

Tesina progetto Tecnologie Web

Corso di laurea in Informatica A.A 2019/2020

Nicholas Bernardoni

TRACCIA

Gestione di una pizzeria ristorante per un possibile uso nel campo ristorativo.

Il progetto consiste nella realizzazione di un sito web che gestisca vari tipi di servizi e in cui gli utenti possono effettuare la registrazione oppure decidere di rimanere anonimi.

I servizi offerti agli **utenti registrati** sono i seguenti:

- possibilità di ordinare d'asporto, selezionando i prodotti che si desidera;
- possibilità di vedere le offerte sui cibi e sulle bevande per un determinato periodo e con il relativo sconto;
- possibilità di prenotare un tavolo, con i campi del nominativo già compilati;
- possibilità di inserire una recensione al ristorante, che tutti gli utenti possono vedere (anche quelli anonimi), ma che solo altri utenti registrati possono commentare;
- possibilità di visitare il proprio profilo, visualizzando le proprie informazioni personali, compresi lo storico delle prenotazioni, le prenotazioni previste per il futuro (con la possibilità di eliminarle), e lo storico degli ordini d'asporto. È possibile, inoltre, modificare le informazioni personali qualora lo si voglia.

I servizi per gli **utenti anonimi**, invece, sono:

- possibilità di visitare la pagina relativa al menù delle bevande e dei cibi (qualora ci fossero);
- possibilità di registrazione al sito, inserendo le proprie informazioni personali, tra cui l'indirizzo;
- possibilità di vedere le recensioni degli utenti registrati;
- possibilità di prenotare un tavolo, senza che alcun campo sia compilato (come avveniva per gli utenti registrati).

Infine, viene gestita una sezione dedicata agli **utenti ristoratori**. Questi utenti possono:

- prenotare i tavoli per i clienti che, per esempio, contattano telefonicamente il ristorante. Non possono fare prenotazioni d'asporto, in quanto si è pensata questa funzionalità solamente per gli utenti registrati al sito;
- aggiungere cibi, bevande e ingredienti al menù, potendo inoltre modificarli ed eliminarli;
- creare offerte personalizzate per i singoli cibi e le singole bevande;
- vedere le recensioni e commentarle;

- visualizzare le informazioni relative alle prenotazioni dei tavoli con possibilità di eliminarle, e tutte le prenotazioni d'asporto di quel giorno.

Un utente registrato che effettua 15 prenotazioni di un tavolo al ristorante, ha diritto ad uno sconto del 15% sul prezzo totale della cena, che verrà scontato dal ristorante al momento del pagamento.

Nella pagina iniziale sono presenti una piccola descrizione a seconda di che tipo di utente si è, con tutti i button e i link relativi per le varie funzionalità descritte. Sempre nella pagina iniziale è stato aggiunto uno slideshow come sfondo.

