

[Table of Contents:](#)

Task Decomposition with Abstract Code:

[Main Menu](#)

[Household Information](#)

[Add Appliance](#)

[View Appliance listing](#)

[Add power generation](#)

[View Power generation](#)

[listing](#)

[Wrapping up](#)

[Reports-Top 25 popular](#)

[manufacturers](#)

[Reports-](#)

[Manufacturer/model](#)

[search](#)

[Reports-Heating/cooling](#)

[method details](#)

[Reports-Water heater](#)

[statistics by state](#)

[Reports-Off-the-grid](#)

[household dashboard](#)

[Reports-Household](#)

[averages by radius](#)

Main Menu

Abstract Code

- There are two links in the main menu: “enter household information” and “View Reports”.
- When the “Enter household information” link is clicked, the user will be provided with the interface to enter their household information into the system.
- When “View reports” link is clicked, a list with links to generate each report will be displayed.

Household Information

Abstract Code

- When the User enters their email ('\$email'), validate that the value is a valid email. Then validate that the email does not already exist in the database:

```
SELECT COUNT(email) FROM Household WHERE email = '$email';
```

If the number of rows returned is zero then the address is valid, else it is invalid and an error message should be displayed.

- When the User enters a postal code('\$postalCode'), validate that it exists in the database:

```
SELECT COUNT(PostalCode) FROM PostalCode WHERE PostalCode = '$postalCode';
```

If the number of rows returned is 1, then the postal code is valid, else it is invalid an error message should be displayed.

- Users choose home type from dropdown list.
- User enters square footage.
 - Validate that the value is a positive whole number.
- User enters heating/cooling thermostat setting:
 - If the user enters either thermostat setting, validate that the value is a positive whole number.
 - If either thermostat setting is empty, validate that the corresponding checkbox is checked. If not, an error should be displayed to the user.
- User select Public Utilities or leave it as unchecked.
- User clicks the Next button to go to next form **Add Appliance**.
 - If there are any form/validation errors, an error message is displayed and the user must correct the errors before moving to the Add Appliance form.

Phase 2 Abstract Code w/SQL | CS 6400 - Summer 2023 | Team 022

- If there are no form errors, the data is inserted into the database:

```
INSERT INTO Household (Email, SquareFootage, HouseholdType, HeatingSetting,
CoolingSetting, PostalCode)
VALUES ($email, $squareFootage, $householdType, $heatingSetting, $coolingSetting,
$postalCode)
```

Get the *householdID* for the newly inserted Household:

```
$householdID = SELECT householdID FROM Household where email = $email
```

Insert the household utilities if any are checked. The query is run once for each checked utility.

```
INSERT INTO PublicUtility (HouseholdID, UtilityName)
VALUES ($householdID, $utilityName)
```

If there are no database/code errors then the Add Appliance form is shown.

Add Appliance

Abstract Code

- User selects Appliance types.
- The manufacturers are retrieved from the database:

```
SELECT ManufacturerName FROM Manufacturer;
```

- If user select Air Handler:
 - User select Manufacturer from dropdown list
 - User enter BTUs number
 - User optionally enters a Model name. (The input field is restricted to the max allowed chars for the database column)
 - User select method which allow multiple values (Air conditioner, Heater, Heat pump)
 - If user select Air conditioner, user need to enter Energy efficiency ratio (EER), decimal number (to the tenth decimal point).
 - If user select Heater, user need to select Energy source: Electric, gas, or thermosolar.
 - If user select Heat pump, Seasonal energy efficiency rating (SEER), decimal number (to the tenth decimal point) and Heating seasonal performance factor (HSPF), decimal number (to the tenth decimal point) are required to be filled in.
- user select energy source from dropdown list.
- User enter RPMs

Phase 2 Abstract Code w/SQL | CS 6400 - Summer 2023 | Team 022

- When the user clicks *Add*, the BTU, EER, and RPM fields are validated. Appropriate error messages are shown if the validation fails. Else, the data is inserted:

Get the next OrderNumber for this household

```
$nextOrderNumber = SELECT COALESCE(Max(orderNumber) + 1, 1) FROM  
Appliance where HouseholdId = $householdID
```

Insert the Appliance and get it's ID:

```
$applianceID = INSERT INTO Appliance  
(HouseholdID, OrderNumber, ManufacturerName, ModelName, BtuRating)  
VALUES ($householdID, $nextOrderNumber, $manufacturer, $modelName,  
$btuRating);  
SELECT LAST_INSERT_ID();
```

Insert the AirHandler and get it's ID:

```
$airHandlerID = INSERT INTO AirHandler (Rpm, ApplianceID) VALUES ($rpm,  
$applianceID);  
SELECT LAST_INSERT_ID();
```

For each selected heating/cooling method, insert the relevant data:

```
INSERT INTO Heater (EnergySource, AirHandlerID)  
VALUES ($energySource, $airHandlerID)  
  
INSERT INTO HeatPump (SEER,HSPF,AirHandlerID)  
VALUES ($seer, $hspf, $airHandlerID)  
  
INSERT INTO AirConditioner (EER,AirHandlerID)  
VALUES ($seer, $airHandlerID)
```

- If user select water heater
 - user select Manufacturer from dropdown list
 - user enter BTUs number
 - user enter Model name
 - user enter Tank size in gallons, decimal value (to the tenth decimal point).
 - user enter the Temperature.
 - user select energy source from dropdown list.
- User click Add button to inset a record in DB:
 - Validate that the tank size and btu rating are supplied and are non-negative.
 - Insert the data into the DB:

Insert the Appliance and get it's ID:

```
$applianceID = INSERT INTO Appliance  
(HouseholdID, OrderNumber, ManufacturerName, ModelName, BtuRating)  
VALUES ($householdID, $nextOrderNumber, $manufacturer, $modelName,  
$btuRating);  
SELECT LAST_INSERT_ID();
```

Insert the water heater:

```
INSERT INTO WaterHeater (ApplianceID, TankSize, CurrentTempSetting, EnergySource)  
VALUES ($applianceID, $tankSize, $currentTempSetting, $energySource)
```

- After adding an appliance, the **appliance listing** is shown.

View Appliance listing

Abstract Code

- After adding an appliance, the appliance listing will list each appliance's details.

```
SELECT a.ApplianceID, 'Water Heater' as Type, a.ManufacturerName, a.ModelName  
FROM WaterHeater w, Appliance a  
WHERE w.ApplianceID = a.ApplianceID and a.HouseholdID = $householdID  
  
UNION  
  
SELECT a.ApplianceID, 'Air Handler' as Type, a.ManufacturerName, a.ModelName  
FROM AirHandler ah, Appliance a  
WHERE ah.ApplianceID = a.ApplianceID and a.HouseholdID = $householdID
```

- There will be a view of listed appliances include appliance number, Type, Manufacturer and Model shown in the interface.
- There is a delete button in the last column on each row . If user click delete button, it will update DB to remove this record.

```
DELETE FROM Appliance WHERE ApplianceID = $applianceID
```

- If user click button of Add another appliance, it will link to **Add appliance** form.
- If user click Next button, it will jump to next form **Add power generation.**

Add power generation

Abstract Code

- Check if the household "off-the-grid":

```
$publicUtilitiesCount = SELECT count(UtilityName) FROM PublicUtility WHERE HouseholdID = $householdID
```

- If the household is NOT "off-the-grid", the user selects *Skip* and the power generation listing form is shown.
- If the household is "off-the-grid", the user enters the data and clicks *Add*:
 - The type will be validated against the allowed values in the dropdown.
 - The monthly kwh value will be validated as a non-negative whole number.
 - The storage kwh value, if provided, will be validated as a non-negative whole number.
 - If the data is valid, it is inserted into the DB:

Get the next available order number for this household.

```
$nextOrderNumber = SELECT COALESCE(Max(OrderNumber) + 1, 1) FROM PowerGenerator where HouseholdID = $householdID
```

