



# Test Plan Document AlcoList

Riferimento	
Versione	1.1
Data	
Destinatario	Prof. Gravino
Presentato da	Baldi Maria Rosaria Conte Melania Di Zenzo Carmine Federico Zaccardi Mario
Approvato da	



## Sommario

<i>Team Members</i> .....	3
<i>Revision History</i> .....	4
1 <i>Introduzione</i> .....	5
1.1 <i>Descrizione del documento</i> .....	5
2 <i>Relazione con altri Documenti</i> .....	5
3 <i>Panoramica del Sistema</i> .....	5
4 <i>Funzionalità da testare e no</i> .....	5
4.1 <i>Matrice di Tracciabilità del Testing</i> .....	6
5 <i>Criteri di Successo Fallimento</i> .....	6
6 <i>Approccio</i> .....	6
6.1 <i>Testing di Unità</i> .....	7
6.2 <i>Testing di Integrazione</i> .....	7
6.3 <i>Testing di Sistema</i> .....	7
7 <i>Strumenti per il testing (Hardware/Software)</i> .....	7
8 <i>Test Case</i> .....	8
9 <i>Specifiche dei Test Case</i> .....	9
9.1 <i>Aggiunta Dipendente;</i> .....	9
9.2 <i>Ricerca Cocktail</i> .....	13
9.3 <i>Aggiunta Cocktail</i> .....	18
9.4 <i>Invio Comanda</i> .....	21
10 <i>Pianificazione del Testing e Assegnazione dei ruoli</i> .....	22



## Team Members

NOME	ACRONIMO	INFORMAZIONI DI CONTATTO
Baldi Maria Rosaria	BMR	m.baldi24@studenti.unisa.it
Conte Melania	CM	m.conte49@studenti.unisa.it
Di Zenzo Carmine Federico	DCF	c.dizenzo2@studenti.unisa.it
Zaccardi Mario	ZM	m.zaccardi@studenti.unisa.it



## Revision History

DATA	VERSIONE	DESCRIZIONE	AUTORI
12/12/2022	0.1	Prima stesura	BMR, CM, DCF, ZM
13/12/2022	0.2	Invio comanda	ZM
14/12/2022	0.3	Aggiunta Cocktail D'Autore	CM
15/12/2022	0.4	Ricerca Cocktail	DCF
16/12/2022	0.5	Aggiungi Dipendente	BMR
17/12/2022	0.6	Prima Revisione	BMR, CM, DCF, ZM
05/01/2023	0.7	Modifica matrice tracciabilità	DCF, ZM
07/01/2023	0.8	Controllo Ricerca Cocktail	DCF
08/01/2023	0.9	Controllo Aggiunta Dipendente	BMR
09/01/2023	1.0	Controllo Cocktail D'Autore	CM
10/01/2023	1.1	Controllo Invio Comanda	ZM

# 1 Introduzione

## 1.1 Descrizione del documento

Il Test Plan del progetto AlcoList presenta la pianificazione e la specifica di quelli che sono i Test Case, ovvero l'insieme di input e di risultati attesi che servono a testare una componente del Sistema. Nel caso in cui delle attività di testing evidenziassero degli errori che possano causare comportamenti diversi da quelli attesi o che possano compromettere il buon utilizzo del Sistema da parte degli utenti, questo può essere sottoposto ad un processo di correzione degli errori individuati, per garantire agli utenti finali un prodotto che rispecchi tutte le specifiche stabilite nelle precedenti fasi di sviluppo.

## 2 Relazione con altri Documenti

Il Test Plan presenta diversi punti di correlazione con i documenti creati durante le fasi precedenti dello sviluppo di AlcoList. In particolare, il presente Documento fa riferimento ai casi d'uso descritti nel Requirement Analysis Document (RAD). Ulteriori dettagli sono specificati nel Test Case Specification Document (TCSD).

## 3 Panoramica del Sistema

Il Sistema AlcoList fornisce tutte le sue funzionalità attraverso una Web Application. Per assicurarsi che si comporti come previsto, bisogna accettarsi che vengano testate tutte le funzionalità offerte dal Sistema. Nel caso in cui si comportino nella maniera prevista, si può ritenere che il Sistema soddisfi gli obiettivi prefissati.

