

Winer

Problem Statement

Versione 1.0.1



Antica enoteca di Roma - Unsplash

Giovanni Prisco

051 21 06534

Indice

Problem Statement	1
Indice.....	2
Revision History	3
1. Introduzione	4
2. Requisiti funzionali	5
2.1 Giuseppe	5
2.1.1 Login	5
2.1.2 Gestione scorte.....	5
2.1.3 Gestione ordini.....	6
2.2 Utente Guest	6
2.2.1 Catalogo.....	6
2.2.2 Dettagli Vino.....	7
2.2.3 Carrello.....	7
2.2.4 Login/Registrazione	7
2.3 Utente Registrato	7
2.3.1 Storico ordini.....	8
2.3.2 Checkout.....	8
3. Requisiti non funzionali	8
4. Target Environment	9
5. Deadlines Proposte.....	9
6. Criteri di Accettazione	10

Revision History

Data	Versione	Descrizione
16/12/2021	1.0	Completata una prima versione pronta per la revisione
17/12/2021	1.0.1	Modifica del documento in seguito ai feedback ricevuti dal professore in merito ai requisiti funzionali e non e la scelta dei criteri di accettazione e del target environment. + Aggiunta dell'indice e della cover

1. Introduzione

Giuseppe è un ragazzo di 27 anni che gestisce un'enoteca, vanta un'ottima selezione di vini in una vasta cantina. Ha avviato la sua attività nel 2015 e da allora non smette di sperimentare ed offrire sempre più vini spaziando dai più conosciuti a quelli più di nicchia.

Molto spesso però, Giuseppe fa fatica a tenere traccia delle varie bottiglie che offre e delle relative scorte e vivendo in un piccolo paesino ha anche difficoltà a scalare il suo business, motivo per cui vorrebbe cominciare a proporre la sua selezione di vini anche altrove.

Non ha attualmente un grande capitale da investire per aprire nuovi punti vendita in altre città, quindi sceglie di cominciare a costruire la sua presenza online tramite un e-commerce che vorrebbe chiamare Winer.

2. Requisiti funzionali

Cominciamo a creare degli scenari che descrivano l'utilizzo di Winer dal punto di vista di Giuseppe.

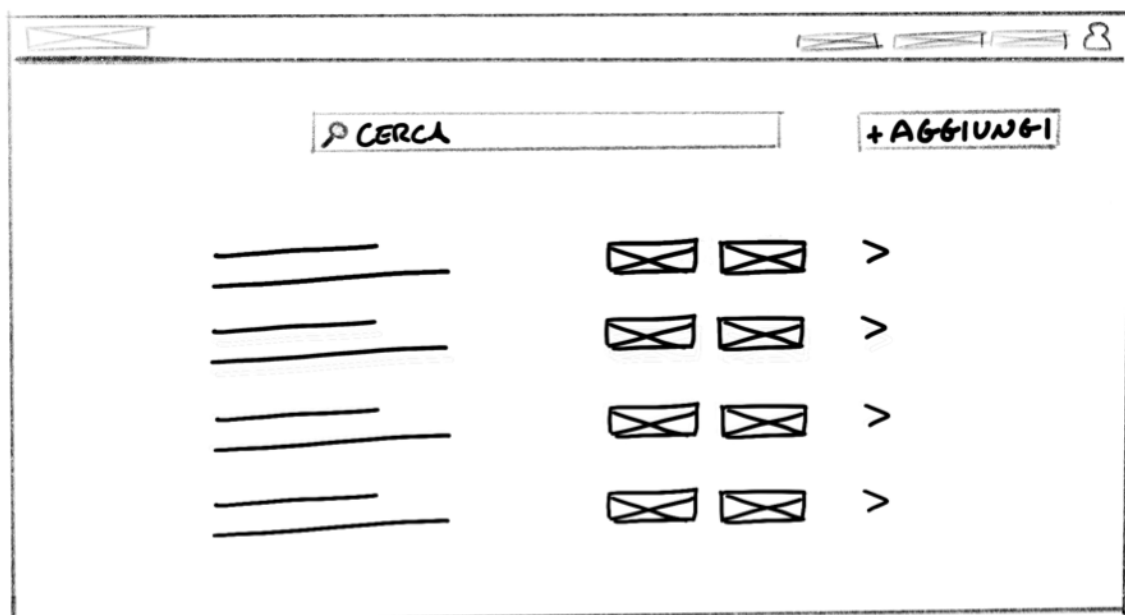
2.1 Giuseppe

2.1.1 Login

Il sistema dovrà fornire funzionalità di autenticazione per la parte gestionale.

Le credenziali devono essere create in fase di configurazione del sistema e subito dopo fornite a Giuseppe.

