



Test Plan

Versione	1.0
Data	15/01/2020
Destinatario	Prof. Andrea De Lucia
Presentato da	Riccardo Martiniello Davide Cresci Alessio Rizzolo Giuseppe Caiazzo

Revision History

data	versione	descrizione	autori
15/01/20	1.0	introduzione	Tutti i membri
15/01/20	1.0	Documenti correlati	Tutti i membri
15/01/20	1.0	Panoramica del sistema	Tutti i membri
15/01/20	1.0	Funzionalità da testare e non	Tutti i membri
15/01/20	1.0	Pass/Fail criteria e approccio	Tutti i membri
15/01/20	1.0	Sospensione e ripristino	Tutti i membri
15/01/20	1.0	Materiale per il testing	Tutti i membri
15/01/20	1.0	Test Cases	Tutti i membri
15/01/20	1.0	Pianificazione dei test	Tutti i membri

Sommario

1. Introduzione	4
2. Documenti Correlati	4
2.1 Relazione con il documento di Analisi Dei Requisiti(RAD)	4
2.2 Relazione con il System Design Document(SDD)	4
2.3 Relazione con l'Object Design Document(ODD)	4
3. Panoramica del sistema	5
4. Funzionalità da testare e non testare	5
5. Pass/Fail criteria	6
6. Approccio	6
6.1 Test di unità	6
6.2 Test di integrazione	6
6.3 Testing di sistema	6
7. Sospensione e ripristino	7
7.1 Criteri di sospensione	7
7.2 Criteri di ripristino	7
8. Materiale per il testing	7
9. Test Cases	7
9.1 Gestione Cliente	7
9.1.1 Aggiungi Carta	7
9.1.2 Aggiungi Indirizzo	9
9.2 Gestione Autenticazione	11
9.2.1 Login	11
9.3 Gestione Registrazione	12
9.3.1 Registrazione	12
9.4 Gestione Dipendenti	13
9.4.1 Scegli Prezzo Prodotti	13
9.4.2 Inserisci Prodotto	14
9.5 Gestione Carrello	15
9.5.1 Aggiungi prodotti al carrello	15
10. Pianificazione dei test	17
10.1 Determinazione dei ruoli	17
10.2 Determinazione dei rischi	17
10.3 Decomposizione gerarchica del sistema	18
10.4 Organizzazione delle attività di testing	18

1. Introduzione

Lo scopo del sistema è quello di pianificare l'attività di testing all'interno della piattaforma Dress-Store per verificare se esistono differenze tra il comportamento atteso e il comportamento osservato. In questa attività andremo a individuare i possibili errori nel codice sorgente causanti incident all'interno del sistema prima che l'utente finale esegua delle interazioni con il sistema. Le attività di testing sono organizzate in modo da poter verificare il funzionamento dei seguenti sottosistemi:

1. Gestione Cliente
2. Gestione Autenticazione
3. Gestione Registrazione
4. Gestione Dipendenti
5. Gestione Carrello

Si noti che saranno oggetto di testing solo alcune funzionalità di tali sottosistemi. I documenti che precedono questa fase saranno di grande importanza per il testing, ci assicureranno che si stia procedendo in modo corretto.

2. Documenti Correlati

La documentazione antecedente è strettamente correlata con la pianificazione dei test, questo perché già nei documenti precedenti abbiamo definito come alcuni servizi dovrebbero funzionare. Questa caratteristica prevede di rilevare informazioni come i comportamenti attesi durante l'esecuzione di alcune funzionalità. Vediamo, quindi, che la fase di testing prevede di verificare se ci siano differenze tra il funzionamento del sistema che si è progettato e il reale funzionamento del sistema implementato.

2.1 Relazione con il documento di Analisi Dei Requisiti(RAD)

La relazione tra la fase di testing e quella di analisi dei requisiti si basa sui requisiti funzionali e non funzionali descritti nel RAD che devono essere rispettati dal sistema durante la fase di testing.

