**UNIEATS**

**divplusplus**

**Feature Documentation**

UniEats aims to provide a seamless and personalized dining experience for students, faculty, and staff. It offers real-time access to dining menus, allows users to manage dietary preferences, and tracks meal plan credits. The app also enables users to make dining reservations and provides a feedback system to collect and manage opinions on dining services and meal quality. UniEats makes use of MongoDB for its database management.

This documentation outlines the key features of the app, detailing their functionality and how they contribute to improving the overall campus dining experience.

**Key features:**

**Feedback system**

User Story:

* As a student, I can write a review on the restaurant so that the restaurant can use my feedback.
* As a student, I can view all previous reviews to a restaurant so that I can see if it’s a good restaurant

User Acceptance Test:

**Write a review for a restaurant**

* **Given**: I have visited a restaurant listed on UniEats.
* **When**: I navigate to the review section for the restaurant and submit my feedback.
* **Then**: My review will be successfully posted for the restaurant to view.

**View all previous reviews for a restaurant**

* **Given**: I am a student logged into the application and viewing a specific restaurant's page.
* **When**: I navigate to the review section of the restaurant's page.
* **Then**: I will see a list of all previous reviews for that restaurant.

The Feedback System allows users to provide input on their dining experience. Users can rate meals, give feedback on service, and suggest improvements, helping the dining facilities enhance quality.

The user can provide ratings for their meals and dining experience and submit written comments, suggestions or complaints. Users can also view other user’s comments and ratings for each restaurant on the restaurant’s homepage before they place their order. This will help motivate the restaurants to put customer satisfaction at the top of their priority list.

A screenshot of a computer

Description automatically generated

The above is a screenshot after “Leave review” button is clicked from the homepage of a restaurant. The user can leave a rating along with a comment. Once the user has clicked “Submit Review” a notification pops up indicating to the user that their review has been made.

A screenshot of a menu

Description automatically generated

On the homepage, in the bottom right, the user can see all reviews for the prospective restaurant that they are about to order from. The user can see what other customers think about the place before they place any orders.

UniEats makes use of a POST request to interact with the endpoint '/addReview'. We receive the parameters ‘{restaurant, review, comment}’ and add the review to the review table.

A white rectangular object with a black border

Description automatically generated with medium confidence

Above is a screenshot of reviewmodels in our database. It shows how a review is stored and what parameters are received from the frontend.

**Menu access**

User Story: As a student, when I click on a restaurant, I want to be able to view all of the items that the restaurant sells as well as the price, ingredients and short description of the item.

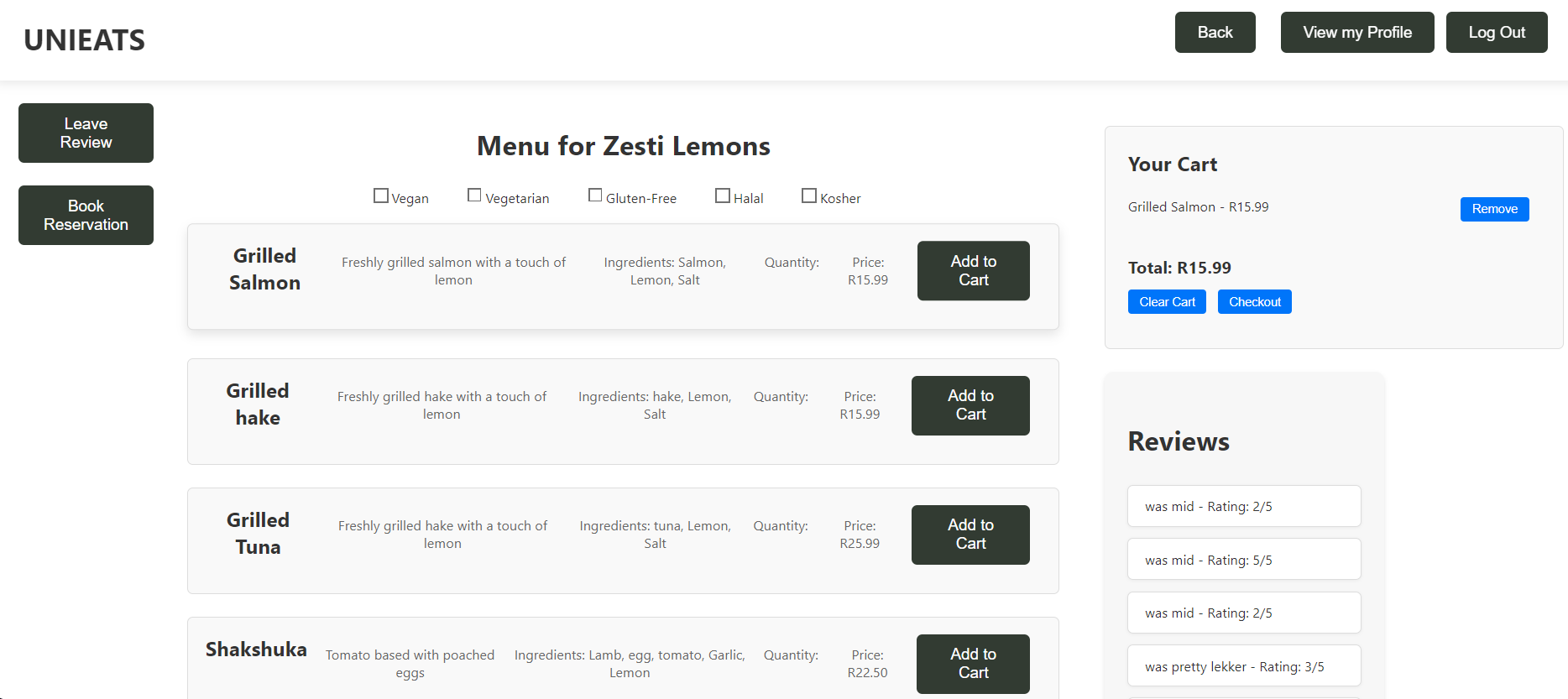
User Acceptance Test:

**View the menu of a selected restaurant**

* **Given**: I am logged into UniEats and viewing the list of available restaurants.
* **When**: I select a specific restaurant from the list.
* **Then**: I will be able to see the full menu for that restaurant.

UniEats menu feature provides users with real-time access to dining menus across all campus dining facilities. It enables users to easily view available meal options, including detailed information about the ingredients for each dish.

The key functionalities are that the menu for the selected restaurant is shown with a short description of the menu item as well as all the ingredients used and the price of the item. If the user wishes to buy an item, then the user can click “Add to Cart” and the selected item will be added to the cart. The menus can also be filtered such that only menu items that match the user’s dietary preferences are shown.



The above picture is a screenshot of the deployed website that shows the menu when Zesti Lemons was selected from the user. The screenshot shows all the key parts of the menu that were needed to be implemented such as the item, description, ingredients and price.

UniEats makes use of a POST request to interact with the endpoint '/viewMenuItems'. The website takes in a restaurant and checks the menuitems table for entries from that restaurant and return all the menu items for that restaurant.

A screenshot of a computer program

Description automatically generated

Above is a screenshot of menuitemmodels that shows how the different menu items are stored in our database.

**Dietary management**

User Story: As a student, I can filter the items on the menu, so that I am able to view the items that fit my dietary requirements.

User Acceptance Test:

**Filter menu items by dietary requirements**

* **Given**: I am viewing the menu of a selected restaurant.
* **When**: I apply a dietary filter (e.g., vegetarian, gluten-free, etc.).
* **Then**: The menu will update to only show items that meet my dietary requirements.

This feature allows users to set, update, and manage their dietary preferences and restrictions. Based on their input, the app automatically filters menus and provides meal suggestions that align with their specific needs.

Users can specify dietary restrictions such as vegan, vegetarian, gluten-free, halal and/or kosher. The system references the menu and shows only the options that meet the user’s preferences. The options that are shown change dynamically when preferences change.

A screenshot of a computer

Description automatically generated

The above picture is the same as the previous screenshot except the user has selected to only see gluten-free menu items and “Shakshuka” is no longer being displayed anymore as it is not a gluten free menu item.

The filtering of menu items based on dietary preferences is done on the frontend. It is done at the time the user is on the menu. The reason for this as opposed to the user defining it at sign up is that he may come to order for a friend who does not have the same preferences as him. It allows for the dietary management to be more versatile. Furthermore, the reason for filtering on the front end is it fastens the process of filtering. Instead of making a fetch every time a filter is set, the front end array is populated with all the menu items and that array is filtered according to the user requirements.

**Meal credits**

User Story:

* As a student, when I go onto my profile, I want to be able to view my credit balance
* As a student, when I enter a voucher code, I want my balance to be increased by the amount according to the voucher.

User Acceptance Test:

**View credit balance on my profile**

* **Given**: I am a student logged into my account on the application.
* **When**: I navigate to my profile page.
* **Then**: I will be able to see my current credit balance.

**Increase balance by entering a voucher code**

* **Given**: I am a student logged into my account.
* **When**: I enter a valid voucher code on my profile or during checkout.
* **Then**: My balance will increase according to the amount specified by the voucher.

