



TP

TEST PLAN



Sommario

1. Introduzione	1
2. Riferimenti e relazioni	1
2.1 Relazioni con il documento di analisi dei requisiti (RAD)	1
2.2 Relazioni con il documento di system design (SDD)	2
2.3 Relazioni con il documento di object design (ODD)	2
3. Panoramica del sistema	2
4. Funzionalità testate e non	3
5. Strumenti per il testing	5
5.1 Strumenti hardware	5
5.2 Strumenti software	5
6. Test case	6
6.1 Registrazione	6
6.1.1 Formal Test Specification	7
6.2 Login	11
6.2.1 Formal Test Specification	12
6.3 Caricamento prodotto	13
6.3.1 Formal Test Specification	14
6.4 Ricerca prodotto per nome	17
6.4.1 Formal Test Specification	17
6.5 Ricerca prodotto per categoria	18
6.5.1 Formal Test Specification	18
6.6 Prenotazione riparazione	18
6.6.1 Formal Test Specification	18
7. Testing schedule	20
7.1 Gestione dei rischi	20
7.2 Organizzazione delle attività	20



1.INTRODUZIONE

Il testing consiste nel trovare le differenze tra il comportamento atteso specificato attraverso il modello di sistema e il comportamento che si ha dopo l'implementazione del sistema.

Lo scopo principale di questa attività è quello di testare il sistema e di rilevare gli errori e i problemi che si vengono a verificare in fase di implementazione. Questo vuol dire che nella fase di testing andiamo a cercare di creare il massimo numero di errori in modo tale che questi possano essere corretti in questa suddetta fase. Lo scopo del documento è quello di andare a definire i test case su cui verranno testate le 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 e quindi appartenenti a classi valide per ogni campo di input che rispettano le condizioni definite nel documento di test plan. Ci sarà inoltre un test case per ogni input che prevede una classe non valida.

2.RIFERIMENTI E RELAZIONI

In questa parte vengono descritte le relazioni con gli altri documenti prodotti durante la fase di sviluppo del progetto software TuttoElettronica. Tali relazioni sono necessarie al fine di produrre una fase di testing che sia coerente con tutte le specifiche dedotte durante l'analisi ed il raffinamento dei requisiti e delle funzionalità che il sistema dovrà fornire. Le relazioni che andiamo a prendere in considerazione sono con:

- RAD – documento di analisi dei requisiti
- SDD – documento di system design
- ODD – documento di object design

2.1 RELAZIONE 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 dei diversi attori descritti nel documento di analisi (RAD):

- Cliente
- Gestore
- Prodotti

Per ciascun cliente e gestore dovrà essere testata la validità dei campi associati.

Per i prodotti dovranno essere testati inserimento, ricerca, quantità.

Inoltre deve essere considerato il flusso di controllo descritto nei diagrammi delle sequenze di operazioni, in modo da testare determinate funzionalità all'interno e all'esterno del proprio flusso standard.



2.2 RELAZIONI CON IL DOCUMENTO DI SYSTEM DESIGN (SDD)

La pianificazione dei test delle componenti rispecchia la decomposizione in sottosistemi specificata nel documento del 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 e quello specificato.

3. PANORAMICA DEL SISTEMA

Il sito di commercio elettronico che si vuole realizzare è concepito sull'idea di dare la possibilità, a dei clienti, di riparare prodotti elettronici. Il sito, inoltre, intende pubblicizzare i prodotti da esso venduti permettendo, inoltre, ai clienti la possibilità di effettuare prenotazioni online.

Il sistema TuttoElettronica segue il modello MVC per la suddivisione dei livelli:

- Model: che rappresenta i dati dell'applicazione;
- View: che rappresenta la visualizzazione degli oggetti;
- Controller: che rappresenta l'insieme di regole che permettono la trasformazione degli input sulle viste in modifiche del modello.

Ogni livello, come descritto nel documento SDD, è stato suddiviso in sottosistemi in modo da raggruppare le componenti.

Ciascun sottosistema, appartenente a ciascun livello, deve essere testato.

