**Project:**  Smart refrigerator

**Description:**

The Smart Refrigerator project introduces an innovative IoT scanning device designed to revolutionize how users manage their kitchen inventory. Placed near or on the refrigerator, the IoT device seamlessly integrates with a user-friendly mobile app. This system aims to not only streamline inventory management but also enhance the overall kitchen experience by providing features such as automated shopping list creation, recipe suggestions, and timely alerts for expiring products.

**Use cases:**

1. **IoT Scanning and Real-time Inventory** 
   * A user wants to effortlessly scan items using the IoT device when placing them in/out of the refrigerator and being able to manage the food inventory.
   * The app will have to facilitate seamless item scanning through the IoT device and update the real-time inventory list accordingly, allowing the user to view and modify the list through the app.
2. **Automated Shopping List: ( not implemented yet )**
   * A user wants to rely on the system to analyze the scanned inventory and generate a shopping list.
   * The app will have to analyze scanned inventory data and generate a shopping list.
3. **Recipe Suggestions: ( not implemented yet )**
   * A user wants to benefit from the system leveraging scanned inventory data to suggest personalized recipes based on available ingredients.
   * The app will have to utilize scanned inventory information to offer personalized recipe suggestions and display them to the user in a convenient way.
4. **Expiration Alerts and Food Waste Reduction: ( not implemented yet )**
   * A user wants to receive timely notifications for upcoming expiration of scanned items.
   * The app will have to send notifications that alert users about expiring items, along with suggestions to reduce food waste through recipe ideas.

**Interfaces:**

**Frontend to Backend**

**1.User Authentication:**

* POST /register : Register a new user.

Example of a Json from the Frontend:

{

    "email" : "dani@gmailcom" ,

    "password" : 12345,

    "first\_name" : "dani",

    "last\_name" : "cohen"

}

* POST /user\_login : authenticate users within the system.

Example of a Json from the Frontend:

{

"email" : "dani@gmailcom" ,

"password" : 12345

}

**2.** **Refrigerator related:**

* GET /linked\_ refrigerators/{user\_id} : get a list of all the id’s of the refrigerators that the user linked to.

Example of a returned Json:

{

"refrigerators": [

{

"nickname": "my refrigerator",

"refrigerator\_id": 1

},

{

"nickname": "mom’s refrigerator",

"refrigerator\_id": 131939297

}

}

* GET /refrigerator\_content/{refrigerator\_id} : get a list of all the products that are in the refrigerator.

Example of a returned Json:

{

"products": [

{

"product\_addedTime": "2024-05-21 15:14:35.647408",

"product\_image": <The image encoded>,

"product\_name": "milk 3%",

"product\_quantity": 8

},

{

"product\_addedTime": "2024-05-14 12:56:18.327383",

"product\_image": <The image encoded>,

"product\_name": "small water bottle",

"product\_quantity": 10

}

],

"refrigerator\_id": "1"

}

* GET /linked\_ refrigerators/{user\_id} : get a list of all the id’s of the refrigerators that the user linked to.

Example of a returned Json:

{

    "refrigerators": [

        {

            "nickname": "my refrigerator",

            "refrigerator\_id": 1

        },

        {

            "nickname": "mom’s refrigerator",

            "refrigerator\_id": 131939297

        }

}

* GET /number\_linked\_refrigerators/{user\_id} : get the number of the the linked refrigerators that the user linked to.

Example of a returned Json:

{

    "number\_linked\_refrigerators": 6

}

* POST /update\_refrigerator\_name/{user\_id}: change the nickname of the relevant refrigerator for the given user.

Example of a Json from the Frontend:

{

“refrigerator\_id”: 1234,

“new\_name”: “new refrigerator”

}

* GET /generate\_shopping\_list/{refrigerator\_id} : generate a shopping list accordingly to the missing products in the refrigerator.

\*This endpoint is not implemented yet.

* GET /generate\_ recipes/{refrigerator\_id} : generate a list of recipes accordingly to the available products in the refrigerator.

\*This endpoint is not implemented yet.

* GET /alerts/{user\_id} : returns the existing alerts related to the user's refrigerators.

\* This endpoint is not implemented yet.

**Embedded to Backend**

* POST /scan : identify scanned product by barcode and add/remove to the relevant refrigerator.

Example of a Json that can be send from the embedded device:

{

“barcode”: “7290110115906”,

“refrigerator\_id” : 123456789,

“mode”: “add”

}

* GET /request\_ refrigerator\_id : first-time request to receive a refrigerator ID within the system.(returns the id itself)
* POST /link : links between a user and a refrigerator.

Example of a Json that can be send from the embedded device:

{

“user\_id”:123,

“refrigerator\_id” : 123456789

}

**Database schemes**

**1.**  The Product table stores information about each product.

**2.** The User table stores information about each user.

**3.** The Refrigerator table stores id of each refrigerator.

**4.** TheRefrigerator\_content table stores the products that available in each refrigerator, linked to a refrigerator via the refrigerator\_id foreign key and linked to a product via barcode foreign key.

**5.** The link table stores the links and nicknames between each user to each refrigerator, linked to a refrigerator via the refrigerator\_id foreign key and linked to a user via user\_id foreign key.

**Product table:**

CREATE\_TABLE product {

barcode char(9) PRIMARY KEY,

product\_name char(255) NOT NULL,

image char(255),

};

**User table:**

CREATE\_TABLE user {

user\_id SERIAL PRIMARY KEY,

email char(255) UNIQUE NOT NULL,

password char(255) NOT NULL,

first\_name char(255) NOT NULL,

last\_name char(255)

};

**Refrigerator table:**

CREATE\_TABLE refrigerator {

refrigerator\_id INTEGER PRIMARY KEY

};

**Refrigerator\_content table:**

CREATE\_TABLE refrigerator\_content {

refrigerator\_id INTEGER REFERENCES refrigerator(refrigerator\_id),

barcode char(9) REFERENCES product(barcode) ,

product\_quantity INTEGER NOT NULL,

oldest\_added\_date DATE NOT NULL,

PRIMARY KEY( refrigerator\_id, barcode)

};

**Link table:**

CREATE\_TABLE link {

user\_id char(255) REFERENCES user(user\_id),

refrigerator\_id INTEGER REFERENCES refrigerator(refrigerator\_id),

nickname char(255),

PRIMARY KEY(email, refrigerator\_id)

};