



13/01/2019

# Test Plan

Team TopHW



Partecipanti

Nome	Cognome	Matricola
<b>Alfonso</b>	Rianna	0512104922
<b>Davide</b>	Zambelli	0512104688
<b>Raffaele</b>	Dragone	0512105016

Scritto da

Nome	Cognome	Matricola
<b>Alfonso</b>	Rianna	0512104922
<b>Davide</b>	Zambelli	0512104688
<b>Raffaele</b>	Dragone	0512105016

## Sommario

<b>1.</b>	<b>INTRODUZIONE .....</b>	<b>5</b>
<b>2.</b>	<b>Relazione con gli altri documenti .....</b>	<b>5</b>
2.1	Relazione con il documento di analisi dei Requisiti .....	5
2.2	Relazione con il documento del System Design.....	5
2.3	Relazione con il documento di Object Design .....	5
<b>3.</b>	<b>Panoramica del Sistema .....</b>	<b>5</b>
<b>4.</b>	<b>Funzionalità da Testare e da non Testare .....</b>	<b>6</b>
4.1	Funzionalità da Testare .....	6
4.2	Funzionalità da non Testare .....	6
<b>5.</b>	<b>Criteri di successo e di insuccesso .....</b>	<b>7</b>
5.1	Criteri di Successo.....	7
5.2	Criteri di Insuccesso.....	7
<b>6.</b>	<b>Approccio .....</b>	<b>7</b>
6.1	Test di unità.....	8
6.2	Test di sistema.....	8
6.3	Test di integrazione .....	8
<b>7.</b>	<b>Sospensione e ripristino dei testing.....</b>	<b>9</b>
7.1	Criteri di sospensione .....	9
7.2	Criteri di ripresa .....	9
<b>8.</b>	<b>Risorse utilizzate per il testing .....</b>	<b>9</b>
<b>9.</b>	<b>Test Case .....</b>	<b>9</b>
9.1	Gestione Autenticazione .....	10
9.1.1	Login .....	10
9.1.1.1	Category Partition.....	10
9.2	Gestione Utente .....	11
9.2.1	Registrazione .....	11
9.2.2	Category Partition .....	11
9.2.3	Test Cases .....	12
9.3	Gestione Catalogo .....	13
9.3.1	Inserimento Prodotto .....	13
9.3.1.2	Category Partition.....	13
9.3.1.3	Test Cases .....	14
9.3.2	Modifica Prodotto .....	15
9.3.2.1	Category Partition.....	15
9.3.1.3	Test Cases .....	15
9.4	Gestione Acquisto .....	17
9.4.1	Ricerca Prodotto .....	17
9.4.1.1	Category Partition.....	17

9.4.1.2 Test Cases .....	17
9.4.2 Inserimento Carta di credito .....	<b>Errore. Il segnalibro non è definito.</b>
9.4.2.1 Category Partition .....	<b>Errore. Il segnalibro non è definito.</b>
9.4.2.2 Test Cases .....	<b>Errore. Il segnalibro non è definito.</b>
9.4.3 Inserimento Indirizzo .....	18
9.4.3.1 Category Partition .....	18
9.3.3 Test Cases .....	19
10. Pianificazione del testing .....	22
10.1 Determinazione dei ruoli .....	22
10.2 Determinazione dei rischi .....	23
10.3 Decomposizione gerarchica del sistema .....	23
10.4 Organizzazione delle attività di testing .....	24

## 1. INTRODUZIONE

Questo documento si basa sulla pianificazione delle attività di testing, delle tecniche utilizzate nel testing, del sito di e-commerce TopHw. In questo senso, il documento mira a valutare l'aspetto qualitativo del software sviluppato, osservando il comportamento del sistema in esecuzione sulla base di particolari dati d'ingresso, quindi verificare se esistono differenze tra risultati attesi e risultati ottenuti. L'attività di testing mira soprattutto ad individuare particolari errori o bug presenti all'interno del codice, in modo tale che essi non si presentino quando il sistema sarà utilizzato dall'utente finale.

## 2. Relazione con gli altri documenti

### 2.1 Relazione con il documento di analisi dei Requisiti

I test dovranno tenere conto delle specifiche espresse nel R.A.D.

In particolar modo si presta molta attenzione a ciò che è stato espresso nei requisiti funzionali e non funzionali e dei casi d'uso descritti in base agli scenari rappresentati.

