



UNIVERSITÀ DEGLI STUDI
DI SALERNO

INGEGNERIA DEL SOFTWARE

A.A 2019/2020



Professore:

Andrea De Lucia

Alunni:

Antonio De Matteo

Girolamo Giordano

1. Introduzione

Lo scopo di questo documento è di pianificare l'attività di test del sistema ETransfer al fine di verificare se esistono differenze tra il comportamento atteso e il comportamento osservato. In questa attività andremo a rilevare gli eventuali errori prodotti all'interno del codice, per evitare che essi si presentino nel momento in cui il sistema verrà utilizzato dall'utente finale. Le attività di test sono state pianificate per le seguenti gestioni:

1. Gestione autenticazione
2. Gestione registrazione
3. Gestione account
4. Gestione autobus
5. Gestione fermata
6. Gestione corsa
7. Gestione tratta

Si noti, tuttavia, che verranno testate esclusivamente le funzionalità implementate e specificate nell'ODD. Oltre alla gestione dei test delle funzionalità, vengono anche pianificate le responsabilità del team e lo scheduling del test. La fase di testing è strettamente legata alle fasi ad essa precedenti; ogni documento, risultato delle differenti fasi di sviluppo, sarà un punto di partenza indispensabile per poter effettuare un testing corretto e adeguato.

2. Documenti correlati

Il test plan ha ovviamente una stretta relazione con il resto dei documenti che sono stati prodotti finora, poiché prima di passare alla fase di testing, oltre ad aver implementato il sistema nella gran parte, esso era stato pianificato nei precedenti documenti. Questo quindi permette di rilevare le eventuali differenze tra il comportamento atteso e quello osservato del sistema. Di seguito verranno riportate le relazioni tra il test plan e la documentazione precedente.

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

La relazione tra test plan e RAD riguarda in particolare i requisiti funzionali e non funzionali del sistema poiché i test che saranno eseguiti su ogni funzionalità terranno conto delle specifiche espresse nel RAD.

2.2 Relazioni con il System Design Document (SDD)

Nel System Design Document abbiamo suddiviso il nostro sistema in sottosistemi e l'architettura in tre livelli: Presentation Layer, Application Layer e Storage Layer. Il test dei vari componenti deve rimanere fedele a queste suddivisioni il più possibile.

2.3 Relazioni con l'Object Design Document (ODD)

Il test d'integrazione farà quanto più riferimento possibile alle interfacce delle classi definite nell'ODD.

3. Panoramica del sistema

Come stabilito nel System Design Document la struttura del nostro sistema è divisa secondo una architettura "Three Layers" cioè a tre livelli: Presentation Layer, Application Layer, Storage Layer. In questo caso il livello più alto interagisce con il livello applicativo che a sua volta si occuperà di eseguire le operazioni nel database di ETransfer, cercando di garantire il più possibile basso accoppiamento e alta coesione tra le varie classi. Il sistema inoltre è stato suddiviso in sottosistemi più piccoli, in particolare è stato diviso per gestioni, definite al paragrafo 1 di questo documento.

Quasi ognuna di queste gestioni prevede principalmente operazioni di inserimento, modifica, cancellazione, visualizzazione e ricerca e saranno proprio queste funzionalità ad essere testate nel corso della fase di testing del sistema.

4. Funzionalità da testare e da non testare

Di seguito saranno elencate per ogni gestione quali sono le funzionalità che saranno testate.

1. Gestione autenticazione

1.1. Login

2. Gestione registrazione

2.1 Registrazione

3. Gestione account

3.1. Modifica password

3.2. Modifica email

3.3. Modifica indirizzo

4. Gestione autobus

4.1. Aggiungi Autobus

5. Gestione fermata

5.1. Crea Fermata

5.2. Richiedi fermata

6. Gestione corsa

6.1. Aggiungi corsa

6.2. Richiedi corsa

7. Gestione tratta

7.1. Aggiungi tratta

7.2. Richiedi tratta

5. Criteri Pass/Failed

I dati di input del test saranno suddivisi in classi di equivalenza, ovvero verranno raggruppati in insiemi dalle caratteristiche comuni, per i quali sarà sufficiente testare un solo elemento rappresentativo. Un input avrà superato un test se l'output risultante sarà quello atteso, cioè quello che è stato specificato dal membro del team che si occuperà del testing su tale test case, il responsabile del testing conosce quale dovrebbe essere l'output corretto.

