

# *Università degli studi di Salerno*

*Corso di Laurea in Informatica*

*Ingegneria del Software*

***FreshFood***

## **Studenti:**

Anna Fulgione

0512105572

Martina Casalnuovo

0512105110

*Anno Accademico: 2019/20*

## Sommario

TEST PLAN .....	3
1. Introduzione .....	3
2. Relazione con altri documenti .....	3
2.1 Relazioni con il documento di Analisi dei Requisiti (RAD) .....	3
2.2 Relazioni con il documento di System Design (SDD) .....	3
2.3 Relazioni con il documento di Object Design (ODD) .....	3
3. Panoramica del sistema .....	3
4. Funzionalità da testare e non testare .....	4
5. Criteri PASS or FAIL .....	4
Pass Criteria .....	4
Fail Criteria .....	4
6. Approccio .....	5
6.1 Testing di unità .....	5
6.2 Testing di integrazione.....	5
6.3 Testing di sistema .....	5
7. Sospensione e ripresa.....	6
7.1 Criteri di sospensione .....	6
7.2 Criteri di ripresa .....	6
8. Materiali di prova .....	6
9. Casi di prova .....	7
9.1 Sottosistema Gestione_utente .....	7
9.1.1 Login .....	7
9.1.2 Visualizzazione area utente .....	7
9.2 Sottosistema Gestione_prodotto .....	8
9.2.1 Inserimento nuovo prodotto .....	8
9.2.2 Acquisto.....	10
9.2.3 Ricerca .....	13
10.Programma dei test .....	14

# TEST PLAN

## 1. Introduzione

Lo scopo del documento è quello di definire i test case nei quali verranno descritti i test sulle funzionalità del sistema. Per ogni funzionalità saranno forniti un numero sufficiente di istanze di input in modo tale da fornire almeno un test case composto da dati corretti.

## 2. Relazione con altri documenti

### 2.1 Relazioni con il documento di Analisi dei Requisiti (RAD)

Tali relazioni riguardano i requisiti funzionali e non funzionali del sistema. I test delle funzionalità terranno conto delle specifiche espresse nel RAD. I test devono tener presente i diversi attori descritti nel documento di analisi (RAD):

- Utente Registrato
- Utente Non Registrato
- Gestore del catalogo
- Amministratore

Gli attori Amministratore, Utente Non Registrato non saranno interessati dalla fase di testing.

### 2.2 Relazioni con il documento di System Design (SDD)

La pianificazione dei test delle componenti rispecchia la decomposizione in sottosistemi specificata nel documento di System Design.

### 2.3 Relazioni con il documento di Object Design (ODD)

La fase di testing deve considerare il documento di Object Design in quanto esso fornisce la base per realizzare l'implementazione. Pertanto è necessario effettuare il testing delle unità per individuare le differenze tra il comportamento atteso specificato.

## 3. Panoramica del sistema

Il test plan descrive i test unitari relativi a particolari sottosistemi della piattaforma FreshFood, già descritti nel documento di System Design, quali:

- Sottosistema *gestione utente*
- Sottosistema *gestione prodotto*

#### 4. Funzionalità da testare e non testare

Si è scelto di testare alcune tra le principali funzionalità del sistema in base ai sottosistemi descritti sopra:

- Sottosistema Gestione\_utente:
  - *login*,
  - *visualizzazione area utente*
- Sottosistema Gestione\_prodotto:
  - *inserimento* di un nuovo prodotto,
  - *rimozione* di un prodotto,
  - *acquisto*,
  - *ricerca*

Non si è scelto, ad esempio, di testare:

- Sottosistema Gestione\_utente:
  - *aggiunta* di uno o più prodotti al carrello,
  - *visualizzazione del carrello*,
  - *logout*

#### 5. Criteri PASS or FAIL

La fase di testing necessita di criteri formali per la determinazione del successo o dell'insuccesso di un determinato test. I criteri, che decreteranno il *pass* o il *fail* di un determinato test, saranno basati sul confronto tra il risultato atteso e quello ottenuto.

##### Pass Criteria

I *pass criteria* indicano l'insuccesso del test svolto su una specifica componente, per cui il comportamento risulta corretto.

Ci sono due casi:

1. L'output del test rileva il comportamento atteso.
2. L'output del test non rileva alcun errore.

##### Fail Criteria

I *fail criteria* indicano il successo del test svolto su una specifica componente, per cui il comportamento risulta non corretto.

Ci sono due casi:

1. L'output del test rileva un errore.
2. L'output del test rileva un comportamento inatteso.

## 6. Approccio

L'approccio usato è suddiviso in più fasi di testing: innanzitutto con il testing di unità, attraverso il quale verranno testate singolarmente le componenti nella loro correttezza.

Dopodiché si effettuerà il testing di integrazione mirato al funzionamento delle interfacce dell'unità.

Infine verrà effettuato il testing di sistema con l'obiettivo di verificare il comportamento dell'intero sistema assemblato e con la verifica sul rispetto delle caratteristiche richieste dal committente.

### 6.1 Testing di unità

In questa fase vengono testate le singole unità con lo scopo di trovare, individualmente, le condizioni di fallimento all'interno delle componenti e verificare i comportamenti per ogni unità.