## 4 Funzionalità da testare e no

Le attività di testing per il Sistema AlcoList prevedono di verificare il corretto funzionamento delle funzionalità del Sistema. Quelle che saranno oggetto dell'attività di testing sono:

1. Aggiunta Dipendente;
2. Ricerca Cocktail;
3. Creazione Cocktail D'Autore;
4. Invio Comanda.

## 4.1 Matrice di Tracciabilità del Testing

REQUISITI	CASI DI TEST			
	TC_1	TC_2	TC_3	TC_4
RF_1				✓
RF_2				
RF_3				
RF_4		✓	✓	
RF_5	✓			
RF_6				
RF_7				
RF_8				
RF_9				
RF_10				
RF_11				

## 5 Criteri di Successo Fallimento

Un caso di test ha esito positivo se l'output osservato è differente dal risultato previsto dall'oracolo; al contrario, un caso di test ha esito negativo se l'output osservato coincide con il risultato previsto dall'oracolo. Pertanto, le attività di test hanno successo nei casi in cui riescono ad individuare dei comportamenti anomali nell'esecuzione delle funzionalità del Sistema. Nel caso in cui uno o più casi di test riscuotono successo, è possibile attuare un'opportuna procedura di correzione del difetto riscontrato e, successivamente, ricorrere ad un test di regressione per testare nuovamente la funzionalità modificata ed accertarsi che il problema sia stato risolto.

## 6 Approccio

L'attività di Testing del Sistema è stata organizzata in tre categorie principali:

- Testing di Unità, che si occupa di testare le singole componenti del Sistema;
- Testing di Integrazione, che si occupa di integrare le unità testate in modo isolato, così da controllare che il sistema funzioni correttamente quando più unità collaborano;
- Testing di Sistema, che include l'attività di Testing Funzionale, la quale si occupa di testare le funzionalità del Sistema definite dai requisiti funzionali. Il Testing di Sistema include anche il Testing di Performance e il Testing Pilota, che non verranno effettuate in quanto il budget a disposizione per il progetto non copre l'ammontare di risorse necessarie.

## 6.1 Testing di Unità

L'attività di Testing di Unità consiste nell'isolare un'unità del Sistema e provare che funzioni correttamente. La strategia prevista per affrontare tale attività consiste nel testare i metodi delle classi del Sistema e, in particolare, quelli che permettono le validazioni dell'input dell'utente, come descritto nei test cases. I casi di test saranno definiti attraverso un approccio Black-box attraverso l'uso del framework JUnit. Inoltre, per isolare le componenti testate è stato utilizzato Mockito.

## 6.2 Testing di Integrazione

L'attività di Testing di Integrazione consiste nell'integrare le componenti testate singolarmente in sottosistemi più grandi, così da rilevare i bug che non sono stati evidenziati con il Testing di Unità. Per questa attività è stata scelta la strategia di testing Bottom-Up, in quanto è ritenuta migliore per i sistemi software Object-Oriented, paradigma utilizzato per AlcoList. In particolare, sono stati testati tutti i metodi delle entità che richiamano i metodi dei repository JPA. Le tecnologie utilizzate sono le stesse di quelle del Testing di Unità, dunque JUnit e Mockito.

## 6.3 Testing di Sistema

L'attività di Testing Funzionale, inclusa in quella di Testing di Sistema, consiste nell'individuare i possibili fault generati dagli input degli utenti. L'approccio utilizzato per testare le funzionalità del Sistema, definite a partire dai requisiti funzionali e dai casi d'uso esposti nel documento "RAD", è quello del "Black-box Testing". La tecnica utilizzata per individuare i casi di test e per descriverli è la "Category Partition", tecnica per l'individuazione dei casi di test partendo da un requisito funzionale e dal caso d'uso collegato. L'attività di Testing di Sistema è invece effettuata tramite Selenium IDE e consiste nell'utilizzare tale tool per verificare che i test case descritti nel Testing Funzionale non rilevino errori.

## 7 Strumenti per il testing (Hardware/Software)

Lo strumento hardware utilizzato per l'attività di testing è il computer poiché il Sistema non è stato ancora rilasciato.

