Table of Contents

Data Types

Household

Attribute	Data Type	Nullable
email	String	Not null
home_type	String	Not null
square_footage	Integer	Not null
occupant_count	Integer	Not null
bedroom_count	Integer	Not null

Location

Attribute	Data Type	Nullable
postal_code	String	Not null
city	String	Not null
state	String	Not null
longitude	String	Not null
latitude	String	Not null

Phone_Number

Attribute	Data type	Nullable
area_code	String	Not Null
phone_number	String	Not Null
phone_type	String	Not Null

Manufacturer

Attribute	Data type	Nullable
manufacturer_name	String	Not Null

Appliance

Attribute	Data type	Nullable
appliance_id	Integer	Not Null
model_name	String	Null

Refrigerator_Freezer

Attribute	Data type	Nullable
refrigerator_type	String	Not Null

Cooker

Attribute	Data type	Nullable
cooker_type	String	Not Null

Oven

Attribute	Data type	Nullable
oven_type	String	Not Null
heat_source	List<String>	Not Null

Cooktop

Attribute	Data type	Nullable
heat_source	String	Not Null

Washer

Attribute	Data type	Nullable
loading_type	String	Not Null

Dryer

Attribute	Data type	Nullable
heat_source	String	Not Null

TV

Attribute	Data type	Nullable
display_type	String	Not Null
display_size	Float	Not Null
max_resolution	String	Not Null

Bathroom

Attribute	Data type	Nullable
bathroom_id	Integer	Not Null
sink_count	Integer	Not Null
commode_count	Integer	Not Null
bidet_count	Integer	Not Null

Full

Attribute	Data type	Nullable
shower_count	Integer	Not Null
bathtub_count	Integer	Not Null
tub_shower_count	Integer	Not Null
is_primary	Boolean	Not Null

Half

Attribute	Data type	Nullable
name	String	Null

Business Logic Constraints

Hemkraft

- The Location entity is not updatable.
- The Location entity will contain data preset by database administrators.
- The Manufacturer entity is not updateable.
- The Manufacturer entity will contain data preset by database administrators.
- Users may not edit or update their household information.
- Users may only enter their household information into Hemkraft once. A unique email is used to identify a single household.
- All users can browse and view reports from the Hemkraft website.
- If the application crashes or the user closes/exits the screen before completing all steps of entering household information (adding an email, optional phone number, household details, bathroom(s), and appliance(s)), all partial data will be deleted from the system.
- Users should provide all the required fields to save the household details. Users cannot skip any of the required fields and save only partial detail.
- Users cannot navigate back to the previous screen and change the details that are already entered.
- The system supports multiple users.
- A single user can have multiple sessions active at a time to view the report, However the same does not apply while entering the household details. A user can have only one active session to enter household information to the system.
- Reports must never display a blank page or empty table. If any reports generate no results, an appropriate message is conveyed to the user.
- All NULL values will be translated into an empty string wherever necessary in any report.
- If parameters are required to view a report, users must click a “Submit” button to generate the reports.
- A half bathroom must have at least one commode, bidet, and/or sink.
- A household can only have one primary bathroom.
- A full bathroom must have at least one bathtub, shower, or tub/shower.

Task Decomposition with Abstract Code

Enter Information

Email Address

Task Decomposition

Lock Type: 1 read lock on Household entity.
Number of Locks: 1.
Enabling Conditions:

- The lookup on the Household entity is triggered by a user clicking the ***Submit*** button on the **Email Address** form.

Frequency: Low – once per household.
Consistency (ACID): is not critical, even if another user is viewing reports while the user is entering their household information.
Subtasks: Mother task is not needed. No decomposition needed.

Abstract Code

```
1. User clicks Enter my household info button on the Main Menu.
2. Display Email Address form.
3. While the email input field is not inputted or valid:
      a. email ('$email') input field is filled and validated
      b. If the Submit button is clicked:
            i. Run the Verify Email task:
                  1. If '$email' is already found in the Household
                     entity
                        a. Display an error message indicating that
                           the email is already in use
                        b. Erase the input and return to step 2.
                  2. Else
                        a. Store '$email' as a session variable.
```

Postal Code

Task Decomposition

Lock Type: 1 read lock on Location entity.
Number of Locks: 1
Enabling Conditions:

- The lookup on the Location entity is triggered by a user clicking the ***Submit*** button on the **Postal Code Form**.

Frequency: Low – once per household.
Consistency (ACID): is not critical, even if another user is viewing reports while the user is entering their household information.
Subtasks: Mother task is not needed. No decomposition needed.

Abstract Code

```
1. Display Postal Code Form.
2. While postal code input field is not inputted or valid:
      a. postal code ('$postal_code') field is inputted
      b. If Submit button is clicked:
            i. Run the Verify Postal Code task:
                  1. If the '$postal_code' is not found in the
                     Location entity
                        a. Display an error message indicating that
                           the postal code input was invalid.
                        b. Erase the input and return to step 1.
                  2. Else
                        a. Display confirmation page with the
                           Location.PostalCode, Location.City and
                           Location.State for the matching record
                           found in the Location entity with Yes
                           button and No button.
                        b. If No button is clicked:
                              i. Jump to step 1.
                        c. Else, if Yes button is clicked:
                              i. Store '$postal_code' as a session
                                 variable.
```

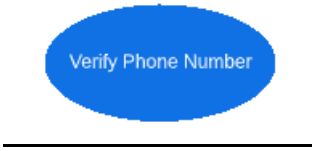
Phone Number

Task Decomposition

Lock Type: 1 read on Phone_Number entity.
Number of Locks: 1.
Enabling Conditions:

- The lookup on the Phone_Number entity is triggered by a user clicking the ***Next*** button on the **Phone Number Form**.

Frequency: Low – once per household.
Consistency (ACID): is not critical, even if another user is viewing reports while the user is entering their household information.
Subtasks: Mother task is not needed. No decomposition needed.



