Phase 1 Report

CS 6400 - Spring 2017

Team 067

SQL Data Types

Table: User

user_id int(16) unsigned NOT NULL AUTO_INCREMENT
site_id unsigned NOT NULL
email varchar(250) NOT NULL
username varchar(50) NOT NULL
password varchar(50) NOT NULL
firstname varchar(50) NOT NULL
lastname varchar(50) NOT NULL

Table: Site

site_id int(16) unsigned NOT NULL AUTO_INCREMENT
short_name varchar(250) NOT NULL
phone varchar(20) NOT NULL
street_address(250) NOT NULL
city varchar(50) NOT NULL
state varchar(20) NOT NULL
zipcode varchar(20) NOT NULL

Table: Soup Kitchen

service_id int(16) unsigned NOT NULL AUTO_INCREMENT
site_id int(16) unsigned NOT NULL
facility_name varchar(250) NOT NULL
hours_of_operation varchar(50) NOT NULL
conditions_for_use varchar(50) NOT NULL
seats_capacity int(16) unsigned NOT NULL
total_seats_available int(16) unsigned NOT NULL

Table: Food Pantry

service_id int(16) unsigned NOT NULL AUTO_INCREMENT
site_id int(16) unsigned NOT NULL
facility_name varchar(250) NOT NULL
hours_of_operation varchar(50) NOT NULL
conditions_for_use varchar(50) NOT NULL

Table: Shelter

service_id int(16) unsigned NOT NULL AUTO_INCREMENT
site_id int(16) unsigned NOT NULL
facility_name varchar(250) NOT NULL
hours_of_operation varchar(50) NOT NULL
conditions_for_use varchar(50) NOT NULL

Table: Bunk

service_id int(16) unsigned NOT NULL
bunk_id int(16) unsigned NOT NULL
bunk_type varchar(50) NOT NULL
available_status varchar(50) NOT NULL

Table: Family Room

service_id int(16) unsigned NOT NULL
room_id int(16) unsigned NOT NULL
available_status varchar(50) NOT NULL

Table: Waitlist

service_id int(16) unsigned NOT NULL
client_id int(16) unsigned NOT NULL
order_index int(16) unsigned NOT NULL
timestamp date NOT NULL

Table: Foodbank

service_id int(16) unsigned NOT NULL AUTO_INCREMENT
site_id int(16) unsigned NOT NULL
facility_name varchar(250) NOT NULL

Table: Item

item_id int(16) unsigned NOT NULL AUTO_INCREMENT
service_id int(16) unsigned NOT NULL
item_name varchar(250) NOT NULL
storage_type varchar(50) NOT NULL
expiration_date date

category varchar(50) NOT NULL
sub_category(50) NOT NULL

Table: Request

request_id int(16) unsigned AUTO_INCREMENT
user_id int(16) unsigned NOT NULL
service_id(16) unsigned NOT NULL
item_name varchar(250) NOT NULL
item_quantity int(16) unsigned NOT NULL
Item_provided int(16) unsigned NOT NULL
status varchar(10) NOT NULL
timestamp date NOT NULL

Table: Client

client_id int(16) unsigned NOT NULL AUTO_INCREMENT
firstname varchar(50) NOT NULL
lastname varchar(50) NOT NULL
id_number varchar(20) NOT NULL
is_head_of_household boolean NOT NULL
description varchar(250) NOT NULL
phone varchar(20)

Table: Field Modified

client_id int(16) unsigned NOT NULL
timestamp date NOT NULL
description varchar(250) NOT NULL

Table: Service Usage

client_id int(16) unsigned NOT NULL
service_id int(16) unsigned NOT NULL
timestamp date NOT NULL
description varchar(250) NOT NULL
service_name varchar(50) NOT NULL
extra_notes varchar(250) NOT NULL

Constraints

User

- Only able to modify data to his/her associated site
- If associated with food pantry/shelter/soup kitchen, he/she can request item from foodbank
- If associated with food bank, he/she can accept/reduce/deny requests

Sites

- Short name will be unique.
- System must allow user to create, delete and edit any services for their site (only).
- A site will have only one service of a particular type.
- Users should not be able to remove the last service associated with their site.

Service

- No constraint

Shelter

- A waitlist entry must consist of a link to a specific client entry in the ASACS who serves as the “head of household”.
- A user of site “A” can not view or modify the room wait list at site “B”.