Il presente test plan relativo al sistema TuttoElettronica si pone i seguenti obiettivi:

- Dettagliare le attività richieste per preparare e condurre il testing;
- Definire le fonti usate per preparare la pianificazione;
- Affinchè il testing riesca a raggiungere al meglio i suoi obiettivi, si è ritenuto necessario non condurre tale processo in modo casuale, cioè guidato solo dalla fantasia e dalla esperienza dei test-engineer, ma definendo i punti generali al quale far riferimento:
 - **Individuare quando deve partire il processo di testing e quale è il criterio di terminazione da utilizzare**
Identificare a priori lo stato di inizio e di fine dell'interno di un processo è fondamentale per evitare che il processo cominci quando ancora non sono presenti tutte le informazioni necessarie o che si prolunghi eccessivamente.
 - **Definire per ogni test i risultati attesi**
 - La descrizione dei risultati ottenuti dall'esecuzione di un test case è determinante poiché, senza una precisa connotazione degli output attesi, si corre il rischio di interpretare come corretti dei risultati che viceversa non lo sono.
 - **Analizzare a fondo i risultati di ciascun test**
 - **Non pianificare il testing assumendo tacitamente che nessun errore sarà individuato**



Questo è un comportamento che deriva direttamente da un'errata interpretazione del documento di testing, laddove si crede che il testing sia la verifica delle funzionalità del sistema, mentre il vero processo di testing è la ricerca del maggior numero di errori.

- **Il miglior test case è quello che ha la più alta probabilità di scoprire errori non ancora rilevati.**

Questo principio è poco pragmatico, in quanto non è possibile a priori valutare la probabilità di scoprire errore da parte di un test case, ma possiamo affermare che si deve sempre tendere ad una produzione limitata il più possibile di test case di qualità, al fine di massimizzare il rapporto tra gli errori scoperti ed i test utilizzati.

Ovviamente non sempre è possibile riuscire a rispettare tutti i principi guida elencati, per cui si cerca, all'interno delle metodologie, di seguire il più possibile le linee guida che essi tracciano.

4. FUNZIONALITA' TESTATE E NON

Le componenti prese in considerazione nella fase di testing rappresentano la maggior parte delle funzionalità di TUTTO-ELETTRONICA. In particolare saranno testate:

- Gestione autenticazione : sarà testata la funzionalità di log-in e registrazione dei vari attori del sistema.

- Gestione prodotti: saranno testate le funzionalità di prenotazione e ricerca di un prodotto.

- Gestione riparazioni: sarà testata la funzionalità per la prenotazione di una riparazione

- Gestione gestore: sarà testata la funzionalità per il caricamento dei vari prodotti

Tutte queste funzionalità richiedono meccanismi dinamici che in base a dati persistenti permettono le diverse operazioni attraverso elaborazioni server.

Andiamo a testare i requisiti funzionali con priorità alta tralasciando quelli con priorità medio-bassa.

PASS/FAIL CRITERIA

La fase di testing necessita di criteri formali per la determinazione del successo o dell'insuccesso di un determinato test. I dati di ogni input relativi ai test verranno divisi in classi di equivalenza: un input appartenente ad una classe specifica supera il test se l'output ottenuto corrisponde ai risultati attesi, specificati nell'oracolo.

PASS CRITERIA

I pass criteria determinano l'insuccesso del test e quindi la correttezza del comportamento della componente testata. I pass criteria per la fase di testing del sistema TUTTO-ELETTRONICA sono essenzialmente raggruppati in due categorie principali, che possono essere raffinate qualora il test necessiti di raffinamento. Le categorie individuate sono le seguenti:

- Comportamento atteso:
 - o L'oracolo rileva che il comportamento della componente rispetti il comportamento atteso
- Nessun errore rilevato dalla componente:
 - o La componente non genera alcuna eccezione non prevista.



FAIL CRITERIA

I fail criteria determinano il successo del test e quindi la presenza di errori nel comportamento della componente testata. I fail criteria per la fase di testing del sistema TUTTO-ELETTRONICA sono essenzialmente raggruppati in 2 categorie principali che possono essere raffinate nei casi specifici. Le categorie individuate sono le seguenti:

- Errore rilevato dalla componente:
 - o La componente genera una eccezione non prevista.
- Comportamento non atteso:
 - o L'oracolo rileva che il comportamento della componente si discosti dal comportamento atteso rispetto agli input.

APPROCCI

L'approccio alla fase di testing si compone di tre fasi:

- Testing di unità, che controlla i singoli componenti (classi, metodi)
- Testing di integrazione, che testa l'integrazione dei vari sottosistemi
- Testing funzionale, che verifica la funzionalità dell'intero sistema assemblato.

TESTING DELLE UNITA'

Con il testing di unità sarà effettuato un controllo delle varie classi e metodi del sistema, quindi saranno ricercate le condizioni di fallimento evidenziando gli errori. Il testing di unità sarà eseguito utilizzando il framework JUnit. In particolare, per ogni classe che esegue operazioni complesse, sarà sviluppata la relativa classe JUnit.

TESTING DI INTEGRAZIONE

Con il testing di integrazione si effettuerà un controllo sull'integrazione delle varie componenti del sistema. Si adotterà una strategia di tipo "Bottom-up".

