

Università degli Studi di Salerno
Corso di Ingegneria del Software

MalTour
Object Design Document
Versione 2.0

MalTour 

Data: 25/12/2021

Progetto: MalTour	Versione: 2.0
Documento: SSD	Data: 25/11/2021

Coordinatore del progetto:

Nome	Matricola
Teresa Vitagliano	052105622

Partecipanti:

Nome	Matricola
Teresa Vitagliano	0512105622

Scritto da:	Teresa Vitagliano
--------------------	-------------------

Revision History

Data	Versione	Descrizione	Autore
24/11/2021	1.0	Prima stesura del documento	Teresa Vitagliano
01/12/2021	1.1	Relazioni con i documenti, criteri	Teresa Vitagliano
02/12/2021	1.2	Test case	Teresa Vitagliano

1.	Introduzione.....	4
2.	Relazioni con altri documenti.....	4
2.1.	Relazione con il RAD	4
2.2.	Relazioni con SDD	4
2.3.	Relazioni con ODD.....	4
3.	Features to be tested/not to be tested	4
3.1	Features da testare.....	4
3.2	Features da non testare	5
4.	Pass/Fail criteria	5
5.	Approccio.....	6
5.1	Testing di Unità.....	6
5.2	Testing di Integrazione.....	6
5.3	Testing di Sistema	6
6.	Regole di sospensione e ripresa del testing	7
6.1	Regole di Sospensione.....	7
6.2	Regole di Ripresa	7
7.	Materiale di testing	7
8.	Test Case.....	7
8.1	TC1_Registrazione.....	7
8.2	TC2_Login.....	9
8.3	TC3_ModificaServizio.....	10
8.4	TC4_AggiungiServizio	11
8.5	TC5_AggiungiOfferta	12
8.6	TC6_ModificaOfferta	16
8.7	TC7_AggiungiComponenti.....	19

1. Introduzione

Lo scopo di questo documento è quello di gestire lo sviluppo e le attività di test riguardanti il sito web “Maltour”. Saranno identificati: gli elementi e le funzionalità da testare, le strategie di testing e gli strumenti utilizzati, il personale responsabile dei test, le risorse e le attività richieste per completare i test e i rischi associati al piano. Lo scopo del testing è quello di rilevare errori in maniera pianificata all'interno del codice realizzato. L'obiettivo del testing consiste nell'evitare che gli errori si presentino durante l'utilizzo del sistema dell'utente finale. I risultati prodotti dai test saranno utilizzati per comprendere dove intervenire per correggere gli errori o apportare modifiche per il migliorare il sistema. In questo documento verranno analizzate, in particolar modo, le seguenti attività:

- Gestione Utente
- Gestione Offerte
- Gestione Servizi
- Gestione Ordini

2. Relazioni con altri documenti

Il Test Plan ha una stretta relazione con i documenti prodotti finora, dato che il sistema è stato pianificato nelle precedenti documentazioni. Per verificare il corretto funzionamento del sito web “Maltour” saranno usati i test cases individuati e documentati precedentemente nel processo di analisi del sistema. I test cases sono basati sulle funzionalità individuate nel documento di raccolta ed analisi dei requisiti.

2.1. Relazione con il RAD

La relazione tra test plan e RAD (Requirement Analysis Document) riguarda in particolare i requisiti funzionali e non funzionali del sistema visto che i test saranno eseguiti su quelle funzionalità tenendo conto delle specifiche espresse nel documento precedente. In particolare, il RAD contiene lo scopo del sistema, l'ambito del sistema e gli obiettivi, evidenziando una panoramica di requisiti funzionali, requisiti non funzionali, scenari, casi d'uso, diagrammi e mock-up del sistema.

2.2. Relazioni con SDD

Tramite il System Design Document sarà possibile definire i sottosistemi e i servizi da porre a Testing.

2.3. Relazioni con ODD

Tramite l'Object Design Document sarà possibile definire le classi e le componenti del sistema da porre a Testing.

3. Features to be tested/not to be tested

Di seguito saranno elencate le funzionalità introdotte nel sistema che saranno sottoposte a test, suddivise per ogni gestione del sistema.

