



# Piano di Testing

Corso di Ingegneria del Software A.A. 2016 – 2017

Docente: Prof. Agostino Cortesi

PandaCode

851777 - Martina Lunardi

852056 – Claudia Farronato

Data: 01/05/2017

# Sommario

I.	<b>INTRODUZIONE</b> .....	2
II.	<b>GLOSSARIO</b> .....	2
III.	<b>PARAMETRI E MODALITÀ</b> .....	3
IV.	<b>PROCESSO DI VERIFICA: THREAD TESTING</b> .....	3
V.	<b>5. SCHEDULE DEL TESTING: TEMPO E RISORSE ALLOCATE</b> .....	3
VI.	<b>6. PROCEDURE PER LA REGISTRAZIONE DEI TEST</b> .....	3
VII.	<b>REQUISITI HARDWARE E SOFTWARE E VINCOLI DI TESTING</b> .....	4
VIII.	<b>TEST E CONCLUSIONI</b> .....	4

# INTRODUZIONE

---

In questo documento si vogliono evidenziare i termini con i quali verrà realizzata l'analisi dell'usabilità dell'applicazione BarbeCrew.

L'importanza di eseguire dei test sull'applicazione è data dal fatto che potenziali bug possono riscontrarsi in sede di pubblicazione dell'apk. I test quindi, verranno eseguiti su dispositivi fisici e virtuali. Effettueremo inoltre test di usabilità immedesimandoci nell'utente finale per individuare eventuali criticità dell'interfaccia e migliorarle. Tuttavia soltanto quando avremo reso pubblica l'applicazione potremo avere un vero riscontro grazie alle valutazioni lasciate dagli utenti su Play Store di Google.

## GLOSSARIO

---

- **Android:** è un sistema operativo per dispositivi mobili sviluppato da Google Inc. e basato sul kernel Linux. Non è da considerarsi, tuttavia, propriamente un sistema unix-like o una distribuzione GNU/Linux, dal momento che la quasi totalità delle utilità GNU è sostituita da software in Java.
- **App:** dicitura abbreviata per indicare un'applicazione software, sia ludica che di utilità, per dispositivi mobili quali smartphone, palmari e tablet.
- **APK:** file dell'applicazione che permette l'installazione della stessa nel device.
- **Crash:** si dice quando il sistema fallisce un'azione o registra un'anomalia che ne arresta il funzionamento, richiedendo un riavvio totale dell'applicazione, con la conseguente perdita dell'attività che si stava svolgendo.
- **Feedback:** designa il processo per cui l'effetto risultante dall'azione di un sistema si riflette sul sistema stesso per variarne o correggerne opportunamente il funzionamento e che si abbia, come risultato finale, l'intensificazione oppure l'attenuazione dell'effetto.
- **Matrice di tracciabilità:** utilizzata per descrivere le dipendenze dei casi d'uso, consiste in una matrice che relaziona i requisiti: quelli in riga utilizzano la funzionalità descritta da quelli in colonna.
- **Piano di Testing:** come si decide di proseguire circa la valutazione di possibili errori all'interno del codice della applicazione.
- **Tester:** coloro che in maniera volontaria si offrono di fare da "cercatori di bug", utilizzando un'applicazione ancora da pubblicare, che si trova in fase embrionale e/o avanzata di progettazione.
- **Thread:** è la più piccola unità di lavoro che un sistema può eseguire.
- **Thread Testing:** uno dei tanti test per verificare la presenza di bug all'interno del codice. La sua efficacia varia in base alla grandezza delle casistiche di eventi che si possono verificare all'interno dell'applicazione: più è vasta la struttura interna dell'applicazione, più saranno le combinazioni di azioni che potranno verificarsi e quindi più alta la possibilità di non verificare un caso malfunzionante.
- **User friendly:** si parla di un'applicazione user friendly quando questa è facilmente comprensibile all'utente sin dal primo utilizzo, rendendo la sua esperienza gratificante e invogliandolo ad utilizzare l'applicazione.

## PARAMETRI E MODALITÀ

---

I test sull'applicazione verranno effettuati sia in corso d'opera con vari debug del codice, sia con test eseguiti avviando l'applicazione su diversi dispositivi smartphone. Una volta completata la fase di testing, i risultati conseguiti ci permetteranno di trarre le conclusioni necessarie sulla validità della nostra applicazione.

Infine, per migliorare l'esperienza d'uso confideremo nei feedback lasciati dagli utenti finali del prodotto sul Play Store una volta resa pubblica l'app.

## PROCESSO DI VERIFICA: THREAD TESTING

---

Il modello di testing che abbiamo deciso di utilizzare per lo sviluppo dell'app è il "Threading Test". Questa tecnica ci permette di verificare le capacità funzionali che svolgono compiti chiave. Questo tipo di tecnica è utile se un'applicazione è di tipo client-server. Il modello segue uno schema logico preciso. Per procedere con la verifica dell'app è necessario partire da situazioni legate ad eventi singoli che si andranno a complicare man mano che si procede. Inoltre, occorre seguire eventi che si sviluppino secondo uno stesso thread logico e non in maniera casuale.

## SCHEDULE DEL TESTING: TEMPO E RISORSE ALLOCATE

---

Il metodo di procedere scelto per l'esecuzione del piano di testing richiederà una fase continua di prove durante le versioni dell'applicazione che precedono quella finale. Una volta testata anche la versione finale del nostro applicativo, procederemo con la pubblicazione della stessa sullo store Google. I tempi richiesti per l'analisi delle versioni alpha e beta sono costanti fino alla fase ultima del codice. Essendo il nostro un team in cui non è prevista alcuna retribuzione, le risorse allocate consistono in forza lavoro.

## PROCEDURE PER LA REGISTRAZIONE DEI TEST

---

Il modo in cui andremo a testare la nostra applicazione consisterà in due momenti distinti, in base allo stato di avanzamento dell'applicazione: nella fase uno verrà utilizzato Android Studio e i device messi a disposizione da parte dell'ambiente di sviluppo e i device di tipo fisico a nostra disposizione (smartphone). Nella fase due, invece, andremo ad utilizzare i dati prodotti (autenticazione di un utente avvenuta con successo, aggiunta di foto e commenti, aggiunta di un luogo ai preferiti, visualizzazione dei punti di interesse) dal database creato con Firebase. Una volta ricevuti i dati, provvederemo ad una analisi degli stessi e agiremo nel codice di conseguenza.

# REQUISITI HARDWARE E SOFTWARE E VINCOLI DI TESTING

---

La nostra applicazione dovrà essere supportata dalla maggior parte dei device ad oggi in commercio. L'hardware richiesto per eseguire l'applicazione consiste in almeno 2 Gb di RAM e 1 Ghz di CPU.

Il sistema operativo richiesto è Android nelle versioni 4.1 e superiori.

I vincoli condizionanti i test riguardano le risorse di tempo che abbiamo a disposizione: eventuali ritardi potrebbero comportare slittamenti sulla data finale di pubblicazione dell'applicazione.

## TEST E CONCLUSIONI

---

Una volta ultimata l'applicazione, eseguiremo dei test su alcuni dispositivi fisici a nostra disposizione dotati di versioni di software ed hardware più o meno aggiornati. Ciò ci consentirà di verificare la validità della nostra applicazione e verificare la qualità delle operazioni che svolge dal punto di vista di device con caratteristiche differenti.