### 2.2 Relazione con il documento del System Design

Il testing dovrà garantire la coerenza tra il software e gli obiettivi di design definiti in fase di System Design, specificati nel SDD.

### 2.3 Relazione con il documento di Object Design

La fase di testing dovrà considerare il contenuto del documento di Object Design, in quanto quest'ultimo rappresenta la base per la realizzazione dell'implementazione, fondamentale per il testing. Pertanto, sarà necessario per individuare le differenze tra ciò che è stato stabilito in fase di Object Design ed il sistema effettivo.

## 3. Panoramica del Sistema

Come stabilito nel System Design Document la struttura del nostro sistema è divisa secondo un'architettura Model-View-Control. Il sistema inoltre è stato suddiviso in sottosistemi più piccoli, in particolare è stato diviso per gestioni. Abbiamo infatti i seguenti sottosistemi:

1. Gestione Autenticazione;

2. Gestione Registrazione;
3. Gestione Catalogo;
4. Gestione Acquisto;

Quasi ognuna delle precedenti gestioni prevede principalmente operazioni di inserimento, modifica, cancellazione, visualizzazione e ricerca e saranno proprio queste funzionalità ad essere testate nel corso della fase di testing del sistema.

## 4. Funzionalità da Testare e da non Testare

### 4.1 Funzionalità da Testare

Di seguito saranno elencate per ogni gestione quali sono le funzionalità che saranno testate.

#### Gestione Utenti:

- Registrazione;

#### Gestione Autenticazione:

- Login;

#### Gestione Catalogo:

- Inserimento prodotto;
- Modifica;

#### Gestione Acquisto:

- Ricerca
- Inserimento Indirizzo

### 4.2 Funzionalità da non Testare

Le funzionalità che non saranno testate sono:

- Funzionalità appartenenti a componenti prettamente grafiche che non offrono funzionalità di spicco per il corretto funzionamento del sistema, cioè la view e le jsp.

- Funzionalità appartenenti a componenti molto semplici, il cui testing è implicito nelle componenti da testare come i metodi get e set dei bean.

## 5. Criteri di successo e di insuccesso

### 5.1 Criteri di Successo

Un input avrà superato un test se l'output risultante sarà quello atteso, cioè quello che è stato specificato sul test case corrispondente.

I criteri di successo del sistema software previsti per la fase di testing sono i seguenti:

- Uguaglianza dal comportamento atteso e previsto dall'oracolo e comportamento effettivo della componente o dell'insieme di componenti.
- Mancanza di errori dovuti a ragioni tecniche.

### 5.2 Criteri di Insuccesso

I criteri di insuccesso del sistema software previsti per la fase di testing sono i seguenti:

- Differenza, anche minima, tra comportamento atteso e previsto dall'oracolo e comportamento effettivo della componente o dell'insieme di componenti.
- Presenza di errori dovuti a ragioni tecniche.
- Presenza di eccezioni generate a causa di una cattiva scrittura del codice.

## 6. Approccio

Le tecniche di testing adottate riguarderanno inizialmente il testing di unità dei singoli componenti, in modo da testare nello specifico la correttezza di ciascuna unità. Sarà applicata la tecnica del category partition che partiziona l'input da testare.

Seguirà il testing d'integrazione, che focalizzerà l'attenzione principalmente sul test delle interfacce delle suddette unità.

Infine verrà eseguito il testing di sistema, che vedrà come oggetto di testing l'intero sistema assemblato nei suoi componenti. Quest'ultimo servirà soprattutto a verificare che il sistema soddisfi le richieste del committente.

## 6.1 Test di unità

Durante questa fase, verranno ricercate le condizioni di fallimento isolando i componenti ed usando test driver e stub, cioè implementazioni parziali di componenti che dipendono o da cui dipendono le componenti da testare.

La strategia utilizzata per il testing si baserà esclusivamente sulla tecnica Black-Box, che si focalizza sul comportamento Input/Output, ignorando la struttura interna della componente.

Al fine di minimizzare il numero di test cases, i possibili input verranno partizionati in classi di equivalenza e per ogni classe verrà selezionato un test case.

Gli stati erronei scovati in questa, come in qualsiasi altra fase di testing, che comporteranno un fallimento del sistema dovranno essere tempestivamente comunicati agli sviluppatori al fine di correggerli e ripristinare il testing al più presto, per comunicare i vari fallimenti identificati durante la fase di testing si utilizzerà un test incident report.