3.1 Features da testare

Gestione Utente

- Registrazione
- Login

Gestione Servizi

- Modifica Servizio
- Aggiungi Servizio
- Rimuovi Servizio

Gestione Offerta

- Modifica Offerta
- Aggiungi Offerta
- Rimuovi Offerta

Gestione Ordini

- AggiungiComponenti
- Pagamento

3.2 Features da non testare

Gestione Utente

- Visualizza Profilo
- Rimuovi Profilo

Gestione Servizi

- Visualizza Servizi

Gestione Offerta

- Visualizza Offerta
- Ricerca Offerta

Gestione Ordini

- AggiungiAlCarrello
- RimuoviDalCarrello
- EffettuaOrdine
- Visualizza Carrello
- Visualizza Ordini
- Convalida Ordine

4. Pass/Fail criteria

Lo scopo del testing è quello di trovare delle failure durante l'esecuzione del sistema. Il testing ha successo se l'output osservato (finale) è diverso dall'output atteso: ciò significa che la fase di testing avrà successo se individua una failure. Nel caso verrà riscontrata una

failure, bisognerà verificare a che tipo di fault è legata, se di tipo meccanico o algoritmico. Al termine dell'individuazione del fault si procederà alla sua correzione. Sarà infine iterata la fase di testing per verificare che la modifica non abbia impattato su altri componenti del sistema. La failure quindi è uno stato di condizione nel quale non si trova l'output desiderato.

5. Approccio

L'approccio alla fase di testing si compone di tre fasi, eseguiti nell'ordine inverso e progettati secondo il seguente ordine:

- Testing di Sistema
- Testing di Integrazione
- Testing di Unità

La progettazione dei casi di test di sistema e di integrazione avverrà prima della fase di implementazione del sistema e, inoltre, verranno raffinati durante la loro esecuzione. La progettazione dei casi di test di unità avverrà durante la fase di implementazione. La loro esecuzione avverrà durante la stesura del codice, in modo parallelo, così da avere un riscontro immediato degli errori di implementazione. Ogni volta che verrà introdotta una nuova funzionalità verranno eseguiti i test ad essa legata e, inoltre, i test di integrazione che la coinvolgono. Ad ogni modifica delle componenti, verranno eseguiti i test di unità e di integrazione delle componenti direttamente dipendenti da quelle modificate.

5.1 Testing di Unità

Per testing di unità si intende il testing delle singole unità software del sistema ovvero si andranno a testare ogni metodo di ciascuna classe del sistema. Il test verrà effettuato con il framework JUnit. Per individuare i test da effettuare si utilizzerà la tecnica Black-Box. Gli unit testing possono essere divisi principalmente in due tipologie: whitebox testing e blackbox testing. Il blackbox testing guarda solo l'input e l'output senza curare il codice, il whitebox testing invece cura solo il codice e la struttura senza guardare l'input e output.

5.2 Testing di Integrazione

L'approccio che verrà utilizzato è il bottom-up, ritenuto il più adatto per i software basati sul paradigma Object Oriented. Si farà utilizzo del framework JUnit per la definizione dei test case in Java.

5.3 Testing di Sistema

Il testing di sistema è stato realizzato tramite il tool Selenium. Questo permette di registrare le azioni che un utente svolge sul browser, in modo tale da implementare ed eseguire i test case di sistema. Per il testing, il server sarà deployato in localhost.

6. Regole di sospensione e ripresa del testing

6.1 Regole di Sospensione

La fase di testing sarà sospesa nel caso verrà rilevato un difetto che può limitare il processo di test. La sospensione del processo dovrà incidere il meno possibile sulle risorse disponibili. La fase di testing può essere sospesa qualora si raggiungeranno gli obiettivi dichiarati, rispettando i tempi fissati.

6.2 Regole di Ripresa

La fase di testing riprenderà quando il difetto verrà risolto con successo. I test verranno ripetuti per controllare se le modifiche non hanno generato nuovi errori.

7. Materiale di testing

Gli strumenti necessari per svolgere le attività di testing sono:

- WebServer Apache Tomcat 9 in locale che gira sul sistema.
- Un DBMS mySQL che gestisce l'utilizzo del database.
- Selenium IDE per il test di integrazione.
- JUnit per il test di unità.

8. Test Case

Per il test cases si utilizzerà la tecnica BLACK-BOX. Con il BLACK-BOX testing ci focalizzeremo sul comportamento dell'input/output delle singole componenti senza tener conto della loro struttura interna. A causa della mancanza di fattibilità di effettuare un test esaustivo per l'ingente quantitativo di dati di input, verrà utilizzata la strategia del Category Partition, che consente di decomporre lo spazio di input in categorie per poi partizionare le "categorie" in classi di equivalenza chiamate "scelte". Al termine saranno specificate le "combinazioni" delle scelte da testare creando delle istanze di casi di test specificando i valori dei dati effettivi per ciascuna scelta e determinare i risultati corrispondenti. Mediante il Category Partition otterremo, quindi, un test efficiente e privo di ridondanze.

