

Table of Contents

[Table of Contents](#)

[SQL Data Types](#)

[Table: Person](#)

[Table: User](#)

[Table: Client](#)

[Table: ClientLogEntry](#)

[Table: Site](#)

[Table: Item](#)

[Table: Supply](#)

[Table: FoodItem](#)

[Table: Request](#)

[Table: Service](#)

[Table: FoodBankService](#)

[Table: NonFoodBankService](#)

[Table: HoursOfOperation](#)

[Table: ConditionsForUse](#)

[Table: Shelter](#)

[Table: FoodPantry](#)

[Table: SoupKitchen](#)

[Constraints](#)

[Task Decomposition with Abstract Code](#)

[View Room/Bunk Availability](#)

[Task Decomp](#)

[Abstract Code](#)

[View Meals Remaining](#)

[Task Decomp](#)

[Abstract Code](#)

[Authenticate User Info](#)

[Task Decomp](#)

[Abstract Code](#)

[Main Menu](#)

[Task Decomp](#)

[Abstract Code](#)

[\(1.1\) Add/modify /delete service details](#)

[Task Decomp](#)

[Abstract Code:](#)

[\(1.2\) Search /Update availability of rooms,bunks](#)

[Task Decomp](#)

[Abstract Code:](#)

[\(1.3\) View/update available seats](#)

[Task Decomp](#)

[Abstract Code](#)

[\(2.1\) Add/update Item details](#)

[Task Decomp](#)

[Abstract Code](#)

[\(3.1\) Add/View/Modify client details](#)

[Task Decomp](#)

[Abstract Code:](#)

[\(3.2\) Search for client](#)

[Task Decomp](#)

[Abstract code](#)

[\(3.3.1\) Add/modify/delete client from waitlist](#)

[Task Decomp](#)

[Abstract code :](#)

[\(3.3.2\) View waitlist](#)

[Task Decomp](#)

[Abstract Code](#)

[\(4.1\) Search for request by type/storage/quantity](#)

[Task Decomp](#)

[Abstract Code](#)

[\(4.2\) Approve/edit/delete Requests from foodbank](#)

[Task Decomp](#)

[Abstract code](#)

[\(4.3\) View item request status](#)

[Task Decomp](#)

[Abstract code](#)

SQL Data Types

Table: Person

Column Name	Data Type	Constraint
ID	int	NOT NULL, UNIQUE
FirstName	varchar(80)	NOT NULL
MiddleName	varchar(80)	
LastName	varchar(80)	NOT NULL

Table: User

Column Name	Data Type	Constraint
UserName	varchar(40)	NOT NULL, UNIQUE
Password	varchar(80)	NOT NULL
EmailAddress	varchar(250)	NOT NULL

Table: Client

Column Name	Data Type	Constraint
Description	varchar(200)	NOT NULL, UNIQUE
PhoneNumber	varchar(80)	NOT NULL

Table: ClientLogEntry

Column Name	Data Type	Constraint
-------------	-----------	------------

Site	varchar(80)	
Description	varchar(255)	NOT NULL
DateCreated	datetime	NOT NULL

Table: Site

Column Name	Data Type	Constraint
ShortName	varchar(80)	NOT NULL, UNIQUE
StreetAddress	varchar(80)	NOT NULL
City	varchar(80)	NOT NULL
State	varchar(80)	NOT NULL
ZipCode	varchar(80)	NOT NULL
PhoneNumber	varchar(80)	NOT NULL

Table: Item

Column Name	Data Type	Constraint
Name	varchar(80)	NOT NULL, UNIQUE
Units	int(16) unsigned	NOT NULL
ExpDate	datetime	NOT NULL
StorageType	varchar(80)	{ Dry Good, Refrigerated, Frozen }

Table: Supply

Column Name	Data Type	Constraint
Type	varchar(80)	{ Shelter, Clothing, Personal Hygiene, Other }

Table: FoodItem

Column Name	Data Type	Constraint
Type	varchar(80)	{ Vegetables, Nuts/Grain/beans, Meat/seafood, Dairy/eggs, Sauce/condiment, Juice/drink }

