



Progetto di Ingegneria del Software 2022/23

Università Ca' Foscari Venezia

Piano di testing

Nome Gruppo
Dokkaebi

07/01/2023



Document Informations

Nome progetto	Dokkaebi
Deliverable	Piano di testing
Data di consegna	07/01/2023
Team Leader	Leonardo Bilardi - 884934@stud.unive.it
Team members	Beatrice Spagnolo - 884798@stud.unive.it Maisha Fahmida Bhuiyan - 887349@stud.unive.it Dumitru Zotea - 887372@stud.unive.it Nicolò Bregantin - 881142@stud.unive.it

Document History

Version	Issue Date	Stage	Changes	Contributors
1.0	12/11/2022	Draft	Creazione documento	Leonardo, Beatrice, Maisha, Dumitru, Nicolò
1.1	07/12/2022	Draft	Inserimento risultati test	Leonardo, Beatrice, Maisha, Dumitru, Nicolò
2.0	07/01/2023	Final	Finalizzazione documento	Leonardo, Beatrice, Maisha, Dumitru, Nicolò



PIANO DI TESTING

VERSIONE 2.0



Indice

1. INTRODUZIONE	5
2. GLOSSARIO	6
3. RIEPILOGO DEI TEST	8
3.1. In ambito di test	8
3.2. Fuori dall'ambito di test	8
4. ANALISI DELLE AREE DI TEST E DEL CAMPO DI APPLICAZIONE	9
4.1. Rilascio del prodotto	9
4.2. Test di regressione	9
4.3. Piattaforma di testing	9
5. ALTRI TEST	10
6. TEST SCHEDULING	11
6.1. Livelli di responsabilità	11
6.2. Approccio e tipi di test	11
6.3. Pianificazione dell'esecuzione dei test	12
6.4. Struttura, dati e piano di fornitura delle risorse	13
6.5. Strumenti di test	13
6.6. Verifica della procedura di testing	13
7. PROCEDURE DI REGISTRAZIONE DEI TEST	14
8. REQUISITI PER L'ATTIVITÀ DI TESTING	16
8.1. Gestione dei difetti	16
9. VINCOLI	16
10. RIFERIMENTI	16



1. INTRODUZIONE

Questo documento racchiude le modalità di verifica per testare in fase di sviluppo il funzionamento del sistema software analizzando le risorse richieste e come verranno effettuati i test.

In particolare:

- Quali possono essere le possibili problematiche del sistema
- Quali sono i test che si possono svolgere
- Che procedura si intende seguire per svolgere questi test
- Come si potranno segnalare gli errori riscontrati
- Come si potranno eventualmente correggere tali errori

Questa analisi servirà per individuare malfunzionamenti e risolverli per soddisfare i requisiti.

Essendo Android la componente fondamentale del progetto, si da per scontato che esso garantisca il funzionamento di tutte le features integrate nell'OS nell'arco di tutta la durata del progetto.

I test verranno eseguiti dai vari membri del team individualmente o insieme a un'altro componente del gruppo.

In particolare ciascun membro del gruppo utilizzerà un dispositivo proprio compatibile per il testing diverso dagli altri componenti del gruppo con diverse versioni di Android e diverse caratteristiche hardware (CPU, RAM, dimensione schermo ecc.) Per avere una varietà maggiore di dispositivi verranno usati gli emulatori forniti da Android Studio.

Seguendo una tabella indicativa contenente tutte le features implementate, ogni membro provvederà a testare ed individuare eventuali errori di funzionamento o di design relativi all'applicazione e segnalargli al resto del gruppo.

Questo permetterà un testing ampio in modo da permettere un debug più completo.

Nella fase finale del testing verrà chiesto a persone terze, non appartenenti al team di sviluppo, di testare l'applicazione e fornire un feedback. Vera offerto ad amici e conoscenti tramite la piattaforma social Discord di installare l'APK e provare le funzionalità dell'applicazione come se fossero un utente qualunque.



