Project Report

Xinxue Guo(xg2407), Jasmine Yuan(cy2571)
Github Repository:https://github.com/jasmineyuan169/SHEMS

I.Introduction

With the rapid development of smart home technology, the integration of energy management system into everyday life has become increasingly important. This document presents a comprehensive design and implementation plan for the Smart Home Energy Management System(SHEMS), an innovative solution aimed at facillating energy consumption monitoring and management for homeowners. The system is designed to cater for diverse needs of customers, empowering them with the ability to manage their energy usage more efficiently.

The SHEMS is a user-friendly, robust platform which allows enrolled users to connect to their service locations (properties receiving eletrical service) and smart devices as well as provides real-time information on energy consumption. Each part of the system, from customer account management to the intricate details of device events and energy usage tracking, is carefully considered to provide a seamless and friendly user experience.

The ability to handle complex data interactions and provide insightful energy consumption analysis lies in the central of the SHEMS. The system incorporates various entities, including Customers, Service Locations, Devices, Device Events, Energy Usage, and Energy Prices. Each serves as a unique role in the system. The interplay between different entities is carefully designed to capture detailed information, ranging from basic customer and device information to the energy usage pattern of a specific device type.

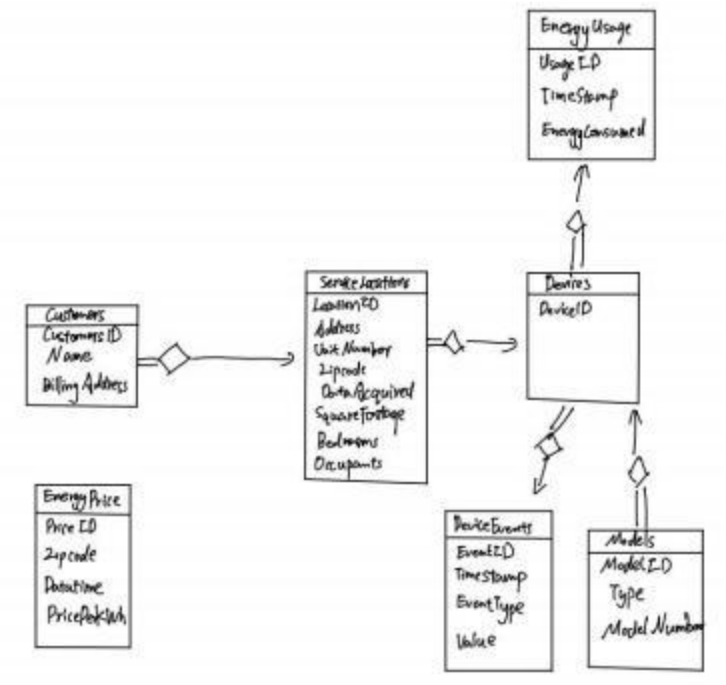
The document delves into the intracacies of the SCHEMS design. The detailed ER diagram illustrates the relationship between different entities, carefully constructed tables and constraints which lay the foundation of data integrity and consistency.

Queries and Procedures are designed to facillitate data definition, maniputation and retrieval, ensuring that the system is intuitive to the user.

Each component of the SCHEMS is rigoriously tested to ensure it meets the high standard of performance and the whole system is reliable and effective. The document carefully selects sample data to make sure that every possible scenarios are included.

In summary, the document serves as a blueprint for the SCHEMS system, including every aspect of its design and functionallity. It is a testament to our effort to deliver an innovative solution that facillitates energy management for customers, paving the way for a more sustainable and efficient future of home energy consumption.

II.ER Diagram



III.Database Schema and Constraint

1.**Customers**(CustomerID, Name, BillingAddress)
Primary Key: CustomerID

2.**ServiceLocations**(LocationID, CustomerID, Address, UnitNumber, Zipcode, DateAcquired, SquareFootage, Bedrooms, Occupants)
Primary Key: LocationID
Foreign Key: CustomerID reference Customers

3.**Models**(ModelID, Type , ModelNumber)
Primary Key: MoeldelID

4.**Devices**(DeviceID, LocationID, ModelID)
Primary Key: DeviceID
Foreign Key: LocationID reference ServiceLocations
ModelID reference Models

5.**DeviceEvents**(EventID, DeviceID , Timestamp, EventType, Value)
Primary Key: EventID
Foreign Key: DeviceID reference Devices

6.**EnergyUsage**(UsageID, DeviceID , Timestamp, EnergyConsumed)
Primary Key: UsageID
Foreign Key: DeviceID reference Devices

7.**EnergyPrices**(PriceID, ZipCode, DateTime, PricePerKWh)
Primary Key: PriceID

IV.Test Data

1.Customers

CustomerID	Name	BillingAddress
1	John Doe	billing address 1
2	Mary Lee	billing address 2
3	Anne Wong	billing address 3