La tecnica di test usata è la cosiddetta **Black Box Testing**:

l'obiettivo è quello di determinare se il programma fa quello che si suppone debba fare (oracolo) in base ai requisiti funzionali. Per la costruzione dei test ci si avvale solo delle specifiche dei requisiti, ignorando completamente come sia stato realizzato il sistema al suo interno. Se ben condotto riesce a determinare il manifestarsi di molti malfunzionamenti funzionali e può evidenziare il mancato rispetto di qualche requisito funzionale.

### 6.2 Testing di integrazione

In questa fase si utilizza una strategia **sandwich** tramite la quale il test viene svolto sull'aggregazione di più componenti e il loro relativo testing. In questo caso il testing mira alla ricerca di malfunzionamenti all'interno delle funzionalità del sistema.

### 6.3 Testing di sistema

In quest'ultima fase, il testing viene rivolto alle principali funzionalità di tutto il sistema e si adotta la tecnica **black-box**, in particolare si verifica che i requisiti e i vincoli di progettazione siano rispettati. Solo dopo quest'ultimo testing si potrà determinare se il sistema è pronto per l'uso da parte del cliente o meno.

## 7. Sospensione e ripresa

La fase di testing del sistema **FreshFood** può essere interrotta e ripresa più volte poiché l'obiettivo è quello di rendere il sistema corretto ed ogni funzionalità completa. Di seguito sono riportati i criteri secondo cui è necessario sospendere la fase di testing e le modalità secondo cui deve poi essere ripresa.

### 7.1 Criteri di sospensione

La sospensione della fase di testing deve avvenire qualora un test abbia esito positivo, ovvero si è riscontrato un errore all'interno di una componente.

### 7.2 Criteri di ripresa

La ripresa della fase di testing avviene a partire dal test case che ne ha causato la sospensione, soltanto dopo che gli sviluppatori hanno corretto l'errore riscontrato.

## 8. Materiali di prova

Per le attività di testing sono necessari:

- **Hardware**
  - Elaboratore su cui eseguire il software.
- **Software**
  - Eclipse IDE
  - MySql DBM
  - Selenium
  - JUnit

## 9. Casi di prova

### 9.1 Sottosistema Gestione\_utente

#### 9.1.1 Login

Parametro: username	
Formato	<code>/^\w+([\.-]?\w+)@\w+([\.-]?\w+)(\.\w{2,3})+\$/</code>
Lunghezza [L]	<ol style="list-style-type: none"><li>&lt;6 [error]</li><li>&gt;6 [correct]</li></ol>
Esistenza [E]	<ol style="list-style-type: none"><li>L'username esiste nel sistema [correct][exist]</li><li>L'username non esiste nel sistema [error]</li></ol>

Parametro: password	
Formato	<code>[A-Za-z0-9.]</code>
Lunghezza [L]	<ol style="list-style-type: none"><li>&lt;6 [error]</li><li>&gt;6 [correct]</li></ol>
Correttezza [C]	<ol style="list-style-type: none"><li>La password coincide [correct][exist]</li><li>La password non coincide [error]</li></ol>

Codice	Combinazione	Esito
TC1.1.1	L[1], L[3]	ERROR
TC1.1.2	L[2], L[4], C[4]	ERROR
TC1.1.3	L[2], L[4], C[3]	CORRECT

#### 9.1.2 Visualizzazione area utente

Il test su questa funzionalità non prevede alcuna compilazione di form o semplici caselle di testo, per cui non è riportata alcuna tabella esplicativa.

Il test ha l'obiettivo di verificare la raggiungibilità della propria area utente dopo aver effettuato l'autenticazione.

## 9.2 Sottosistema Gestione\_prodotto

### 9.2.1 Inserimento nuovo prodotto

Parametro: Nome	
Formato	[A-Za-z.]
Lunghezza [L]	1. <2 [error] 2. >2 [correct]
Correttezza [C]	1. Non rispetta il formato [error] 2. Rispetta il formato [correct]

Parametro: Descrizione	
Formato	[A-Za-z.]
Lunghezza [L]	3. <4 [error] 4. >4 [correct]
Correttezza [C]	3. Non rispetta il formato [error] 4. Rispetta il formato [correct]

Parametro: Tipo	
Formato	Selezione da menu a tendina
Correttezza [C]	5. Non rispetta il formato [error] 6. Rispetta il formato [correct]

Parametro: Quantità	
Formato	[1-999]
Lunghezza [L]	7. <1 [error] 8. >1 [correct]
Correttezza [C]	7. Non rispetta il formato [error] 8. Rispetta il formato [correct]

Parametro: Stagionalità	
Formato	Selezione da menu a tendina
Lunghezza [L]	9. <4 [error] 10. >4 [correct]
Correttezza [C]	9. Non rispetta il formato [error] 10. Rispetta il formato [correct]



Parametro: Immagine	
Formato	<code>^https://[a-zA-Z0-9]*(\\.it/\\.com/\\.city/\\.org/\\.net)/([a-zA-Z0-9\\_ - ]/(\\.png/\\.jpg))*\$</code>
Correttezza [C]	11. Non rispetta il formato [error] 12. Rispetta il formato [correct]

