

**GINusJuice
Test Plan
Versione 1.0**



Data: 14/12/2024

Coordinatore del progetto:

Nome	Matricola
D'Avino Salvatore	0512118435
Vitulano Antonio	0512116776

Partecipanti:

Nome	Matricola
ANTONIO VITULANO	0512116776
SALVATORE D'AVINO	0512118435

Scritto da:	ANTONIO VITULANO, SALVATORE D'AVINO
--------------------	-------------------------------------

Revision History

Data	Versione	Descrizione	Autore
14/12/2024	1.0	Introduzione	Vitulano
14/12/2024	1.0	Rapporto con altri documenti	Vitulano
14/12/2024	1.0	Panoramica del sistema	Vitulano
14/12/2024	1.0	Caratteristiche da testare e non da testare	Vitulano
14/12/2024	1.0	Criteri successo e di fail	Vitulano
14/12/2024	1.0	Approccio	D'avino
14/12/2024	1.0	Sospensione e di ripresa	D'avino
14/12/2024	1.0	Testing e materials	D'Avino
14/12/2024	1.0	Testing Cases	D'Avino
14/12/2024	1.0	Testing Schedule	D'Avino
15/12/2024	1.0	Riunione per approvare	D'avino Vitulano
18/12/2024	1.1	Casi di test Acquisto	Vitulano
18/12/2024	1.1	Casi di test login	Vitulano
18/12/2024	1.1	Casi di test registrazione	D'avino
18/12/2024	1.1	Casi di test chiedi al barista	D'Avino
18/12/2024	1.1	Casi di test Cerca prodotto	Vitulano

18/12/2024	1.1	Revisione	Vitulano D'Avino
------------	-----	-----------	---------------------

Sommario

INTRODUZIONE	4
Definizioni	4
1.1.2 Acronimi e Abbreviazioni	4
1.1.3 Riferimenti	4
Rapporto con altri documenti	4
Relazione con il documento di raccolta ed analisi dei requisiti (RAD)	4
Relazione con il System Design Document (SDD)	4
Relazione con l'Object Design Document (ODD)	5
Panoramica del sistema	5
Caratteristiche da testare/non da testare	5
Criteri di successo/fail	5
Approccio	5
Testing di unità	5
Approccio scelto	5
Testing di integrazione	5
Approccio scelto	6
Testing di sistema	6
Approccio scelto	6
Sospensione e ripresa.....	6
Testing materials	6
Test Cases	6
Testing schedule.....	7

INTRODUZIONE

Definizioni

- Branch Coverage: tecnica adoperata durante la fase di testing, che prevede l'esecuzione di tutti i rami del programma almeno una volta durante la fase di testing.
- Failure: mancata o scorretta azione di un determinato servizio atteso.
- Fault: causa che ha generato una failure.
- Model View Control: è un metodo architetturale che prevede la divisione dell'applicazione di tre parti. Tale divisione viene effettuata per separare la rappresentazione delle informazioni interne del sistema dal meccanismo in cui le informazioni sono presentate all'utente

1.1.2 Acronimi e Abbreviazioni

- RAD: Requirement Analysis Document.
- SDD: System Design Document.
- ODD: Object Design Document.
- TP: Test plan.
- MVC: Model View Controller.
- DB: Database.
- API: Application Programming Interface.
- GUI: Graphical User Interface.

1.1.3 Riferimenti

- Object-Oriented Software Engineering (Using UML, Patterns, and Java) Third Edition.
Autori: Bernd Bruegge & Allen H. Dutoit
- PMBOK Guide and Software Extention to the PMBOK Guide, Fifth Ed, Project Management Institute, 2013;
- Documentazione del progetto:
RAD,SDD,ODD

Rapporto con altri documenti

Questo documento è correlato a tutti i documenti prodotti fino al rilascio del sistema, quindi verranno modificati in futuro dopo il rilascio di altri documenti non ancora prodotti. I test case sono basati sulle funzionalità del sistema, individuate e raccolte nel RAD.e nei sottosistemi dell'SDD

Relazione con il documento di raccolta ed analisi dei requisiti (RAD)

La relazione tra questo documento e il RAD riguarda la fase di raccolta dei requisiti funzionali e non funzionali, infatti in questo documento oltre a essere presenti i requisiti si possono trovare anche scenari, casi d'uso, diagrammi di sequenza e mockups.