Table: Request

Column Name	Data Type	Constraint
User	varchar(80)	NOT NULL
QuantityRequested	int(16) unsigned	NOT NULL
QuantityFulfilled	int(16) unsigned	
Status	varchar(80)	{ Pending, Closed }

Table: Service

Column Name	Data Type	Constraint

Table: FoodBankService

Column Name	Data Type	Constraint

Table: NonFoodBankService

Column Name	Data Type	Constraint
FacilityName	varchar(80)	NOT NULL
AddressInfo	varchar(80)	
PhoneNumber	varchar(80)	

Table: HoursOfOperation

Column Name	Data Type	Constraint
Description	varchar(255)	NOT NULL

Table: ConditionsForUse

Column Name	Data Type	Constraint
Description	varchar(255)	NOT NULL

Table: Shelter

Column Name	Data Type	Constraint
MaleBunksAvailable	int(16) unsigned	NOT NULL
FemaleBunksAvailable	int(16) unsigned	NOT NULL
MixedBunksAvailable	int(16) unsigned	NOT NULL
RoomsAvailable	int(16) unsigned	NOT NULL

Table: FoodPantry

Column Name	Data Type	Constraint
-------------	-----------	------------

--	--	--

Table: SoupKitchen

Column Name	Data Type	Constraint
TotalSeatingCapacity	int(16) unsigned	NOT NULL
AvailableSeats	int(16) unsigned	NOT NULL

Constraints

- Users are only allowed to make modifications to the data for the specific site they are associated with
- Users at any site with a food pantry, shelter or soup kitchen may use the system to request items from one or more food banks
- Users at a site with a food bank can approve or edit requests for their site's food bank
- If a site provides multiple services, all site users have full control of all data for all services
- A site may have no more than 4 services (one of each type)
- Users should not be able to remove the last service associated with their site
- A site must provide at least one service to be listed in the ASACS
- A user of site "A" can not view or modify the room wait list at site "B"
- Clients can not be deleted by users, but their name and description can be modified
- Users are never shown a full list of all clients. They must provide a search string that matches at least one client in the database, but it may NOT match more than 5 clients in the database
- If at any time the count of an item reaches zero, its record should be removed from inventory completely

Task Decomposition with Abstract Code

View Room/Bunk Availability



View Room/Bunk
Availability

Task Decomp

Lock Types: Read only on Shelter table

Enabling Conditions: None

Frequency: Around thousands of views per day

Schema elements involved: Two schemas


Consistency: not critical, order is not critical

Subtasks: Mother task is not needed, No decomposition needed.

Abstract Code

- User enters page which displays a list of all shelters that have available bunks or rooms.
- The list will include the shelter name, location, phone number, hours of operation, conditions, number of male bunks, female bunks, mixed bunks, and rooms available.
- If no bunks/rooms are available through the whole ASA system, a message "Sorry, all shelters are currently at maximum capacity."

View Meals Remaining



View Meals
Remaining

Task Decomp

Lock Types: Read only on FoodBank table

Enabling Conditions: None

Frequency: Around thousands of views per day

Schema elements involved: Two schemas

Consistency: not critical, order is not critical

Subtasks: Mother task is not needed, No decomposition needed.

Abstract Code

- User enters page which displays the number of meals remaining in inventory in all food banks in the ASA system.
- The report also shows what types of donations are needed to provide more meals.

Authenticate User Info



Task Decomp

Lock Types: Read only on User table

Enabling Conditions: None

Frequency: Around 100 logins per day

Schema elements involved: Single

Consistency: not critical, order is not critical

Subtasks: Mother task is not needed, No decomposition needed.

Abstract Code

- User enters *username* (\$Username), *password* (\$Password) input fields.
- If data validation is successful for both *username* and *password* input fields, then:
 - When **Enter** button is clicked:
 - If User record is found but User.password != '\$Password':
 - Go back to Login form, with error message.
 - Else:
 - Store login information as session variable \$UserID.
 - Go to Main Menu form.
- Else Username and password input fields are invalid. Display Login form with an error message.