Insert the power generation method.

```
INSERT INTO PowerGenerator (HouseholdID, OrderNumber, BatteryStorageCapacity, AvgMonthlyKwHours, GenerationType)
VALUES ($householdID, $nextOrderNumber, $batteryStorageCapacity, $avgMonthlyKwHours, $powerGenerationType)
```

View Power generation listing

Abstract Code

- View power generation: lookup power generation records and show a table includes columns of the generator type, monthly kilowatt hours, and battery storage capacity.

```
SELECT OrderNumber as Num, AvgMonthlyKwHours as 'Monthly kWh', BatteryStorageCapacity as 'Battery kWh'
FROM PowerGenerator
WHERE HouseholdID = $householdID
```

- There is a delete button in the last column on each row. If user click delete button, it will remove this record from the DB.

```
DELETE FROM PowerGenerator
WHERE HouseholdID = $householdID AND OrderNumber = $orderNumber
```

Phase 2 Abstract Code w/SQL | CS 6400 - Summer 2023 | Team 022

- If the user clicks the button *Add more power* link, they will be taken back to the **Add power generation** form.
- If the user clicks Next button, validate whether all generators are deleted, and household is “off-the-grid”:
 - This query sums the total utilities and power generators for the household. If the result is zero, then the house is off-the-grid with no power generators. The user cannot leave this screen until at least one generator has been added. A message indicating this is presented to the user.

```
SELECT SUM(Total) as Total
FROM
(
    SELECT count(UtilityName) as Total
    FROM PublicUtility
    WHERE HouseholdID = $householdID

    UNION

    SELECT count(HouseholdID) as Total
    FROM PowerGenerator
    WHERE HouseholdID = $householdID
) UtilitiesAndPowerGenerators
```

- Else the user can click *Finish* and will be directed to the **Wrapping Up** form.

Wrapping up

Abstract Code:

- After the user has finished adding or has skipped power generation information:
 - A thank you message should be displayed to them
 - If user click the link to the main menu, it will return to form **Main Menu**.

Reports-Top 25 popular manufacturers

Abstract Code:

- Lookup and count the appliances number of each manufacturer. Order top 25 manufacturer by the number of appliances.

```
SELECT ManufacturerName, count(ManufacturerName) as TotalAppliances
FROM Appliance
GROUP BY ManufacturerName
ORDER BY TotalAppliances DESC
```

LIMIT 25;

- There is a button from the parent report's row for that manufacturer to provide an option to view a drilldown report for a particular manufacturer.
 - When user click the drilldown button, it will look up the appliances produced by certain manufacturer name.
 - Count appliances belonging to each appliance type produced by that manufacturer (as an integer)
 - Display the manufacturer name at the top, with a table listing the count of appliances belonging to each appliance type produced by that manufacturer.

```
SELECT 'Water Heater' as Type, COUNT(*) as Total
FROM WaterHeater w, Appliance a
WHERE a.ManufacturerName = $manufacturerName
      AND w.ApplianceID = a.ApplianceID
```

UNION

```
SELECT 'Air Handler' as Type, COUNT(*) as Total
FROM AirHandler ah, Appliance a
WHERE a.ManufacturerName = $manufacturerName
      AND ah.ApplianceID = a.ApplianceID
```

Reports- Manufacturer/model search

Abstract Code:

- User enters the search string.
- User clicks the search button:
 - Validate that the search string is not empty.
 - Search for all manufacturers or models that match the entered value(\$searchString).

```
SELECT ManufacturerName, ModelName
FROM Appliance
WHERE LOWER(ManufacturerName) LIKE CONCAT('%', LOWER($searchString), '%')
      OR LOWER(ModelName) LIKE CONCAT('%', LOWER($searchString), '%')
ORDER BY ManufacturerName ASC, ModelName ASC;
```

- Display the resulting manufacturers and models.
- If user click the link to go back to the main menu, return to the Main Menu.

