



Laurea Triennale in informatica-Università di Salerno
Corso di *Ingegneria del Software*- Prof. C.Gravino



TP Test Plan

UniPass

Versione	1.0
Data	16/12/2022
Destinatario	Prof. C. Gravino
Presentato da	Barretta Alessandro Di Lauro Antonio Malafronte Sabato Zampaglione Gerardo
Approvato da	



Revision History

Data	Versione	Cambiamenti	Autori
10/12/2022	0.1	Prima stesura	[Gruppo]
10/12/2022	0.2	Aggiunta vincoli dei TC	Antonio Di Lauro
10/12/2022	0.3	Aggiunta dei Test Frames	Sabato Malafronte
11/12/2022	0.4	Stesura capitoli 1, 8, 9	Gerardo Zampaglione
12/12/2022	0.5	Stesura capitoli 2, 3, 5	Antonio Di Lauro
13/12/2022	0.6	Stesura capitoli 4, 7	Sabato Malafronte
13/12/2022	0.7	Stesura capitoli 6, 10	Alessandro Barretta
16/12/2022	1.0	Revisione pre-consegna intermedia	[Gruppo]



Sommario

I.	Revision History	2
II.	1. Introduzione	4
III.	2. Documenti correlati	4
	Relazione con il Requirement Analysis Document.....	4
	Relazione con il System Design Document.....	4
IV.	3. Panoramica del sistema	5
V.	4. Funzionalità da testare	5
VI.	5. Criteri Pass/Failed	5
VII.	6. Approccio	6
	Testing di sistema	6
	Testing di unità	6
VIII.	7. Sospensione e ripresa	6
	Criteri di sospensione	6
	Criteri di ripresa	6
IX.	8. Materiale per il testing	7
X.	9. Test cases	7
	9.1 Registrazione.....	7
	9.2 Autenticazione.....	8
	9.3 Viaggio	10
XI.	10. Testing Schedule	12



1. Introduzione

UniPass è un sistema che si affianca agli studenti che solitamente rientrano a casa con i servizi di trasporto pubblico locale, per aiutarli nel viaggio dal Campus di Fisciano fino al loro domicilio. Gli studenti automuniti possono iscriversi all'applicativo con la propria auto e creare viaggi. I viaggi saranno reperibili dagli utenti attraverso la ricerca, dopo la quale potranno scegliere quello a cui prenotarsi.

L'obiettivo è quello di evitare i disagi, attualmente presenti con gli enti di trasporto locale, per tornare a casa dal Campus e di alleggerire le spese per chi solitamente si reca all'Università con la propria auto.

Il documento di Test Plan ha l'obiettivo di descrivere ed analizzare le attività di Testing per la piattaforma proposta con l'obiettivo di garantire la correttezza di ogni funzionalità.

All'interno del documento sono presenti:

- Le strategie di testing adottate
- La dichiarazione delle funzionalità che saranno testate
- Gli strumenti scelti per effettuare la rilevazione degli errori

I risultati dei test che verranno eseguiti saranno presi in considerazione per la rilevazione degli eventuali errori e le loro cause (fault). Successivamente questi verranno corretti, in modo da eliminare i malfunzionamenti precedentemente individuati.

2. Documenti correlati

L'individuazione dei Test Case da eseguire è stata fatta sulla base dei documenti già realizzati.

Relazione con il Requirement Analysis Document

Il documento principale a cui si fa riferimento è il Requirement Analysis Document, che contiene una descrizione dettagliata di requisiti funzionali, scenari e use case, i quali costituiscono la fonte principale della pianificazione dei Test Case.

Relazione con il System Design Document

Il planning dei Test Case avverrà in accordo con la decomposizione in sottosistemi del System Design Document. In particolare, l'attenzione verrà posta sul livello logico di business, di cui si andranno a testare le funzionalità dei relativi sottosistemi.