2.2 Relazione con il System Design Document(SDD)

La relazione tra la fase di testing e la fase di design del sistema si basa sulla suddivisione del sistema in sottosistemi e la divisione in strati (Presentation layer, Application layer e Storage layer). Il testing deve essere fedele alla suddivisione progettata in fase di design in modo tale da rimanere coerente con il sistema progettato. La struttura del sistema verrà analizzata al fine di scoprire le differenze presenti tra essa e quella prevista durante la fase di System Design.

2.3 Relazione con l'Object Design Document (ODD)

La relazione tra la fase di testing e la fase di object design si basa principalmente sul riferimento della verifica dei contratti e dei componenti raffinati all'interno di tale documento.

3. Panoramica del sistema

Come riportato nel System Design Document (SDD) la struttura del nostro sistema è suddivisa secondo un'architettura strutturata su tre livelli (Three-Tier): User Interface Layer, Application Layer, Storage Layer.

- User Interface Layer – si occupa della gestione delle interfacce utente
- Application Layer – si occupa di fornire all'applicazione specifiche funzionalità
- Storage Layer – si occupa della gestione dei database di sistema

Il sistema, inoltre, è stato diviso in sottosistemi a secondo della gestione di una particolare categoria di servizi. La suddivisione è fatta nel seguente modo:

- Gestione Cliente
- Gestione Autenticazione
- Gestione Registrazione
- Gestione Dipendenti
- Gestione Carrello

Quasi tutte le gestioni prevedono operazioni di inserimento, modifica, cancellazione, visualizzazione e ricerca di dati. Saranno proprio tali funzionalità ad essere oggetto di testing del sistema.

4. Funzionalità da testare e non testare

Di seguito sono elencate tutte le funzionalità da testare in relazione ai sottosistemi a cui appartengono:

1. Gestione Cliente:
 - Aggiungi Carta
 - Aggiungi Indirizzo
2. Gestione Autenticazione:
 - Login
3. Gestione Registrazione:
 - Registrazione
4. Gestione Dipendenti:
 - Scegli prezzo prodotti
 - Inserisci Prodotto
5. Gestione Carrello:
 - Aggiungi prodotti al carrello

5. Pass/Fail criteria

Nel caso si verificassero degli errori durante la fase di testing, si interverrà sulle porzioni di codice che causano il problema e si rieseguirà il testing in modo da verificare che le modifiche non abbiano impattato sul comportamento di altre componenti del sistema.

Un input supererà il test solo se i valori attesi e quelli ottenuti combaceranno.

6. Approccio

Verrà applicata una strategia "Bottom-up". Si inizierà con il testing di unità dei singoli componenti in modo da testare, in modo esaustivo, le unità più piccole.

Successivamente si procederà con il test di integrazione, nel quale l'attenzione sarà rivolta sulle interfacce dell'unità. Infine si eseguirà il testing di sistema, dove l'attenzione sarà rivolta alla verifica del corretto funzionamento del sistema.

Durante la fase di testing, per testare il codice verrà usato il software JUnit, mentre sarà utilizzato Selenium per testare l'integrazione delle componenti.

6.1 Test di unità

In questa fase verranno ricercate le condizioni di fallimento delle singole componenti. Verranno usati Test Driver e Test Stub che rappresenteranno l'implementazione parziale di componenti che dipendono o da cui dipendono le componenti testate. La strategia di testing si baserà sulla strategia "Black-box". Il testing sarà quindi strutturato in un'analisi Input/Output delle singole componenti. In questa fase si avrà particolare attenzione sulla suddivisione delle classi degli input così da poter verificare ogni componente su qualsiasi tipo di input.

6.2 Test di integrazione

Questa fase prevede l'aggregazione delle singole componenti e il loro testing. Per questa fase di test, verrà adottata una strategia black-box. Questa fase prevede l'iterazione con l'interfaccia del sistema legato ai servizi delle funzionalità del sistema da testare. Il testing viene effettuato per individuare dei possibili fault e per testare la logica applicativa del software.