2.ServiceLocations

LocationID	CustomerID	Address	UnitNumber	Zipcode	DateAcquired	SquareFootage	Bedrooms	Occupants
1	1	location address 1	3001	12345	2021-05-18	924	2	3
2	2	location address 2	2533	12354	2020-10-21	876	2	2
3	1	location address 3	1242	52738	2019-08-07	1052	3	3
4	3	location address 4	1823	63724	2022-01-04	792	1	1
5	3	location address 5	1510	24253	2018-04-23	839	2	3

3.Models

ModelID	Type	ModelNumber
1	AC System	FTXS18HVJU
2	AC System	CS-E12RKUAW
3	light	9290012573A
4	light	26792
5	light	P-BDG-PKG1W
6	refrigerator	RF28R7201SR
7	refrigerator	LFXS26596S
8	refrigerator	WRS325SDHZ

4.Devices

DeviceID	LocationID	ModelID
1	1	1
2	1	3
3	1	6
4	2	1
5	2	2
6	2	4
7	2	5
8	2	7
9	3	2
10	3	3
11	3	4
12	3	6
13	4	1
14	4	4
15	4	5
16	4	8
17	5	2
18	5	4
19	5	3
21	5	7

5.DeviceEvents

EventID	DeviceID	Timestamp	EventType	Value
1	1	2022-11-30 14:30:00	temp lowered	78
2	1	2022-12-04 12:21:30	energy use	32
3	1	2022-08-24 11:24:08	energy use	24
4	1	2021-06-17 13:52:08	temp lowered	23
5	4	2022-08-16 21:13:42	energy use	33
6	4	2022-09-12 07:32:11	energy use	53
7	4	2021-08-09 16:24:33	switched on	NULL
8	6	2022-08-03 12:13:55	energy use	12
9	6	2022-08-10 11:28:07	energy use	23.14
10	6	2022-09-13 09:22:57	energy use	53.24
11	10	2021-12-27 00:00:00	energy use	11
12	10	2022-04-06 17:25:33	switched on	NULL
13	12	2022-09-04 12:23:44	energy use	44
14	12	2022-08-01 14:22:47	energy use	27
15	12	2022-08-13 23:11:28	energy use	31
16	17	2021-06-09 16:23:49	switched on	NULL
17	17	2022-08-01 15:24:33	energy use	27
18	17	2022-08-09 14:14:14	energy use	25
19	19	2022-08-12 09:37:52	energy use	36
20	19	2022-09-15 10:36:24	energy use	22
21	19	2022-09-22 21:33:56	energy use	30
22	21	2022-09-06 07:08:22	energy use	26
23	21	2022-08-11 17:23:34	door opened	NULL
24	1	2023-11-30 22:50:34	energy use	18
25	6	2023-11-30 22:50:56	energy use	9
26	21	2023-11-30 22:51:24	energy use	29
27	10	2023-11-30 23:01:32	energy use	10
28	21	2023-11-27 08:00:00	door opened	NULL
29	21	2023-11-27 08:20:00	door closed	NULL
30	21	2023-11-27 09:00:00	door opened	NULL
31	21	2023-11-27 09:45:00	door closed	NULL
32	21	2023-11-27 10:00:00	door opened	NULL

6.EnergyUsage

UsageID	DeviceID	Timestamp	EnergyConsumed
1	1	2022-12-04 12:21:30	32.00
2	1	2022-08-24 11:24:08	24.00
3	4	2022-08-16 21:13:42	33.00
4	4	2022-09-12 07:32:11	53.00
5	6	2022-08-03 12:13:55	12.00
6	6	2022-08-10 11:28:07	23.14
7	6	2022-09-13 09:22:57	53.24
8	10	2021-12-27 00:00:00	11.00
9	12	2022-09-04 12:23:44	44.00
10	12	2022-08-01 14:22:47	27.00
11	12	2022-08-13 23:11:28	31.00
12	17	2022-08-01 15:24:33	27.00
13	17	2022-08-09 14:14:14	25.00
14	19	2022-08-12 09:37:52	36.00
15	19	2022-09-15 10:36:24	22.00
16	19	2022-09-22 21:33:56	30.00
17	21	2022-09-06 07:08:22	26.00
18	1	2023-11-30 22:50:34	18.00
19	6	2023-11-30 22:50:56	9.00
20	21	2023-11-30 22:51:24	29.00
21	10	2023-11-30 23:01:32	10.00

7. EnergyPrices

PriceID	ZipCode	DateTime	PricePerKWh
1	12345	2022-12-04 12:00:00	1.50
2	12345	2022-08-24 11:00:00	1.50
3	12354	2022-08-16 21:00:00	1.20
4	12354	2022-09-12 07:00:00	1.20
5	12354	2022-08-03 12:00:00	1.40
6	12354	2022-08-10 11:00:00	1.40
7	12354	2022-09-13 09:00:00	1.40
8	52738	2021-12-27 00:00:00	0.70
9	52738	2022-09-04 12:00:00	1.60
10	52738	2022-08-01 14:00:00	1.60
11	52738	2022-08-13 23:00:00	0.80
12	24253	2022-08-01 15:00:00	1.50
13	24253	2022-08-09 14:00:00	1.50
14	24253	2022-08-12 09:00:00	1.50
15	24253	2022-09-15 10:00:00	1.50
16	24253	2022-09-22 21:00:00	1.30
17	24253	2022-09-06 07:00:00	1.30
18	12345	2023-11-30 22:00:00	1.20
19	12354	2023-11-30 22:00:00	1.00
20	24253	2023-11-30 22:00:00	1.00
21	52738	2023-11-30 23:00:00	0.70