2. GLOSSARIO

- **Activity:** un'activity rappresenta una schermata con l'interfaccia utente di un'applicazione ed il suo scopo è quello di permettere l'interazione con l'utente;
- **Android:** sistema operativo basato su kernel Linux per dispositivi mobili, quali smartphone e tablet, nel nostro caso verranno considerate solamente versioni di Android superiori alla 9.0;
- **Android Studio:** Android è l'ambiente di sviluppo integrato dove si svilupperà la nostra applicazione;
- **API:** è l'acronimo di Application Programming Interface, serve ad indicare un gruppo di procedure già disponibili al programmatore per la realizzazione di un determinato compito;
- **App:** abbreviazione della parola "applicazione" per indicare un software per dispositivi mobile
- **Backup:** con backup si intende la messa in sicurezza dei dati di un sistema informatico attraverso la creazione di ridondanza delle informazioni stesse da utilizzare in caso di guasto per fare il ripristino dei dati;
- **Bug:** il bug è un problema che porta ad un malfunzionamento del software, producendo ad esempio un risultato inatteso o errato, tipicamente dovuto ad un errore di scrittura del codice. Un codice con un numero ragionevolmente alto di bug è detto "buggato";
- **Database:** rappresenta la locazione in cui vengono memorizzati tutti i dati riguardanti gli eventi o i posti da visitare nel database. I dati verranno strutturati in modo tale da facilitare l'inserimento e l'aggiornamento degli stessi in modo dinamico;
- **Firebase:** è un database NoSQL sviluppato da Google che permette di avere un database in tempo reale per la gestione dei dati, inoltre è una piattaforma per la creazione di applicativi mobile e web;
- **GitHub:** GitHub è un servizio di hosting di repository Git, ma offre molte più funzionalità. GitHub fornisce un'interfaccia grafica basata sul web, il controllo dell'accesso e diverse funzionalità di collaborazione, come strumenti di base per la gestione delle attività per ogni progetto;
- **GPS:** si intende il sistema di geolocalizzazione all'interno del dispositivo che il giocatore utilizza per individuare la propria posizione



- **Java:** linguaggio di programmazione orientato agli oggetti, progettato per essere più indipendente dalla piattaforma di esecuzione;
- **Smartphone:** dispositivo dotato di tecnologie hardware avanzate rispetto ad un comune cellulare, quali uno schermo touch, una memoria RAM e spazio di archiviazione minimi per installare svariate app, un processore di discreta potenza per far girare il sistema operativo e le app che verranno successivamente installate, un sistema di sensoristica avanzato (gps, giroscopio, accelerometro, ecc.) ed una connessione ad internet per permettere la comunicazione con internet;
- **Unit testing:** chiamato anche “test sulle unità”, con unit testing si intende l'attività di testing di singole unità software;



3. RIEPILOGO DEI TEST

3.1. In ambito di test

L'ambiente di testing sarà supervisionato dal responsabile della qualità e del testing Nicolò, tuttavia ogni membro del gruppo testerà l'applicazione per individuare eventuali malfunzionamenti.

Le funzionalità la cui implementazione deve essere corretta sono:

- login
- registrazione
- modifica dei dati dell'utente
- georilevazione e geolocalizzazione
- corretta interfaccia grafica
- corretta visualizzazione del browser integrato nell'app
- corretta visualizzazione delle mappe integrate nell'app
- inizializzazione di app di sistema come calendario e mappe

3.2. Fuori dall'ambito di test

Fuori dall'ambito di test verranno considerati i commenti del Play Store e il feedback fornito dagli utenti e amici che sono stati contattati su Discord per testare l'applicazione



4. ANALISI DELLE AREE DI TEST E DEL CAMPO DI APPLICAZIONE

4.1. Rilascio del prodotto

L'applicazione verrà sviluppata nell'ambiente di sviluppo Android Studio utilizzando le librerie principali di Java e di Android (SDK) compatibili con le versioni di Android 9.0 e successive.

Per garantire una comunicazione affidabile con un database esterno si sfruttano le API di Firebase.

L'applicazione sarà un software in grado di operare all'interno di uno smartphone Android dotato di connessione a internet e opportuna sensoristica come il GPS, il WI-FI, SIM eccetera, per permettere la lettura e scrittura di dati in un database e la geolocalizzazione dell'utente.