Abstract Code

1. Display **Phone Number Form**.
2. Display prompt with ***Yes*** or ***No*** toggle to enter a phone number.
    a. If ***Yes*** toggle is clicked:
        i. While the *area code* input field is not inputted, and the *phone number* is not inputted, and the *phone type* is not inputted
            1. The *area code* ('$area_code') input field is filled by user.
            2. The *number* ('$phone_number') input field is filled by user.
            3. The *phone type* ('$phone_type') is selected from drop down menu.
            4. If the *area code*, *phone number*, and *phone type* inputs are all filled:
                a. Jump to step 2.a.ii.
            5. Else
                a. Jump to step 2a
        ii. If the ***Next*** button is clicked:
            1. Run the **Verify Phone Number** task:
                a. If the combination of the '$area_code' and '$phone_number' input fields is found in the Phone_Number entity:

```
        i. Display an error message stating that
           '$phone_number' is already in use.
       ii. Erase the input and return to step 1.
   b. Else:
        i. Store '$area_code' and '$phone_number'
           as session variables
```

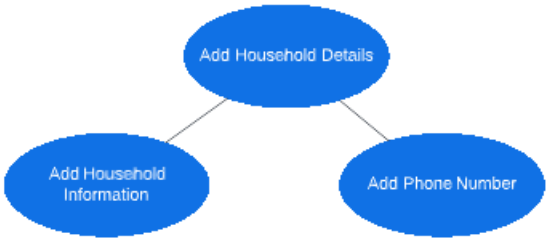
Household Information

Task Decomposition

Lock Type: 1 write lock on Household entity. 1 write lock on Phone_Number entity.
Number of Locks: 2.
Enabling Conditions:

- The write-lock on the Household entity is triggered by a user clicking the ***Next*** button on the **Household Details Form**.
- The write-lock on the Phone_Number entity is triggered after the household information is entered into the Household entity.

Frequency: Low – both locks have the same frequency.
Consistency (ACID): is not critical, even if another user is viewing reports while the user is entering their household information.
Subtasks: Mother task is needed to coordinate subtasks for entering household information. Task decomposition is needed to ensure that the subtasks are sequenced correctly.



Abstract Code

1. Display **Household Details Form**.
    a. User selects *home type* (‘$home_type’) from dropdown.
    b. *square footage* (‘$square_footage’) input field is filled by the user
    c. *occupant count* (‘$occupant_count’) input field is filled by the user
    d. *bedroom count* (‘$bedroom_count’) input field is filled by the user
    e. If the ***Next*** button is clicked:
        i. If *square footage*, *occupant count*, and *bedroom count* input fields are not inputted or are invalid:
            1. Display an error message indicating that *square footage*, *occupant count*, and *bedroom count* input fields are invalid.
            2. Erase inputs and return to step 8.
        ii. Else

1. Store '$home_type', '$square_footage', '$occupant_count', and '$bedroom_count' as session variables.
2. Jump to step 2.

2. If the ***Next*** button is clicked on the **Household Details Form**:
   a. Add the household information to the Household and Phone_Number entities by running the **Add Household Details** task:
      i. Run **Add Household Information** task: Add a new household in the Household entity with the '$email' as the Household.Email, the '$home_type' as the Household.HomeType, '$square_footage' as the Household.SquareFootage, '$occupant_count' as the Household.OccupantCount, and '$bedroom_count' as the Household.BedroomCount, associate the household with the record in the Location entity where Location.PostalCode == '$postal_code'
      ii. Run **Add Phone Number** task: Add the '$area_code' and '$phone_number' as the phone number to the Phone_Number entity for the household with the matching email from the Household entity.

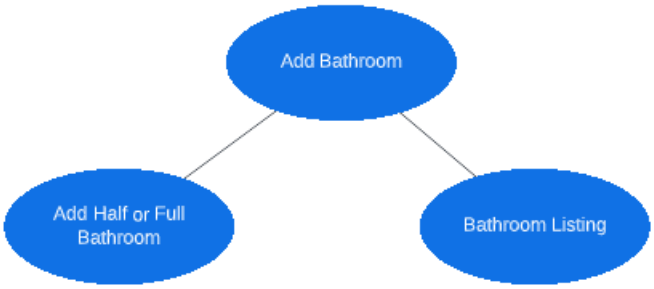
Add Bathroom

Task Decomposition

Lock Types: 2 read locks on Full entity. 1 read lock on Bathroom. 1 read lock on Half entity. 1 write lock on Half entity, 1 write lock on Full entity, and 1 write lock on Bathroom entity.
Number of Locks: 7.
Enabling Conditions:

- The first lookup (read lock) on the Full entity is triggered by the user clicking "Full" as the bathroom type on the **Add Bathroom Form** to check if the household already has a primary bathroom.
- The write-lock on the Full entity is triggered by the user clicking the ***Add*** button on the **Add Bathroom Form** and has selected "Full" as the bathroom type.
- The write-lock on the Half entity is triggered by the user clicking the ***Add*** button on the **Add Bathroom Form** and has clicked "Half" as the bathroom type.
- The write-lock on the Bathroom entity is triggered by the user clicking the ***Add*** button on the **Add Bathroom Form**, regardless of the selection of bathroom type.
- The second lookup (read lock) on the Full entity is triggered by the user clicking the ***Add*** button on the **Add Bathroom Form** and the write-lock on the Full entity has completed.
- The lookup (read lock) on the Bathroom entity is triggered by the user clicking the ***Add*** button on the **Add Bathroom Form** and the write-lock on the Bathroom entity has completed.
- The lookup (read lock) on the Half entity is triggered by the user clicking the ***Add*** button on the **Add Bathroom Form** and the write-lock on the Half entity has completed.

Frequency: Medium - a single user can enter details of multiple bathrooms while entering their household information.
Consistency (ACID): Order is not critical.
Subtasks: Mother task is needed to coordinate subtasks. **Add bathroom** is the mother task. The subtask **Add Half or Full Bathroom** must execute before the **Bathroom Listing** task. Order is critical.



