

Piano di testing

Alessia Marostica
Diletta Oliaro
Gregory Sech
Simone Scaboro

31 gennaio 2018

Proposta Finale

Indice

1	Introduzione	3
2	Processo di testing	4
2.1	Descrizione Processo	4
2.2	Scheduling Processo	4
3	Tracciabilità dei requisiti	6
4	Elementi testati	7
5	Procedure di registrazione	8
6	Requisiti Hardware e Software	9
7	Vincoli	10
7.1	Vincoli temporali	10
7.2	Vincoli architetturali	10

Capitolo 1

Introduzione

Questo documento è finalizzato a descrivere la fase di testing. Di seguito la specifica delle differenti sezioni presentate:

Processo di testing

Presentazione delle diverse tipologie dei processi utilizzati e il loro scheduling (suddivisione del tempo) per poter effettuare la fase di testing.

Tracciabilità dei requisiti

Analisi di come le specifiche dell'applicazione descritte nel documento *analisi dei requisiti* sono state realizzate, indicando come vengono testate e cosa viene preso in esame.

Elementi testati

Elementi dell'applicazione che vengono sottoposti a maggiori controlli, a causa della loro criticità , e definizione di successo del test.

Procedure di registrazione

In questa sezione verranno descritte le modalità di verifica che verranno effettuate.

Requisiti Hardware e Software

Descrizione dei principali vincoli per compiere la fase di test sull'applicazione. Verranno presentati i vincoli temporali e architetturali.

Vincoli

Vincoli temporali e architetturali per poter effettuare i test.

Capitolo 2

Processo di testing

2.1 Descrizione Processo

Il processo di testing che intendiamo applicare è suddiviso in due parti che si conseguono temporalmente:

1. **Alfa Testing** Prima fase di testing con strategia *Bottom-Up*, ci aspettiamo di iniziare questa fase prima della terminazione del periodo di implementazione dell'applicazione. Sfruttando il paradigma Bottom-Up inizieremo a testare i moduli man mano che vengono implementati. Il testing in questa fase avverrà con dei *Walkthrough* al termine dell'implementazione di ogni modulo, se il modulo testato è dipendente da altri non ancora implementati essi verranno considerati funzionanti. Ulteriormente, prima del termine di questa fase, ogni modulo deve aver superato un *Black-Box* testing.
2. **Beta Testing** Seconda fase con strategia di *Stress testing*. Durante la fase di Beta Testing avverranno delle Ispezioni per simulare uno stress di utilizzo dell'applicazione. Un'ulteriore modalità di testing sarà rendere l'applicazione accessibile da utenti esterni, il che ci permetterà di ottenere dei *report* da parte degli utilizzatori.

Ora segue un glossario della terminologia utilizzata per descrivere il processo di testing:

- **Modulo** Componente atomica del progetto che fornisce delle funzionalità a delle entità che possono accedervi tramite una interfaccia.
- **Strategia Bottom-Up** Prevede di testare un modulo una volta che tutti i moduli sui quali dipende vengono testati.
- **Walkthrough** Metodologia statica di testing che prevede la simulazione dell'esecuzione del codice tramite una discussione di gruppo tra sviluppatori ed ispettori.
- **Black-Box Testing** Metodologia di testing che vede il modulo come una scatola nera, il modulo viene sottoposto a un test e vengono registrati i risultati da un tester. Dopo di che vengono comparati i risultati ottenuti con i risultati attesi dal test, in caso di incongruenza il test è fallito e lo sviluppatore dovrà provvedere alla correzione.
- **Stress Testing** Metodologia di testing con l'obiettivo finale di verificare che il sistema sia in grado di sopportare il carico massimo previsto in fase di progettazione.
- **Report** Notifica di comportamento non atteso (bug) inviata da un utilizzatore dell'applicazione agli sviluppatori.