Inoltre, la versione di rilascio deve contenere tutti i requisiti previsti dai documenti precedenti: Piano di Progetto e Documento dei Requisiti.

4.2. Test di regressione

I test di regressione garantiranno che le vecchie funzionalità rimarranno utilizzabili nelle nuove versioni.

Per identificare eventuali errori creati durante il debugging o modifiche atte a sistemare problematiche e/o bug verrà messo a prova l'intero protocollo di testing.

Ciò serve per assicurarsi che con l'introduzione di nuove funzionalità o modifiche al codice, le funzioni già implementate e funzionanti non vengano compromesse.

Una volta assicuratosi che le modifiche sono state implementate con successo, il codice verrà pushato e uploadato su GitHub in modo tale che tutti i componenti del gruppo abbiano la stessa versione aggiornata dell'applicazione

4.3. Piattaforma di testing

Per testare l'applicazione si utilizzerà l'emulatore Android integrato in Android Studio e su più dispositivi fisici possibili, in modo da garantire una vasta compatibilità.



5. ALTRI TEST

Come scritto nell'introduzione del documento, il responsabile dei test sarà Nicolò.

<i>Test n°</i>	<i>Ambito</i>	<i>Dettaglio</i>
1	Sicurezza	La confidenzialità delle credenziali degli utenti è garantita dal fatto che la password sarà crittografata all'interno del database. A livello di dati, essi non verranno mai gestiti per scopi illeciti o di marketing ma, al più, per statistiche mostrate in chiaro all'utente.
2	Stress test	Le prestazioni dell'app e del database saranno messe a dura prova attraverso un uso prolungato ed intensivo che tenterà di violare i limiti realistici d'uso dell'applicazione. Per i test verrà utilizzato, se e quando possibile, il servizio "Firebase Test Lab"
3	Connettività	Viene utilizzata una connessione a internet (per comunicare col database in tempo reale) e una connessione satellitare (per geolocalizzare l'utente). Sebbene Firebase sia già testato per prevenire errori in caso di disconnessioni, è consigliabile testare l'app in caso di disconnessione improvvisa, per controllare che i dati vengano mantenuti consistenti e siano recuperati del tutto (o che perlomeno la perdita sia minima).
4	Backup	I dati dell'utente verranno mantenuti sul database di Firebase che fa affidamento sui server di Google con i propri servizi di ripristino dei dati. Verranno effettuati backup manuale dei dati archiviati sul server in formato JSON ad intervalli regolari. Per quanto riguarda la documentazione del progetto si farà affidamento sul cloud e locale per l'archiviazione della documentazione e dei file sorgente. In particolare si utilizzeranno Google Drive e GitHub.
5	Unit Test	Il testing, effettuato sia tramite emulatore che dispositivi fisici, sarà concomitante con lo sviluppo dell'applicazione. Una "closed beta" e' prevista nella fase di testing dove l'APK verrà fornito ad alcuni utenti e amici



6. TEST SCHEDULING

6.1. Livelli di responsabilità

Tabella esplicativa dei livelli di responsabilità dei test che verranno applicati.

Livello di test	Team di progetto	Professore	Esterno
<i>Sicurezza</i>	P	P	S
<i>Stress test</i>	P	P	S
<i>Connettività</i>	P	S	S
<i>Backup</i>	P	–	–
<i>Unit Test</i>	P	S	S

Legenda: P - responsabilità primaria; S - responsabilità secondaria

6.2. Approccio e tipi di test

Gli obiettivi sono verificare che l'applicazione:

- Soddisfi i requisiti definiti;
- Funzioni in modo accurato;
- Gestisca correttamente le condizioni di errore;

Inoltre, si deve verificare che:

- Le interfacce funzionino correttamente;
- Il caricamento dei dati si concluda correttamente.

Considerato che l'applicazione si focalizza soprattutto su funzionalità visive poco onerose in termini di prestazioni, la modalità di test sarà soprattutto **bottom up**.

Questo approccio, anche se non è il più adatto per risolvere bug ad alto livello una volta che il progetto diventa complesso, serve per testare e identificare malfunzionamenti o incompatibilità.