6. Approccio

Le tecniche di testing adottate riguarderanno inizialmente il testing di unità dei singoli componenti, in modo da testare nello specifico la correttezza di ciascuna unità. Seguirà il testing di integrazione, che focalizzerà l'attenzione principalmente sul test delle interfacce delle suddette unità. Infine verrà eseguito il testing di sistema, che vedrà come oggetto di testing l'intero sistema assemblato nei suoi componenti. Quest'ultimo servirà soprattutto a verificare che il sistema soddisfi le richieste del committente.

6.1 Testing di Unità

Durante questa fase, verranno ricercate le condizioni di fallimento, isolando i componenti ed usando test driver e stub, cioè implementazioni parziali di componenti che dipendono o da cui dipendono le componenti da testare. La strategia utilizzata per il testing si baserà esclusivamente sulla tecnica Black-Box, che si focalizza sul comportamento Input/Output, ignorando la struttura interna della componente. Al fine di minimizzare il numero di test cases, i possibili input verranno partizionati in classi di equivalenza e per ogni classe verrà selezionato un test case. Gli stati erronei scovati in questa, come in qualsiasi altra fase di testing, che comporteranno un fallimento del sistema dovranno essere tempestivamente comunicati agli sviluppatori al fine di correggerli e ripristinare il testing al più presto, per comunicare i vari fallimenti identificati durante la fase di testing si utilizzerà un test incident report.

6.2 Testing di Integrazione

In questa fase si procederà all'integrazione delle componenti di una funzionalità che verranno testate nel complesso attraverso una strategia Bottom-Up. Si passerà, poi, alla funzionalità successiva fino ad esaurire le funzionalità implementate. Quest'approccio mira principalmente a ridurre le dipendenze tra funzionalità differenti e a facilitare la ricerca di errori nelle interfacce di comunicazione tra sottosistemi.

6.3 Testing di Sistema

Lo scopo di questa fase di testing è quello di dimostrare che il sistema soddisfi effettivamente i requisiti richiesti e sia, quindi, pronto all'uso. Come per il testing di unità, si cercherà di testare le funzionalità più importanti per l'utente e quelle che hanno una maggiore probabilità di fallimento. Si noti che, come per il testing di unità, si procederà attraverso tecnica Black-Box.

7. Sospensione e ripresa

7.1 Criteri di sospensione

La fase di testing del sistema verrà sospesa quando si raggiungerà un compromesso tra qualità del prodotto e costi dell'attività di testing. Il testing verrà quindi portato avanti quanto più possibile nel tempo senza però rischiare di ritardare la consegna finale del progetto.

7.2 Criteri di ripresa

In seguito alle modifiche o correzioni delle componenti che introdurranno errori o fallimenti, i test case verranno sottoposti nuovamente al sistema assicurandosi così di aver risolto effettivamente il problema.

8. Materiale per il testing

L'hardware necessario per l'attività di test è un pc non necessariamente avente connessione ad internet, in quanto il sistema non è stato ancora rilasciato

9. Test Cases

1.0 LOGIN:

Specification: Login deve ricevere esattamente una username ed una password di tipo stringa

Scelte:

PARAMETRO: Username	
MATCH [MT]	1. Username non presente nel db [ERRORE] 2. Username presente nel db [OK]

PARAMETRO: Password	
MATCH [MT]	3. Se password combacia con l'username nel db se l'username è presente nel db [OK] 4. Se password combacia con l'username nel db se l'username è presente nel db [ERRORE]

TEST CASES:

CODICE	COMBINAZIONE	ESITO
TC 1.0_1	[MT2][MT4]	ERRATO
TC 1.0_2	[MT2][MT3]	CORRETTO

1.1 REGISTRAZIONE:

Specification: Registrazione deve ricevere esattamente un Nome, Cognome, username, password, conferma password, e-mail, numero di telefono, indirizzo, città di residenza, formati da stringhe tranne il numero di telefono che deve essere un intero

Categorie: Username, e-mail, numero di telefono non devono appartenere al database

Scelte:

PARAMETRO: Nome	
FORMATO: [A-Za-z]{2,}\$;	
LUNGHEZZA [LN]	1. < 3 or > 25 [ERRORE] 2. >= 3 && <= 25 [LN OK]
FORMATO [FM]	5. Se rispetta il formato [OK] 6. Se non rispetta il formato [ERRORE]

PARAMETRO: Cognome	
FORMATO: /^[a-zA-Z0-9]+ [a-zA-Z0-9]+\s{1}[a-zA-Z0-9]{1,} [a-zA-Z0-9]+\s{1}[a-zA-Z0-9]{3,}\s{1}[a-zA-Z0-9]{1,})\$/	
LUNGHEZZA [LN]	3. < 3 or > 25 [ERRORE] 4. >= 3 && <= 25 [LN OK]
FORMATO [FM]	7. Se rispetta il formato [OK] 8. Se non rispetta il formato [ERRORE]