Relazione con il System Design Document (SDD)

In questo documento è presente l'architettura del sistema (MVC), la struttura dei dati e i servizi offerti da Ogni sottosistema.

Relazione con l'Object Design Document (ODD)

Nell'ODD sono contenuti i package e le classi del sistema.

Panoramica del sistema

Come definito nel System Design Document, il sistema avrà una struttura MVC (Model View Controller). La componente fondamentale di questo approccio è il controller che si occuperà della logica esecutiva di ogni sottosistema, nel model verranno indicate le entità persistenti del DB, infine nella view verranno mostrate le interfacce utente.

Caratteristiche da testare/non da testare

Per trovare le caratteristiche da testare ci concentriamo sui requisiti funzionali del sistema presenti nel rad, e nelle varie funzionalità dei sottosistemi individuati nel SDD, in generale daremo molta più importanza alle funzionalità di sicurezza e che noi (Vitulano Antonio, D'Avino Salvatore) riteniamo importanti per il software che stiamo progettando come ad esempio il login e l'aggiunta al carrello, le caratteristiche non andremo a testare sono quelle funzionalità ancora non implementate e quei requisiti non funzionali che risultano difficili da testare per due studenti universitari.

Criteri di successo/fail

Dopo aver individuato tutti i dati di input del sistema, quest'ultimi verranno raggruppati insieme in base alle caratteristiche in comune. Questa tecnica ci servirà per poter diminuire il numero di test da dover effettuare. Diremo che la fase di test ha successo se viene individuata effettivamente una failure all'interno del sistema, cioè l'output atteso per quel determinato input non è lo stesso previsto dall'oracolo. Successivamente la failure sarà analizzata e si passerà eventualmente alla sua correzione e verranno eseguiti nuovamente tutti i test necessari per verificare l'impatto che la modifica ha avuto sull'intero sistema. Diremo invece che il testing fallirà se l'output mostrato dal sistema coincide con quello previsto dall'oracolo.

Approccio

Testing di unità

Lo scopo del testing di unità è quello di testare ogni singola funzione presente all'interno del sistema. Ogni funzione rappresenta quella che viene definita come "unità".

Approccio scelto

L'approccio scelto è di tipo white-box, con tale tipo di approccio quindi andremo a testare il sistema conoscendone il funzionamento interno.

Testing di integrazione

Lo scopo del testing di integrazione è quello di "mettere insieme" le componenti testate precedentemente tramite il test di unità per vedere come funzionano una volta integrate tra loro.

Approccio scelto

Per effettuare il testing di integrazione si è scelto di adoperare un approccio bottom-up. Il vantaggio fondamentale di questa tipologia di testing è quello della riusabilità del codice. Questo tipo di approccio prevede però la costruzione driver per simulare l'ambiente chiamante. È stato scelto quindi questo tipo di approccio perché sembra quello più intuitivo e semplice.

Testing di sistema

Lo scopo del testing di sistema è quello di verificare che i requisiti richiesti dal cliente siano stati effettivamente rispettati e che il cliente risulti soddisfatto del sistema stesso. In questo tipo di testing si vanno a verificare le funzionalità utilizzate più spesso da parte del cliente e quelle che risultano più "critiche".

Approccio scelto

Useremo dei file, andremo a utilizzare il sistema come dei veri e propri clienti .

Sospensione e ripresa

Criteri di sospensione

La fase di testing verrà sospesa nel momento in cui saranno raggiunti i risultati previsti in accordo con quello che è il budget a disposizione o dovremo andare all'esame

Criteri di ripresa

Tutte le attività di testing riprenderanno nel momento in cui verranno effettuate modifiche all'interno del sistema.

Testing materials

Per effettuare il testing non avremo bisogno di altro che un PC senza nemmeno la connessione ad internet in quanto ogni tipo di test verrà effettuato sull'applicazione che girerà in locale.