Come anticipato nei precedenti paragrafi, gli strumenti software utilizzati per le attività di testing sono:

- JUnit: framework open source usato per scrivere ed eseguire Testing di Unità per Java;
- Mockito: framework utilizzato per la realizzazione di oggetti mock durante il Testing di Unità;
- Selenium IDE: tool open source utilizzato per la gestione automatizzata dei browser, utilizzato come framework di testing. Esso permette di registrare le azioni che un utente può effettuare nell'interazione con il Sistema, così da poter eseguire i casi di test.



## 8 Test Case

Per sviluppare i Test Case sarà utilizzato il metodo del Category Partition. Questo metodo consiste nell'identificare per ogni funzionalità da testare dei parametri; per ogni parametro verranno individuate delle categorie, le quali poi saranno suddivise in scelte. Alle scelte verrà assegnato un valore. I Test Case verranno definiti nel documento di Test Cases Specification (TCS).



## 9 Specifiche dei Test Case

### 9.1 Aggiunta Dipendente;

#### TC\_1 Aggiunta Dipendente

**N.B.** La password non viene considerata un campo da controllare in quanto viene generata in automatico dal sistema.

<b>Parametro:</b> Nome Dipendente <b>Formato:</b> $^{\wedge}[\text{a-zA-Z} \cdot']^{+}\$$	
CATEGORIE	POSSIBILI SCENARI
Campo vuoto cvn	<ol style="list-style-type: none"> <li>1. Il campo non è stato compilato [errore]</li> <li>2. Il campo è stato compilato [property <b>campoVuotoCVNok</b>]</li> </ol>
Lunghezza ln	<ol style="list-style-type: none"> <li>1. Lunghezza <math>\leq 1</math> [if <b>campoVuotoCVNok</b>] [errore]</li> <li>2. Lunghezza <math>&gt; 30</math> [if <b>campoVuotoCVNok</b>] [errore]</li> <li>3. <math>0 &lt; \text{Lunghezza} \leq 30</math> [if <b>campoVuotoCVNok</b>] [property <b>lunghezzaLNok</b>]</li> </ol>
Formato fn	<ol style="list-style-type: none"> <li>1. Non rispetta il formato [if <b>campoVuotoCVNok</b> and <b>lunghezzaLNok</b>] [errore]</li> <li>2. Rispetta il formato [if <b>campoVuotoCVNok</b> and <b>lunghezzaLNok</b>] [property <b>formatoFNok</b>]</li> </ol>

<b>Parametro:</b> Cognome Dipendente <b>Formato:</b> $^{\wedge}[\text{a-zA-Z} \cdot']^{+}\$$	
CATEGORIE	POSSIBILI SCENARI
Campo vuoto cvc	<ol style="list-style-type: none"> <li>1. Il campo non è stato compilato [errore]</li> <li>2. Il campo è stato compilato [property <b>campoVuotoCVNok</b>]</li> </ol>
Lunghezza lc	<ol style="list-style-type: none"> <li>1. Lunghezza <math>\leq 1</math> [if <b>campoVuotoCVNok</b>] [errore]</li> <li>2. Lunghezza <math>&gt; 30</math> [if <b>campoVuotoCVNok</b>] [errore]</li> <li>3. <math>0 &lt; \text{Lunghezza} \leq 30</math> [if <b>campoVuotoCVNok</b>] [property <b>lunghezzaLNok</b>]</li> </ol>
Formato fc	<ol style="list-style-type: none"> <li>1. Non rispetta il formato [if <b>campoVuotoCVNok</b> and <b>lunghezzaLNok</b>] [errore]</li> <li>2. Rispetta il formato [if <b>campoVuotoCVNok</b> and <b>lunghezzaLNok</b>] [property <b>formatoFNok</b>]</li> </ol>