6.3 Testing di sistema

Anche per questa fase di testing verrà adottata una strategia black-box. Verranno testate le funzionalità dell'intero sistema e si verificherà se i requisiti sono stati implementati. Ci si baserà sul testing delle funzionalità principali basate sulle priorità dei requisiti individuati durante la fase di analisi.

7. Sospensione e ripristino

Durante il testing potrebbero esserci casi in cui debba essere sospeso e/o ripristinato.

7.1 Criteri di sospensione

Comprendono tutti quei casi critici in cui gli errori hanno un impatto dannoso sul progresso dell'attività di testing. Esempi possono essere:

- Crash del server
- Crash del database
- Fallimento di funzionalità interne
- Problemi relativi all'ambiente di sviluppo del testing

Se il testing di un'unità viene sospeso, si procederà alla correzione della stessa.

7.2 Criteri di ripristino

I criteri di ripristino rappresentano le azioni da apportare in seguito ad una modifica o ad una correzione di qualche componente durante la fase di testing. La ripresa del sistema avviene solo quando tali errori vengono risolti, ripartendo dal test case che ha causato l'errore.

8. Materiale per il testing

Per il testing servirà un computer con installato il codice che si vuole testare, una IDE (Eclipse preferibilmente) con installato JUnit ed un web browser per testare il sistema con Selenium.

9. Test Cases

9.1 Gestione Cliente

9.1.1 Aggiungi Carta

9.1.1.1 Category Partition

Parametro: Nome	
Formato: [A-Za-z.]	
Formato [FORNOM]	<ol style="list-style-type: none">1. rispetta il formato [propertyFormatoFORNOMOK, rispetta il formato [A-Za-z.]]2. non rispetta il formato [error]

Parametro: Cognome	
Formato: [A-Za-z.]	
Formato [FORCOG]	<ol style="list-style-type: none"> rispetta il formato [propertyFormatoFORCOGOK, rispetta il formato [A-Za-z.]] non rispetta il formato [error]

Parametro: Numero carta di credito	
Formato: [0-9]	
Lunghezza [LUNCRE]	<ol style="list-style-type: none"> = 16 [propertyLunghezzaLUNCREOK] ! = 16 [error]
Formato [FORCRE]	<ol style="list-style-type: none"> rispetta il formato [ifLunghezzaLUNCREOK] [propertyFormatoFORCOK, rispetta il formato [0-9]] non rispetta il formato [ifLunghezzaLUNCREOK] [error]

Parametro: CVV	
Formato: [0-9]	
Lunghezza [LCVV]	<ol style="list-style-type: none"> = 3 [propertyLunghezzaLCVVOK] ! = 3 [error]
Formato [FCVV]	<ol style="list-style-type: none"> Rispetta il formato [ifLunghezzaLCVVOK] [propertyFormatoFCVVOK, rispetta il formato [0-9]] Non rispetta il formato [ifLunghezzaLCVVOK] [error]

Parametro: Data di Scadenza	
Formato: AAAA/MM/DD	
Formato [FORSC]	<ol style="list-style-type: none"> rispetta il formato [propertyFormatoFORSCOK, rispetta il formato [AAAA/MM/DD.]] non rispetta il formato [error]
Corretto [CRSC]	<ol style="list-style-type: none"> La data di scadenza deve essere successiva a quella odierna [ifFormatoFORSCOK] [propertyCorrettoCRSCOK] La data di scadenza è precedente o uguale a quella odierna [ifFormatoFORSCOK] [error]

9.1.1.2 Test Case

Codice	Combinazione	Esito
TC_1.1_1	FORNOM2	Errato
TC_1.1_2	FORFOM1, FORCOG2	Errato
TC_1.1_3	FORNOM1, FORCOG1, ILUNCRE2	Errato
TC_1.1_4	FORNOM1, FORCOG1, LUNCRE1, FORCRE2	Errato
TC_1.1_5	FORNOM1, FORCOG1, LUNCRE1, FORCRE1, LCVV2	Errato
TC_1.1_6	FORNOM1, FORCOG1, LUNCRE1, FORCRE1, LCVV1, FCVV2	Errato
TC_1.1_7	FORNOM1, FORCOG1, LUNCRE1, FORCRE1, LCVV1, FCVV1, FORSC2	Errato
TC_1.1_8	FORNOM1, FORCOG1, LUNCRE1, FORCRE1, LCVV1, FCVV1, FORSC1, CRSC2	Errato
TC_1.1_9	FORNOM1, FORCOG1, LUNCRE1, FORCRE1, LCVV1, FCVV1, FORSC1, CRSC1	Corretto

