Phase 1 Report
CS6400 – Spring 2017
Team 081

Table of Contents

Atlanta Sharing Alliance Coordination System Data Types

Table: Account

UserName varchar(25) NOT NULL
Password varchar(50) NOT NULL

Table: User
UserName varchar(25) NOT NULL
Email varchar(250) NOT NULL
FirstName varchar(50) NOT NULL
LastName varchar(50) NOT NULL
Role varchar(25) NOT NULL
SiteId int unsigned NOT NULL

Table: Site
Id int unsigned NOT NULL AUTO_INCREMENT
ShortName varchar(50) NOT NULL
StreetAddress varchar(250) NOT NULL
City varchar(100) NOT NULL
State varchar(2) NOT NULL
Zip varchar(10) NOT NULL
PhoneNumber (12) NOT NULL
Services ENUM NOT NULL {Shelter, SoupKitchen, FoodPantry, FoodBank}

Table: FoodBank
SiteId int NOT NULL
Category varchar(25) NOT NULL
ItemId int NOT NULL

Table: Requested Item
ItemId NOT NULL
RequesteeSiteId int NOT NULL
UserID int NOT NULL
Status ENUM NOT NULL {Pending, Closed}
NumRequested int NOT NULL
NumFilled in (16) NOT NULL
ReqDateTime TIMESTAMP NOT NULL

Table: FoodItem
ItemId unsigned NOT NULL AUTO_INCREMENT
ExpireDate DATE NOT NULL
NumberUnits int NOT NULL
Name varchar(250) NOT NULL
StorageType ENUM NOT NULL {Dry Good, Refrigerated, Frozen}
Food Category ENUM NOT NULL {Vegetables, nuts/grains/beans, Meat/seafood, Dairy/eggs, Sauce/Condiment/Seasoning, Juice/Drink}

Table: SupplyItem
ItemId NOT NULL AUTO_INCREMENT
ExpireDate DATE NOT NULL
NumberUnits int NOT NULL
Name varchar(250) NOT NULL
StorageType ENUM NOT NULL {Dry Good, Refrigerated, Frozen}
Supply Category ENUM NOT NULL {Personal hygiene, Clothing, Shelter, other}

Table: FoodPantry
SiteId int NOT NULL
Description varchar(500)
HoursOperation varchar (500) NOT NULL

ConditionUse varchar (500) NOT NULL

Table: SoupKitchen
SiteId int NOT NULL
Description varchar(500)
HoursOperation varchar (500) NOT NULL
ConditionUse varchar (500) NOT NULL
AvailableSeats int

Table: Shelter
SiteId int NOT NULL
Description varchar(500)
HoursOperation varchar (500) NOT NULL
ConditionUse varchar (500) NOT NULL
FamilyRoomCount int NOT NULL
MaleBunkCount int NOT NULL
FemaleBunkCount int NOT NULL
MixedBunkCount int NOT NULL
TotalBunkCount int NOT NULL

Table: WaitList
ClientId int NOT NULL
SiteId int NOT NULL
Position int NOT NULL

Table: Client
Identifier int NOT NULL AUTO_INCREMENT
ID/Description varchar(500) NOT NULL
Name varchar(100) NOT NULL
Phone varchar(12)

Table: LogEntry
ClientId int NOT NULL
SiteId int NOT NULL
Time/Date TIMESTAMP NOT NULL
Notes varchar (500)
DescOfService varchar(500)

Atlanta Sharing Alliance Coordination System Constraints

User:
- Users must be authenticated.
- Users can only make modifications to the data for the specific site with which they are associated.

Sites:
- A site must provide at least one service.

WaitList:
- The default ranking for waitlist is first come first served.

FoodBank:
- Items cannot have a count of less than 1; items whose counts reach zero are deleted.

- A meal is one unit each of a vegetable, nuts/grains/beans, and Meat/seafood or Dairy/eggs.

Clients:

- Users cannot see information for more than five clients at a time.

