

ANALISI DEI REQUISITI

RF = Requisiti Funzionali

RNF = Requisiti Non Funzionali

| ID | Requisito Funzionale |
|-----------------|--|
| RF-1.0.1 | Il sistema deve permettere l'accesso tramite email e password, reindirizzando automaticamente l'utente all'interfaccia corretta in base al suo ruolo (Studente/Esercente). |
| RF-1.0.2 | Il sistema deve impedire l'accesso se le credenziali non sono valide, fornendo un feedback visivo immediato. |
| RF-1.3.1 | Il sistema deve recuperare e visualizzare in una vista dedicata i dati anagrafici e di contatto attualmente associati all'account dell'utente loggato. |
| RF-1.3.2 | Il sistema deve rendere modificabili i campi informativi dell'utente (es. telefono, indirizzo, password), permettendo la sovrascrittura dei valori esistenti. |
| RF-1.3.3 | Il sistema deve validare la correttezza formale dei nuovi dati inseriti (es. sintassi email valida, complessità password) prima di procedere al salvataggio. |

| | |
|-----------------|---|
| RF-1.3.4 | Il sistema deve aggiornare in modo persistente le informazioni dell'utente, garantendo che le modifiche siano immediatamente effettive per le sessioni future. |
| RF-1.3.5 | Il sistema deve fornire un feedback visivo esplicito al termine dell'operazione (messaggio di conferma) |
| RF-1.1.1 | L'utente deve poter aggiungere, rimuovere o modificare la quantità dei prodotti nel carrello prima dell'ordine. |
| RF-1.1.2 | Il sistema deve garantire il vincolo di unicità dell'esercente: deve essere impossibile aggiungere un prodotto di un Esercente B se il carrello contiene già prodotti dell'Esercente A (a meno che il carrello non venga svuotato). |
| RF-1.1.3 | Il sistema deve permettere all'utente di modificare la quantità dei singoli prodotti già inseriti o di rimuoverli completamente dalla lista. |
| RF-1.1.4 | Il sistema deve ricalcolare e aggiornare visivamente il totale complessivo dell'ordine in tempo reale ad ogni singola variazione (aggiunta, rimozione o modifica quantità). |

| | |
|-----------------|--|
| RF-1.1.5 | Il sistema deve inibire l'aggiunta al carrello di prodotti che, seppur visibili nel menu, risultano temporaneamente non disponibili. |
| RF-1.5.1 | Il sistema deve impedire l'avvio della procedura di ordine se il carrello è vuoto. |
| RF-1.5.2 | Il sistema deve verificare in tempo reale lo stato dell'esercente al momento della conferma, bloccando l'ordine se il locale è chiuso o se mancano meno di 10 minuti alla chiusura. |
| RF-1.5.3 | Il sistema deve validare la disponibilità di ogni singolo prodotto nel carrello prima di registrare l'ordine. In caso di prodotti non disponibili, il sistema deve impedire l'ordine, segnalare l'anomalia e aggiornare il carrello. |
| RF-1.5.4 | Il sistema deve registrare l'ordine nel database associandolo all'utente e all'esercente, impostando lo stato iniziale: "In Attesa di Approvazione" |
| RF-1.5.5 | Il sistema deve rendere immediatamente visibile il nuovo ordine nella dashboard dell'esercente destinatario. |
| RF-1.5.6 | Il sistema deve garantire che la funzione di invio ordine sia accessibile |

| | |
|-----------------|---|
| | esclusivamente agli utenti autenticati (Login effettuato). |
| RF-1.6.1 | Il sistema deve verificare lo stato di giacenza attuale di tutti i prodotti presenti nel carrello nell'esatto momento in cui viene richiesto l'ordine. |
| RF-1.6.2 | Il sistema deve segnalare la presenza o la mancanza dei prodotti. |
| RF-1.7.1 | Il sistema deve fornire un campo di input testuale dedicato (es. area di testo) all'interno della vista del carrello per l'inserimento di richieste speciali. |
| RF-1.7.2 | Il sistema deve trattare il campo commento come opzionale, permettendo all'utente di procedere con l'ordine anche se il campo viene lasciato vuoto. |
| RF-1.7.3 | Il sistema deve accettare e validare l'input testuale dell'utente, imponendo un limite massimo di caratteri (es. 250) per garantire la leggibilità da parte dell'esercente. |
| RF-1.7.4 | Il sistema deve associare in modo persistente il testo inserito all'oggetto "Ordine" corrente, garantendo che la |