Food Bank

- All users of the ASACS system can put in a request for up to the maximum amount available of items.

Item

- A “meal” as one unit each of a Vegetable, nuts/grains/beans, and Meat/seafood OR Dairy/eggs.

Request

- It will be for a specific item in the inventory, with a specific number requested.

Client

- Clients can not be deleted by users.

Available Bunks/Rooms Report

- No constraint

Meals Remaining Report

- No constraint

Client Search / Report

- Only provided to logged in users at any site.
- User are never shown a full list of all clients.
- User 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.

Wait List Report

- Only provided to logged in users who is associated with a site that has a Shelter.

Item Search Report

- Only provided to logged in users.
- A user should NOT be able to request an item from the Food Bank associated with their site, even though they can see items from their site.
- If the count of an item reaches zero, its record should be removed from inventory.

Outstanding Request Report

- Only provided to logged in users who is associated with a site that has a Food Bank.
- If the total number of items requested is greater than the total number available, the report should indicate this shortfall in inventory to the user.

Request Status Report

- Only provided to logged in users.

Task Decomposition with Abstract Code

Login

Task Decomp

Lock Types: Read-only on User table
Enabling Conditions: None
Frequency: Very High
Schemas: Single
Consistency: Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

- Display buttons:
    - **View Web Reports**
    - **Login**
- Upon:
    - Click **Web Reports** button
        - Jump to the **View Web Reports**
- User enters *user name* ($UserName), *password* ($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 $UserName.
            - Go to Main Menu form.
- Else *username* and *password* input fields are invalid, display **Login** form, error message. (See Reference 1 pg.22)

Web Reports

Task Decomp

Lock Types: Read only

Enabling Conditions: None
Frequency: Very High
Schemas: None
Consistency: Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

- Display buttons:
    - **View Available Bunks/Rooms Report**
    - **View Meals Remaining Report**
- Upon:
    - Click **View Available Bunks/Rooms Report** button
        - Jump to the **View Available Bunks/Rooms Task**
    - Click **View Meals Remaining Report** button
        - Jump to the **View Meals Remaining Task**

View Available Bunks/Rooms Report

Task Decomp

Lock Types: Read-only on shelter table
Enabling Conditions: None
Frequency: Very High
Schemas: Single
Consistency: Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

- Display all Shelters with available bunks or rooms.
- If a Shelter in the ASA System has an available bunks or rooms display:
    - Name
    - Site location
    - Phone number
    - Hours of operation and conditions
    - Number of male/female/mixed bunks and rooms available
- Else

    - Display “Sorry all shelters are currently at maximum capacity” message.
- Click **Back** button
    - Brings unauthenticated clients back to **Web Reports** form.

View Meals Remaining Report

Task Decomp

Lock Types: Read-only on the foodbank table
Enabling Conditions: None
Frequency: Very High
Schemas: Single
Consistency: Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

- Display all meals remaining in inventory in all Food Banks in ASA system.
- Display:
    - Type of donations are most needed to provide more meals (Either vegetables, nuts/grains/beans, or one of meat/seafood or dairy or eggs)
- Click **Back** button
    - Brings unauthenticated clients back to **Web Reports** form.

Main Menu

Task Decomp

Lock Types: Lookup Users ServiceID’s associated to their SiteID, all are Read-only.
Enabling Conditions: Logged in Users
Frequency: High
Schemas: Multiple
Consistency: Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

