



TEST PLAN

THE SPOON



Versione	0.8
Data	12/12/2023
Destinatario	Esame di Ingegneria del Software 2023/24
Presentato da	Alessandro Pascarella (0512109149) Vincenzo Catone (0512106704) Jacopo Gennaro Esposito (0512121285)



Revision History

Data	Versione	Descrizione	Autori
10/12/2023	0.1	Prima stesura	Alessandro Pascarella Vincenzo Catone Jacopo Gennaro Esposito
11/12/2023	0.2	Realizzazione dell'introduzione, relazione con altri documenti e la panoramica del sistema.	Alessandro Pascarella Vincenzo Catone Jacopo Gennaro Esposito
13/12/2023	0.3	Stesura delle features da testare e non testare, pass/fail di criteria e l'approccio	Alessandro Pascarella Vincenzo Catone Jacopo Gennaro Esposito
14/12/2023	0.4	Stesura della sospensione e ripristino, test cases	Alessandro Pascarella Vincenzo Catone Jacopo Gennaro Esposito
16/12/2023	0.5	Revisione	Alessandro Pascarella Vincenzo Catone Jacopo Gennaro Esposito
17/12/2023	0.6	Revisione	Alessandro Pascarella Vincenzo Catone Jacopo Gennaro Esposito
18/12/2023	0.7	Revisione	Alessandro Pascarella Vincenzo Catone Jacopo Gennaro Esposito
02/01/2024	0.8	Revisione	Alessandro Pascarella Vincenzo Catone Jacopo Gennaro Esposito



Team Members

Nome	Ruolo nel progetto	Acronimo	Informazioni di contatto
Alessandro Pascarella	Team Member	AP	a.pascarella9@studenti.unisa.it
Vincenzo Catone	Team Member	VC	v.catone@studenti.unisa.it
Jacopo Gennaro Esposito	Team Member	JE	j.esposito9@studenti.unisa.it



Sommario

1. Introduzione.....	5
2. Relazione con altri documenti.....	5
3. Panoramica del sistema.....	5
4. Features da testare/da non testare.....	6
5.Pass/Fail criteria.....	6
6. Approccio.....	7
7. Sospensione e ripristino.....	7
8. Test Cases.....	7
8.2 Gestione ristorante.....	10
8.3 Gestione prenotazione.....	11
8.4 Gestione ordini.....	12
8.4.1 Inserimento ordine(asporto).....	12
8.4.2 Inserimento ordine(tavolo).....	13



1. Introduzione

The Spoon nasce con l'obiettivo di rivoluzionare l'esperienza gastronomica, migliorando notevolmente sia il lato del consumatore che quello del gestore. Nell'era moderna, contrassegnata da ritmi sempre più frenetici, il tempo è diventato un bene prezioso. Troppo spesso, i clienti si trovano a passare lunghi periodi di attesa prima di potersi sedere in un ristorante o una pizzeria, solo per dover poi aspettare nuovamente per ordinare il proprio pasto. Questa situazione non solo genera insoddisfazione tra i clienti, ma comporta anche perdite economiche e stress per i gestori. Ecco perché nasce The Spoon, la piattaforma di riferimento per affrontare e risolvere tali problematiche. Gli utenti che scelgono di utilizzare The Spoon hanno la possibilità di prenotare facilmente un tavolo presso i ristoranti e le pizzerie aderenti al servizio, garantendo un arrivo puntuale e eliminando l'attesa e la frustrazione. Inoltre tale piattaforma potrà essere utilizzata anche direttamente all'interno dei ristoranti: grazie all'installazione di totem su ogni tavolo, i clienti potranno ordinare le loro pietanze preferite e riceverle da un cameriere nel minor tempo possibile. E se invece un cliente vuole ordinare del cibo da asporto? Nessun problema! Attraverso The Spoon anche questo è possibile.