Dato che l'applicazione verrà implementata con un linguaggio object-oriented, verrà effettuato un **testing incrementale** per permettere di testare le singole funzionalità a basso livello, che sarebbero più difficili da rilevare in seguito una volta che il progetto diventa complesso.

Per migliorare l'efficacia nel scovare bug di progettazione, useremo la tecnica del **"Black Box"** dove useremo l'applicazione come se fossimo degli utenti qualunque, oppure verrà chiesto a una persona di terze parti di farlo.

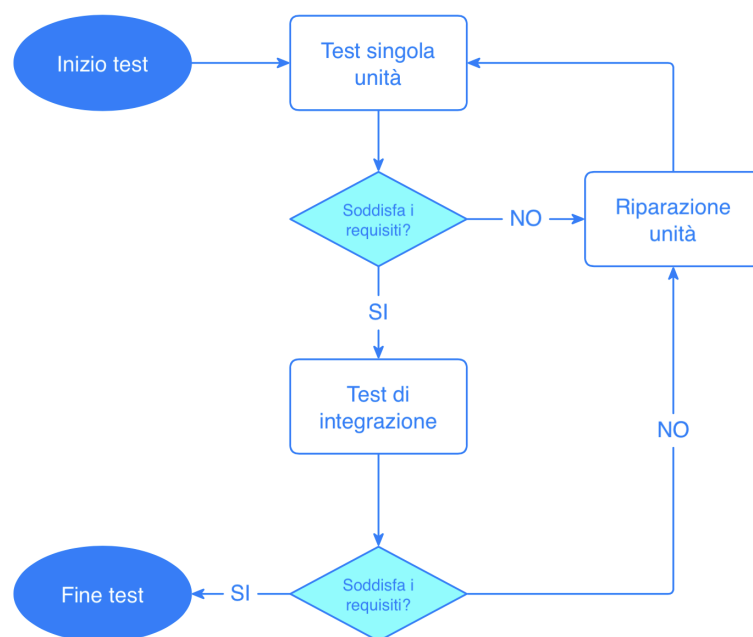


Tale test ci permette di rilevare bug di vario tipo più facilmente per poi permettere di risalire alla causa.

Per alcune funzionalità, come la fluidità dello scorrimento della mappa e la robustezza del database o che per lo più riguardano la performance dell'app, useremo un approccio di **stress-testing**.

Infine, per il test d'ambiente, apertura app e compatibilità utilizziamo un test di tipo **top-down**.

Per ogni test, la procedura sarà la seguente:



6.3. Pianificazione dell'esecuzione dei test

L'applicazione verrà testata in modo critico anche per quanto riguarda l'aspetto grafico e l'usabilità dell'applicazione stessa per l'utente medio, ovvero la User Experience.

I test verranno eseguiti in due fasi:

1. Inizialmente l'applicazione verrà testata dai membri del team e dagli sviluppatori che lavorano sul codice dell'app;
2. L'applicazione verrà poi testata da familiari / conoscenti dei membri del team per avere un feedback più ampio (fase fondamentale per avere un feedback da utenti totalmente estranei allo sviluppo);

I test verranno effettuati seguendo la tabella dei livelli di responsabilità (vedi [6.1](#)).



6.4. Struttura, dati e piano di fornitura delle risorse

Non viene definita una struttura precisa per effettuare i test.

I test verranno svolti al completamento dell'applicazione per individuare eventuali nuovi bug. I test verranno effettuati con l'utilizzo dell'emulatore di Android Studio e con ulteriori 4 dispositivi Android per un testing multiplatforma.

6.5. Strumenti di test

In questo paragrafo verranno dichiarati gli strumenti che saranno utilizzati dal team per i test.

Processo	Strumento
Creazione casi di test	Documenti Google - Piano di testing
Tracciamento casi di test	LibreOffice Calc
Esecuzione casi di test	Manualmente
Gestione casi di test	LibreOffice Calc
Gestione difetti	LibreOffice Calc

6.6. Verifica della procedura di testing

Il seguente documento verrà modificato per mantenere attiva la tracciabilità dei casi di test. I risultati della procedura di testing verranno analizzati dall'intero gruppo e se necessario verranno apportate delle modifiche al codice.

