



Università Ca' Foscari di Venezia

Dipartimento di Scienze Ambientali,  
Informatica e Statistica

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

Docente: Prof. Agostino Cortesi

## **PIANO DI TESTING**

Data di consegna: 15/03/17

Versione 1.1



# MASSIVE DYNAMICS

Paolo Vedorin  
Via San Pelajo 139  
31100 Treviso

345.8052880

[849942@stud.unive.it](mailto:849942@stud.unive.it)

Mariagiovanna Czarnecki  
Via per Cavriè 34  
31030 Breda di Piave

346.0262298

[854098@stud.unive.it](mailto:854098@stud.unive.it)

Dora Pavan  
Strada delle Acquette 19  
31100 Treviso

347.0358079

[860980@stud.unive.it](mailto:860980@stud.unive.it)

# Sommario

<b>1</b>	<b>INTRODUZIONE .....</b>	<b>3</b>
<b>2</b>	<b>GLOSSARIO.....</b>	<b>4</b>
<b>3.</b>	<b>PARAMETRI E MODALITÀ.....</b>	<b>5</b>
<b>4.</b>	<b>PROCESSO DI VERIFICA: THREAD TESTING .....</b>	<b>5</b>
<b>5.</b>	<b>SCHEDULE DEL TESTING: TEMPO E RISORSE ALLOCATE .....</b>	<b>6</b>
<b>6.</b>	<b>PROCEDURE PER LA REGISTRAZIONE DEI TEST.....</b>	<b>6</b>
<b>7.</b>	<b>REQUISITI HARDWARE E SOFTWARE E VINCOLI DI TESTING.....</b>	<b>6</b>
<b>8.</b>	<b>TEST E CONCLUSIONI.....</b>	<b>7</b>

# 1 INTRODUZIONE

---

Nel seguente documento, si vogliono evidenziare i termini con i quali si procederà con l'analisi dell'usabilità della nostra applicazione, in base a dei punti da noi decisi in sede di riunione.

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 verranno eseguiti sia in corso d'opera, con vari debug del codice, sia nella release version dell'applicazione per mezzo di "Firebase – Test Lab". Tale strumento è messo a disposizione all'interno del sito "console.firebase.google.com" e ha la funzione di testare su diversi dispositivi l'apk che viene caricato nell'apposita sezione. I dispositivi si dividono in fisici e virtuali e coprono la maggior parte degli SDK presenti in commercio. Una volta eseguito il test, i risultati verranno mostrati in una decina di minuti. Riteniamo importante eseguire questo tipo di test in quanto è possibile controllare eventuali errori anche in SDK che non sono a nostra disposizione.

## 2 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 che permette l'installazione dell'applicazione nel device.
- **Release-APK:** file che permette l'installazione dell'applicazione nel device. Questa sarà anche la versione che verrà pubblicata nel Play Store.
- **Piano di Testing:** come si decide di proseguire circa la valutazione di possibili errori all'interno del codice della applicazione.
- **Thread:** tradotto "filo". Si associa il thread ad una discussione all'interno di un forum. Nel caso di un applicativo si parla di sequenza di istruzioni di un programma in corso di esecuzione.
- **Thread Testing:** uno dei tanti test per verificare la presenza di bug all'interno del codice. La sua bontà 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.

### 3. PARAMETRI E MODALITÀ

---

Come detto in precedenza, la debug mode e il test – lab di Firebase costituiscono la base rispettivamente dello sviluppo del codice e della parte finale di test legata alla pubblicazione finale dell’APK. Ci avvarremo anche dei commenti lasciati dagli utenti sul Play Store una volta resa pubblica l’applicazione.

### 4. PROCESSO DI VERIFICA: THREAD TESTING

---

Nel mentre dello sviluppo, il piano che abbiamo deciso di seguire è quello del **“Thread Testing”** dal momento che ci è parso il più appropriato per la nostra applicazione. Date le dimensioni non eccessive di “Picnicly”, riteniamo possibile un “Threading Test” completo e avere una valutazione totale della qualità del codice da noi scritto in sede di programmazione. 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. Tuttavia, abbiamo deciso di accostare a questo test anche quello di Firebase, poiché lo riteniamo più affidabile e sicuro. Come detto in precedenza, questo test verrà compiuto ad avvenuta pubblicazione del release-APK.

## 5. SCHEDULE DEL TESTING: TEMPO E RISORSE ALLOCATE

---

Il modus operandi scelto per procedere col piano di testing richiederà una fase continua di prove durante le fasi alpha e beta, precedenti a 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. Come team senza portafoglio, non allocheremo un quantitativo di risorse monetarie ma solo di forza lavoro.

## 6. PROCEDURE PER LA REGISTRAZIONE DEI TEST

---

Il modo in cui andremo a testare la nostra applicazione consisterà in due momenti distinti, in relazione alla fase in cui il codice si troverà: nella fase uno (alpha e beta) utilizzeremo Android Studio e i device messi a disposizione da parte dell'ambiente di sviluppo (in alternativa, potremo testarne uno o più di tipo fisico). Nella fase due, invece, andremo ad utilizzare i dati prodotti dalle verifiche condotte da Firebase. Una volta ricevuti i dati, provvederemo ad una analisi degli stessi e agiremo nel codice di conseguenza.

## 7. REQUISITI HARDWARE E SOFTWARE E VINCOLI DI TESTING

---

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

Il software richiesto, invece, è Android Kitkat 4.2 o superiore.

I vincoli condizionanti i test sotto i quali agiremo sono prettamente legati alle risorse di tempo che avremo a disposizione: un ritardo anche minimo potrebbe determinare uno slittamento di alcuni giorni dei test causando un problema con le varie fasi di prove.

## 8.TEST E CONCLUSIONI

---

Una volta ultimata l'applicazione, eseguiremo dei test per verificare la bontà di quanto prodotto. Per eseguire tali prove abbiamo scelto di utilizzare "Firebase – Test Lab", in cui è possibile selezionare dei device, divisi tra fisici e virtuali, per eseguire delle prove di testing. I vari dispositivi sono caratterizzati da API di livello differente, cosicché sia possibile testare varie versioni del sistema operativo e riscontrare eventuali errori. Tutto ciò di cui si ha bisogno per eseguire questi test è un APK funzionante da caricare nell'apposita sezione di test. In seguito, si procede con la selezione dei vari device su cui si vuole testare l'APK. Nel giro di una decina di minuti verranno restituiti i risultati, comprensivi di errori e altre informazioni inerenti l'applicazione. Nella versione gratuita di Firebase, i test sono limitati a 15 al giorno suddivisi in 5 test legati a dispositivi fisici e 10 a quelli virtuali.