Test Cases

Tutti i casi che andremo a testare li prendiamo dal SDD in particolare andremo a testare:

Svuota Carrello

Login

Logout

registrazione

effettua ordine

aggiungi al carrello

rimuovi al carrello

modifica quantità

Visualizza ordine

Aggiungi recensione

Chiedi al barista

Rispondi all'utente

Rimuovi utente

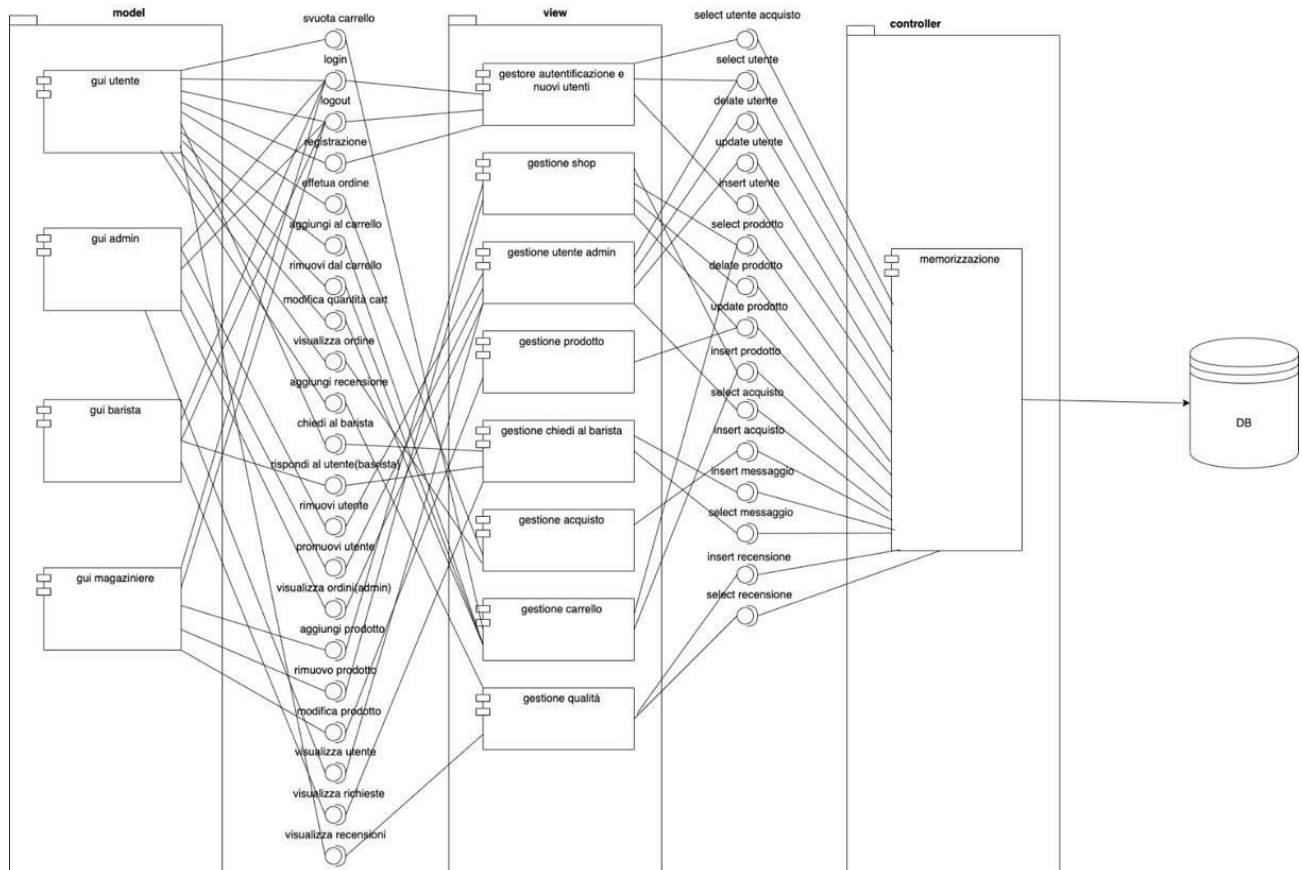
Promuovi utente

Visualizza ordini

Aggiungi prodotto

Rimuovi prodotto
 Modifica prodotto
 Visualizza utente
 Visualizza richieste
 Visualizza recensioni