Per effettuare questi test di integrazione, spesso sarà necessario l'utilizzo di driver, dato che tale strategia va ad integrare man mano i sottosistemi partendo dal layer che si trova più in basso nella scala gerarchica.

TESTING DI SISTEMA

Nella sessione di testing di TuttoElettronica sarà utilizzato un approccio di tipo "BLACK BOX", andando a utilizzare la strategia "category partition", il quale prevede che i test siano effettuati ad un livello di astrazione più alto, in modo da non entrare nei dettagli del codice, ma basandosi sulle specifiche delle funzionalità da testare. Il motivo di tale scelta è quella di selezionare il giusto numero di casi di test, esplicitando le relazioni tra le variabili da testare, evitando di selezionare casi di test inutili. Il sistema viene scomposto in funzionalità; Le funzionalità vengono scomposte in parametri; Ogni parametro ha delle categorie, che sono delle proprietà o delle caratteristiche della funzionalità.



SOSPENSIONE E RIPRISTINO

La fase di testing del sistema TuttoElettronica può essere interrotta e ripresa più volte, se il fine ultimo è di rendere il sistema corretto e ogni funzionalità completa. Di seguito sono riportati i criteri secondo cui è necessario sospendere la fase di testing e le modalità secondo cui poi deve essere ripresa.

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.

Questi criteri comprendono tutti gli errori che hanno un impatto dannoso sul progresso dell'attività di testing, come il crash del database, il crash del sistema operativo, il fallimento di intere funzionalità ed anche problemi relativi all'ambiente di sviluppo del testing.

CRITERI DI RIPRISTINO

La ripresa del test avviene soltanto quando tutti i problemi relativi alla sospensione del test sono stati risolti. L'attività di testing riprenderà a partire dal test case che ha causato la sospensione.

5.STRUMENTI PER IL TESTING

Gli strumenti per il testing verranno divisi in due tipi differenti:

- Strumenti hardware
- Strumenti software

5.1 STRUMENTI HARDWARE

Il sistema sarà testato su un personal computer con la necessaria dotazione software. Non sono richieste particolari specifiche hardware se non quelle adatte a far funzionare il DBMS server e il Web server apache.

5.2 STRUMENTI SOFTWARE

Per i test delle unità e di integrazione si cercherà di adoperare JUnit. JUnit è un framework interamente sviluppato in Java Open-Source. Fornisce un insieme di API che assistono gli sviluppatori nel crearti in modo semplice ed automatico dei test per il proprio software. Permette di separare i test dal codice.

Fornisce una struttura di asserzioni per confrontare i risultati attesi (oracolo) con quelli ottenuti.

Fornisce una interfaccia grafica molto elementare per la valutazione dei risultati dei test: barra verde se i test case in esecuzione falliscono, barra rossa se qualche test case della test suite fallisce.

Consente di definire:

- Test case: singolo test su di una specifica funzionalità (un metodo, una classe, etc.).
- Test suite: una collezione di test case che possono essere raggruppati in base a caratteristiche omogenee e possono essere eseguite in blocco.

In particolare i software che si utilizzeranno per l'attività di testing sono i seguenti:

- JUnit



- MySQL
- Apache Tomcat 8 o superiore
- Java Runtime Environment (JRE) 7 o superiore
- Java Development kit (JDK) 7 o superiore
- Java Enterprise Edition (J2EE)
- Selenium IDE per l'esecuzione dei test case;
- Browser (Google Chrome, FirefoxMozilla, Safari, Opera, Internet Explorer, etc.).

6. TEST CASE

I test case sono mirati a scoprire eventuali malfunzionamenti o comportamenti errati da parte del sistema. Per fare ciò è necessario testare il sistema su diverse istanze di input, ognuna diretta a testare comportamenti del sistema in determinate condizioni, un tipo o classe di input piuttosto di un'altra. Il testing viene quindi strutturato in base alle funzionalità fornite dal sistema. Per ognuna di queste verranno fornite delle istanze di input che costituiscono i test case dei vari scenari. I test case potranno appartenere a tre diverse categorie di input in base al tipo di dati da cui sono costituiti relativamente allo use case in esame che possono essere:

- Grammaticalmente non validi: fanno parte di questa categoria gli input che contengono caratteri non validi, che sono a loro volta suddivisi in classi di equivalenza.
- Logicamente non validi: fanno parte di questa categoria i dati che potrebbero essere grammaticalmente validi ma che nel contesto non lo sono, ad esempio in un form che richiede l'inserimento di due numeri per la definizione di un range di valori, due input logicamente non validi possono essere un lowerbound (il valore inferiore del range) più alto dell'upperbound (valori superiori del range).
- Validi: fanno parte di questa categoria i dati validi, cioè che sono grammaticalmente validi e hanno senso nel contesto in cui sono utilizzati.

