UNIVERSITÁ DEGLI STUDI DI SALERNO

Ingegneria del Software

TEST PLAN

ANNO ACCADEMICO 2018/2019

a

TOP MANAGER:

Nome
Prof. Andrea De Lucia

PARTECIPANTI:

Nome	Matricola
Mario Sessa	0512104650
Luigi Crisci	0512104740
Pasquale Ambrosio	0512104704

HISTORY:

Data	Versione	Cambiamenti	Autori
24/01/2019	1.0	Inserimento sezioni con esclusione dei Test Case	Mario Sessa
13/02/2019	1.1	Completamento del Test Plan	Mario Sessa
15/02/2019	1.2	Correzione Test Plan	Mario Sessa

Indice

1. Introduzione

Lo scopo del sistema è quello di pianificare l'attività di testing all'interno della piattaforma YouLearn con lo scopo di verificare se esistono differenze tra il comportamento atteso) con il comportamento osservato. In questa attività andremo ad individuare i possibili errori nel codice sorgente causanti incident all'interno del sistema prima che l'utente finale esegua delle interazioni con il sistema. Le attività di testing sono organizzate in modo da poter verificare il funzionamento dei seguenti sottosistemi:

1. Gestione utente
2. Gestione corso
3. Gestione lezioni

Si noti che non vi saranno attività di testing per i sottosistemi non descritti all'interno dell'Object design document e sviluppati in fase di implementazione. Quindi i sottosistemi seguenti saranno esclusi dalle verifiche delle funzionalità di sistema:

1. Gestione pagamenti
2. Gestione mail

I documenti precedenti al testing sono di grande importanza per la fase di testing, questo perché ogni documento, sviluppato secondo un differente livello di progettazione, saranno di grande importanza per la correttezza del testing.

2. Documenti correlati

La documentazione precedente è strettamente correlata con la pianificazione dei test, questo perché già nei documenti precedenti abbiamo definito come alcuni servizi devono funzionare. Questa caratteristica prevede di rilevare informazioni come i comportamenti attesi durante l'esecuzione di alcune funzionalità. Vediamo, quindi, che la fase di testing prevede di verificare se ci siano differenze tra il funzionamento del sistema che si è progettato e il reale funzionamento del sistema implementato.

2.1 Relazione con il documento di analisi dei requisiti (RAD)

La relazione tra la fase di testing e la fase di analisi dei requisiti si basano sui requisiti funzionali e non funzionali descritti nel RAD che devono essere rispettati dal sistema durante la fase di testing.

2.2 Relazione con il System design document (SDD)

La relazione tra la fase di testing e la fase di design del sistema si basa sulla suddivisione del sistema in sottosistemi e la divisione in strati (Presentation layer, Application layer, Storage layer). Il testing deve essere fedele alla suddivisione progettata in fase di design in modo tale da rimanere coerente con il sistema che si è progettati.

2.3 Relazioni con l'Object design document (ODD)

La relazione fra la fase di testing e la fase di object design si basa principalmente sul riferimento della verifica dei contratti e dei componenti raffinati all'interno di tale documento.

3 Panoramica del Sistema

La struttura definita dentro il System design document (SDD) prevede un architettura client-server three-trial basata sulla suddivisione in 3 strati:

- Presentation Layer – Strato che si occupa della presentazione dei risultati di una computazione agli utenti del sistema e di raccogliere gli input degli utenti
- Application layer – si occupa di fornire all'applicazione specifiche funzionalità
- Storage Layer – si occupa della gestione dei database di sistema

Il sistema, inoltre, è stato diviso in sottosistemi a secondo della gestione di una particolare categoria di servizi. La suddivisione è fatta nel seguente modo:

- Gestione utente
- Gestione corso
- Gestione lezioni
- Gestione pagamenti
- Gestione mail

Quasi tutte le gestioni prevedono operazioni di inserimento, modifica, cancellazione, visualizzazione e ricerca di dati. Saranno proprio tali funzionalità ad essere oggetto di testing del sistema. Ricordiamo, infine, che solamente i primi tre sottosistemi verranno testati prima del rilascio del progetto, qualsiasi servizio legato alla gestione dei pagamenti e di mail dipendenti dai sottosistemi da testare saranno sviluppati come stub o driver a secondo del tipo di dipendenza.

