



Laurea Triennale in informatica-Università di Salerno
Corso di Ingegneria del Software-Prof. C. Gravino



Test Plan

SushiStar

Riferimento

TCS

Versione

1.0

**Data
Destinatario**

17/11/2024
Prof. C. Gravino

**Presentato
da**

Antonio Avino, Antonio Bisogno,
Matteo Coraggio, Paolo Balestriere



Revision History

Data	Versione	Descrizione	Autori
27/12/2024	0.1	Stesura Test Plan	PB
29/12/2024	1.0	Revisione Test Plan	AA,AB,MC,PB

Team Members

Nome	Acronimo	Contatto
Antonio Avino	AA	a.avino29@studenti.unisa.it
Antonio Bisogno	AB	a.bisogno53@studenti.unisa.it
Paolo Balestriere	PB	p.balestriere@studenti.unisa.it
Matteo Coraggio	MC	m.coraggio8@studenti.unisa.it



Sommario

Revision History	2
Team Members	2
1 Introduzione	4
2 Relazione con altri documenti	4
3 Panoramica del sistema	4
4 Features da testare/non testare	5
5 Pass/fail criteria	5
6 Approccio	6
7 Sospensione e ripristino	6
8 Materiale di testing	6
9 Test case	6
10 Testing schedule	7



1 Introduzione

La piattaforma SushiStar ha come obiettivo quello di fornire ai clienti della catena di ristoranti SushiStar la possibilità di consultare il menu ed effettuare ordinazioni di prodotti tramite una applicazione web sempre accessibile.

Il documento di Test Plan descrive le attività di testing effettuate sul sistema SushiStar al fine di verificare il corretto funzionamento della piattaforma.

Nel documento sono quindi riportate le strategie di testing adottate, le funzionalità che saranno testate e gli strumenti utilizzati per rilevare eventuali errori presenti nel sistema.

In particolare, verranno analizzate le seguenti sezioni del sistema:

- Gestione Utente
- Gestione Navigazione Utente
- Gestione Prodotti

2 Relazione con altri documenti

Per l'individuazione di test case efficaci, si fa riferimento ad altri documenti prodotti.

RELAZIONI CON IL REQUIREMENTS ANALYSIS DOCUMENT (RAD)

I test case riportati nel Test Plan sono ideati in base ai requisiti funzionali e i requisiti non funzionali riportati nel RAD.

RELAZIONI CON IL SYSTEM DESIGN DOCUMENT (SDD)

I test case riportati nel Test Plan rispettano la suddivisione in sottosistemi presentata nell'SDD

3 Panoramica del sistema

Il sistema prodotto si basa su un'architettura three-tier.

La sezione di front-end e di generazione delle view utilizza il framework Angular.

La sezione di back-end utilizza il framework Java SPRING per implementare la logica applicativa.

Per la gestione dei dati persistenti saranno utilizzati:

- MySQL come software di gestione della base di dati
- AWS per l'utilizzo del database in fase di deployment e di produzione



- Spring JPA per il collegamento al database

4 Features da testare/non testare

Di seguito la lista di features per cui si effettuerà il testing per le varie gestioni:

- Gestione Utente
 - Recupero password
- Gestione Navigazione Utente
 - Ricerca di un prodotto nel menu
 - Acquisto di un prodotto
- Gestione Prodotti (Menu)
 - Aggiunta di un prodotto al menu

Si andranno dunque a testare funzionalità a cui è stata assegnata una priorità elevata durante la raccolta dei requisiti funzionali. Si è invece deciso di escludere dalla fase iniziale di testing le funzionalità con media o bassa priorità e tutte le funzionalità di visualizzazione dati che non prevedano input da parte degli utenti.

5 Pass/fail criteria

L'attività di testing ha come scopo l'individuazione di bug (errori) presenti nel sistema software. pertanto, un test ha un esito positivo se riesce ad individuare un errore.

La valutazione di un test si effettua tramite l'utilizzo di un oracolo, ovvero la raccolta di risultati attesi per quel test. Se il risultato ottenuto è diverso da quello previsto dell'oracolo, il test ha avuto successo. Viceversa, un test fallisce se il risultato ottenuto è uguale a quello previsto (contenuto nell'oracolo).

Di seguito sono elencati i criteri di validazione dell'attività di testing:

- Esecuzione dei Regression Test a seguito dell'introduzione di nuove caratteristiche o correzioni al sistema



6 Approccio

La fase di testing del sistema prevede l'esecuzione di testing di unità per i componenti utilizzati per realizzare le funzionalità elencate al punto 4 di questo documento.

I casi di test verranno progettati durante la fase di implementazione del sistema.

Testing di unità

L'approccio adottato per i test di unità sarà quello black-box: saranno quindi testati tutti i metodi delle classi implementate che richiedono dei parametri in input. Inoltre, i casi di test saranno documentati nel codice attraverso il framework JUnit.

Un altro strumento che verrà utilizzato è Maven per le build del progetto e l'esecuzione automatica dei test.

7 Sospensione e ripristino

Di seguito sono specificati i criteri di sospensione e di ripristino dell'attività di testing.

Criteri di sospensione

Il testing non verrà sospeso fino alla sua terminazione, tranne nel caso di individuazione di errori nella definizione dei test stessi.

Criteri di ripristino

Il testing verrà ripreso dopo aver risolto i fault individuati.

8 Materiale di testing

L'hardware necessario per l'attività di testing è un semplice computer connesso ad internet per poter interagire con la base di dati in cui sono memorizzati i dati persistenti.

9 Test case

Tutti i test case sono riportati nel documento Test Case Specification.



10 Testing schedule

Le attività di testing sono pianificate per essere eseguite in contemporanea con la fase di scrittura del codice, pertanto la progettazione dei casi di test avverrà subito dopo la terminazione dei documenti tecnici (RAD e SDD).

Una volta concluso lo sviluppo, tutti i test saranno rieseguiti per garantirne il corretto funzionamento e produrre i report finali.