## 6.2 Test di integrazione

In questa fase si procederà all'integrazione delle componenti di una funzionalità che verranno testate nel complesso attraverso una strategia Bottom-Up, poiché le componenti sono state implementate in modo specifico e poi connesse tra di loro in modo da creare una componente unica.

Si passerà, poi, alla funzionalità successiva fino ad esaurire le funzionalità implementate. Quest'approccio mira principalmente a ridurre le dipendenze tra funzionalità differenti e a facilitare la ricerca di errori nelle interfacce di comunicazione tra sottosistemi.

## 6.3 Test di sistema

Lo scopo di questa fase di testing è quello di dimostrare che il sistema soddisfi effettivamente i requisiti richiesti e sia, quindi, pronto all'uso. Come per il testing di unità, si cercherà di testare le funzionalità più importanti per l'utente e quelle che hanno una maggiore probabilità di fallimento. Si noti che, come per il testing di unità, si procederà attraverso tecnica Black-Box



## 7. Sospensione e ripristino dei testing

### 7.1 Criteri di sospensione

La fase di testing del sistema verrà sospesa quando si raggiungerà un compromesso tra qualità del prodotto e costi dell'attività di testing. Il testing verrà quindi portato avanti quanto più possibile nel tempo senza però rischiare di ritardare la consegna finale del progetto.

### 7.2 Criteri di ripresa

In seguito ad ogni modifica o correzione delle componenti che genereranno errori o fallimenti, i test case verranno sottoposti nuovamente al sistema assicurandosi così di aver risolto effettivamente il problema.

## 8. Risorse utilizzate per il testing

**Software:** un browser firefox, selenium IDE come estensione su firefox, office word per stilare un verbale sulle attività di testing svolte.

**Hardware:** due pc (Desktop o Laptop), di cui uno per effettuare test, l'altro per stilare il documento.

## 9. Test Case

## 9.1 Gestione Autenticazione

### 9.1.1 Login

#### 9.1.1.1 Category Partition

<b>Parametro: E-mail</b> <b>Formato: [A-Za-z0-9._]@[a-z.]</b>	
<b>Lunghezza[LE]</b>	1. < 5 and > 50 [error] 2. >=5 and <= 50 [propertyLEOK]
<b>Formato[FE]</b>	1. rispetta il formato [ifLEOK] [propertyFUOK,rispecchia il formato[A-Za- z0-9._]@[a-z.]] 2. non rispetta il formato [ifLEOK] [error]
<b>Presenza[PE]</b>	1. email presente nel database [PEOK] 2. email non presente nel database [error]

<b>Parametro: Password</b> <b>Formato: [A-Za-z0-9._]</b>	
<b>Lunghezza[LP]</b>	1. < 8 and > 32 [error] 2. >= 8 and <= 32 [propertylunghezzaLPOK]
<b>Formato[FP]</b>	1. Rispetta il formato[iflunghezzaLPOK][propertyformatoFPOK, rispecchia il formato [A-Za-z0-9.]] 2. Non rispetta il formato[iflunghezzaLPOK] [error]

Codice	Combinazione	Esito
TC_1.1_1	LE1	error
TC_1.1_2	LE2,FE2	error
TC_1.1_3	LE2, FE1, PE2	error
TC_1.1_4	LE2, FE1 , PE1, LP1	error
TC_1.1_5	LE2, FE1, PE1, LP2, FP2	error
TC_1.1_6	LE2, FE1, PE1, LP2, FP1	correct

## 9.2 Gestione Utente

### 9.2.1 Registrazione

#### 9.2.2 Category Partition

<b>Parametro: Nome</b> <b>Formato: [A-Za-z]</b>	
<b>Lunghezza[LN]</b>	1. < 2 and > 50 [error] 2. >=2 and <= 50 [property lunghezzaLNOK]
<b>Formato[FN]</b>	1. rispetta il formato [ifLNOK] [property formatoFNOK, rispecchia il formato [A-Za-z] 2. non rispetta il formato [ifLNOK] [error]

<b>Parametro: Cognome</b> <b>Formato: [A-Za-z]</b>	
<b>Lunghezza[LC]</b>	1. < 2 and > 50 [error] 2. >= 2 and <= 50 [property LCOK]
<b>Formato[FC]</b>	1. Rispetta il formato [ifLCOK] [property formatoFCOK, rispetta il formato [A-Za-z]] 2. Non rispetta il formato [ifLCOK] [error]

