



MERENDERO

Relazione di Analisi, Sviluppo e Test software



A cura di Concettoni Kevin, Grussu Andrea e Jeferson Morales Mariciano
4°CI A.S. 2019/20

Indice

Indice.....	1
Abstract	2
Analisi.....	2
Progettazione.....	3
Sviluppo	6
Test.....	8
Conclusione.....	8

Abstract

Applicativo gestionale delle prenotazioni delle merende del bar della scuola con salvataggio dei dati su database e generazione dinamica dei controlli.

Analisi

Il software “Merendero” ha lo scopo di fornire al bar della scuola dell’istituto Volterra Elia un metodo funzionale per la gestione delle ordinazioni delle merende, in modo da minimizzare gli assembramenti nella zona del bar in rispetto delle nuove norme sul distanziamento dovute al virus COVID19.

Il software, all’avvio, offre un’interfaccia intuitiva e user-friendly con stile minimal, ultimamente di tendenza, e una palette di colori chiara che risulta in un armonioso contrasto che rende leggibile tutto ciò che è scritto su schermo, a discapito dell’eleganza ma molto spesso poco chiara modalità notte. La schermata principale di login permette, inserendo username e password, di accedere alle interfacce dedicate all’admin, barista e utente.

MERENDERO LOGIN

Username

Password



LOGIN

- **Admin:** permette la gestione degli account e dei prodotti all’interno dell’applicazione. Per quanto riguarda la gestione dei prodotti gli è consentito CRUD (create, read, update, delete), di fatto l’admin è in grado di apportare modifiche alla lista dei prodotti disponibili.

- **Barista:** il barista gestisce le ordinazioni ricevute dai clienti, rimuovendo un quantitativo di prodotti dalla dispensa, ed è in grado di rifornirsi alla semplice richiesta.

- **Cliente:** Seleziona i prodotti che vuole ordinare con le rispettive quantità dalla schermata dell’ordinazione, semplicemente cliccano sul prodotto desiderato. Una volta confermata la prenotazione, riceve una notifica sul suo spazio personale.

Form di login

Progettazione

Il software è progettato nel linguaggio di programmazione C# su framework .Net provvisto da Microsoft e quindi propenso ad un sistema operativo di ambiente Windows. Il DBMS utilizzato è un MySql. La programmazione era orientata agli oggetti con paradigma OOP. Per l'ideazione e la strutturazione della GUI del software abbiamo seguito il principio di costruzione di una buona UX, il quale enuncia che minore è il numero di click necessari per portare a termine una determinata azione di utilizzo fondamentale e basilare, migliore sarà il riscontro dell'applicativo con l'utenza. A livello di layout, l'applicazione è basata su una Form principale la quale mostra i bottoni dedicati alle varie funzioni delle entità (admin, barista, cliente) inoltre sono presenti icone rappresentanti i vari social (Facebook, Instagram) e il sito della scuola.

Una volta che una persona avrà acceduto:

- Nel caso dell'admin: troverà tre bottoni, aggiunta e modifica prodotti nel menu, rimozione prodotti dal menu e gestione account