2. Relazione con altri documenti

Per consentire una corretta identificazione dei test-case si riportano in questa sezione i riferimenti con altri documenti:

- **Relazioni con il Requirements Analysis Document (RAD)**

I test case individuati e pianificati nel Test Plan sono stati realizzati in riferimento ai requisiti funzionali e non funzionali descritti nel documento del Requirements Analysis Document.

- **Relazioni con il System Design Document (SDD)**

I test case pianificati nel documento del Test Plane devono procedere a rispettare la suddivisione dei documenti individuati nel documento System Design Document.

3. Panoramica del sistema

Il sistema proposto basa la sua architettura sul sistema three-tier, nello specifico usando Spring MVC e quindi un sistema Model View Control.

Verranno usati:

- **HTML5 e CSS3** per la gestione per la parte di front-end e la generazione delle view.
- Per la logica applicativa sarà utilizzato **Java Spring**
- Per la gestione del database sarà utilizzato **MySql**.



4. Features da testare/da non testare

Nella seguente sezione si riportano i features di cui si effettuerà il testing per le varie gestioni precedentemente individuate:

Gestione Utente:

- Registrazione ristorante
- Login

Gestione Ristorante:

- Inserimento ristorante

Gestione prenotazione:

- Creazione nuova prenotazione

Gestione ordini:

- Creazione nuovo ordine(asporto)
- Creazione nuovo ordine(tavolo)

Le funzionalità con una bassa o media priorità non saranno oggetti dell'attività di testing, così come non saranno considerate le funzionalità che non prevedano l'input manuale dell'utente e le funzionalità di semplice visualizzazione

5.Pass/Fail criteria

Le attività di testing sono finalizzate a identificare la presenza di errori nel sistema, con l'obiettivo di svolgere successive attività per procedere a rimuovere gli errori. I risultati dei casi di test vengono stabiliti attraverso un oracolo, definito come il risultato previsto dall'esecuzione sulla base dei requisiti. Un test può essere considerato valido se soddisfa tutti i criteri stabiliti nelle specifiche e produce un output conforme a quello previsto dall'oracolo. Al contrario, un test può essere considerato fallito se non soddisfa uno o più criteri o se produce un output diverso da quello previsto dall'oracolo. I criteri pass e fail sono estremamente importanti per garantire che il sistema sia funzionante e che rispetti i requisiti stabiliti. Tutti i test sono considerati validi se vengono soddisfatte tutte le seguenti restrizioni:

- Testare tutti i requisiti funzionali ad alta priorità;
- Eseguire test di regressione ogni volta che vengono aggiunte nuove funzionalità al sistema o vengono apportate modifiche alle funzionalità esistenti;



6. Approccio

La verifica del sistema è composta da tre fasi: test di unità, test di integrazione e test di sistema. Saranno progettati in ordine inverso rispetto a come sono stati descritti, ma verranno eseguiti nell'ordine appena menzionato. Prima della fase di implementazione del sistema, verranno creati i casi di test per il sistema, che saranno successivamente perfezionati durante la loro esecuzione.

7. Sospensione e ripristino

Il testing non sarà interrotto fino al suo completamento, anche in caso di rilevamento di un fallimento. Tuttavia, il testing potrebbe essere temporaneamente sospeso se si verifica un errore nella definizione di uno dei test stessi durante la sua esecuzione. In tal caso, sarà necessario correggere l'errore prima di poter riprendere il testing. È importante completare il testing per garantire che il sistema sia funzionante e che soddisfi i requisiti stabiliti. In caso di modifiche al sistema, la fase di testing verrà ripresa. I test case verranno nuovamente eseguiti per verificare che il sistema continui a funzionare correttamente anche dopo le modifiche apportate. È importante effettuare nuovamente i test per garantire che il sistema mantenga la sua funzionalità anche in seguito a eventuali cambiamenti.

8. Test Cases

L'approccio utilizzato per definire i test frame è il category partition. Al fine di minimizzare il numero di test case, gli input verranno partizionati in classi di equivalenza.