Abstract Code

```
1. While the Hemkraft site is still open:
      a. Display Add Bathroom Form with options of 'Half' or 'Full'
         for the bathroom type
      b. While Add button is not clicked, do nothing.
      c. If user selects 'Half' as bathroom type ('$bathroom_type')
         in Add Bathroom Form.
```

i. *sink count* (‘$sink_count’) input field is filled
ii. *commode count* (‘$commode_count’) input field is filled
iii. *bidet count* (‘$bidet_count’) input field is filled
iv. *name* (‘$name’) input field is filled
v. If ***Add*** button is clicked, do the following:
    1. If any of the *sink count*, *commode count*, *bidet count* fixtures are negative
        a. Display **Add Bathroom Form** with error message indicating that none of the *sink count*, *commode count*, *bidet count* input fields can be negative.
        b. Erase input and jump to step 1c.
    2. Else if the sum of count of the ‘$sink_count’, ‘$commode_count’, and ‘$bidet_count’ is not greater than 0
        a. Display **Add Bathroom Form** with error message indicating that the total of the sum of *sink count*, *commode count*, *bidet count* should be greater than zero.
        b. Erase input and jump to step 1c.
    3. Else
        a. Run **Add Half or Full Bathroom** task: Add a bathroom in the Bathroom entity for the household where Household.Email == ‘$email’ with the ‘$bidet_count’ as Bathroom.BidetCount, ‘$sink_count’ as Bathroom.SinkCount, ‘$commode_count’ as Bathroom.CommodeCount, and ‘$name’ as Half.name to Half entity if the user provided the optional half bathroom name, where the half bathroom is the same as the bathroom.

d. Else if user selects ‘Full’ as *bathroom type* (‘$bathroom_type’) in **Add Bathroom Form**.
    i. For all bathrooms in the Bathroom entity where the bathroom belongs to the household where Household.Email == ‘$email’ and the *bathroom type* is Full
        1. If Full.IsPrimary == 1,
            a. Disable the ***This bathroom is a primary bathroom*** checkbox
    ii. While ***Add*** button is not clicked, do nothing.
    iii. User enters *sink count* (‘$sink_count’), *commode count* (‘$commode_count’), *bidet count* (‘$bidet_count’), *bathtub count* (‘$bathtub_count’), *shower/tub count* (‘$showertub_count’), *shower count* (‘$shower_count’)

and if the checkbox is enabled, the user also enters if bathroom *is primary* ('$is_primary').

iv. When ***Add*** button is clicked, do the following:

1. If any of the of the '$sink_count', '$commode_count', '$bidet_count', '$bathtub_count', '$showertub_count', or '$shower_count' values are negative
    a. Display **Add Bathroom Form** with error message indicating that none of the *sink count*, *commode count*, *bidet count*, *bathtub count*, *shower/tub count*, or *shower count* input fields can be negative.
    b. Erase input and jump to step 1d
2. Else if the sum of the '$bathtub_count', '$showertub_count', and '$shower_count' is not greater than zero
    a. Display **Add Bathroom Form** with error message indicating that the sum of the *bathtub count*, *shower/tub count*, and *shower count* input fields must be greater than zero.
    b. Erase input and jump to step 1d
3. Else
    a. Run **Add Half or Full Bathroom** task: Add a bathroom in the Bathroom entity for the household where Household.Email == '$email' with the '$bidet_count' as Bathroom.BidetCount, '$sink_count' as Bathroom.SinkCount, '$commode_count' as Bathroom.CommodeCount to Bathroom entity and add '$bathtub_count' as Full.BathtubCount, '$showertub_count' as Full.TubShowerCount, '$shower_count' as Full.ShowerCount, '$is_primary' as Full.IsPrimary in the Full entity, where the full bathroom is the same as the bathroom.
    b. Jump to step 1e

e. Run **Bathroom Listing** task:

i. For each record in Bathroom entity for the household where Household.Email == '$email':

1. Display the Bathroom.BathroomID
2. If the Bathroom.BathroomID is in the Full entity
    a. Display "Full" for the bathroom type and "Yes" if Full.IsPrimary else display an

empty string for if the bathroom is a primary bathroom

3. Else if the Bathroom.BathroomID is in the Half entity
    a. Display "Half" for the bathroom type and an empty string for if the bathroom is a primary bathroom

ii. Jump to step 1f.

f. Display a button to ***Add another bathroom***, and a ***Next*** button.

i. If the ***Add another bathroom*** button is clicked,
    1. Display **Add Bathroom Form**
    2. Jump to step 1a.

ii. Else if the ***Next*** button is clicked,
    1. Display **Add Appliance Form**.

2. If the user closes the Hemkraft website or the Hemkraft website unexpectedly crashes before a bathroom is added:
    a. Clear all partial data for the household where the '$email' == Household.Email from the Household and Phone_Number entities.

Add Appliance

Task Decomposition

Lock Types: 2 read locks on the Manufacturer entity. 1 read lock on the Appliance entity. 1 read lock on each of the Appliance subtype entities: Refrigerator_Freezer, Cooker, Oven, Cooktop, Washer, Dryer, and/or TV. 1 write lock on one of the Appliance subtype entities: Refrigerator_Freezer, Cooker, Oven, Cooktop, Washer, Dryer, and/or TV. 1 write lock on the Appliance entity.