Merendero

LOGIN

Aggiungi Prodotto

Elimina Prodotto

Gestione Account

Loggato come: admin

Aggiungi Prodotto

Nome
sprite

Descrizione
desc sprite

Costo
1,00

Categoria
BIBITA

Anteprima Foto

Seleziona File

PULISCI CAMPI

AGGIUNGI

MODIFICA

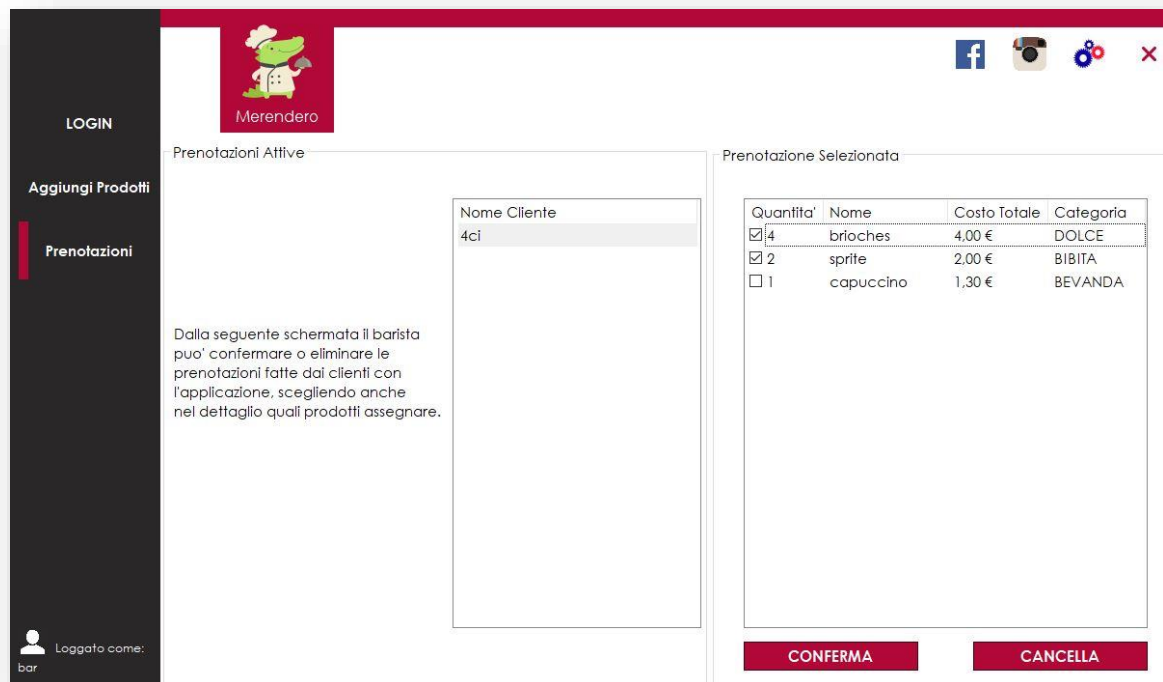
Dalla seguente schermata l'admin può aggiungere un prodotto, includendolo alla lista del catalogo dei prodotti disponibili e offerti dal bar.

ATTENZIONE: Il nome del prodotto non è modificabile

Nome	Descrizione	Costo	Categoria
brioche	desc brlo	1,00 €	DOLCE
sprite	desc sprite	1,00 €	BIBITA

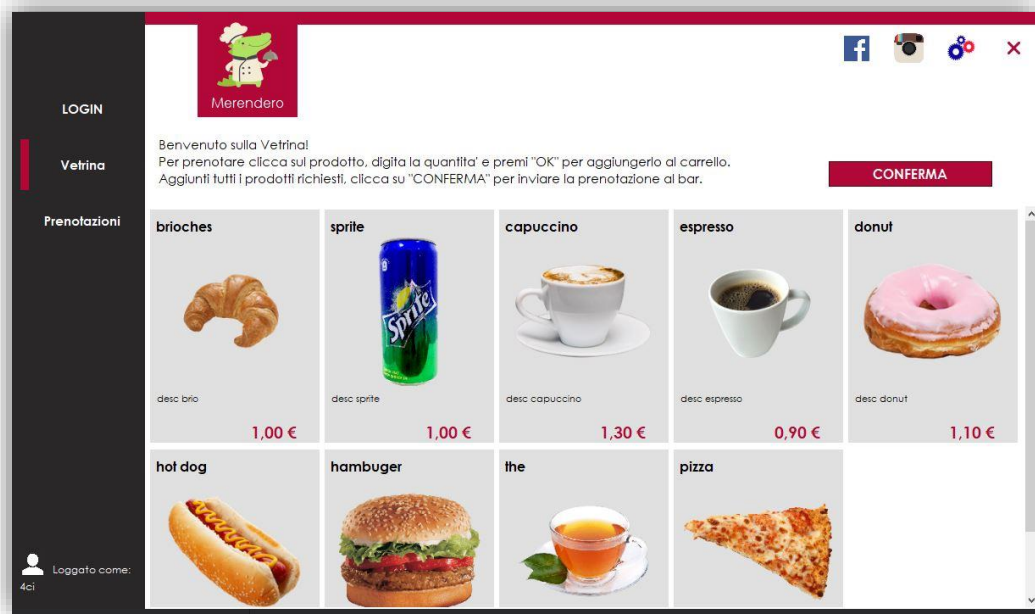
Sezione aggiungi prodotto dell'amministratore

- Nel caso del barista: troverà due bottoni, aggiunta prodotti e gestione prenotazioni