Quindi per ogni funzionalità verranno presentati un insieme di test. Ognuno di questi insiemi è diviso a sua volta in tre sottoinsiemi:

- Input grammaticalmente non validi, ma logicamente validi
- Input logicamente non validi, ma grammaticalmente validi
- Input grammaticalmente e logicamente validi.

6.1 REGISTRAZIONE

CLASSI DI EQUIVALENZA

INPUT	Nome, cognome	
CLASSI VALIDE	C_0001	Stringa alfabetica
CLASSI NON VALIDE	C_0002	Stringa contiene numeri
	C_0003	Stringa minore di 1 carattere.

INPUT	Indirizzo	
CLASSI VALIDE	C_0004	Stringa alfanumerica con lunghezza compresa tra 1 e 40 caratteri.
CLASSI NON VALIDE	C_0005	Stringa maggiore di 40 caratteri.
	C_0006	Stringa alfanumerica che non contiene almeno un carattere e un numero



INPUT	Codice fiscale	
CLASSI VALIDE	C_0007	Stringa alfanumerica con lunghezza fissa di 16 caratteri.
CLASSI NON VALIDE	C_0008	Stringa alfanumerica minore di 16 caratteri.
	C_0009	Stringa alfanumerica maggiore di 16 caratteri.
	C_0010	Stringa alfanumerica discorde con la forma standard di un codice fiscale.

INPUT	e-mail	
CLASSI VALIDE	C_0011	Stringa alfanumerica con formato "nomeUtente"@provider"
CLASSI NON VALIDE	C_0012	Stringa alfanumerica discorde dal formato.

INPUT	Password	
CLASSI VALIDE	C_0013	Stringa alfanumerica almeno di 5 caratteri
CLASSI NON VALIDE	C_0014	Stringa alfanumerica minore di 5 caratteri.
	C_0015	Stringa alfanumerica vuota

INPUT	Username	
CLASSI VALIDE	C_0016	Stringa alfanumerica.
CLASSI NON VALIDE	C_0017	Stringa alfanumerica vuota.
	C_0018	Stringa alfanumerica che non contiene almeno un carattere e un numero

6.1.1 FORMAL TEST SPECIFICATION

INPUT	Nome, cognome	
ID_CODICE	SPECIFICA FORMALE	
C_0001	[propertynome_cognome_indirizzo OK]	
C_0002	[errore]	
C_0003	[errore]	

INPUT	Indirizzo	
ID_CODICE	SPECIFICA FORMALE	
C_0004	[propertynome_cognome_indirizzo OK]	
C_0005	[errore]	
C_0006	[errore]	

INPUT	Codice fiscale	
ID_CODICE	SPECIFICA FORMALE	
C_0007	[propertycodice_fiscaleOK]	
C_0008	[errore]	
C_0009	[errore]	
C_0010	[errore]	

INPUT	Email	
-------	-------	--



ID_CODICE	SPECIFICA FORMALE
C_0011	[propertyEmailOK]
C_0012	[errore]

INPUT	Password
ID_CODICE	SPECIFICA FORMALE
C_0013	[propertyPasswordOK]
C_0014	[errore]
C_0015	[errore]

INPUT	Username
ID_CODICE	SPECIFICA FORMALE
C_0016	[propertyusernameOK]
C_0017	[errore]
C_0018	[errore]

TEST CASE

TEST CASE	TC_Registrazione_01	
PARAMETRO	SCelta	VALORE
Codice fiscale	C_0007	SSSGPP95E10H703E
Nome	C_0001	Raffaele
Cognome	C_0001	Villani
Email	C_0011	giuseppe@hotmail.it
Password	C_0013	Universita95
Indirizzo	C_0004	Nocera Via Iroma 105
Username	C_0016	Raff97
TEST CASE	TC_Registrazione_02	
PARAMETRO	SCelta	VALORE
Codice fiscale	C_0010	SSSGPP95E10H70AE
Nome	C_0001	Raffaele
Cognome	C_0001	Villani
Email	C_0011	giuseppe@hotmail.it
Password	C_0013	Universita95
Indirizzo	C_0004	Nocera Via Iroma 105
Username	C_0016	Raff97