8.1 Gestione Utente

8.1.1 Registrazione Ristoratore

Parametro: Password	
Nome categoria	Scelte per la categoria
Lunghezza[LP]:	1. Lunghezza < 8 = false[error] 2. Lunghezza >= true [PROPERTY LP_OK]
Parametro: Nome	
Formato ^[A-zÀ-ù '-]{2,30}\$	
Nome Categoria	Scelte per la categoria
Formato[FNU]	1. Rispetta il formato = false[ERROR]



	2. Rispetta il formato = true[PROPERTY_FTP_OK]
Parametro : Nome	
Nome categoria	Scelta per la categoria
Lunghezza [LNU]	1. Lunghezza <= 2 && Lunghezza >= 30 = false [ERROR] 2. Lunghezza >= = true[PROPERTY_LNU_OK]
Parametro:Cognome	
Formato: ^[A-zÀ-ù ' -]{2,30}\$	
Nome Categoria	Scelte per la categoria
Formato [FCU]	1. Rispetta il formato = false [ERROR] 2. Rispetta il formato = true [PROPERTY_FCO_OK]
Parametro: Data di Nascita	
Nome Categoria	Scelte per la categoria
Data [DNN]	1. Data<DataCorrente=false[ERROR] 2. Data>=DataCorrente=true[PROPERTY_DNN_OK]
Parametro: Data di Nascita	
Nome Categoria	Scelte per la categoria
Correttezza [DNU]	1. Età < 18 = false[ERROR] 2. Età >= 18 = true [PROPERTY_DNU_OK]
Nome Categoria	Scelte per la categoria
Telefono [FT]	1. Rispetta il formato=false[ERROR] 2. Rispetta il formato=true[PROPERTY_FT_OK]
Parametro: Email	
Formato: ^[A-z0-9._%+-]+@[A-z0-9.-]+\.[A-z]{2,10}\$	
Nome Categoria	Scelte per la categoria
Formato [FEM]	1. Rispetta il formato=false[ERROR] 2. Rispetta il formato=true[PROPERTY_FEM OK]



Parametro: Email	
Nome Categoria	Scelte per la categoria
Correttezza [FMU]	<ol style="list-style-type: none">1. email = emailRegistrata = false[ERROR]2. email != emailRegistrata = true[PROPERTY_FMU_OK]

Test case	Test frame	Esito
1.1.1	LN1	Errato: nome troppo lungo
1.1.2	LN2, FEM1	Errato: formato email non valido
1.1.3	LN2, FEM2, LP1	Errato: password troppo corta
1.1.4	LN2, FEM2, LP2, FNU1	Errato: formato nome non valido
1.1.5	LN2, FEM2, LP2, FNU2, FCU1	Errato: formato cognome non valido
1.1.6	LN2, FEM2, LP2, FNU2, FCU2, FT1	Errato: formato numero di telefono errato
1.1.7	LN2, FEM2, LP2, FNU2, FCU2, FT2, FMU1	Errato: email già utilizzata
1.1.8	LN2, FEM2, LP2, FNU2, FCU2, FT2, FMU2, DNU1	Errato: età non valida
1.1.9	LN2, FEM2, LP2, FNU2, FCU2, FT2, FM2, DNU2	Corretto

8.1.2 Login

Parametro: Email	
Nome Categoria	Scelte per la categoria
Match [ME]	<ol style="list-style-type: none">1. match email = false[ERROR]2. match email = true [PROPERTY_ME_OK]
Parametro: Password	
Nome Categoria	Scelte per la categoria



Match [MP]	<ol style="list-style-type: none">1. match password = false[ERROR]2. match password = true [PROPERTY_MP_OK]
------------	--

Test case	Test frame	Esito
1.2.1	ME1	Errato: email errata
1.2.2	ME2, MP1	Errato: password errata
1.2.3	ME2, MP2	Corretto