Number of Locks: 6

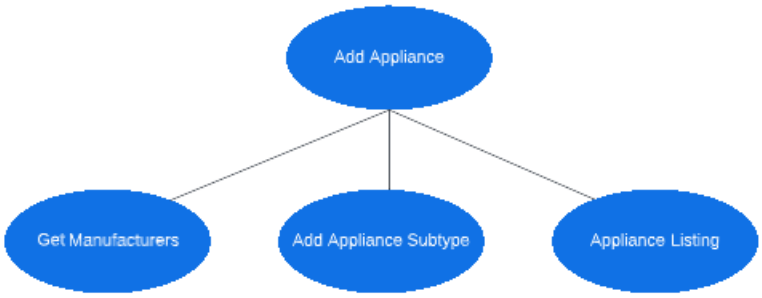
Enabling Conditions:

- The first lookup (read lock) on the Manufacturer entity is triggered by the user clicking the ***Next*** button on the **Bathroom Listing** page, when the page is navigated to the **Add Appliance Form.**
- The write lock on the Appliance entity is triggered by the user clicking the ***Add*** button on the **Add Appliance Form**.
- The write lock on one of the Appliance subtype entities (Refrigerator_Freezer, Cooker, Oven, Cooktop, Washer, Dryer, and/or TV) is triggered by the user clicking the ***Add*** button on the **Add Appliance Form** after the write lock for the Appliance entity has completed.
- The second lookup (read lock) on the Manufacturer entity is triggered by the user clicking the ***Add*** button on the **Add Appliance Form** and the write lock has completed to display the **Appliance Listing** page.
- The lookup (read lock) on the Appliance entity is triggered by the user clicking the ***Add*** button on the **Add Appliance Form** and the write lock has completed to display the **Appliance Listing** page.
- The lookup (read lock) on one of the Appliance subtype entities (Refrigerator_Freezer, Cooker, Oven, Cooktop, Washer, Dryer, and/or TV) is triggered by the user clicking the ***Add*** button on the **Add Appliance Form** and the write lock has completed to display the **Appliance Listing** page.

Frequency: Medium - a single user can enter details of multiple bathrooms while entering their household information.

Consistency (ACID): not critical. Order is not critical.

Subtasks: Mother task is required to coordinate subtasks. **Add Appliances** is the mother task. The subtasks for retrieving manufacturers and adding an appliance type must be run in that order before the **Appliance Listing** subtask executes.



Abstract Code

1. While the Hemkraft website is still open:
    a. Populate the *appliance type* dropdown with 'Refrigerator/Freezer', 'Cooker', 'Washer', 'Dryer', and 'TV'.
    b. Populate the *manufacturer name* input field with all Manufacturer.ManufacturerName from the Manufacturer entity
    c. While the *appliance type* is not selected and *manufacturer name* is not selected:
        i. *appliance type* ('$appliance_type') is selected from dropdown by the user
        ii. *manufacturer name* ('$manufacturer_name') is selected from dropdown by the user
        iii. *model name* ('$model_name') input field is optionally filled by the user.
        iv. If the *appliance type* and *manufacturer name* fields are selected:
            1. Jump to step 1d.
        v. Else
            1. Jump to step 1c.
    d. If '$appliance_type' == 'Cooker':
        i. Display **Add Appliance – Cooker Form**.
        ii. While ***Add*** button is not clicked, do nothing.
        iii. If ***oven*** is checked,
            1. The user selects one or more *oven heat source* ('$oven_heat_source') options
            2. The user selects the *oven type* ('$oven_type') from the dropdown options of 'Convection' or 'Conventional'
        iv. If ***cooktop*** is checked,
            1. The user selects the *cooktop heat source* ('$cooktop_heat_source') from the dropdown.

v. If ***Add*** button is clicked, do the following:
    1. If ***oven*** is checked and *oven heat source* is selected,
        a. Run the **Add Appliance Subtype** task:
            i. Add an appliance to the Appliance entity with the '$model_name' as Appliance.ModelName if '$model_name' is not empty, and oven to the Oven entity where the oven is the same as the appliance and the oven is a Cooker with '$oven_heat_source' as Oven.HeatSource, '$oven_type' as Oven.OvenType
        b. Jump to step 1i.
    2. Else if *oven* is checked and *oven heat source* is not selected
        a. Display error message indicating that *oven heat source* for the oven must be selected.
        b. Jump to step 1.d.iii.
    3. If *cooktop* is checked and *cooktop heat source* is selected,
        a. Run the **Add Appliance Subtype** task:
            i. Add an appliance to the Appliance entity with the '$model_name' as Appliance.ModelName if '$model_name' is not empty, and cooktop to the Cooktop entity where the cooktop is the same as the appliance and the cooktop is a Cooker with '$cooktop_heat_source' as Cooktop.HeatSource
        b. Jump to step 1i.
    4. Else if *cooktop* is checked and *cooktop heat source* is not selected
        a. Display error message indicating that *cooktop heat source* for the cooktop must be selected.
        b. Jump to step 1.d.iv.

e. Else if '$appliance_type' == 'TV':
    i. Display **Add Appliance – TV Form**.
    ii. While ***Add*** button is not clicked, do nothing.
    iii. While the *display type* is not selected and *display size* input field is not filled and *maximum resolution* is not selected:

1. *display type* (‘$display_type’) is selected from dropdown
2. *display size* (‘$display_size’) input field is filled
3. *max resolution* (‘$max_resolution’) is selected from dropdown.
4. If the *display type* is selected and *display size* input field is filled and *maximum resolution* is selected:
    a. Jump to step 1f.
5. Else
    a. Display error message indicating that all *display type* and *display size* and *max resolution* inputs must be filled.
    b. Jump to step 1.e.iii.

iv. If ***Add*** button is clicked, do the following:
1. Run the **Add Appliance Subtype** task:
    a. Add an appliance to the Appliance entity with the ‘$model_name’ as Appliance.ModelName if ‘$model_name’ is not empty, and a television to the TV entity where the television is the same as the appliance and the television is a TV with ‘$display_type’ as TV.DisplayType, ‘$display_size’ as TV.DisplaySize, ‘$max_resolution’ as TV.MaxResolution,
    b. Jump to step 1i.

f. Else if ‘$appliance_type’ == ‘Washer’:
i. Display **Add Appliance – Washer Form**.
ii. While ***Add*** button is not clicked, do nothing.
iii. While the *loading type* is not selected from the dropdown of option types:
1. User selects *loading type* (‘$loading_type’) from dropdown
2. If the *loading type* is selected:
    a. Jump to step 1.f.iv.
3. Else
    a. Display error message indicating that a *loading type* must be selected.
    b. Jump to step 1.f.iii.