8.1 TC1_Registrazione

Parametro: e-mail Formato: <code>^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w+)+\$</code>	
CATEGORIE	SCELTE
Lunghezza e-mail - LE	<ol style="list-style-type: none">1. Lunghezza = 0 - campo vuoto [error]2. Lunghezza > 1 - [property LE_OK]
Formato e-mail - FE	<ol style="list-style-type: none">1. Non rispetta il formato [if LE_OK] [error]2. Rispetta il formato [if LE_OK] [property FE_OK]
Esiste email - EE	<ol style="list-style-type: none">1. Esiste nel DB [if LE_OK AND FE_OK] [error]

	2. Non esiste nel DB [if LE_OK AND FE_OK] [property EE_OK]
--	---

Parametro: nome Formato: ^[a-zA-Z\u00C0-\u00ff]+\$	
CATEGORIE	SCELTE
Lunghezza nome - LN	1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza > 1 - [property LN_OK]
Formato nome - FN	1. Non rispetta il formato [if LN_OK AND FN_OK] [error] 2. Rispetta il formato [if LN_OK AND FN_OK] [Property FN_OK]

Parametro: username Formato: ^[0-9a-zA-Z]+\$	
CATEGORIE	SCELTE
Lunghezza username - LU	1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza > 0 and < 6 - [error] 3. Lunghezza >= 6 - [property LU_OK]
Formato username - FU	1. Non rispetta il formato [if LU_OK] [error] 2. Rispetta il formato [if LU_OK] [property FU_OK]
Esiste username - EU	1. Esiste nel DB [if LU_OK AND FU_OK] [error] 2. Non esiste nel DB [if LU_OK AND FU_OK] [property EU_OK]

Parametro: password Formato: *[0-9].*	
CATEGORIE	SCELTE
Lunghezza password - LP	1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza > 0 and < 7 - [error] 3. Lunghezza > 7 [property LP_OK]
Formato password - FP	1. Non rispetta il formato [if LP_OK] [error] 2. Rispetta il formato [if LP_OK] [property LP_OK]
Corrispondenza password - CP	1. Lunghezza ConfermaPassword = 0 - [error] 2. ConfermaPassword non corrisponde a password [if LP_OK AND FP_OK] [error] 3. ConfermaPassword corrisponde a password [if

	LP_OK AND FP_OK] [property CP_OK]
--	-----------------------------------

CODICE	COMBINAZIONE	ESITO
TC_Registrazione_1	LN1	Errore: campo email vuoto
TC_Registrazione_2	LN2, FN1	Errore: formato non valido
TC_Registrazione_3	LN2, FN2, EE1	Errore: email esistente
TC_Registrazione_4	LN2, FN2, EE2, LN1	Errore: campo nome vuoto
TC_Registrazione_5	LN2, FN2, EE2, LN2, FN1	Errore: formato non valido
TC_Registrazione_6	LN2, FN2, EE2, LN2, FN2, LU1,	Errore: campo username vuoto
TC_Registrazione_7	LN2, FN2, EE2, LN2, FN2, LU2,	Errore: lunghezza username non valida
TC_Registrazione_8	LN2, FN2, EE2, LN2, FN2, LU3, FU1	Errore: formato non valido
TC_Registrazione_9	LN2, FN2, EE2, LN2, FN2, LU3, FU2, EU1	Errore: username esistente
TC_Registrazione_10	LN2, FN2, EE2, LN2, FN2, LU3, FU2, EU2, LP1	Errore: campo password vuoto
TC_Registrazione_11	LN2, FN2, EE2, LN2, FN2, LU3, FU2, EU2, LP2	Errore: lunghezza password non valida
TC_Registrazione_12	LN2, FN2, EE2, LN2, FN2, LU3, FU2, EU2, LP3, FP1	Errore: formato non valido
TC_Registrazione_13	LN2, FN2, EE2, LN2, FN2, LU3, FU2, EU2, LP3, FP2, CP1	Errore: corrispondenza password vuoto
TC_Registrazione_14	LN2, FN2, EE2, LN2, FN2, LU3, FU2, EU2, LP3, FP2, CP2	Errore: corrispondenza password lunghezza non valida
TC_Registrazione_15	LN2, FN2, EE2, LN2, FN2, LU3, FU2, EU2, LP3, FP2, CP3	Corretto, registrazione avvenuta