3. Panoramica del sistema

Il sistema proposto si basa sull'architettura Three-Tier, un modello che ha come caratteristica la presenza di tre layer: presentazione, che riguarda l'interfaccia utente; business, dove le informazioni ottenute dal livello di presentazione vengono processate; persistenza, dove le informazioni vengono conservate. Lo schema di riferimento è riportato nel System Design Document nella sezione "Diagramma architetturale". Per ogni layer verranno usati diversi strumenti:

- Per la parte di front-end si utilizzeranno HTML, CSS e JavaScript;
- Per la parte di back-end si utilizzerà Java SPRING;
- Per la gestione della persistenza verrà usato MySQL.

4. Funzionalità da testare

Successivamente viene presentata la lista delle features di cui si effettuerà il testing per le varie gestioni:

- Sottosistema Registrazione
 - Registrazione
- Sottosistema Autenticazione
 - Modifica Profilo
- Sottosistema Viaggio
 - Crea viaggio
 - Cerca viaggi

La scelta di queste funzionalità per il testing di sistema è dettata dall'intenzione di mantenere la tracciabilità tra i vari documenti progettuali.

Le altre funzionalità non saranno testate a causa dei vincoli tecnici del progetto.

5. Criteri Pass/Failed

La fase di testing ha come obiettivo quello di individuare faults all'interno del Sistema. Dunque, il test ha successo (Pass) nel momento in cui l'output ottenuto è diverso dal risultato che si voleva ottenere. Il test, invece, fallisce (Fail) quando l'output ottenuto concorda con il risultato atteso.

6. Approccio

Il Testing dell'intero Sistema è stato pianificato seguendo i vincoli progettuali, e sarà quindi composto da due fasi: Testing di sistema e Testing di unità. Di seguito si descrivono le strategie adottate per la loro progettazione ed esecuzione.

Testing di sistema

In questa fase il focus principale è quello di testare le funzionalità del Sistema rispetto ai possibili input degli utenti. Viene utilizzata la tecnica del Category Partition per ridurre all'essenziale i casi di test.

Siccome il Sistema sarà interamente accessibile sulla Rete, verrà utilizzato il tool Selenium, adatto a registrare le interazioni dell'utente con i siti Web.

Testing di unità

In questa fase si provvederà ad effettuare il testing di alcuni metodi delle classi sviluppate, non includendo quelle esclusivamente caratterizzate da metodi getters e setters.

Per l'esecuzione dei test verrà utilizzata la strategia di black-box testing e in particolare, come fatto per il Testing di sistema, verrà seguita la tecnica del Category Partition. Dato che il progetto sarà sviluppato in linguaggio Java, si è scelto di utilizzare il framework JUnit, dedicato specificatamente al Testing di unità.

7. Sospensione e ripresa

Nella successiva sezione saranno definiti i criteri di sospensione del test e le operazioni che dovranno essere ripetute quando il testing riprende.

Criteri di sospensione

Non è prevista la sospensione del testing in caso di rilevamento di una failure. L'eventuale sospensione momentanea è prevista nel caso in cui venga restituito un errore nella definizione di uno dei test.

Criteri di ripresa

Il testing riprenderà una volta risolti i fault emersi.

8. Materiale per il testing

Siccome l'applicativo non è ancora stato rilasciato, per eseguire il testing necessario alla verifica degli eventuali problemi servirà un qualunque computer in grado di eseguire localmente il Sistema.

9. Test cases

Per ogni sottosistema mostriamo le funzionalità che andremo a testare e ad ognuna sarà associata una tabella dove definiamo:

- eventuale formato da seguire
- il nome del parametro da testare
- il nome della categoria del testing da effettuare (ad esempio lunghezza, formato ecc.)
- le possibili scelte che l'utente può intraprendere

9.1 Registrazione

Per il sottosistema Registrazione si andrà a testare la funzionalità di registrazione di un nuovo utente

Tale funzionalità prevede la compilazione di un Form da parte dell'utente in cui sono richiesti i seguenti dati:

- Nome: stringa con un numero di caratteri compreso tra 1 e 64
- Cognome: stringa con un numero di caratteri compreso tra 1 e 64
- E-mail: stringa che deve rispettare un certo formato
- Password: stringa con un numero di caratteri compreso tra 8 e 32
- Telefono: stringa contenente 10 numeri.

