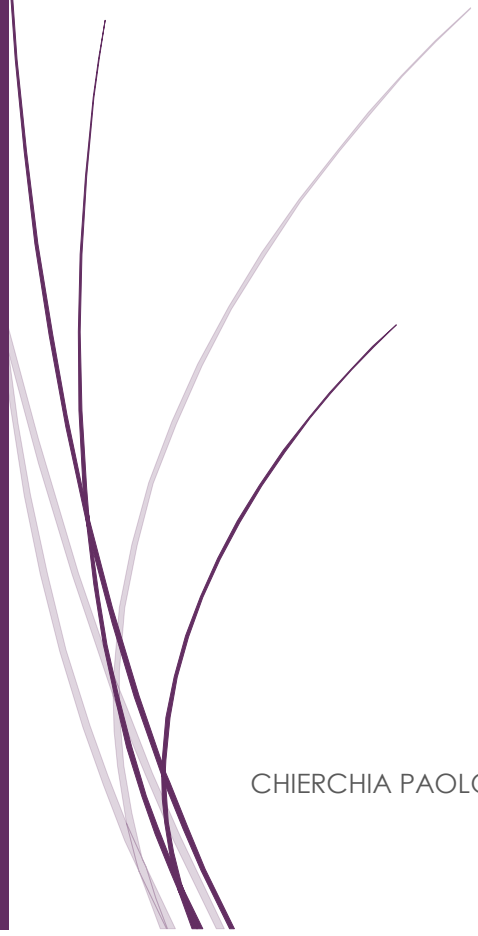
2024/25

TSR

Test Summary Report Digitronics



Test Summary Report

CHIERCHIA PAOLO GIOVANNI, DI MICCO VINCENZO, ZOCCOLA DOMENICO

Sommario

Revision History

Data	Versione	Descrizione	Autori
15/01/2025	0.1	Stesura documento	Tutto il team
16/01/2025	1.0	Revisione finale	Tutto il team

Team Members

Nome	Ruolo nel Progetto	Acronimo	Informazioni di contatto
Giovanni Paolo Chierchia	Team Member	GPC	g.chierchia8@studenti.unisa.it
Vincenzo Di Micco	Team Member	VDM	v.dimicco4@studenti.unisa.it
Domenico Zoccola	Team Member	DZ	d.zoccola3@studenti.unisa.it

1. Introduzione

Digitronics è un e-commerce che si occupa della vendita di dispositivi elettronici come tablet, smartphone, smartwatch e relativi accessori.

Il Test Plan Digitronics ha lo scopo di descrivere e analizzare le attività di testing per la piattaforma, assicurando che ogni componente funzioni correttamente.
Il documento mostra le strategie di testing adottate, le funzionalità oggetto di verifica e gli strumenti selezionati per l'identificazione dei difetti, al fine di consegnare al cliente finale una piattaforma completamente funzionante e priva di anomalie.
Sono state pianificate le attività di testing per le seguenti funzionalità:

- Aggiunta recensione

- Modifica prodotto
- Checkout prodotti

2. Relazione con altri documenti

Relazione con il Test Plan
Il Test Summary Report si basa sulle attività di testing descritte nel Test Plan.

Relazione con il Test Case Specification
Il Test Summary Report contiene un riassunto dell'esecuzione dei test di sistema definiti nella Test Case Specification.

Relazione con il Test Incident Report
Il Test Summary Report include un riassunto dei risultati dell'esecuzione riportati nel Test Incident Report.

3. Testing unitario e di integrazione

I test di unità e di integrazione sono stati organizzati in due distinte classi di test. Prima di effettuare una pull request, ogni membro del gruppo doveva verificare che tutti i test scritti avessero esito positivo. Una volta aperta la pull request, **Travis CI** eseguiva automaticamente tutti i test per verificarne il successo. Se uno o più test non superavano il controllo, il merge veniva bloccato. In tal caso, spettava al membro del gruppo che aveva aperto la pull request apportare le necessarie modifiche per risolvere i problemi.
Per verificare le metriche di coverage di test è stato utilizzato lo strumento di **Code Coverage integrato in Intellij**.
Ecco le coverage delle classi che si occupano di realizzare le funzionalità che nel test plan si era deciso di implementare:

- Classe GestioneOrdineServlet.java

Branche Coverage	Line Coverage
85%	100%

- Classe GestioneOrdineService.java

Branche Coverage	Line Coverage
100%	100%

- Classe ModificaProdottoServlet.java

Branche Coverage	Line Coverage
81%	100%

- Classe GestioneProdottoService.java

Branche Coverage	Line Coverage
70%	100%

- Classe AggiungiRecensioneServletTest.java

Branche Coverage	Line Coverage
75%	97%

- Classe RecensioneServiceImplTest.java

Branche Coverage	Line Coverage
100%	100%

4. Testing di sistema

Per quanto riguarda il test di sistema, sono state definite 3 test suite utilizzando il tool **Selenium IDE** per Chrome. In particolare, è stata creata una test suite per ogni funzionalità da testare. Ogni esecuzione di una test suite ha richiesto il riavvio del sistema, a causa della necessità di aggiornare lo storage. Di seguito vengono riportati i risultati delle esecuzioni dei test.

Esecuzione	#Fallimenti	#Successi
Esecuzione 1 15/01/2025	1	31
Esecuzione 2 15/01/2025	0	32