<b>Parametro:</b> email Dipendente <b>Formato:</b> $^{\wedge}[\backslash w-\backslash.]+\@([\backslash w-]+\backslash.)+[\backslash w-]\{2,4\}\$$	
CATEGORIE	POSSIBILI SCENARI
Campo vuoto cve	<ol style="list-style-type: none"> <li>1. Il campo non è stato compilato [errore]</li> <li>2. Il campo è stato compilato [property <b>campoVuotoCVNok</b>]</li> </ol>
Lunghezza le	<ol style="list-style-type: none"> <li>1. Lunghezza <math>\leq 8</math> [if <b>campoVuotoCVNok</b>] [errore]</li> <li>2. Lunghezza <math>&gt; 30</math> [if <b>campoVuotoCVNok</b>] [errore]</li> <li>3. <math>0 &lt; \text{Lunghezza} \leq 30</math> [if <b>campoVuotoCVNok</b>] [property <b>lunghezzaLNok</b>]</li> </ol>
Formato fe	<ol style="list-style-type: none"> <li>1. Non rispetta il formato [if <b>campoVuotoCVNok</b> and <b>lunghezzaLNok</b>] [errore]</li> <li>2. Rispetta il formato [if <b>campoVuotoCVNok</b> and <b>lunghezzaLNok</b>] [property <b>formatoFNok</b>]</li> </ol>

<b>Parametro:</b> Ruolo (Dropdown menù) <b>Formato:</b> /	
CATEGORIE	POSSIBILI SCENARI
Scelta sr	<ol style="list-style-type: none"> <li>1. Scelta non effettuata [error]</li> <li>2. Scelta effettuata [property <b>SceltaScOk</b>]</li> </ol>



CODICE	COMBINAZIONE	ESITO
TC_1.1_01	Cvn1	FAIL
TC_1.1_02	Cvn2.ln1	FAIL
TC_1.1_03	Cvn2.ln2	FAIL
TC_1.1_04	Cvn2.ln3.f1	FAIL
TC_1.1_05	Cvn2.ln3.f2.cvc1	FAIL
TC_1.1_06	Cvn2.ln3.f2.cvc2.lc1	FAIL
TC_1.1_07	Cvn2.ln3.f2. cvc2.lc2	FAIL
TC_1.1_08	Cvn2.ln3.f2.cvc2.lc3.fc1	FAIL
TC_1.1_13	Cvn2.ln3.f2.cvc2.lc3.fc2. cve1	FAIL
TC_1.1_14	Cvn2.ln3.f2.cvc2.lc3.fc2.cve2.le1	FAIL
TC_1.1_15	Cvn2.ln3.f2.cvc2.lc3.fc2.cve2.le2	FAIL
TC_1.1_16	Cvn2.ln3.f2.cvc2.lc3.fc2.cve2.le3.fe1	FAIL
TC_1.1_17	Cvn2.ln3.f2.cvc2.lc3.fc2.cve2.le3.fe2.sr1	FAIL
TC_1.1_18	Cvn2.ln3.f2.cvc2.lc3.fc2.cve2.le3.fe2.sr2	PASS

## 9.2 Ricerca Cocktail

### TC\_2 Ricerca Cocktail

Poiché la ricerca di un cocktail può essere effettuata specificando uno o più campi tra nome, gusto, alcolico (o analcolico) o compilando tutti i campi, il test case “TC\_2 Ricerca Cocktail” è l’aggregazione di quattro test case componenti:

- 1) TC\_2.1 RicercaSoloNome;
- 2) TC\_2.2 RicercaSoloGusto;
- 3) TC 2.3 RicercaSoloAlcolico
- 4) TC\_2.4 RicercaCompleta

Il motivo di tale suddivisione dei test case è che, nel caso in cui venga specificato solo uno dei campi della ricerca, non si tratterà di un errore.

Le ricerche RicercaNomeAlcolico, RicercaGustoAlcolico e RicercaNomeGusto non sono considerate in quanto rientrano nei test case precedentemente svolti.

Inoltre, nel caso in cui, tutti i campi fossero vuoti, la ricerca stamperà tutti i cocktail.