9.1.2 Aggiungi Indirizzo

9.1.2.1 Category Partition

Parametro: Nome	
Formato: [A-Za-z.]	
Formato [FORNOME]	<ol style="list-style-type: none">rispetta il formato [propertyFormatoFORNOMEOK, rispetta il formato [A-Za-z.]]non rispetta il formato [error]

Parametro: Cognome	
Formato: [A-Za-z.]	
Formato [FORCOGN]	<ol style="list-style-type: none">rispetta il formato [propertyFormatoFORCOGNOK, rispetta il formato [A-Za-z.]]non rispetta il formato [error]

Parametro: Cellulare	
Formato: [0-9]	
Lunghezza [LUCCELL]	<ol style="list-style-type: none"> = 10 [propertyLunghezzaLUCELLOK] != 10 [error]
Formato [FORCELL]	<ol style="list-style-type: none"> rispetta il formato [ifLunghezzaLUCELLOK] [propertyFormatoFORCELLOK, rispetta il formato [0-9]] non rispetta il formato [ifLunghezzaLUCELLOK] [error]

Parametro: Città	
Formato: [A-Za-z.]	
Formato [FORCITT]	<ol style="list-style-type: none"> rispetta il formato [propertyFormatoFORCITTOK, rispetta il formato [A-Za-z.]] non rispetta il formato [error]

Parametro: Provincia	
Formato: [A-Z] [A-Z]	
Formato [FORPRO]	<ol style="list-style-type: none"> rispetta il formato [propertyFormatoFORPROOK, rispetta il formato [A-Za-z.]] non rispetta il formato [error]

Parametro: CAP	
Formato: [0-9]	
Lunghezza[LUCAP]	<ol style="list-style-type: none"> = 5 [propertyLunghezzaLUCAPOK] != 5 [error]
Formato [FORCOGNO]	<ol style="list-style-type: none"> rispetta il formato [ifLunghezzaLUCAPOK] [propertyFormatoFORCOGNOOK, rispetta il formato [0-9]] non rispetta il formato [error]

Parametro: Via	
Formato: via [A-Za-z.]	
Formato [FORVIA]	<ol style="list-style-type: none"> rispetta il formato [propertyFormatoFORVIAOK, rispetta il formato: via [A-Za-z.]] non rispetta il formato [error]

9.1.2.2 Test Case

Codice	Combinazione	Esito
TC_1.2_1	FORNOME2	Errato
TC_1.2_2	FORNOME1, FORCOGN2	Errato
TC_1.2_3	FORNOME1, FORCOGN1, LUCCELL2	Errato
TC_1.2_4	FORNOME1, FORCOGN1, LUCCELL1, FORCECELL2	Errato
TC_1.2_5	FORNOME1, FORCOGN1, LUCCELL1, FORCECELL1, FORCITT2	Errato
TC_1.2_6	FORNOME1, FORCOGN1, LUCCELL1, FORCECELL1, FORCITT1, FORPRO2	Errato
TC_1.2_7	FORNOME1, FORCOGN1, LUCCELL1, FORCECELL1, FORCITT1, FORPRO1, LUCAP2	Errato
TC_1.2_8	FORNOME1, FORCOGN1, LUCCELL1, FORCECELL1, FORCITT1, FORPRO1, LUCAP1, FORCOGNO2	Errato
TC_1.2_9	FORNOME1, FORCOGN1, LUCCELL1, FORCECELL1, FORCITT1, FORPRO1, LUCAP1, FORCOGNO1, FORVIA2	Errato
TC_1.2_10	FORNOME1, FORCOGN1, LUCCELL1, FORCECELL1, FORCITT1, FORPRO1, LUCAP1, FORCOGNO1, FORVIA1	Corretto