DIAGRAMMA DEGLI USE CASE

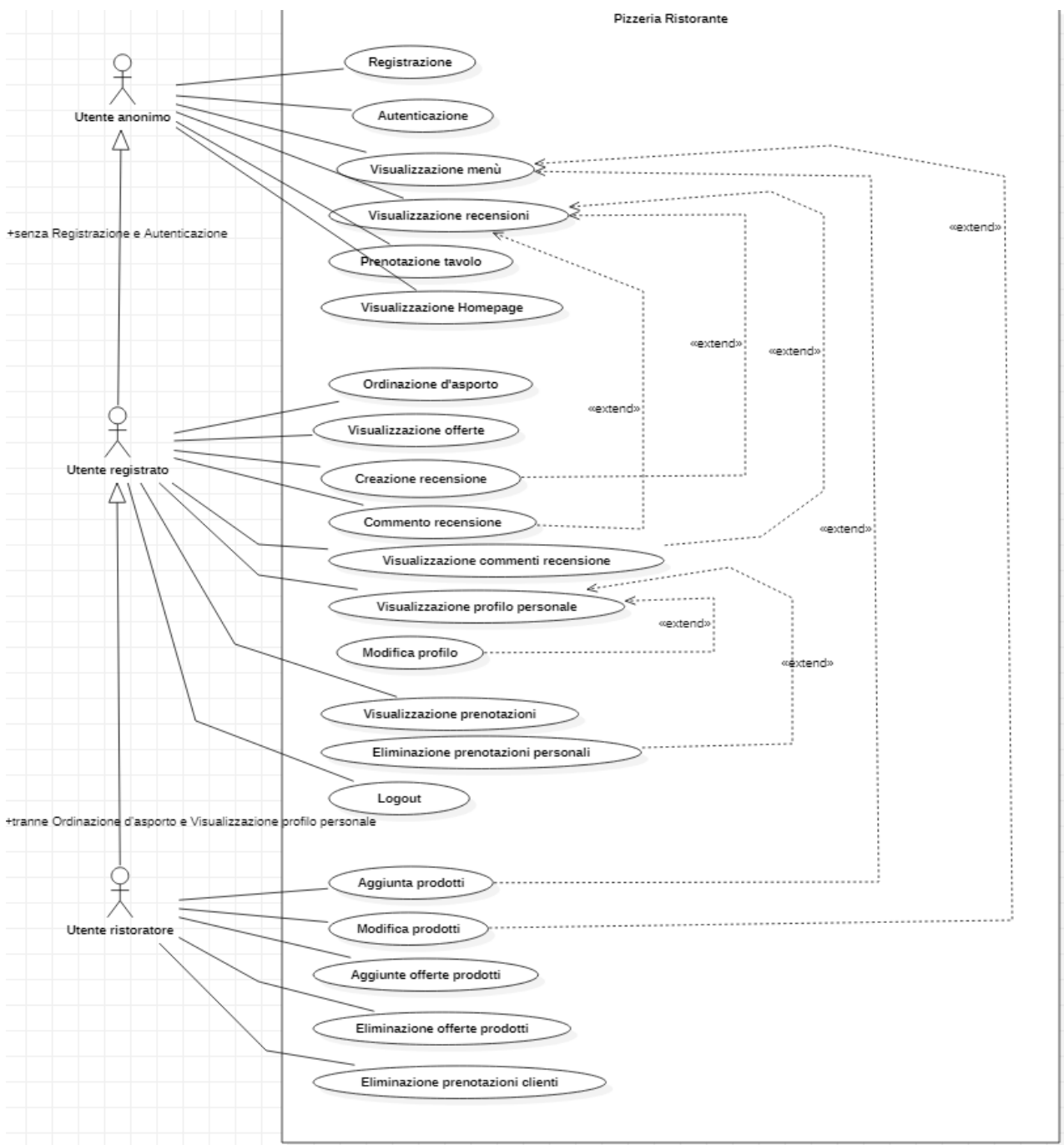