Reports-Heating/cooling method details

Abstract Code:

- Lookup records grouped and ordered by household type
 - Count *air conditioners*, average air conditioner *BTUs* (as a whole number, rounded), average *RPM* (as a decimal number, rounded to tenths) and the average *EER* (as a decimal number, rounded to tenths).

```
SELECT
  h.HouseholdType,
  COUNT(DISTINCT ac.AirConditionerID) AS AirConditionerCount,
  ROUND(AVG(a.BtuRating)) AS AvgAirConditionerBTU,
  ROUND(AVG(ah.Rpm), 1) AS AvgAirHandlerRPM,
  ROUND(AVG(ac.EER), 1) AS AvgAirConditionerEER
FROM Household h LEFT JOIN Appliance a ON h.HouseholdID = a.HouseholdID
  LEFT JOIN AirHandler ah ON a.ApplianceID = ah.ApplianceID
  LEFT JOIN AirConditioner ac ON ah.AirHandlerID = ac.AirHandlerID
GROUP BY h.HouseholdType
ORDER BY h.HouseholdType ASC;
```

- Count of heaters, average heater *BTUs* (as a whole number, rounded), average *RPM* (as a decimal number, rounded to tenths) and the most common energy source.

```
SELECT
  h.HouseholdType,
  COUNT(DISTINCT he.HeaterID) AS HeaterCount,
  ROUND(AVG(a.BtuRating)) AS AvgHeaterBTU,
  ROUND(AVG(ah.Rpm), 1) AS AvgHeaterRPM,
  (SELECT EnergySource FROM Heater WHERE AirHandlerID = ah.AirHandlerID
GROUP BY EnergySource ORDER BY COUNT(*) DESC LIMIT 1) AS
MostCommonEnergySource
FROM Household h
  LEFT JOIN Appliance a ON h.HouseholdID = a.HouseholdID
  LEFT JOIN AirHandler ah ON a.ApplianceID = ah.ApplianceID
  LEFT JOIN Heater he ON ah.AirHandlerID = he.AirHandlerID
GROUP BY h.HouseholdType
ORDER BY h.HouseholdType ASC;
```

- Count of heat pumps, average heat pump *BTUs* (as a whole number, rounded), average *RPM* (as a decimal number, rounded to tenths), the average *SEER* (as a decimal number, rounded to tenths) and the average *HSPF*(as a decimal number, rounded to tenths).

```
SELECT
  h.HouseholdType,
```

```
COUNT(DISTINCT hp.HeatPumpID) AS HeatPumpCount,  
ROUND(AVG(a.BtuRating)) AS AvgHeatPumpBTU,  
ROUND(AVG(ah.Rpm), 1) AS AvgHeatPumpRPM,  
ROUND(AVG(hp.SEER), 1) AS AvgHeatPumpSEER,  
ROUND(AVG(hp.HSPF), 1) AS AvgHeatPumpHSPF  
FROM Household h  
      LEFT JOIN Appliance a ON h.HouseholdID = a.HouseholdID  
      LEFT JOIN AirHandler ah ON a.ApplianceID = ah.ApplianceID  
      LEFT JOIN HeatPump hp ON ah.AirHandlerID = hp.AirHandlerID  
GROUP BY h.HouseholdType  
ORDER BY h.HouseholdType ASC;
```

- Display count output and household email information on a table.
- If user click the link to go back to the main menu, return to the Main Menu.

Reports-Water heater statistics by state

Abstract Code:

- Look up all records in water heater appliance type and group by state
 - Calculate the average water heater tank size (a whole number, rounded),
 - Calculate the average water heater BTUs (a whole number, rounded),
 - Calculate the average water heater temperature setting (a decimal number, rounded to tenths)
 - count of water heaters where a temperature setting has been provided,
 - count of water heaters where no temperature setting has been provided,
 - Display all output in a table and sorted by state abbreviation ascending

```
SELECT  
  pc.State,  
  ROUND(AVG(wh.TankSize)) AS AvgTankSize,  
  ROUND(AVG(a.BtuRating)) AS AvgWaterHeaterBTU,  
  ROUND(AVG(wh.CurrentTempSetting), 1) AS AvgTemperatureSetting,  
  COUNT(CASE WHEN wh.CurrentTempSetting IS NOT NULL THEN 1 END)  
AS TempSettingProvidedCount,  
  COUNT(CASE WHEN wh.CurrentTempSetting IS NULL THEN 1 END) AS  
TempSettingNotProvidedCount  
FROM WaterHeater wh  
  JOIN Appliance a ON wh.ApplianceID = a.ApplianceID  
  JOIN Household hh ON a.HouseholdID = hh.HouseholdID  
  JOIN PostalCode pc ON hh.PostalCode = pc.PostalCode  
GROUP BY pc.State  
ORDER BY pc.State;
```

