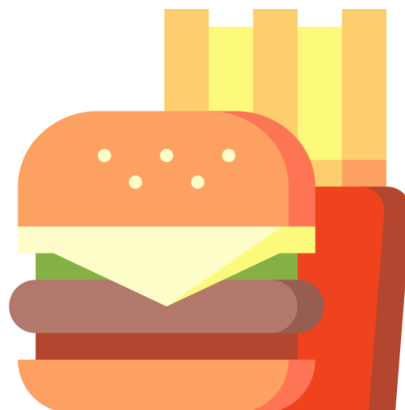


Food to Go!



Progetto d'esame
Applicazioni Web e Cloud

Gabriele Zarcone

22 settembre 2020

INDICE

Le 5 fasi del lavoro	2
1 - Analisi dei requisiti	2
2 - Identificazione delle funzionalità da sviluppare	2
3 - Progettazione della struttura e della presentazione delle pagine web	4
4 - Progettazione della sorgente di informazioni statica o dinamica	5
5 - Implementazione delle pagine Web.	7
Come vengono gestiti i due scenari principali	8
Gestione dei menù	8
Gestione degli ordini	9
Le principali funzionalità richieste	10
Visualizzazione di informazioni relative ai piatti, ai clienti ai ristoratori registrati e agli acquisti.	10
Visualizzazione dei prodotti	10
Ricerca dei piatti	11
Login al sito e acquisto di un piatto a scelta.	11
Visualizzazione degli acquisti presenti e passati per un cliente.	12
Conclusioni e possibili espansioni future	13

Le 5 fasi del lavoro

1 - Analisi dei requisiti

Il lavoro per la realizzazione del progetto d'esame è partita, come si conviene, dall'analisi dei requisiti necessari. Partendo dalle specifiche si sono analizzate le parole chiave e le funzionalità necessarie alla realizzazione del progetto.

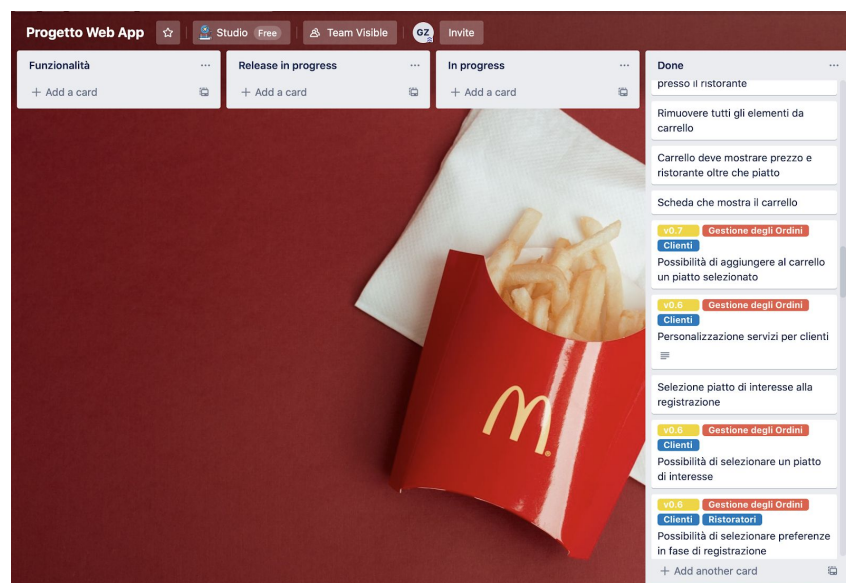
Partendo dalle keywords sono stati estrapolati i concetti fondamentali di tutti gli oggetti che avrebbero avuto bisogno di essere memorizzati ed utilizzati dall'applicazione web, insieme anche alle funzionalità necessarie all'utente finale.

Infine si è valutato con quali strumenti affrontare il progetto. Partendo dall'utilizzo di Javascript, HTML5 e CSS3 indicato dalle specifiche, si è ponderato sull'utilizzo o meno di framework quali Bootstrap o UiKit per CSS e React per Javascript e l'intera webapp. Si è preferito però non utilizzare nessuno di essi e mantenere il progetto il più possibile vicino alle funzionalità base dei linguaggi e alle nozioni imparate durante il corso. In questo modo infatti si è potuto approfondire maggiormente ciò che sta alla base dei framework sopracitati e quindi imparare nozioni svincolate dalla libreria che si è deciso di usare per un determinato progetto e dare a questo lavoro uno spettro più ampio e didattico che una semplice produzione di contenuto.

2 - Identificazione delle funzionalità da sviluppare

Partendo quindi dalle specifiche si è prodotta una lista di note dettagliata di tutte le funzionalità base e accessorie indicate.

Ad ausilio di questa fase si è pensato di utilizzare il software Trello che ha consentito di arricchire ogni nota con tag, commenti e check list che sono stati inseriti sia in fase di identificazione delle specifiche che in fase di sviluppo del progetto.