| | |
|-----------------|--|
| | nota sia visualizzabile nei dettagli dell'ordine ricevuto dall'Esercente. |
| RF-1.8.1 | Il sistema deve recuperare e far visualizzare all'utente, l'elenco completo di tutti gli ordini, includendo sia quelli conclusi (passati) che quelli eventualmente in corso (attuali). |
| RF-1.8.2 | Il sistema deve presentare la lista degli ordini ordinata cronologicamente (es. dal più recente al meno recente) per facilitare la consultazione. |
| RF-1.8.3 | Per ogni voce in elenco, il sistema deve mostrare le informazioni sintetiche essenziali: Nome Esercente, Data/Ora dell'ordine, Importo Totale e Stato Corrente. |
| RF-1.8.4 | Nel caso in cui non esistano ordini precedenti associati all'account (storico vuoto), il sistema deve visualizzare un messaggio informativo specifico (es. "Nessun ordine trovato") anziché una schermata vuota. |
| RF-1.9.1 | Il sistema deve permettere all'utente di selezionare un singolo ordine dalla lista dello storico per espanderne le informazioni. |

| | |
|-----------------|---|
| RF-1.9.2 | Il sistema deve recuperare e visualizzare tutte le informazioni specifiche dell'ordine selezionato, inclusi: elenco dei prodotti, quantità per prodotto, prezzi singoli, totale complessivo e la data/ora di effettuazione. |
| RF-1.9.3 | Il sistema deve mostrare chiaramente lo stato attuale di lavorazione dell'ordine. |
| RF-2.0.1 | Il sistema deve permettere all'Esercente loggato di visualizzare l'elenco completo dei propri prodotti presenti nel menù. |
| RF-2.0.2 | Il sistema deve permettere all'Esercente di modificare la disponibilità di un singolo prodotto (renderlo ordinabile o nasconderlo) in qualsiasi momento, indipendentemente dallo stato di apertura dell'esercizio. |
| RF-2.0.3 | Il sistema deve permettere all'Esercente di Aggiungere , Modificare (nome, descrizione, prezzo) o Eliminare un prodotto esclusivamente quando lo stato dell'esercizio è impostato su "CHIUSO". |
| RF-2.0.4 | Il sistema deve bloccare (disabilitare i comandi di interfaccia e rifiutare le richieste API) le operazioni di aggiunta, modifica strutturale ed eliminazione |

| | |
|-----------------|---|
| | prodotti quando l'esercizio è in stato "APERTO". |
| RF-2.0.5 | Il sistema deve validare i dati inseriti dall'Esercente prima del salvataggio, impedendo l'inserimento di prezzi negativi, nulli o di campi obbligatori vuoti. |
| RF-2.1.1 | Il sistema deve permettere all'Esercente di visualizzare i dati attuali del proprio profilo pubblico (Nome, Descrizione, Indirizzo e Logo corrente). |
| RF-2.1.2 | Il sistema deve permettere all'Esercente di modificare i campi testuali informativi (Nome, Descrizione, Indirizzo) che verranno mostrati agli utenti. |
| RF-2.1.3 | Il sistema deve permettere all'Esercente di effettuare l'upload di una nuova immagine da utilizzare come Logo dell'attività. |
| RF-2.1.4 | Il sistema deve validare il file immagine caricato, accettando esclusivamente formati grafici supportati (es. JPG, PNG) e rifiutando formati non validi (es. PDF) o file che superano la dimensione massima consentita. |

| | |
|-----------------|--|
| RF-2.1.5 | Il sistema deve impedire il salvataggio delle modifiche se uno o più campi obbligatori (es. Nome Esercizio) risultano vuoti. |
| RF-2.1.6 | Il sistema deve aggiornare in tempo reale la scheda dell'esercente visibile nella lista pubblica (lato Cliente) dopo il salvataggio delle modifiche. |
| RF-2.2.1 | Il sistema deve permettere all'Esercente di visualizzare una lista dedicata contenente esclusivamente gli ordini nello stato "In attesa di approvazione", completi di dettagli (elenco piatti, totale e orario ricezione). |
| RF-2.2.2 | Il sistema deve permettere all'Esercente di Accettare un ordine specifico. A seguito di questa azione, il sistema deve aggiornare lo stato dell'ordine in "Accettato" e abilitare il passaggio alla fase successiva (preparazione). |
| RF-2.2.3 | Il sistema deve permettere all'Esercente di Rifiutare un ordine specifico. A seguito di questa azione, il sistema deve aggiornare lo stato dell'ordine in "Rifiutato" e impedire ulteriori modifiche su di esso. |
| RF-2.2.4 | Il sistema deve rendere immediatamente visibile il cambio di stato (da "In attesa" a "Accettato" o |