iv. If ***Add*** button is clicked, do the following:
1. Run the **Add Appliance Subtype** task:
    a. Add an appliance to the Appliance entity with the ‘$model_name’ as Appliance.ModelName if ‘$model_name’ is not

empty, and a washer to the Washer entity where the washer is the same as the appliance and the washer is a Washer with '$loading_type' as Washer.LoadingType

2. Jump to step 1i.

g. Else if '$appliance_type' == 'Dryer':
   i. Display **Add Appliance – Dryer Form**.
   ii. While ***Add*** button is not clicked, do nothing.
   iii. While the *dryer heat source* is not selected from the dropdown of option types:
      1. User selects *dryer heat source* ('$dryer_heat_source') from dropdown
      2. If the *dryer heat source* is selected:
         a. Jump to step 1.g.iv.
      3. Else
         a. Display error message indicating that a *dryer heat source* must be selected.
         b. Jump to step 1.g.iii.
   iv. When ***Add*** button is clicked, do the following:
      1. Run the **Add Appliance Subtype** task:
         a. Add an appliance to the Appliance entity with the '$model_name' as Appliance.ModelName if '$model_name' is not empty, and a dryer to the Dryer entity where the dryer is the same as the appliance and the dryer is a Dryer with '$dryer_heat_source' as Dryer.HeatSource
      2. Jump to step 1i.

h. Else if '$appliance_type' == 'Refrigerator/Freezer':
   i. Display **Add Appliance – Refrigerator/Freezer Form**.
   ii. While ***Add*** button is not clicked, do nothing.
   iii. While the *refrigerator type* is not selected from the dropdown of option types:
      1. User selects *refrigerator type* ('$refrigerator_type') from dropdown
      2. If the *refrigerator type* is selected:
         a. Jump to step 1.h.iv.
      3. Else
         a. Display error message indicating that a *refrigerator type* must be selected.
         b. Jump to step 1.h.iv.
   iv. When ***Add*** button is clicked, do the following:
      1. Run the **Add Appliance Subtype** task:
         a. Add an appliance to the Appliance entity with the '$model_name' as

Appliance.ModelName if '$model_name' is not empty, and a refrigerator to the Refrigerator_Freezer entity where the refrigerator is the same as the appliance and the refrigerator is a Refrigerator_Freezer with '$refrigerator_type' as Refrigerator_Freezer.RefrigeratorType

2. Jump to step 1i.

i. Run the **Appliances Listing** task:

i. For each appliance in the household where Household.Email == '$email'

1. Display the Appliance.ApplianceID from the Appliance entity
2. If the Appliance.ApplianceID is in Refrigerator_Freezer entity
   a. Display 'Refrigerator/Freezer' as "Type"
3. Else if the Appliance.ApplianceID is in Oven or Cooktop entities
   a. Display 'Cooker' as "Type"
4. Else if the Appliance.ApplianceID is in TV entity
   a. Display 'TV' as "Type"
5. Else if the Appliance.ApplianceID is in Dryer entity
   a. Display 'Dryer' as "Type"
6. Else if the Appliance.ApplianceID is in Washer entity
   a. Display 'Washer' as "Type"
7. Display the Manufacturer.ManufacturerName from the Manufacturer entity for the appliance.
8. Display the Appliance.ModelName from the Appliance entity if Appliance.ModelName is not NULL else display an empty string

ii. Display a button to add another appliance, and a ***Next*** button.

iii. If the ***Add another appliance*** button is clicked,

1. Display the **Add Appliance Form**.

iv. Else if the ***Next*** button is clicked,

1. Go to the Submission Complete page.
2. If the ***Return to main menu*** button is clicked.
   a. Go to the **Main Menu** page.

2. If the user closes the Hemkraft website or the Hemkraft website unexpectedly crashes before an appliance is added:

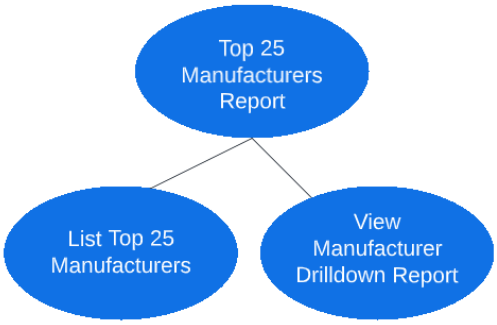
a. Clear all partial data for the household where the '$email' == Household.Email from the Household and Phone_Number entities.

Reports

Top 25 Popular Manufacturers

Task Decomposition

Lock Types: Read-only locks needed on Manufacturer and Appliance entities.
Number of Locks: 2.
Enabling Conditions: Enabled when the user selects ***Top 25 Manufacturers*** on the **View Reports** form.
Frequency: Medium – several users may view this report each day.
Consistency (ACID): is not critical, even if another user is entering appliances for their household while the user is viewing this report.
Subtasks: Mother task is needed coordinate subtasks. Task decomposition is necessary. Order of the task execution is critical; **List Top 25 Manufacturers** task must be executed before the **View Manufacturer Drilldown Report** task.



Abstract Code

1. User selects ***Top 25 Popular Manufacturers*** on the **View Reports** form.
2. Run the **List Top 25 Manufacturers** task
    a. For each Manufacturer.ManufacturerName from the Manufacturer entity
        i. Count the number of appliances from the Appliance entity whose associated manufacturer has the same manufacturer name as Manufacturer.ManufacturerName
        ii. Link is provided for drilldown report
    b. Sort the list in descending order of appliance count
    c. If the length of the list is greater than 25
        i. Truncate the list to the first 25 in the sorted list.
        ii. Display the results
        iii. Jump to step 3
    d. Else if the length of the list is 0
        i. Display "There are no appliances to generate the Top 25 Manufacturers"