8.2 TC2_Login

Parametro: username Formato: ^[0-9a-zA-Z]+\$	
CATEGORIE	SCELTE

Lunghezza username - LU	1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >=6 - [property LU_OK]
Esiste username - EU	1. Esiste nel DB [if LU_OK AND FU_OK] [error] 2. Non esiste nel DB [if LU_OK AND FU_OK] [property EU_OK]

Parametro: password Formato: .*[0-9].*	
CATEGORIE	SCELTE
Lunghezza password - LP	1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >= 8 - [LP_OK]
Corrispondenza password e-mail - CPE	1. Non c'è corrispondenza tra password ed e-mail nel database [if LP_OK] [error] 2. C'è corrispondenza tra password ed e-mail nel database [if LP_OK] [property CPE_OK]

CODICE	COMBINAZIONE	ESITO
TC_Login_1	LE1	Errore: campo e-mail vuoto
TC_Login_2	LE2, EE1	Errore: e-mail esiste nel database
TC_Login_3	LE2, EE2, LP1	Errore: campo password vuoto
TC_Login_4	LE2, EE2, LP2, CPE1	Errore: corrispondenza non valida
TC_Login_5	LE, EE2, LP2, CPE2	Corretto: login avvenuto

8.3 TC3_ModificaServizio

Parametro: nome	
CATEGORIE	SCELTE
Lunghezza nome - LN	1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >=1 - [property LN_OK]

Parametro: nome e descrizione	
CATEGORIE	SCELTE
Esiste servizio - EC	1. Esiste nel DB [error] 2. Non esiste nel DB [property EC_OK]

--	--

Parametro: descrizione	
CATEGORIE	SCELTE
Lunghezza descrizione - LD	1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >=1 - [property LD_OK]

CODICE	COMBINAZIONE	ESITO
TC_ModificaServizio_1	LN1	Errore: campo nome vuoto
TC_ModificaServizio _2	LN2, EC1	Errore: servizio esiste nel database
TC_ModificaServizio _3	LN2, EC2, LD1	Errore: campo descrizione vuoto
TC_ModificaServizio _4	LN2, EC2, LD2	Corretto: modifica cservizio avvenuta con successo

8.4 TC4_AggiungiServizio

Parametro: nome	
CATEGORIE	SCELTE
Lunghezza nome - LN	3. Lunghezza = 0 - campo vuoto [error] 4. Lunghezza >=1 - [property LN_OK]
Esiste nome - EN	1. Esiste nel DB [if LN_OK] [error] 2. Non esiste nel DB [if LN_OK] [property EN_OK]

Parametro: descrizione	
CATEGORIE	SCELTE
Lunghezza descrizione - LD	1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >=1 - [property LD_OK]

CODICE	COMBINAZIONE	ESITO
TC_AggiungiServizio_1	LN1	Errore: campo nome vuoto
TC_AggiungiServizio _2	LN2, EN1	Errore: nome esiste nel database

TC_AggiungiServizio _3	LN2, EN2, LD1	Errore: campo descrizione vuoto
TC_AggiungiServizio _4	LN2, EN2, LD2	Corretto: aggiunto servizio avvenuto con successo

8.5 TC5_AggiungiOfferta

Parametro: id	
CATEGORIE	SCELTE
Lunghezza id - LI	1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >=1 - [property LD_OK]
Esiste id - EI	1. Esiste nel DB [error] 2. Non esiste nel DB [property EI_OK]

Parametro: destinazione	
CATEGORIE	SCELTE
Lunghezza destinazione- LDE	1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >=1 - [property LDE_OK]

Parametro: descrizione	
CATEGORIE	SCELTE
Lunghezza descrizione - LD	1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >=1 - [property LD_OK]

Parametro: data_partenza	
CATEGORIE	SCELTE
Lunghezza data_partenza - LDP	1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >=1

	- [property LDP_OK]
--	---------------------

Parametro: ora_partenza	
CATEGORIE	SCELTE
Lunghezza ora_partenza - LOP	1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >=1 - [property LOP_OK]

Parametro: data_ritorno	
CATEGORIE	SCELTE
Lunghezza data_ritorno - LDR	1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >=1 - [property LD_OK]

Parametro: ora_ritorno	
CATEGORIE	SCELTE
Lunghezza ora_ritorno - LDR	1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >=1 - [property LOR_OK]

Parametro: partenza_da	
CATEGORIE	SCELTE
Lunghezza partenza_da- LPDA	1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >=1 - [property LPD_OK]

