



## 1. Introduzione

3GTech nasce come azienda specializzata nella vendita di prodotti tecnologici quali console, smartphone, elettrodomestici ed accessori.

Con il documento di Test Planning si vanno a definire, descrivere ed analizzare le attività di testing per il sistema 3GTech, al fine di garantire il corretto funzionamento dello stesso. Le funzionalità soggette a testing:

- Gestione Utente
- Gestione Prodotto
- Gestione Carrello
- Gestione Ordini

Per effettuare tali attività di testing sono definite, all'interno del documento, le strategie di testing scelte e gli strumenti adottati per la rilevazione degli errori.

## 2. Relazione con altri documenti

- **Relazione con il Requirements Analysis Document (RAD):** i test case pianificati all'interno di questo documento sono stati definiti in base ai requisiti funzionali e non funzionali descritti all'interno del RAD.
- **Relazione con il System Design Document (SDD):** i test case sviluppati devono testare tutte le funzionalità di ogni sottosistema definito nel SDD.
- **Relazione con l' Object Design Document (ODD):** verranno effettuati dei test per garantire l'efficacia dell'interfaccia.

## 3. Panoramica del sistema

Il sistema che stiamo sviluppando ha un'architettura di tipo THREE-TIER ovvero:

- **Presentation tier**  
Nel presentation layer utilizzeremo le tecnologie HTML per la scrittura delle pagine web, CSS per dare delle grafiche personalizzate al sito, Javascript per delle animazioni e Ajax per alcune funzioni specifiche
- **Application tier**  
Si codificherà il codice in linguaggio java e tale codice verrà deployato sul container-server TOMCAT di apache con l'utilizzo delle librerie GSON per convertire gli oggetti in json, json per appunto utilizzarlo, mysql-connector per permettere a tomcat di comunicare con il database mysql
- **Data tier**  
Il database utilizzato è MySQL

## 4. Funzionalità da testare/ da non testare

Le features che verranno testate sono le seguenti, catalogate in base ai servizi definiti nel documento SDD.3. Sono state scelte principalmente le funzionalità che prevedano input da parte dell'utente.

### -Gestione Utente

Login  
Registrazione  
ModificaPassword  
ModificaEmail  
ModificaIndirizzo

#### -Gestione Prodotto

InserisciProdotto  
RimuoviProdotto  
Cerca  
ModificaDescrizione  
ModificaPrezzo

#### -GestioneOrdini

VisualizzaOrdiniDiUnUtente  
Pagamento

## 5. Criteri di successo e insuccesso

Un test si può considerare di successo nel caso in cui si ottiene una (o più) failure, ovvero quando il risultato ottenuto è diverso da quello atteso e specificato nell'oracolo. Dunque in tal caso si procederà alla correzione della funzione.

Analogamente, un test si considera fallito nel caso in cui l'output ottenuto rispetta le previsioni dell'oracolo, pertanto non sono state identificate failures.

## 6. Approccio

Il test si compone di tre fasi:

- Testing di Unità, con l'obiettivo di testare ogni sottosistema in modo da verificare la correttezza del codice e delle singole funzionalità. Verrà svolto con l'ausilio di JUnit, mediante l'implementazione di metodi annotati con @ParameterizedTest.
- Test di Integrazione, con l'obiettivo di testare gruppi di sottosistemi.
- Test di Sistema, per testare il sistema nella sua interezza.

Il Testing di Unità verrà svolto utilizzando la tecnica Black-Box, dunque si avrà cura solo ed esclusivamente di input/output e non del codice in sé per esaminare il corretto funzionamento delle singole unità.

Attueremo la tecnica del Black-Box mediante un Equivalence-Testing, in modo da ridurre il numero di test case creando delle classi di equivalenza.

Il Testing di Integrazione porterà a galla errori non rilevati nella fase precedente, integrando vari sottosistemi che sono risultati corretti nel Testing di Unità.

Verrà utilizzata la tecnica Bottom-Up, testando prima le componenti del livello più basso e successivamente integrando unità del livello sovrastante.

Il Test di Sistema infine ci consente di verificare che il sistema rispetti i requisiti funzionali e non funzionali inizialmente definiti. Verrà utilizzato l'IDE Selenium per effettuare i vari test direttamente sul browser.

## 7. Sospensione e ripresa

Definizione dei criteri di sospensione del test e delle attività da ripetere nel momento in cui si riprende il test.

- **Criteri di sospensione:** la fase di testing non verrà mai sospesa, a meno che si verifichi un errore del test stesso. Nel caso in cui venga trovato un failure questo verrà riportato nel file contenente i Report.
- **Criteri di ripresa:** il test verrà ripreso dopo aver ovviato ad eventuali problemi dello stesso.