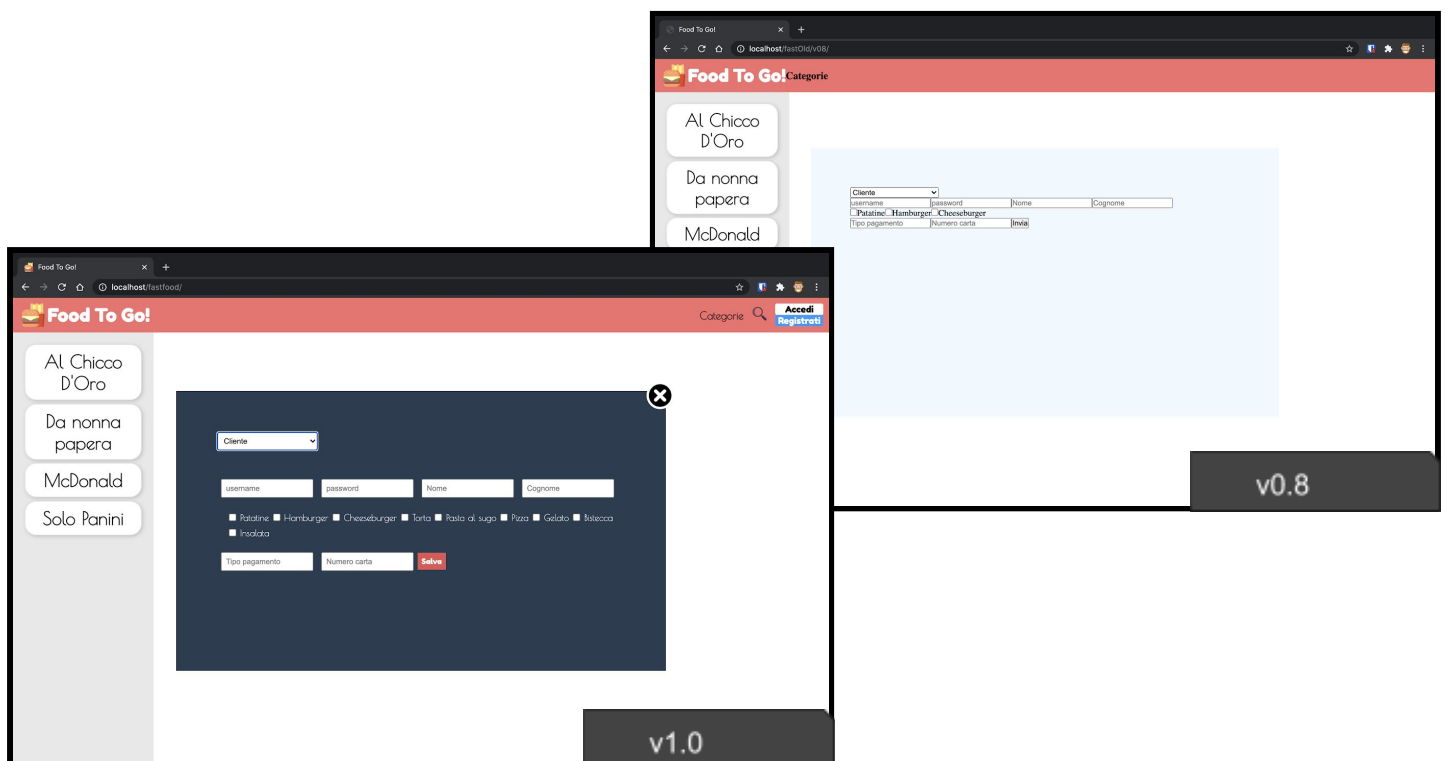
Questo ha consentito di dividere il lavoro in macroblocchi in base alle funzionalità, all'importanza della feature e altri criteri più o meno specifici.

L'utilizzo di Trello ha consentito anche di adottare un approccio stile Kanban alla produzione di ogni funzionalità. Ognuna di esse, infatti, veniva spostata dalla lista "Da implementare" a quella "In sviluppo" per finire poi in quella "Completati".

Questo ha consentito non solo una maggiore organizzazione del flusso di lavoro, ma anche maggiore flessibilità poiché ogni funzionalità poteva essere aggiunta o modificata in ogni momento tenendo traccia di ogni cosa.

L'utilizzo dei tag inoltre ha consentito di raggruppare le diverse funzionalità in diverse *release* ognuna delle quali consegnava un prodotto finito e già utilizzabile seppur con poche funzionalità. L'ordine in cui sono state fatte le diverse funzionalità quindi ha guardato anche al prodotto finale, in modo tale che ogni release potesse essere funzionante anche da sola. Un esempio di questo approccio è stato l'ordine con cui si è prodotto il CSS: il sito fino all'ultima release ha mantenuto uno stile molto basico ma funzionale alle funzionalità che venivano realizzate. Solamente con l'ultima release si è prestata attenzione ai colori e a dare un aspetto più moderno al tutto.