### TC\_2.1 RicercaSoloNome

<b>Parametro:</b> Nome <b>Formato:</b> $^{\wedge}[a-zA-Z \cdot ]^{+}\$$	
CATEGORIE	POSSIBILI SCENARI
Campo c	<ol style="list-style-type: none"> <li>1. Il campo è vuoto [<b>error</b>]</li> <li>2. Il campo è stato compilato [property <b>CampoCOk</b>]</li> </ol>
Formato fn	<ol style="list-style-type: none"> <li>1. Non rispetta il formato [if <b>CampoCOk</b>] [<b>errore</b>]</li> <li>2. Rispetta il formato [property <b>formatoFnOk</b>]</li> </ol>
Lunghezza len	<ol style="list-style-type: none"> <li>1. Lunghezza <math>\leq 2</math> [if <b>CampoCOk</b> and <b>FormatoFnOk</b>] [<b>errore</b>]</li> <li>2. Lunghezza <math>&gt; 30</math> [if <b>CampoCOk</b> and <b>FormatoFnOk</b>] [<b>errore</b>]</li> <li>3. <math>2 &lt; \text{Lunghezza} \leq 30</math> [if <b>CampoCOk</b> and <b>FormatoFnOk</b>] [property <b>lunghezzaLenOk</b>]</li> </ol>

CODICE	COMBINAZIONE	ESITO
TC_2.1_01	c2.fn1	FAIL
TC_2.1_02	c2.fn2.len1	FAIL
TC_2.1_03	c2.fn2.len2	FAIL
TC_2.1_04	c2.fn2.len3	PASS

### TC\_2.2 RicercaSoloGusto



<b>Parametro:</b> Gusto (Dropdown menù) <b>Formato:</b> /	
CATEGORIE	POSSIBILI SCENARI
Scelta sc	<ol style="list-style-type: none"><li>3. Scelta non effettuata [<b>error</b>]</li><li>4. Scelta effettuata [property <b>SceltaScOk</b>]</li></ol>

CODICE	COMBINAZIONE	ESITO
TC_2.2_01	sc2	PASS

#### TC\_2.3 RicercaSoloAlcol

<b>Parametro:</b> Tipologia (Dropdown menù) <b>Formato:</b> /	
CATEGORIE	POSSIBILI SCENARI
Scelta al	<ol style="list-style-type: none"><li>1. Scelta non effettuata [<b>error</b>]</li><li>2. Scelta effettuata [property <b>SceltaAlSingola</b>]</li></ol>

CODICE	COMBINAZIONE	ESITO
TC_2.3_01	al2	PASS

## TC\_2.4 RicercaCompleta

<b>Parametro:</b> Nome <b>Formato:</b> $^{\wedge}[a-zA-Z \cdot ]^{+}\$$	
CATEGORIE	POSSIBILI SCENARI
Campo c	<ol style="list-style-type: none"> <li>1. Il campo è vuoto [<b>error</b>]</li> <li>2. Il campo è stato compilato [property <b>CampoCOk</b>]</li> </ol>
Formato fn	<ol style="list-style-type: none"> <li>1. Non rispetta il formato [if <b>CampoCOk</b>] [errore]</li> <li>2. Rispetta il formato [property <b>formatoFnOk</b>]</li> </ol>
Lunghezza len	<ol style="list-style-type: none"> <li>1. Lunghezza <math>\leq 2</math> [if <b>CampoCOk</b> and <b>FormatoFnOK</b>] [errore]</li> <li>2. Lunghezza <math>&gt; 30</math> [if <b>CampoCOk</b> and <b>FormatoFnOK</b>] [errore]</li> <li>3. <math>2 &lt; \text{Lunghezza} \leq 30</math> [if <b>CampoCOk</b> and <b>FormatoFnOK</b>] [property <b>lunghezzaLenOk</b>]</li> </ol>

<b>Parametro:</b> Gusto (Dropdown menù) <b>Formato:</b> /	
CATEGORIE	POSSIBILI SCENARI
Scelta sc	<ol style="list-style-type: none"> <li>1. Scelta non effettuata [<b>error</b>]</li> <li>2. Scelta effettuata [property <b>SceltaScOk</b>]</li> </ol>



**Parametro:** Tipologia (Dropdown menù)  
**Formato:** /

CATEGORIE	POSSIBILI SCENARI
Scelta al	<ol style="list-style-type: none"> <li>Scelta non effettuata [<b>error</b>]</li> <li>Scelta effettuata [property <b>SceltaAlSingola</b>]</li> </ol>