## 8. Materiale di prova (requisiti hardware/software)

Per testare il sistema in locale l'utente ha bisogno dei seguenti requisiti:

- **Hardware :**
  - o Un computer di qualsiasi genere con architettura x86 e che abbia almeno:
    - processore intel i3/ryzen 3
    - 8gb di memoria ram
    - 50gb su spazio disco
- **Software preinstallati:**
  - o JVM (java virtual machine) versione 11
  - o TOMCAT versione 9
  - o MySQL ultima versione sul mercato
  - o JUnit e Mockito per l'implementazione di Test di Unità e di Integrazione
  - o Selenium IDE per lo svolgimento del Test di Sistema

## 9. Casi di test

L'approccio che verrà utilizzato per il test frame sarà il category partition, con la partizione degli input in classi di equivalenza.

### 9.1 Gestione Utente

#### 9.1.1 TC1 – Registrazione

Parametro: Password	
Categoria	Scelta
Lunghezza [LP]	1. LP<=20 [OK] 2. LP> 20 [ERROR]
Formato	<code>/^(?=.*[A-Za-z])(?=.*\d)(?=.*[\$@-!%*#?&amp;])[A-Za-z\d\$@-!%*#?&amp;]{8,}\$/</code>
Formato [FP]	1. Formato Rispettato = true [OK] 2. Formato Rispettato = false [ERROR]
Parametro: Email	
Lunghezza [LE]	1. Lunghezza <=30 [OK] 2. Lunghezza > 30 [ERROR]
Formato	<code>/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)(\.\w{2,9})+\$/</code>
Formato [FE]	1. Formato Rispettato = true [OK] 2. Formato Non Rispettato = false [ERROR]
Validità [VE]	1. La mail non è presente nel Database [OK] 2. La mail è presente nel Database [ERROR]

Parametro: Nome	
Formato	/^[A-Za-z]+\$ /
Lunghezza [LNO]	1. LNO > 0 && LNO <= 30 [OK] 2. LNO > 30    LNO < 1 [ERROR]
Formato [FNO]	1. Formato rispettato = true [OK] 2. Formato rispettato = false[ERROR]
Parametro : Cognome	
Lunghezza [LC]	1. LC> 0 && LC <= 30 [OK] 2. LC > 30    LC < 1 [ERROR]
Formato	/^[A-Za-z]+\$ /
Formato [FC]	1. Formato Rispettato = true [OK] 2. Formato Rispettato = false[ERROR]
Parametro: Indirizzo	
Lunghezza [LI]	1. LI > 0 && LI <= 30 [OK] 2. LI <1    LI > 30 [ERROR]
Formato	/^[A-z]+ [A-z]+ \d{1,3}\$ /
Formato [FI]	1. Formato Rispettato = true[OK] 2. Formato Rispettato = false[ERROR]
Parametro : Telefono	
Lunghezza [LT]	1. LT = 10 [OK] 2. LT != 10 [ERROR]
Formato	/^\d{10}\$ /
Formato [FT]	1. Formato Rispettato = true [OK] 2. Formato Rispettato = false[ERROR]

Codice TC	Test Frame	Esito
TC1.1	LP1, FP1, LE1, FE1, VE1, LNO1, FNO1, LC1, FC1, LI1, FI1, LT1, FT1	CORRETTO: utente registrato
TC1.2	LP2	ERRATO: password troppo lunga
TC1.3	LP1, FP2	ERRATO: formato password non rispettato
TC1.4	LP1, FP1,LE2	ERRATO: email troppo lunga

TC1.5	LP1, FP1, LE1, FE2	ERRATO: formato email non rispettato.
TC1.6	LP1, FP1, LE1, FE1, VE2	ERRATO: email già presente all'interno del sistema
TC1.7	LP1, FP1, LE1, FE1, VE1, LNO2	ERRATO: nome troppo lungo
TC1.8	LP1, FP1, LE1, FE1, VE1, LNO1, FNO2	ERRATO: formato nome non rispettato
TC1.9	LP1, FP1, LE1, FE1, VE1, LNO1, FNO1, LC2	ERRATO: cognome troppo lungo
TC1.10	LP1, FP1, LE1, FE1, VE1, LNO1, FNO1, LC1, FC2	ERRATO: formato cognome non rispettato
TC1.11	LP1, FP1, LE1, FE1, VE1, LNO1, FNO1, LC1, FC1, LI2	ERRATO: indirizzo troppo lungo
TC1.12	LP1, FP1, LE1, FE1, VE1, LNO1, FNO1, LC1, FC1, LI1, FI2	ERRATO: formato indirizzo non rispettato
TC1.13	LP1, FP1, LE1, FE1, VE1, LNO1, FNO1, LC1, FC1, LI1, FI1, LT2	ERRATO: lunghezza telefono non rispettata
TC1.14	LP1, FP1, LE1, FE1, VE1, LNO1, FNO1, LC1, FC1, LI1, FI1, LT1, FT2	ERRATO: formato telefono non rispettato

