JOHN FRY'S FOOD TRUCK EXTRAVAGANZA - SYSTEM MANUAL

John Fry's Food Truck Extravaganza (abbreviated JFFTE) consists of a client application for end users (client), an editor app to add and modify data (editor), and a Google Firebase database (database). Note: Instructions for *using* the client are fairly simple and self-explanatory and can be found in the user manual.

Both the client and editor are each made of three files: an HTML file, a JavaScript file, and a CSS file. The client also makes use of an image file, logo.png.

JFFTE is designed to hold information about a series of food trucks on Drexel's campus, as well as a series of food items on that truck's menu. A template for the format of each truck and each food item is as follows; the trucks are stored in JSON format:

```
var truck = {
        "id": <a unique identifier for the truck>,
        "name": <the name of the truck>,
        "location": <truck's general location on Drexel's campus>,
        "address": <the closest approximate street address>,
        "cuisine": <description of what kind of food the truck serves>,
        "lat": <latitude: a decimal number from -90.0 to 90.0, inclusive>,
        "lng": <longitude: a decimal number from -180.0 to 180.0, inclusive>,
        "time": {  <integers, when trucks open and close, ranges inclusive>
                "open": {"hour":<0 - 23>, "minute":<0 – 59> },
                "closed": {"hour":<0 – 23>, "minute":<0 – 59> }
        },
        "menu": [0] <an array of foodItems>
};
```

```
var foodItem = {
        "name": <name of the food item>,
        "price": <positive decimal – price of the food item>
};
```

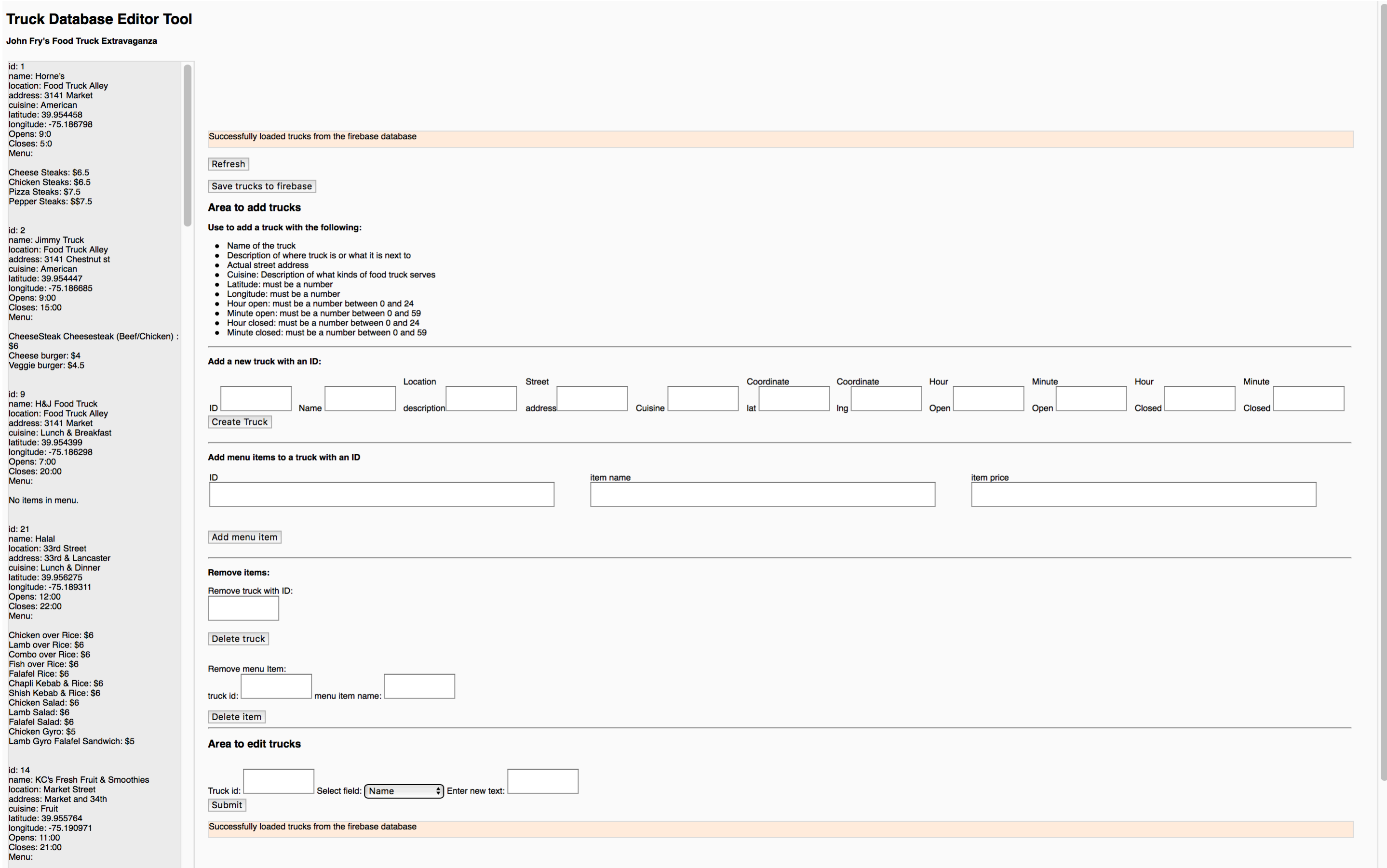
The editor is essentially a user interface for an administrator (admin) to create a series of truck objects. When created, the trucks are added to a local array. The admin then adds foodItems to the trucks, one by one. The admin can then push the trucks to the database.