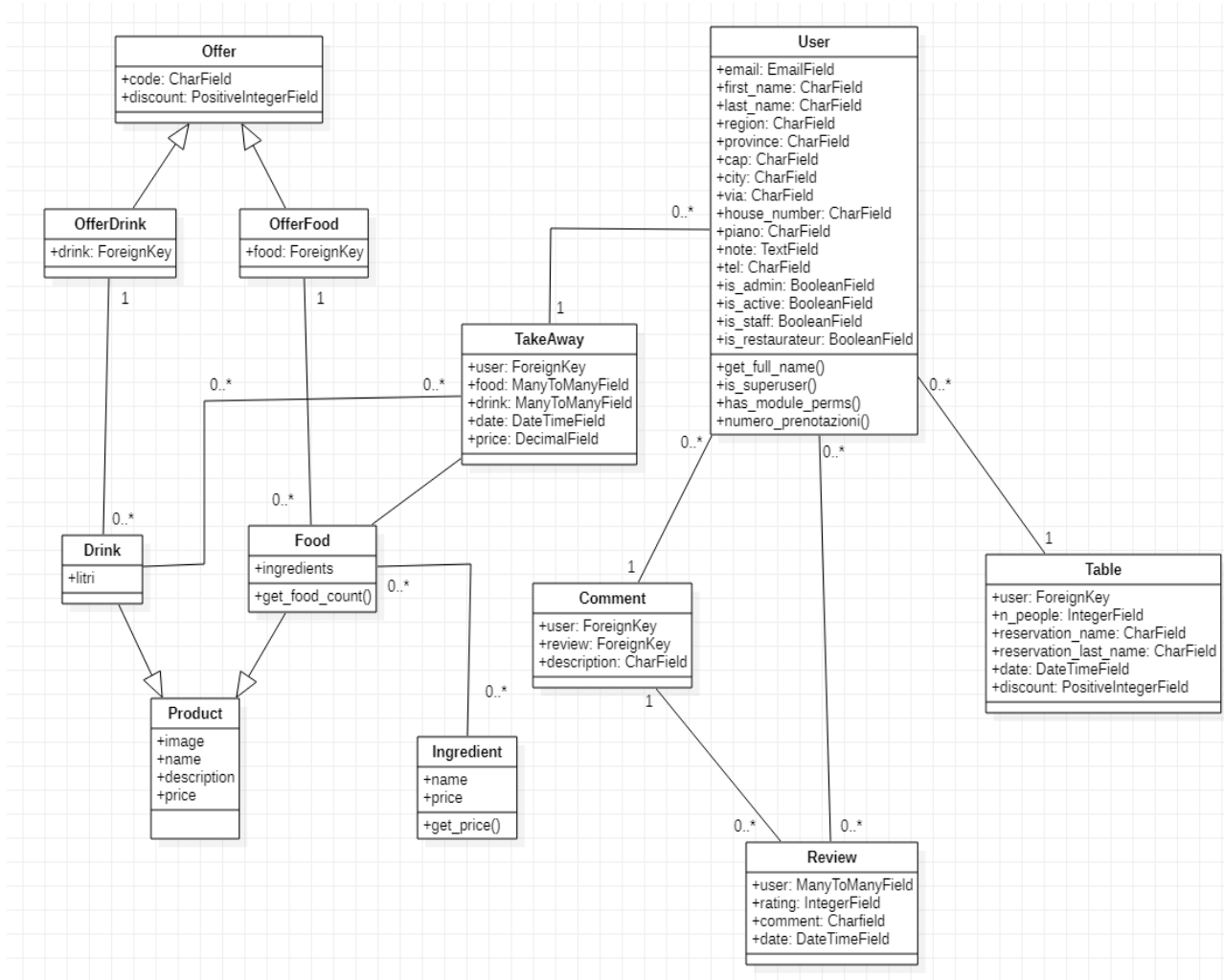