PARAMETRO: Username	
FORMATO: /^[a-zA-Z0-9]+([a-zA-Z0-9](_ -)[a-zA-Z0-9])*[a-zA-Z0-9]+\$/	
LUNGHEZZA [LN]	5. < 3 or > 25 [ERRORE] 6. >= 3 && <= 25 [LN OK]
FORMATO [FM]	9. Se rispetta il formato [OK] 10. Se non rispetta il formato [ERRORE]
MATCH [MT]	11. Username già presente nel db [ERRORE] 12. Username non presente nel db [OK]

PARAMETRO: Password	
FORMATO: (?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,};	
LUNGHEZZA [LN]	7. < 3 or > 25 [ERRORE] 8. >= 3 && <= 25 [LN OK]
FORMATO [FM]	13. Se rispetta il formato [OK] 14. Se non rispetta il formato [ERRORE]

PARAMETRO: Conferma password	
FORMATO: (?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,};	
LUNGHEZZA [LN]	9. < 3 or > 25 [ERRORE] 10. >= 3 && <= 25 [LN OK]
FORMATO [FM]	15. Se rispetta il formato [OK] 16. Se non rispetta il formato [ERRORE]
MATCH[MT]	17. Se è uguale a Password [OK] 18. Se è diverso a Password [ERRORE]

PARAMETRO: E-mail	
FORMATO: [a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,3}\$;	
LUNGHEZZA [LN]	11. < 3 or > 25 [ERRORE] 12. >= 3 && <= 25 [LN OK]
FORMATO [FM]	19. Se rispetta il formato [FM OK] 20. Se non rispetta il formato [ERRORE]

PARAMETRO: Numero di telefono	
FORMATO: [0-9]{9,11}\$	
LUNGHEZZA [LN]	13. < 9 or > 10 [ERRORE] 14. >= 9 && <= 10 [LN OK]

FORMATO [FM]	21. Se rispetta il formato [FM OK] 22. Se non rispetta il formato [ERRORE]
---------------------	---

PARAMETRO: Indirizzo	
FORMATO: [A-Za-z]{2,}\$;	
LUNGHEZZA [LN]	15. < 3 or > 25 [ERRORE] 16. >= 3 && <= 25 [LN OK]
FORMATO [FM]	23. Se rispetta il formato [FM OK] 24. Se non rispetta il formato [ERRORE]

PARAMETRO: Città di residenza	
FORMATO: [A-Za-z]{2,}\$;	
LUNGHEZZA [LN]	17. < 3 or > 25 [ERRORE] 18. >= 3 && <= 25 [LN OK]
FORMATO [FM]	25. Se rispetta il formato [FM OK] 26. Se non rispetta il formato [ERRORE]

TEST CASES:

CODICE	COMBINAZIONE	ESITO
TC 1.1_1	[LN1][FM1][LN3][FM3][LN6][FM6][MT8][LN8][FM9] [LN10][FM11][FM13][MT13][LN12][FM15][LN14][FM17] [LN16][FM20][LN18][FM21]	ERRATO
TC 1.1_2	[LN2][FM1][LN4][FM3][LN6][FM5][MT8][LN8][FM9] [LN10][FM11][MT13][LN12][FM15][LN14][FM17] [LN16][FM19][LN18][FM21]	CORRETTO

1.2 MODIFICA PASSWORD:

Specification: Modifica password deve ricevere esattamente vecchia password, nuova password e conferma nuova password, tutti i parametri devono essere di tipo stringa

Categorie:

Scelte:

PARAMETRO: Nuova password	
FORMATO: (?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,};	
LUNGHEZZA [LN]	19. < 3 or > 25 [ERRORE] 20. >= 3 && <= 25 [LN OK]
FORMATO [FM]	27. Se rispetta il formato [OK] 28. Se non rispetta il formato [ERRORE]

PARAMETRO: Conferma nuova password	
FORMATO: (?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,};	
LUNGHEZZA [LN]	21. < 3 or > 25 [ERRORE] 22. >= 3 && <= 25 [LN OK]
FORMATO [FM]	29. Se rispetta il formato [OK] 30. Se non rispetta il formato [ERRORE]
MATCH[MT]	31. Se è uguale a Nuova password [OK] 32. Se è diversa a Nuova password [ERRORE]

TEST CASES:

CODICE	COMBINAZIONE	ESITO
TC 1.2_1	[LN20][FM27][LN22][FM32][FM29]	ERRATO
TC 1.2_2	[LN20][FM27][LN22][FM29][MT31]	CORRETTO