4 Funzionalità da testare e non testare

Di seguito saranno elencate tutte le funzionalità da testare in relazione al sottosistema a cui appartengono.

1. Gestione utente
    - Login
    - Cambio E-mail
    - Cambio Password
2. Gestione corsi
    - Crea corso
    - Modifica corso
    - Verifica corso
    - Visualizzazione corso
    - Iscrizione corso
3. Gestione lezioni
    - Elimina lezione
    - Inserimento lezione
    - Visualizza lezione
    - Inserimento commento
    - Cancellazione commento

5 Criteri Pass/Failed

I dati di test verranno raggrupati in classi di equivalenza in grado di raggruppare elementi con le stesse caratteristiche, per tali classi sarà sufficiente testare le funzionalità per un singolo elemento per classe. Un input supererà il test se il comportamento atteso e quello risultate combaceranno, ossia se l'output desiderato è uguale all'output ottenuto. Il responsabile del testing saprà a priori quale sia l'output desiderato.

6 Approccio

Verrà applicata una strategia di testing bottom-up. Si inzierà con il testing di unità dei singoli componenti, in modo da testare nello specifico le unità atomiche nella loro correttezza. Seguirà il testing di integrazione che focalizzerà l'attenzione sulle interfacce dell'unità. Infine, verrà eseguito il testing di sistema che vedrà la verifica del comportamento dell'interno sistema assemblato partendo dalle sue componenti principali. Il testing di sistema è importante per verificare se le caratteristiche richieste dal committente vengono rispettate o meno.

6.1 Testing di unità

Durante questa fase, verranno ricercate le condizioni di fallimento all'interno delle singole componenti e, usando test driver e stub raffiguranti implementazioni parziali di componenti che dipendono o da cui dipendono le componenti testate, si verificano i comportamenti della singola unità. La strategia utilizzata per il testing si baserà esclusivamente sulla tecnica Black-Box. Questa scelta strutturerà il testing unitario in un'analisi Input/Ouput delle singole componenti andando ad astrarre la verifica della struttura interna. Per minimizzare i casi di test, gli input verranno divisi in classi di equivalenza e ogni componente avrà un singolo caso di test per ogni classe di equivalenza strutturata. In questa fase, quindi, si avrà particolare attenzione sulla suddivisione delle classi degli input cosi da poter verificare ogni componente su ogni possibile tipo di input del dominio. La gestione del caso di errore, ossia lo stato in cui il comportamento atteso non è equivalente al comportamento ottenuto, comporterà un aggiornamento del documento di Incident Report; Tale documento verrà utilizzato per informare gli sviluppatori della presenza di un fault in modo tale da poterlo correggerlo tempestivamente per poi ripassare ad una verifica della correzione. La strategia del report di fault verrà estesa anche per le altre fasi di testing.

6.2 Testing di integrazione

Questa fase di test prevede l'aggregazione delle singole componenti e il loro testing adottando una strategia black-box. Tale fase prevede l'iterazione con l'interfaccia del sistema legato ai servizi delle funzionalità da testare, tale testing si basarà sulla ricerca di possibili fault all'interno delle funzionalità del sistema e, in particolare, nella logica applicativa del software.

6.3 Testing di sistema

In questa fase verrà utilizzata nuovamente la strategia black-box per il testing delle funzionalità dell'intero sistema e verificare se i requisiti e i vincoli di progettazione sono stati rispettati. In seguito a tale test, se superato in maniera corretta, si avrà un sistema pronto all'uso per l'utente finale. Ci si concentrerà principalmente sul testing delle funzionalità principali basate sulle priorità dei requisiti specificati durante la fase di analisi.