2.1.2 Gestione scorte



Wireframe Gestione Scorte

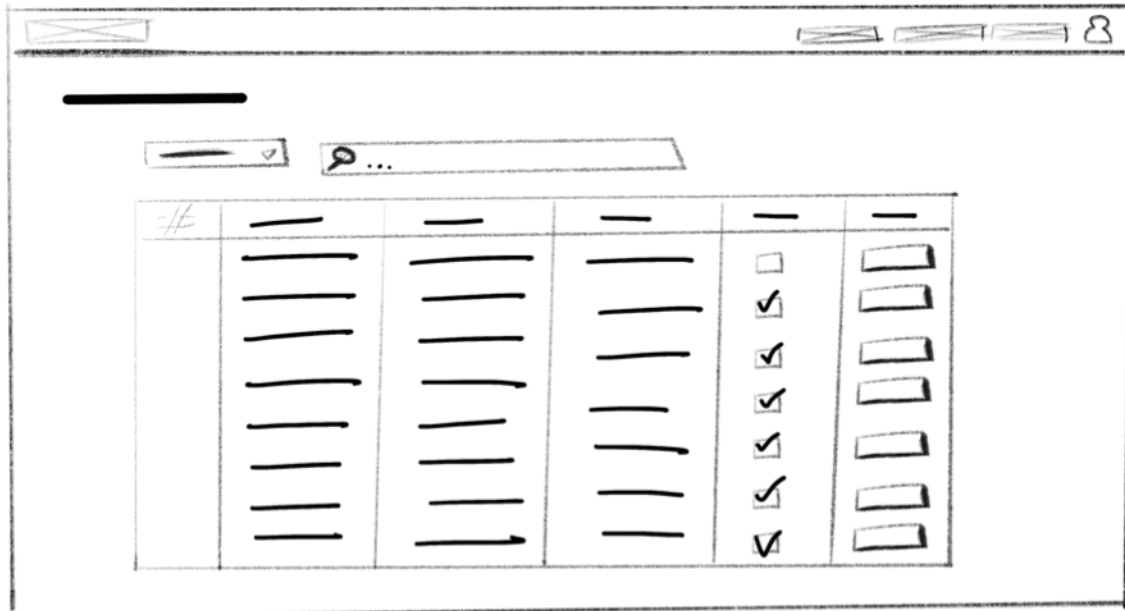
Il sistema dovrà mostrare il catalogo di vini a Giuseppe in modalità di modifica e avere gli appositi pulsanti per eseguire operazioni CRUD sui vini del catalogo

Ovviamente il sistema deve validare i dati e avvertire Giuseppe in caso di errore di inserimento di alcuni di questi, altrimenti il sistema deve renderli persistenti.

Aggiunta vino

Il sistema dovrà permettere a Giuseppe di aggiungere un vino tramite la pressione di un apposito pulsante e presentargli un form con i dati necessari all'aggiunta di questo.

2.1.3 Gestione ordini



Wireframe Gestione Ordini

In questa pagina il sistema dovrà mostrare a Giuseppe gli ordini che devono essere presi in carico.

Da qui dovrà anche permettere a Giuseppe di segnare un ordine come “Preso in carico” o annullarlo.

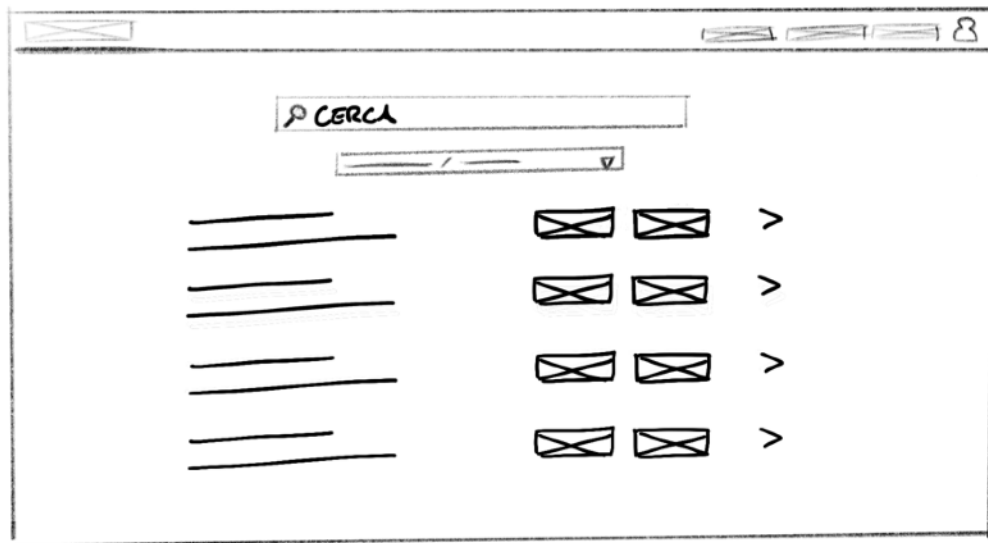
2.2 Utente Guest

Ora passiamo a descrivere le funzionalità offerte da Winer agli utenti visitatori.