V.Create Command

```
CREATE TABLE Customers (
CustomerID INT AUTO_INCREMENT PRIMARY KEY,
Name VARCHAR(255),
BillingAddress VARCHAR(255)
);

CREATE TABLE ServiceLocations (
LocationID INT AUTO_INCREMENT PRIMARY KEY,
CustomerID INT,
Address VARCHAR(255),
UnitNumber INT,
Zipcode VARCHAR(10),
DateAcquired DATE,
SquareFootage INT,
Bedrooms INT,
Occupants INT,
FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);

CREATE TABLE Models(
ModelID INT AUTO_INCREMENT PRIMARY KEY,
Type VARCHAR(50),
ModelNumber VARCHAR(50)
);

CREATE TABLE Devices (
DeviceID INT AUTO_INCREMENT PRIMARY KEY,
LocationID INT,
ModelID INT,
FOREIGN KEY (LocationID) REFERENCES ServiceLocations(LocationID),
```

```
FOREIGN KEY (ModelID) REFERENCES Models (ModelID)
);

CREATE TABLE DeviceEvents (
EventID INT AUTO_INCREMENT PRIMARY KEY,
DeviceID INT,
Timestamp DATETIME,
EventType VARCHAR(50),
Value VARCHAR(255),
FOREIGN KEY (DeviceID) REFERENCES Devices(DeviceID)
);

CREATE TABLE EnergyUsage (
UsageID INT AUTO_INCREMENT PRIMARY KEY,
DeviceID INT,
Timestamp DATETIME,
EnergyConsumed DECIMAL(10, 2),
FOREIGN KEY (DeviceID) REFERENCES Devices(DeviceID)
);

CREATE TABLE EnergyPrices (
PriceID INT AUTO_INCREMENT PRIMARY KEY,
ZipCode VARCHAR(10),
DateTime DATETIME,
PricePerKWh DECIMAL(10, 2)
);

DELIMITER //
CREATE TRIGGER InsertEnergyUsage
AFTER INSERT ON DeviceEvents
FOR EACH ROW
BEGIN
   IF NEW.EventType = 'energy use' THEN
      INSERT INTO EnergyUsage (DeviceID, Timestamp, EnergyConsumed)
      VALUES (NEW.DeviceID, NEW.Timestamp, NEW.Value);
   END IF;
END;
//
DELIMITER ;
```

Explanation:

Whenever the smart device transmits energy consumption related data to the database, specifically labeled as ‘energy use’, the information is first stored in ‘DeviceEvents’ because it represents a typical event. Subsequently, a trigger which will be activated when detecting the label is implemented to make sure the pertinent details will be automatically transferred to ‘EnergyUsage’ table. This streamlines the process and ensures data consistency of the system.

VI. SHEMS Web-Based User Interface Design

1. Login Interface

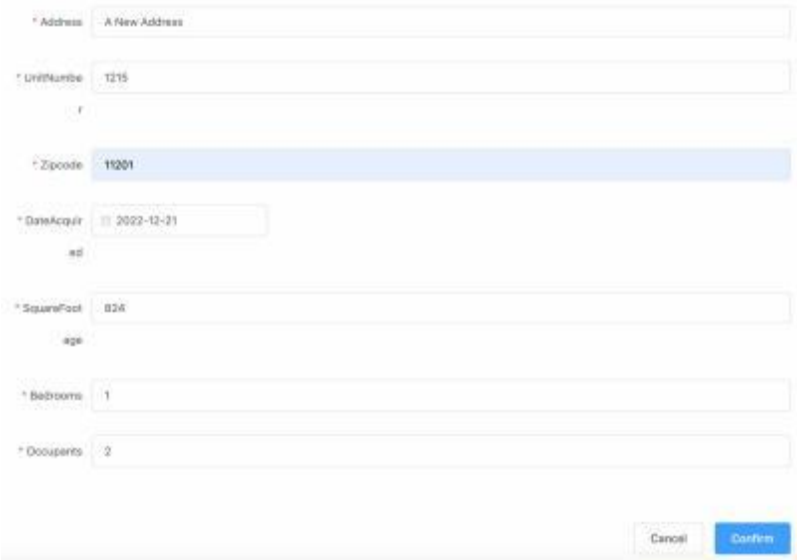
Scenario 1. When entering the wrong username/password.

Scenario 2.When login without password.