CODICE	COMBINAZIONE	ESITO
TC_2.4_01	c2.fn1	FAIL
TC_2.4_02	c2.fn2.len1	FAIL
TC_2.4_03	c2.fn2.len2	FAIL
TC_2.4_04	c2.fn2.len3.sc1	FAIL
TC_2.4_05	c2.fn2.len3.sc2.al1	FAIL
TC_2.4_06	c2.fn2.len3.sc2.al2	PASS

## 9.3 Aggiunta Cocktail D'Autore

### TC\_3 Aggiunta Cocktail D'Autore

<b>Parametro:</b> Nome Cocktail <b>Formato:</b> $^{\wedge}[a-zA-Z.']+ \$$	
CATEGORIE	POSSIBILI SCENARI
Campo vuoto cvn	<ol style="list-style-type: none"> <li>1. Il campo non è stato compilato [errore]</li> <li>2. Il campo è stato compilato [property <b>campoVuotoCVNok</b>]</li> </ol>
Lunghezza ln	<ol style="list-style-type: none"> <li>1. Lunghezza <math>\leq 0</math> [if <b>campoVuotoCVNok</b>] [errore]</li> <li>2. Lunghezza <math>&gt; 30</math> [if <b>campoVuotoCVNok</b>] [errore]</li> <li>3. <math>0 &lt; \text{Lunghezza} \leq 30</math> [if <b>campoVuotoCVNok</b>] [property <b>lunghezzaLNok</b>]</li> </ol>
Formato fn	<ol style="list-style-type: none"> <li>1. Non rispetta il formato [if <b>campoVuotoCVNok</b> and <b>lunghezzaLNok</b>] [errore]</li> <li>2. Rispetta il formato [if <b>campoVuotoCVNok</b> and <b>lunghezzaLNok</b>] [property <b>formatoFNok</b>]</li> </ol>

<b>Parametro:</b> Path immagine <b>Formato:</b> $^{\wedge}[a-zA-Z.']+ \$$	
CATEGORIE	POSSIBILI SCENARI
Campo vuoto cvp	<ol style="list-style-type: none"> <li>1. Il campo non è stato compilato [errore]</li> <li>2. Il campo è stato compilato [property <b>campoVuotoCVPok</b>]</li> </ol>

<b>Parametro:</b> Descrizione <b>Formato:</b> $^{\wedge}[a-zA-Z.']+\$$	
CATEGORIE	POSSIBILI SCENARI
Campo vuoto cvd	<ol style="list-style-type: none"> <li>1. Il campo non è stato compilato [errore]</li> <li>2. Il campo è stato compilato [property <b>campoVuotoCVDok</b>]</li> </ol>

<b>Parametro:</b> Prezzo <b>Formato:</b> $^{\wedge}[1-9]+[0-9]*\$$	
CATEGORIE	POSSIBILI SCENARI
Campo vuoto cvpr	<ol style="list-style-type: none"> <li>1. Il campo non è stato compilato [errore]</li> <li>2. Il campo è stato compilato [property <b>campoVuotoCVDok</b>]</li> </ol>
Formato fp	<ol style="list-style-type: none"> <li>1. Non rispetta il formato [if <b>campoVuotoCVPRok</b>] [errore]</li> <li>2. Rispetta il formato [if <b>campoVuotoCVPRok</b>] [property <b>formatoFPok</b>]</li> </ol>

CODICE	COMBINAZIONE	ESITO
TC_3_01	cvn1	FAIL
TC_3_02	cvn2.ln1	FAIL
TC_3_03	cvn2.ln2	FAIL
TC_3_04	cvn2.ln3.fn1	FAIL
TC_3_05	cvn2.ln3.fn2	FAIL
TC_3_06	cvn2.ln3.fn3.cvp1	FAIL
TC_3_07	cvn2.ln3.fn3.cvp2.cvd1	FAIL
TC_3_08	cvn2.ln3.fn3.cvp2.cvd2.cvpr1	FAIL
TC_3_09	cvn2.ln3.fn3.cvp2.cvd2.cvpr2.fp1	FAIL
TC_3_09	cvn2.ln3.fn3.cvp2.cvd2.cvpr2.fp2	PASS