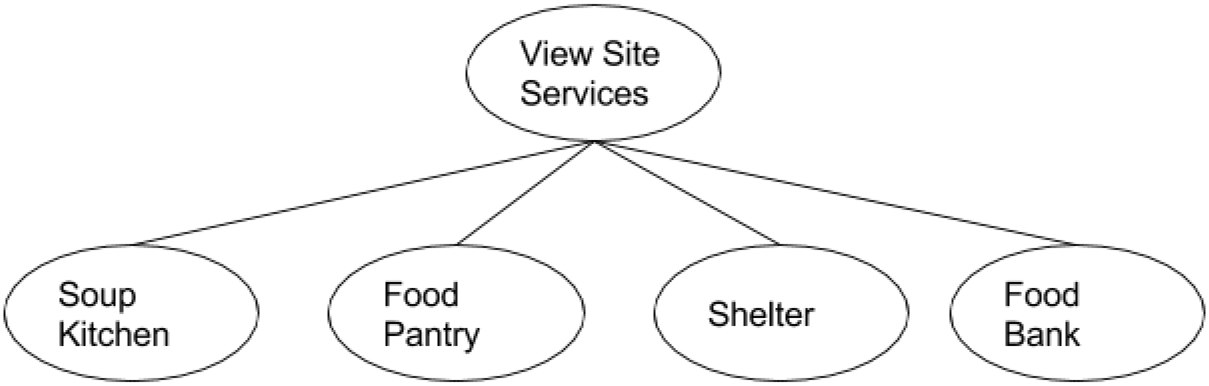
Abstract Code

- Look up $UserID and show the following buttons contingent on what $ServicesID are connected to the users $SiteID.
- Display buttons:
    - **View Site Services**

    - **Add Site Services**
    - **View Client Search/Report**
    - **View Item Search/Report**
    - **View Waitlist Report** (if Users $SiteID is associated with a Shelter)
    - **View Outstanding Requests Report** (if Users $SiteID is associated with a Food Bank)
    - **View Requests Status Report**
    - **Log Off**
- Click on **View Site Services** button
    - Jump to **View Site Services** Task
- Click on **Add Site Services** button
    - Jump to **Add Site Services** Task
- Click on **View Client Search/Report** button
    - Jump to **View Client Search/Report** Task
- Click on **View Item Search/Report** button
    - Jump to **View Item Search/Report** Task
- Click on **View Waitlist Report** button
    - Jump to **View Waitlist Report** Task.
- Click on **View Outstanding Requests Report** button
    - Jump to **View Outstanding Requests Report** Task.
- Click on **View Requests Status Report** button
    - Jump to **View Requests Status Report** Task.
- Click **Log Off** button
    - Invalidate login session and go back to the **Login** form.

View Site Services

Task Decomp



Lock Types: Read-Only lookup of Site Services for a User.
Enabling Conditions: Both are enabled by a user’s login.

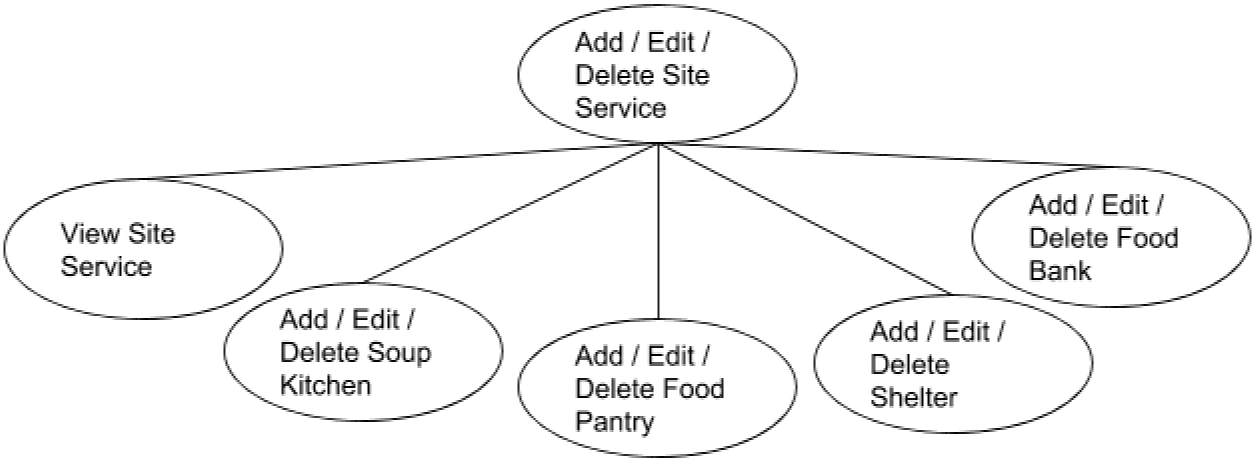
Frequency: Medium
Schemas: Multiple different schema constructs are needed
Consistency: Not Critical, order is not critical.
Subtasks: Lookup up multiple services for a site. Mother task is required to coordinate subtasks.

Abstract Code

- Get all the services which are associated with a $SiteID which are associated with $UserID.
- Lookups all Services associated with the current User's $SiteID.
- For each Service the User is associated with, display all the basic information for each Service.
- Click on a service
    - Jump to **Edit Site Service** Task
- Click **Back** button
    - Brings User back to **Main Menu** form.