1.3 MODIFICA INDIRIZZO:

Specification: Modifica indirizzo deve ricevere esattamente un indirizzo di tipo stringa

Scelte:

PARAMETRO: Indirizzo	
FORMATO: [A-Za-z]{2,}\$;	
LUNGHEZZA [LN]	23. < 3 or > 25 [ERRORE] 24. >= 3 && <= 25 [LN OK]
FORMATO [FM]	33. Se rispetta il formato [OK] 34. Se non rispetta il formato [ERRORE]

TEST CASES:

CODICE	COMBINAZIONE	ESITO
TC 1.3_1	[LN23][FM33]	ERRATO
TC 1.3_2	[LN24][FM33]	CORRETTO

1.4 AGGIUNGI FERMATA:

Specification: Aggiungi fermata deve ricevere esattamente una fermata antecedente scelta dall'elenco e deve ricevere una fermata di tipo stringa

Scelte:

PARAMETRO: Nome fermata	
FORMATO: [A-Za-z]\$;	
LUNGHEZZA [LN]	25. < 3 or > 25 [ERRORE] 26. >= 3 && <= 25 [LN OK]
FORMATO [FM]	35. Se rispetta il formato [OK] 36. Se non rispetta il formato [ERRORE]

TEST CASES:

CODICE	COMBINAZIONE	ESITO
TC 1.4_1	[LN25][FM35]	ERRATO
TC 1.4_2	[LN26][FM35]	CORRETTO

1.5 MODIFICA E-MAIL:

Specification: Modifica e-mail deve ricevere esattamente una nuova e-mail e conferma nuova e-mail, entrambi devono essere di tipo stringa

Categorie: L'email non deve essere già presente nel db

Scelte:

PARAMETRO: Nuova e-mail	
FORMATO: [a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,3}\$;	
LUNGHEZZA [LN]	27. < 3 [ERRORE] 28. >= 3 && <= 25 [LN OK]
FORMATO [FM]	37. Se rispetta il formato [FM OK] 38. Se non rispetta il formato [ERRORE]
MATCH [MT]	39. Se non è presente nel db [MT OK] 40. Se è presente nel db [ERRORE]

TEST CASES:

CODICE	COMBINAZIONE	ESITO
TC 1.5_1	[LN27][FM38][MT39]	ERRATO
TC 1.5_2	[LN28][FM37][MT39]	CORRETTO

1.6 AGGIUNGI AUTOBUS:

Specification: Aggiungi autobus deve ricevere esattamente un modello di tipo stringa, anno di immatricolazione, chilometri e posti di tipo intero

Categorie: L'autobus non deve essere già memorizzato all'interno del database

Scelte:

PARAMETRO: Modello	
FORMATO: [A-Za-z0-9._-]{3,20}\$	
LUNGHEZZA [LN]	29. < 3 or > 20 [ERRORE] 30. >= 3 && <= 20 [LN OK]
FORMATO [FM]	41. Se rispetta il formato [FM OK] 42. Se non rispetta il formato [ERRORE]

PARAMETRO: Immatricolazione	
FORMATO: [0-9]	
LUNGHEZZA [LN]	31. != 4 [ERRORE] 32. == 4 [LN OK]
FORMATO [FM]	43. Se rispetta il formato [FM OK] 44. Se non rispetta il formato [ERRORE]

PARAMETRO: Chilometri	
FORMATO: [0-9]	
LUNGHEZZA [LN]	33. < 0 >9 [ERRORE] 34. >=1 && <=9 [LN OK]
FORMATO [FM]	45. Se rispetta il formato [FM OK] 46. Se non rispetta il formato [ERRORE]

PARAMETRO: Numero di posti	
FORMATO: [0-9]	
LUNGHEZZA [LN]	1. < 0 >3 [ERRORE] 2. >=1 && <=3 [LN OK]
FORMATO [FM]	47. Se rispetta il formato [FM OK] 48. Se non rispetta il formato [ERRORE]

TEST CASES:

CODICE	COMBINAZIONE	ESITO
TC 1.6_1	[LN29][FM41] [LN31][FM43] [LN34][FM45] [LN2][FM47]	ERRATO
TC 1.6_2	[LN30][FM41] [LN32][FM43] [LN34][FM45] [LN2][FM47]	CORRETTO

1.7 AGGIUNGI AUTISTA:

Specification: Aggiungi autista deve ricevere esattamente nome, cognome, città che sono di tipo stringa ed una data di tipo data

Categorie: L'autista non deve essere già memorizzato all'interno del database

Scelte:

PARAMETRO: Nome	
FORMATO: [A-Za-z]{2,}\$;	
LUNGHEZZA [LN]	35. < 3 or > 25 [ERRORE] 36. >= 3 && <= 25 [LN OK]
FORMATO [FM]	49. Se rispetta il formato [OK] 50. Se non rispetta il formato [ERRORE]

PARAMETRO: Cognome	
FORMATO: [A-Za-z]{2,}\$;	
LUNGHEZZA [LN]	37. < 3 or > 25 [ERRORE] 38. >= 3 && <= 25 [LN OK]
FORMATO [FM]	51. Se rispetta il formato [OK] 52. Se non rispetta il formato [ERRORE]

PARAMETRO: Username	
FORMATO: /^[a-zA-Z0-9]+([a-zA-Z0-9](_ -) [a-zA-Z0-9])*[a-zA-Z0-9]+\$ /	
LUNGHEZZA [LN]	39. < 3 or > 25 [ERRORE] 40. >= 3 && <= 25 [LN OK]
FORMATO [FM]	53. Se rispetta il formato [OK] 54. Se non rispetta il formato [ERRORE]
MATCH [MT]	55. Username già presente nel db [ERRORE] 56. Username non presente nel db [OK]

PARAMETRO: Password	
FORMATO: (?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,};	
LUNGHEZZA [LN]	41. < 3 or > 25 [ERRORE] 42. >= 3 && <= 25 [LN OK]
FORMATO [FM]	57. Se rispetta il formato [OK] 58. Se non rispetta il formato [ERRORE]

TEST CASES:

CODICE	COMBINAZIONE	ESITO
TC 1.7_1	[LN35][FM49] [LN37][FM51] [LN40][FM53][MT56] [LN42][FM57]	ERRATO
TC 1.7_2	[LN36][FM49] [LN38][FM51] [LN40][FM53][MT56] [LN42][FM57]	CORRETTO

1.8 AGGIUNGI TRATTA:

Specification: Aggiungi fermata deve ricevere esattamente una città di partenza ed arrivo, un bus e delle fermate che sono di tipo stringa

Categorie: La tratta non deve essere già presente all'interno del sistema

Scelte:

PARAMETRO: Città di partenza	
FORMATO: [a-zA-Z]{2,}\$	
LUNGHEZZA [LN]	43. < 3 or > 25 [ERRORE] 44. >= 3 && <= 25 [LN OK]
FORMATO [FM]	59. Se rispetta il formato [FM OK] 60. Se non rispetta il formato [ERRORE]
MATCH [MT]	61. Se non è presente nel db [MT OK] 62. Se è presente nel db [ERRORE]

PARAMETRO: Città di arrivo	
FORMATO: [a-z0-9._%+~]+@[a-z0-9.-]+\.[a-z]{2,3}\$;	
LUNGHEZZA [LN]	45. < 3 or > 25 [ERRORE] 46. >= 3 && <= 25 [LN OK]
FORMATO [FM]	63. Se rispetta il formato [FM OK] 64. Se non rispetta il formato [ERRORE]
MATCH [MT]	65. Se è diversa a città di partenza [MT OK] 66. Se è uguale a città di partenza [ERRORE]

TEST CASES:

CODICE	COMBINAZIONE	ESITO
TC 1.8_1	[LN43][FM60] [LN61][FM86][MT87] [LN45][FM63] [MT65]	ERRATO
TC 1.8_2	[LN44][FM60] [LN61][FM86][MT87] [LN46][FM63] [MT65]	CORRETTO

1.9 RICHIEDI CORSA:

Specification: Aggiungi corsa deve ricevere esattamente una tratta, una o più fermate, orario di tipo orario

Scelte:

PARAMETRO: Data	
FORMATO: <code>/^([0-2][0-9] (3)[0-1])(\\)((([0-9]) ([1][0-2]))(\\)\\d{4})\$/</code>	
LUNGHEZZA [LN]	47. != 10 [ERRORE] 48. == 10 [LN OK]
FORMATO [FM]	67. Se rispetta il formato [OK] 68. Se non rispetta il formato [ERRORE]

PARAMETRO: Ora partenza	
FORMATO: <code>^\\d{2}\\:\\d{2}\\-\\d{2}\\:\\d{2}\$</code>	
LUNGHEZZA [LN]	49. !=5 [ERRORE] 50. ==5 [LN OK]
FORMATO [FM]	69. Se rispetta il formato [FM OK] 70. Se non rispetta il formato [ERRORE]

TEST CASES:

CODICE	COMBINAZIONE	ESITO
TC 1.9_1	[LN47][FM67] [LN50][FM69]	ERRATO
TC 1.9_2	[LN48][FM67] [LN50][FM69]	CORRETTO