DIAGRAMMA DELLE CLASSI E DESCRIZIONE MODELLI

Il database è gestito nel seguente modo:



Offer: rappresenta il model astratto per le offerte.

OfferDrink: rappresenta le offerte che vengono create per una determinata bevanda.

OfferFood: rappresenta le offerte che vengono create per un determinato cibo.

Drink: rappresenta la bevanda che è presente nel menù e a cui può essere associata ad un'offerta con un certo codice e un certo sconto.

Food: rappresenta il cibo che viene mostrato nel menù e che può essere associato anche questo ad una determinata offerta.

Ingredient: questo model è collegato al fatto che un determinato cibo contiene un certo numero di ingredienti con un certo prezzo e vengono mostrati in una lista.

User: rappresenta l'utente che si iscrive al sito ed è usato durante l'autenticazione. Contiene un attributo booleano (is_restaurateur) che consente di distinguere se l'utente è un cliente o un ristoratore.

Review: rappresenta il fatto che un determinato utente può effettuare una recensione riguardante il ristorante, dato una valutazione da 1 a 5 stelle e potendo lasciare anche un commento nel caso lo desiderasse.

Comment: rappresenta il commento che gli utenti possono lasciare sotto le recensioni di altri utenti.

Table: rappresenta il fatto che un utente, autenticato o non, possa effettuare una prenotazione di un tavolo per un certo numero di persone, ad un certo orario ed ad una certa ora. In caso un utente registrato effettui 15 prenotazioni di un tavolo, questo riceve automaticamente uno sconto del 15% che verrà poi sottratto al prezzo della cena.

TakeAway: rappresenta la possibilità di un utente di poter effettuare prenotazioni d'asporto di determinati prodotti con un sovrapprezzo fisso la consegna.

Oltre agli utenti richiesti e a quelli presenti nel diagramma, c'è un utente aggiuntivo, l'**admin**, che ha la possibilità di accedere alla zona di amministrazione e gestire l'applicativo, assegnando anche i permessi agli utenti.