Add/Edit/Delete Site Service

Task Decomp


Add / Edit / Delete Site Service
View Site Service
Add / Edit / Delete Soup Kitchen
Add / Edit / Delete Food Pantry
Add / Edit / Delete Shelter
Add / Edit / Delete Food Bank

Lock Types: Lookup, Update, Delete and Add Services for a Site.
Enabling Conditions: Enabled by a user's login.
Frequency: Low
Schemas: Multiple different schema constructs are needed
Consistency: Not Critical, order is not critical.
Subtasks: Lookup, Update, Delete and Add multiple services for a site. Mother task is required to coordinate subtasks.

Abstract Code

- View Site Services.
- Add, Edit, Delete the services for a site that the user is associated with.
- Run the View Site Service again.
- Click **Back** button
  - Brings User back to **Main Menu** form.

View Client Search / Report

Task Decomp

Lock Types: Read-only on Client table
Enabling Conditions: Logged in Users
Frequency: Medium
Schemas: Single
Consistency: Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

- Display input boxes
  - Client name
  - ID number
- Display **Search** and **Add** buttons
- Upon:
  - Click on **Search** button
    - Run **Search for Client Record** task
    - Display maximum 5 clients that matched the query. Each client row shows:
      - Client name
      - ID number
      - Description
    - Click on a client
      - Jump to **View/Edit Client**
  - Click on **Add** button
    - Jump to **Add Client**

Add client

Task Decomp

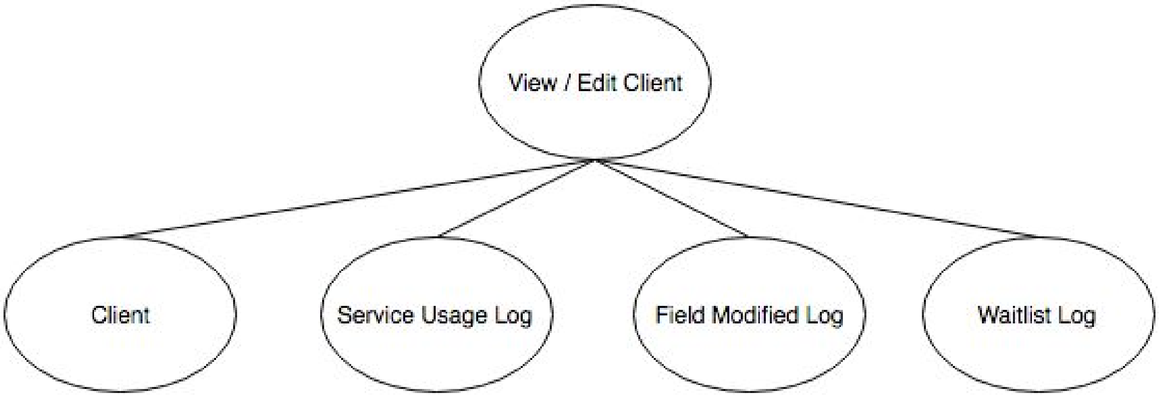
Lock Types: Update on Client table
Enabling Conditions: Enable by a user request to add client through Client Search Report
Frequency: Medium
Schemas: Single
Consistency: Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

- Display input boxes
    - Client name
    - ID number
    - Description
    - Contact phone number
- Display **Cancel** and **Submit** buttons
- Upon:
    - Click on **Cancel** button
        - Jump to back to **View Client Search / Report**
    - Click on **Submit** button
        - Run **Add Client Entry** task
        - Jump to back to **View Client Search / Report**

View/Edit Client

Task Decomp



Lock Types: Read, Update on Client, Service Usage Log, Waitlist Log, Field Modified Log table

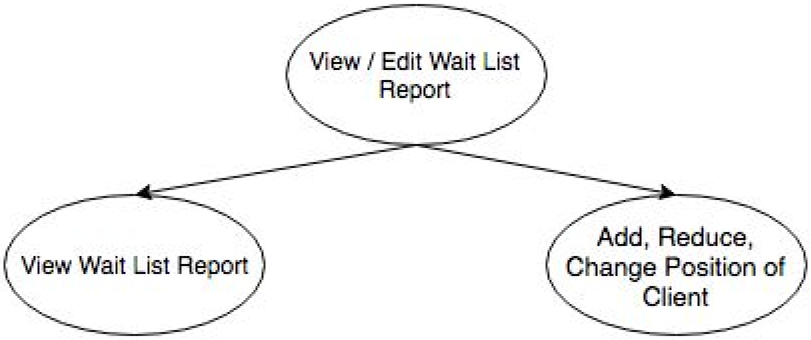
Enabling Conditions: Enabled by user request to view/edit client through Client Search Report
Frequency: High
Schemas: Several different schema constructs are needed
Consistency: Critical
Subtasks: All tasks must be done, but can be done in parallel. Mother task is required to coordinate subtasks. Order is not necessary.