3. If the ***View Manufacturer Drilldown Report*** is clicked for a manufacturer name ('$manufacturer_name')
    a. Run **View Manufacturer Drilldown Report** task
        i. Find the manufacturer from the Manufacturer entity whose manufacturer name == '$manufacturer_name'
        ii. For each appliance type in the Refrigerator_Freezer, Cooker, Washer, Dryer, and/or TV entities
            1. Count the number of appliances from the *appliance type* associated with the '$manufacturer_name' as an integer
        iii. If the count for all appliance types is 0
            1. Display a message stating "There are no appliances associated with that manufacturer name"
        iv. Else
            1. Display the results

Manufacturer/Model Search

Task Decomposition

Lock Types: Read-only on Manufacturer and Appliance entities.
Number of Locks: 2.
Enabling Conditions: Enabled when the user clicks the ***Submit*** button on the **View Reports** form and has selected "Manufacturer/Model Search" as the report type and has specified a search term.
Frequency: Medium – several users may view this report each day.
Consistency (ACID): is not critical, even if another user is entering appliances for their household while the user is viewing this report.
Subtasks: Mother task is not needed. No decomposition needed.

Abstract Code

1. User selects ***Manufacturer/Model Search*** on the **View Reports** form.
2. While the *search term* ('$search_term') input is not filled
    a. User enters a *search term* ('$search_term') into the input field
    b. If the *search term* input field is filled
        i. Jump to step 3.
    c. Else
        i. Jump to step 2.
3. While the ***Submit*** button is not clicked, do nothing.
4. If the ***Submit*** button is clicked:
    a. Run the **Search** task, using the '$search_term'
        i. Find the set of all distinct Manufacturer.ManufacturerName from the Manufacturer entity where Manufacturer.ManufacturerName contains the '$search_term' and all Appliance.ModelName where the appliance's manufacturer == Manufacturer.ManufacturerName UNION the set of all distinct Appliance.ModelName from the Appliance entity where Appliance.ModelName contains the '$search_term' and the Manufacturer.ManufacturerName for the appliance
        ii. Sort the set by Manufacturer.ManufacturerName ascending and Appliance.ModelName ascending.
        iii. If the length of the set is greater than 0:
            1. Jump to step 4b.
        iv. Else

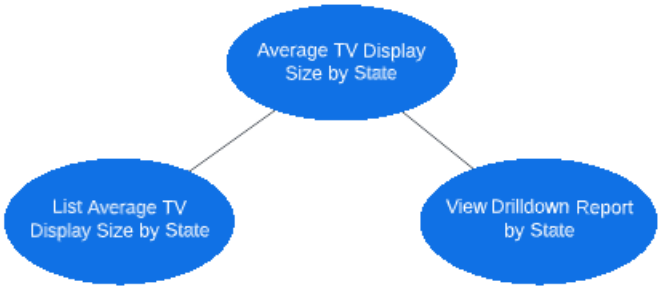
1. Display "There were no Manufacturers or Model Names that matched your search."

b. For each manufacturer name and model name in the results of the **Search** task:
   i. If the model name is NULL:
      1. Replace the model name in the result with an empty string
   ii. If the manufacturer name contains the '$search_term':
      1. Highlight the cell in green
   iii. Else if the model name contains the '$search_term':
      1. Highlight the cell in green
   iv. Display the results

Calculate Average TV Display Size by State

Task Decomposition

Lock Type: Read-only on Location and TV entities.
Number of Locks: 2.
Enabling Conditions: Enabled when the user selects ***Average TV Display Size by State*** on the **View Reports** form.
Frequency: Medium – several users may view this report each day.
Consistency (ACID): is not critical, even if another user is entering appliances for their household while the user is viewing this report.
Subtasks: Mother task is needed to coordinate tasks. Task decomposition is necessary. Order is critical; **List Average TV Display Size by State** task must be done first.



Abstract Code

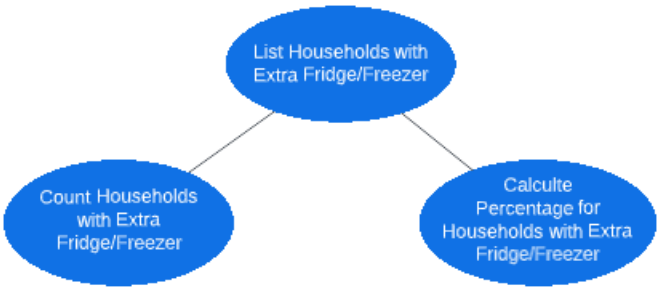
1. User selects the ***Average TV display size by State*** report on the **View Reports** form.
2. Run the **List Average TV Display Size by State** task:
   a. Find the Location.State and average TV.DisplaySize rounded up to the tenths decimal point from the Location, TV, Appliance, Household entities from the set of all households where the postal code for the household is in the Location.State and the TV is an appliance associated with the household and sort by Location.State ascending.
   b. A link is provided for the Drilldown Report by State for each state
3. If the ***Average TV Display Report by State Drilldown Report*** is clicked for a state ('$state'):
   a. Display the '$state'
   b. Display the TV.DisplayType, TV.MaxResolution, and average TV.DisplaySize as '$avg_display_size' rounded up to the tenths decimal point from Location, TV, Appliance, Household entities from the set of all households where the postal code for the household is in the Location. State as the Location.State == '$state' and the TV is an appliance associated with the household and group the results by

'$state', TV.DisplayType, TV.MaxResolution, and sort by '$avg_display_size' descending.

Extra Fridge/Freezer Report

Task Decomposition

Lock Type: Read-only on Location and Refrigerator_Freezer entities associate with each email in Household entity.
Number of Locks: 2.
Enabling Conditions: Enabled when the user selects ***List Households with Extra Fridge/Freezer*** on the **View Reports** form.
Frequency Medium – several users may view this report each day.
Consistency (ACID): is not critical, even if another user is entering appliances for their household while the user is viewing this report.
Subtasks: Mother task is needed to get to the two subtasks. Task decomposition is needed. Order for subtasks is not critical.