Qui sotto, ad esempio, la differenza tra la pagina di registrazione di un nuovo utente della versione 0.8 e la versione 1.0:



Per poter tener traccia di tutte le funzionalità e le diverse modifiche apportate durante il corso dello sviluppo si è utilizzato inoltre git come tool di versioning coadiuvato da git-flow per la gestione delle feature e delle diverse release.

3 - Progettazione della struttura e della presentazione delle pagine web

Lo sviluppo dell'applicazione è partito con l'intento di utilizzare solamente una pagina web, senza che l'utente avesse bisogno di navigare tra svariate pagine per ottenere tutte le funzionalità. La home page è, in fin dei conti, la pagina che contiene tutte le funzionalità del sito. Si è optato per questa scelta per rendere il sito molto più vicino ad un'app di quelle che si usano tutti i giorni sul computer piuttosto che un classico sito web.

Tutto ciò con cui interagisce l'utente è generato dinamicamente dal codice javascript che sfrutta come base un singolo file html che fa da template iniziale.

La pagina presenta una barra di navigazione superiore utile per il login, per la ricerca dei piatti e, una volta loggati, l'utilizzo di funzionalità come la lista degli ordini recenti e il carrello.

Sulla sinistra è sempre presente una spalla contenente tutti i ristoranti registrati al sito. Il suo contenuto è variabile in base al tipo di utente che accede. Ogni qualvolta che ad accedere è un *cliente* questa si popola con tutti i ristoranti presenti in memoria, mentre se ad accedere è un *ristoratore* questo può visualizzare solamente i ristoranti di cui è il proprietario.

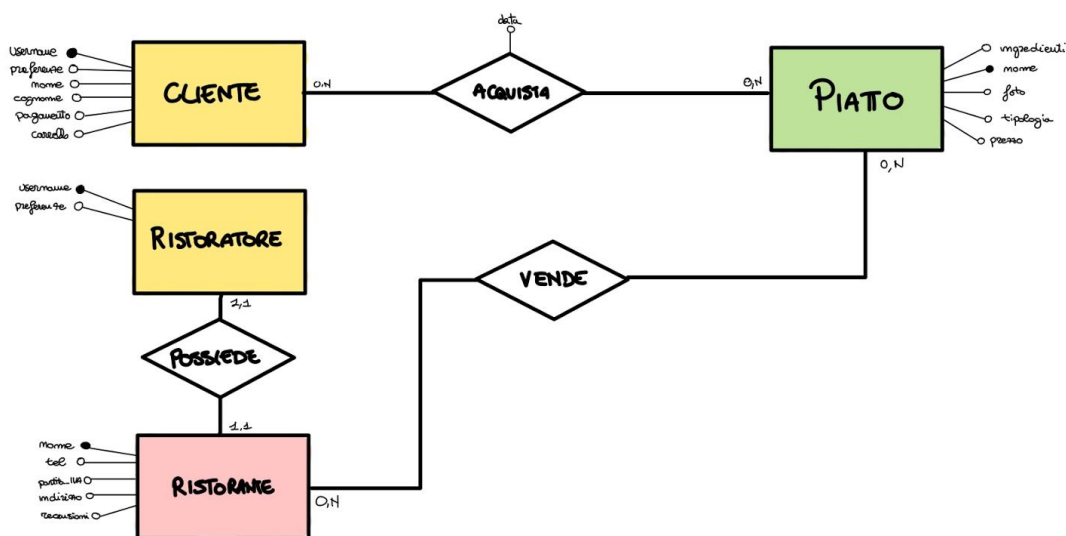
Il click su uno dei ristoranti popola la sezione più grande presente nella pagina. Qui vengono mostrati i piatti ogni volta che ve ne è la necessità. Difatti la sezione viene utilizzata quando si seleziona un ristorante e vengono mostrati i piatti venduti, quando ci si trova nella homepage e viene visualizzato l'elenco di tutti i piatti preferiti dell'utente o quando viene effettuata una ricerca e vengono mostrati i risultati.

Sono stati implementati inoltre altri due tipi di contenitori. Il primo è una tendina che dalla barra di navigazione viene verso il basso ed è utilizzata ad esempio per mostrare i piatti contenuti nel carrello o le informazioni personali dell'utente. Il secondo contenitore invece è una finestra pop-up posizionata al centro dello schermo e utilizzata per funzionalità che richiedono più spazio come la registrazione di utenti o la creazione di un nuovo ristorante.

Tutte queste soluzioni insieme consentono di utilizzare una singola pagina web per ogni funzionalità richiesta.

4 - Progettazione della sorgente di informazioni statica o dinamica

Come già accennato precedentemente, si è iniziata l'analisi dei requisiti del progetto attraverso l'analisi delle parole chiave delle specifiche. Da queste sono stati estrapolati degli oggetti che avrebbero dovuto essere memorizzati per il mantenimento dello stato dell'applicazione. Per poter costruire uno schema efficace di memorizzazione di ognuno di questi oggetti è stato utilizzato uno schema molto simile ad un Diagramma ER che viene utilizzato per i database relazionali



Nonostante la modalità di stoccaggio delle informazioni che si è poi utilizzata sia lontana da quella di un database, questo schema ha consentito di organizzare il lavoro, comprendere le relazioni tra i vari oggetti e capire quali informazioni dovessero essere immagazzinate.

