USAGE MODEL

L'utente si registra ed ha tre possibilità: non creare nessun gruppo, creare un nuovo gruppo, aggiungersi ad un gruppo già esistente (inserendo il codice che gli è stato inviato via mail, tramite invito da parte di un membro del gruppo).

Effettuando il login si accede alla **home** nella quale sono presenti:

- **Post-it**: permette di aggiungere post-it ed eliminarli tramite shake del telefono oppure semplicemente selezionando il post-it.
- **Spesa**: gli elementi si aggiungono tramite tastiera oppure attraverso voice recognition, l'eliminazione avviene tramite shake del telefono o selezionando l'elemento della lista.
- **Pulizie**: si possono aggiungere dei consigli tramite il bottone '+' (ed eliminarli con lo shake), inserire dei compiti da eseguire (ogni coinquilino inserisce il proprio) e far ruotare i compiti tramite l'apposito bottone.
- **Calendario**: gli eventi vengono aggiunti selezionando la data dal calendario.
- **Finanze**: permette di aggiungere una spesa (tab *aggiungi*). Nella tab *crediti* vengono visualizzate le spese effettuate dall'utente e selezionando un singolo elemento della lista si può visualizzare la spesa completa e selezionare i coinquilini che hanno dato i soldi. Selezionando tutti i coinquilini sparirà la spesa dalla lista. Nella tab *debiti* l'utente visualizza i debiti che ha nei confronti degli altri inquilini.

La **nav bar** laterale dà la possibilità di accedere alle impostazioni:

- **Profilo utente**: cambio password e visualizzazione dei dati personali.
- **Impostazioni gruppo**: si può invitare un nuovo utente o eliminarne uno già esistente.
- **Logout ed eliminazione del proprio account**.

(le schermate principali sono presenti nelle ultime due pagine)

## ARCHITECTURE DESIGN

L'applicazione è stata implementata in modo da poter essere usata a partire dalla versione di Android 4.3 . Per l'utilizzo dell'applicazione è necessario essere connessi alla rete.

Le componenti principali dell'applicazione sono:

**Client:** il lato client è costituito dall'applicazione Android.

**Server php:** Il lato server è implementato in php. Nel database vengono salvate tutte le informazioni necessarie al funzionamento dell'applicazione. Nel telefono si salva un cookie e un token di sessione che servono a verificare se l'utente è loggato o meno.

**Google server:** per effettuare le richieste GCM. Questa componente è necessaria per l'implementazione delle notifiche.

La comunicazione fra client Android e server avviene tramite richieste HTTP (i dati vengono inviati tramite POST e richiesti tramite GET).

## IMPLEMENTATION

Per implementare il progetto è stato utilizzato *Android Studio* per la parte di Android, mentre per la parte in php è stato utilizzato *Sublime Text*.

Le funzionalità principali sono state implementate nel seguente modo:

- *ServerRequest*: classe implementata per gestire le richieste http del server utilizzando la libreria *org.apache.http*.
- Le risposte ricevute dal server sono state interpretate utilizzando *JSONObject* e *JSONArray*, mentre per l'invio delle informazioni si è utilizzato principalmente "*List<NameValuePair>*".
- Per gestire i singoli elementi di ogni lista (o griglia) sono stati utilizzati degli adapter (*BaseAdapter extends ListAdapter*).
- Eventi di shake del telefono sono gestiti dalla classe *ShakeEventListener* implementando *SensorEventListener* (ho trovato difficile calibrare questo sensore).
- Per la parte *speechToText* si è utilizzata la classe *startVoiceRecognitionActivity()*.
- Per la parte di GCM si è importata la libreria *com.google.gms* utilizzata per implementare le classi : *RegistrationIntentService*, *MyGmListener Service*, *NotificationReciver* (estende *BroadCast Reciver*), *MyInstanceIdListenerService*.

## EVALUATION

L'applicazione è stata testata utilizzando due dispositivi con due versioni differenti di Android (Nexus5 con Android 5.1 e Samsung S3 con Android 4.3). L'applicazione è stata provata anche su tablet, ma l'interfaccia è pensata per essere usata su smartphone.

L'applicazione è stata utilizzata da 8 persone (le mie 4 coinquiline e un altro gruppo di inquilini).

Gli utenti hanno definito l'app interessante e utile poiché non porta via molto tempo per essere utilizzata. Ho ricevuto alcuni suggerimenti(es: sincronizzare il calendario con google calendar, possibilità di mettere un check sulle pulizie effettuate). L'aspetto negativo sottolineato dagli utenti è quello relativo alla lentezza di risposta da parte del server.

## LIMITATION

La principale limitazione è data dalla lentezza delle risposte del server.

Un'altra limitazione di questa applicazione sono le notifiche tramite gcm (lato php) che non sono riuscite a portare a termine (principalmente per mancanza di tempo).

## TEAM MEMBERS CONTRIBUTION

L'applicazione è stata fatta solo da me, sia la parte di php che quella di Android.

Le righe di codice scritte sono all'incirca: 7000 Java, 2400 xml, 1500 PHP.

## LESSON LEARNED

Avendo lavorato da sola ho compreso l'importanza di avere un team con cui confrontarsi e suddividere il lavoro.

Se dovessi rifare il progetto mi soffermerei di più sulla parte di progettazione, sia quella relativa alla progettazione grafica che quella relativa all'implementazione.

## REFERENCES:

Per la soluzione dei problemi mi sono basata sulla ricerca online:

- stackoverflow
- google developer (soprattutto per la parte gcm):
- per layout e icone ho trovato molto utile e interessante:

<https://www.google.com/design/spec/material-design/introduction.html>