Abstract Code

1. User selects the ***List Households with Extra Fridge/Freezer*** report on the **View Reports** form.
2. Run the **Count Households with Extra Fridge/Freezer** task:
    a. Run **Count Households with Extra Fridge/Freeze** task:
        i. Display the count of the number of households from the Household entity where the count of Refrigerator_Freezer appliances that are associated with the household is greater than 1.
    b. Run **Calculate Percentage for Households with Extra Fridge/Freeze** task:
        i. Find the Location.State and the count of the number of households from the Household entity as ('$household_count'), where the household's postal code is in Location.State and count of Refrigerator_Freezer appliances that are associated with the household is greater than 1. Sort by '$household_count' descending.
        ii. For each state in the results

1. Count the number of households where Refrigerator_Freezer.RefrigeratorType == 'chest freezer' as '$chest_freezer_count', the number of households Refrigerator_Freezer.RefrigeratorType == 'upright freezer' ('$upright_freezer_count'), and the number of households Refrigerator_Freezer.RefrigeratorType != 'chest freezer' and Refrigerator_Freezer.RefrigeratorType != 'upright freezer' as ('$other_count')
2. Calculate the percentage of households with each type of fridge/freezer as ('$chestFreezerPercent'), ('$uprightFreezerPercent'), and ('$otherPercent'). Round all values to whole number.

iii. Display the first 10 records in the results

Laundry Center Report

Task Decomposition

Lock Type: Read-only on Washer entity, Dryer entity, Household entity, and Location entity
Number of Locks: 4.
Enabling Conditions Enabled when the user selects ***Laundry Center*** on the **View Reports** form.
Frequency Frequency Medium – several users may view this report each day.
Consistency (ACID): is not critical, even if another user is entering appliances for their household while the user is viewing this report.
Subtasks: Mother task is needed. The task is decomposed into subtasks. Laundry Center Report is the mother task.

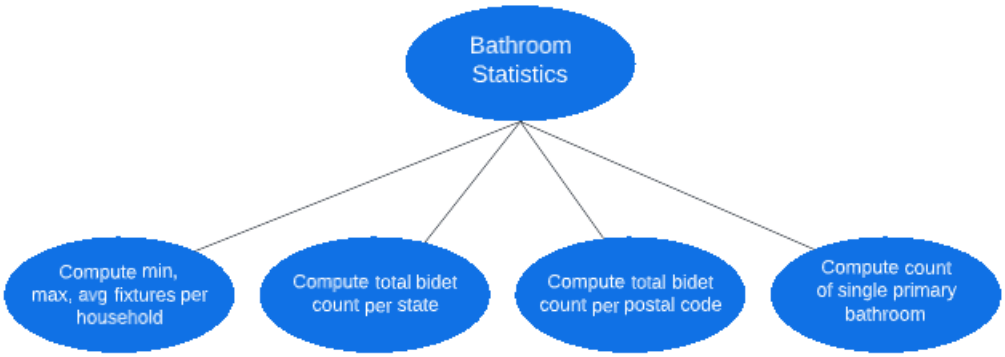
Abstract Code

1. User clicks on the button for ***Laundry Center*** after choosing ***View Reports***
2. Run the **Get Washer Type and Heat Source** task:
    a. Find the most common Washer.LoadingType and Dryer.HeatSource) from all households from the Household entity for each state in the Location entity where the household has a Washer surrogate, and the household has a Dryer surrogate and the household's postal code is in the Location.State.
    b. Sort by Location.State ascending.
3. Run the **Get Households with no dryer** task:
    a. Display the count of all households as ('$household_count') from the Household entity where there is a Washer surrogate associated with the household, but no Dryer surrogate associated with the household for each state in the Location entity where the household's postal code is in the Location.State.
    b. Sort by '$household_count' descending.

Bathroom Statistics

Task Decomposition

Lock Types: Read-only on Full, Half, Bathroom, Location, and Household entities.
Number of Locks: 5
Enabling Conditions: Enabled when the user selects ***Bathroom Statistics*** on the **View Reports** form.
Frequency Medium – several users may view this report each day.
Consistency (ACID): is not critical, even if another user is entering appliances for their household while the user is viewing this report.
Subtasks: Mother task is needed. The task is decomposed into sub tasks. Bathroom statistics is the mother task. Calculating min, max, avg count of bathrooms, fixtures per household task is done first and the following tasks, i.e., calculate total bidet count per state, calculate total bidet count per postal code, calculate single primary bathroom household can be parallelized.



Abstract Code

1. User clicks on ***Bathroom Statistics*** button.
2. Run the **Compute min, max, average statistics per household** task:
    a. List the minimum, maximum, average count of all bathrooms as min, max, avg on total count of Bathroom.BathroomID per household as '$total_bathroom_count'. Round the average count of bathrooms to tenths decimal point.
    b. List the minimum, maximum, average count of full bathrooms as min, max, avg on total count of Bathroom.BathroomID if the bathroom id is in Full entity per household as '$total_full_bathroom_count'. Round the average count of full bathrooms to tenths decimal point.
    c. List the minimum, maximum, average count of half bathrooms as min, max, avg on total count of Bathroom.BathroomID if the bathroom id is in Half entity per household as '$total_half_bathroom_count'. Round the average count of half bathrooms to tenths decimal point.