The Meal Credits feature allows users to track and manage their meal plan credits. It provides a real-time overview of the user’s current balance and recent transactions. The user also can add more meal credits using vouchers.

The user can view their credit balance, view a history of their transactions including past meals and the respective prices as well as add credits with the use of vouchers. In the “View My Profile” tab, the user can enter in a voucher code and get a certain number of credits added to their balance based on the voucher code entered.

A screenshot of a computer

Description automatically generated

The above screenshot is of the “User Profile” page. It shows the credits balance, the previous order and the price of that respective order as well as where the user would enter a voucher code to increase their credit balance.

UniEats makes use of a POST request to interact with the endpoint '/addCredits'. UniEats takes the voucher and userid as input. We then check if the voucher is a valid voucher, then add the credits to that user by userid in the database.

A screenshot of a computer code

Description automatically generated

Above is a screenshot of how each type of voucher is stored in our database.

**Dining reservations**

User Story: ​

* As a student, I can make a reservation at my selected restaurant so that I can secure my spot at the restaurant.
* As a student, I can view my reservations so that I can prove my reservations.

User Acceptance Test:

**Make a reservation at a restaurant**

* **Given**: I am viewing the menu of my selected restaurant.
* **When**: I fill in the reservation details (date, time, number of guests, special requests) and submit the form.
* **Then**: I will receive confirmation that my reservation has been made.

**View my reservations**

* **Given**: I am a student logged into my account.
* **When**: I navigate to the reservations section of my profile.
* **Then**: I will see a list of my past and upcoming reservations.

This feature allows users to reserve tables at campus dining facilities. It simplifies the process of making and managing reservations, ensuring users have a dining spot during peak times.

The user can make a reservation for a table at a certain time or date for a restaurant as well as specify the number of guests and any special requests. The user can cancel bookings as needed and can view all their bookings on the “User Profile” page.

A screenshot of a computer

Description automatically generated

The above picture is a screenshot after clicking the “Book Reservation” button on the Zesti Lemons home page. The user can specify the date, time, number of guests and any special requests that the user might have. Once the user has clicked “Submit Reservation” a message pops up letting the user know that their reservation has been made successfully.

A screenshot of a user profile

Description automatically generated

This picture is a screenshot of the “User Profile” page that shows all the user’s reservations and allows the user to view the details of their reservations as well as delete any reservations that they no longer need.

UniEats utilizes a POST request to interact with the endpoints ‘/addReservation’, '/deleteReservation' and '/viewReservations' to add, delete and view a reservation respectively.

**Secure login**

User Story: As a student, I want to have secure access to my account so that no one else can log on to my profile and spend my credits.

User Acceptance Test:

**Secure access to account**

* **Given**: I am a student with a registered account on the UniEats platform.
* **When**: I log in using my correct credentials (username and password),
* **Then**: I will securely access my profile, and no unauthorized users will be able to log in and spend my credits without the correct login information.

The Secure Login feature ensures that users can safely access UniEats using a robust authentication system powered by Auth0. This feature provides a secure way to authenticate users, protecting their data, including meal credits, dietary preferences, and feedback submissions. Auth0 facilitates the implementation of modern authentication standards such as OAuth 2.0and JWT, allowing users to log in using Google accounts or traditional email and password credentials.

A screenshot of a login form

Description automatically generated

The above screenshot is what is shown when the user clicks “Login”. The user can either login traditionally with an email address and a password or using their Google account.

A computer screen shot of a computer code

Description automatically generated

The above screenshot is how a User is stored in our database. When logging in we make use of the userid to check, firstly if the user exists and secondly to get the user’s information.

**Add to cart**

User Story:

* As a student, when I add to cart and checkout, I want to be notified when my order is ready so that I can collect my order
* As a student, when I add to cart and checkout, I want to be able to view my previous orders so that I can see what I ordered.

User Acceptance Test:

**Receive notification when order is ready**

* **Given**: I am a student who has added items to my cart and completed the checkout process.
* **When**: My order is ready for collection,
* **Then**: I will receive a notification informing me that my order is ready to be picked up.

**View previous orders after checkout**

* **Given**: I am a student who has completed previous orders on the platform.
* **When**: I navigate to the order history section of my profile,
* **Then**: I will be able to view a list of all my previous orders, including the details of what I ordered.

The Add to Cart feature allows users to select items from the dining menu and place an order if they have sufficient meal credits. Upon placing an order, the credits are automatically deducted from the user’s balance. The placed order is then available for users to view on their profile page, where they can track order history and see details of past transactions.