<b>Parametro: E-mail</b> <b>Formato: [A-Za-z0-9._]@[a-z.]</b>	
<b>Lunghezza[LE]</b>	3. < 5 and > 50 [error] 4. >=5 and <= 50 [property LEOK]
<b>Formato[FE]</b>	3. rispetta il formato [ifLEOK] [property FUOK, rispecchia il formato [A-Za-z0-9._]@[a-z.]] 4. non rispetta il formato [ifLEOK] [error]

<b>Presenza[PE]</b>	3. email non presente nel database [PEOK] 4. email già presente nel database [error]
---------------------	---

<b>Parametro: Password</b> <b>Formato: [A-Za-z0-9._]</b>	
<b>Lunghezza[LP]</b>	1. < 8 and > 32 [error] 2. >= 8 and <= 32 [property lunghezzaLPOK]
<b>Formato[FP]</b>	1. Rispetta il formato[iflunghezzaLPOK][property formatoFPOK, rispecchia il formato [A-Za-z0-9._]] 2. Non rispetta il formato[iflunghezzaLPOK] [error]

### 9.2.3 Test Cases

Codice	Combinazione	Esito
TC_3.1_1	LN1	error
TC_3.1_2	LN2, FN2	error
TC_3.1_3	LN2, FN1, LC1	error
TC_3.1_4	LN2, FN1, LC2, FC2	error
TC_3.1_5	LN2, FN1, LC2, FC1, LE1	error
TC_3.1_6	LN2, FN1, LC2, FC1, LE2, FE2	error
TC_3.1_7	LN2, FN1, LC2, FC1, LE2, FE1, PE2	error
TC_3.1_8	LN2, FN1, LC2, FC1, LE2, FE1, PE1, LP1	error
TC_3.1_9	LN2, FN1, LC2, FC1, LE2, FE1, PE1, LP1,FP2	error
TC_3.1_10	LN2, FN1, LC2, FC1, LE2, FE1, PE1, LP2,FP1	correct

## 9.3 Gestione Catalogo

### 9.3.1 Inserimento Prodotto

#### 9.3.1.2 Category Partition

<b>Parametro: Nome</b> <b>Formato: [A-Za-z]</b>	
<b>Lunghezza[LN]</b>	1. < 2 and > 50 [error] 2. >=2 and <= 50 [property lunghezzaLNOK]
<b>Formato[FN]</b>	1. rispetta il formato [ifLNOK] [property formatoFNOK, rispetta il formato [A-Za-z]] 2. non rispetta il formato [ifLNOK] [error]
<b>Presenza[PN]</b>	1. prodotto non presente nel database [PNOK] 2. prodotto già presente nel database [error]

<b>Parametro: Descrizione</b> <b>Formato: ^[^\]*\$</b>	
<b>Lunghezza[LD]</b>	1. < 20 and >100 [error] 2. >=20 and <= 100 [property lunghezzaLDOK]
<b>Formato[FD]</b>	1. Rispetta il formato [iflunghezzaLDOK] [property formatoFDOK, rispetta il formato ^[^\]*\$] 2. Non rispetta il formato [iflunghezzaLDOK] [error]

<b>Parametro: Immagine</b> <b>Formato: [A-Za-z0-9]+\.\png\.\jpg</b>	
<b>Lunghezza[LI]</b>	1. < 5 and >50 [error] 2. >=5 and <= 50 [property lunghezzaLIOK]

<b>Formato[FI]</b>	1. Rispetta il formato[iflunghezzaLIOK] [propertyformatoFIOK, rispetta il formato [A-Za-z0-9]+\.\png 2. Non rispetta il formato [iflunghezzaLIOK] [error]
--------------------	---

### 9.3.1.3 Test Cases

Codice	Combinazione	Esito
TC_4.2_1	LN1	Error
TC_4.2_2	LN2, FN2	Error
TC_4.2_3	LN2, FN1, PN2	Error
TC_4.2_4	LN2, FN2, PN1,LD1	Error
TC_4.2_5	LN2, FN1, PN1, LD2,FD2	Error
TC_4.2_6	LN2, FN1, PN1, LD2,FD1,LI1	Error
TC_4.2_7	LN2, FN1, PN1, LD2,FD1,LI2,FI2	Error
TC_4.2_8	LN2, FN1, PN1, LD2,FD1,LI2,FI1	Error
TC_4.2_9	LN2, FN1, PN1, LD2,FD1,LI2,FI1	Correct