   d. List the minimum, maximum, average count of bidets as min, max, avg on total count of Bathroom.BidetCount per household '$total_bidet_count'. Round the average count of bidets to tenths decimal point.
   e. List the minimum, maximum, average count of sinks as min, max, avg on total count of Bathroom.SinkCount per household as '$total_sink_count'. Round the average count of sinks to tenths decimal point.
   f. List the minimum, maximum, average count of commodes as min, max, avg on total count of Bathroom.CommodeCount per household as '$total_commode_count'. Round the average count of commodes to tenths decimal point.
   g. List the minimum, maximum, average count of bathtubs as min, max, avg on total count of Bathroom.Full.TubCount per household as '$total_bathtub_count'. Round the average count of bathtubs to tenths decimal point.
   h. List the minimum, maximum, average count of all showers as min, max, avg on total count of Bathroom.Full.ShowerCount per household as '$total_shower_count'. Round the average count of showers to tenths decimal point.
   i. List the minimum, maximum, average count of tub/showers as min, max, avg on total count of Bathroom.Full.TubShowerCount per household as '$total_tubshower_count'. Round the average count of tub/showers to tenths decimal point.
3. Run **Compute the total bidet count per state** task:
   a. For each distinct Location.State in the Location entity, count the total Bathroom.BidetCount as 'total_bidet_count_per_state' from all bathrooms from all households from the Household entity where the household's postal code is in the Location.State
   b. Group by Location.State.
   c. Sort the results based on 'total_bidet_count_per_state' in descending order.
   d. Display the Location.State and '$total_bidet_count_per_state' of the first record.
4. Run **Compute the total bidet count per postal Code** task:
   a. For a household, find the corresponding postal code, '$postal_code' and total count of Bathroom.BidetCount '$total_bidet_count' computed in **Compute min, max, average statistics per household.**
   b. Group by '$postal_code' and find the total bidet count per postal code 'total_bidet_count_per_postal_code'.
   c. Sort the results based on 'total_bidet_count_per_postal_code' in descending order.
   d. Display the '$postal_code' and '$total_bidet_count' of the first record.
5. Run **Compute the count of Single primary bathroom** task:

   a. For each household, find the total count of Bathroom.BathroomID per household '$total_bathroom_count' computed in **Compute min, max, average statistics per household** where '$total_bathroom_count' is equal to 1 and Full.IsPrimary is true and Bathroom.BathroomID is in Full entity.
6. Display the total number of records retrieved.

Household Averages by Radius

Task Decomposition

Lock Types: 1 read lock on Household entity, 1 read lock on Location entity, 1 read lock on Household entity, 1 read lock on Bathroom entity, 1 read lock on Cooktop entity, 1 read lock on Oven entity, 1 read lock on Dryer entity.
Number of Locks: 7
Enabling Conditions: Enabled when the user clicks the ***Household Averages by Radius*** button on the **View Reports** form.
Frequency: Low
Consistency (ACID): Order is critical. The houses in a postal code must are found first and then the average bathroom count, the average bedroom count, the average occupant count, the ratio of commodes to occupants, the average number of appliances, and the most common heat source are determined.
Subtasks: Mother task is not needed.

Abstract Code

1. Display **Household Averages by Radius** form.
2. While ((*postal code* ('$postal_code') is not inputted) or (*postal code* ('$postal_code') is invalid) or (*search radius* ('$search_radius') is not inputted)):
   a. *postal code* ('$postal_code') is inputted
   b. *search radius* ('$search_radius') is inputted
   c. If ***Submit*** button is clicked:
      i. If the '$postal_code' is not found in the Location entity
         1. Display an error message indicating that the *postal code* input was invalid and empty the input fields.
         2. Jump to step 2.
      ii. Else
         1. Jump to step 3.
3. If '$search_radius' == 0:
   a. Run the **Household Averages by Radius** task:
   b. Find all households from the Household entity where the household is lives in the '$postal_code'

c. Round the average count of all bathrooms on total count of bathrooms per household (‘$average_bathroom_count’) to the nearest integer.
d. Round the average count of all bedrooms on total count of bedrooms per household in (‘$average_bedroom_count’) to the nearest integer. Display ‘$average_bedroom_count’.
e. Round the average count of all occupants per household in (‘$average_occupant_count’) to the nearest hundredth. Display ‘$average_occupant_count’.
f. Round the division of the sum of all Household.OccupantCount and the total count of Bathroom.CommodeCount for each household where the bathroom is associated with the household and the household is in the ‘$postal_code’ to determine (‘$ratio_commodes_to_occupants’). Display the ratio as “1: ‘$ratio_commodes_to_occupants’.”
g. Divide the count of all Appliance.ApplianceID (‘$total_number_of_appliances’) by the count of unique Household.Email (‘$number_of_households’) to determine ‘$average_number_of_appliances’. Round ‘$average_number_of_appliances’ to the nearest tenths decimal point. Display ‘$average_number_of_appliances’.
h. Find all ‘$heat_source’ on Dryer, Cooktop, and Oven entities along associated with Appliance.ApplianceID where the appliance is associated with the household and the household lives in the ‘$postal_code’
    i. Display the most common ‘$heat_source’.

4. Else:
    a. For each ‘$postal_code’ in the set of all unique Location.PostalCode from the Location entity and their Location.Longitude and Location.Latitude
        i. Calculate the ‘$haversine_distance’ to determine if the ‘$postal_code’ is in the specified ‘$search_radius’
        ii. If the ‘$haversine_distance’ is less than or equal to the search radius:
            1. Run the **Household Averages by Radius** task:
            2. Find all households from the Household entity where the household is lives in the ‘$postal_code’
            3. Round the average count of all bathrooms on total count of bathrooms per household (‘$average_bathroom_count’) to the nearest integer.

4. Round the average count of all bedrooms on total count of bedrooms per household in ('$average_bedroom_count') to the nearest integer. Display '$average_bedroom_count'.
5. Round the average count of all occupants per household in ('$average_occupant_count') to the nearest hundredth. Display '$average_occupant_count'.
6. Round the division of the sum of all Household.OccupantCount and the total count of Bathroom.CommodeCount for each household where the bathroom is associated with the household and the household is in the '$postal_code' to determine ('$ratio_commodes_to_occupants'). Display the ratio as "1: '$ratio_commodes_to_occupants'."
7. Divide the count of all Appliance.ApplianceID ('$total_number_of_appliances') by the count of unique Household.Email ('$number_of_households') to determine '$average_number_of_appliances'. Round '$average_number_of_appliances' to the nearest tenths decimal point. Display '$average_number_of_appliances'.
8. Find all '$heat_source' on Dryer, Cooktop, and Oven entities along associated with Appliance.ApplianceID where the appliance is associated with the household and the household lives in the '$postal_code'
    a. Display the most common '$heat_source'.