Effettivamente poi le informazioni sono inserite in un file sotto formato JSON contenuto nella cartella *data* (vi è anche un altro file di backup per tornare ad una situazione iniziale).

In fase di primo avvio o di ricarica della pagina tutto il contenuto del file viene utilizzato per inizializzare delle variabili del LocalStorage del sito. Sono state quindi sviluppate delle funzioni di salvataggio e di ricerca delle informazioni all'interno del localStorage. Infatti ogni informazione di cui hanno bisogno i vari elementi dell'applicazione viene recuperata da queste variabili e non dal file. Quest'ultimo quindi rimane solamente come memoria dello stato dell'applicazione, di modo che ad un nuovo avvio non si ritorni ad un vecchio stato iniziale.

Ogni operazione che comporta una modifica delle informazioni dello stato comporta una chiamata XMLHttpRequest verso il server che contiene tutte le informazioni dello stato attuale. Il server, con un semplice script php, andrà poi a sovrascrivere il file JSON. (Affinché il php funzioni ci sarà bisogno di dare al server i permessi di scrittura del file data.json).

Le tre funzioni che sono state implementate per gestire le informazioni dell'applicazione sono:

- `upload_initial_state()`: viene lanciata ogni volta che viene caricato il sito e inserisce all'interno delle variabili di LocalStorage i vari oggetti memorizzati nel file JSON sul server.
- `db(table)`: viene usata per ricavare le informazioni. Se un componente dell'applicazione necessita di interrogare lo stato attuale usa questa funzione specificando il nome dell'oggetto di cui si ha bisogno. La funzione restituisce un oggetto JSON.
- `update_db(table, new_db)`: usata per modificare le informazioni salvate. Quando viene invocata aggiorna la variabile di LocalStorage corrispondente all'argomento "table" con il contenuto di "new_db" che deve essere un oggetto JSON del tutto simile a quello che sostituisce. Una volta aggiornata la variabile viene chiamata la funzione `updateServer()`.
- `updateServer()`: la sua esecuzione invoca una chiamata sincrona di XMLHttpRequest verso il server. La richiesta POST contiene tutti gli oggetti contenuti nelle variabili di LocalStorage sotto forma di stringa e viene inviata al file `service/upload_data.php` che con una piccola procedura sovrascrive il contenuto del file JSON allineando così lo stato contenuto in LocalStorage con quello contenuto all'interno del file.

Le operazioni eseguite da `updateServer` sono state mantenute in maniera sincrona in modo che non vi siano cambi di stato durante l'allineamento delle informazioni tra LocalStorage e file JSON.

Come ultimo accorgimento per il passaggio di informazioni tra applicazione e server si è usata la coppia di funzioni **JSON.parse** e **JSON.stringify**. Le informazioni JSON infatti vengono incapsulate all'interno di una stringa grazie a `JSON.stringify` per essere lavorabile come testo dalle funzioni php e vengono poi trasformate nuovamente in oggetto grazie a `JSON.parse` quando vengono salvate all'interno del LocalStorage.

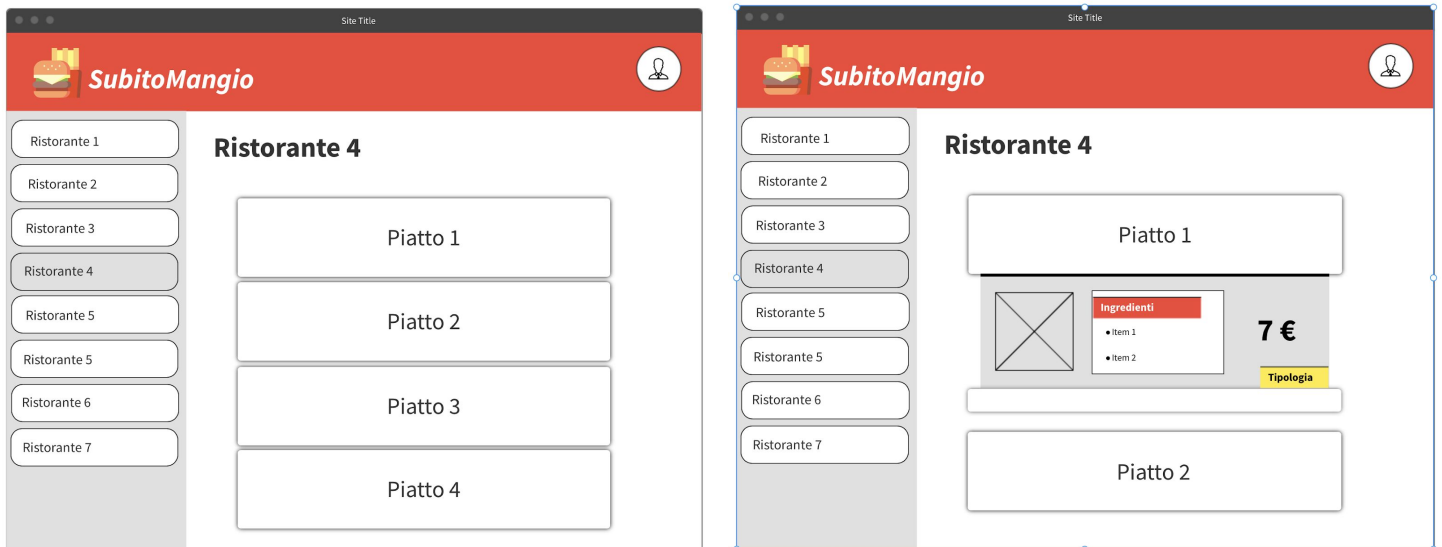
Durante l'utilizzo dell'applicazione da parte degli utenti viene usato anche lo spazio di SessionStorage in cui vengono immagazzinate informazioni che non necessitano di persistenza sul server, ma che sono utili finché non viene chiusa l'applicazione. Tra queste informazioni troviamo ad esempio l'username dell'utente loggato o il tipo di utente. Ognuna di queste informazioni non viene salvata nel file JSON e viene persa alla fine della sessione.