7 Sospensione e ripresa

7.1 Criteri di sospensione

La fase di testing di sistema verrà sospesa quando si raggiungerà un compromesso di qualità deò prodotto e costi delle attività di testing. Il testing verrà propagato il più possibile in modo da garantire una migliore affidabilità del funzionamento del sistema andando, però, a considerare i tempi di consegna del progetto.

7.2 Criteri di ripresa

I criteri di ripresa rappresentano le azioni da apportare in seguito ad una modifica o ad una correzione di qualche componente durante la fase di testing. Tale attività prevede la creazione di nuovi test case sottoposti nuovamente al sistema per verificare la correttezza della modifica apportata.

8 Materiale per il testing

L'hardware necessario per l'attività di testing deve essere necessariamente un pc. Non vi è bisogno di una connessione poiché ogni componente può essere testata in locale.

9 Test cases

9.1 Gestione utente

9.1.1 Login

9.1.1 Category Partition

Parametro: E-mail **Formato: [A-Za-z0-9.]@[a-z.]**	
Lunghezza [LU]	1. <8 [error] [ifformatoFUOK] 2. >=8 [ifFormatoFUK][property lunghezzaLUOK]
Esistenza [FU]	1. L'e-mail esiste nel sistema [ifLunghezzaLUOK][property EsistenzaFUOK] 2. L'e-mail non esiste nel sistema [iflunghezzaLUOK][error]

Parametro: Password	
Correttezza [CP]	1. La password non è corretta [error] 2. La password è corretta [property CorrettezzaCPOK]

9.1.1.2 Test Case

Codice	Combinazione	Esito
TC_1.1_1	LU1	Errato
TC_1.2_2	LU2, FU1	Errato
TC_1.3_3	LU2, FU2, CP1	Errato
TC_1.4_4	LU2, FU2, CP2	Corretto

9.1.2 Cambio e-mail

9.1.2.1 Category Partition

Parametro: E-mail **Formato: [A-Za-z0-9.]@[a-z.]**	
Lunghezza [LRC]	1. <8 [error] [ifformatoFRCOK] 2. >=8 [ifFormatoFUK][property lunghezzaLRCOK]
Formato [FRC]	1. rispecchia il formato [propertyformatoFRCOK, rispecchia il formato [A-Za-z0- 9.]@[a-z.]] 2. non rispetta il formato [error]
Correttezza [DBNotPresent]	1. L'email non deve essere presente all'int65erno del database [ifLunghezzaLRCOK][property DBNotPresentOK] 2. L'email è già presente dentro il DB[ifLunghezzaLRCOK][error]

9.1.2.2 Test Case

Codice	Combinazione	Esito
TC_1.2_1	FRC1	Errato
TC_1.2_2	FRC2, LRC1	Errato
TC_1.2_3	FRC2, LRC2, DBNotPresent1	Errato
TC_1.2_4	FRC2, LRC2, DBNotPresent2	Corretto

9.1.3 Modifica Password

9.1.3.1 Category Partition

Parametro: NuovaPassword	
Lunghezza [LPM]	2. >=8 and <= 45 [ifValoreVPROK] [property lunghezzaLPROK] 3. <8 and > 45[ifValoreVPROK] [error]
Valore (VPRM)	1. != RipetiPassword [error] 2. == RipetiPassword [propertyvaloreVPROK, è uguale al valore Password successivo]

Parametro: ConfermaPassword	
Valore (VPRM)	1. == Password [propertyvaloreVRPOK, è uguale al valore Password precedente] 2. != Password [iflunghezzaLRPOK] [error]

9.1.3.2 Test Case

Codice	Combinazione	Esito
TC_1.3_1	LPM1	Errato
TC_1.3_2	LP2, VPRM1	Errato
TC_1.3_3	LP2, VPRM2	Corretto