Phase 2 Abstract Code w/SQL | CS 6400 - Summer 2023 | Team 022

- User click button from the parent report's row for that state to generate a drilldown report for that particular state
 - Select this particular state as report header group by energy source
 - Calculate the minimum water heater tank size (a whole number, rounded)
 - Calculate the average water heater tank size (a whole number, rounded)
 - Calculate the maximum water heater tank size (a whole number, rounded)
 - Calculate the minimum temperature setting
 - Calculate maximum temperature setting
 - Calculate the average temperature setting (decimal number, rounded to tenths)
 - Energy sources should be ordered in ascending order.

```
SELECT
  ROUND(MIN(wh.TankSize)) AS MinTankSize,
  ROUND(AVG(wh.TankSize)) AS AvgTankSize,
  ROUND(MAX(wh.TankSize)) AS MaxTankSize,
  MIN(wh.CurrentTempSetting) AS MinTempSetting,
  MAX(wh.CurrentTempSetting) AS MaxTempSetting,
  AVG(wh.CurrentTempSetting) AS AvgTempSetting
FROM WaterHeater wh
JOIN Appliance a ON wh.ApplianceID = a.ApplianceID
JOIN Household hh ON a.HouseholdID = hh.HouseholdID
JOIN PostalCode pc ON hh.PostalCode = pc.PostalCode
WHERE pc.State = '$state'
ORDER BY EnergySource
```

- If user click the link to go back to the main menu, return to the Main Menu.

Reports-Off-the-grid household dashboard

Abstract Code:

- Look up the state with the most off-the-grid households and calculate its count.

```
SELECT
  pc.State,
  COUNT(DISTINCT hh.HouseholdID) AS OffGridHouseholds
FROM Household AS hh
JOIN PostalCode AS pc ON hh.PostalCode = pc.PostalCode
WHERE hh.HouseholdID NOT IN (SELECT pu.HouseholdID
FROM PublicUtility AS pu)
GROUP BY pc.State
ORDER BY OffGridHouseholds DESC LIMIT 1;
```

Phase 2 Abstract Code w/SQL | CS 6400 - Summer 2023 | Team 022

- Retrieve all the off-the-grid households:
 - Calculate the average battery storage capacity.

```
SELECT      ROUND(AVG(pg.BatteryStorageCapacity),      0)      as
AverageBatteryStorageCapacity
FROM PowerGenerator pg
WHERE pg.HouseholdID IN (
    SELECT HouseholdID FROM Household
    WHERE HouseholdID NOT IN (
        SELECT HouseholdID FROM PublicUtility)
```

- Calculate the percentages for each power generation type.

```
SELECT
    gt.GenerationType,
    COALESCE(
        NULLIF(
            ROUND(
                100 * (COUNT(pg.GenerationType) * 1.0 / (SELECT COUNT(GenerationType) FROM
PowerGenerator) * 1.0)
            ,1),
            0), '0%'
        ) as '%'
FROM (
    select 'Solar' as GenerationType union all
    select 'Wind-turbine' union all
    select 'Mixed') gt
LEFT OUTER JOIN PowerGenerator pg
ON LOWER(pg.GenerationType) = LOWER(gt.GenerationType)
GROUP BY gt.GenerationType
```