Parametro: arrivo_a	
CATEGORIE	SCELTE
Lunghezza arrivo_a - LAA	1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >=1 - [property LAA_OK]

Parametro: pernottamento	
CATEGORIE	SCELTE
Lunghezza pernottamento - LP	1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >=1

	- [property LP_OK]
--	--------------------

Parametro: posti_disponibili	
CATEGORIE	SCELTE
Quantità Posti_disponibili - LPDI	1. quantità = 0 - campo vuoto [error] 2. quantità >=1 - [property LPDI_OK]

Parametro: prezzo	
CATEGORIE	SCELTE
Inserimento prezzo - IP	1. Inserimento = false [error] 2. Inserimento = true - [property IP_OK]

CODICE	COMBINAZIONE	ESITO
TC_AggiungiOfferta_1	LI1	Errore: campo id vuoto
TC_AggiungiOfferta_2	LI2, EI1	Errore: l'id esiste nel db
TC_AggiungiOfferta_3	LI2, EI2, LDE1	Errore: destinazione campo vuoto
TC_AggiungiOfferta_4	LI2, EI2, LDE2, LD1	Errore: destinazione campo vuoto
TC_AggiungiOfferta_5	LI2, EI2, LDE2, LD2, LDP1	Errore: data partenza campo vuoto
TC_AggiungiOfferta_6	LI2, EI2, LDE2, LD2, LDP2, LOP1	Errore: campo ora partenza vuoto
TC_AggiungiOfferta_7	LI2, EI2, LDE2, LD2, LDP2, LOP2, LDR1	Errore: campo data ritorno vuoto
TC_AggiungiOfferta_8	LI2, EI2, LDE2, LD2, LDP2, LOP2, LDR2, LOR1	Errore: campo ora ritorno vuoto
TC_AggiungiOfferta_9	LI2, EI2, LDE2, LD2, LDP2, LOP2, LDR2, LOR2, LPDA1	Errore: campo partenza da vuoto
TC_AggiungiOfferta_10	LI2, EI2, LDE2, LD2, LDP2, LOP2, LDR2, LOR2, LPDA2, LAA1	Errore: campo arrivo a vuoto
TC_AggiungiOfferta_11	LI2, EI2, LDE2, LD2, LDP2, LOP2, LDR2, LOR2, LPDA2, LAA2, LP1	Errore: campo pernottamento vuoto
TC_AggiungiOfferta_12	LI2, EI2, LDE2, LD2, LDP2, LOP2,	Errore: campo posti disponibili vuoto

	LDR2, LOR2, LPDA2, LAA2, LP2, LPDI1	
TC_AggiungiOfferta_13	LI2, EI2, LDE2, LD2, LDP2, LOP2, LDR2, LOR2, LPDA2, LAA2, LP2, LPDI2, IP1	Errore: campo prezzo vuoto
TC_AggiungiOfferta_15	LI2, EI2, LDE2, LD2, LDP2, LOD2, LDR2, LPD2, LAA3, LP2, LPDI2, IP2	Corretto, aggiunta offerta con successo

8.6 TC6_ModificaOfferta

Parametro: id	
CATEGORIE	SCELTE
Lunghezza id - LI	3. Lunghezza = 0 - campo vuoto [error] 4. Lunghezza >=1 - [property LD_OK]
Esiste id - EI	3. Esiste nel DB [error] 4. Non esiste nel DB [property EI_OK]

Parametro: destinazione	
CATEGORIE	SCELTE
Lunghezza destinazione- LDE	1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >=1 - [property LDE_OK]

Parametro: descrizione	
CATEGORIE	SCELTE
Lunghezza descrizione - LD	1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >=1 - [property LD_OK]

Parametro: data_partenza	
CATEGORIE	SCELTE
Lunghezza data_partenza - LDP	3. Lunghezza = 0 - campo vuoto [error] 4. Lunghezza >=1 - [property LDP_OK]

Parametro: ora_partenza	
CATEGORIE	SCELTE
Lunghezza ora_partenza - LOP	3. Lunghezza = 0 - campo vuoto [error] 4. Lunghezza >=1 - [property LOP_OK]

Parametro: data_ritorno

CATEGORIE	SCELTE
Lunghezza data_ritorno - LDR	3. Lunghezza = 0 - campo vuoto [error] 4. Lunghezza >=1 - [property LD_OK]

Parametro: ora_ritorno

CATEGORIE	SCELTE
Lunghezza ora_ritorno - LDR	3. Lunghezza = 0 - campo vuoto [error] 4. Lunghezza >=1 - [property LOR_OK]