9.2 Gestione corso

9.2.1 Crea corso

9.2.1.1 Category Partition

Parametro: TitoloCorso Formato: [A-Za-z]	
Lunghezza [LTCC]	1. <5 and > 30 [error] 2. >=5 and <= 30 [property lunghezzaLTCCOK]
Formato [FTCC]	1. rispecchia il formato [iflunghezzaFTCCOK] [propertyformatoFTCCOK, rispecchia il formato [A-Za-z]] 2. non rispetta il formato [iflunghezzaLTCCOK] [error]

Parametro: Descrizione	
Lunghezza [LTCD]	1. <10 and > 1048 [error] 2. >=10 and <= 1048 [property lunghezzaLTCDOK]

Parametro: ScadenzaCorso Formato: DD/MM/AAAA	
Settato [LNSCC]	1. La data viene settata [property SettatoLNSCCOK] 2. La data non viene settata [error]
Corretto [CSCC]	1. La data deve essere successiva a quella odierna e l'anno compreso tra 2019 e 2030 [ifSettatoLNSCCOK][property CorrettoCSCCOK] 2. La data è precedente o uguale alla data odierna oppure l'anno è inferiore al 2019 e superiore al 2030[ifSettatoLNSCCOK] [error]

Parametro: Prezzo Formato: [0-9]	
Formato [FPC]	1. rispecchia il formato [[0-9] [propertyformatoFPCOK] 2. non rispetta il formato [error]

Parametro: ImmagineCopertina	
Inserito [INIMG]	1. File inserito [OK] 2. File non inserito [error]

9.2.1.2 Test Case

Codice	Combinazione	Esito
TC_1.4_1	LTCC1	Errato
TC_1.4_2	LTCC2, FTCC1	Errato
TC_1.4_3	LTCC2, FTCC2, LTCD1	Errato
TC_1.4_4	LTCC2, FTCC2, LTCD2, LNSCC1	Errato
TC_1.4_5	LTCC2, FTCC2, LTCD2, LNSCC2, CSCC1	Errato
TC_1.4_6	LTCC2, FTCC2, LTCD2, LNSCC2, CSCC2, FPC1	Errato
TC_1.4_7	LTCC2, FTCC2, LTCD2, LNSCC2, CSCC2, FPC2, INIMG1	Errato
TC_1.4_8	LTCC2, FTCC2, LTCD2, LNSCC2, CSCC2, FPC2, INIMG2	Corretto

9.2.2 Modifica corso

9.2.2.1 Category Partition

Parametro: TitoloCorso Formato: [A-Za-z]	
Lunghezza [LTCCM]	1. <5 and > 30 [error] 2. >=5 and <= 30 [property lunghezzaLTCCMOK]
Formato [FTCCM]	1. rispecchia il formato [iflunghezzaFTCCMOK] [propertyformatoFTCCMOK, rispecchia il formato [A-Za-z]] 2. non rispetta il formato [iflunghezzaLTCCMOK] [error]

Parametro: Descrizione	
Lunghezza [LTCDM]	1. <10 and > 1048 [error] 2. >=10 and <= 1048 [property lunghezzaLTCCMOK]

Parametro: ScadenzaCorso Formato: DD/MM/AAAA	
Settato [LNSCCM]	1. La data viene settata [property SettatoLNSCCOK] 2. La data non viene settata [error]
Corretto [CSCCM]	1. La data deve essere successiva a quella odierna [ifSettatoLNSCCOK][property CorrettoCSCCOK] 2. La data è precedente o uguale alla data odierna[ifSettatoLNSCCOK] [error]

Parametro: Prezzo Formato: [0-9]	
Formato [FPCM]	1. rispecchia il formato [[0-9] [propertyformatoFPCMOK] 2. non rispetta il formato [error]

Parametro: ImmagineCopertina	
Inserito [INIMGM]	1. File inserito [OK] 2. File non inserito [error]