8.2 Gestione ristorante

Parametro: Nome	
Nome Categoria	Scelte per la categoria
Lunghezza [LNR]	<ol style="list-style-type: none">2. Lunghezza >30=false[ERROR]3. Lunghezza <=30=true[PROPERTY_LNR_OK]
Parametro: Indirizzo (Via, NCivico, CAP, Provincia)	
Formato Via: Lunghezza <30 NCivico: Lunghezza <3 CAP: Lunghezza=5 Provincia: Lunghezza=2	
Nome categoria	Scelte per la categoria
Formato[FIR]	<ol style="list-style-type: none">1. Rispetta il formato = false[ERROR]2. Rispetta il formato= true[PROPERTY_FIR_OK]
Parametro: Telefono	
Formato: ^\d{10}\$	
Nome Categoria	Scelte per la categoria
Formato[FTR]	<ol style="list-style-type: none">1. Rispetta il formato = false[ERROR]2. Rispetta il formato= true[PROPERTY_FTR_OK]



Test case	Test frame	Esito
TC_2.1	LNR1,	Errato: Nome troppo lungo
TC_2.2	LNR2, FIR1	Errato: Indirizzo non valido
TC_2.3	LNR2, FIR2, FTR1	Errato: Telefono non valido
TC_2.4	LNR2, FIR2, FTR2	Corretto

8.3 Gestione prenotazione

Parametro: NPersone	
Nome categoria	Scelte per la categoria
Numero[NPP]	<ol style="list-style-type: none">1. numero<=0 OR numero>99=false[ERROR]2. numero>0 AND numero<99=true[PROPERTY_NPP_OK]
Parametro: Email	
Formato: ^[A-z0-9._%+-]+@[A-z0-9.-]+\.[A-z]{2,10}\$	
Nome categoria	Scelte per la categoria
Formato[FEP]:	<ol style="list-style-type: none">1. Rispetta il formato=false[ERROR]2. Rispetta il formato=true[PROPERTY_FEP_OK]
Parametro: Cellulare	
Formato ^\d{10}\$	
Nome Categoria	Scelte per la categoria
Formato[FTP]	<ol style="list-style-type: none">1. Rispetta il formato=false[ERROR]2. Rispetta il formato=true[PROPERTY_FTP_OK]
Parametro:Ora	
Nome Categoria	Scelte per la categoria



Correttezza[OP]	1. Ora non inserita= false[ERROR] 2. Ora inserita= true[PROPERTY_OP_OK]
Parametro: Data	
Nome Categoria	Scelte per la categoria
Data[DP]	1. Data<DataCorrente=false[ERROR] 2. Data>=DataCorrente=true[PROPERTY_DP_OK]

Test case	Test frame	Esito
TC_3.1	NPP1	Errato: Npersone non valido
TC_3.2	NPP2,FEP1	Errato: email non valida
TC_3.3	NPP2, FEP2, FTP1	Errato: cellulare non valido
TC_3.4	NPP2, FEP2, FTP2,vOP1	Errato: Ora non valida
TC_3.5	NPP2, FEP2, FTP2, OP2, DP1	Errato: Data non valida
TC_3.6	NPP2, FEP2, FTP2, OP2, DP2	Corretto

8.4 Gestione ordini

8.4.1 Inserimento ordine

Parametro: Prodotto	
Nome Categoria	Scelte per la categoria
Correttezza [CP]	1. Prodotto != Prodotto_Esistente[ERROR] 2. Prodotto == Prodotto esistente[PROPERTY_CQ_OK]
Parametro: Totale	
Nome categoria	Scelte per la categoria
Correttezza [CT]	1. Totale!=Totale_Atteso[ERROR] 2. Totale==Totale_Atteso[PROPERTY_OD_OK]



Test case	Test frame	Esito
TC_4.1	CP1	Errato: La quantità di prodotto aggiunta all'ordine risulta diversa da 1
TC_4.2	CP2, CT1	Errato: La somma del totale dell'ordine risulta diversa dalla somma prevista
TC_4.3	CP2, CT2	Corretto