L'applicazione finale non dovrà contenere anomalie irrisolte per permettere all'utente di avere una buona esperienza all'interno dell'applicazione.



7. PROCEDURE DI REGISTRAZIONE DEI TEST

La seguente tabella contiene i test effettuati con i rispettivi risultati ottenuti, e verrà compilata man mano che procede il testing dell'applicazione.

Come già specificato, il supervisore del testing è Nicolò e i test sono stati effettuati da tutti i componenti del gruppo e da persone esterne al gruppo (beta testers).

Specifica testata	Caso d'uso	Modalità di test	Risultato atteso test
Test 00	Ambiente / Apertura app	Top - down	/
Test 01	Registrazione	Stress test	l'utente viene registrato con successo
Test 02	Login utente registrato	Stress test	l'utente registrato si logga con successo
Test 03	Login utente guest	Stress test	l'utente riesce ad accedere nell'applicazione come ospite con successo.
Test 04	Guida	Bottom-up	visualizzazione corretta della guida dopo il primo accesso all'app
Test 05	Visualizzazione delle notizie	Bottom-up	le notizie sono visualizzabili correttamente
Test 06	Visualizzazione della lista degli eventi	Bottom-up	gli eventi sono visualizzabili correttamente
Test 07	Visualizzazione dei posti da visitare	Bottom-up	i posti da visitare sono visualizzabili correttamente
Test 08	Visualizzazione della mappa e della propria posizione	Bottom-up	la mappa è visualizzabile in modo corretto
Test 09	Ricerca nella mappa	Bottom-up	la ricerca di un luogo nella mappa funziona correttamente con i luoghi



Test 10	Visualizzazione del profilo utente	Bottom-up	il profilo utente è visualizzabile correttamente
Test 11	Visualizzazione di una notizia	Bottom-up	una singola notizia è cliccabile e visualizzabile correttamente
Test 12	Visualizzazione di un evento	Bottom-up	l'evento singolo è visualizzato correttamente
Test 13	Visualizzazione di un luogo	Bottom-up	il luogo singolo si apre in dettaglio e mostra la sua descrizione, l'indirizzo e la foto



8. REQUISITI PER L'ATTIVITÀ DI TESTING

Prima di iniziare con i vari test, è necessario controllare i requisiti d'ambiente:

- **Dispositivi Android:** Essendo l'applicazione sviluppata interamente per dispositivi Android, è necessario che i tester abbiano tutti un telefono con questo sistema operativo.
- **Posizione:** I dispositivi utilizzati dovranno avere la possibilità di migliorare la precisione di localizzazione, senza questa opzione è possibile usare le mappe, ma è probabile che la geolocalizzazione sia meno precisa e di conseguenza l'utente farebbe fatica ad orientarsi a Venezia.
- **Ambiente di test:** L'area di test dovrà essere ben coperta dal segnale telefonico in modo da avere condizioni ottimali per quanto riguarda il segnale. Il primo test si svolgerà interamente in un'area all'aperto, mentre nel secondo, sarà possibile anche entrare negli edifici, in modo da vedere come si comporta l'applicazione a seconda della situazione.

8.1. Gestione dei difetti

Quando un tester trova un difetto in fase di test cosa fa?

Viene segnalato ai membri del gruppo che si occupano dello sviluppo del codice.

Il responsabile cerca di risolvere il problema eventualmente con l'aiuto degli altri membri del team.

Ogni settimana ci si trova in un meeting per aggiornarsi su ulteriori difetti e/o bug individuati e su una possibile soluzione di tali difetti

9. VINCOLI

Il test dell'applicazione deve essere concluso prima della pubblicazione della stessa nel Play Store.

Lo sviluppo dell'applicativo deve essere terminato con dovuto anticipo prima della data di consegna in modo da poter testare al meglio tutte le funzionalità.

10. RIFERIMENTI

Per creare questo documento sono stati utilizzati come riferimento:

- "piano di testing" di alcuni gruppi degli anni passati;
- documenti già stilati dal team (piano di progetto e documento di analisi e specifica);
- template di un "piano di testing" messo a disposizione dal professore.