

TESTING

Il testing è una fase essenziale del processo di sviluppo di un programma robusto ed efficiente. La probabilità di rilevare malfunzionamenti aumenta all'aumentare dei test eseguiti. Si vuole dunque rendere il software realizzato funzionante e funzionale.

Sono stati effettuati dei testing di tipo "Unit Test", attenzionando quindi i metodi, visti nella loro singolarità.

I metodi sono stati testati in modo esaustivo, analizzando per ciascuno, ove possibile, tutte le casistiche che si potrebbero verificare.

In particolar modo i testing sono stati realizzati tramite l'ausilio del framework JUnit5: un JUnit Test è una classe java contenente un insieme di metodi attraverso i quali verificare la correttezza del codice in analisi. Tra questi metodi JUnit, quelli più utilizzati sono gli appartenenti alla categoria *Assert*, utili a confrontare il risultato del metodo in analisi con un risultato atteso, in base anche al tipo di *Assert* stesso. Oltre a questo tipo di test sono stati svolti dei test manuali del sistema lanciando l'applicazione. Non sono emersi errori nel funzionamento.

CameraTests

- **TEST CALCOLA PREZZO TOTALE:**
 - **testCalcolaPrezzo:** Calcola il prezzo totale di un soggiorno. Il prezzo totale di un soggiorno di due notti in una camera che ha un prezzo per notte di 20 euro è pari a 42 euro (comprensivo delle tasse). Si verifica allora il corretto calcolo del totale.
- **TEST VERIFICA TIPO:**
 - **testVerificaTipo:** Nel test, viene creata un'istanza di Struttura e di camera, viene quindi chiamato il metodo *verificaTipo (tipologiaCamera)*. Il test utilizza gli assert *assertTrue ()* e *assertFalse ()* per verificare se la funzione restituisce il risultato atteso. Nel caso di un test positivo (la camera è dello stesso tipo di quello passato come argomento), ci si aspetta che la funzione restituisca true mentre nel caso negativo ci si aspetta che la funzione restituisca false.

StrutturaTests

- **VERIFICA ESISTENZA CAMERA:**
 - **testVerificaEsistenzaCamera:** Nel test, viene creata un'istanza di Struttura e di camera, viene quindi chiamato il metodo *verificaEsistenzaCamera (numCamera)*. Il test utilizza gli assert *assertTrue ()* e *assertFalse ()* per verificare se la funzione restituisce il risultato atteso. Nel caso di un test positivo (la struttura possiede la camera passata come argomento), ci si aspetta che la funzione restituisca true mentre nel caso negativo ci si aspetta che la funzione restituisca false.
- **TEST VERIFICA STRUTTURA BY CITTA':**
 - **testVerificaStrutturaByCittà:** Nel test, viene creata un'istanza di Struttura, viene quindi chiamato il metodo *verificaStrutturaByCittà (Città)*. Il test utilizza gli assert *assertTrue ()* e *assertFalse ()* per verificare se la funzione restituisce il risultato atteso. Nel caso di un test positivo (la struttura risiede nella città passata come argomento), ci si aspetta che la funzione restituisca true mentre nel caso negativo ci si aspetta che la funzione restituisca false.
- **TEST VERIFICA STRUTTURA BY GESTORE:**
 - **testVerificaStrutturaByGestore:** Nel test, viene creata un'istanza di Struttura, viene quindi chiamato il metodo *verificaStrutturaByIdGestore (idGestore)*. Il test utilizza gli assert *assertTrue ()* e *assertFalse ()* per verificare se la funzione restituisce il risultato atteso. Nel caso di un test positivo (la struttura è gestita dal gestore passato come argomento), ci si aspetta che la funzione restituisca true mentre nel caso negativo ci si aspetta che la funzione restituisca false.