Parametro: E-mail	
FORMATO	
^[A-z0-9._%+-]+@[A-z0-9.-]+\.[A-z]{2,10}\$	
Nome Categoria	Scelta per la categoria
Correttezza [CE]	1. Rispetta il formato = false [errore] 2. Rispetta il formato = true [PROPERTY_CE_OK]
Parametro: Password	
Nome Categoria	Scelta per la categoria
Lunghezza [LP]	1. Lunghezza < 8 OR Lunghezza > 32 [errore] 2. Lunghezza >= 8 AND Lunghezza <= 32 [PROPERTY_LP_OK]
Parametro: Nome	
Nome Categoria	Scelta per categoria
Lunghezza [LN]	1. Lunghezza < 1 OR Lunghezza > 64 [errore] 2. Lunghezza >= 1 AND Lunghezza <= 64 [PROPERTY_LN_OK]



Parametro: Cognome	
Nome Categoria	Scelta per categoria
Lunghezza [LC]	1. Lunghezza < 1 OR Lunghezza > 64 [errore] 2. Lunghezza >= 1 AND Lunghezza <= 64 [PROPERTY_LC_OK]
Parametro: Telefono	
FORMATO [0-9]{10}	
Nome Categoria	Scelta per categoria
Correttezza [CT]	1. Rispetta il formate = false [errore] 2. Rispetta il formato = true [PROPERTY_CT_OK]

Test Case ID	Test frame	Esito
TC_1.1_1	CE1	Errato: L'email non rispetta il formato
TC_1.1_2	CE2, LP1	Errato: La password ha una lunghezza non valida
TC_1.1_3	CE2, LP2, LN1	Errato: Il nome ha una lunghezza non valida
TC_1.1_4	CE2, LP2, LN2, LC1	Errato: Il cognome ha una lunghezza non valida
TC_1.1_5	CE2, LP2, LN2, LC2, CT1	Errato: il numero di telefono non rispetta il formato
TC_1.1_6	CE2, LP2, LN2, LC2, CT2	Corretto

9.2 Autenticazione

Per il sottosistema Autenticazione si andrà a testare la funzionalità di modifica del profilo da parte di un utente

Tale funzionalità prevede la modifica delle informazioni associate al profilo dell'utente, modificabili compilando un Form contenente i seguenti dati:

- Nome: stringa con un numero di caratteri compreso tra 1 e 64
- Cognome: stringa con un numero di caratteri compreso tra 1 e 64
- E-mail: stringa che deve rispettare un certo formato
- Password: stringa con un numero di caratteri compreso tra 8 e 32
- Telefono: stringa contenente 10 numeri.



Parametro: E-mail	
FORMATO	
^[A-z0-9._%+-]+@[A-z0-9.-]+\.[A-z]{2,10}\$	
Nome Categoria	Scelta per la categoria
Correttezza [CE]	3. Rispetta il formato = false [errore] 4. Rispetta il formato = true [PROPERTY_CE_OK]
Parametro: Password	
Nome Categoria	Scelta per la categoria
Lunghezza [LP]	3. Lunghezza < 8 OR Lunghezza > 32 [errore] 4. Lunghezza >= 8 AND Lunghezza <= 32 [PROPERTY_LP_OK]
Parametro: Nome	
Nome Categoria	Scelta per categoria
Lunghezza [LN]	3. Lunghezza < 1 OR Lunghezza > 64 [errore] 4. Lunghezza >= 1 AND Lunghezza <= 64 [PROPERTY_LN_OK]
Parametro: Cognome	
Nome Categoria	Scelta per categoria
Lunghezza [LC]	3. Lunghezza < 1 OR Lunghezza > 64 [errore] 4. Lunghezza >= 1 AND Lunghezza <= 64 [PROPERTY_LC_OK]
Parametro: Telefono	
FORMATO	
[0-9]{10}	
Nome Categoria	Scelta per categoria
Correttezza [CT]	3. Rispetta il formate = false [errore] 4. Rispetta il formato = true [PROPERTY_CT_OK]