9.2.2.2 Test Case

Codice	Combinazione	Esito
TC_1.5_1	LTCCM1	Errato
TC_1.5_2	LTCCM2, FTCCM1	Errato
TC_1.5_3	LTCCM2, FTCCM2, LTCDM1	Errato
TC_1.5_4	LTCCM2, FTCCM2, LTCDM2, LNSCCM1	Errato
TC_1.5_5	LTCCM2, FTCCM2, LTCDM2, LNSCCM2, CSCCM1	Errato
TC_1.5_6	LTCCM2, FTCCM2, LTCDM2, LNSCCM2, CSCCM2, FPCM1	Errato
TC_1.5_7	LTCCM2, FTCCM2, LTCDM2, LNSCCM2, CSCCM2, FPCM2, INIMGM1	Errato
TC_1.5_8	LTCCM2, FTCCM2, LTCDM2, LNSCCM2, CSCCTM2, FSCM2, INIMGM2	Corretto

9.2.2 Iscrizione Corso

9.2.2.1 Category Partition

Parametro: CodiceDiSicurezza Formato: [0-9]	
Lunghezza [LUSEC]	1. !=3 [error] 2. ==3 [property lunghezzaLUSECOK]
Formato [FCSEC]	1. Rispecchia il formato [iflunghezzaLUSECOK][propertyformatoFCSECOK, rispecchia il formato [0-9] 2. Non rispecchia il formato [iflunghezzaLUSECOK] [error]

9.2.2.1 Test Case

Codice	Combinazione	Esito
TC_1.6_1	LUSEC1	Errato
TC_1.6_2	LUSEC2, FCSEC1	Errato
TC_1.6_3	LUSEC2, FCSEC2	Corretto

9.3 Gestione lezioni

9.3.1 Inserimento lezione

9.3.1.1 Category Partition

Parametro: NomeLezione Formato: [A-Za-z]	
Lunghezza [LTCL]	1. <5 and > 30 [error] 2. >=5 and <= 30 [property lunghezzaLTCLOK]
Formato [FTCL]	1. rispecchia il formato [iflunghezzaFTCLOK] [propertyformatoFTCLOK, rispecchia il formato [A-Za-z]] 2. non rispetta il formato [iflunghezzaLTCLOK] [error]

Parametro: VideoLezione Formato: [A-Za-z].[mp4]	
Inserito [INVID]	1. Inseriti da 1 a 500 MB di file video nel campo [OK] 2. 0 MB di file video presenti nel campo[error]

9.3.1.2 Test Case

Codice	Combinazione	Esito
TC_1.7_1	LTCL1	Errato
TC_1.7_2	LTCL2, FTCL1	Errato
TC_1.7_3	LTCL2, FTCL2, INVID1	Errato
TC_1.7_4	LTCL2, FTCL2, INVID2	Corretto

9.3.2 Inserimento commento

9.3.2.1 Category Partition

Parametro: Testo	
Lunghezza [LTCLC]	1. Lunghezza = 0 [error] 2. Lunghezza > 0 [property LunghezzaLTCLO]

9.3.2.2 Test Case

Codice	Combinazione	Esito
TC_1.8_1	LTCLC1	Errato
TC_1.8_2	LTCLC2	Corretto

10 Pianificazione dei test

Il team per il testing deve essere composto da persone che hanno una completa conoscenza del dominio applicativo e del dominio delle soluzioni del sistema. Inoltre, devono sapere in maniera completa tutte le tecniche di testing utilizzate e nominate all'interno del Test Plan e Test case specification. Le attività che comportano tale fase devono essere fatte nei tempi, nei costi e nei vincoli di qualità specificati. Solitamente gli sviluppatori e i tester non sono relazionati alla stessa persona, l'unico caso in cui tali ruoli combaciano con la stessa persona è quando si verifica un fault che porta ad un massivo cambiamento di una funzionalità. Il team dedicato al controllo dei vincoli di qualità sarà anche responsabile delle attività di testing e, quindi, della ricerca di possibili fault presenti nel sistema. La documentazione dei fault trovati verrà inviata agli sviluppatori per consentire la correzione del sistema. Il sistema revisionato dovrà, successivamente alla correzione, essere verificato attraverso altri casi di test per consentire di assicurarsi che le modifiche sono state compiute in maniera corretta e verificare se tali cambiamenti hanno introdotto nuovi errori. L'attività di testing è fondamentale nello sviluppo di un sistema software in quanto la mancanza di tale attività o una cattiva gestione di essa può portare al completo fallimento del sistema e, in casi estremi, dell'intero progetto. Data l'importanza del testing, la schedulazione delle sue attività sono fondamentali.