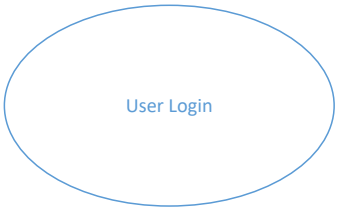
Requests:

- Users cannot request an item from the Food Bank associated with their site.
- Pending requests for an item that is deleted (generally because its NumberUnits has reached zero) are given the status closed with a NumFilled value of 0.

Task Decomposition with Abstract Code

Login/Authenticate User

Task Decomp



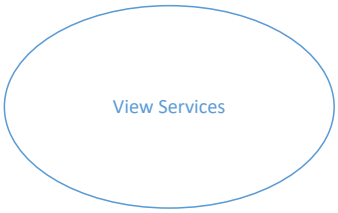
Lock Types: Read-only on Account table
Enabling Conditions: None
Schemas: Single
Consistency: Order is not critical
Subtasks: Mother Task is not needed. No decomposition needed
Frequency: Average user logons per day: 500

Abstract Code

- User enters *username* and *password* into form fields
- If data validation for *username* and *password(*$*password)* fields is successful:
    - When ***Logon*** button is pressed:
        - If Account record is not found by username or if hash of Account.password != $*password* then:
            - Go back to **Login** form with error message
        - Else
            - Store User information into session variable $user
            - Retrieve Site record for $user and store in $siteid
            - Go to **Manage Services** form

View Services

Task Decomp



Lock Types: 3 Read-only lookups for User, Site, and Service
Enabling Conditions: Successful login
Schemas: Single
Consistency: Order is not critical
Subtasks: Mother Task is not needed. No decomposition needed
Frequency: Medium

Abstract Code

- Show the list of available Services associated with the Site by $siteid
- Upon choice of:
    - *Food Bank* link, go to **Food Bank Request** form
    - *Soup Kitchen* link, go to **Modify Seat Capacity** form
    - *Shelter* link, go to **Modify Available Bunks** form

Add/Edit/Remove Service

Task Decomp

Lock Types: Inserts or Deletes to tables for a service
Enabling Conditions: Service selected on **Manage Services** Form
Schemas: 2 Schema constructs at a time
Consistency:
Subtasks: Mother Task is not needed. No decomposition needed
Frequency: Rare

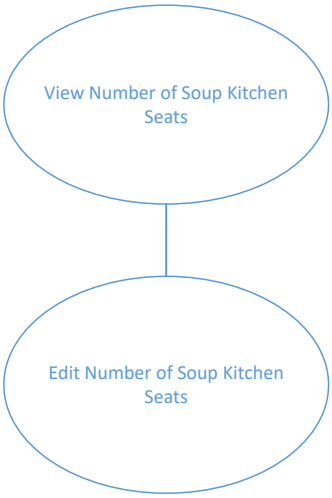
Abstract Code

- User selects a Service by Site table's ServiceCategory to be changed as $serviceCategory
- If the ***Add*** button is chosen:
    - If the Site already has an association to $serviceCategory then:
        - Go to **Manage Services** Form, with error
    - Else
        - If $serviceCategory == "FoodBank"
            - Insert record into FoodBank table for $siteid
        - If $serviceCategory == "FoodPantry"

        - Insert record into FoodPantry table for $siteid
    - If $serviceCategory == "Shelter"
        - Insert record into Shelter for $siteid
    - If $serviceCategory == "SoupKitchen"
        - Insert record into SoupKitchen for $siteid
- Else if ***Remove*** is chosen:
    - Existing Service records for $siteid are read into $serviceList
    - If the count of is $serviceList <= 1
        - Go to **Manage Services** Form, with error "Cannot remove last service"
    - Else
        - If $serviceCategory == "FoodBank"
            - Delete record from FoodBank table for $siteid
        - If $serviceCategory == "FoodPantry"
            - Delete record from FoodPantry table for $siteid
        - If $serviceCategory == "Shelter"
            - Delete record from Shelter for $siteid
        - If $serviceCategory == "SoupKitchen"
            - Delete record from SoupKitchen for $siteid

View/Edit Number of Soup Kitchen Seats

Task Decomp



Lock Types:
Enabling Conditions: User's site must have a soup kitchen
Schemas: Single
Consistency: View Number of seats, then edit
Subtasks: Viewing number of seats, Editing number of seats
Frequency: Rare