PrenotazioneTests

- **TEST VERIFICA DISPONIBILITA':**
 - **testVerificaDisponibilita:** La funzione *verificaDisponibilita (dataArrivoReq, dataPartenzaReq)* restituisce un valore booleano true se la prenotazione coincide con la data richiesta dall'utente altrimenti false. Nel test viene creata un'istanza di Prenotazione, viene quindi chiamato il metodo *verificaDisponibilita (dataArrivoReq, dataPartenzaReq)* con due date differenti, nel caso di test positivo (la data passata come argomento coincide con quella della prenotazione), ci si aspetta che la funzione restituisca true viceversa ci si aspetta che la funzione restituisca false.
- **TEST VERIFICA DATA ELIMINAZIONE:**
 - **testVerificaDataEliminazione:** La funzione *verificaDataEliminazione ()* restituisce un valore booleano true se la prenotazione ha una distanza di almeno tre giorni dalla data della prenotazione altrimenti false. Nel test viene creata un'istanza di Prenotazione, viene quindi chiamato il metodo *verificaDataEliminazione ()*, nel caso di test positivo (la differenza di giorni è superiore a tre), ci si aspetta che la funzione restituisca true viceversa ci si aspetta che la funzione restituisca false.

JourneyHouseTests

- **TEST COSTRUTTORE:**
 - **testJourneyHouse:** È stata istanziata un'unica classe JourneyHouse ed è stato controllato il corretto funzionamento del costruttore stesso.
- **TEST RICERCA CAMERA:**
 - **testRicercaCamera:** La funzione *ricercaCamera (città, dataArrivo, dataPartenza, tipologiaCamera)* stampa le camere disponibili in base alle richieste del cliente e restituisce una lista di interi contenenti gli id delle camere disponibili. Nel test si verifica che le camere disponibili per le richieste inserite siano in numero pari a 2.
- **TEST SCELTA CAMERA E PAGAMENTO ONLINE:**
 - **testFinePrenotazione:** Viene scelta una camera e viene scelto il pagamento tramite carta di credito, si verifica infine che la prenotazione si andata a buon fine.
- **TEST INSERIMENTO CAMERA:**
 - **testInserisciCamera:** La funzione *inserisciCamera (numeroCamera, numeroPiano, tipoCamera, prezzoPerNotte)* restituisce un valore booleano true se l'inserimento della camera è avvenuto con successo viceversa false nel caso in cui il numero della camera inserita dall'utente è già esistente. Viene chiamata la funzione *inserisciCamera (numeroCamera, numeroPiano, tipoCamera, prezzoPerNotte)*, viene utilizzato l'id corretto e dunque viene inserita la camera, si utilizza dunque *assertTrue* per verificare la restituzione del parametro true.
- **TEST MODIFICA CAMERA:**
 - **testModificaCamera:** La funzione *modificaCamera (idCamera, prezzoPerNotte)* restituisce un valore booleano true se la modifica della camera è avvenuta con successo viceversa false nel caso in cui l'id della camera non è esistente. Viene chiamata la funzione *modificaCamera (idCamera, prezzoPerNotte)*, viene utilizzato l'id corretto e dunque viene modificato il prezzo della camera, si utilizza dunque *assertTrue* per verificare la restituzione del parametro true.
- **TEST ELIMINA PRENOTAZIONE:**
 - **testEliminaPrenotazione:** La funzione *eliminaPrenotazione (idPrenotazione)* restituisce un valore booleano true se l'eliminazione della prenotazione è avvenuta con successo viceversa false nel caso in cui l'id della prenotazione inserito dall'utente non dovesse esistere o nel caso in cui l'utenteCorrente non è l'autore della prenotazione che si desidera eliminare. Viene

chiamata tre volte la funzione *eliminaPrenotazione (idPrenotazione)*, la prima volta viene utilizzato l'id corretto e dunque viene eliminata la prenotazione, si utilizza dunque *assertTrue* per verificare la restituzione del parametro true, nel secondo test si inserisce un id errato e si utilizza *assertFalse* per verificare se la funzione restituisce false, nel terzo test si inserisce l'id di una prenotazione che non appartiene all'utenteCorrente.

MessaggioRecensioneHandlerTests

- TEST INSERIMENTO RECENSIONE:

- **testInserimentoRecensione:** La funzione *inserisciRecensione (testo, voto, idStruttura)* restituisce un valore booleano true se l'inserimento della recensione è avvenuto con successo viceversa false nel caso in cui l'id inserito dall'utente non dovesse esistere. Viene chiamata due volte la funzione *inserisciRecensione (testo, voto, idStruttura)*, la prima volta viene utilizzato l'id corretto e dunque viene inserita la recensione, si utilizza dunque *assertTrue* per verificare la restituzione del parametro true, nel secondo test si inserisce un id errato e si utilizza *assertFalse* per verificare se la funzione restituisce false.