In particolare andremo a testare tutte le funzioni descritte in questa tabella



naturalemnte andremo a testare solamente le funzioni più importanti

Casi di Test per ACQUISTO

PARAMETRI PRODOTTO(QUANTITA')
 OGGETTI DELL'AMBIENTE: DATABASE

CATEGORIA

- CATEGORIA 1: presenza della quantità in magazzino
- CATEGORIA 2 importo del pagamento
- CATEGORIA 3: Carta se esiste nella vita reale
- CATEGORIA 4: Password della carta
- CATEGORIA 5 Saldo della carta

SCELTE

Scelte per la categoria 1:

- Pre1: presenza nel database
- Pre2: Assenza nel database

Scelte per la categoria 2:

- Pag1: la carta ha effettivamente i soldi
- Pag2: la carta non ha effettivamente i soldi

Scelte per la categoria 3

- Esi1: la carta esiste nella vita reale
- Esi2: La carta non esiste nella vita reale

Scelte per la categoria 4:

pass1:la password della carta è corretta
pass2:la password della carta non è corretta

VINCOLI ASSOCIATI ALLE SCELTE

1. **Pre1**: Presenza della quantità in magazzino [property quantità_presente]
2. **Pre2**: Assenza della quantità in magazzino
3. **Esi1**: La carta esiste nella vita reale [property carta_esistente]
4. **Esi2**: La carta NON esiste nella vita reale
5. **Pass1**: Password corretta [if carta_esistente]
6. **Pass2**: Password NON corretta [if carta_esistente]
7. **Pag1**: Importo disponibile sulla carta [if carta_esistente and password_corretta]
8. **Pag2**: Importo NON disponibile sulla carta

TEST FRAME

1. **Pre1, Esi1, Pass1, Pag1** - ORACOLO: Acquisto completato con successo.
2. **Pre1, Esi1, Pass1, Pag2** - ORACOLO: Visualizzazione messaggio "Saldo insufficiente".
3. **Pre1, Esi1, Pass2, Pag1** - ORACOLO: Visualizzazione messaggio "Password carta errata".
4. **Pre1, Esi2, Pass1, Pag1** - ORACOLO: Visualizzazione messaggio "Carta non esistente".
5. **Pre2, Esi1, Pass1, Pag1** - ORACOLO: Visualizzazione messaggio "Prodotto non disponibile in magazzino".

TEST CASE

TC1: Quantità in magazzino: **Presente**, Carta: **Esiste**, Password: **Corretta**, Saldo: **Sufficiente** –
ORACOLO: Acquisto completato con successo.

TC2: Quantità in magazzino: **Presente**, Carta: **Esiste**, Password: **Corretta**, Saldo: **Insufficiente** –
ORACOLO: Messaggio "Saldo insufficiente".

TC3: Quantità in magazzino: **Presente**, Carta: **Esiste**, Password: **Errata**, Saldo: **Sufficiente** –
ORACOLO: Messaggio "Password carta errata".

TC4: Quantità in magazzino: **Presente**, Carta: **Non esiste**, Password: **Corretta**, Saldo: **Sufficiente** –
ORACOLO: Messaggio "Carta non esistente".

TC5: Quantità in magazzino: **Assente**, Carta: **Esiste**, Password: **Corretta**, Saldo: **Sufficiente** –
ORACOLO: Messaggio "Prodotto non disponibile in magazzino".

Specifica dei casi di test per il caso d'uso chiedi al barista

PARAMETRI: Messaggio

Oggetti dell'ambiente il messaggio()

Categoria

Categoria1:il messaggio è stato scritto dall'utente

Categoria 2 il messaggio è stato scritto dal barista

SCELTE

Per categoria 1

MES1U: il messaggio è stato effettivamente scritto dall'utente

MES2U: il messaggio non è stato scritto dall'utente

Per categoria 2

MES1B il barista ha scritto la risposta

MES2B il barista non ha scritto la risposta

VINCOLI ASSOCIATI ALLE SCELTE

U1=utente loggato [property stato_loggato]

U2=utente non loggato

TEST FRAME

MES2U MESS2B -ORACOLO non succede nulla nessuno ha scritto nulla

MES1U MESS2B- ORACOLO l'utente non vede la risposta

MESS2U MESS1B Oracolo- il barista ha commesso un errore nessuno gli ha inviato nulla

MESS1U- MESS1B Oracolo messaggio e risposta ricevuta

U1 Oracolo-Devi essere registrato

TEST CASE

TC1: Il messaggio non è stato scritto dall'utente, e il barista non ha scritto alcuna risposta – **ORACOLO:** Non succede nulla. Nessun messaggio viene visualizzato.