Main Menu

Task Decomp

Lock Types: Lookup User's Name and Site, all are Read-only

Enabling Conditions: Trigger by successful login

Frequency: User detail and Menu options have the same frequency

Schemas: Two schemas

Consistency: not critical, order is not critical

Subtasks: Mother task is not needed, No decomposition needed.

Abstract Code

- Show “**Site details**”, “**Items**”, “**Clients**”, “**Requests**”, “**Reports**”, and “**Logout**” as tabs
Upon :

1. **Tab: Site details:**

Clicking of **Site details** tab displays menu items representing forms as a dropdown menu:

- **Site details** form
- **Rooms/Bunks available** form
- **Soup kitchen seats** form

- 1.1. Clicking on the **Site details** form will take the user to the task **Add /modify/delete service details.**
- 1.2. Clicking on **Rooms/Bunks Availability** form will take the user to the task **Search /Update availability of rooms/bunks.**
- 1.3. Clicking on **Soup kitchen seats** form will take the to the task **View/update available seats.**

2. **Tab:Items:**

Clicking of **Items** tab displays one menu items representing a form **Item details entry.**

- 2.1. Clicking on **Item details entry** will take the user to the task **Add/update Item details.**

3. **Tab: Clients:**

Clicking of **Clients** tab displays menu items representing forms as as a dropdown menu:

- **Client Check-in** form
 - **Client Search** form
 - **Client Waitlist** form
- 3.1. Clicking on **Client Check-in** form will take the user to the task **Add/View /modify client details.**
 - 3.2. Clicking on **Client Search** form will take the user to the task **Search for client.**
 - 3.3. Clicking on **Client waitlist** form will take the user to the tasks (3.3.1) **Add/modify/delete client** from waitlist and (3.3.2) **View waitlist.**

4. **Tab : Requests:**

Clicking of **Requests** tab displays menu items representing forms as a dropdown menu:

- **Item Request** form
- **Outstanding Requests** Report
- **Item Request status** report

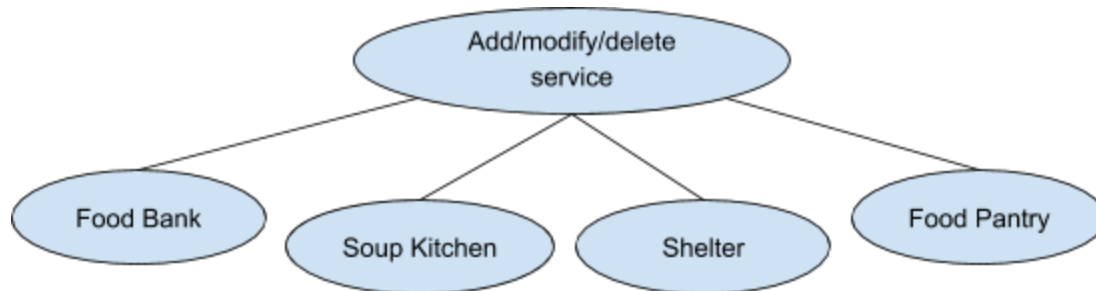
- 4.1. Clicking on **Item Request** will take the user to the task **Search for request by type/storage/quantity.**
- 4.2. Clicking on **Outstanding Requests** will take the user to the task **Approve/edit/delete requests from foodbank.**

4.3. Clicking on **Item Request status** will take the user to the task **View item request status**.

5. **Tab:Logout**

Clicking of **Logout** tab would would invalidate the login session and take the user back to the **Login** form.

(1.1) Add/modify /delete service details



Task Decomp

Lock Types: Read/write site Service information

Enabling Conditions: On click of **Site details form** button within sub-menu

Frequency: Medium

Schemas: One schema

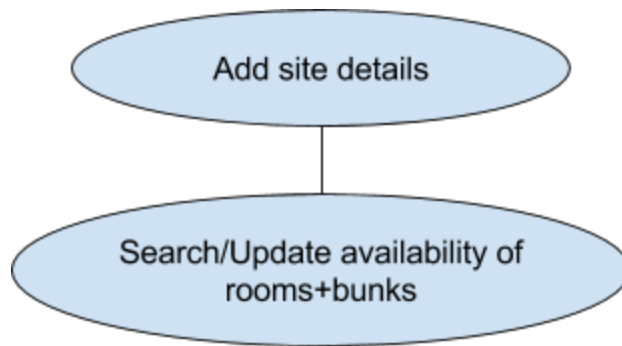
Consistency: Non-critical, Each site may provide multiple services.