9.2 Gestione Autenticazione

9.2.1 Login

9.2.1.1 Category Partition

Parametro: E-mail	
Formato: [A-Za-z0-9.] @ [a-z.]	
Lunghezza [LU]	<ol style="list-style-type: none"> <8 [error] [iformatoESOK] >=8 [iformatoESOK][property lunghezza LUOK]

Esistenza [ES]	<ol style="list-style-type: none"> 1. L'email esiste nel sistema [ifLunghezzaLUOK][propertyEsistenzaESOK] 2. L'email non esiste nel sistema [ifLunghezzaLUOK][error]
----------------	--

Parametro: Password	
Formato: [A-Za-z0-9.]	
Lunghezza [LU]	<ol style="list-style-type: none"> 1. <10 [error] [iformatoCROK] 2. >=10 [iformatoCROK][property lunghezza LUOK]
Correttezza [CR]	<ol style="list-style-type: none"> 1. La password non è corretta [ifLunghezzaLUOK][error] 2. La password è corretta [ifLunghezzaLUOK][propertyCorrettezzaCROK]

9.2.1.2 Test Case

Codice	Combinazione	Esito
TC_1.3_1	LU1	Errato
TC_1.3_2	LU2, ES1	Errato
TC_1.3_3	LU2, ES2, CR1	Errato
TC_1.3_4	LU2, ES2, CR2	Corretto

9.3 Gestione Registrazione

9.3.1 Registrazione

9.3.1.1 Category Partition

Parametro: Nome	
Formato: [A-Za-z.]	
Formato [FORN]	<ol style="list-style-type: none"> 1. rispetta il formato [propertyFormatoFORNOK, rispetta il formato [A-Za-z.]] 2. non rispetta il formato [error]

Parametro: Cognome	
Formato: [A-Za-z.]	
Formato [FORC]	<ol style="list-style-type: none"> rispetta il formato [propertyFormatoFORCOK, rispetta il formato [A-Za-z.]] non rispetta il formato [error]

Parametro: Data di Nascita	
Formato: AAAA/MM/DD	
Formato [FORDN]	<ol style="list-style-type: none"> rispetta il formato [propertyFormatoFORDNOK, rispetta il formato [AAAA/MM/DD.]] non rispetta il formato [error]
Corretto [CRDN]	<ol style="list-style-type: none"> La data di nascita dell'utente deve essere ≥ 18 a partire dall'anno corrente [ifFormatoFORDNOK][propertyCorrettoCRDDNOK] La data di nascita dell'utente è inferiore a 18. [ifFormatoFORDNOK][error]

9.3.1.2 Test Case

Codice	Combinazione	Esito
TC_1.4_1	FORN2	Errato
TC_1.4_2	FORN1, FORC2	Errato
TC_1.4_3	FORN1, FORC1, FORDN2	Errato
TC_1.4_4	FORN1, FORC1, FORDN1, CRDN2	Errato
TC_1.4_5	FORN1, FORC1, FORDN1, CRDN1	Corretto

9.4 Gestione Dipendenti

9.4.1 Scegli Prezzo Prodotti

9.4.1.1 Category Partition

Parametro: Prezzo	
Inserito [INSP]	<ol style="list-style-type: none"> Prezzo inserito [OK] Prezzo non inserito [error]

9.4.1.2 Test Case

Codice	Combinazione	Esito
TC_1.5_1	INSP2	Errato
TC_1.5_2	INSP1	Corretto

9.4.2 Inserisci Prodotto

9.4.2.1 Category Partition

Parametro: Foto	
Formato: [A-Za-z.].jpg]	
Formato [FORFOT]	<ol style="list-style-type: none">rispetta il formato [propertyFormatoFORFOTOK, rispetta il formato [A-Za-z.].jpg]]non rispetta il formato [error]