TC2: Il messaggio è stato scritto dall'utente, ma il barista non ha scritto la risposta – **ORACOLO:** L'utente non vede la risposta del barista.

TC3: Il messaggio non è stato scritto dall'utente, ma il barista ha scritto una risposta – **ORACOLO:** Il barista ha commesso un errore, nessuno gli ha inviato nulla.

TC4: Il messaggio è stato scritto dall'utente, e il barista ha scritto la risposta – **ORACOLO:** Messaggio e risposta ricevuti con successo.

TC5 l'utenete scrive ma non è registrato Oracolo:devi essere registrato

Specifica dei casi di test per i casi d'uso relativi all'Autenticazione dell'utente con Category Partition

PARAMETRI: username, password

Oggetti dell'ambiente: Database

CATEGORIE

categoria 1: username presente nel database

categoria 2: password associata a username nel database

SCELTE

Scelte per categoria 1

UP1: username presente nel database

UP2: username NON presente nel database

Scelte per categoria 2

PP1: password associata a username nel database

PP2: password NON associata a username nel database

VINCOLI ASSOCIATI ALLE SCELTE

UP1: username presente nel database [property username_presente]

UP2: username NON presente nel database

PP1: password associata a username nel database [if username_presente]

PP2: password NON associata a username nel database

TEST FRAME

UP1, PP1 – ORACOLO: Utente autenticato e indirizzato alla sua home page

UP1, PP2 – ORACOLO: Visualizzazione pagina di autenticazione con messaggio di errore

“username o password non corretta”

UP2, PP2 – ORACOLO: Visualizzazione pagina di autenticazione con messaggio di errore

“username o password non corretta”

TEST CASE

TC1: username: “pippo”, password: “pluto” – in DB presente la coppia (username = “pippo”, password = “pluto”) – ORACOLO: Utente autenticato e indirizzato alla sua home page

TC2: username: “pippo”, password: “pluto” – in DB presente la coppia (username = “pippo”, password = “topolino”) – ORACOLO: Visualizzazione pagina di autenticazione con messaggio di errore “username o password non corretta”

TC3: username: “pippo”, password: “pluto” – in DB non presente nessun utente con username “pippo” – ORACOLO: Visualizzazione pagina di autenticazione con messaggio di errore “username o password non corretta”

Specifica dei casi di test per i casi d’uso relativi ad una generica Registrazione di un account con Category Partition (la username è l’e-mail)

PARAMETRI: username, password,

Oggetti dell’ambiente: Database

CATEGORIE

categoria 1: formato email per la username

categoria 2: username presente nel database

categoria 3: lunghezza password (dalla specifica: la password deve avere almeno 8 caratteri)

categoria 4: formato password (dalla specifica: la password deve contenere lettere maiuscole e minuscole, cifre e caratteri speciali -!_\$(+)

SCELTE

Scelte per categoria 1

EM1: email corretta

EM2: email non corretta

Scelte per la categoria 2

UP1: username presente nel database

UP2: username NON presente nel database

Scelte per categoria 3

LP1: lunghezza password corretta (≥ 8)

LP2: lunghezza password NON corretta

Scelte per categoria 4

LP1: formato password corretto (contiene lettere maiuscole e minuscole, cifre e i caratteri speciali -!_\$.+)

LP2: formato password NON corretto

VINCOLI ASSOCIATI ALLE SCELTE

EM1: email corretta [property email_corretta]

EM2: email non corretta

UP1: username presente nel database [if email_corretta]

UP2: username NON presente nel database [property username_non_presente]

LP1: lunghezza password corretta (≥ 8) [property lunghezza_password_corretta]

LP2: lunghezza password NON corretta [if email_corretta and username_non_presente]