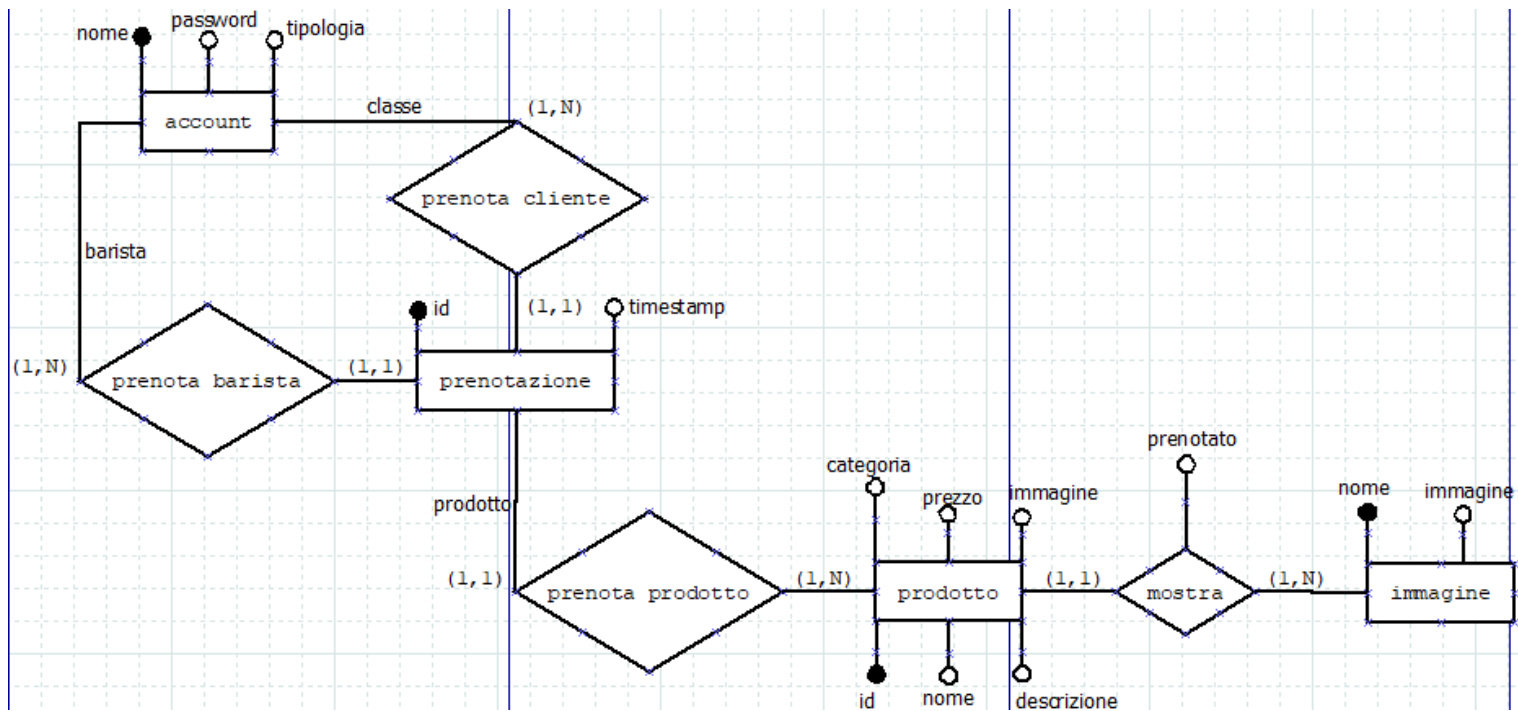
Sezione Prenotazioni del Bar

- Nel caso del cliente: troverà due bottoni, uno con la vetrina dei prodotti e un secondo che riepiloga i vari ordini effettuati.