- Calculate the percentages for each household type.

```
SELECT
    ht.HouseholdType,
    COALESCE(
        NULLIF(
            ROUND(
                100 * (COUNT(hh.HouseholdType) * 1.0 / (SELECT COUNT(HouseholdType) FROM
Household) * 1.0)
            ,1),
            0), '0%'
        ) as '%'
FROM (
    select 'House' as HouseholdType union all
    select 'Apartment' union all
```

```
select 'Townhome' union all
select 'Condominium' union all
select 'Modular home' union all
select 'Mixed') ht
```

```
LEFT OUTER JOIN Household hh
ON LOWER(hh.HouseholdType) = LOWER(ht.HouseholdType)
GROUP BY ht.HouseholdType
```

- Calculate the average water heater tank size.

```
SELECT
ROUND(AVG(wh1.TankSize), 1) as AvgOffGridTankSize,
ROUND(AVG(wh2.TankSize), 1) as AvgOnGridTankSize
FROM Appliance ap1, WaterHeater wh1, Appliance ap2, WaterHeater wh2

WHERE ap1.ApplianceID = wh1.ApplianceID
AND ap1.HouseholdID IN (
    SELECT HouseholdID FROM Household
    WHERE HouseholdID NOT IN (
        SELECT HouseholdID FROM PublicUtility)
    )

AND ap2.HouseholdID NOT IN (
    SELECT HouseholdID FROM Household
    WHERE HouseholdID NOT IN (
        SELECT HouseholdID FROM PublicUtility)
    )
```

- Calculate the minimum, average, and maximum BTU values.

```
SELECT
'Water Heater' as ApplianceType,
IFNULL( ROUND(MIN(a.BtuRating), 0), 0) as MinBTU,
IFNULL(ROUND(MAX(a.BtuRating), 0), 0) as MaxBTU,
IFNULL(ROUND(AVG(a.BtuRating), 0), 0) as AvgBTU
FROM Appliance a, WaterHeater w
WHERE a.ApplianceID = w.ApplianceID
GROUP BY ApplianceType

UNION

SELECT
'Air Handler' as ApplianceType,
IFNULL(ROUND(MIN(a2.BtuRating), 0), 0) as MinBTU,
IFNULL(ROUND(MAX(a2.BtuRating), 0), 0) as MaxBTU,
IFNULL(ROUND(AVG(a2.BtuRating), 0), 0) as AvgBTU
FROM Appliance a2, AirHandler ah
WHERE a2.ApplianceID = ah.ApplianceID
GROUP BY ApplianceType
```

- Calculate the average water heater gallon capacity.

- If user click the link to go back to the main menu, return to the Main Menu.

Reports-Household averages by radius

Abstract Code:

- User enter the postal code to center the search on, and the search radius (a whole number, with the following choices available: 0, 5, 10, 25, 50, 100, and 250).

When the User enters a postal code('\$postalCode'), validate that it exists in the database:

```
SELECT COUNT(PostalCode) FROM PostalCode WHERE PostalCode = '$postalCode';
```

- User click search to trigger validation for both the postal code and radius input fields
 - If validation succeed,
 - Retrieve the longitude and latitude data from the postal code table
 - Calculate the distances between the postal codes using the haversine formula
 - Lookup all households whose postal codes in calculated range

```
SELECT
    h.PostalCode,
    COUNT(h.HouseholdID),
    h.HouseholdType,
    AVG(HeatingSetting, 0),
    AVG(CoolingSetting)
FROM Household h
WHERE h.PostalCode IN
(
    SELECT
        DISTINCT(p.PostalCode)
        FROM PostalCode p, PostalCode p2
        WHERE p.PostalCode = '11745'
            AND (acos(sin(radians(p.Latitude))*sin(radians(p2.Latitude)))
                + cos(radians(p.Latitude))*cos(radians(p2.Latitude))*cos(radians(p2.Longitude)-
                    radians(p.Longitude))) * 3958.75) <= 5
)
GROUP BY h.HouseholdType
```

- If the data validation fails for any of the input fields, an error message is displayed.
- If user click the link to go back to the main menu, return to the Main Menu.