Parametro: Prezzo	
Formato	[1-99,999]
Lunghezza [L]	13. <1 [error] 14. >1 [correct]
Correttezza [C]	13. Non rispetta il formato [error] 14. Rispetta il formato [correct]

Codice	Combinazione	Esito
TC2.1.1	L[1] C[1], L[3] C[3], L[5] C[5], L[7] C[7], L[9] C[9], L[11] C[11], L[13] C[13]	ERROR
TC2.1.2	L[2]&C[2], L[4]&C[4], L[6]&C[6], L[8]&C[8], L[10]&C[10], L[12]&C[12], L[14]&C[14]	CORRECT

### 9.2.2 Acquisto

Parametro: Indirizzo	
Formato	[A-Za-z0-9.]
Lunghezza [L]	1. <3 [error] 2. >3 [correct]
Correttezza [C]	1. Non rispetta il formato [error] 2. Rispetta il formato [correct]

Parametro: Città	
Formato	[A-Za-z.]
Lunghezza [L]	3. <3 [error] 4. >3 [correct]
Correttezza [C]	3. Non rispetta il formato [error] 4. Rispetta il formato [correct]

Parametro: Provincia	
Formato	[A-Za-z.]
Lunghezza [L]	5. <2 [error] 6. >2 [correct]
Correttezza [C]	5. Non rispetta il formato [error] 6. Rispetta il formato [correct]

Parametro: CAP	
Formato	[0-9.]
Lunghezza [L]	7. <5 [error] 8. =5 [correct]
Correttezza [C]	7. Non rispetta il formato [error] 8. Rispetta il formato [correct]

Parametro: Telefono	
Formato	[+0-9] +

<b>Lunghezza [L]</b>	9. <10 [error] 10. =10 [correct]
<b>Correttezza [C]</b>	9. Non rispetta il formato [error] 10. Rispetta il formato [correct]

<b>Parametro: Tipo Carta</b>	
<b>Formato</b>	Selezione da menu a tendina
<b>Correttezza [C]</b>	11. Non rispetta il formato [error] 12. Rispetta il formato [correct]

<b>Parametro: Numero Carta</b>	
<b>Formato</b>	^(?:4[0-9]{12}(?:[0-9]{3})? # Visa (?:5[1-5][0-9]{2} # MasterCard
<b>Lunghezza [L]</b>	13. <16 [error] 14. =16 [correct]
<b>Correttezza [C]</b>	13. Non rispetta il formato [error] 14. Rispetta il formato [correct]

<b>Parametro: Mese Scadenza</b>	
<b>Formato</b>	[0-9].
<b>Lunghezza [L]</b>	15. <2 [error] 16. =2 [correct]
<b>Correttezza [C]</b>	15. Non rispetta il formato [error] 16. Rispetta il formato [correct]

<b>Parametro: Anno Scadenza</b>	
<b>Formato</b>	[0-9]+
<b>Lunghezza [L]</b>	17. <4 [error] 18. =4 [correct]
<b>Correttezza [C]</b>	17. Non rispetta il formato [error] 18. Rispetta il formato [correct]

<b>Parametro: CVV</b>	
<b>Formato</b>	[0-9]+
<b>Lunghezza [L]</b>	19. <3 [error] 20. =3 [correct]

<b>Correttezza [C]</b>	19. Non rispetta il formato [error] 20. Rispetta il formato [correct]
------------------------	--

<b>Codice</b>	<b>Combinazione</b>	<b>Esito</b>
TC2.2.1	L[1]   C[1], L[3]   C[3], L[5]   C[5], L[7]   C[7], L[9]   C[9], L[11]   C[11], L[13]   C[13], L[15]   C[15], L[17] C[17], L[19]   C[19]	ERROR
TC2.2.2	L[2]&C[2], L[4]&C[4], L[6]&C[6], L[8]&C[8], L[10]&C[10], L[12]&C[12], L[14]&C[14], L[16]&C[16], L[18]&C[18], L[20]&C[20]	CORRECT

### 9.2.3 Ricerca

Parametro: casella di testo	
<b>Formato</b>	[A-Za-z.]
<b>Lunghezza [L]</b>	1. <2 [error] 2. >2 [correct]
<b>Correttezza [C]</b>	1. Non rispetta il formato [error] 2. Rispetta il formato [correct]

Codice	Combinazione	Esito
TC2.3.1	L[1], C[1]	ERROR
TC2.3.2	L[2], C[1]	ERROR
TC2.3.3	L[2], C[2]	CORRECT

## 10.Programma dei test

Il programma necessario per le attività di testing consiste nell'apprendimento di diversi software da utilizzare come JUnit, Selenium ecc.

Trattandosi di un progetto universitario non esiste badget, ma il programma fornisce un arricchimento della formazione e delle conoscenze acquisite dai componenti del gruppo.

Il principale rischio è legato al tempo necessario per imparare ad utilizzare i tools, che può rallentare il lavoro effettivo dedicato al progetto, con conseguente aumento del ritardo per la consegna del progetto.