Users can browse the daily menu and select items to add to their cart. Before confirming an order, the system checks if the user has enough meal credits to cover the cost. Upon placing an order, the required number of credits is deducted from the user’s meal credit balance. Once the transaction is successful, the order is processed and recorded in the system. Users can view their past orders in the "Profile" section, including details such as date, time, items ordered, and credits spent.

To mock the restaurants cooking the order and letting the user know that the user’s order is ready, we send an email to the user after 30 minutes letting the user know that their order is ready and to come collect it.

A screenshot of a computer

Description automatically generated

The above screenshot is of an email that the user would receive 30 minutes after placing an order at Zesti Lemons.

UniEats utilizes a POST request to interact with the endpoints ‘/addOrder’ and the endpoint '/completeOrder'. The former is takes in these parameters { userID, total, items, orderID, email,restaurant,date, time } and adds it to the ordermodels table. The latter takes in the parameter {ordered} and updates the status of order in the order table to completed once the user clicks completed.



Above is a screenshot of the ordermodels table in our database and it shows all the parameters sent when a new order is made by the user.

**All User Stories**

1. As a student, when I login to my profile, I will be able to view a list of all the restaurants available.
2. As a student, when I select my desired restaurant, I can see the menu.
3. As a student, I can filter the items on the menu, so that I am able to view the items that fit my dietary requirements.
4. As a student, I can make a reservation at my selected restaurant so that I can secure my spot at the restaurant.
5. As a student, I can write a review on the restaurant so that the restaurant can use my feedback.
6. As a student, I can view all previous reviews for a restaurant so that I can see if it's a good restaurant.
7. As a student, when I go onto my profile, I want to be able to view my credit balance.
8. As a student, when I enter a voucher code, I want my balance to be increased by the amount according to the voucher.
9. As a student, I can view my reservations so that I can confirm my reservations.
10. As a student, when I add items to the cart and checkout, I want to be notified when my order is ready so that I can collect it.
11. As a student, when I add items to the cart and checkout, I want to be able to view my previous orders so that I can see what I ordered.
12. As a student, I want to have secure access to my account so that no one else can log on to my profile and spend my credits.

**All User Acceptance Tests**

1. **View a list of all restaurants upon login**

* **Given**: I am a student with a valid account.
* **When**: I log into my profile on the UniEats platform.
* **Then**: I will see a list of all the restaurants available.

1. **View the menu of a selected restaurant**

* **Given**: I am logged into UniEats and viewing the list of available restaurants.
* **When**: I select a specific restaurant from the list.
* **Then**: I will be able to see the full menu for that restaurant.

1. **Filter menu items by dietary requirements**

* **Given**: I am viewing the menu of a selected restaurant.
* **When**: I apply a dietary filter (e.g., vegetarian, gluten-free, etc.).
* **Then**: The menu will update to only show items that meet my dietary requirements.

1. **Make a reservation at a restaurant**

* **Given**: I am viewing the menu of my selected restaurant.
* **When**: I fill in the reservation details (date, time, number of guests, special requests) and submit the form.
* **Then**: I will receive confirmation that my reservation has been made.

1. **Write a review for a restaurant**

* **Given**: I have visited a restaurant listed on UniEats.
* **When**: I navigate to the review section for the restaurant and submit my feedback.
* **Then**: My review will be successfully posted for the restaurant to view.

1. **View all previous reviews for a restaurant**

* **Given**: I am a student logged into the application and viewing a specific restaurant's page.
* **When**: I navigate to the review section of the restaurant's page.
* **Then**: I will see a list of all previous reviews for that restaurant.

1. **View credit balance on my profile**

* **Given**: I am a student logged into my account on the application.
* **When**: I navigate to my profile page.
* **Then**: I will be able to see my current credit balance.

1. **Increase balance by entering a voucher code**

* **Given**: I am a student logged into my account.
* **When**: I enter a valid voucher code on my profile or during checkout.
* **Then**: My balance will increase according to the amount specified by the voucher.

1. **View my reservations**

* **Given**: I am a student logged into my account.
* **When**: I navigate to the reservations section of my profile.
* **Then**: I will see a list of my past and upcoming reservations.

1. **Receive a notification when my order is ready**

* **Given**: I have added items to my cart, completed the checkout process, and my order is being prepared.
* **When**: My order is ready for collection.
* **Then**: I will receive a notification informing me that my order is ready.

1. **View previous orders**

* **Given**: I am a student logged into my account and have placed orders in the past.
* **When**: I navigate to the order history section of my profile.
* **Then**: I will see a list of my previous orders with details of what I ordered.

1. **Secure log in**

* **Given**: I am a student logged into my account and have placed orders in the past.
* **When**: I navigate to the order history section of my profile.
* **Then**: I will see a list of my previous orders with details of what I ordered.