2.Register Interface

3. Check all service locations registered by the customer

Scenario 1. Add a new record



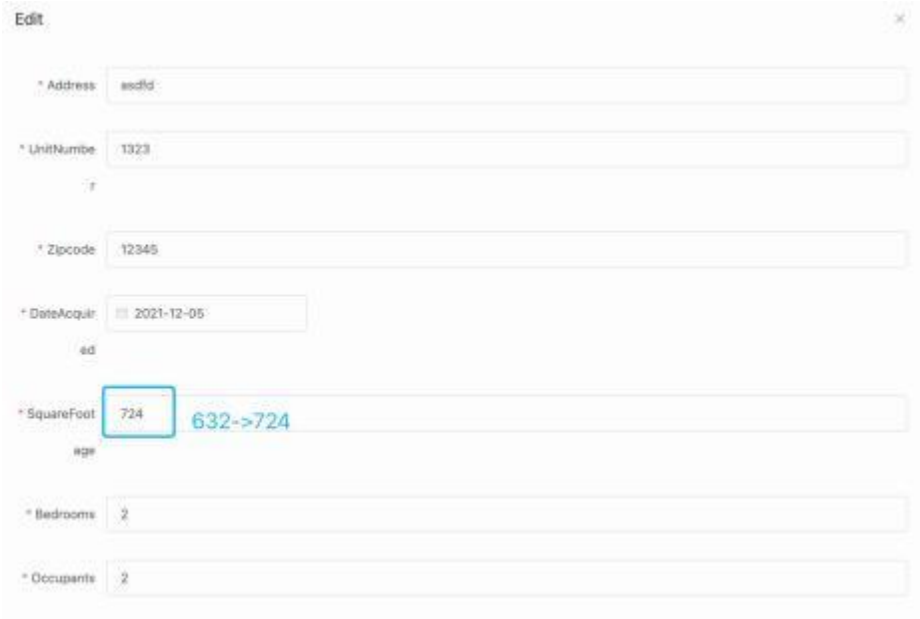
After Insertion


```
mysql> select * from servicelocations where CustomerID=1;
+------------+------------+-----------------+------------+---------+--------------+---------------+----------+-----------+
| LocationID | CustomerID | Address         | UnitNumber | Zipcode | DateAcquired | SquareFootage | Bedrooms | Occupants |
+------------+------------+-----------------+------------+---------+--------------+---------------+----------+-----------+
|          3 |          1 | location address 3 |       1213 | 52738   | 2019-08-07   |          1052 |        3 |         3 |
|          6 |          1 | location address 6 |       2610 | 07302   | 2022-11-02   |           732 |        1 |         2 |
|         11 |          1 | asdfd           |       1323 | 12345   | 2021-12-06   |           632 |        2 |         2 |
|         12 |          1 | A New Address   |       1215 | 11201   | 2022-12-21   |           824 |        1 |         2 |
+------------+------------+-----------------+------------+---------+--------------+---------------+----------+-----------+
4 rows in set (0.00 sec)
```

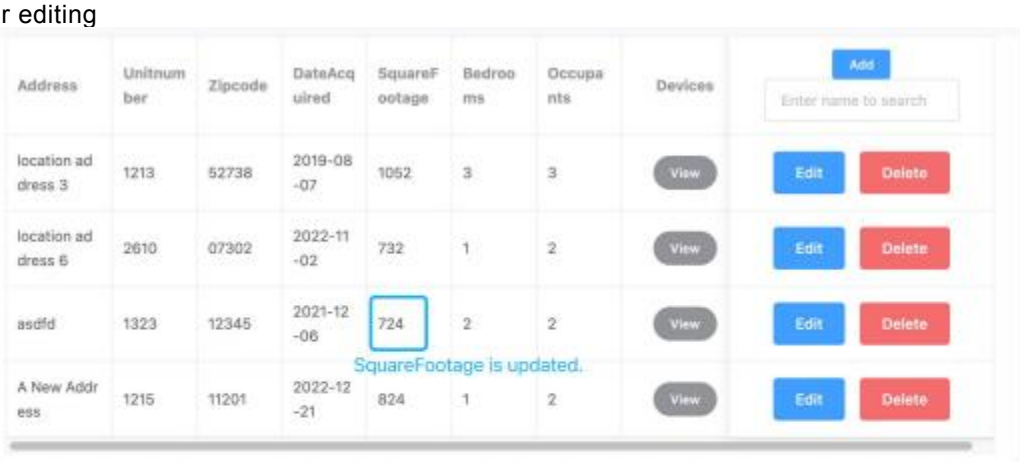
New tuple is also inserted into the backend database.

Scenario 2. Edit a record

Edit the tuple whose Address is ‘asdfd’ from the previous form.