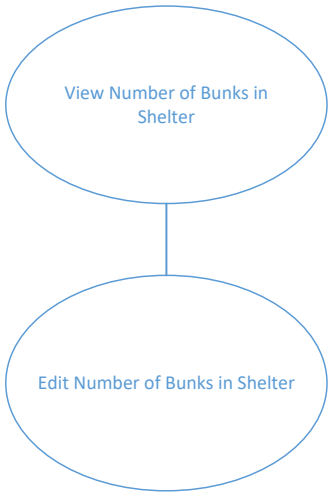
Abstract Code

- User selects the option to **view** total seats available in the soup kitchen.
    - Application shows number of seats available.
- User selects the option to **edit** the number of available seats in the soup kitchen

- Textbox displaying total number of seats is shown and enable for editing. ($availableSeats)
- If user hits ***Save*** button:
    - Value in textbox is checked to ensure validity (e.g. value is a number, not greater than total seat capacity of the soup kitchen) and saved to database
- Else If user hits ***Cancel*** button:
    - Value in textbox is discarded, no changes are made to database.

View/Edit Number of Bunks in Shelter

Task Decomp



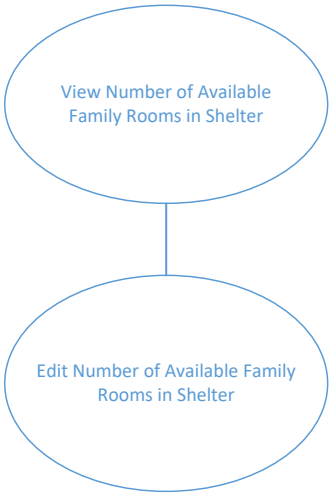
Lock Types: Read-only lookup on Shelter if viewing, Update lock if editing
Enabling Conditions: User's site must have a shelter
Schemas: Single
Consistency: View number of bunks, then edit
Subtasks: Viewing number of bunks, Editing number of bunks
Frequency: Often

Abstract Code

- User selects option to **view** number of bunks currently available in shelter
- Application queries Shelter table for *$maleBunkCount, $femaleBunkCount,* and *$mixedBunkCount,* derives *$totalBunkCount,* and displays form showing the total number of bunks available as well as a breakdown of male, female, and mixed-gender bunks available.
- If user selects option to **edit** the number of bunks in the shelter:
    - User is given option to edit either male/female/mixed bunks.
    - Textbox displaying number of selected bunk types is shown and is enabled for editing.
    - If user hits ***Save*** button:
        - Value in textbox is checked to ensure validity and value(s) are updated in the Shelter table.
    - Else If user hits ***Cancel*** button:
        - Value in textbox is discarded, no changes are made to database.

View/Edit Number of Family Rooms in Shelter

Task Decomp



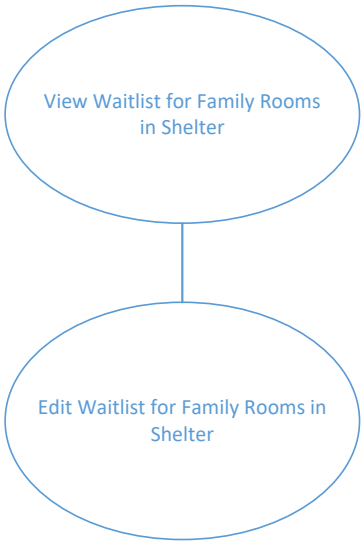
Lock Types: Read-only lookup on Shelter if viewing, Update lock if editing
Enabling Conditions: User's site must have a shelter with family rooms
Schemas: Single
Consistency:
Subtasks: Viewing number of family rooms available, editing number
Frequency: Often

Abstract Code

- User selects option to view number of family rooms currently available in shelter
- Application queries the Shelter table, returns the number of family rooms available (*$familyRoomCount*), and displays to user.
- If user selects option to **edit** the number of family rooms in the shelter:
  - Textbox displaying number of family rooms is enabled for editing.
  - If user hits ***Save*** button:
    - Value in textbox is checked to ensure validity and the value is updated in the Shelter table.
  - Else If user hits ***Cancel*** button:
    - Value in textbox is discarded, no changes are made to database.