5 - Implementazione delle pagine Web.

La implementazione del sito ha mantenuto la separazione tra struttura della pagina e contenuto.

Si è infatti utilizzato HTML e Javascript per definire ciò che doveva essere **contenuto** all'interno della pagina. Il primo per il contenuto statico e il secondo per il contenuto dinamicamente generato.

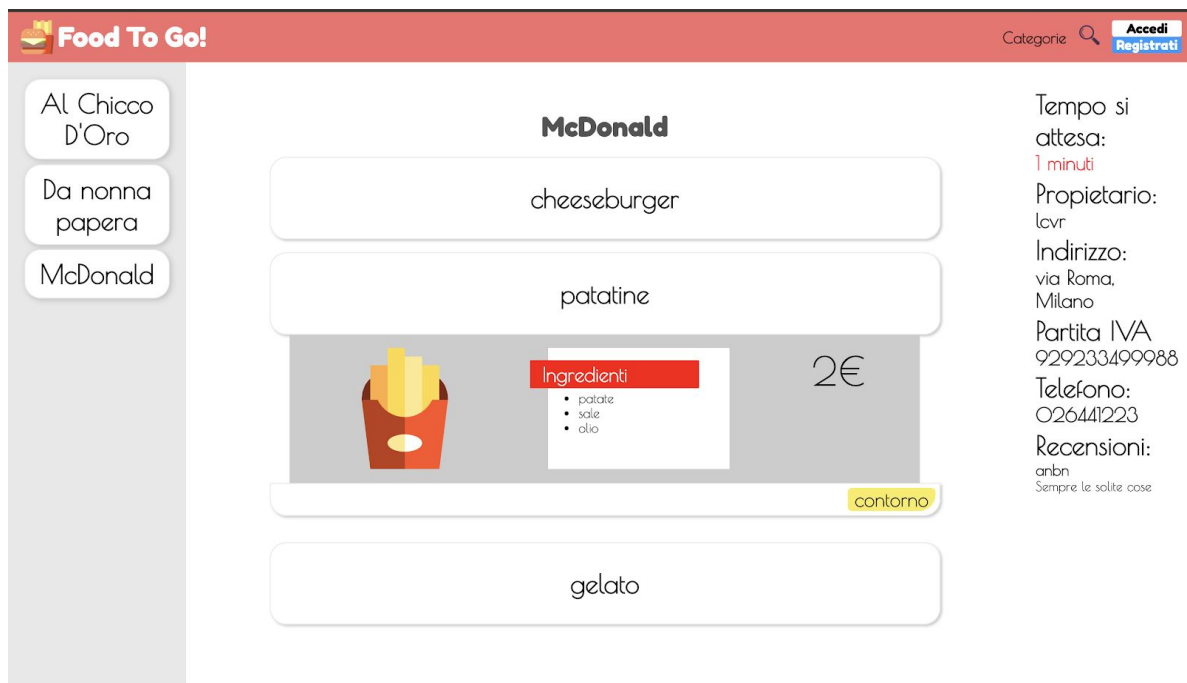
Per la **struttura** e l'organizzazione della pagina è stato invece utilizzato CSS che ha consentito tramite l'uso di classi e identificativi di andare a creare l'intero layout. La scrittura del CSS è stata preceduta da un lavoro di mockup. Sono stati utilizzati dei software di disegno per pensare al design del sito prima dell'effettivo sviluppo di esso. Qui sotto un esempio:



Come vengono gestiti i due scenari principali

Gestione dei menù

Come da specifica, ogni ristoratore può attingere ad una lista predefinita di piatti e può aggiungerli a piacere all'interno del menù dei propri ristoranti.



