

# **Employee Meal Tracking Assistant**

## **Documentation**

### **Description of the Functionalities**

#### **Classes:**

Class StudentRepository – Contains all the Data Layer (Database) functionalities.

Class StudentController – Contains the API Services and calls them.

Class StudentDetails, MealRecords, DefaultersDetails, ClockLogs – All of these classes provide constructors and, getter and setter methods.

#### **Functions:**

getStudentDetailsFromDB – This function helps the administrator to get the student details (S\_ID, First\_Nmae, Last\_Name, Employee\_Store, Employer\_name, Schedule) from the database.

addMealRecord – This function helps the student order his meal from the Meal\_store. It takes S\_ID, Meal\_Order, Meal\_Store as input and stores the students meal in the database.

generateDefaultersList – This procedure generates a list of all students who have taken an invalid meal. The reasons for invalid meal is either ‘the student has taken a meal without attending his shift’ or ‘Student is taken a meal even though he has worked for more than 20 hours per week’

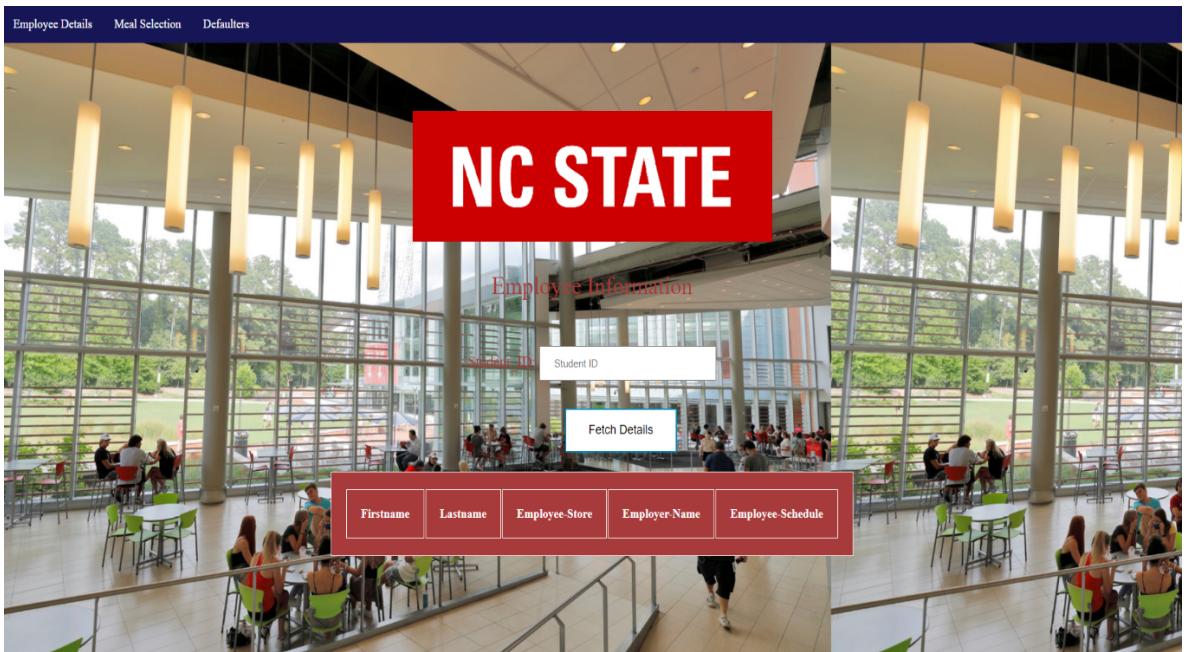
fetchTotalHoursForID – This function gets the total number of hours worked by the student per week.

checkShiftVisited – This function checks if the student has gone for the shift on that particular day.

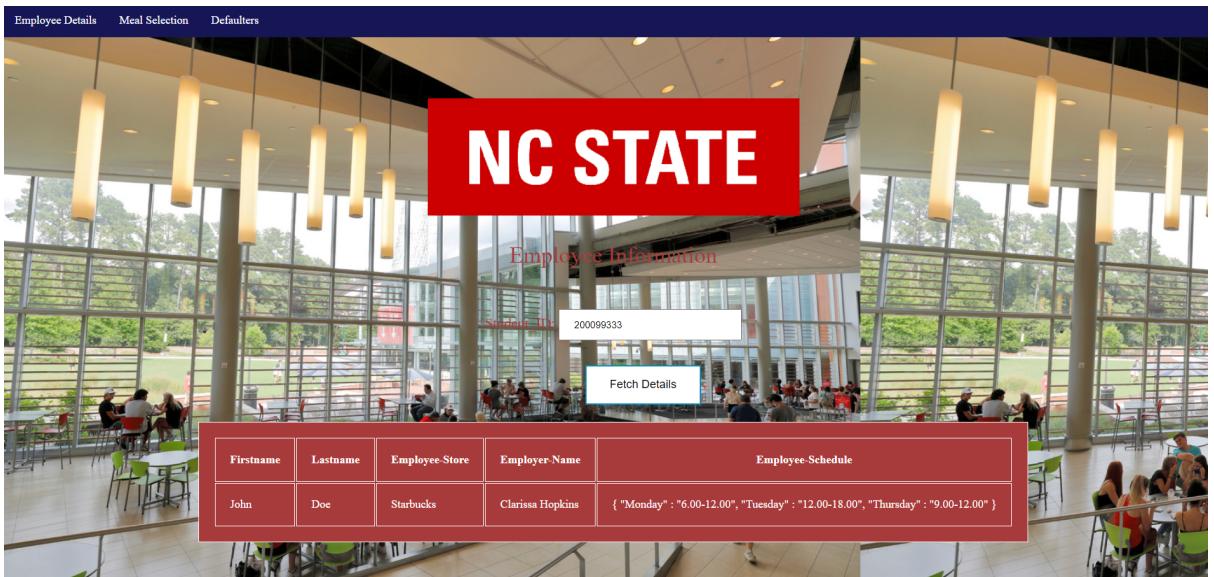
getDefaulterIds – This method is used by the generateDefaultersList method to get all the S\_IDs of all the defaulters, so that the administrator can generate the report.