| | |
|-----------------|--|
| | "Rifiutato") nella scheda dell'ordine visualizzata dal Cliente. |
| RF-2.3.1 | Il sistema deve permettere all'Esercente di visualizzare l'elenco degli ordini attivi in lavorazione. |
| RF-2.3.2 | Il sistema deve permettere all'Esercente di avanzare lo stato di un ordine da "Accettato" a "In Preparazione" per segnalare l'inizio della produzione. |
| RF-2.3.3 | Il sistema deve permettere all'Esercente di avanzare lo stato di un ordine da "In Preparazione" a "Pronto per il ritiro". |
| RF-2.3.4 | Al momento del passaggio allo stato "Pronto per il ritiro", il sistema deve generare automaticamente un codice univoco di ritiro e renderlo visibile al Cliente nei dettagli dell'ordine. |
| RF-2.3.5 | Il sistema deve fornire all'Esercente una funzionalità per verificare il codice di ritiro fornito dal Cliente. Se il codice è corretto, il sistema deve permettere il passaggio allo stato finale "Ritirato". |
| RF-2.3.6 | Il sistema deve aggiornare in tempo reale la vista del Cliente ad ogni cambio |

| | |
|-----------------|--|
| | di stato ("In Preparazione", "Pronto", "Ritirato"). |
| RF-2.3.7 | Il sistema deve permettere all'Esercente di chiudere forzatamente un ordine impostando lo stato su "Non Ritirato" nel caso in cui il Cliente non si presenti entro l'orario di chiusura. |
| RF-1.2.1 | Il sistema deve permettere al Cliente loggato di visualizzare il listino completo di un esercente selezionato, mostrando per ogni prodotto: nome, prezzo e descrizione. |
| RF-1.2.2 | Il sistema deve presentare i prodotti organizzati visivamente per categorie (es. Primi, Secondi, Bevande) per facilitare la consultazione. |
| RF-1.2.3 | Il sistema deve permettere al Cliente di filtrare i prodotti visualizzati in base alla categoria. |
| RF-1.2.4 | Il sistema deve distinguere graficamente i prodotti Disponibili da quelli Non Disponibili (esauriti), inibendo la possibilità di aggiungere questi ultimi al carrello. |
| RF-1.2.5 | Il sistema deve permettere al Cliente di selezionare un prodotto per |

| | |
|-----------------------|--|
| | visualizzarne la scheda di dettaglio, che deve includere informazioni aggiuntive come ingredienti e allergeni. |
| RF-1.2.6 | Nel caso in cui l'esercente selezionato sia in stato "Chiuso", il sistema deve mostrare il menu in modalità "Sola Lettura", visualizzando un avviso informativo e disabilitando tutte le funzioni di acquisto. |
| RF-1.2.7 | Nel caso in cui il menu dell'esercente risulti vuoto (lista prodotti vuota), il sistema deve fornire un feedback visivo esplicito all'utente ("Nessun prodotto disponibile"). |
| RF-1.10. 1 | Il sistema deve permettere la registrazione di un nuovo utente richiedendo l'inserimento di nome, cognome, email istituzionale e password. |
| RF-1.10. 2 | Il sistema deve validare il dominio dell'indirizzo email inserito, consentendo la registrazione esclusivamente agli indirizzi appartenenti al dominio universitario |
| RF-1.10. 3 | Il sistema deve verificare che l'indirizzo email non sia già associato a un |

| | |
|-------------------|--|
| | account esistente (controllo di unicità), impedendo la creazione di duplicati. |
| RF-1.10. 4 | Nel caso di tentativo di registrazione con email già esistente, il sistema deve bloccare l'operazione e mostrare un messaggio che suggerisce all'utente di effettuare il login o recuperare la password. |
| RF-1.10. 5 | Il sistema deve validare la correttezza formale dei dati inseriti (es. campi obbligatori mancanti, password troppo breve) fornendo un messaggio di errore specifico in corrispondenza del campo non valido. |
| RF-1.10. 6 | Al termine di una registrazione conclusa con successo, il sistema deve effettuare automaticamente il login dell'utente, rendendo attiva la sessione senza richiedere un nuovo inserimento delle credenziali. |
| RF-1.4.1 | Il sistema deve permettere al Cliente loggato di visualizzare l'elenco completo degli esercenti convenzionati presenti nella piattaforma. |
| RF-1.4.2 | Il sistema deve mostrare, per ogni esercente in lista, le informazioni riepilogative essenziali: Nome |