TEST CASE	TC_Registrazione_03	
PARAMETRO	SCelta	VALORE
Codice fiscale	C_0009	SSSGPP95E10H70AEFDSSFD
Nome	C_0001	Raffaele
Cognome	C_0001	Villani
Email	C_0011	giuseppe@hotmail.it
Password	C_0013	Universita95
Indirizzo	C_0004	Nocera Via Iroma 105



Username	C_0016	Raff97
----------	--------	--------

TEST CASE	TC_Registrazione_04	
PARAMETRO	SCelta	VALORE
Codice fiscale	C_0008	SSSGPP95
Nome	C_0001	Raffaele
Cognome	C_0001	Villani
Email	C_0011	giuseppe@hotmail.it
Password	C_0013	Universita95
Indirizzo	C_0004	Nocera Via Iroma 105
Username	C_0016	Raff97

TEST CASE	TC_Registrazione_05	
PARAMETRO	SCelta	VALORE
Codice fiscale	C_0007	SSSGPP95E10H703E
Nome	C_0003	
Cognome	C_0001	Villani
Email	C_0011	giuseppe@hotmail.it
Password	C_0013	Universita95
Indirizzo	C_0004	Nocera Via Iroma 105
Username	C_0016	Raff97

TEST CASE	TC_Registrazione_06	
PARAMETRO	SCelta	VALORE
Codice fiscale	C_0007	SSSGPP95E10H703E
Nome	C_0001	Raffaele
Cognome	C_0003	
Email	C_0011	giuseppe@hotmail.it
Password	C_0013	Universita95
Indirizzo	C_0004	Nocera Via Iroma 105
Username	C_0016	Raff97

TEST CASE	TC_Registrazione_07	
PARAMETRO	SCelta	VALORE
Codice fiscale	C_0007	SSSGPP95E10H703E
Nome	C_0002	Raffa5f
Cognome	C_0001	Villani
Email	C_0011	giuseppe@hotmail.it
Password	C_0013	Universita95
Indirizzo	C_0004	Nocera Via Iroma 105
Username	C_0016	Raff97



TEST CASE	TC_Registrazione_08	
PARAMETRO	SCELTA	VALORE
Codice fiscale	C_0007	SSSGPP95E10H703E
Nome	C_0001	Raffaele
Cognome	C_0002	Villani65
Email	C_0011	giuseppe@hotmail.it
Password	C_0013	Universita95
Indirizzo	C_0004	Nocera Via Iroma 105
Username	C_0016	Raff97

TEST CASE	TC_Registrazione_09	
PARAMETRO	SCELTA	VALORE
Codice fiscale	C_0007	SSSGPP95E10H703E
Nome	C_0001	Raffaele
Cognome	C_0001	Villani
Email	C_0011	giuseppe@hotmail.it
Password	C_0013	Università95
Indirizzo	C_0005	
Username	C_0016	Raff97

TEST CASE	TC_Registrazione_10	
PARAMETRO	SCELTA	VALORE
Codice fiscale	C_0007	SSSGPP95E10H703E
Nome	C_0001	Raffaele
Cognome	C_0001	Villani
Email	C_0011	giuseppe@hotmail.it
Password	C_0014	12345
Indirizzo	C_0006	Nocera Via Iroma
Username	C_0016	Raff97

TEST CASE	TC_Registrazione_11	
PARAMETRO	SCELTA	VALORE
Codice fiscale	C_0007	SSSGPP95E10H703E
Nome	C_0001	Raffaele
Cognome	C_0001	Villani
Email	C_0011	giuseppe@hotmail.it
Password	C_0015	
Indirizzo	C_0004	Nocera Via Iroma 105
Username	C_0016	Raff97

TEST CASE	TC_Registrazione_12	
PARAMETRO	SCELTA	VALORE
Codice fiscale	C_0007	SSSGPP95E10H703E
Nome	C_0001	Raffaele



Cognome	C_0001	Villani
Email	C_0011	giuseppe@hotmail.it
Password	C_0014	asd
Indirizzo	C_0004	Nocera Via Iroma 105
Username	C_0016	Raff97

TEST CASE	TC_Registrazione_13	
PARAMETRO	SCELTA	VALORE
Codice fiscale	C_0007	SSSGPP95E10H703E
Nome	C_0001	Raffaele
Cognome	C_0001	Villani
Email	C_0012	giuseppe@hotmail
Password	C_0013	Universita95
Indirizzo	C_0004	Nocera Via Iroma 105
Username	C_0016	Raff97

TEST CASE	TC_Registrazione_14	
PARAMETRO	SCELTA	VALORE
Codice fiscale	C_0007	SSSGPP95E10H703E
Nome	C_0001	Raffaele
Cognome	C_0001	Villani
Email	C_0011	giuseppe@hotmail.it
Password	C_0013	Universita95
Indirizzo	C_0004	Nocera Via Iroma 105
Username	C_0017	