### 9.3.2 Modifica Prodotto

#### 9.3.2.1 Category Partition

<b>Parametro: Nome</b> <b>Formato: [A-Za-z]</b>	
<b>Lunghezza[LN]</b>	1. < 2 and > 50 [error] 2. >=2 and <= 50 [property lunghezzaLNOK]
<b>Formato[FN]</b>	1. rispetta il formato [ifLNOK] [property formatoFNOK, rispecchia il formato [A-Za-z] 2. non rispetta il formato [ifLNOK] [error]
<b>Presenza[PN]</b>	

<b>Parametro: Descrizione</b> <b>Formato: ^[^\]*\$</b>	
<b>Lunghezza[LD]</b>	1. < 20 and >100 [error] 2. >=20 and <= 100 [property lunghezzaLDOK]
<b>Formato[FD]</b>	1. Rispetta il formato [iflunghezzaLDOK] [property formatoFDOK, rispetta il formato ^[^\]*\$ 2. Non rispetta il formato [iflunghezzaLDOK] [error]

#### 9.3.1.3 Test Cases

Codice	Combinazione	Esito
TC_5.2_1	LN1	Error
TC_5.2_2	LN2, FN2	Error
TC_5.2_3	LN2, FN1, LD1	Error
TC_5.2_4	LN2, FN1, LD2, FD2	Error

TC_5.2_5	LN2, FN1, LD2,FD1	Correct
----------	-------------------	---------



## 9.4 Gestione Acquisto

### 9.4.1 Ricerca Prodotto

#### 9.4.1.1 Category Partition

<b>Parametro:</b> Nome	
<b>Formato:</b> [a-zA-Z0-9]	
<b>Lunghezza[LN]</b>	
<b>Formato[FN]</b>	1. Rispetta il formato[iflunghezzaLNOK] [propertyformatoFNOK, rispetta il formato [a-zA-Z0-9]] 2. Non rispetta il formato [iflunghezzaLNOK] [error]

#### 9.4.1.2 Test Cases

Codice	Combinazione	Esito
TC_2.1_1	FN2	Error
TC_2.1_2	FN1	Correct

## 9.4.2 Inserimento Indirizzo

### 9.4.2.1 Category Partition

<b>Parametro: Via</b> <b>Formato: [a-zA-Z]</b>	
<b>Lunghezza[LV]</b>	1. < 3 and > 50 [error] 2. >=3 and <= 50 [property lunghezzaLVOK]
<b>Formato[FV]</b>	3. Rispetta il formato[iflunghezzaLVOK] [propertyformatoFVOK, rispetta il formato [a-zA-Z]] 4. Non rispetta il formato [iflunghezzaLVOK] [error]

<b>Parametro: Civico</b> <b>Formato: [0-9]</b>	
<b>Lunghezza[LCI]</b>	1. < 1 and > 5 [error] 2. >=1 and <= 5 [property lunghezzaLCIOK]
<b>Formato[FCI]</b>	3. Rispetta il formato[iflunghezzaLCIOK] [propertyformatoFCIOK, rispetta il formato [a-zA-Z]] 4. Non rispetta il formato [iflunghezzaLCIOK] [error]

<b>Parametro: Città</b> <b>Formato: [a-zA-Z]</b>	
<b>Lunghezza[LC]</b>	1. < 3 and > 50 [error] 2. >=3 and <= 50 [property lunghezzaLCOK]
<b>Formato[FC]</b>	3. Rispetta il formato[iflunghezzaLCOK] [propertyformatoFCOK, rispetta il formato [a-zA-Z]] 4. Non rispetta il formato [iflunghezzaLCOK] [error]

<b>Parametro: Provincia</b> <b>Formato: [a-zA-Z]</b>	
<b>Lunghezza[LP]</b>	1. < 3 and > 50 [error] 2. >=3 and <= 50 [property lunghezzaLPOK]
<b>Formato[FP]</b>	3. Rispetta il formato[iflunghezzaLPOK] [propertyformatoFPOK, rispetta il formato [a-zA-Z]] 4. Non rispetta il formato [iflunghezzaLPOK] [error]