2.2 Scheduling Processo

- **Alfa Testing** Questa fase inizierà ufficialmente il 26 Dicembre in questa fase avverranno i seguenti walkthrough, le date sono state scelte pensando al ritardo massimo atteso nella stesura della code-base del sistema dunque saranno attesi possibili anticipi:
 - **Walkthrough Requisito 4.1:** 27 Dicembre 2017

- **Walkthrough Requisito 4.2:** 1 Gennaio 2018
- **Walkthrough Requisito 4.3:** 3 Gennaio 2018
- **Walkthrough Requisito 4.4:** 5 Dicembre 2018
- **Walkthrough Requisito 4.5:** 7 Dicembre 2018

Successivamente al superamento di ogni walkthrough inizieranno dei periodi di Black-Box testing. Ci aspettiamo di riuscire per ogni requisito a superare i test black box entro la data prevista per la pubblicazione della versione beta: 13 Dicembre.

- **Beta Testing** Questa fase inizierà ufficialmente il 13 Dicembre dopo la pubblicazione della versione beta dell'applicazione sul PlayStore. Prevediamo di fare una ispezione per ogni modulo alle seguenti date:

- **Inspection Requisito 4.1:** 15 Gennaio 2018
- **Inspection Requisito 4.2:** 16 Gennaio 2018
- **Inspection Requisito 4.3:** 17 Gennaio 2018
- **Inspection Requisito 4.4:** 18 Gennaio 2018
- **Inspection Requisito 4.5:** 19 Gennaio 2018

Al termine di ogni Inspection e successivamente ad ogni modifica avverrà un nuovo test black box per il requisito.

In questa fase saremo anche esposti a dei bug report da parte di utenze esterne, sarà dunque compito degli sviluppatori correggere gli errori trovati e verificare sempre con dei test black box il corretto funzionamento dei vari requisiti prima di confermare la correzione dell'errore e dunque aggiornare la versione dell'applicazione.

Capitolo 3

Tracciabilità dei requisiti

I requisiti che verranno analizzati sono ripresi dal documento dei requisiti [Ref. *Documento dei requisiti Sezione 4*] precedentemente redatto. Verranno analizzati solo i requisiti funzionali suddivisi in macroaree e ci si riferirà ad essi tramite il numero di sottoparagrafo utilizzato nel sopracitato documento. Per ogni requisito verrà segnalato il relativo test che verrà effettuato .

Requisito	Test
4.1	Il requisito è verificato se l'utente riesce ad accedere correttamente senza la presenza di errori
4.2	Questi requisiti sono soddisfatti se l'utente riesce ad accedere con successo alle aree indicate
4.3	Requisiti superati se l'utente riesce a modificare i propri dati con successo
4.4	L'utente riesce a visualizzare i dati correnti e coerenti con la sua scelta
4.5	Requisito soddisfatto se l'utente riesce a trovare correttamente un avversario e giocare al gioco senza errori. Le domande devono essere create coerentemente a quanto specificato nei documenti precedenti

Capitolo 4

Elementi testati

Gli elementi testati saranno i seguenti:

1. Tempo di risposta delle seguenti operazioni:
 - Avvio dell'applicazione
 - Creazione delle domande nella sezione gioco
 - Caricamento della mappa iniziale
 - Caricamento di tutti i dati visibili all'utente
2. Usabilità dell'applicazione:
 - L'utente deve capire con semplicità come utilizzare le interfacce e tutti gli elementi al suo interno quindi attraverso gli utenti finali verrà testata la user experience
3. Stabilità dell'applicazione:
 - L'applicazione deve rispondere agli errori in modo rapido ed efficiente per mantenere la sua stabilità
4. Correttezza all'interno dell'applicazione di:
 - Contenuti: dati presentati agli utenti, domande generate nella sezione gioco, mappa e classifica dei punteggi
 - Operazioni: ogni azione fatta dall'utente deve corrispondere all'operazione richiesta al sistema

Capitolo 5

Procedure di registrazione

In questa sezione saranno descritte le modalità di verifica scelte per la fase di testing che saranno opportunamente documentate.