Parametro: partenza_da

CATEGORIE	SCELTE
Lunghezza partenza_da- LPDA	3. Lunghezza = 0 - campo vuoto [error] 4. Lunghezza >=1 - [property LPD_OK]

Parametro: arrivo_a

CATEGORIE	SCELTE
Lunghezza arrivo_a - LAA	3. Lunghezza = 0 - campo vuoto [error] 4. Lunghezza >=1 - [property LAA_OK]

Parametro: pernottamento

CATEGORIE	SCELTE
Lunghezza pernottamento - LP	3. Lunghezza = 0 - campo vuoto [error] 4. Lunghezza >=1 - [property LP_OK]

Parametro: posti_disponibili

CATEGORIE	SCELTE
Quantità Posti_disponibili - LPDI	3. quantità = 0 - campo vuoto [error] 4. quantità >=1 - [property LPDI_OK]

Parametro: prezzo

CATEGORIE	SCELTE
Inserimento prezzo - IP	3. Inserimento = false [error] 4. Inserimento = true - [property IP_OK]

CODICE	COMBINAZIONE	ESITO
TC_ModificaOfferta_1	LI1	Errore: campo id vuoto
TC_ModificaOfferta_2	LI2, EI1	Errore: l'id esiste nel db
TC_ModificaOfferta_3	LI2, EI2, LDE1	Errore: destinazione campo vuoto
TC_ModificaOfferta_4	LI2, EI2, LDE2, LD1	Errore: destinazione campo vuoto
TC_ModificaOfferta_5	LI2, EI2, LDE2, LD2, LDP1	Errore: data partenza campo vuoto
TC_ModificaOfferta_6	LI2, EI2, LDE2, LD2, LDP2, LOP1	Errore: campo ora partenza vuoto
TC_ModificaOfferta_7	LI2, EI2, LDE2, LD2, LDP2, LOP2, LDR1	Errore: campo data ritorno vuoto
TC_ModificaOfferta_8	LI2, EI2, LDE2, LD2, LDP2, LOP2, LDR2, LOR1	Errore: campo ora ritorno vuoto
TC_ModificaOfferta_9	LI2, EI2, LDE2, LD2, LDP2, LOP2, LDR2, LOR2, LPDA1	Errore: campo partenza da vuoto
TC_ModificaOfferta_10	LI2, EI2, LDE2, LD2, LDP2, LOP2, LDR2, LOR2, LPDA2, LAA1	Errore: campo arrivo a vuoto
TC_ModificaOfferta_11	LI2, EI2, LDE2, LD2, LDP2, LOP2, LDR2, LOR2, LPDA2, LAA2, LP1	Errore: campo pernottamento vuoto
TC_ModificaOfferta_12	LI2, EI2, LDE2, LD2, LDP2, LOP2, LDR2, LOR2, LPDA2, LAA2, LP2, LPDI1	Errore: campo posti disponibili vuoto
TC_ModificaOfferta_13	LI2, EI2, LDE2, LD2, LDP2, LOP2, LDR2, LOR2, LPDA2, LAA2, LP2, LPDI2, IP1	Errore: campo prezzo vuoto
TC_ModificaOfferta_15	LI2, EI2, LDE2, LD2, LDP2, LOD2, LDR2, LPD2, LAA3, LP2, LPDI2, IP2	Corretto, offerta modificata con successo

8.7 TC7_AggiungiComponenti

Parametro: nome	
CATEGORIE	SCELTE
Lunghezza nome - LN	5. Lunghezza = 0 - campo vuoto [error] 6. Lunghezza >=1 - [property LN_OK]

Parametro: cognome	
CATEGORIE	SCELTE
Lunghezza cognome - LC	3. Lunghezza = 0 - campo vuoto [error] 4. Lunghezza >=1 - [property LC_OK]

Parametro: data_nascita	
CATEGORIE	SCELTE
Lunghezza data_nascita - LD	1. Lunghezza = 0 - campo vuoto [error] 2. Lunghezza >=1 - [property LD_OK]

CODICE	COMBINAZIONE	ESITO
TC_AggiungiComponenti_1	LN1	Errore: campo nome vuoto
TC_AggiungiComponenti_2	LN2, LC1	Errore: campo cognome vuoto
TC_AggiungiComponenti_3	LN2, LC2, LD1	Errore: campo data_nascita vuoto
TC_AggiungiComponenti_4	LN2, LC2, LD2	Corretto: aggiungo componente viaggio