TEST CASE	TC_Registrazione_15	
PARAMETRO	SCELTA	VALORE
Codice fiscale	C_0007	SSSGPP95E10H703E
Nome	C_0001	Raffaele
Cognome	C_0001	Villani
Email	C_0011	giuseppe@hotmail.it
Password	C_0013	12346
Indirizzo	C_0004	Nocera Via Iroma 105
Username	C_0018	Raff

6.2 LOGIN

CLASSI DI EQUIVALENZA

INPUT	Username	
CLASSI VALIDE	C_0027	Stringa alfanumerica.
CLASSI NON VALIDE	C_0028	Stringa alfanumerica vuota.
	C_0029	Stringa alfanumerica che non contiene almeno un carattere e un numero

INPUT	Password	
CLASSI VALIDE	C_0030	Stringa alfanumerica almeno di 5 caratteri.



CLASSI NON VALIDE	C_0031	Stringa alfanumerica minore di 5 caratteri.
	C_0032	Stringa alfanumerica vuota.

6.2.1 FORMAL TEST SPECIFICATION

INPUT	Username	
ID_CODICE	SPECIFICA FORMALE	
C_0027	[propertyusernameOK]	
C_0028	[errore]	
C_0029	[errore]	

INPUT	Password	
ID_CODICE	SPECIFICA FORMALE	
C_0030	[propertyPasswordOK]	
C_0031	[errore]	
C_0032	[errore]	

TEST CASE

TEST CASE	TC_Login_01	
PARAMETRO	SCelta	VALORE
Username	C_0027	admin
Password	C_0030	123456

TEST CASE	TC_Login_02	
PARAMETRO	SCelta	VALORE
Username	C_0028	
Password	C_0030	Babbeo12

TEST CASE	TC_Login_03	
PARAMETRO	SCelta	VALORE
Username	C_0029	Raff
Password	C_0030	Babbeo12

TEST CASE	TC_Login_04	
PARAMETRO	SCelta	VALORE
Username	C_0027	Gae1998
Password	C_0031	123

TEST CASE	TC_Login_05	
PARAMETRO	SCelta	VALORE
Username	C_0027	Raff97
Password	C_0032	



6.3 CARICAMENTO PRODOTTO (GESTORE PRODOTTI)

CLASSI DI EQUIVALENZA

INPUT	Nome	
CLASSI VALIDE	C_0044	Stringa alfanumerica.
CLASSI NON VALIDE	C_0045	Stringa alfanumerica che non contiene almeno un carattere e un numero
	C_0046	Stringa alfanumerica vuota.

INPUT	Marchio	
CLASSI VALIDE	C_0047	Stringa alfanumerica.
CLASSI NON VALIDE	C_0048	Stringa alfanumerica che non contiene almeno un carattere e un numero
	C_0049	Stringa alfanumerica vuota.

INPUT	Categoria	
CLASSI VALIDE	C_0050	Stringa alfanumerica.
CLASSI NON VALIDE	C_0051	Stringa alfanumerica che non contiene almeno un carattere e un numero
	C_0052	Stringa alfanumerica vuota.

INPUT	Prezzo	
CLASSI VALIDE	C_0053	Numero decimale con due numeri dopo la virgola.
CLASSI NON VALIDE	C_0054	Carattere alfabetico all'interno del prezzo.

INPUT	Descrizione	
CLASSI VALIDE	C_0057	Stringa alfanumerica di lunghezza arbitraria.
CLASSI NON VALIDE	C_0058	Stringa alfanumerica vuota.

INPUT	Quantità	
CLASSI VALIDE	C_0059	Numero intero maggiore o uguale a 0.
CLASSI NON VALIDE	C_0060	Numero intero minore di 0.
	C_0061	Carattere alfabetico all'interno della quantità.

6.3.1 FORMAL TEST SPECIFICATION

INPUT	Nome	
ID_CODICE	SPECIFICA FORMALE	
C_0044	[propertycodice_pezzoOK]	
C_0045	[errore]	
C_0046	[errore]	



INPUT	Marchio	
ID_CODICE	SPECIFICA FORMALE	
C_0047	[propertymarchioOK]	
C_0048	[errore]	
C_0049	[errore]	

INPUT	Categoria	
ID_CODICE	SPECIFICA FORMALE	
C_0050	[propertymodelloOK]	
C_0051	[errore]	
C_0052	[errore]	

INPUT	Prezzo	
ID_CODICE	SPECIFICA FORMALE	
C_0053	[propertyPrezzoOK]	
C_0054	[errore]	