- TEST MODIFICA RECENSIONE:

- **testModificaRecensione:** La funzione *modificaRecensione (idRecensione, testo, voto)* restituisce un valore booleano true se la modifica della recensione è avvenuta con successo viceversa false nel caso in cui l'id della recensione inserito dall'utente non dovesse esistere o nel caso in cui l'utenteCorrente non è l'autore della recensione da eliminare. Viene chiamata tre volte la funzione *modificaRecensione (idRecensione, testo, voto)*, la prima volta viene utilizzato l'id corretto e dunque viene modificata la recensione, si utilizza dunque *assertTrue* per verificare la restituzione del parametro true, nel secondo test si inserisce un id errato e si utilizza *assertFalse* per verificare se la funzione restituisce false, nel terzo test si inserisce l'id di una recensione che non appartiene all'utenteCorrente.

- TEST ELIMINA RECENSIONE:

- **testEliminaRecensione:** La funzione *eliminaRecensione (idRecensione)* restituisce un valore booleano true se l'eliminazione della recensione è avvenuta con successo viceversa false nel caso in cui l'id della recensione inserito dall'utente non dovesse esistere o nel caso in cui l'utenteCorrente non è l'autore della recensione di cui si desidera effettuare la modifica. Viene chiamata tre volte la funzione *eliminaRecensione (idRecensione)*, la prima volta viene utilizzato l'id corretto e dunque viene eliminata la recensione, si utilizza dunque *assertTrue* per verificare la restituzione del parametro true, nel secondo test si inserisce un id errato e si utilizza *assertFalse* per verificare se la funzione restituisce false, nel terzo test si inserisce l'id di una recensione che non appartiene all'utenteCorrente.

- TEST INSERIMENTO MESSAGGIO:

- **testInserimentoMessaggio:** La funzione *inserisciMessaggio (testo, idUtenteDestinatario)* restituisce un valore booleano true se l'inserimento del messaggio è avvenuto con successo viceversa false nel caso in cui l'id inserito dall'utente non dovesse esistere. Viene chiamata due volte la funzione *inserisciMessaggio (testo, idUtenteDestinatario)*, la prima volta viene utilizzato l'id corretto e dunque viene inserito il messaggio, si utilizza dunque *assertTrue* per verificare la restituzione del parametro true, nel secondo test si inserisce un id di un utente destinatario errato e si utilizza *assertFalse* per verificare se la funzione restituisce false.

LoginRegGestReportHandlerTests

- **TEST LOGIN:**
 - **testAccesso:** Nel test viene creata un'istanza di Utente (Cliente, Gestore o Admin) viene quindi chiamato il metodo *accesso(id, password)* effettuando il test inserendo id e password corretti, id corretto ma password sbagliata, id errato ed infine viene differenziata la tipologia di login, ad esempio, un utente cliente non può effettuare il login da admin. Nel caso di test positivo (accesso effettuato correttamente), ci si aspetta che la funzione restituisca true viceversa ci si aspetta che la funzione restituisca false.
- **TEST MODIFICA PASSWORD:**
 - **testModificaPassword:** La funzione *modificaPassword(idUtente, nuovaPassword)* restituisce un valore booleano true se l'operazione di modifica è stata eseguita con successo viceversa false. Viene chiamata due volte la funzione *modificaPassword(idUtente, nuovaPassword)*, la prima volta viene utilizzato l'id corretto e dunque viene assegnata una nuova password, si utilizza dunque *assertTrue* per verificare la restituzione del parametro true nel secondo test si inserisce un id errato e si utilizza *assertFalse* per verificare se la funzione restituisce false.
- **TEST MODIFICA EMAIL:**
 - **testModificaEmail:** La funzione *modificaEmail(idUtente, nuovaEmail)* restituisce un valore booleano true se l'operazione di modifica è stata eseguita con successo viceversa false. Viene chiamata due volte la funzione *modificaEmail(idUtente, nuovaEmail)*, la prima volta viene utilizzato l'id corretto e dunque viene assegnata una nuova e-mail, si utilizza dunque *assertTrue* per verificare la restituzione del parametro true nel secondo test si inserisce un id errato e si utilizza *assertFalse* per verificare se la funzione restituisce false.