After editing




```
mysql> select * from servicelocations where CustomerID=1;
+------------+------------+---------------------+------------+---------+--------------+---------------+----------+-----------+
| LocationID | CustomerID | Address             | UnitNumber | Zipcode | DateAcquired | SquareFootage | Bedrooms | Occupants |
+------------+------------+---------------------+------------+---------+--------------+---------------+----------+-----------+
|          3 |          1 | location address 3  |       1213 | 52738   | 2019-08-07   |          1052 |        3 |         3 |
|          6 |          1 | location address 6  |       2610 | 07302   | 2022-11-02   |           732 |        1 |         2 |
|         11 |          1 | asdfd               |       1323 | 12345   | 2021-12-06   |           724 |        2 |         2 |
|         12 |          1 | A New Address       |       1215 | 11201   | 2022-12-21   |           824 |        1 |         2 |
+------------+------------+---------------------+------------+---------+--------------+---------------+----------+-----------+
4 rows in set (0.00 sec)
```

The backend database is also updated

Scenario 3. Delete a record

Delete the record whose address is ‘asdfd’ from the previous form.

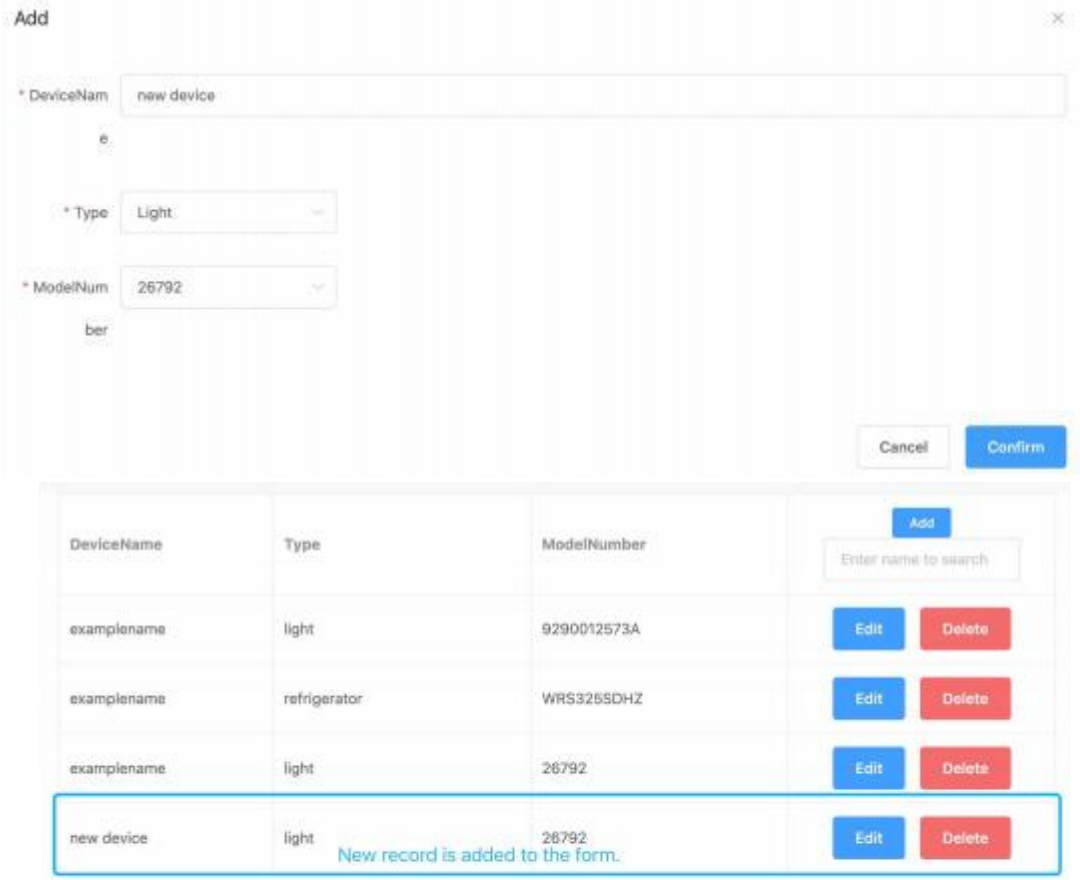
```
mysql> select * from servicelocations where CustomerID=1;
+------------+------------+---------------------+------------+---------+--------------+---------------+----------+-----------+
| LocationID | CustomerID | Address             | UnitNumber | Zipcode | DateAcquired | SquareFootage | Bedrooms | Occupants |
+------------+------------+---------------------+------------+---------+--------------+---------------+----------+-----------+
|          3 |          1 | location address 3  |       1213 | 52738   | 2019-08-07   |          1052 |        3 |         3 |
|          6 |          1 | location address 6  |       2610 | 07302   | 2022-11-02   |           732 |        1 |         2 |
|         12 |          1 | A New Address       |       1215 | 11201   | 2022-12-21   |           824 |        1 |         2 |
+------------+------------+---------------------+------------+---------+--------------+---------------+----------+-----------+
3 rows in set (0.01 sec)
```

It is also deleted from the database.

