



Università degli Studi di Salerno

Corso di Ingegneria del Software

Time-Platform Test Plan Versione 2.0



Progetto: Time-Platform	Versione : 2.0
Documento: Test plan	Data : 08/01/2016

Partecipanti:

Nome	Matricola
Angellotti Fabio [AF]	0512103314
Angellotti Francesco [AFR]	0512101988
Milito Mariano [MM]	0512101430

Scritto da:	Angellotti Fabio
--------------------	------------------

Revision history

Data	Versione	Descrizione	Autore
4/01/2016	1.0	Prima bozza del file	[AF]
5/01/2016	1.1	Scelta approccio per il test	[AF],[AFR]
6/01/2016	1.2	Revisione introduzione e approccio	[AFR],[MM]
7/01/2016	1.3	Creazione tabelle test	[AF],[AFR],[MM]
8/01/2016	2.0	Revisione completa del documento	[AF],[AFR],[MM]

Indice

1. Introduzione
2. Relazione con altri documenti
3. Panoramica del Sistema
4. Funzionalità da Testare e non
5. Pass/fail criteria
6. Approccio
 - 6.1. Testing di Unità
 - 6.2. Testing di accettazione
7. Test cases
8. Specifica dei test Cases

Introduzione

Il test è un procedimento che fa parte del ciclo di vita del software utilizzato per individuare le carenze di correttezza, completezza ed affidabilità; Ai fini di ciò presentiamo questo documento con lo scopo di gestire e analizzare le attività di testing riguardanti Time-Platform e i relativi sviluppi. L'obiettivo primario è quello di trovare possibili bug nelle componenti del sistema. In base a questi risultati sarà possibile capire dove bisognerà intervenire, e quindi apportare eventuali modifiche e correggere eventuali errori, così da migliorare i vari sottosistemi e rendere efficiente il prodotto. L'iterazione del processo sarà effettuata fino al momento in cui non si otterranno gli esiti richiesti rispettando i tempi di sviluppo previsti.

Relazione con altri documenti

Per la documentazione di ogni test descritto si rimanda al documento TCS_Time-Platform, esso contiene gli input e gli output attesi.

Panoramica del sistema

Time-Platform è una piattaforma web-based per la gestione dei turni di lavoro dei propri dipendenti, che in particolare permette ad ogni utente del servizio di accedere al sistema, previa autenticazione, e di controllare i turni e gli orari lavorativi. Così facendo si ottiene la possibilità di interagire istantaneamente con l'amministrazione affinché vi sia una corretta e veloce gestione dei turni onde evitare perdite di tempo e probabili gap amministrativi.

Funzionalità da testare e non

Le componenti che saranno testate sono:

- Login
- Invio richiesta ferie
- Invio SOS
- Invio richiesta cambio linea
- Invio richiesta cambio orario
- Invio richiesta cambio turno
- Invio avviso malattia

Le componenti che non saranno testate sono :

- Area amministratore
- Area capo fabbrica
- Avvisi Broadcast

Pass/fail criteria

L'obiettivo della fase di test è quello di progettare i test in modo tale da riscontrare delle discordanze fra il comportamento atteso e quello implementato, quindi si cerca di dimostrare che l'implementazione sia inconsistente con il modello del sistema.

Quindi in questa fase devono essere trovati il maggior numero di errori che poi dovranno essere corretti dagli sviluppatori.

Approccio

I casi di test sono progettati sulla base della descrizione del sistema, ovvero partendo dal documento di specifiche del sistema; l'approccio che verrà utilizzato è **Black box testing**, quindi non sarà controllata la struttura interna del sistema ma il comportamento di esso in base agli input e output.

Dall'insieme dei dati di input possibili si individua il sottoinsieme che può rivelare la presenza di difetti nel sistema in modo da progettare casi di test efficaci.

Individuati i bug e successivamente corretti, le componenti saranno pronte per essere integrate in sottosistemi più grandi. Utilizzando un approccio **Big-bang** le componenti corrette saranno testate individualmente e poi tutte insieme come un unico sistema, nel caso in cui si dovessero riscontrare nuovi bug, non individuati durante il test di unità (black box), verranno aggiunte nuove componenti per correggerli.

Dopodiché il sistema sarà testato nella sua completezza per assicurarci che sia conforme ai requisiti funzionali.

Test Case

Per selezionare gli input per il test del sistema verrà effettuata la partizione in classi equivalenti. Gli input e output saranno raggruppati in classi in cui tutti i membri di una classe sono relazionati, cioè il programma effettua elaborazioni simili per ciascun membro della stessa classe.