<b>Parametro: Nazione</b> <b>Formato: [a-zA-Z]</b>	
<b>Lunghezza[LN]</b>	1. < 3 and > 50 [error] 2. >=3 and <= 50 [property lunghezzaLNOK]
<b>Formato[FN]</b>	3. Rispetta il formato[iflunghezzaLNOK] [propertyformatoFNOK, rispetta il formato [a-zA-Z]] 4. Non rispetta il formato [iflunghezzaLNOK] [error]

<b>Parametro: CAP</b> <b>Formato: [a-zA-Z]</b>	
<b>Lunghezza[LCA]</b>	1. < 3 and > 50 [error] 2. >=3 and <= 50 [property lunghezzaLCAOK]
<b>Formato[FCA]</b>	3. Rispetta il formato[iflunghezzaLCAOK] [propertyformatoFCAOK, rispetta il formato [a-zA-Z]] 4. Non rispetta il formato [iflunghezzaLCAOK] [error]

### 9.3.3 Test Cases

Codice	Combinazione	Esito
TC_7.2_1	LV2	Error
TC_7.2_2	LV2 , FV3	Error
TC_7.2_3	LV2 , FV3 , LCI2	Error
TC_7.2_4	LV2 , FV3 , LCI2, FCI3	Error
TC_7.2_5	LV2 , FV3 , LCI2, FCI3, LC2	Error
TC_7.2_6	LV2 , FV3 , LCI2, FCI3, LC2 , FC3	Error
TC_7.2_7	LV2 , FV3 , LCI2, FCI3, LC2 , FC3, LP2	Error

TC_7.2_8	LV2 , FV3 , LCI2, FCI3, LC2 , FC3, LP2, FP3	Error
TC_7.2_9	LV2 , FV3 , LCI2, FCI3, LC2 , FC3, LP2, FP3, LN2	Error
TC_7.2_10	LV2 , FV3 , LCI2, FCI3, LC2 , FC3, LP2, FP3, LN2, FN3	Error
TC_7.2_11	LV2 , FV3 , LCI2, FCI3, LC2 , FC3, LP2, FP3, LN2, FN3 , LCA2	Error
TC_7.2_12	LV2 , FV3 , LCI2, FCI3, LC2 , FC3, LP2, FP3, LN2, FN3 , LCA2, FCA3	Error
TC_7.2_13	LV2 , FV3 , LCI2, FCI3, LC2 , FC3, LP2, FP3, LN2, FN3 , LCA2, FCA3	Correct

## 10. Pianificazione del testing

Il team per il testing deve essere composto da persone che hanno una completa e approfondita conoscenza del sistema e delle tecniche di testing con i documenti associati, quali Test plan e Test case specification. Tali tecniche devono essere applicate nei tempi, nel budget e nei vincoli di qualità stabiliti. Solitamente i componenti del team di testing non sono coinvolti nella fase di sviluppo, in quanto non deve esistere una reticenza nel decomporre il sistema. Il team dedicato al controllo della qualità è responsabile dell'attività di testing e quindi della ricerca di fault. La documentazione dei fault trovati è inviata agli sviluppatori per consentire la correzione del sistema. Il sistema revisionato è poi testato nuovamente non solo per verificare se gli errori trovati in precedenza sono stati eliminati ma soprattutto per verificare che non ne siano stati introdotti dei nuovi. L'attività di testing è fondamentale nello sviluppo di un sistema software in quanto la mancanza di tale attività o una cattiva interpretazione di essa può portare al completo fallimento del sistema. Data l'importanza del testing ne risulta fondamentale la schedulazione.