Nel progetto si è deciso di mantenere la homepage nella directory principale (restaurant). Le altre funzionalità sono state divise in tre app: **users**, **products** e **offers**.

La prima si occupa della creazione degli utenti e delle azioni che possono eseguire, come la prenotazione di un tavolo, di un ordine d'asporto, della scrittura di una recensione e dei commenti relativi a questa. Una feature interessante che è stata implementata, è la gestione degli ordini di asporto: si è deciso di utilizzare l'url per passare i prodotti ad un'altra view, aspettando che l'utente confermi la sua scelta per l'ordine d'asporto, senza aggiungere gli elementi al database. Una volta che l'utente ha dato conferma, l'ordine viene confermato, salvando i dati nel modello. Altro meccanismo interessante che è stato utilizzato all'interno di questa app è il meccanismo riguardante le recensioni: un utente può decidere di lasciare una sola recensione al ristorante e questa viene aggiunta dinamicamente al sito attraverso meccanismi ajax che permettono l'aggiornamento dinamico della pagina web.

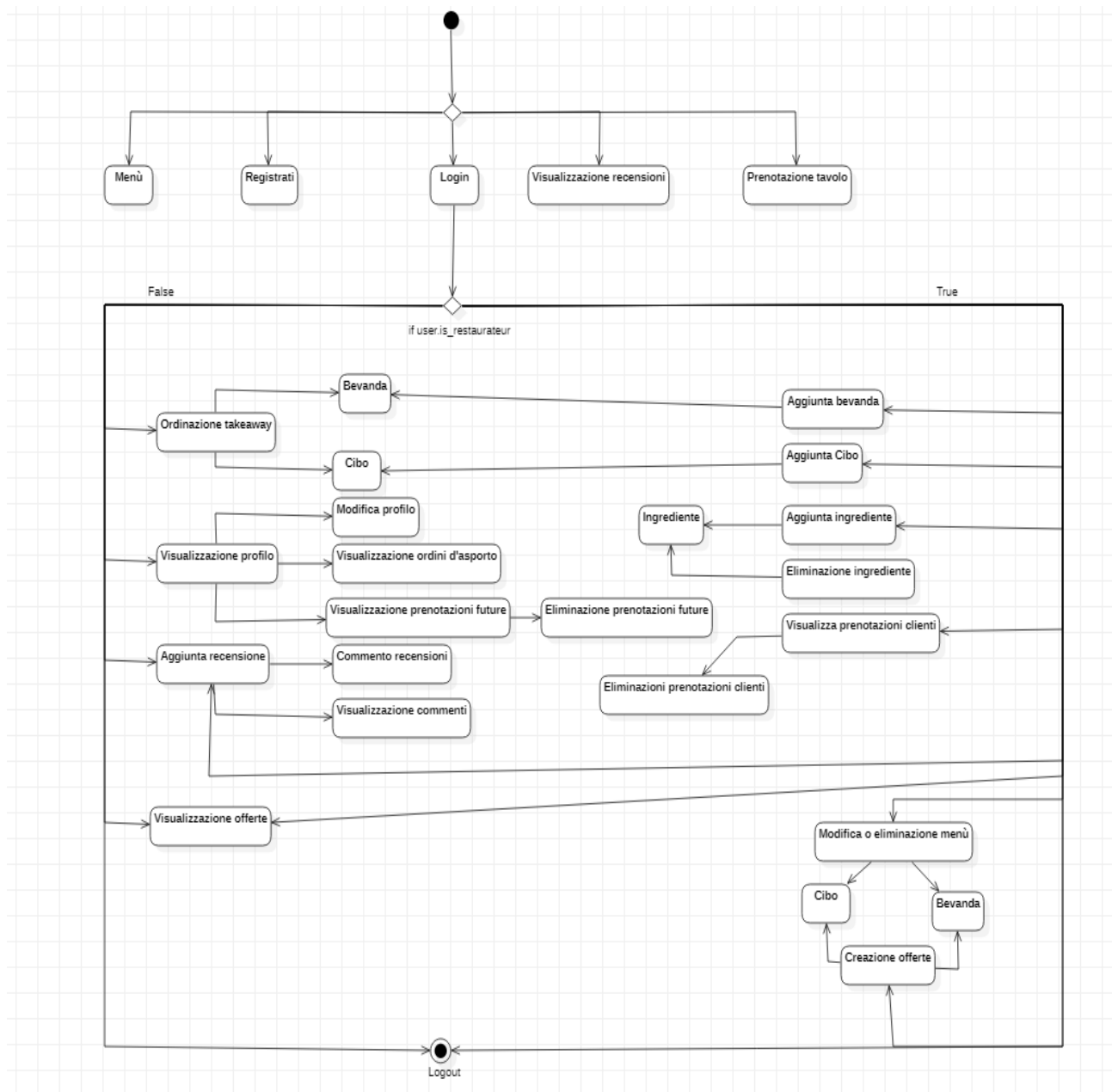
La seconda contiene tutte le features riguardanti i prodotti, come la creazione, l'eliminazione o la modifica di questi.