INPUT	Descrizione	
ID_CODICE	SPECIFICA FORMALE	
C_0057	[propertyDescrizioneOK]	
C_0058	[errore]	

INPUT	Quantità	
ID_CODICE	SPECIFICA FORMALE	
C_0059	[propertyQuantitàOK]	
C_0060	[errore]	
C_0061	[errore]	

TEST CASE

TEST CASE	TC_Caricamento_01	
PARAMETRO	SCELTA	VALORE
Nome	C_0044	Batteria 12A
Marchio	C_0047	Duracell
Categoria	C_0050	batterie
Prezzo	C_0053	2.00
Descrizione	C_0057	Batteria per telecomandi
Quantità	C_0059	10

TEST CASE	TC_Caricamento_02	
PARAMETRO	SCELTA	VALORE
Nome	C_0044	Arduino shield
Marchio	C_0047	arduino
Categoria	C_0052	



Prezzo	C_0053	18.00
Descrizione	C_0058	Scheda arduino
Quantità	C_0059	10

TEST CASE	TC_Caricamento_03	
PARAMETRO	SCELTA	VALORE
Nome	C_0044	Arduino shield
Marchio	C_0047	Arduino
Categoria	C_0052	Arduino
Prezzo	C_0053	18.00
Descrizione	C_0058	
Quantità	C_0059	10

TEST CASE	TC_Caricamento_04	
PARAMETRO	SCELTA	VALORE
Nome	C_0044	Arduino shield
Marchio	C_0047	Arduino
Categoria	C_0051	0
Prezzo	C_0053	18.00
Descrizione	C_0058	
Quantità	C_0059	10

TEST CASE	TC_Caricamento_05	
PARAMETRO	SCELTA	VALORE
Nome	C_0046	Batteria 12A
Marchio	C_0047	Phonocar
Categoria	C_0050	Accessori
Prezzo	C_0053	10.15
Descrizione	C_0057	Mascherina auto
Quantità	C_0060	-1

TEST CASE	TC_Caricamento_06	
PARAMETRO	SCELTA	VALORE
Nome	C_0046	Batteria 12A
Marchio	C_0047	Phonocar
Categoria	C_0050	Accessori
Prezzo	C_0053	10.15
Descrizione	C_0057	Mascherina auto
Quantità	C_0060	1ab

TEST CASE	TC_Caricamento_07	
PARAMETRO	SCELTA	VALORE
Nome	C_0045	
Marchio	C_0047	Phonocar
Categoria	C_0050	Accessori
Prezzo	C_0053	10.15
Descrizione	C_0057	Mascherina auto
Quantità	C_0060	-1



TEST CASE	TC_Caricamento_08	
PARAMETRO	SCELTA	VALORE
Nome	C_0045	0
Marchio	C_0048	0
Categoria	C_0050	Illuminazione
Prezzo	C_0053	15.00
Descrizione	C_0057	Lampadina 16A
Quantità	C_0059	10

TEST CASE	TC_Caricamento_09	
PARAMETRO	SCELTA	VALORE
Nome	C_0045	0
Marchio	C_0048	
Categoria	C_0050	Illuminazione
Prezzo	C_0053	15.00
Descrizione	C_0057	Lampadina 16A
Quantità	C_0059	10

TEST CASE	TC_Caricamento_010	
PARAMETRO	SCELTA	VALORE
Nome	C_0046	Lamp
Marchio	C_0047	
Categoria	C_0052	
Prezzo	C_0053	14.60
Descrizione	C_0058	Lampadina 16A
Quantità	C_0061	10

TEST CASE	TC_Caricamento_11	
PARAMETRO	SCELTA	VALORE
Nome	C_0046	Lamp
Marchio	C_0047	0
Categoria	C_0052	illuminazione
Prezzo	C_0053	
Descrizione	C_0058	Lampadina 16A
Quantità	C_0061	10

TEST CASE	TC_Caricamento_12	
PARAMETRO	SCELTA	VALORE
Nome	C_0046	Batteria 12A
Marchio	C_0047	Duracell
Categoria	C_0052	Batterie
Prezzo	C_0054	14.0a
Descrizione	C_0058	Batteria per telecomandi
Quantità	C_0061	10



6.4 RICERCA PRODOTTO PER NOME

CLASSI DI EQUIVALENZA

INPUT	Nome	
CLASSI VALIDE	C_0035	Stringa alfanumerica.
CLASSI NON VALIDE	C_0036	Stringa alfanumerica vuota.
	C_0037	Stringa alfanumerica non presente nel sistema.