**N.B.** Step 2 di aggiunta cocktail

<b>Parametro:</b> Quantità <b>Formato:</b> $^{\wedge}[1-9]+[0-9]*\$$	
CATEGORIE	POSSIBILI SCENARI
Formato fq	<ol style="list-style-type: none"> <li>Non rispetta il formato [errore]</li> <li>Rispetta il formato [property <b>formatoFQok</b>]</li> </ol>
Quantità qn	<ol style="list-style-type: none"> <li>Una o più quantità superano quelle del corrispettivo prodotto  <math>q_1 &gt; qp_1 \vee q_2 &gt; qp_2 \vee \dots \vee q_n &gt; qp_n</math> [if <b>formatoFQok</b>] [errore] </li> <li>Tutte le quantità sono nulle  <math>q_1 \leq 0 \wedge q_2 \leq 0 \wedge \dots \wedge q_n \leq 0</math> [if <b>formatoFQok</b>] </li></ol> <p>OR</p> <p>I dati sono inseriti correttamente:  <math>\exists q_x \text{ in } q_1, \dots, q_n \mid q_x &gt; 0 \wedge q_x &lt; qp_x</math>            (dove <math>qp_x</math> è la quantità totale del corrispettivo prodotto) </p> <p>[property <b>quantityQNok</b>]</p>

CODICE	COMBINAZIONE	ESITO
TC_3_01	fq1	FAIL
TC_3_02	fq2.qn1	FAIL
TC_3_04	fq2.qn2	PASS

## 9.4 Invio Comanda

<b>Parametro:</b> Cocktail (Checkbox) <b>Formato:</b> /	
CATEGORIE	SCELTE
<b>Ceckbox ckd</b>	<ol style="list-style-type: none"> <li>1. Non selezionato [<b>errore</b>]</li> <li>2. Selezionato [property <b>CeckboxIsCkd</b>]</li> </ol>

<b>Parametro:</b> Quantità <b>Formato:</b> $^{\wedge}[1-9]+[0-9]*\$$	
CATEGORIE	SCELTE
<b>Campo cam</b>	<ol style="list-style-type: none"> <li>1. Il campo non è stato compilato [<b>errore</b>]</li> <li>2. Il campo è stato compilato correttamente [property <b>campoCamIsOk</b>]</li> </ol>
<b>Formato for</b>	<ol style="list-style-type: none"> <li>1. Formato errato [if <b>campoCamIsOk</b>] [<b>errore</b>]</li> <li>2. Rispetta il formato [if <b>campoCamIsOk</b>] [property <b>formatoForIsOk</b>]</li> </ol>
<b>Lunghezza len</b>	<ol style="list-style-type: none"> <li>1. Lunghezza <math>\leq 0</math> or Lunghezza <math>\geq</math> “Disponibilità degli ingredienti” [if <b>campoCamIsOk</b> and <b>formatoForIsOk</b>] [<b>errore</b>]</li> <li>2. Lunghezza <math>&gt; 0</math> and Lunghezza <math>&lt;</math> “Disponibilità degli ingredienti” [if <b>campoCamIsOk</b> and <b>formatoForIsOk</b>] [property <b>lunghezzaLenIsOk</b>]</li> </ol>

CODICE	COMBINAZIONE	ESITO
TC_4_01	Ckd1	FAIL
TC_4_03	Ckd2.Cam1	FAIL
TC_4_04	Ckd2.Cam2.For1	FAIL
TC_4_05	Ckd2.Cam2.For2.Len1	FAIL
TC_4_06	Ckd2.Cam2.For2.Len2	PASS

## 10 Pianificazione del Testing e Assegnazione dei ruoli

Per documentare l'organizzazione adottata, in merito alle attività di pianificazione del testing e di assegnazione dei ruoli, presentiamo la tabella sottostante che funge da matrice di tracciabilità per facilitare l'identificazione delle informazioni. In particolare, le attività di testing inizieranno con il Testing Funzionale, quindi con l'individuazione dei Test Cases, e a seguire saranno realizzate tutte le altre categorie di testing descritte.

Test Case	Autore
TC_01	Baldi Maria Rosaria
TC_02	Di Zenzo Carmine Federico
TC_03	Conte Melania
TC_04	Zaccardi Mario