21/01/2022







Corso

[CT0090] INGEGNERIA DEL SOFTWARE (CT3) - a.a. 2021- 22

Team  
****

Membri

ANDREA BRION 860595

FABIO DANESIN 882805

MARCO TAMISARI 865233  
TANJIN HABIBUR 875737

TOMMASO GOLFETTO 876451



**PIANO DI TESTING**

VERSIONE 1.1



**INDICE**

**1. INTRODUZIONE**

1.1*Analisi delle aree di test e del campo di applicazione*

**2. GLOSSARIO**

**3. TRACCIABILITA’ DEI REQUISITI**

**4. ELEMENTI TESTATI**

**5. SCHEDULE DEL TESTING**

**6. PROCEDURE DI REGISTRAZIONE DEI TEST**

**7. REQUISITI HARDWARE E SOFTWARE**

**8. VINCOLI PER I TESTING**

**9. RIFERIMENTI**



1. INTRODUZIONE

Questo documento racchiude per intero le modalità di verifica, le quali serviranno per testare il sistema e verificarne l’integrità.

Il team, attraverso le modalità di testing, identifica eventuali malfunzionamenti del sistema per garantire che sia in grado di soddisfare i punti del documento di analisi dei requisiti e specifica.

Verranno trattati nello specifico:

* Processo di testing, qui illustrate, dopo una breve descrizione delle procedure di testing a disposizione, le tipologie dei processi usate per effettuare le procedure scelte.
* Ambito, aree di interesse e obiettivi dei test.
* Rischi, problemi e dipendenze dei test.
* Responsabilità dei test.
* Tracciabilità dei requisiti, dove verranno analizzati i requisiti funzionali del sistema in accordo con le specifiche indicate nel documento di analisi e specifica in modo da essere certi riguardo la validità del requisito funzionale.
* Elementi testati: un elenco delle sezioni dell’applicazione da testare.
* Schedule di testing e milestones associate, qui vedremo l’organizzazione delle varie ’attività di testing in relazione non solo ad alpha e beta testing, ma anche la distribuzione delle risorse e del tempo dedicato ad ogni attività di controllo.
* Procedure di registrazione dei test, sezione dedicata agli esiti dei test effettuati ed eventuali annotazioni/appunti.
* Requisiti hardware e software utilizzati.
* Vincoli per il testing, qui vengono indicati i vincoli da rispettare per rendere l’attività di testing attendibile.

1.1 ANALISI DELLE AREE DI TEST E DEL CAMPO DI APPLICAZIONE

L’applicazione verrà sviluppata nei seguenti ambienti di sviluppo: Visual Studio Code e Webstorm, utilizzando le librerie principali di JavaScript.

Tale applicazione sarà un' App in grado di operare all’interno di uno smartphone Android dotato di connessione a internet e Geolocalizzazione (GPS, WI-FI) per permettere la lettura e scrittura di dati in un database.

L’applicazione deve essere in grado di:

* Ottenere i dati sulla propria posizione attraverso il GPS e di poterli usare, aggiornandoli in maniera continua.
* Connettersi al database esterno, leggere i contenuti interessati e aggiornarli.



2. GLOSSARIO

* **Android:** Sistema operativo basato su kernel Linux per dispositivi mobili, quali smartphone e tablet, nel nostro caso l’app sarà compatibile con le versioni 5.0 (Lollipop) in poi.
* **API:** 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 dotato di particolari procedure per raggiungere un determinato obiettivo.
* **Database:** Rappresenta la locazione in cui vengono memorizzati tutti i dati riguardante l’applicazione.
* **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 giocare.
* **JavaScript:** È un linguaggio di programmazione orientato agli oggetti e agli eventi, comunemente utilizzato nella programmazione web lato client(esteso poi anche a lato server) per la creazione di siti web e applicazioni web.

Fonte: <https://it.wikipedia.org/wiki/JavaScript>

* **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 verrannosuccessivamente installate, un sistema di sensoristica avanzato (gps, giroscopio,accelerometro, ecc.) ed una connessione ad internet per permettere la comunicazione con internet.
* **White-Box testing:** Forma di test che verifica il funzionamento interno di un componente software, piuttosto che le sue funzionalità.
* **Black-Box testing:** Il test effettuato accedendo al software solamente tramite l’interfaccia utente , oppure tramite l’interfaccia di comunicazione tra processi. Può essere manuale oppure automatico.
* **RF**:Irequisiti funzionali si presentano come elenchi di funzionalità o servizi che il sistema deve fornire. Essi descrivono anche il comportamento del sistema a fronte di particolari input e come esso dovrebbe reagire in determinate situazioni.
* **RNF**:I requisiti non funzionali rappresentano i vincoli e le proprietà/caratteristiche relative al sistema, come vincoli di natura temporale, vincoli sul processo di sviluppo e sugli standard da adottare. I requisiti non funzionali non riguardano solo il sistema software che si sta sviluppando, alcuni possono vincolare il processo usato per sviluppare il sistema.