FP1: formato password corretto (contiene lettere maiuscole e minuscole, cifre e i caratteri speciali -!_\$.+)

FP2: formato password NON corretto [if lunghezza_password_corretta and email_corretta and username_non_presente]

VINCOLI ASSOCIATI ALLE SCELTE (Alternativa 2: le scelte [Error] non vanno combinate tra di loro)

EM1: email corretta

EM2: email non corretta [Error]

UP1: username presente nel database [Error]

UP2: username NON presente nel database

LP1: lunghezza password corretta (≥ 8)

LP2: lunghezza password NON corretta [Error]

FP1: formato password corretto (contiene lettere maiuscole e minuscole, cifre e i caratteri speciali -!_\$.+)

FP2: formato password NON corretto [Error]

TEST FRAME

- EM1, UP1, LP1, FP1 -- ORACOLO: Visualizzazione della pagina di registrazione con i campi già compilati e messaggio di errore accanto al campo username: "username già presente"
- EM1, UP2, LP1, FP1 -- ORACOLO: Registrazione effettuata
- EM1, UP2, LP2, FP1 -- ORACOLO: Visualizzazione della pagina di registrazione con i campi già compilati e messaggio di errore accanto al campo password: "password deve essere"

almeno 8 caratteri”

- EM1, UP2, LP1, FP2 -- ORACOLO: Visualizzazione della pagina di registrazione con i campi già compilati e messaggio di errore accanto al campo password: “password non rispetta il formato”
- EM2, UP2, LP1, FP1 – ORACOLO: Visualizzazione della pagina di registrazione con i campi già compilati e messaggio di errore accanto al campo username: “email non corretta”

Test Case 1

Parametri:

- Username: "pippo"
- Password: "pluto"
- Database: presente la coppia (username = "pippo", password = "pluto")

Risultato Atteso:

- Sei già registrato

Test Case 2

Parametri:

- Username: "pippo"
- Password: "pluto"
- Database: presente la (username = "pippo"

Risultato Atteso:

- Cambia username

Test Case 3

Parametri:

- Username: "pippo"
- Password: "Pluto123!"
- Database: presente password = "pluto"

Risultato Atteso:

- Registrazione avvenuta con successo

Test Case 4

Parametri:

- Username: "pippo"
- Password: "pluto123"

Risultato Atteso:

- Manca il carattere grande

Test Case 5**Parametri:**

- Username: "pippo"
- Password: "Pluto"
- Database: non presente la coppia

Risultato Atteso:

- Mancano i Numeri

Test Case 5**Parametri:**

- Username: "pippo"
- Password: "Pluto123"
- Database: non presente la coppia

Risultato Atteso:

- Mancano i caratteri speciali

Test Case 6**Parametri:**

- Username: "pippo"
- Password: "Pluto123!"
- Database: non presente la coppia

Risultato Atteso:

- Registrato correttamente

Specifica dei casi di test per i casi d'uso relativi all'Aggiunta al carrello dell'utente

PARAMETRI: prodotto,

quantità,

carrello

Oggetti dell'ambiente: Database, carrello dell'utente

CATEGORIE

- categoria 1: prodotto disponibile

- categoria 2: quantità valida
- categoria 3: carrello

SCELTE

Scelte per categoria 1

- PR1: prodotto disponibile
- PR2: prodotto esaurito

Scelte per categoria 2

- Q1: quantità valida (≥ 1)
- Q2: quantità non valida (≤ 0)

Scelte per categoria 3

- C1: carrello esistente
- C2: carrello vuoto

VINCOLI ASSOCIATI ALLE SCELTE

- PR1: prodotto disponibile
- PR2: prodotto esaurito
- Q1: quantità positiva
- Q2: quantità non valida
- C1: carrello già presente
- C2: carrello vuoto