## 9.1.2 - TC2 - Login

Parametro: Password	
Categoria	Scelta
Lunghezza [LP]	1. LP<=20 [OK] 2. LP> 20 [ERROR]
Parametro: Email	
Lunghezza [LE]	1. Lunghezza <=30 [OK] 2. Lunghezza > 30 [ERROR]
Formato	/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)(\.\w{2,9})+\$/
Formato [FE]	1. Formato Rispettato = true [OK] 2. Formato Non Rispettato = false [ERROR]
Validità [VE]	1. La mail è presente nel Database [OK] 2. La mail non è presente nel Database [ERROR]

Codice TC	Test Frame	Esito
TC2.1	LP1, FP1, LE1, FE1	CORRETTO: l'utente effettua il login
TC2.2	LP2	ERRATO: password troppo lunga
TC2.3	LP1, LE2	ERRATO: email troppo lunga
TC2.4	LP1, LE1, FE2	ERRATO: formato email non rispettato.
TC2.5	LP1, LE1, FE1, VE2	ERRATO: la mail non è presente nel database

### 9.1.3 - TC3 - Modifica Password

Parametro: Password	
Categoria	Scelta
Lunghezza [LP]	1. LP<=20 [OK] 2. LP> 20 [ERROR]
Formato	<code>/^(?=.*[A-Za-z])(?=.*\d)(?=.*[\$@-_%*#?&amp;])[A-Za-z\d\$@-_%*#?&amp;]{8,}\$</code>
Formato [FP]	1. Formato Rispettato = true [OK] 2. Formato Rispettato = false [ERROR]

Codice TC	Test Frame	Esito
TC3.1	LP1, FP1	CORRETTO: password modificata con successo
TC3.2	LP2	ERRATO: password troppo lunga
TC3.3	LP1, FP2	ERRATO: formato password non rispettato

### 9.1.4 - TC4 - Modifica Email

Parametro: Email	
Categoria	Scelta
Lunghezza [LE]	1. Lunghezza <=30 [OK] 2. Lunghezza > 30 [ERROR]
Formato	<code>/^[w+([\.-]?w+)*@w+([\.-]?w+)(\.[w]{2,9})+\$/</code>
Formato [FE]	1. Formato Rispettato = true [OK] 2. Formato Non Rispettato = false [ERROR]
Validità [VE]	1. La mail non è presente nel Database [OK] 2. La mail è presente nel Database [ERROR]

Codice TC	Test Frame	Esito
TC4.1	LE1, FE1, VE1	CORRETTO: email modificata con successo
TC4.2	LE2	ERRATO: email troppo lunga
TC4.3	LE1, FE2	ERRATO: formato email non rispettato
TC4.4	LE1, FE1, VE2	ERRATO: email già presente nel sistema

### 9.1.5 - TC5 - Modifica Indirizzo

Parametro: Indirizzo	
Categoria	Scelta
Lunghezza [LI]	1. LI > 0 && LI <= 30 [OK] 2. LI <1    LI > 30 [ERROR]
Formato	/^[A-z]+ [A-z]+ \d{1,3}\$/
Formato [FI]	1. Formato Rispettato = true[OK] 2. Formato Rispettato = false[ERROR]

Codice TC	Test Frame	Esito
TC5.1	LI1, FI1	CORRETTO: Indirizzo modificato con successo
TC5.2	LI2	ERRATO: indirizzo troppo lungo
TC5.3	LI1, FI2	ERRATO: formato indirizzo non rispettato

## 9.2. Gestione Prodotto

### 9.2.1 TC6 – InserisciProdotto

Parametro: Nome	
Categoria	Scelta
Lunghezza [LNP]	1. LNP>0 && LNP<=100 [OK] 2. LNP==0    LNP> 100 [ERROR]
Parametro: Descrizione	
Lunghezza [LDP]	1. LDP<=300 [OK] 2. LDP> 300 [ERROR]
Parametro: Prezzo	
Prezzo [PDP]	1. PDP > 0 [OK] 2. PDP <= 0 [ERROR]

Codice TC	Test Frame	Esito
TC6.1	LNP1, LDP1, PDP1	CORRETTO , prodotto aggiunto.
TC6.2	<b>LNP2</b>	ERRATO, inserire nome correttamente.
TC6.3	LNP1, <b>LDP2</b>	ERRATO, la descrizione è troppo lunga.
TC6.4	LNP1, LDP1, <b>PDP2</b>	ERRATO, inserisci un prezzo valido.