4. Check all devices of a specific service location.

Let's check the devices of the service location whose LocationID is 3.

Scenario 1. Add a record

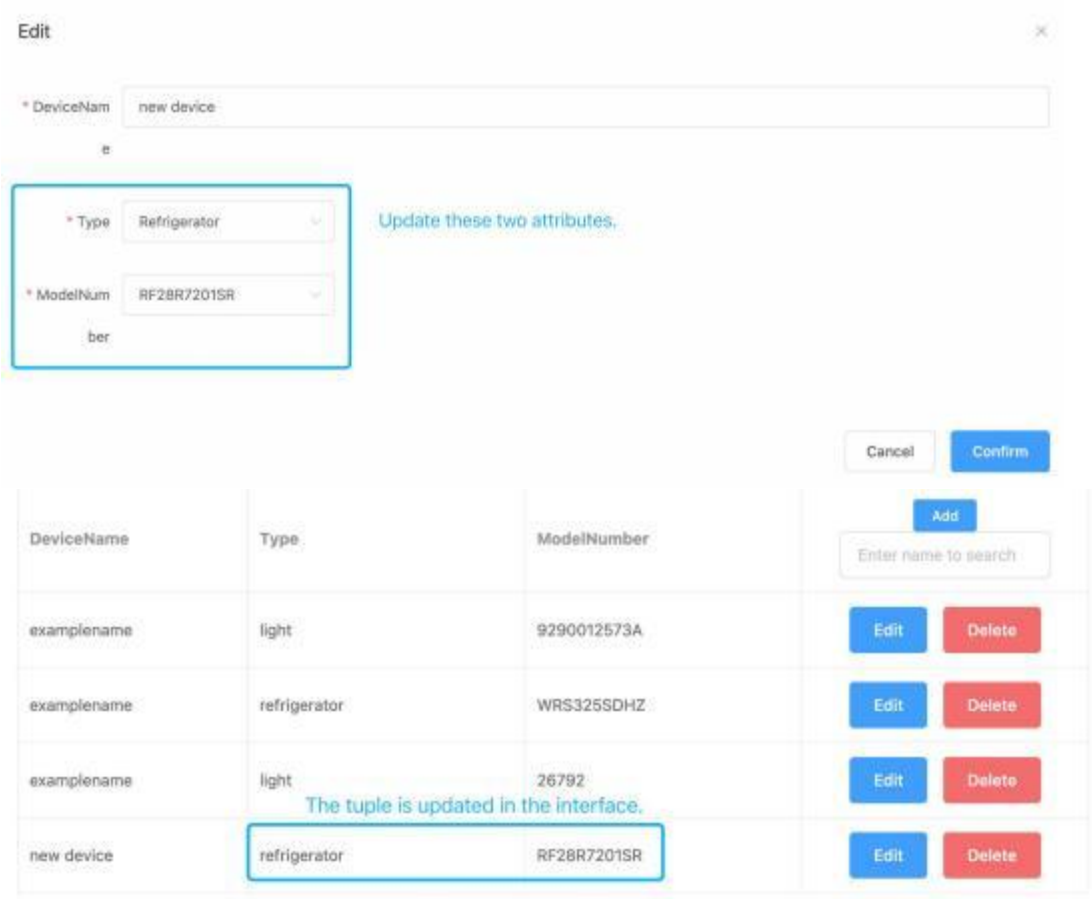



```
mysql> select * from Devices Natural Join Models where LocationID=3;
+---------+----------+------------+-------------+--------------+-------------+
| ModelID | DeviceID | LocationID | DeviceName  | Type         | ModelNumber |
+---------+----------+------------+-------------+--------------+-------------+
|       3 |        9 |          3 | examplename | light        | 9290012573A |
|       8 |       10 |          3 | examplename | refrigerator | WRS325SDHZ  |
|       4 |       11 |          3 | examplename | light        | 26792       |
|       4 |       33 |          3 | new device  | light        | 26792       |
+---------+----------+------------+-------------+--------------+-------------+
```

New device is also inserted into the database

Scenario 2. Edit a record

Edit the tuple whose DeviceName is 'new device'




```
mysql> select * from Devices Natural Join Models where LocationID=3;
+---------+----------+------------+-------------+--------------+-------------+
| ModelID | DeviceID | LocationID | DeviceName  | Type         | ModelNumber |
+---------+----------+------------+-------------+--------------+-------------+
|       3 |        9 |          3 | examplename | light        | 9290012573A |
|       8 |       10 |          3 | examplename | refrigerator | WRS325SDHZ  |
|       4 |       11 |          3 | examplename | light        | 26792       |
|       6 |       33 |          3 | new device  | refrigerator | RF28R7201SR |
+---------+----------+------------+-------------+--------------+-------------+
4 rows in set (0.00 sec)
```