Parametro: Modello	
Formato: [A-Za-z.]	
Formato [FORMOD]	<ol style="list-style-type: none">rispetta il formato [propertyFormatoFORMODOK, rispetta il formato [A-Za-z.]]non rispetta il formato [error]

Parametro: Marca	
Formato: [A-Za-z.]	
Formato [FORMAR]	<ol style="list-style-type: none">rispetta il formato [propertyFormatoFORMAROK, rispetta il formato [A-Za-z.]]non rispetta il formato [error]

Parametro: Quantità	
Formato: [1-9]	
Formato [FORQT]	<ol style="list-style-type: none">rispetta il formato [propertyFormatoFORMODOK, rispetta il formato [1-9.]]non rispetta il formato [error]

Parametro: Codice	
Formato: [A-Za-z0-9.]	
Lunghezza [LUCOD]	1. =13 [propertyLunghezzaLUCODOK] 2. <13 and >13 [error]
Formato [FORCOD]	1. Rispetta il formato[ifLunghezzaLUCODOK][propertyFormatoFORCODOK,rispetta il formato[A-Za-z0-9.]] 2. Non rispetta il formato [ifLunghezzaLUCODOK][error]

Parametro: Descrizione	
Lunghezza [LUDES]	1. <=200 [propertyLunghezzaLUDESOK] 2. >200 [error]

9.4.2.2 Test Case

Codice	Combinazione	Esito
TC_1.6_1	FORFOT2	Errato
TC_1.6_2	FORFOT1, FORMOD2	Errato
TC_1.6_3	FORFOT1, FORMOD1, FORMAR2	Errato
TC_1.6_4	FORFOT1, FORMOD1, FORMAR1, FORQT2	Errato
TC_1.6_5	FORFOT1, FORMOD1, FORMAR1, FORQT1, LUCOD2	Errato
TC_1.6_6	FORFOT1, FORMOD1, FORMAR1, FORQT1, LUCOD1, FORCOD2	Errato
TC_1.6_7	FORFOT1, FORMOD1, FORMAR1, FORQT1, LUCOD1, FORCOD1, LUDES2	Errato
TC_1.6_8	FORFOT1, FORMOD1, FORMAR1, FORQT1, LUCOD1, FORCOD1, LUDES1	Corretto

9.5 Gestione Carrello

9.5.1 Aggiungi prodotti al carrello

9.5.1.1 Category Partition

Parametro: Taglia	
Inserito [INSTA]	<ol style="list-style-type: none">1. Taglia inserita [OK]2. Taglia non inserita [default]

9.5.1.2 Test Case

Codice	Combinazione	Esito
TC_1.8_1	INSTA2	Default
TC_1.8_2	INSTA1	Corretto

10. Pianificazione dei test

Il team dedito alla compilazione del testing deve essere composto da persone che hanno una completa conoscenza del dominio applicativo e del dominio delle soluzioni del sistema. Inoltre, devono conoscere in maniera completa tutte le tecniche di testing utilizzate e nominate all'interno del Test Plan e del Test case specification. Le attività che comportano tale fase dovranno essere fatte nei tempi, nei costi e nei vincoli di qualità specificati.

Il sistema revisionato dovrà, successivamente alla correzione, essere verificato attraverso altri casi di test per consentire di assicurarsi che le modifiche sono state compiute in maniera corretta e verificare se tali cambiamenti hanno introdotto nuovi errori. L'attività di testing è fondamentale nello sviluppo di un sistema software in quanto la mancanza di tale attività o una cattiva gestione della stessa può portare al completo fallimento del sistema e, in casi estremi, dell'intero progetto. Data l'importanza del testing, la schedulazione delle sue attività sono fondamentali.