| | |
|-----------------|--|
| | dell'attività, Logo/Immagine e breve descrizione. |
| RF-1.4.3 | Il sistema deve indicare visivamente lo stato di apertura corrente di ogni esercente (es. Aperto/Chiuso) direttamente nella lista, per orientare la scelta dell'utente. |
| RF-1.4.4 | Il sistema deve filtrare la lista mostrando esclusivamente i ristoranti marcati come "Attivi" nel database, nascondendo quelli dismessi o sospesi. |
| RF-1.4.5 | Il sistema deve permettere all'utente di selezionare un esercente dalla lista per accedere alla visualizzazione del relativo menu (collegamento al caso d'uso Consultazione Menu). |

| ID | Requisito Non Funzionale |
|---|---|
| RNF-1.1 (Tecnologici e Architetturali) | Il Backend deve essere sviluppato in PHP nativo senza l'utilizzo di framework (es. Laravel). Il Frontend deve essere sviluppato in HTML, CSS e JavaScript (è consentito jQuery), senza framework reattivi come React o Angular. |

| | |
|---|--|
| RNF-1.2 (Tecnologici e Architetturali) | L'applicazione lato client deve rispettare rigorosamente il pattern architettonale Model-View-Presenter (MVP), separando la logica di presentazione dalla struttura HTML e dai dati. |
| RNF-1.3 (Tecnologici e Architetturali) | La comunicazione tra Frontend e Backend deve avvenire esclusivamente tramite RESTful API asincrone, utilizzando il formato JSON per lo scambio dati. |
| RNF-1.4 (Tecnologici e Architetturali) | È vietata la generazione di codice HTML lato server (Server-Side Rendering). Il PHP deve restituire solo dati grezzi; la costruzione dell'interfaccia è demandata al JavaScript client-side. |
| RNF-1.5 (Tecnologici e Architetturali) | Il sistema deve essere fornito all'interno di container Docker configurati (file docker-compose.yml), garantendo l'esecuzione su qualsiasi ambiente ospite senza installazioni manuali. |
| RNF-1.6 (Tecnologici e Architetturali) | Il codice sorgente e la documentazione devono essere gestiti tramite repository GitHub, dimostrando un corretto flusso di lavoro collaborativo. |
| RNF-2.1 (Sicurezza) | Il sistema deve gestire l'autenticazione tramite Token (Stateless) generati dal server e memorizzati nel client. È vietato l'uso delle sessioni server-side native di PHP. |
| RNF-2.2 (Sicurezza) | Le password degli utenti devono essere memorizzate nel database esclusivamente sotto forma di hash crittografico sicuro. |

| | |
|---|---|
| RNF-2.3 (Sicurezza) | Tutti gli endpoint API che permettono la modifica dei dati o l'accesso a informazioni personali devono richiedere un token di autenticazione valido nell'header della richiesta HTTP. |
| RNF-3.1 (Usabilità e Interfaccia) | L'interfaccia utente deve essere Responsive, adattando automaticamente il layout per dispositivi mobili, desktop e tablet. |
| RNF-3.2 (Usabilità e Interfaccia) | Il sistema deve fornire un feedback visivo in risposta a qualsiasi azione dell'utente (es. messaggi di errore, conferma inserimento carrello). |
| RNF-4.1 (Affidabilità e Correttezza) | Il sistema deve garantire la consistenza dei dati in caso di accessi concorrenti, impedendo l'ordine di prodotti esauriti se richiesti simultaneamente da più utenti. |
| RNF-4.2 (Affidabilità e Correttezza) | Il Backend deve implementare una validazione ridondante di tutti i dati in ingresso, non affidandosi esclusivamente ai controlli effettuati dal JavaScript lato client. |
| RNF-5.1 (Dominio) | Il sistema non deve gestire transazioni monetarie elettroniche; il pagamento è previsto esclusivamente in loco al momento del ritiro. |

