UNIVERSITÀ DEGLI STUDI DI SALERNO

Ingegneria del Software

Test Plan

ANNO ACCADEMICO 2017/2018

Versione 1.1

Partecipanti:

NOME	MATRICOLA
Della Porta Raffaele	0512102538

Revision History:

DATA	VERSIONE	DESCRIZIONE	AUTORE
16/02/18	1.0	Descrizione del Test plan con gli obiettivi prefissati.	Raffaele Della Porta
17/02/18	1.1	Revisione del Test Plan.	Raffaele Della Porta

Indice

1. Introduzione al documento

1.1 Acronimi ed Abbreviazioni

Nel presente documento verranno utilizzati i seguenti acronimi:
PFL: PlayForLearn.
TP: Test Plan.

1.2 Naming

Per una migliore gestione dei documenti di test che saranno prodotti, i documenti di test devono rispettare il naming come definito nella figura 1.1:

xxx indica la Feature.
yyy indica il test Suite.
zzz indica il test Case.

figura 1.1 Diagramma con regole di naming

Nel dettaglio:

- "PFL" identifica il progetto, ovvero PlayForLearn
- "xxx" identifica la *feature*, ovvero a quale task il documento di test fa riferimento. L'elenco delle feature ed il codice associato è reperibile nella sezione 2.5;
- "yyy" identifica i *test suite*;
- "xxx" identifica i *test case*;

1.3 Riferimenti

Per la realizzazione del sistema sono stati utilizzati i seguenti materiali di riferimento:

- Android guida: https://developer.android.com/index.html
- Android Programming: The Big Nerd Ranch Guide: http://www.bignerdranch.com/we-write/android-programming.html
- B.Bruegge, A.H. Dutoit, Object Oriented Software Engineering - Using UML, Patterns and Java, Prentice Hall, 3rd edition, 2009.

2. Test Plan

2.1 Purpose

Il presente documento di Test Plan (TP) del progetto PlayForLearn descrive lo scopo, gli approcci, le risorse e le tempistiche per le attività di testing. Inoltre identifica gli items e le features da testare, i testing task che devono essere effettuati, il rischio associato a questo stesso plan.

Il processo di testing segue di pari passo la metodologia di sviluppo agile che il team ha deciso di intraprendere (Scrum). Per ulteriori dettagli si rimanda alla sezione 2.7 "Approccio"

2.2 Outline

Il test plan presenta la seguente struttura:

- Introduzione;
- Test items;
- Features da testare;
- Features da non testare;
- Approccio;
- Item pass/fail criteria;
- Test deliverables
- Testing tasks and schedule
- Ambiente richiesto;
- Testing tasks and schedule

3 Introduzione al test plan

3.1 Obiettivi

Il presente documento di test plan per il PFL, ha i seguenti obiettivi:

- Dettagliare le attività richieste per preparare e condurre i test al sistema;

- Comunicare a tutti i responsabili i compiti che devono effettuare e la loro tempistica;
- Definire la provenienza delle informazioni per preparare il piano;
- Definire gli strumenti di test e l'ambiente necessario per effettuare i test al sistema;

3.2 Contesto

Il progetto PFL mira alla creazione di una serie di giochi da far utilizzare ai bambini per poterli istruire. Ha la caratteristica di essere giocato attraverso l'app, accedendo al server di gioco attraverso un programma client.

3.3 Test Items

Per *unità* si intende il più piccolo contenitore di lavoro per un singolo programmatore, che può essere facilmente tracciata e pianificata. Essendo in un contesto object-oriented, le unità sono identificabili come le *classi*. In questa sezione vengono riportati i componenti del sistema che devono essere testati.

3.4 Features da testare

La seguente lista descrive le features che saranno testate:

Test Unità

- **Feature001**: Visualizzazione disposizione degli elementi sullo schermo e controllo input sul login.

Test Funzionali

- **Feature002**: Gestione Creazione Account Utente.
- **Feature003**: Gestione test login.
- **Feature004 : Fase di New Game.**
- **Feature005 : Fase di Load Game.**

Altri Test

- **Feature006**: Database.

3.5 Approccio

Verranno ora trattati in dettaglio le tipologie di test utilizzate nel corso del processo di sviluppo.

Test di Unità: I test di questa tipologia vengono effettuati parallelamente allo sviluppo del sistema. I test di unità devono risultare sempre positivi.
I test suite di questa tipologia sono raccolti nel documento TestUnitàPFL..

Per ogni test suite deve essere specificato:

- Nome del test suite;
- Feature da testare (Classe);

per ogni metodo

- Nome metodo e descrizione;
- Procedura;
- Pass/Fail Criteria;

Test funzionale: Questa tipologia di test viene effettuata negli ultimi giorni di ogni *sprint* del processo scrum. Si dovrà verificare che ogni funzionalità completata funzioni correttamente in conformità alle specifiche; I test suite di questa tipologia sono raccolti nel documento TestFunzionalePFL.

Per ogni test suite deve essere specificato:

- Nome del test suite;
- Feature da testare;
- Procedura;
- Casi di test eseguiti;
- Pass/Fail Criteria;

Test di accettazione: Questa tipologia di test viene effettuata ad ogni nuova release del prodotto. Il sistema sarà fatto provare ai committenti che potranno verificare di persona se le funzionalità introdotte corrispondono ai loro requisiti.

3.6 Item pass/fail criteria

I criteri per considerare un test passato o fallito vengono singolarmente stabilite in ogni documento di test case.

3.7 Documenti di Test

I Documenti di Test devono essere inseriti nella cartella "TestingPFL", facendo riferimento ai file template appositamente creati. Tutti i file di una feature devono essere inseriti in una directory con nome **"Feature000"** dove "000" è il numero della feature.

3.8 Ambiente richiesto

E' stabilito che:

- Il linguaggio utilizzato dai test è **Java**;
- I **Test Funzionali** sono eseguiti (ove possibile)
- I **Test di Unità** sono eseguiti con il tool JUnit.
- I file dei test risiedono nelle cartella predisposte:
    - **Test Funzionali (files *.xml*)** : directory TestFunzionalePFL.
    - **Test Unità (files delle JUnit)**: directory TestUnitàPFL.