3. TRACCIABILITA’ DEI REQUISITI

Qui verranno verificati i requisiti precedentemente identificati nel documento di analisi e specifica dei requisiti, seguendo le direttive ivi contenute (gli identificatori di questi saranno i medesimi).

Così facendo possiamo garantire la validità del requisito funzionale testato.

Nel testare i requisiti verrà implementato l’approccio Black-Box/White-Box.

| **ID requisito** | **Nome requisito** | **Test** | **Risultato** |
| --- | --- | --- | --- |
| **RF 01** | Geolocalizzazione | Test accuratezza della posizione, Unit Testing API | Tutti gli Unit Testing devono passare e le posizioni coincidere con la realtà |
| **RF 02** | Segnalazione positività/negatività | Ponendo l’utente positivo o negativo (guarito), gli utenti che hanno partecipato agli stessi eventi dell’utente verranno avvisati | Positivo se tutti gli utenti previsti ricevono l’avviso |
| **RF 03** | Suggerimento inserimento luogo | Creando un’attività proviamo diversi indirizzi e località | Tutti gli indirizzi presenti in Google maps devono ricevere suggerimenti |
| **RF 04** | Anteprima luogo | Accedendo ad un’attività dove il luogo è stato inserito verifichiamo la presenza dell’anteprima | L’anteprima deve essere presente e corretta, inoltre il link deve portare alla pagina di Google Maps del luogo indicato |
| **RNF 01** | Risposta rapida | Sistema testato in tutte le sue funzionalità per controllare che l’esecuzione delle stesse avvenga in un tempo limite | Positivo se l’esecuzione avviene entro 5 secondi |
| **RNF 02** | Modalità d’uso user-friendly | Vengono interrogate persone esterne al progetto  riguardo l’interfaccia e modalità d’uso | Positivo se l’utilizzatore riconosce un’interfaccia e modalità d’uso facili ed intuitive |
| **RNF 03** | Robustezza | Vengono verificate tutte le funzionalità dell’applicazione | Positivo se l’applicazione non subisce crash |



4. ELEMENTI TESTATI

Terminati gli sviluppi delle funzionalità prefissate, inizieranno le fasi di testing, per verificare che ognuna di esse agisca opportunamente.

Nel nostro caso, è stato deciso di dividere i test in due categorie: **alpha testing e beta testing**.

La prima, fa riferimento alle funzionalità più importanti e che richiederanno più tempo e risorse per essere testate, la seconda, invece, verte su tutto ciò che può essere migliorato nella fase finale del progetto.



5. SCHEDULE DEL TESTING

Di seguito, viene riportato uno schema per facilitare la comprensione del flusso di testing.



Per effettuare tutti i test, il team avrà bisogno di almeno una giornata di lavoro tra il dividersi degli alpha test e beta test, presumendo il successo di tutti questi.

Nel caso che i test riscontrino malfunzionamenti o anomalie, dovrà essere aggiunto del tempo per compierli di nuovo.

Per i costi teniamo lo studio effettuato nel piano di progetto considerando quindi un costo di 14€ per ora a persona, quindi ottenendo un costo di circa 118€.



6. PROCEDURE DI REGISTRAZIONE DEI TEST

I test verranno registrati tramite un modello comune a tutti i test, qui le specifiche fanno riferimento a quelle nel documento dei requisiti.

Ogni specifica può essere soggetta a vari test, di seguito l’archetipo da seguire:

| Specifica testata | Tester | Risultato | Annotazioni varie |
| --- | --- | --- | --- |
| ID requisito | Nome del componente che esegue il test | Risultato del test |  |
| ... | ... | ... |  |

I risultati dei test saranno oggetto di analisi per determinare una panoramica degli eventuali problemi che il sistema può sopportare, inoltre ogni membro riporta le proprie considerazioni, rispetto alle post-condizioni attese, per ogni test.



7. REQUISITI HARDWARE E SOFTWARE

I seguenti requisiti sono necessari sui propri dispositivi per poter testare l'applicazione:

* Sistema Android Lollipop o successivi (5.0).
* In alternativa è possibile scaricare Bluestacks sul proprio PC.

* Connessione ad internet.



8. VINCOLI PER I TESTING

Tutti i test dovranno essere conclusi entro la scadenza del 15/01/2022 e ovviamente dopo che le varie componenti del sistema sono state sistemate dal team in seguito alle modifiche effettuate.

Durante le quali la applicazione sarà migliorata nel caso i test segnino dei malfunzionamenti o anomalie per essere all’altezza delle azioni citate nel documento di analisi è specifica.



9. RIFERIMENTI

Per la realizzazione di questo documento sono stati utilizzati come riferimento:

* Documenti di alcuni gruppi degli anni precedenti.
* Materiale messo a disposizione del professore.
* Materiale ricercato in rete.

Per la realizzazione di questo documento sono state impiegate:

* Tommaso Golfetto (8h)
* Andrea Brion (8h)
* Marco Tamisari (8h)
* Tanjin Habibur (2h)