10.1 Determinazione dei ruoli

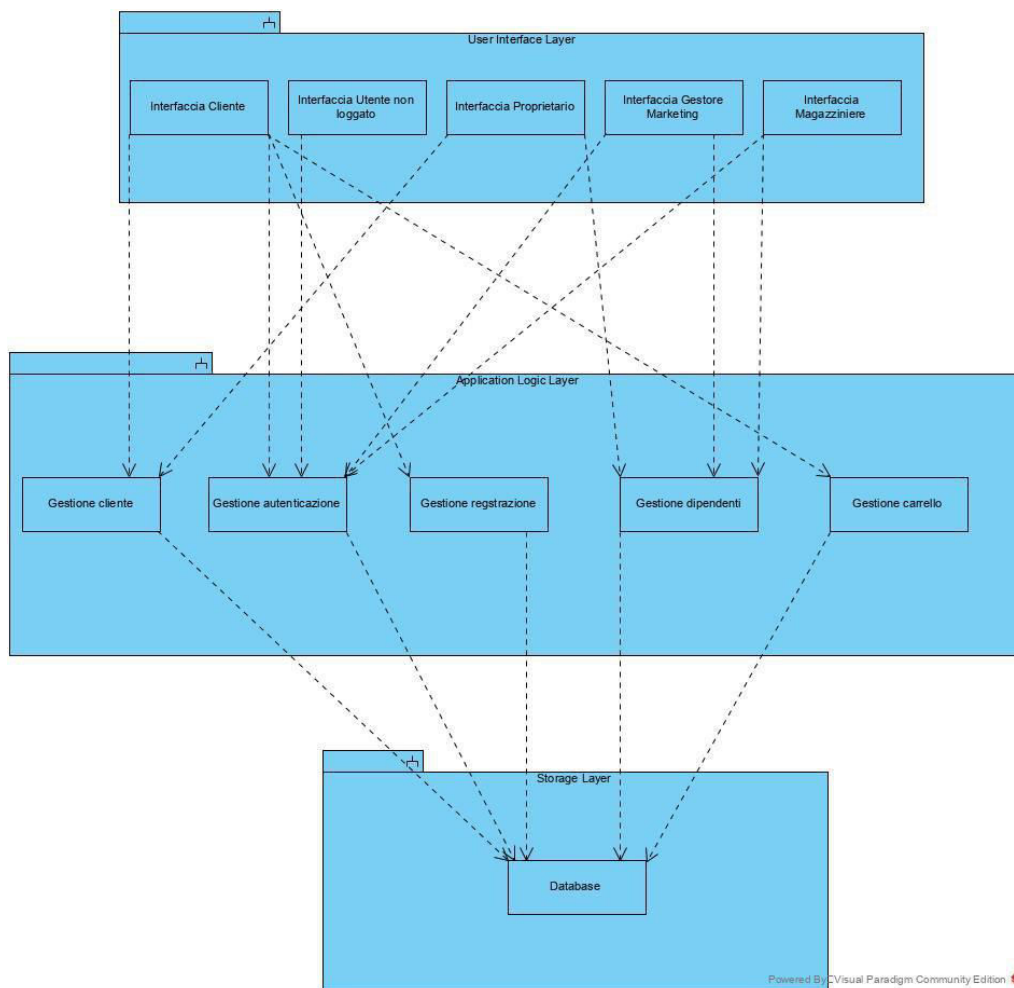
Tutta l'attività di testing viene eseguita da Giuseppe Caiazzo. Le attività relative al testing di unità come correzioni e cambiamenti verranno delegate agli sviluppatori che si occuperanno delle modifiche a livello implementativo. Tale organizzazione serve principalmente per alleggerire il carico di lavoro da parte del tester che potrà dedicarsi maggiormente sul lavoro di testing funzionale.

10.2 Determinazione dei rischi

Nel caso in cui vi sia un'elevata quantità di errori, il sistema rischia di essere consegnato in ritardo. Per sovvenire a questa situazione, nel caso in cui si verifichi, si è deciso di effettuare una pianificazione verticale dei test funzionali. Tale approccio consente di rilasciare un numero minore di funzionalità nei tempi previsti, ma in maniera completa e funzionale.

10.3 Decomposizione gerarchica del sistema

La divisione gerarchica del sistema è stata mappata in 3 livelli gerarchici come nel seguente diagramma:



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.