View/Edit Waitlist for Family Rooms in Shelter

Task Decomp



Lock Types: Read-only lookup on Waitlist if viewing, Update lock if editing
Enabling Conditions: User's site must have a shelter with family rooms
Schemas: Single
Consistency:
Subtasks: Viewing waitlist, then edit
Frequency: Often

Abstract Code

- User selects the ***View Waitlist*** button for family rooms currently available in shelter
- Application queries the Waitlist table and displays records associated with the user's shelter. Records are sorted with highest $Position value on top.
- If user clicks the ***Add*** button:
  - Form for adding a client to the waitlist is displayed
  - The new record is inserted into the Waitlist table with the $ClientID and a $Position value, which is incremented from the current highest value.
- If user clicks the ***Update*** button:
  - All $Position values displayed in the waitlist will become editable.
  - User clicks on the row they wish to edit and may change the client's position in the waitlist.
  - If user hits the ***Save*** button:
    - The record for the selected client in the Waitlist table will be Updated.
    - All $Position values for clients with a position on the waitlist greater than the new position of the selected client will be incremented by one.
  - Else if the user hits the ***Cancel*** button:
    - Any values entered into the textboxes for any of the clients will be discarded and no changes will be made.
- If user clicks the ***Delete*** button:
  - A confirmation box will display to make sure that the user wants to delete the selected client.
  - If the user clicks ***Yes***, the selected client record will be deleted from the Waitlist table.

Grant/Deny Food Bank Request

Task Decomp

Lock Types: Table lock on RequestedItem and Item
Enabling Conditions: User's Site has Food Bank, requested item exists
Schemas: 2 Schema constructs needed
Consistency: RequestedItem.status, Item.NumberUnits may be updated. Other pending requests for item may be closed.
Subtasks: Mother Task is not needed. No decomposition needed
Frequency: Medium usage

Abstract Code

- Run the **Grant/Deny Food Bank Request** task for Site associated with $user
- Find the current RequestedItem records for User's FoodBank; Display list by Site.ShortName, Item.Name, and RequestedItem.NumRequested
- For each user selected record:

  If ***Grant*** button chosen and *quantity* field ($quantityToGrant) validated

  - update RequestedItem table record NumFilled = $quantityToGrant, Status = 'closed', and Item table NumberUnits = $availableUnits - $quantityToGrant for selected Item.ItemId.
  - If Item table NumberUnits = 0 for Item.ItemId
    - Find all RequestItem.ItemId = Item.ItemId
    - Update NumFilled = 0 and Status = 'closed'
- If ***Deny*** button chosen
  - update RequestedItem table NumFilled = 0 and Status = 'closed'

Make Food Bank Request

Task Decomp

Lock Types: None, insert into RequestedItem
Enabling Conditions: User is authenticated and request is not being made to FoodBank at user's own site
Schemas: Single
Consistency: Order is not critical
Subtasks: Mother Task is not needed. No decomposition needed
Frequency: Medium

Abstract Code

- Run the **Make Food Bank Request** task for Site associated with $user
- Find the current FoodBank record; Display list by Site.ShortName
- Find the available Item records for FoodBank record $foodbank; Display list by Item records Name and NumberUnits
- If ***Request*** button chosen
    - Insert into RequestedItem table NumRequested = $requestedUnits, Status = 'pending, $foodbank, $user
    - Goto **Food Bank Request** form