## Common uses cases and working example of the system:

- The employee details webpage:  
Asks for the Employee Student ID.



- As and when we enter the student ID of the employee, and click on “Fetch details” button, the API call will fetch the student employee details and display it in a tabular format as shown below. This helps the store owner to check the basic details of the student employee.



- The restaurant menu webpage

This webpage is designed to add the student employees meal details and store it in the database.

The screenshot shows a "Food Menu" page with a header for "Student ID" and "Store name". Below the header is a table titled "Food Menu" with columns: Product Name, Quantity, Price, Total Price, and Operation (with an "Add to cart" button). The table lists eight items: French Fries, Spinach Veggie Wrap, Garden Hummus, Black Bean Burger, Veggie Bowl, Jason's Tots, Soft Drink, and Black Bean Burger (repeated). The total cost is displayed as \$13. An "Add" button is located at the bottom left.

Product Name	Quantity	Price	Total Price	Operation
French Fries	<input type="text" value="1"/>	\$3.00	\$3.00	Add to cart
Spinach Veggie Wrap	<input type="text" value="1"/>	\$5.00	\$5.00	Add to cart
Garden Hummus	<input type="text" value="1"/>	\$5.00	\$5.00	Add to cart
Black Bean Burger	<input type="text" value="1"/>	\$7.00	\$7.00	Add to cart
Veggie Bowl	<input type="text" value="1"/>	\$8.00	\$8.00	Add to cart
Jason's Tots	<input type="text" value="1"/>	\$5.00	\$5.00	Add to cart
Soft Drink	<input type="text" value="1"/>	\$2.50	\$2.50	Add to cart
Black Bean Burger	<input type="text" value="1"/>	\$7.00	\$7.00	Add to cart

Total Cost: 13

Add

- After the student employee comes to receive his/her work day meal, we enter the Employee Student ID, the store name from where he/she is ordering their meal and the food items that need to be added to the cart. The system then calculates the total meal cost for this employee and the “ADD” button will store the required data.

The screenshot shows the same "Food Menu" page with updated quantities. The "Garden Hummus" row now has a quantity of 2, resulting in a total price of \$10.00. The other items remain the same. The total cost is now \$13. An "Add" button is located at the bottom left.

Product Name	Quantity	Price	Total Price	Operation
French Fries	<input type="text" value="1"/>	\$3.00	\$3.00	Add to cart
Spinach Veggie Wrap	<input type="text" value="1"/>	\$5.00	\$5.00	Add to cart
Garden Hummus	<input type="text" value="2"/>	\$5.00	\$10.00	Add to cart
Black Bean Burger	<input type="text" value="1"/>	\$7.00	\$7.00	Add to cart
Veggie Bowl	<input type="text" value="1"/>	\$8.00	\$8.00	Add to cart
Jason's Tots	<input type="text" value="1"/>	\$5.00	\$5.00	Add to cart
Soft Drink	<input type="text" value="1"/>	\$2.50	\$2.50	Add to cart
Black Bean Burger	<input type="text" value="1"/>	\$7.00	\$7.00	Add to cart

Total Cost: 13

Add

- The defaulter's list webpage

When we click on the “Defaulters” button, present at the top navigation bar, the API generates a list of the current day employee meal defaulters. This list highlights the students details who have invalidly taken a meal, along with the reason for the violation.

The screenshot shows a web application interface for meal defaulters. At the top, there is a navigation bar with three items: "Employee Details", "Meal Selection", and "Defaulters". Below the navigation bar is a large red banner with the white text "NC STATE". Underneath the banner, the text "Defaulter Information" is displayed in a smaller red font. A table follows, listing five defaulter entries. The table has columns for "Sid", "Name", "Employed Store", "Employer Name", and "DefaulterReason". The data is as follows:

Sid	Name	Employed Store	Employer Name	DefaulterReason
200099555	Emily Cooper	Jasons Deli	Tiffany James	Weekly limit of hours crossed
200088444	Elijah Tapia	Starbucks	Clarissa Hopkins	No Shift Visited
200055444	Sincere Carillo	Tuffys Diner	Angela Wang	No Shift Visited
200055555	Hillary Stankovic	Port City Java	Monica	Weekly limit of hours crossed

The background of the page shows a photograph of a modern cafeteria or dining hall with large windows, people sitting at tables, and a red "NC STATE" sign on the wall.