### 9.2.2. TC7 - RimuoviProdotto

Parametro: Codice	
Categoria	Scelta
Validità[VCP]	1. VCP è l'ID di un prodotto presente nel database[OK] 2. VCP non è l'ID di un prodotto presente nel database [ERROR]

Codice TC	Test Frame	Esito
TC7.1	VCP1	CORRETTO, prodotto rimosso.
TC7.2	VCP2	ERRATO, inserire un codice valido.yy

### 9.2.3. TC8 - Cerca

Parametro: Nome	
Categoria	Scelta
Lunghezza [LNP]	1. LNP $\geq$ 2 [OK] 2. LNP $<$ 2 [ERROR]
Validità [VNP]	1. VNP corrisponde ad una sottostringa di un prodotto [OK] 2. VNP non corrisponde alla sottostringa di alcun prodotto [ERROR]

Codice TC	Test Frame	Esito
TC8.1	LNP1, VNP1	CORRETTO, il prodotto (o i prodotti) trovati vengono mostrati.
TC8.2	LNP2	ERRATO, inserire una stringa di almeno due caratteri.
TC8.3	LNP1, VNP2	ERRATO, il prodotto cercato non è in catalogo.

### 9.2.4. TC9 - ModificaDescrizione

Parametro: Codice	
Categoria	Scelta
Validità [VCP]	1. VCP è l'ID di un prodotto presente nel database[OK] 2. VCP non è l'ID di un prodotto presente nel database [ERROR]
Parametro: Descrizione	
Lunghezza [LDP]	1. LDP $\leq$ 300 [OK] 2. LDP $>$ 300 [ERROR]

Codice TC	Test Frame	Esito
TC9.1	VCP1, LDP1	CORRETTO, prodotto modificato correttamente.
TC9.2	<b>VCP2</b>	ERRATO, il codice inserito non corrisponde ad alcun prodotto.
TC9.3	VCP1, <b>LDP2</b>	ERRATO, la descrizione è troppo lunga.

### 9.2.5. TC10 – ModificaPrezzo

Parametro: Codice	
Categoria	Scelta
Validità [VCP]	1. VCP è l'ID di un prodotto presente nel database[OK] 2. VCP non è l'ID di un prodotto presente nel database [ERROR]
Parametro: Prezzo	
Prezzo [PP]	1. PP > 0 [OK] 2. PP <= 0 [ERROR]

Codice TC	Test Frame	Esito
TC10.1	VCP1, PP1	CORRETTO, prodotto modificato correttamente.
TC10.2	<b>VCP2</b>	ERRATO, il codice inserito non corrisponde ad alcun prodotto.
TC10.3	VCP1, <b>PP2</b>	ERRATO, inserire un prezzo valido.

## 9.3. Gestione Ordini

### 9.3.1. TC11 - VisualizzaOrdiniDiUnUtente

Parametro: Email	
Validità[VCP]	1. VCP è l'e-mail di un cliente esistente [OK] 2. VCP è un'e-mail che non esiste nel database [ERROR]

Codice TC	Test Frame	Esito
TC11.1	VCP1	CORRETTO, visualizzazioni degli ordini di quel cliente.
TC11.2	<b>VCP2</b>	ERRATO, non è possibile visualizzare gli ordini

### 9.3.3. TC12 - ProcediAlPagamento

Parametro: Numerocarta	
Lunghezza [LN]	1. LN == 10 [OK] 2. LN != 10 [ERRORE]
Parametro: Cvc	
Lunghezza [LC]	1. LC == 3[OK] 2. LC != 3[ERRORE]
Parametro: Titolare	
Lunghezza [LT]	1. LT <= 30 && LT>=0 [OK] 2. LT > 30    LT<0[ERRORE]

Codice TC	Test Frame	Esito
TC12.1	LN1, LC1, LT1	CORRETTO
TC12.2	<b>LN2</b>	[ERRATO] Lunghezza NUMERO CARTA non corretta
TC12.3	LN1, <b>LC2</b>	[ERRATO] Lunghezza CVC non corretta
TC12.4	LN1, LC1, <b>LT2</b>	[ERRATO] Lunghezza NOME titolare non corretta.