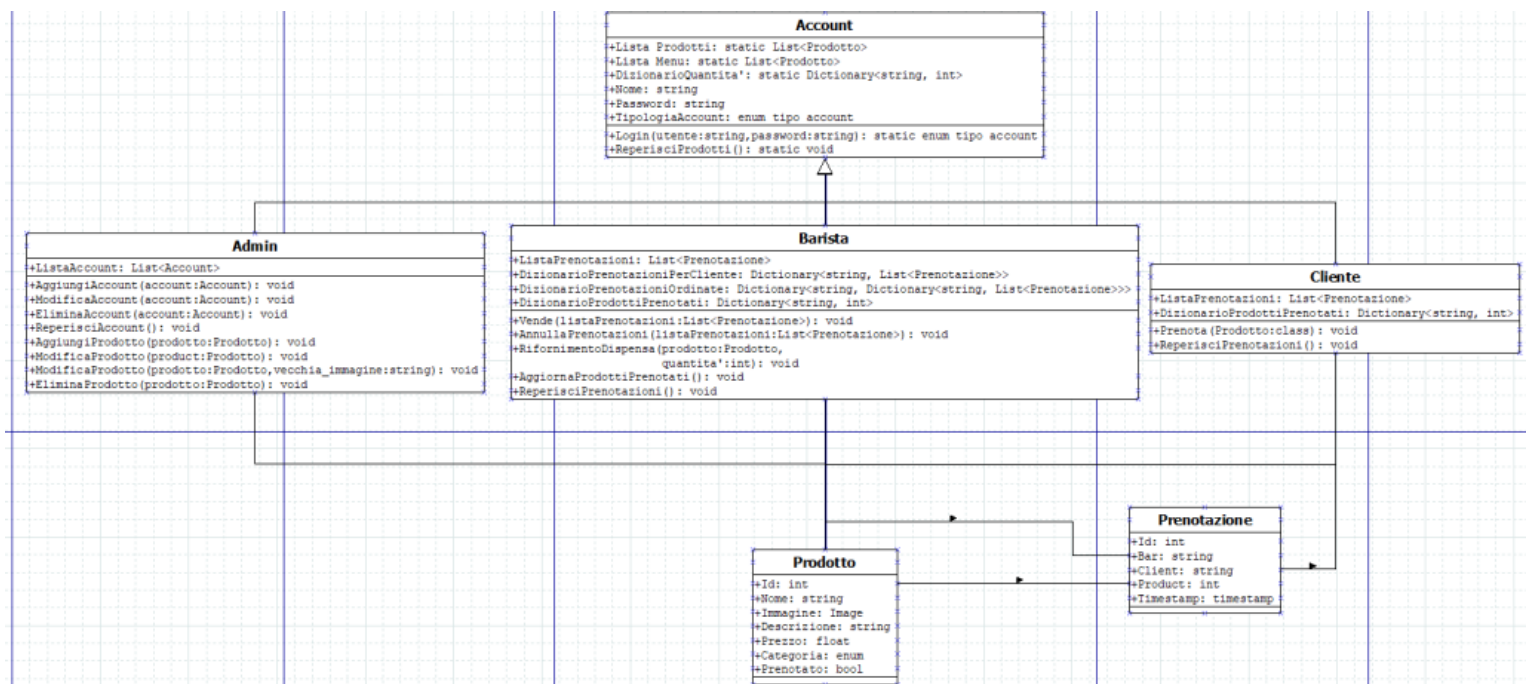


Sezione vetrina del Cliente

Sviluppo



Modello E/R rifinito



Modello UML rifinito

Come prima cosa, abbiamo progettato le classi con il diagramma UML e il database con lo schema delle relazioni E/R, cercando di rendere lo sviluppo del progetto semplice ma allo stesso tempo efficiente, valutando tutte le possibili sfumature senza tralasciare dettagli importanti. Nonostante lo sforzo di comprendere nei nostri primi schemi UML ed E/R la

maggior parte dei dettagli possibile, il progetto con lo sviluppo è andato evolvendosi e mutando alcuni aspetti, ampliandosi per lo più. Per questo non risulta avere ogni caratteristica di quelle pensate durante l'analisi bensì spesso di più e con il taglio di feature di scarso rilievo come la comunicazione tra admin e bar per le richieste. Difatti è sempre meglio lasciare svolgere il lavoro a chi di competenza, in questo caso ironicamente potremmo immaginare Whatsapp per i promemoria bar-admin. Si è quindi rivelato fondamentale ridefinire i diagrammi delle classi UML e lo schema relazionale E/R a ciò che l'applicazione stava diventando. Ad inizio paragrafo difatti si trovano i diagrammi ridefiniti a progetto terminato. Per lo sviluppo di questo programma siamo partiti cercando di dare forma a quella che è la parte grafica del progetto disegnando le varie Form e UserControl per poi traslarle nel progetto vero e proprio. In seguito, il gruppo si è concentrato sulla scelta della palette dei colori i quali, seguendo degli studi, provocano risposte emotive positive e influenzano inconsciamente l'utente, ciò è chiamato "psicologia del colore".

