# Testing

## File AuthenticationTest

In tale file si è testato il corretto funzionamento delle funzioni di login e registrazione presenti nella classe “FitnessReady”. Prima di eseguire i test di seguito descritti viene eseguita una funzione di setUp che genera una lista di utenti di prova necessari per la verifica dei test e con cui si recupera l’unica istanza presente del sistema FitnessReady.

### **Login Test**

In questo test è stato visionato il corretto funzionamento del metodo effettuaLogin(…) del sistema FitnessReady, la quale prende in ingresso l’email e la password dell’utente, sia esso Cliente o Amministratore, e restituisce i privilegi dell’utente loggatosi da cui derivano le funzioni che esso potrà effettuare grazie al suddetto sistema. Poiché il metodo effettuaLogin(…) esegue dei controlli per determinare che i parametri passatigli siano corretti essa lancia diversi tipi di eccezioni (appositamente create per le diverse situazioni), si è quindi deciso di basare il test sia sul lancio di tali eccezioni verificandole tramite la funzione assertThrows(), con cui si confrontano l’eccezione attesa con quella effettivamente ricevuta, sia sul corretto login dei due utenti (un Cliente ed un Amministratore) tramite la funzione assertEquals(), con cui si confrontano i privilegi attesi con quelli recuperati.

### **Sing-In Test**

In questo test è stato visionato il corretto funzionamento del metodo registraCliente(…) del sistema FitnessReady, il quale prende in ingresso nome, cognome, e-mail e password del cliente oltre a confermaPassword (necessario al cliente per il corretto inserimento della sua password). Dopo aver specificato queste informazioni, il metodo generà ed aggiungerà un nuovo cliente con tali attributi alla lista di utenti posseduta dal sistema. Poiché anche nel metodo sopra descritto vengono lanciate diverse eccezioni in caso d’inserimento dati scorretto, per effettuare il test sono stati forniti dati intenzionalmente errati o già presenti per provocare il lancio delle relative eccezioni, per la cui verifica si è ricorso nuovamente alla funzione assertThrows(), per poi creare un nuovo cliente con cui confrontare quello inserito nella lista utenti usando il metodo registraCliente() tramite la funzione assertEquals().

## File PurchaseTest

In tale file si è testato il corretto funzionamento della classe “GestoreAcquisto”, che gestisce tutte le operazioni relative all’acquisto di uno o più prodotti. Poiché uno stesso gestore acquisto permane solo per la durata (o l’annullamento) di un singolo acquisto di uno o più prodotti, per una corretta esecuzione dei test di seguito descritti viene innanzitutto recuperata l’unica istanza del sistema FitnessReady ed in più, prima di ogni test, il gestore acquisto viene re inizializzato.

### **Add-To-Cart Test**

In questo test è stato visionato il corretto funzionamento del metodo aggiungiAlCarrello(…) del GestoreAcquisto, il quale ha il compito di inserire il prodotto selezionato nell’apposito carrello d’acquisto e per questo tale metodo prende in ingresso il suddetto prodotto e la quantità desiderata dal cliente. Il metodo in questione si occupa sia di effettuare dei controlli sui parametri d’ingresso i quali se quest’ultimi non sono corretti, lanciano diversi tipi di eccezioni a seconda dell’errore riscontrato, sia di ritornare un parametro booleano per determinare se il prodotto che si cerca di aggiungere sia o no già presente nel carrello. Per questa ragione il test si concentra prima sulla corretta ricezione di tali eccezioni in caso di errato inserimento o selezione del prodotto tramite la funzione assertThrows(), e poi sulla verifica del corretto inserimento di un prodotto appositamente creato, cercando d’inserire quest’ultimo due volte e verificandone l’esito tramite le funzioni asserTrue(), per il primo tentativo, e assertFalse(), per il secondo tentativo.

## **Execute-Purchase Test**

Questo test ci permette di controllare il corretto funzionamento dell’effettiva esecuzione di un nuovo acquisto relativo ai prodotti inseriti nel carrello durante la fase di selezione. Il metodo della classe GestoreAcquisto interessato in questo test è effettuaAcquisto(), il quale restituisce l’acquisto che viene concluso in modo tale che venga aggiunto alla lista acquisti relativa del cliente che lo effettua. Nel test viene prima verificata l’impossibilità di effettuare un acquisto nel caso in cui il carrello sia vuoto tramite la funzione assertThrows(), poiché in tali condizioni il metodo lancia un’eccezione, e poi viene controllata la corretta esecuzione dell’acquisto confrontando il risultato ottenuto tramite il metodo qui trattato con un acquisto appositamente creato per il confronto grazie alla funzione assertEquals().