Retrieve Food Bank Item Data

Task Decomp

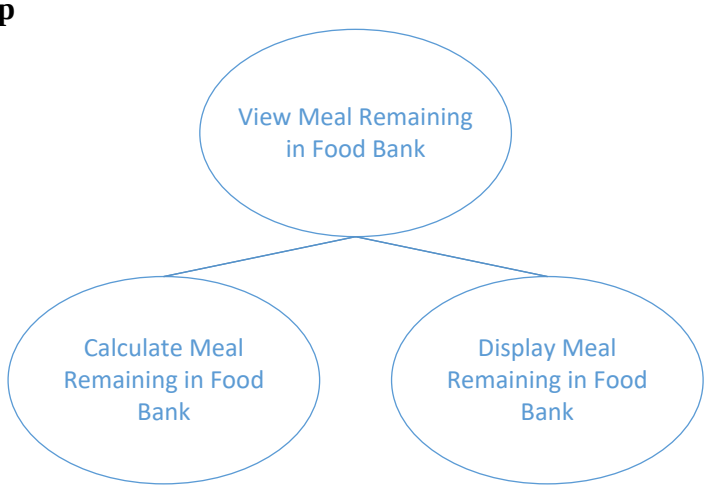
Lock Types: Read-only
Enabling Conditions: User is authenticated
Schemas: Single
Consistency: No required order
Subtasks: Mother Task is not needed. No decomposition needed
Frequency: Medium

Abstract Code

- Run **Retrieve Food Bank Item Data** task with $siteId
- Display Item record's Name, StorageType, NumberUnits, and ExpireDate

View Meal Remaining in Food Bank

Task Decomp



Lock Types: Read-only
Enabling Conditions: User is authenticated
Schemas: 2 Schema constructs
Consistency: not critical
Subtasks: Both tasks must be done sequentially. Mother task is required to coordinate subtasks.
Frequency: Medium

Abstract Code

- Run the **View Meals Remaining in Food Bank** task with $siteid
- Get the sum of each Item record's NumberUnits for each FoodItem record with FoodCategory of "Vegetable" as $vegCount, "nuts/grains/beans" as $nutsCounts, "Meat/seafood" as $meatCount, and "Dairy/eggs" as $dairyCount
- Calculate $totalMealCount as the minimum of [$vegCount, $nutsCounts,($meatCount + $dairyCount)]
- Display $totalMealCount

Add Inventory to Food Bank

Task Decomp

Lock Types: Insert into FoodItem or Supplyitem
Enabling Conditions: User is authenticated
Schemas: Single
Consistency:
Subtasks: Mother Task is not needed. No decomposition needed
Frequency: Frequent

Abstract Code

- The ***Add Donation*** button is pressed and the *category* ($category), *name* ($name), *quantity* ($quantity) and *expiration date* ($expireDt) fields are valid
- If the $category == "personal hygiene", "clothing", "shelter", or "other"
    - Insert a Supplyitem for $category, $name, $quantity, $expireDt
- Else if the $category == ""vegetables", "nuts/grains/beans", "meat/seafood", "dairy/eggs", "sauce/condiment/seasoning", "juice/drink"
    - Insert a Fooditem for $category, $name, $quantity, $expireDt

.

Add/Update Client

Task Decomposition

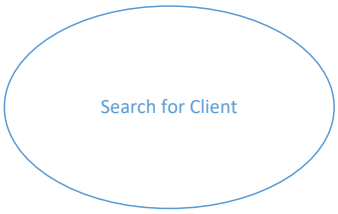
Lock Types: Insert lock on Client if adding, update lock if updating.
Enabling Conditions:
Schemas: Single
Consistency: Mother Task is not needed. No decomposition needed
Subtasks:
Frequency: Often

Abstract Code

- If user selects the ***Add Client*** button:
  - The **Add Client** Form will be displayed.
  - User will enter the Client's details: $Name, $ID_Description and $PhoneNum.
  - If User clicks ***Save***:
    - A new record with a unique $ClientID and the Client's details will be inserted into the Client table.
  - Else if User clicks ***Cancel***:
    - All values entered into the **Add Client** Form will be discarded and no changes will be made to Client table.

Search for Client

Task Decomposition



Lock Types: Read-only lookup on Client
Enabling Conditions:
Schemas: Single
Consistency: Mother Task is not needed. No decomposition needed
Subtasks:
Frequency: Often

Abstract Code

- If user selects the ***Search for Client*** button:
  - The **Search for Client** form will be displayed.

- The form contains empty textboxes for $ClientID, $ID_Description, $Name and $PhoneNum.
- If the user hits the ***Search*** button:
  - The Client table will be queried with the values entered into the textboxes on the form.
  - A table showing all records returned in the search will be displayed to the user.
- Else if the user hits the ***Cancel*** button:
  - The **Search for Client** form will be closed and no search will be performed.