Inspection

Gli obiettivi principali sono quelli di rilevare la presenza di difetti ed eseguire una lettura mirata del codice scritto. La verifica del codice verrà effettuata da chi non ha scritto quella specifica parte per permettere una visione più oggettiva della revisione. La prima fase sarà quella della pianificazione, seguita dalla definizione dei punti da analizzare (errori più comuni o possibili errori) proseguendo poi con la lettura effettiva del codice e la correzione degli eventuali difetti ed errori. Questa verifica si basa su errori presupposti ovvero sulla lista di errori possibili che è stata redatta, si tratta quindi di una metodologia di verifica molto veloce. Per effettuare un processo di testing tramite inspection verranno utilizzati i seguenti strumenti:

- Checklist degli errori possibili

ID	Descrizione
1	Corretto utilizzo del ciclo di vita di Activity e Fragment
2	Strutturazione del modello dati in accordo con la documentazione Firebase
3	Corretta formazione di Intent per le chiamate ad Activity
4	Errato utilizzo di Transaction
5	Errato utilizzo di addChildEventListener
6	Errato utilizzo di addListenerForSingleValueEvent
7	Errato utilizzo di addValueEventListener
8	Corretto utilizzo del FragmentManager di Android per il cambio di Fragments
9	Corretto riferimento al giocatore durante le fasi di gioco, assegnamento dei punti e ricostruzione della partita
10	Corrispondenza della terminologia utilizzata nel codice Android e nel database di Firebase
11	Sfasamento dei widgets nelle schermate causate da diverse dimensioni display

- Form per la raccolta dati

Inspection Form

*Campo obbligatorio

Errore Trovato: *

La tua risposta

File: *

La tua risposta

Funzione: *

La tua risposta

Righe File: *

La tua risposta

Note:

La tua risposta

INVIA

Non inviare mai le password tramite Moduli Google.

Walkthrough

In questa fase di verifica gli obiettivi principali sono la rilevazione della presenza di difetti e l'esecuzione di una lettura critica del codice. La strategia da applicare è quella di percorrere il codice simulandone l'esecuzione. La prima fase è ancora una volta la pianificazione, segue poi la lettura del codice e la discussione tra i componenti del team per effettuare poi una correzione su tutti i vari errori. Questa verifica si basa sull'esperienza dei verificatori e quindi è un processo più lento rispetto all'inspection.

- Form per Walkthrough

Walkthrough Form

*Campo obbligatorio

File: *

La tua risposta

Function: *

La tua risposta

Rows: *

La tua risposta

Error Description: *

La tua risposta

INVIA

Non inviare mai le password tramite Moduli Google.

Capitolo 6

Requisiti Hardware e Software

Nella fase di testing verranno utilizzati i seguenti dispositivi hardware per verificare la compatibilità con i vari sistemi operativi Android:

- Samsung Galaxy A5 con sistema operativo Android 7.0 Nougat
- LG Nexus 5 con sistema operativo Android 6.0.1 Marshmallow
- Lenovo Tab2 A10-70 per testare la compatibilità su tablet Android, sistema operativo Android 6.0 Marshmallow
- Huawei p10 Lite con sistema operativo Android 7.0 Emotion UI 5.1 Nougat
- OnePlus X con sistema operativo Android 6.0.1 Oxygen

Per testare i minimum ed i target SDK e le altre versioni del sistema operativo saranno utilizzati dei software di emulazioni così da verificare le funzionalità anche su sistemi Android con versioni non presenti tra quelle elencate precedentemente.

Capitolo 7

Vincoli

7.1 Vincoli temporali

Con vincoli temporali si intendono tutti quei vincoli legati alle tempistiche di realizzazione della fase di testing. Tutti gli elementi testati verranno analizzati in contemporanea dai componenti del gruppo (sui diversi dispositivi [Ref. 6.0]) per verificare che tutti gli elementi siano correttamente coesi. I vincoli relativi al testing si distinguono in base alla fase di testing nella quale ci troviamo:

- **Alfa Testing**, durante questa fase il vincolo da rispettare è il termine della stesura della code-base da parte degli sviluppatori, le date descritte in [2.2] (Scheduling Processo) sono state scelte tenendo conto dell'analisi esposta nel Piano di Progetto dov'è stato tenuto conto del ritardo massimo accettabile.
- **Beta Testing**, durante questa fase ci aspettiamo di far durare una Inspection non più di una giornata, dunque ci aspettiamo di terminare l'arco di Inspections per tutti e cinque i requisiti in non più di cinque giorni. Fare sempre riferimento alle date in [2.2] (Scheduling Processo).

7.2 Vincoli architetturali

Verranno di seguito descritti i pre-requisiti necessari per la corretta procedura di testing.

- Connessione ad internet
- Utilizzo di un dispositivo che rispetti i requisiti hardware e software [Ref. 6.0]
- Utilizzo dell'App nella sua versione più recente