Subtasks: Mother Task is needed, Only valid users can view the site information.

Abstract Code:

- The add/modify /delete site details task operates from the site details form.
- The user can add site details information like sitename, address(street address,city ,state,zipcode, phonenumber)
- The site details form will have an option in the form of a drop-down list to select the type of service.(Foodbank,Shelter,Soupkitchen , FoodPantry)
- If the service selected is FoodPantry, Shelter or Soupkitchen ,then enter details like description string (which has the facility name, Address), hours of operation (Time (from,to),days of week), Conditions_foruse(Document name(prrof of id), Weather condition(temperature),Clientphysicalcondition(intoxicated(true/false))
- If soup kitchen is selected , enter/view the total count of seats .(this is a value set to the total seating capacity)
- If shelter is selected, it has additional fields to be entered. The total number of room /bunk types for that site is to be entered.

(1.2) Search /Update availability of rooms,bunks



Task Decomp

Lock Types: read/write Shelter info

Enabling Conditions: On click of Rooms/Bunks availability form

Frequency: medium

Schemas: one schema

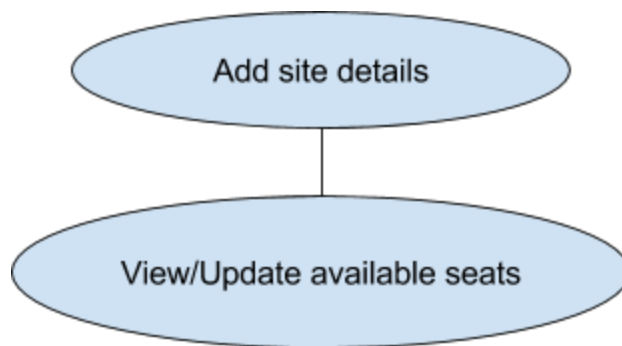
Consistency: non-critical, Search is performed on the entered sites to check for availability of rooms/bunks

Subtasks: mother task is needed. Previously if the rooms /bunks details are entered .only then it can be searched and retrieved.

Abstract Code:

- Search for the available rooms or bunks at a site by specifying the *sitename*, *location address*
- A user of a site can modify the room available number at only the site where the user has admin control using update task.

(1.3) View/update available seats



Task Decomp

Lock Types: read/write site details form

Enabling Conditions: On click of button “view available seats” in site details- soup kitchen form

Frequency: medium

Schemas: one schema

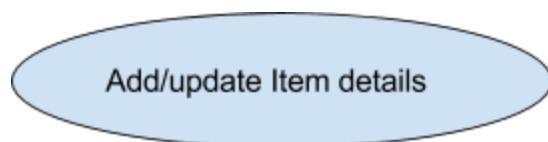
Consistency: non-critical, Search is performed on the sitename and location address entered to check for availability of seats

Subtasks: mother task is needed. Previously if the available seats for soup kitchen are entered only then it can be searched and retrieved.

Abstract Code

- Search for the available seats at a soup kitchen by specifying the sitename , location address
- A user of a site can lower the available seats at only the site where the user has admin control using update task.

(2.1) Add/update Item details



Task Decomp

Lock Types: Read and Update Item

Enabling Conditions: Click on Items menu item, select Item details entry form to open the form

Frequency: medium

Schemas: one schema

Consistency: non-critical,

Subtasks: mother task is not needed.

Abstract Code

- View all the existing items assigned to the current site
- Select an item by *name* to view the complete details for that item entry, loading the **Item Details** form
- Enter details for *foodbank*, *item name*, *number of units*, *expiration date*, *item_category*, *storage type*. Click on **Add new item details** to add item to database.
- To update a particular item details, first retrieve the item details using the *item name*. Make the changes and update it back to the database.