10.1 Determinazione dei ruoli

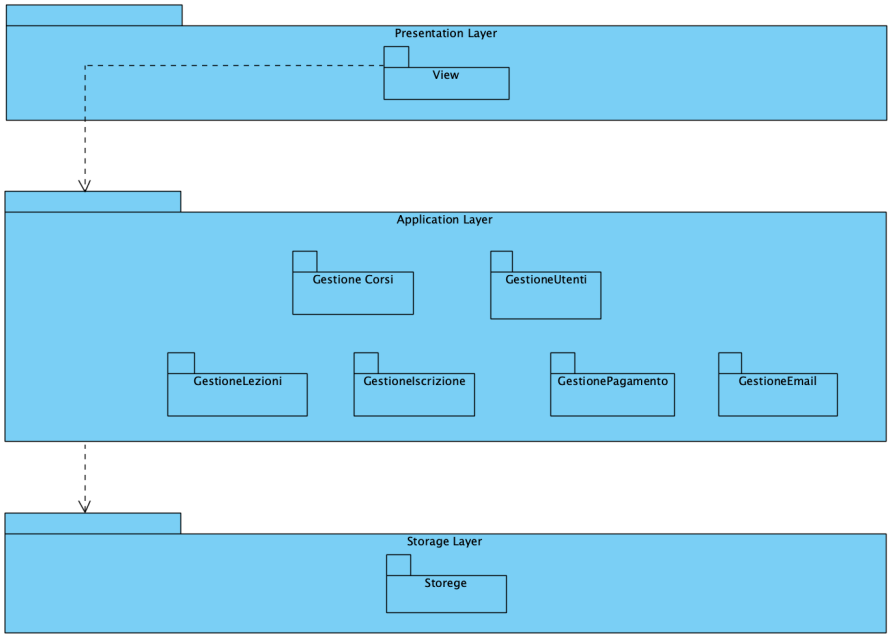
Tutta l'attività di testing viene eseeguita da Mario Sessa. Le attività relative al testing di unità come correzioni e cambiamenti verranno delegate agli sviluppatori che si occuperanno delle modifiche a livello implementativo. Tale organizzazione serve principalmente per allegerire il carico di lavoro da parte del tester che potrà dedicarsi maggiormente sul lavoro di testing funzionale.

10.2 Determinazione dei rischi

I rischi di un completo fallimento che prevedono la presenza di una quantità di errori elevata, può portare ad un ritardo del progetto. Per risolvere questa situazione, sempre se si verifichi, si è deciso di effettuare una pianificazione verticale dei test funzionale. Tale approccio ci permetterà di rilasciare un numero minore di funzionalità nei tempi previsti ma in maniera completa e funzionale.

10.3 Decomposizione gerarchica del sistema

La divisione gerarchica del sistema è stata mappata in 3 livelli gerarchici come nel seguente diagramma:



10.4 Organizzazione delle attività di testing

Le attività di testing verranno organizzate secondo uno schema che effettuerà una divisone funzionale di tipo verticale. In questo modo al termine di ogni attività si avrà una funzionalità completamente testata nei suoi livelli gerarchici. I vantaggi principali sono che in caso di ritardi dovuti al ritrovamento di numerosi failure il sistema verrà rilasciato con mno componenti, ma interamente testate e funzionanti.