Test Case ID	Test frame	Esito
TC_2.1_1	CE1	Errato: L'email non rispetta il formato
TC_2.1_2	CE2, LP1	Errato: La password ha una lunghezza non valida
TC_2.1_3	CE2, LP2, LN1	Errato: Il nome ha una lunghezza non valida
TC_2.1_4	CE2, LP2, LN2, LC1	Errato: Il cognome ha una lunghezza non valida
TC_2.1_5	CE2, LP2, LN2, LC2, CT1	Errato: il numero di telefono non rispetta il formato
TC_2.1_6	CE2, LP2, LN2, LC2, CT2	Corretto



9.3 Viaggio

Per il sottosistema Viaggio si andranno a testare due funzionalità: Creazione viaggio e Cerca viaggi

La prima funzionalità mette a disposizione dei guidatori la possibilità di creare dei viaggi inserendo all'interno di un Form le seguenti informazioni:

- Comune di destinazione: stringa selezionata da un insieme di stringhe predefinite
- Data di partenza: insieme all'ora di partenza costituisce la data e l'ora di partenza del viaggio.
- Ora di partenza: insieme alla data di partenza costituisce la data e l'ora di partenza del viaggio.
- Numero posti: numero intero che indica il numero di posti a disposizione dei passeggeri. Deve essere compreso tra 1 e il numero di posti a sedere del veicolo – 1.
- Prezzo: numero che indica il costo (in euro) per prenotarsi al viaggio. Deve essere compreso tra 0 e 50.

Parametro: Data e ora	
Nome Categoria	Scelta per la categoria
Correttezza [CD]	1. $DataOra < (DataOraCorrente + 1\ ora)$ [errore] 2. $DataOra \geq (DataOraCorrente + 1\ ora)$ [PROPERTY_CD_OK]
Parametro: Numero Posti	
Nome Categoria	Scelta per la categoria
Valore [VNP]	1. $NumeroPosti < 1$ OR $NumeroPosti > (PostiDisponibili - 1)$ [errore] 2. $NumeroPosti \geq 1$ AND $NumeroPosti \leq (PostiDisponibili - 1)$ [PROPERTY_VNP_OK]
Parametro: Prezzo	
Nome Categoria	Scelta per la categoria
Valore [VP]	1. $Prezzo < 0$ OR $Prezzo > 50$ [errore] 2. $Prezzo \geq 0$ AND $Prezzo \leq 50$ [PROPERTY_VP_OK]



Test Case ID	Test frame	Esito
TC_3.1_1	CD1	Errato: Data nel passato
TC_3.1_2	CD2, VNP1	Errato: Valore non valido per numero posti
TC_3.1_3	CD2, VNP2, VP1	Errato: Valore per il prezzo invalido
TC_3.1_4	CD2, VNP2, VP2	Corretto

La seconda funzionalità mette a disposizione dei passeggeri la possibilità di cercare dei viaggi inserendo all'interno di un Form le seguenti informazioni:

- Comune di destinazione: stringa selezionata da un insieme di stringhe predefinite
- Data di partenza: giorno, mese e anno a cui la data di partenza dei viaggi risultati deve corrispondere.
- Ora di partenza: L'orario di partenza minimo, costituito da ora e minuti, che i viaggi risultati devono avere.
- Prezzo: numero che indica il costo massimo che si è disposti a pagare per la prenotazione di un viaggio. Dev'essere compreso tra 0 e 50.

Parametro: Data e ora	
Nome Categoria	Scelta per la categoria
Correttezza [CD]	1.DataOra < (DataOraCorrente + 40 minuti) [errore] 2.DataOra >= (DataOraCorrente + 40 minuti) [PROPERTY_CD_OK]
Parametro: Prezzo	
Nome Categoria	Scelta per la categoria
Valore [VP]	1.Prezzo < 0 OR Prezzo > 50 [errore] 2.Prezzo <= 50 AND Prezzo >= 0 [PROPERTY_VP_OK]

Test Case ID	Test frame	Esito
TC_3.2_1	CD1	Errato: Data e ora non ammissibili
TC_3.2_2	CD2, VP1	Errato: Prezzo incorretto
TC_3.2_3	CD2, VP2	Corretto



10. Testing Schedule

I test verranno eseguiti sia durante che dopo l'implementazione del Sistema. Inoltre, per garantire il suo corretto funzionamento, tutti i test saranno rieseguiti al termine della fase di sviluppo.