MATRICE DI TRACCIABILITÀ

Area Pubblica (Cliente UniPR)

| ID Caso d'Uso | Nome Caso d'Uso | Requisiti Funzionali (RF) | Requisiti Non Funzionali (RNF) |
|----------------------|-----------------------------------|--|--|
| UC-1.0 | Login | RF-1.0.1, RF-1.0.2 | RNF-2.1 (Generazione Token), RNF-2.2 (Hash Password), RNF-1.3 (API JSON), RNF-3.2 (Feedback) |
| UC-1.10 | Registrazione | RF-1.10.1, RF-1.10.2, RF-1.10.3, RF-1.10.4, RF-1.10.5, RF-1.10.6 | RNF-2.2 (Hash Password), RNF-4.2 (Validazione Backend), RNF-3.2 (Feedback) |
| UC-1.3 | Gestire Profilo | RF-1.3.1, RF-1.3.2, RF-1.3.3, RF-1.3.4, RF-1.3.5 | RNF-2.1 (Token richiesto), RNF-2.3 (Protezione API PUT), RNF-4.2 (Validazione dati) |
| UC-1.4 | Consultare Lista Esercenti | RF-1.4.1, RF-1.4.2, RF-1.4.3, RF-1.4.4, RF-1.4.5 | RNF-2.1 (Token richiesto), RNF-3.1 (Responsive Mobile), RNF-1.4 (No Server HTML), RNF-1.3 (API GET) |
| UC-1.2 | Consultazione Menu | RF-1.2.1, RF-1.2.2, RF-1.2.3, RF-1.2.4, RF-1.2.5, RF-1.2.6, RF-1.2.7 | RNF-2.1 (Token richiesto), RNF-4.1 (Disponibilità Real-time), RNF-3.1 (Responsive), RNF-1.2 (MVP View) |
| UC-1.1 | Gestire Carrello | RF-1.1.1, RF-1.1.2, RF-1.1.3, RF-1.1.4, RF-1.1.5 | RNF-2.1 (Token richiesto), RNF-1.2 (Logica JS Presenter), RNF-3.2 (Feedback immediato), RNF-1.4 (Calcolo locale/API) |

| | | | |
|---------------|--|---|--|
| UC-1.7 | Aggiungere Commento | RF-1.7.1, RF-1.7.2, RF-1.7.3, RF-1.7.4 | RNF-2.1 (Token richiesto), RNF-4.2 (Validazione lunghezza Backend) |
| UC-1.5 | Ordinare | RF-1.5.1, RF-1.5.2, RF-1.5.3, RF-1.5.4, RF-1.5.5, RF-1.5.6 | RNF-2.1 (Token richiesto), RNF-2.3 (Protezione API POST), RNF-4.1 (Transazione/Concorrenza), RNF-5.1 (Pagamento in loco) |
| UC-1.6 | Verificare disponibilità prodotti | RF-1.6.1, RF-1.6.2 | RNF-2.1 (Token richiesto), RNF-4.1 (Integrità Dati concorrenti) |
| UC-1.8 | Visualizzare Storico ordini | RF-1.8.1, RF-1.8.2, RF-1.8.3, RF-1.8.4 | RNF-2.1 (Token richiesto), RNF-1.3 (API JSON) |
| UC-1.9 | Visualizzare dettagli ordine | RF-1.9.1, RF-1.9.2, RF-1.9.3 | RNF-2.1 (Token richiesto), RNF-3.1 (Layout Responsive) |

Area Gestionale (Esercente)

| ID Caso d'Uso | Nome Caso d'Uso | Requisiti Funzionali (RF) | Requisiti Non Funzionali (RNF) |
|---------------|--------------------------------|---|---|
| UC-2.1 | Modifica Info Esercente | RF-2.1.1, RF-2.1.2, RF-2.1.3, RF-2.1.4, RF-2.1.5, RF-2.1.6 | RNF-2.1 (Token richiesto), RNF-2.3 (Protezione API), RNF-4.2 (Validazione Formato File), RNF-3.2 (Feedback salvataggio) |

| | | | |
|---------------|---|--|--|
| UC-2.0 | Gestire Menu | RF-2.0.1, RF-2.0.2, RF-2.0.3, RF-2.0.4, RF-2.0.5 | RNF-2.1 (Token richiesto), RNF-4.2 (Validazione Prezzi), RNF-1.3 (CRUD API) |
| UC-2.2 | Gestire Richieste Ordini | RF-2.2.1, RF-2.2.2, RF-2.2.3, RF-2.2.4 | RNF-2.1 (Token richiesto), RNF-3.2 (Feedback immediato), RNF-3.1 (Desktop View), RNF-4.1 (Consistenza Stato) |
| UC-2.3 | Gestire Stato Ordini | RF-2.3.1, RF-2.3.2, RF-2.3.3, RF-2.3.4, RF-2.3.5, RF-2.3.6, RF-2.3.7 | RNF-2.1 (Token richiesto), RNF-1.3 (Aggiornamento asincrono), RNF-2.3 (Protezione Azioni), RNF-4.1 (Sequenza stati corretta) |