TEST FRAME

- **PR1, Q1, C1 – ORACOLO:** Prodotto aggiunto al carrello, quantità aggiornata nel carrello esistente.
- **PR1, Q2, C1 – ORACOLO:** Visualizzazione messaggio di errore "Quantità non valida" nel carrello esistente.
- **PR1, Q1, C2 – ORACOLO:** Prodotto aggiunto al carrello, creazione di un nuovo carrello.
- **PR1, Q2, C2 – ORACOLO:** Visualizzazione messaggio di errore "Quantità non valida" con carrello vuoto.
- **PR2, Q1, C1 – ORACOLO:** Visualizzazione messaggio di errore "Prodotto esaurito" nel carrello esistente.
- **PR2, Q2, C1 – ORACOLO:** Visualizzazione messaggio di errore "Prodotto esaurito" nel carrello esistente.
- **PR2, Q1, C2 – ORACOLO:** Visualizzazione messaggio di errore "Prodotto esaurito" con carrello vuoto.
- **PR2, Q2, C2 – ORACOLO:** Visualizzazione messaggio di errore "Prodotto esaurito" con carrello vuoto.

TEST CASE

- **TC1:** prodotto: "vodka", quantità: 2, carrello: esistente – ORACOLO: Prodotto aggiunto al carrello, aggiornamento quantità nel carrello.
- **TC2:** prodotto: "vodka", quantità: 0, carrello: esistente – ORACOLO: Visualizzazione messaggio di errore "Quantità non valida".
- **TC3:** prodotto: "vodka", quantità: 1, carrello: vuoto – ORACOLO: Prodotto aggiunto al carrello, creazione di un nuovo carrello.
- **TC4:** prodotto: "vodka", quantità: 3, carrello: esistente (prodotto esaurito) – ORACOLO: Visualizzazione messaggio di errore "Prodotto esaurito".

Specifica dei casi di test della funzione ricerca prodotti

Funzione di ricerca prodotti

PARAMETRI: stringa(il prodotto)

Oggetti dell'ambiente: Database, lista dei prodotti

CATEGORIE

- **categoria 1:** prodotto presente
- **categoria 2:** prodotto non presente

SCELTE

Scelte per categoria 1

- **PR1:** prodotto disponibile
- **PR2:** prodotto non disponibile

VINCOLI ASSOCIATI ALLE SCELTE

- **PR1:** prodotto presente nel database
- **PR2:** prodotto non presente nel database

TEST FRAME

PR1 – ORACOLO: I prodotti correlati vengono visualizzati nella lista dei risultati.

PR2 – ORACOLO: Visualizzazione messaggio "Nessun prodotto trovato".

TEST CASE

- **TC1:** parola chiave: "smartphone" – prodotto presente nel database con il nome "smartphone"
ORACOLO: Visualizzazione dei prodotti correlati (smartphone) nella lista dei risultati.
- **TC2:** parola chiave: "tablet" – prodotto non presente nel database
ORACOLO: Visualizzazione messaggio di errore "Nessun prodotto trovato".

- **TC3:** parola chiave: "vodka" – prodotto presente nel database con il nome "laptop"
ORACOLO: Visualizzazione dei prodotti correlati (laptop) nella lista dei risultati.
- **TC4:** parola chiave: "vodka" – prodotto presente nel database con il nome "smartwatch"
ORACOLO: Visualizzazione dei prodotti correlati (smartwatch) nella lista dei risultati.
- **TC5:** parola chiave: "vodka" – prodotto presente nel database con il nome "wireless headphones"
ORACOLO: Visualizzazione dei prodotti correlati nella lista dei risultati, anche se la parola chiave non corrisponde esattamente al nome.
- **TC6:** parola chiave: "voka" – prodotto non presente nel database
ORACOLO: Visualizzazione messaggio di errore "Nessun prodotto trovato".
- **TC7:** parola chiave: "voka" – prodotto presente nel database con il nome "voka 4k"
ORACOLO: Visualizzazione dei prodotti correlati (vodka 4k) nella lista dei risultati.

Testing schedule

Il testing verrà effettuato parallelamente allo sviluppo del sistema, al fine di individuare e correggere errori nel momento in cui essi verranno introdotti. L'attività di testing è prerogativa di tutti i membri del team, i quali dovranno occuparsi della generazione e dell'esecuzione dei casi di test relativi alle funzionalità che hanno implementato. Tale attività è fondamentale nello sviluppo di un sistema software in quanto la sua mancanza può portare al completo fallimento del sistema. Data tale premessa ne risulta fondamentale la schedulazione e la successiva documentazione.