Ogni ristoratore può avere più di un ristorante che può gestire attraverso la sua pagina personale. Dopo essersi loggato il ristoratore può quindi selezionare uno dei ristoranti di sua proprietà e dalla sezione “Aggiungi un piatto” selezionare la nuova pietanza da un menù a tendina.

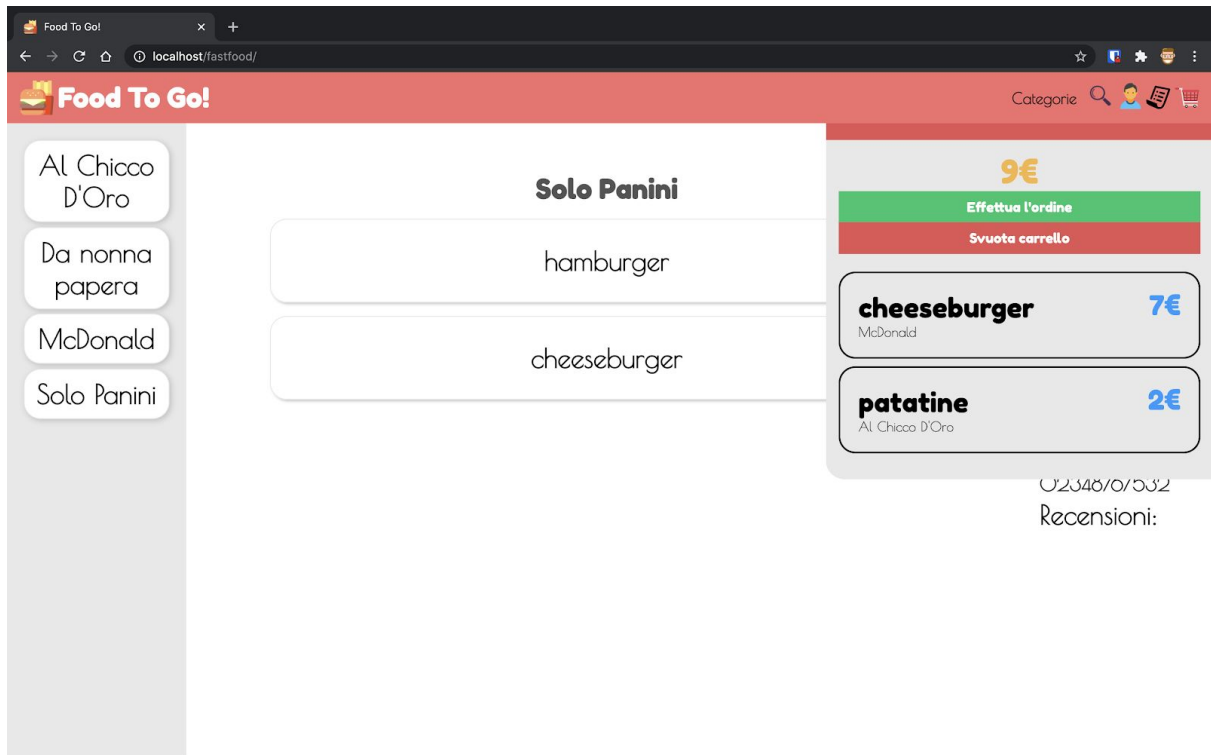
La relazione tra piatti venduti e il ristorante che li vende è definita dal JSON contenuto nella variabile di LocalStorage “vende”.

I menù così generati sono poi visibili sia dagli utenti loggati che da quelli che accedono al sito senza un account. Selezionando uno dei ristoranti dalla spalla a sinistra si potranno visionare i piatti di quel ristorante.

In questa schermata per l'utente basterà selezionare un qualsiasi piatto per poterne visionare le informazioni e, una volta loggato, anche aggiungerlo al carrello.

Gestione degli ordini

Accedendo alla sezione del carrello è possibile effettuare due operazioni sui piatti al suo interno. La prima è l'eliminazione di tutti gli articoli presenti, la seconda è invece l'acquisto degli stessi.



Procedendo verso l'acquisto si viene portati ad una schermata in cui l'utente può scegliere se farsi portare i piatti a casa o se vuole ritirarli direttamente al ristorante. La seconda opzione mostra anche il tempo d'attesa che l'utente troverà in ogni ristorante che vende i piatti che ha ordinato.

Ogni ordine effettuato viene quindi memorizzato e mostrato nello storico degli acquisti fatti dal cliente, il quale è facilmente accessibile dall'icona a forma di lista presente nella barra di navigazione.

Le principali funzionalità richieste

Visualizzazione di informazioni relative ai piatti, ai clienti ai ristoratori registrati e agli acquisti.

Parlando di parole chiave di questo progetto sicuramente tali come “piatto”, “ristoratore”, “cliente” o “ristorante” sono sicuramente tra le più importanti. Non da meno infatti per ognuna di queste è stato pensato uno degli oggetti presenti all’interno del file JSON di cui si è parlato fin’ora. Ogni informazione a loro relativa fa parte dello stato dell’applicazione ed è quindi salvata insieme ad esso.