Warning*: *Important information regarding the editor:

- At startup, the editor loads all trucks from the database into its local truck array. In the current build of the editor, when the admin clicks save, the editor pushes all the trucks from its local truck array to the database, replacing the entire database. *Thus, the editor should not be used by multiple people simultaneously.*
- Empty arrays in the editor are represented by [0], an array with a single item with value 0. A function exists to check if an array is empty by checking if the item at index 0 is 0 (a function isArrayEmpty exists to check this). Thus, all functions that add or remove items to arrays within the editor, whether the array is the main local truck array, or an array of food items specific to a truck, must check the following:
    - If adding an item to an array, use isArrayEmpty to see if the array is empty. If it is, replace the first item in the array (the 0) with the item you are adding. Otherwise, just push the item you are adding to the array.
    - If subtracting an item from an array, the length of the array must be checked; if the length is 1, then the only item in the array should simply be replaced with a 0. Otherwise, the array should be spliced at the index of the item you wish to remove, and only a single item should be removed.

- This was done to avoid the annoying and inconsistent behavior regarding empty arrays in JavaScript; the hoops we are jumping through here with [0] are, in the author's opinion, less tedious that the hoops we'd be jumping through otherwise.
- Since the client application does not and should never modify data under any circumstance, [0] is not used within the client and no functions account for it. Currently, it is up to the editor to ensure that all trucks are created and uploaded in the proper format.

When opening the editor, the admin is presented with the following interface:



Truck Database Editor Tool

John Fry's Food Truck Extravaganza

```
id: 1
name: Horne's
location: Food Truck Alley
address: 3141 Market
cuisine: American
latitude: 39.954458
longitude: -75.186798
Opens: 9:0
Closes: 5:0
Menu:

Cheese Steaks: $6.5
Chicken Steaks: $6.5
Pizza Steaks: $7.5
Pepper Steaks: $$7.5


id: 2
name: Jimmy Truck
location: Food Truck Alley
address: 3141 Chestnut st
cuisine: American
latitude: 39.954447
longitude: -75.186685
Opens: 9:00
Closes: 15:00
Menu:

CheeseSteak Cheesesteak (Beef/Chicken) :
$6
Cheese burger: $4
Veggie burger: $4.5


id: 9
name: H&J Food Truck
location: Food Truck Alley
address: 3141 Market
cuisine: Lunch & Breakfast
latitude: 39.954399
longitude: -75.186298
Opens: 7:00
Closes: 20:00
Menu:

No items in menu.


id: 21
name: Halal
location: 33rd Street
address: 33rd & Lancaster
cuisine: Lunch & Dinner
latitude: 39.956275
longitude: -75.189311
Opens: 12:00
Closes: 22:00
Menu:

Chicken over Rice: $6
Lamb over Rice: $6
Combo over Rice: $6
Fish over Rice: $6
Falafel Rice: $6
Chapli Kebab & Rice: $6
Shish Kebab & Rice: $6
Chicken Salad: $6
Lamb Salad: $6
Falafel Salad: $6
Chicken Gyro: $5
Lamb Gyro Falafel Sandwich: $5


id: 14
name: KC's Fresh Fruit & Smoothies
location: Market Street
address: Market and 34th
cuisine: Fruit
latitude: 39.955764
longitude: -75.190971
Opens: 11:00
Closes: 21:00
Menu:
```

Successfully loaded trucks from the firebase database

Area to add trucks

Use to add a truck with the following:

- Name of the truck
- Description of where truck is or what it is next to
- Actual street address
- Cuisine: Description of what kinds of food truck serves
- Latitude: must be a number
- Longitude: must be a number
- Hour open: must be a number between 0 and 24
- Minute open: must be a number between 0 and 59
- Hour closed: must be a number between 0 and 24
- Minute closed: must be a number between 0 and 59

Add a new truck with an ID:

Add menu items to a truck with an ID

Remove items:

Area to edit trucks

Truck id: Select field: Name Enter new text:

Successfully loaded trucks from the firebase database

A list of all trucks is printed out on the side of the page. When a change is made to the database, the list is reprinted. The orange bars are status bars; they exist to indicate to the admin that a successful or unsuccessful action has taken place, such as a food truck being added, a menu item being deleted, etc. Here is what an admin can do with the editor:

- Refresh: Reprints the list of trucks in the gray display area on the left side of the screen.
- Save trucks to database: this button will update the contents of the database by replacing them with the contents of the local array. This is not an ideal solution, obviously, and had we further developed the application it would have been changed for a safer system.

- Add or remove trucks: A set of text fields where the admin can type in the information required to create a new truck. An admin can enter a unique ID for the truck, as well as a name, location description, etc. When the admin clicks the "create truck" button:
    - The editor checks whether the ID is in use,
    - The editor checks and parses each field,
    - If the ID is unused and all fields are acceptable, it adds a new truck to the local array.
        - Trucks are created with an empty menu.
- Add a menu item: An admin types in the ID of the truck he wants to add a menu item to, as well as the item name and the item price. When the "add menu item" button is clicked:
    - The editor makes sure a truck exists with the ID the admin typed in
    - If the truck exists, the editor checks each item in its menu to make sure none of the existing (if any) items has a name that matches the item name the admin wants to add. In other words, every item on the menu must have a unique name.
    - The editor checks to ensure the item price is a decimal number.
    - If all fields are acceptable, a menu item is added to the specified truck in the local array.
- Remove a truck: the admin types the ID of the truck to be removed. If a truck with the ID exists, it is deleted from the local array.
- Remove a menu item: the admin types the ID of the truck they want to remove an item from, and the name of that item. If the truck exists and an item within the truck has a matching name, it is removed.
- Edit a truck: this allows the admin to type the ID of the truck they want to edit, select from a drop-down menu a field from that truck to edit, and type in what they want to replace that field with. If the truck exists, and the text the admin typed is acceptable for the field selected from the drop-down menu, the truck's field is modified.
    - As of 2018.06.02, there is currently a bug that the author of this document has not fixed that prevents the text from being accepted. I'm still not sure where the bug is, but trust me, this part would be working right now if it wasn't for that bug.
    - Editing menu items is still not implemented, but at this point it makes more sense to delete the item you want to edit and re-add it.