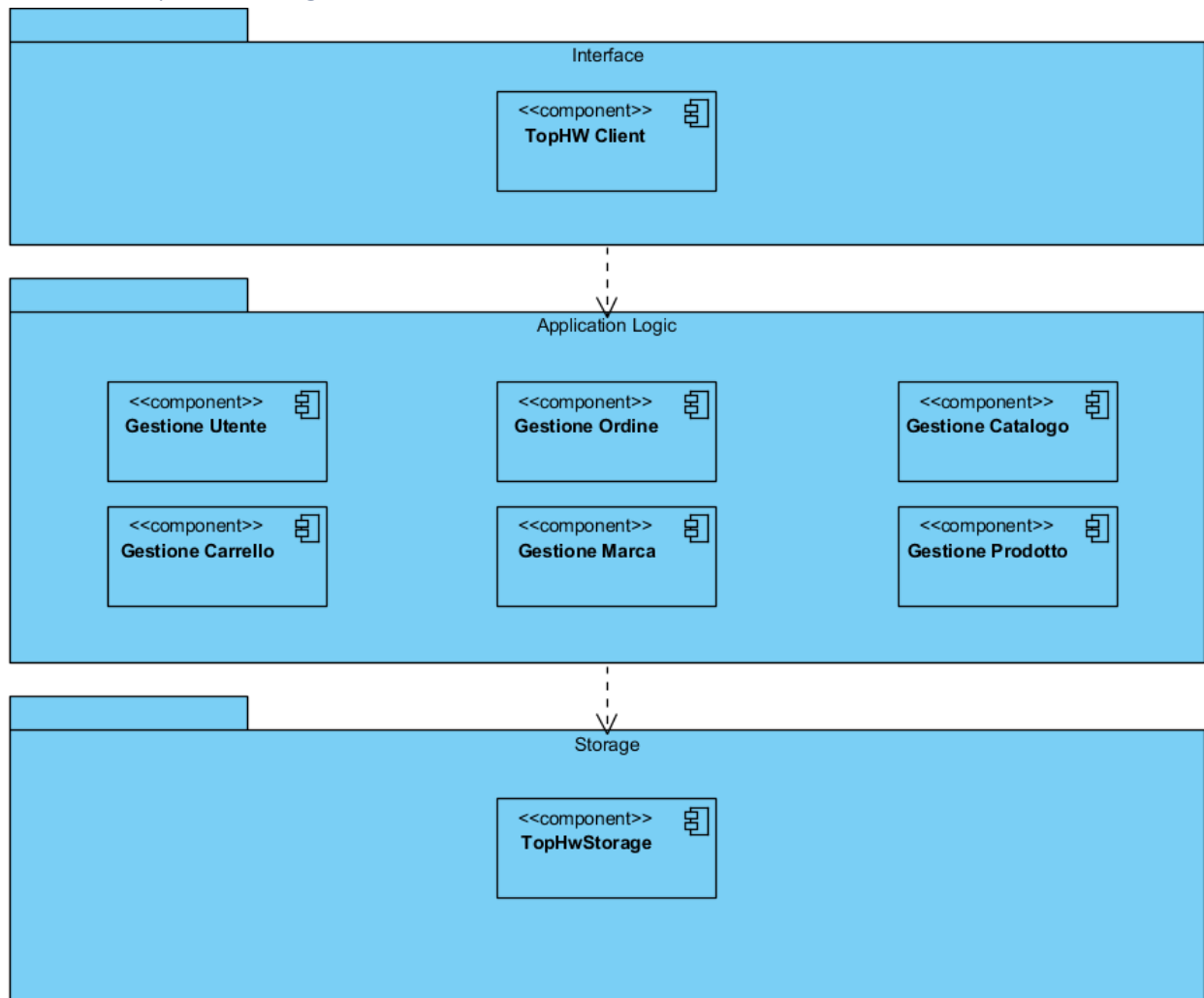
### 10.1 Determinazione dei ruoli

Il team del Testing sarà composto da Rianna Alfonso, Zambelli Davide e Dragone Raffaele. Le attività relative al testing di unità saranno delegate agli stessi sviluppatori delle componenti, in modo tale da alleggerire il lavoro del team di testing, che potrà quindi dedicarsi ad un più attento lavoro di testing funzionale.

### 10.2 Determinazione dei rischi

I rischi di un completo fallimento verranno minimizzati effettuando una pianificazione verticale delle attività di testing funzionale. Questo permetterà in caso di ritardi, dovuti ad una grande quantità di failure trovati, di rilasciare meno funzionalità del previsto, ma completamente testate.

### 10.3 Decomposizione gerarchica del sistema



### 10.4 Organizzazione delle attività di testing

Le attività di testing verranno organizzate secondo uno schema che effettuerà una divisione funzionale di tipo verticale. In questo modo al termine di ogni attività si avrà una funzionalità completamente testata nei suoi livelli gerarchici. I vantaggi principali sono che in caso di ritardi dovuti al ritrovamento di numerosi failure il sistema verrà rilasciato con meno componenti, ma interamente testate e funzionanti.