Specifica dei Test cases

TC_1.0 Login

Parametri: Matricola. Formato: [1-9]{4}	
Categorie	Scelte
Lunghezza Lm	1:Lunghezza ==0 [errore] 2:lunghezza==4 [property lunghezzaLmOk] 3:Lunghezza>4 [errore]
Formato Fm	1:rispetta il formato [if lunghezzaLmOk][property formatoFmOk] 2: non rispetta il formato [if !lunghezzaLmOk][errore]
Esiste Em	1: esiste nel DB [if lunghezzaLmOk and formatoFmOk][property EsisteEmOk] 2: non esiste nel DB [if lunghezzaLmOk

	and !formatoFmOk][errore]
--	---------------------------

Parametri: Password Formato: [az-AZ-09]{8}	
Categorie	Scelte
Lunghezza Lp	1:Lunghezza ==0 [errore] 2:lunghezza==8 [property lunghezzaLpOk] 3:Lunghezza>8 [errore]
Formato Fp	1:rispetta il formato [if lunghezzaLpOk][property formatoFpOk] 2: non rispetta il formato [if !lunghezzaLpOk][errore]
Esiste Ep	1: esiste nel DB [if lunghezzaLpOk and formatoFpOk][propety EsisteEpOk] 2: non esiste nel DB [if lunghezzaLpOk and !formatoFpOk][errore]

Codice	Combinazione	Esito
TC_1.0_1	Lm1	Errore
TC_1.0_2	Lm3	Errore
TC_1.0_3	Lp1	Errore
TC_1.0_4	Lp3	Errore
TC_1.0_5	Lm2 Fm1 Em2	Errore
TC_1.0_6	Lp2 Fp2 Ep2	Errore
TC_1.0_7	Lm2 Fm1 Em1 Lp1 Fp2 Ep2	Errore
TC_1.0_8	Lm2 Fm1 Em1 Lp3 Fp2 Ep2	Errore
TC_1.0_9	Lm1 Lm2 Em2 Lp1 Fp2 Ep2	Errore
TC_1.0_10	Lm1 Lm2 Em2 Lp3 Fp2 Ep2	Errore
TC_1.0_11	Lm2 Fm1 Em1 Lp2 Fp1 Ep1	Login Effettuato

TC_1.1 Invio SOS

Parametri: Descrizione. Formato: [az-AZ-1-9]{200}	
Categorie	Scelte
Lunghezza Ld	1:lunghezza==0 lunghezza==200 [property lunghezzaLdOk] 2:Lunghezza>200 [errore]
Formato Fd	1:rispetta il formato [if lunghezzaLdOk][property formatoFdOk] 2: non rispetta il formato [if !lunghezzaLdOk][errore]

Codice	Combinazione	Esito
TC_1.1_1	Ld2	Descrizione tagliata.
TC_1.1_2	Fd2	Descrizione tagliata.
TC_1.1_3	Ld1 Fd1	OK

TC_1.2 Invio richiesta cambio linea

Parametri: Descrizione. Formato: [az-AZ-1-9]{200}	
Categorie	Scelte
Lunghezza Ld	1:lunghezza==0 lunghezza==200 [property lunghezzaLdOk] 2:Lunghezza>200 [errore]
Formato Fd	1:rispetta il formato [if lunghezzaLdOk][property formatoFdOk] 2: non rispetta il formato [if !lunghezzaLdOk][errore]

Codice	Combinazione	Esito
TC_1.2_1	Ld2	Descrizione tagliata.
TC_1.2_2	Fd2	Descrizione tagliata.
TC_1.2_3	Ld1 Fd1	OK

TC_1.3 Invio richiesta cambio orario

Parametri: Descrizione. Formato: [az-AZ-1-9]{200}	
Categorie	Scelte
Lunghezza Ld	1:lunghezza==0 lunghezza==200 [property lunghezzaLdOk] 2:Lunghezza>200 [errore]
Formato Fd	1:rispetta il formato [if lunghezzaLdOk][property formatoFdOk] 2: non rispetta il formato [if !lunghezzaLdOk][errore]

Codice	Combinazione	Esito
TC_1.31_1	Ld2	Descrizione tagliata.
TC_1.3_2	Fd2	Descrizione tagliata.
TC_1.3_3	Ld1 Fd1	OK

TC_1.4 Invio richiesta cambio turno

Parametri: Descrizione. Formato: [az-AZ-1-9]{200}	
Categorie	Scelte
Lunghezza Ld	1:lunghezza==0 lunghezza==200 [property lunghezzaLdOk] 2:Lunghezza>200 [errore]
Formato Fd	1:rispetta il formato [if lunghezzaLdOk][property formatoFdOk] 2: non rispetta il formato [if !lunghezzaLdOk][errore]

Codice	Combinazione	Esito
TC_1.4_1	Ld2	Descrizione tagliata.
TC_1.4_2	Fd2	Descrizione tagliata.
TC_1.4_3	Ld1 Fd1	OK

TC_1.5 Invio Avviso malattia

Parametri: Descrizione. Formato: [az-AZ-1-9]{200}	
Categorie	Scelte
Lunghezza Ld	1:lunghezza==0 lunghezza==200 [property lunghezzaLdOk] 2:Lunghezza>200 [errore]
Formato Fd	1:rispetta il formato [if lunghezzaLdOk][property formatoFdOk] 2: non rispetta il formato [if !lunghezzaLdOk][errore]

Codice	Combinazione	Esito
TC_1.5_1	Ld2	Descrizione tagliata.
TC_1.5_2	Fd2	Descrizione tagliata.
TC_1.5_3	Ld1 Fd1	OK