Successivamente, pensato lo scheletro e la struttura del software, abbiamo iniziato la stesura del codice. In primo luogo, abbiamo iniziato con la creazione delle classi, fondamentali per il funzionamento del programma in quanto cuore pulsante che gestisce le entità. Allo stesso tempo, mentre si creavano le classi, abbiamo provveduto alla creazione del database e delle proprie tabelle seguendo il modello E/R sviluppato, indispensabile per la gestione dei dati in quanto sarebbe stato impensabile manovrare tutto quel quantitativo di informazioni su file. Una volta creata la base di dati e le classi, abbiamo proseguito gestendo le tre entità presenti all'interno del nostro software (Cliente, Barista e Admin). La scelta degli User Control è prettamente stilistica e di comodità in quanto ci sembrava poco efficiente gestire innumerevoli Form, bensì nettamente più rapido e funzionale utilizzare una singola Form per poi andargli ad abbinare un diverso User Control ogni qual volta che si compie una determinata azione. Gli User Control hanno già del layout preimpostato per essere mostrati su form ed ereditano parte degli utili più indispensabili dei comuni controlli disponibili.

Abbiamo voluto sfruttare al massimo la programmazione ad oggetti e le strutture dati, le quali sono tornate utili qual ora abbiamo realizzato un dizionario che tiene in ordine le prenotazioni per cliente e successivamente per nome prodotto. Inoltre, tali strutture dati hanno aumentato la semplicità del codice scritto rendendolo decisamente più leggibile. Le strutture dati in comune tra gli account Cliente, Bar e Amministratore sono stati messi successivamente nella

classe padre Account nel quale sono stati piazzati sotto visibilità pubblica a carattere statico. Di conseguenza c'è stata una saggia e meditata implementazione della classe padre e delle caratteristiche dell'eredità per la gestione ad esempio delle liste di account impiegando tecniche di downcasting e upcasting per agevolare la compatibilità delle liste.

Test

Una volta aver completato la fase di scrittura del codice, abbiamo iniziato la fase più importante e tediosa di tutto il progetto: la fase di debug. Questo periodo delicato del progetto ci ha portato via circa 4 giorni di lavoro intenso con la scoperta che, nonostante la portata importante del progetto, il maggiore del lavoro era la risoluzione di questi errori nella logica del programma. Alcuni bug sono stati riscontrati anche durante lo sviluppo del software ma la maggior parte sono stati identificati proprio durante questa fase di Alpha Test. Non abbiamo riscontrato grandi bug che facessero terminare l'applicazione bensì dei piccoli errori che interferivano con il corretto funzionamento del software. Di seguito una lista degli errori trovati durante il debugging:

- Lo scontrino non veniva resettato anche se si accedeva con un'altra classe
- Quando si provava ad aggiungere un prodotto senza immagine si generava un errore nel Database
- Quando si provava a rimuovere un prodotto con l'admin non era possibile farlo perché la chiave non era presente nel dizionario
- Irreperibilità dei prodotti prenotati da un account non confermati, di cui l'account di seguito eliminato
- Prodotto prenotabile a quantità non disponibile sulla dispensa bar
- Errato conteggio prodotti prenotati causavano quantità negative sulla dispensa del bar
- Errata estrapolazione delle prenotazioni ordinate per cliente e per prodotto su dizionario

Conclusione

Dopo alcune settimane di lavoro possiamo dire che questa è stata un'esperienza assolutamente interessante e formativa. Non bisogna negare che ci sono state alcune

difficoltà quale il fatto di lavorare a distanza e il rispetto dei tempi di consegna i quali, però, grazie ad una proroga, sono stati rispettati. Questo progetto è stato molto utile per capire e adattarsi a futuri lavori di gruppo, anche in ottica lavoro, dove ogni componente ha cercato di dare il proprio contributo venendo aiutato dagli altri membri in caso di difficoltà. In conclusione, possiamo ritenerci soddisfatti del prodotto finale che presenta delle caratteristiche molto interessanti e innovative e possiamo affermare senz'ombra di dubbio, che questa attività ci ha saputo coinvolgere, intrattenere e mettere in gioco con tutte le conoscenze di programmazione acquisite durante il corso degli anni di studi superiori.