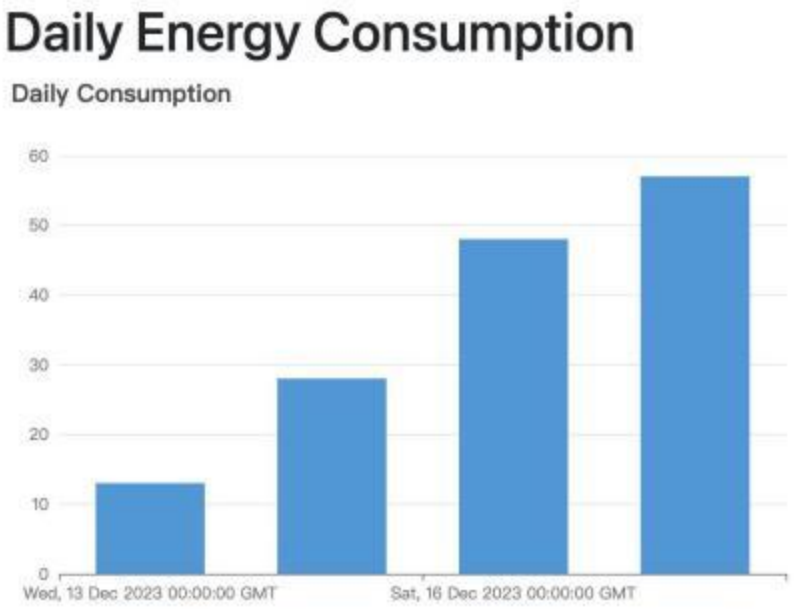
Also updated from the database.

Scenario 4.Delete a record
Delete the record whose ModelNumber is ‘26792’

```
mysql> select * from Devices Natural Join Models where LocationID=3;
+---------+----------+------------+-------------+---------------+-------------+
| ModelID | DeviceID | LocationID | DeviceName  | Type          | ModelNumber |
+---------+----------+------------+-------------+---------------+-------------+
|       3 |        9 |          3 | examplename | light         | 9290012573A |
|       8 |       10 |          3 | examplename | refrigerator  | WRS325SDHZ  |
|       6 |       33 |          3 | new device  | refrigerator  | RF28R7201SR |
+---------+----------+------------+-------------+---------------+-------------+
3 rows in set (0.00 sec)
```

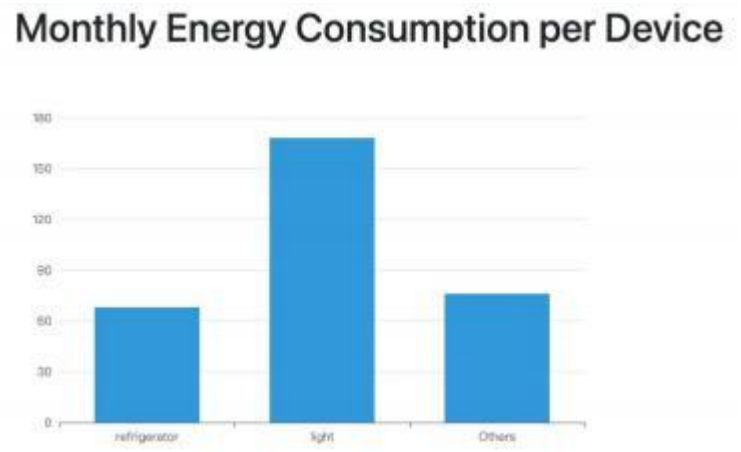
5. Energy Usage Pattern Charts

Chart 1 Daily Energy Consumption



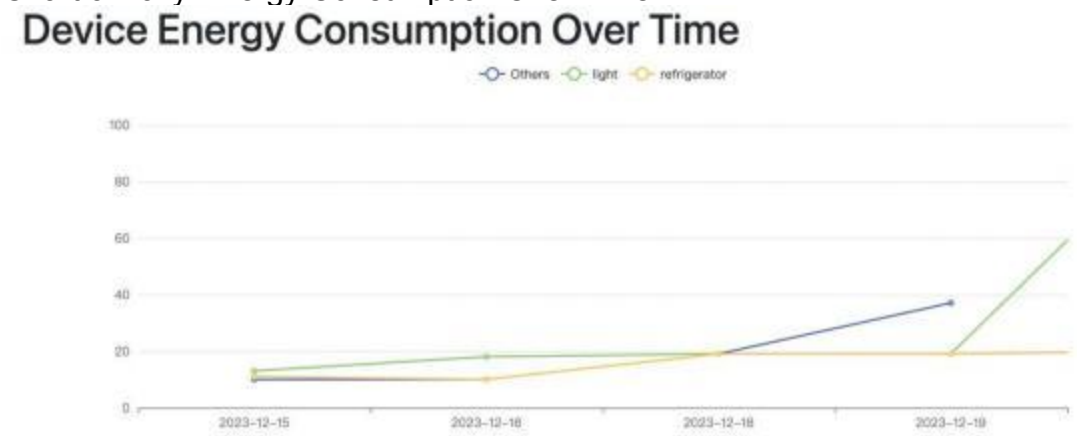
The application supports GET and POST requests on '/daily'. For GET, it displays a date selection form. For POST, it processes the date range, validates it, and fetches energy consumption data from a MySQL database. The SQL code correlates energy usage with devices and locations per customer ID, using JOINs across 'EnergyUsage', 'Devices', and 'ServiceLocations' tables to calculate daily energy consumption, filtered by customer ID. The output is a set of tuples for each day's total consumption, visualized in a bar chart using ECharts. The chart, showing energy usage from December 13-16, 2023, reveals daily consumption trends, aiding users in identifying unusual usage patterns and optimizing energy efficiency.

