

# FAST FOOD

## 1 Premessa

La specifica del problema che deve essere affrontato è per sua natura incompleta e può essere ambigua. Il candidato deve essere in grado di valutare eventuali soluzioni alternative e giustificare le scelte implementative adottate. Le motivazioni delle scelte fatte vanno inoltre documentate nel progetto. Il lavoro consiste di cinque fasi principali: *i)* analisi dei requisiti; *ii)* identificazione delle funzionalità da sviluppare; *iii)* progettazione della struttura e della presentazione delle pagine web; *iv)* progettazione del backend software per la gestione delle informazioni statiche e dinamiche; *v)* implementazione dell'applicazione stessa. Lo svolgimento del progetto è una prova d'esame da svolgere individualmente.

## 2 Requisiti

Il progetto si pone l'obiettivo di sviluppare l'applicazione web **FastFood** che implementa un sito di ordinazione online per ristoranti. **FastFood** gestisce il processo di ordinazione all'interno di un ristorante di una catena di Fast Food ed 'e composto di 4 macro-scenari principali:

- Gestione del profilo dell'utente (registrazione e scelta della tipologia, modifica dei dati personali, rimozione del profilo, etc.);
- Gestione del Ristorante;
- Gestione degli Ordini;
- Gestione delle consegne.

Di seguito sono analizzate in dettaglio le caratteristiche dei 4 macro-scenari introdotti.

Il primo macro-scenario (**Gestione del profilo dell'utente**) consiste nella gestione classica di un profilo utente, con l'acquisizione dei dati principali, la loro modifica ed eventualmente anche la rimozione del profilo stesso. Esistono due tipologie di utenti: ristoratore e cliente.

L'applicazione deve prevedere una fase di registrazione utente dove verranno collezionate informazioni quali ad esempio nome utente, indirizzo

email, password, e soprattutto la scelta della tipologia di Utente (Ristoratore o Cliente) .

Il secondo macro-scenario (**Gestione del Ristorante**) consiste nella gestione del ristorante dalla gestione delle informazioni principali, quali nome, luogo etc sino alla scelta del menu in vendita. Gli utenti ristoratori devono potersi collegare all'applicazione, modificare i propri dati/preferenze e cancellarsi. Per ogni ristorante si dovranno gestire informazioni quali nome del ristorante, numero di telefono, partita iva, indirizzo, etc. Un ristorante, una volta registrato, può collegarsi all'applicazione e inserire i prodotti in vendita selezionandoli da una lista comune a tutti i ristoranti (l'elenco di tutti i piatti disponibile è fornito in allegato tramite il file meal.json). Per ogni prodotto (piatto) in vendita dovranno essere gestiti gli ingredienti da cui composto e dovrà essere presente una foto illustrativa. Si dovranno inoltre gestire informazioni quali tipologia, nome e prezzo. Il ristorante ha la possibilità inoltre di inserire piatti scelti da lui e quindi ha la possibilità per ognuno di specificare ogni possibile informazione sopra indicata

Il terzo macro-scenario (**Gestione degli Ordini**) consiste nella classica gestione delle attività dei clienti all'interno di un'applicazione di ordini online. I clienti devono potersi registrare, modificare i propri dati/preferenze, cancellarsi. Per ogni utente si devono memorizzare informazioni personali (ad es., nome, cognome) e informazioni relative all'account generato. Al cliente vengono anche associate informazioni riguardanti il meccanismo di pagamento dei prodotti (ad es., carta di credito o carta prepagata). Inoltre, in fase di registrazione gli utenti possono selezionare delle preferenze per la personalizzazione dei servizi (ad es., offerte speciali in bacheca per la tipologia di prodotto preferito). In questo scenario, gli utenti registrati possono fare login al sito, selezionare uno o più piatti di interesse, aggiungerli al carrello e concludere l'acquisto dei piatti nel carrello. facendo la consegna. Il flusso di stato di un ordine è: i) ordinato ii) in preparazione iv) in consegna v) consegnato Il ritiro dell'ordine può essere fatto in due modalità:

- presso il ristorante. In questo caso deve essere fornito un tempo di attesa che prenda in considerazione gli ordini in coda presso il ristorante. Quando un ordine è pronto, il ristorante segnala che esso è stato preparato e viene rimosso dalla coda delle preparazioni di quel ristorante e passa direttamente nello stato di consegnato.
- consegna a domicilio (Solo per gruppi di due persone). Un utente in fase di ordine chiede la consegna a domicilio e quindi indica il luogo di consegna. In questo caso si deve inserire anche un costo di consegna

dipendente dalla distanza del ristorante e del luogo di consegna stesso. In questo caso utilizzando le API di OpenStreetMap va calcolata la distanza dal ristorante al punto di consegna e calcolato il costo in base ai KM da percorrere per la consegna.

L'ultimo scenario (**Gestione delle consegne**) prevede che l'utente finale quando riceve l'ordine lo segnala e quindi l'ordine stesso passa dallo stato di inconsegna allo stato di consegnato.

## 2.1 Operazioni da svolgere

Le operazioni base che devono essere presentate al momento della discussione del progetto sono le seguenti:

- Registrazione e login al sito;
- Visualizzazione di informazioni relative ai piatti, ai clienti e ai ristoratori registrati, agli acquisti;
- Ricerca dei ristoranti (le ricerche di base previste sono: luogo, nome del ristorante)
- Ricerca dei piatti (le ricerche di base previste sono: tipologia, nome, prezzo);
- Login al sito e ordine di uno o più piatti a scelta.;
- Gestione consegne
- Visualizzazione statistiche per ristorante;
- Visualizzazione degli acquisti presenti e passati per un cliente.

Le operazioni obbligatorie per i gruppi che devono essere presentate al momento della discussione del progetto sono le seguenti:

- ricerca del ristorante per piatto
- ricerca di piatti per ingredienti
- ricerca di piatti per allergie
- consegna a domicilio

Allo startup dell'applicazione, tutti i dati necessari per la fase iniziale devono essere disponibili nel backend che verrà utilizzato dall'applicazione web (come sopra descritto è prevista una fase di setup in cui verrà caricato il JSON fornito).

Operazioni e funzionalità aggiuntive possono essere implementate a piacere. Le pagine web devono essere implementate utilizzando HTML5, CSS3 e JavaScript, e devono seguire un paradigma di separazione tra la struttura (HTML5) e la rappresentazione (CSS3) della pagina web.

La parte di Backend deve essere realizzato usando NodeJS e MongoDB. E' necessario fornire anche lo swagger delle API che vengono implementate.

Le informazioni visualizzate all'interno delle pagine dell'applicazione web devono essere memorizzate e accedute tramite il paradigma REST e fornite tramite il backend sviluppat. Devono perciò essere previste operazioni per la presentazione e modifica delle informazioni.

### 3 Informazioni Generali

Il progetto è valido per l'anno accademico 2024/2025 (L'ultimo appello utile per la consegna è quello di Settembre 2026). Prima di iniziare il progetto bisogna inviare una mail a [valerio.bellandi@unimi.it](mailto:valerio.bellandi@unimi.it) con la specifica dei componenti del gruppo (anche per gruppi composti da una sola persona). Una volta terminato, il progetto deve essere caricato all'indirizzo [upload.di.unimi.it](http://upload.di.unimi.it). È necessario presentare:

1. Il codice sorgente.
2. Una relazione dettagliata (in formato pdf) che illustra la struttura e presentazione del sito web, come sono state realizzate le operazioni richieste e le scelte implementative che sono state fatte.
3. Delle prove di funzionamento, consistenti in una serie di schermate dimostrative comprovanti la corretta esecuzione delle operazioni previste.

Per ogni ulteriore chiarimento: [valerio.bellandi@unimi.it](mailto:valerio.bellandi@unimi.it), [antongia-como.polimeno@unimi.it](mailto:antongia-como.polimeno@unimi.it)