(3.1) Add/View/Modify client details



Task Decomp

Lock Types: read/write client form

Enabling Conditions: Click on Clients menu tab , select client check-in form

Frequency: medium

Schemas: one schema

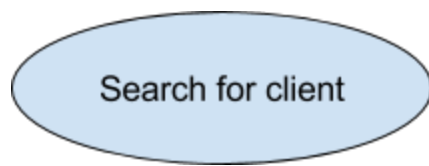
Consistency: non-critical,

Subtasks: mother task is not needed.

Abstract Code:

- Users can add a client by entering *client_fullname*, *client_IDnumber*, *contact_number*.
- Users can modify a client details by searching a client based on the *client_ID number* and *client_fullname*.
- A user can check a client to use a site service like shelter by entering the *sitename*, *dateof checkin*, *type of room or bunk selected* for client using *client_fullname*, *client_IDnumber*.

(3.2) Search for client



Task Decomp

Lock Types: client info, read only

Enabling Conditions: Click on Clients menu tab , select client search form

Frequency: medium

Schemas: one schema

Consistency: non-critical,

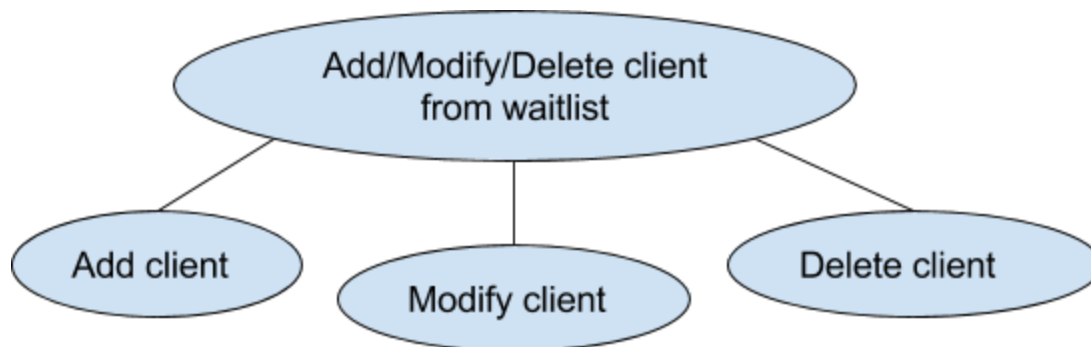
Subtasks: mother task is needed. Client details should have been entered.

Abstract code

- Search a client based on the *client_IDnumber* and *client_fullname*.
- If client details are found, display client details
Count the number of client's details displayed and store it in num.
If num < 5 , display a list of client_fullname and client_IDnumber

If client details are not found, display message "client not found"

(3.3.1) Add/modify/delete client from waitlist



Task Decomp

Lock Types: read/write lock

Enabling Conditions: Click on Clients menu tab , select client waitlist form menu option

Frequency: medium

Schemas: one schema

Consistency: non-critical,

Subtasks: mother task is not needed.

Abstract code :

- To add a client to waitlist, a user of a site shelter can add a client to the waitlist by entering the following details in the waitlist form: *Waitlist entry consists of client_fullname , date_placing request, client_ID, type of room_request* . A position number or ranking number will then be associated with the client when placed in waitlist.
- A client details can be modified to a waitlist. The client's position number can be modified by the user
- The user can also delete a client from the waitlist by first entering the *client_fullname, client_ID*. once the client's waitlist details are displayed . The client details can be removed from the waitlist on user confirmation.

(3.3.2) View waitlist



Task Decomp

Lock Types: read/write lock

Enabling Conditions: Click on Clients menu tab , select client waitlist form menu option

Frequency: medium

Schemas: one schema

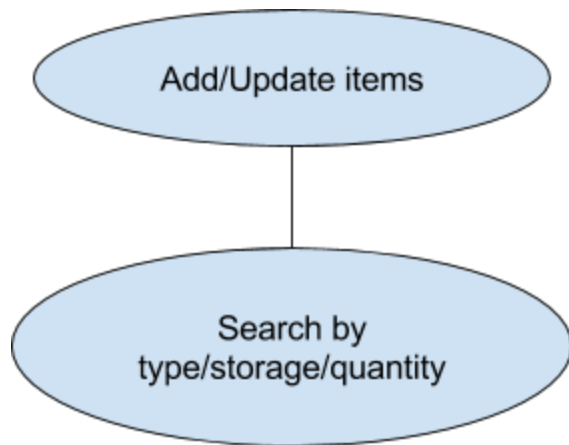
Consistency: non-critical,

Subtasks: mother task is needed. Clients waitlist should be present/added first to view the waitlist.