6.4.1 FORMAL TEST SPECIFICATION

INPUT	Nome	
ID_CODICE	SPECIFICA FORMALE	
C_0035	[propertynomeOK]	
C_0036	[errore]	
C_0037	[errore]	

TEST CASE

TEST CASE	TC_Ricerca_01	
PARAMETRO	SCELTA	VALORE
Nome	C_0035	Arduino UNO

TEST CASE	TC_Ricerca_02	
PARAMETRO	SCELTA	VALORE
Nome	C_0036	

TEST CASE	TC_Ricerca_03	
PARAMETRO	SCELTA	VALORE
Nome	C_0037	Dwfwioonef

6.5 RICERCA PRODOTTO PER CATEGORIA

CLASSI DI EQUIVALENZA

INPUT	Categoria	
CLASSI VALIDE	C_0038	Stringa alfanumerica.
CLASSI NON VALIDE	C_0039	Stringa alfanumerica non presente nel sistema.
	C_0040	Stringa alfanumerica vuota.



6.5.1 FORMAL TEST SPECIFICATION

INPUT	Categoria	
ID_CODICE	SPECIFICA FORMALE	
C_0038	[propertycategoriaOK]	
C_0039	[errore]	
C_0040	[errore]	

TEST CASE

TEST CASE	TC_Ricerca_04	
PARAMETRO	SCELTA	VALORE
Categoria	C_0038	Arduino

TEST CASE	TC_Ricerca_05	
PARAMETRO	SCELTA	VALORE
Categoria	C_0039	edcferverev

TEST CASE	TC_Ricerca_06	
PARAMETRO	SCELTA	VALORE
Categoria	C_0040	

6.6 PRENOTAZIONE RIPARAZIONE

CASI DI EQUIVALENZA

INPUT	Descrizione	
CLASSI VALIDE	C_0062	Stringa alfanumerica.
CLASSI NON VALIDE	C_0064	Stringa alfanumerica vuota.

INPUT	Categoria	
CLASSI VALIDE	C_0065	Stringa alfanumerica.
CLASSI NON VALIDE	C_0067	Stringa alfanumerica vuota.

6.6.1 FORMAL TEST SPECIFICATION

INPUT	Descrizione	
ID_CODICE	SPECIFICA FORMALE	
C_0062	[propertycodice_pezzoOK]	
C_0064	[errore]	



INPUT	Categoria	
ID_CODICE	SPECIFICA FORMALE	
C_0065	[propertymarchioOK]	
C_0067	[errore]	

TEST CASE

TEST CASE	TC_PrenotazioneRiparazione_01	
PARAMETRO	SCELTA	VALORE
Descrizione	C_0062	Tv rotta
Categoria	C_0065	televisori

TEST CASE	TC_PrenotazioneRiparazione_02	
PARAMETRO	SCELTA	VALORE
Descrizione	C_0062	Radio rotta
Categoria	C_0067	

TEST CASE	TC_PrenotazioneRiparazione_03	
PARAMETRO	SCELTA	VALORE
Descrizione	C_0064	
Categoria	C_0067	

TEST CASE	TC_PrenotazioneRiparazione_04	
PARAMETRO	SCELTA	VALORE
Descrizione	C_0064	
Categoria	C_0065	televisori

7 TESTING SCHEDULE

Di seguito sono elencate la gestione dei rischi che occorre seguire durante la fase di testing, l'organizzazione delle attività di testing e la schedulazione delle attività durante il periodo stabilito per il testing.

7.1 GESTIONE DEI RISCHI

I possibili rischi generati dalle attività di testing sono stati minimizzati diminuendo le componenti del sistema da implementare e, quindi, testare. Inoltre, effettuando un testing di tipo funzionale viene limitato lo sviluppo di stub e driver per il testing delle singole componenti e quindi l'introduzione di nuovi errori nel nuovo codice di cui sarebbero composti. D'altro canto il testing di funzionalità rallenta l'individuazione di errori qualora un caso di test avesse esito positivo, poiché l'utilizzo di più componenti per il test di una singola funzionalità viene eseguita ma non in modo



completamente corretto. Qualora la fase di testing evidenziasse un numero di errori maggiore rispetto alla media attesa, viene pianificato un impegno maggiore dei membri del team sulle attività di testing ed in casi estremi l'abbandono delle altre attività finchè errori gravi (funzionalità non corretta, risultati errati, modifiche apportate in modo errato) non vengano risolti.

7.2 ORGANIZZAZIONE DELLE ATTIVITA'

Le attività di testing devono svolgersi sulle singole funzionalità divise nei livelli di suddivisione del sistema, rispettando le direttive indicate dal documento di system design.