La visualizzazione di queste informazioni è diversa per ognuna di esse. Per i ristoranti ad esempio, accedendo al loro menù, è stata posizionata sulla destra una sezione in cui sono inserite informazioni quali nome del proprietario, indirizzo, recensioni o tempo d’attesa.

Il tempo d’attesa, in particolare, si è deciso di renderlo pseudocasuale. Infatti ogni volta che si accede al ristorante questo viene generato in maniera casuale ma senza che ci siano troppe differenze di tempo d’attesa da un momento all’altro. La funzione che genera questa informazione usa infatti la probabilità. È molto probabile che il tempo d’attesa cambi di 1 minuto o non cambi affatto, mentre è molto difficile che ci sia uno sbalzo di 5 minuti. Questo è stato fatto per rendere più credibile il tempo d’attesa di ristoranti che in realtà non esistono.

Per quanto riguarda i piatti invece le informazioni sono mostrate ogni volta che ne viene selezionato uno. Caratteristiche come ingredienti, tipologia e prezzo sono tutti contenuti all’interno di un unico contenitore che si apre a tendina selezionando il piatto.

Infine sono facilmente accessibili le informazioni dell’utente. Cliccando infatti sull’icona dell’utente si può accedere ad esse e anche modificarle facilmente cliccando sull’apposito pulsante.

Visualizzazione dei prodotti

Il fulcro dell’applicazione è rappresentato dai menù dei ristoranti. Questi sono una componente fondamentale e, difatti, sono accessibili in qualsiasi momento e in qualsiasi stato il sistema si trovi. La spalla con tutti i ristoranti non scompare mai e basta un click per poter visualizzare tutti i prodotti venduti dai ristoranti registrati.

Accedere ai menù dei vari ristoranti non è l’unica maniera per i clienti di trovare i piatti che cercano. Sono state implementate anche due modalità di ricerca quali la ricerca per nome del piatto e quella per tipologia.

Il modo per trovare i piatti più semplice però rimane la pagina principale che si carica nel momento in cui l’utente effettua il log in. Infatti ogni cliente può selezionare dei piatti a

scelta in base dei quali viene popolata la lista dei suoi piatti preferiti. La lista mostra non solo le pietanze, ma anche i ristoranti che le vendono.

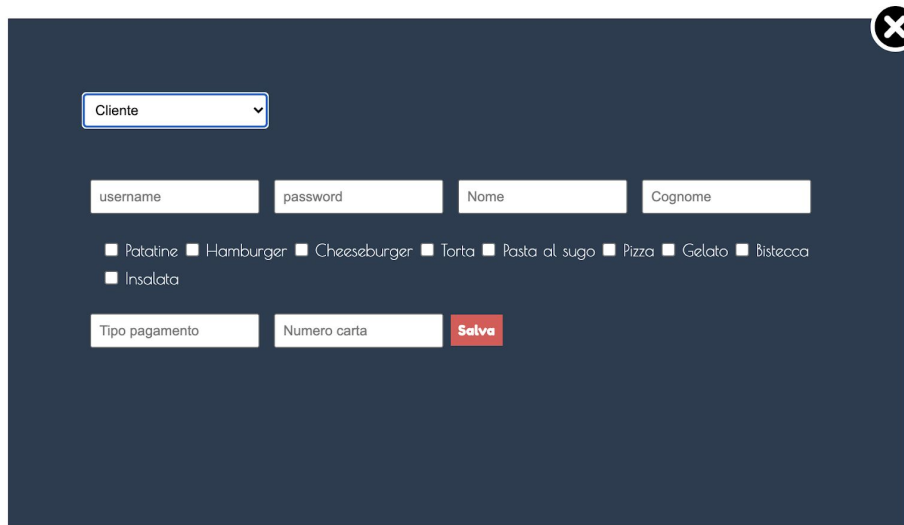
Ricerca dei piatti

Le modalità di ricerca implementate sono fondamentalmente due: ricerca per nome del piatto e quella per tipologia. La prima è implementata come una classica ricerca attraverso un input di testo e un pulsante cerca, mentre la seconda utilizza un menù a tendina sulla barra di navigazione che mostra tutti i tipi di piatto contenuti all'interno della memoria.

Login al sito e acquisto di un piatto a scelta.

L'accesso al sito e la navigazione dei prodotti è stata pensata per non essere vincolata alla creazione di un'utenza. Difatti effettuare un login è necessario solamente nel momento in cui si voglia effettuare un acquisto.

Un utente che non possiede delle credenziali di accesso può crearne di nuove cliccando il tasto registrati sulla barra di navigazione. Questo consente di aprire una finestra in cui deciderà se vorrà creare un'utenza da Cliente o da Ristoratore. Una volta selezionato la tipologia dell'utente si potranno compilare le proprie informazioni personali.