2.2.1 Catalogo

Il sistema dovrà presentare all'utente i vini del catalogo offerti da Giuseppe.

Il sistema dovrà permettere all'utente di ordinare la collezione di vini in base a vari parametri (es. prezzo in senso crescente) oltre a permettergli di filtrare i vini per nome tramite una casella input in cui può digitare del testo ed ottenere suggerimenti di autocompletamento.

*Wireframe Catalogo*

Il sistema dovrà fornire all'utente la funzionalità di aggiunta al carrello (carrello di cui parleremo tra poco) anche dal catalogo.

2.2.2 Dettagli Vino

Il sistema dovrà permettere all'utente di cliccare su un vino per visualizzarne i dettagli. In questa pagina dovrà mostrare tutti i dati, di cui parleremo in altri documenti, correlati a quel singolo vino oltre a mostrare vini correlati come suggerimento per mettere l'utente in condizione di poter continuare la navigazione. Anche in questa pagina dovrà essere offerta la funzionalità di aggiunta al carrello.

2.2.3 Carrello

Il sistema dovrà fornire come accennato al punto **2.2.1** e **2.2.2** funzionalità di gestione di un carrello che altro non è che un insieme di vini che l'utente ha intenzione di acquistare.

2.2.4 Login/Registrazione

Il sistema dovrà permettere all'utente di registrarsi o effettuare il login alla piattaforma. Queste operazioni sono richieste in alcuni casi come vedremo tra poco, come Checkout e storico ordini.

2.3 Utente Registrato

Il sistema fornirà funzionalità aggiuntive agli utenti registrati come accennato sopra, in particolare fornirà la possibilità di finalizzare l'acquisto dei vini nel carrello tramite un'apposita pagina di Checkout e la possibilità di visualizzare gli ordini effettuati.

2.3.1 Storico ordini

Il sistema dovrà permettere all'utente di visualizzare il suo storico ordini opportunamente palinato e permettere all'utente di ordinarli per data.

2.3.2 Checkout

Il sistema dovrà permettere all'utente di finalizzare l'ordine dei vini presenti nel carrello pagando tramite carta di credito.

3. Requisiti non funzionali

- **Usabilità**

Il sistema dovrà essere il più accessibile possibile, per questo motivo dovrà essere responsive, ovvero correttamente visualizzabile e funzionante su qualsiasi dispositivo.

- **Performance**

Il sistema dovrà offrire tempi di risposta inferiori ai 2 secondi anche per le operazioni più onerose quali transazioni e registrazione di nuovi utenti.

- **Manutenibilità**

Il sistema deve essere facilmente mantenibile ed estendibile per sviluppi futuri, quindi dovrà basarsi su uno sviluppo modulare.

- **Sicurezza**

Oltre a garantire la persistenza delle password nel database con algoritmi di crittografia "one-way", tutti i dati degli utenti dovranno viaggiare su protocolli sicuri.

- **Robustezza**

Il sistema dovrà assicurarsi che tutte le operazioni che saranno progettate siano eseguibili solo ed esclusivamente se sono rispettate tutte le precondizioni.

4. Target Environment

Il progetto sarà sviluppato e deployato in ambiente web, un server gestirà la logica di business che verrà implementata da un client.

Di seguito alcune tecnologie che verranno utilizzate per lo sviluppo del sistema:

- Node.js per i servizi server-side
- React per avere un client che sia una Single-Page Application (SPA) e che implementi tramite un'interfaccia da concordare i servizi offerti dal server
- Webserver NGINX
- Database MySQL

5. Deadlines Proposte

- 07-10: Project Proposal
- **16-12: Problem Statement**
- 16-12: Requisiti e casi d'uso
- Da stabilire: Requirements Analysis Document
- Da stabilire: System Design Document
- Da stabilire: Piano di test e specifica interfacce dei moduli del sistema
- Altri documenti alla consegna finale del progetto, tra cui: esecuzione dei test, object design document e codice.

6. Criteri di Accettazione

- L'interfaccia dovrà essere responsive
- Una percentuale maggiore o uguale all'80% del sistema dovrà superare la fase di verifica e di convalida
- Ovviamente tutti i requisiti discussi sopra dovranno essere correttamente rispettati