Chart 2 Monthly Energy Consumption per Device



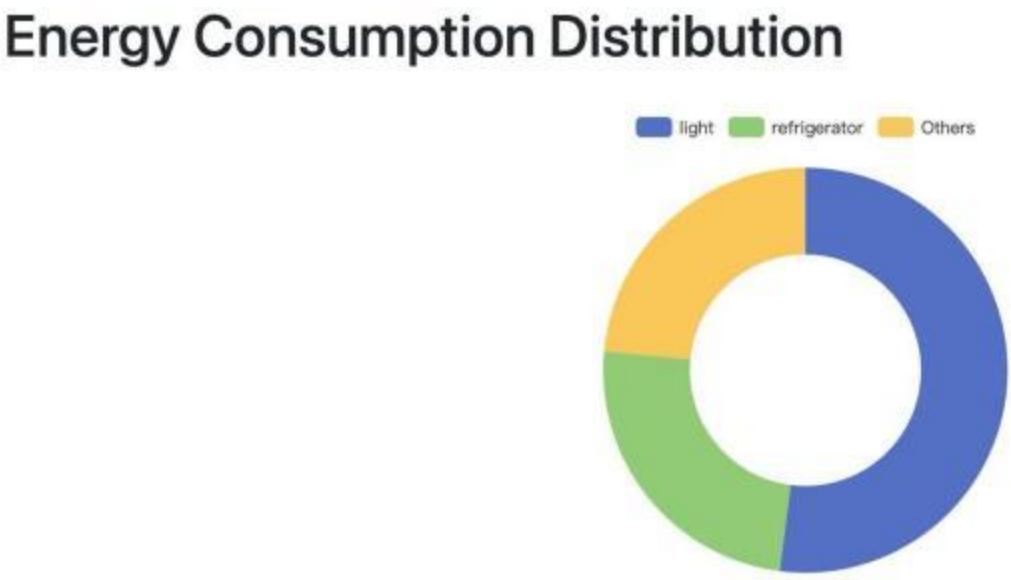
The /monthly_consumption endpoint in the application queries the 'SHEMS' MySQL database to merge data from EnergyUsage, Devices, Models, and ServiceLocations tables. This forms a detailed view of a customer's monthly energy consumption by device, identified by customer_id. The SQL query retrieves data for the last complete month, grouping it by device type from the Models table and summing up energy use per type.
The resulting data is displayed in a bar chart on the monthly_consumption.html template using the ECharts library. The chart categorizes energy consumption for devices like 'refrigerator', 'light', and 'Others', highlighting 'light' as the top consumer. This visual aids users in understanding their energy usage, potentially influencing them to optimize device usage or invest in efficient appliances for cost and environmental benefits.

Chart 3 Daily Energy Consumption Over Time



The /device_consumption_over_time Flask route retrieves energy consumption data from a MySQL database for a specific customer's devices. It executes a SQL query to gather device name, date, and daily energy consumption from EnergyUsage, Devices, Models, and ServiceLocations, filtering by ServiceLocations.CustomerID and organizing data by device and date.
The code structures this data into chart_data for visualization with the ECharts library in the device_consumption_over_time.html template, resulting in a line chart titled "Device Energy Consumption Over Time". The chart, covering December 15th to 19th, 2023, shows energy trends for 'Others', 'light', and 'refrigerator' devices, with significant energy use spikes in 'Others' and moderate increase in 'refrigerator' on the last day.
This visual helps users track energy patterns, identify unusual usage, and guides informed decisions on energy management, cost reduction, and environmentally conscious choices.

Chart 4 Energy Consumption Distribution



The /energy_consumption_distribution Flask route visualizes a customer's energy consumption by device type, using data from the 'SHEMS' MySQL database. A SQL query joins EnergyUsage, Devices, Models, and ServiceLocations tables to calculate total energy use per device, identified by customer_id. The data, grouped by device type, is processed into labels (device types) and values (energy consumed).

This is visualized as a donut chart in the energy_consumption_distribution.html template, using ECharts. The chart, titled "Energy Consumption Distribution," color-codes segments for 'light', 'refrigerator', and 'Others', showing the proportion of energy each type uses. It helps users quickly identify major energy-consuming devices, aiding in energy management and sustainability decisions, like switching to energy-efficient lighting.