Cliente

username password Nome Cognome

☐ Patatine ☐ Hamburger ☐ Cheeseburger ☐ Torta ☐ Pasta al sugo ☐ Pizza ☐ Gelato ☐ bistecca
☐ Insalata

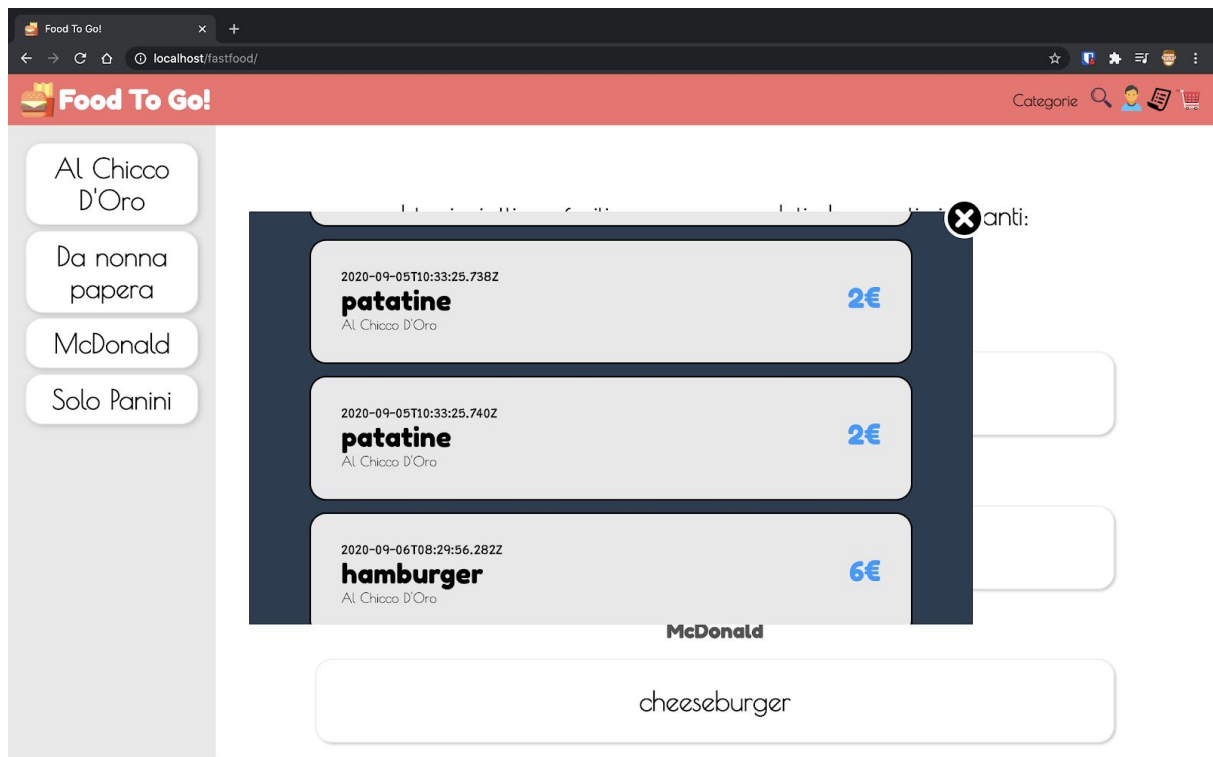
Tipo pagamento Numero carta **Salva**

Un utente in possesso di un account invece potrà accedervi cliccando il pulsante Accedi vicino a Registrati. Selezionata la tipologia di utente e inseriti username e password corretti l'applicazione consentirà all'utente di inserire i piatti nel carrello e di effettuare gli acquisti.

Tutte le utenze con le loro informazioni sono salvate all'interno degli oggetti "Clienti" e "Ristoratori".

Visualizzazione degli acquisti presenti e passati per un cliente.

Ogni volta che un ordine viene effettuato l'acquisto viene memorizzato nel oggetto "acquisto" del file JSON. Lo stesso oggetto è utilizzato dalla pagina che mostra lo storico degli ordini.



Cliccando sull'icona della lista sulla barra di navigazione infatti si accede ad una finestra che mostra lo storico di ogni acquisto fatto dal cliente, mostrandone il giorno e l'ora, il piatto acquistato e il prezzo che quel piatto aveva in quel momento.

Conclusioni e possibili espansioni future

Le funzionalità implementate dall'applicazione ne consentono l'utilizzo da parte degli utenti per ricercare e acquistare piatti dai vari ristoranti. Affianco a queste però potrebbero essere sviluppate anche ulteriori funzioni.

- Potrebbero essere aggiunte numerose animazioni per le transizioni delle finestre, dei menu a tendina o per la navigazione tra ristoranti.
- La funzione per la generazione del tempo d'attesa potrebbe essere modificata affinché non cambi solamente quando viene selezionato un ristorante, ma che venga chiamata con un timer senza che l'utente faccia nulla.
- Aggiungere la possibilità di eliminare un elemento alla volta dal carrello

Insieme a queste piccole modifiche si potrebbe pensare anche di sostituire il file JSON con un vero e proprio database, mantenendo il paradigma di comunicazione implementato con lo scambio di stringhe JSON, ma rendendo in questo modo più efficienti le operazioni.