L'ultima app, prevede la gestione delle offerte dei prodotti che sono stati aggiunti al progetto e di cui gli utenti possono usufruire.

Le foto aggiunte al sito vengono salvate nella cartella "media/img/" che è una sottodirectory della cartella principale.

Si è optato per questa suddivisione delle applicazioni per separare logicamente le varie applicazioni, in modo tale da rendere il codice più chiaro e facile da modificare.

DIAGRAMMA DI ATTIVITA'



TECNOLOGIE UTILIZZATE

Django: si è scelto di utilizzare all'interno di questo progetto Django, un framework MVC (Model View Controller) per lo sviluppo Web ad alto livello con molte caratteristiche che lo rendono uno dei più utilizzati. Prima tra tutte, lo stesso paradigma MVC permette la suddivisione del codice in una struttura modulare. Inoltre, è scritto in Python, un linguaggio molto versatile e di facile apprendimento iniziale rispetto a linguaggi più classici. Altre caratteristiche di Django sono:

- presenza di numerosi package che permettono di importare molte funzioni, risparmiando sulla scrittura di codice e quindi sul tempo che questo richiede;
- presenza di una documentazione completa in tutti i suoi aspetti;
- possibilità di organizzare il codice in app indipendenti e ciò permette un riutilizzo del codice stesso;
- presenza di un sistema di testing automatizzato.

Ogni applicazione Django contiene:

- **Model:** sono collezioni di classi Python che vanno a rappresentare le tabelle del database;
- **View:** comprende funzioni Python che gestiscono il flusso dell'applicazione e permettono di definire comportamenti che le pagine adotteranno in funzione all'interazione con l'utente;
- **Url:** permette di mappare la varie url dell'applicazione.

Python: dato che viene utilizzato Django, per la programmazione lato server dovrà essere utilizzato Python, un linguaggio dinamico ad alto livello ed orientato agli oggetti, intuitivo, con un vasto numero di librerie che possono essere incluse nell'applicazione

Javascript: è un linguaggio di programmazione, interpretato ed orientato agli oggetti, ad alto livello e utilizzato per la programmazione lato client.

AJAX: Asynchronous Javascript and XML, è un insieme di tecniche per la realizzazione di applicazioni web interattive. Lo sviluppo di applicazioni HTML e AJAX si basa su uno scambio asincrono di dati in background fra web browser e server, che consente l'aggiornamento dinamico di una pagina web senza ricaricamento esplicito da parte dell'utente.

Bootstrap: framework per applicazioni web. Essa contiene modelli basati su HTML e CSS per lo sviluppo di applicazioni web responsive.

TEST DEL METODO (product/test.py)

Si è deciso di effettuare un test sull'aggiunta di un prodotto. Nella fase iniziale di setup, viene creato un utente ristoratore con tutte le sue informazioni. Successivamente viene creato un oggetto Ingredient con il relativo prezzo e salvato. Viene poi creato il prodotto, con tutte le informazioni aggiunte.

```
class Tests_add_product(TestCase):

    def setUp(self):
        self.user = User()
        self.user.email = 'ciccio@gmail.com'
        self.user.first_name = 'ciccio'
        self.user.last_name = 'marcio'
        self.user.region = 'Emilia Romagna'
        self.user.province = 'MO'
        self.user.cap = '41054'
        self.user.city = 'Modena'
        self.user.via = 'Della pace'
        self.user.house_number = '38'
        self.user.piano = '1'
        self.user.tel = '3332134567'
        self.user.is_restaurateur = True
        self.user.save()
        self.client.force_login(self.user)

    def test_add_product(self):
        self.ingredients = Ingredient()
        self.ingredients.name = 'tartufo'
        self.ingredients.price = 10
        self.ingredients.save()
        response = self.client.post(
            reverse('products:product-create'),
            data={'name': 'Tagliolini al tartufo', 'ingredients': [self.ingredients.pk], 'price': 10}
        )
        self.assertTrue(Food.objects.filter(name="Tagliolini al tartufo").exists())
        self.assertEqual(response.status_code, 302)
```