Abstract Code

- The user of a site shelter can enter the sitename , location address to view the room waitlist info for that site
- if waitlists are present at that site location address, display waitlist of clients in with respect to the client position number(or rank number).
- A particular client's waitlist number can then be obtained.

(4.1) Item request



Task Decomp

Lock Types: read only

Enabling Conditions: Click on Requests menu tab, select the menu item from dropdown list Request Form

Frequency: medium

Schemas: one schema

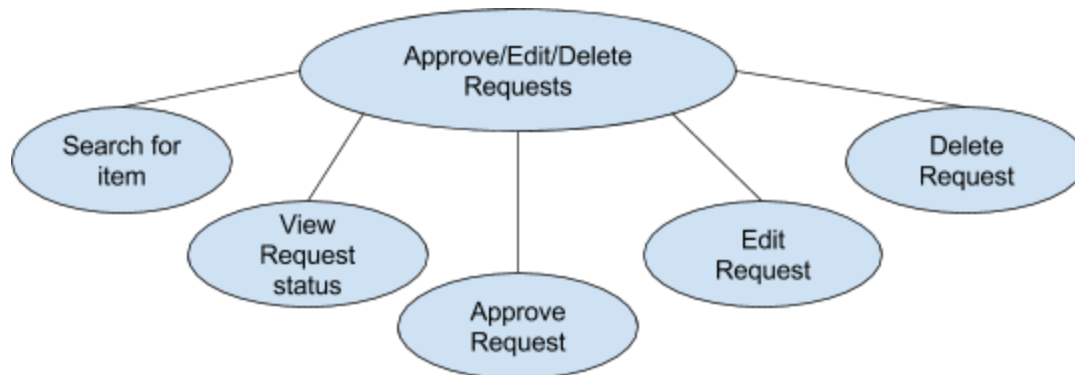
Consistency: non-critical, can be retrieved only if items are entered in the foodbank inventory

Subtasks: mother task is needed. Add/update item details task must be executed first before calling the search task.

Abstract Code

- To search an item by category type, storage, quantity
 - enter the item category type as food or supplies
 - enter the storage type as drygood/refrigerated / frozen
 - enter the quantity as number of boxes/ bags/cartons/units
- If the given search criteria finds a match, the item details are displayed
- else if no match is found, message is displayed a " item details not found"

(4.2) Outstanding Requests



Task Decomp

Lock Types: Read and Update Request

Enabling Conditions: Click on **Requests** menu, select from dropdown list "**Outstanding Requests**"

Frequency: medium

Schemas: one schema

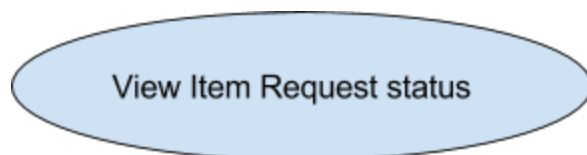
Consistency: non-critical, can be retrieved only if items are present in the foodbank inventory

Subtasks: mother task is needed. First the user search for a item, Views the item request status and then works on this task which can be subdivided into 5 subtasks.

Abstract code

- Users search for items and check for the pending request report
- If a request is accepted/approved, the number of items from the foodbank is reduced
- If a request is denied, items provided will be zero

(4.3) View item request status



Task Decomp

Lock Types: read only

Enabling Conditions: Click on **Requests** menu tab, select the menu item from dropdown list **View Item Request Status**

Frequency: medium

Schemas: one schema

Consistency: non-critical, can be retrieved using the Request form

Subtasks: none

Abstract code

- Enter the details on the **View Requests** form to view pending or closed requests.
- Pending requests represented as “P “and closed requests represented as “C”