TEST DELLA VIEW (users/test.py)

```
def test_profile_view(self):
    response = self.client.get(reverse('users:user-profile'))
    self.assertEqual(response.status_code, 200) #se non ci sono errori nella richiesta HTTP
    for p in response.context['prenotazioniDopo']:
        self.assertGreater(p.date, timezone.now())
    for p in response.context['prenotazioniPrima']:
        self.assertLess(p.date, timezone.now())

    self.assertQuerysetEqual(response.context['takeaways'], [repr(self.takeaway)])
    self.assertEqual(response.context['user'], self.user)
```

Dopo una fase di setup iniziale in cui sono stato creato l'utente, una prenotazione di un tavolo, un ingrediente, un cibo e una bevanda, viene controllato che le prenotazioni vengano inserite nel modo corretto rispetto alla data in cui vengono effettuate e che vengano inseriti gli ordini d'asporto nel modo corretto.

SCREENSHOT APPLICAZIONE

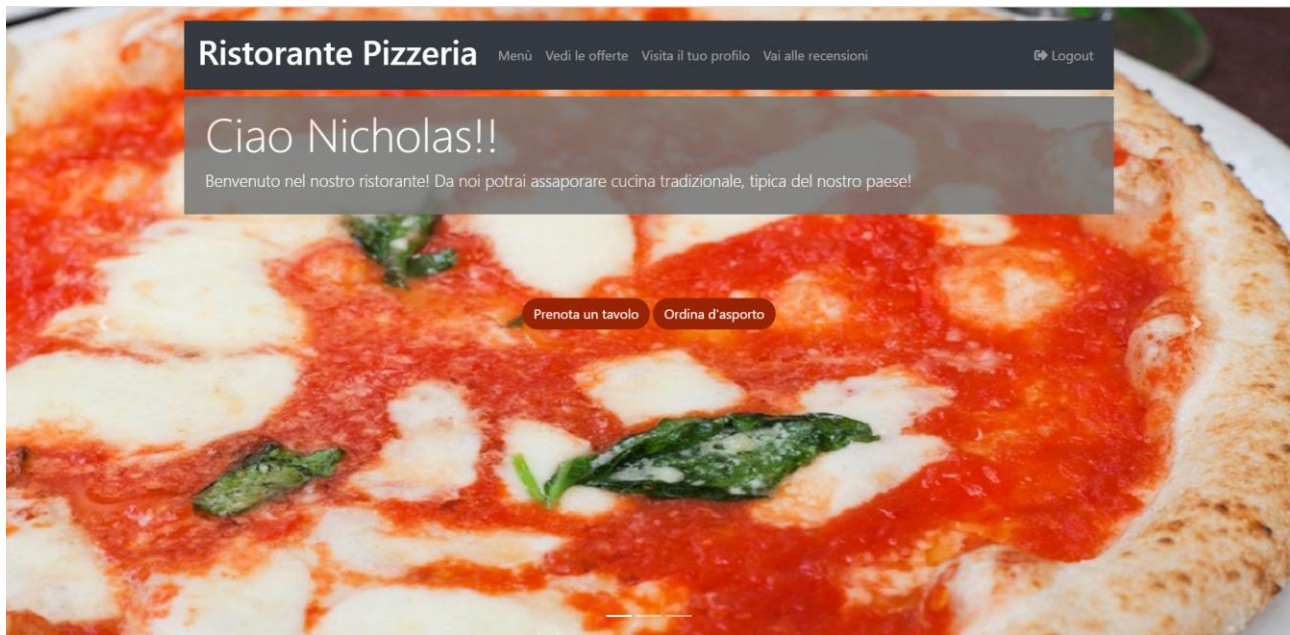





Figura 1: Homepage sito

Cibi

#Cibo	Immagine	Nome	Descrizione	Ingredienti	Prezzo	Azioni
1		Bucatini all'amatriciana	L'Amatriciana (matriciana in romanesco) è un condimento per la pasta tipico della tradizione gastronomica di Amatrice, cittadina in provincia di Rieti, nella regione Lazio.	Pecorino Guanciale	12.00	Modifica Elimina
2		Carbonara	La pasta alla carbonara è un piatto caratteristico del Lazio, e più in particolare di Roma, preparato con ingredienti popolari e dal gusto intenso. I tipi di pasta tradizionalmente più usati sono gli spaghetti e i rigatoni.	Pancetta Uovo	14.00	Modifica Elimina

Bevande

#Bevanda	Immagine	Nome	Descrizione	Litri	Prezzo	Azioni
1		Campanone	Il Campanone, denominazione data dal suo utilizzo come torre civica, chiamata in passato Torre Campanaria o Torre dell'Orologio, consiste in una torre di epoca medievale riattata nel corso dei secoli.	0.75	6.00	Modifica Elimina

[Indietro](#) [Aggiungi un nuovo cibo](#) [Aggiungi una nuova bevanda](#) [Guarda la lista degli ingredienti](#)

Figura 2: Pagina visualizzazione menù da parte del ristoratore

Inserisci il numero di persone della prenotazione:

Inserisci il nome per la prenotazione:

Inserisci il cognome per la prenotazione:

Inserisci la data per la prenotazione:

Inserisci data e l'ora nel formato YYYY-MM-DD hh:mm

Figura 3: Prenotazione di un tavolo

nicholasberna98@gmail.com

Nome:

nicholas

Cognome:

bernardoni

Indirizzo:

puzzole, 45, Piano 3 43356, MO

Regione:

Emilia Romagna

Telefono:

3333333333

Note

d

Indietro

Modifica

Storico prenotazioni

Data:

Oct. 21, 2020, 12:10 p.m.

Numero di posti:

4

Prenotazioni

Data:

Nov. 21, 2020, 12:10 p.m.

Numero di posti:

4

Elimina prenotazione

Data:

Nov. 21, 2020, 12:10 p.m.

Numero di posti:

2

Elimina prenotazione

Data:

Nov. 21, 2020, 12:10 p.m.

Numero di posti:

2

Elimina prenotazione

Storico ordini d'asporto

Data:

Nov. 12, 2020, 10:57 p.m.

Cibi:

Drink:

Totale ordine:

36.50€

Data:

Nov. 13, 2020, 11:01 a.m.

Cibi:

Bucatini all'amatriciana - 12.00€

Drink:

Campanone - 6.00€

Totale ordine:

20.00€

Figura 4: Visualizzazione profilo utente

Giovanni Caselli, Nov. 13, 2020, 11:07 a.m.

☆☆☆

Un peccato per il cibo, location molto carina.

Lascia un commento

Vedi commenti

nicholas bernardoni, Nov. 13, 2020, 11:05 a.m.

☆☆☆☆☆

Ristorante molto carino e staff gentilissimo. Consiglio a tutti!!

Lascia un commento

Vedi commenti

Indietro

Figura 5: Visualizzazione recensioni

Prenotazioni Tavoli

Nome prenotazione:

Tavolo riservato per 4 a nome nicholas bernardoni

Data:

Nov. 21, 2020, 12:10 p.m.

Numero di posti riservati:

4

Elimina prenotazione

Nome prenotazione:

Tavolo riservato per 2 a nome Gigi ddd

Data:

Nov. 21, 2020, 12:10 p.m.

Numero di posti riservati:

2

Elimina prenotazione

Nome prenotazione:

Tavolo riservato per 4 a nome Gigi Calcagno

Data:

Nov. 21, 2020, 12:10 p.m.

Numero di posti riservati:

4

Elimina prenotazione

Nome prenotazione:

Tavolo riservato per 2 a nome nicholas bernardoni

Data:

Nov. 21, 2020, 12:10 p.m.

Numero di posti riservati:

2

Elimina prenotazione

Figura 6: Gestione delle prenotazioni da parte dell'utente