Abstract Code

- Display text boxes
    - Client name
    - ID number
    - Description
    - Contact phone number
- Display **Edit** button
    - Click on **Edit** button
        - Text boxes become input boxes
- Display table of Service Usage Log
- Display table of Waitlist Log
- Display table of Field Modified Log
- Display table of available services as a scrollable view
    - Each service row should display
        - Type
        - Description
        - Hours of Operation
        - Conditions
        - Etc.
- Upon:
    - Click on a service row
        - If the service is available, highlight the row
        - If the service is not available, show a message
- Display **Cancel** and **Submit** buttons
- Upon:
    - Click on **Cancel** button
        - Jump to back to **View Client Search / Report**
    - Click on **Submit** button
        - Run **Edit Client** task
        - Run **Record Client Check Into Service** task
        - Jump to back to **View Client Search / Report**

View Wait List Report

Task Decomp


View / Edit Wait List Report
View Wait List Report
Add, Reduce, Change Position of Client

Lock Types: Read, Update on Waitlist table
Enabling Conditions: Logged in Users who is associated with a site that has a Shelter
Frequency: Medium
Schemas: Several different schema constructs are needed
Consistency: Not critical, order is not critical.
Subtasks: Mother task is required to coordinate subtasks. Decomposition needed.

Abstract Code

- Run **View Wait List** task on $SiteID
- If no clients on the waitlist, display “The wait list is empty”
- Else, display list of clients. Each client row should show:
    - Client’s name
    - ID number
    - Description
    - Contact phone number
- Able to drag and drop a client row to change its order and save in the database
- Display **Delete** button in each client row
- Upon:
    - Click on **Delete** button, client row disappears
- Display Reset and Submit buttons
- Upon:
    - Click on **Reset** button
        - Revert back to previous form

    - Click on **Submit** button
        - Apply changes - Run **Add, Reduce, Change Position Of Client** task

View Item Search/Report

Task Decomp

Lock Types: Read-only on Item table
Enabling Conditions: Logged in Users
Frequency: Medium
Schemas: Single
Consistency: Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

- Display filters:
    - Food Bank (Default: All)
    - Expiration Date (Default: All)
    - Storage Type (Default: All)
    - Food/Supply (Default: Food)
    - Sub-category (Default: depending on Food/Supply)
    - Item name / Description input box
- Run query based on the above filters to list items. Item with 0 # of units will not be displayed.
- Upon:
    - Click on an item
        - If owned, jump to **View Owned Item**
        - If not owned, jump to **View Item Request**

View Owned Item

Task Decomp

Lock Types: Read, Update on Item table
Enabling Conditions: the user click on an item which is owned by his/her associated site which has Food bank.
Frequency: Medium
Schemas: Single

Consistency: Critical
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

- Run **View Owned Item** task based on $ItemID
- Display Current Item's # of units
- Display New Item's # of units input box
- Display **Cancel** and **Submit** buttons
- Upon:
    - Click on **Cancel** button
        - Jump back to **View Item Search / Report**
    - Click on **Submit** button
        - Run **Edit Item's # of Units** task
        - Jump back to **View Item Search / Report**

View Item Request

Task Decomp

Lock Types: Read, Update on Request table
Enabling Conditions: the user click on an item which is not owned from the Item Search Report
Frequency: Medium
Schemas: Single
Consistency: Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

- Run **Search for Items** task based on $ItemID
- Display Current Item's # of units
- Display # Items requested input box
- Display **Cancel** and **Submit** buttons
- Upon:
    - Click on **Cancel** button
        - Jump back to **View Item Search / Report**
    - Click on **Submit** button
        - Run **Request for Item** task
        - Jump back to **View Item Search / Report**

View Outstanding Requests Report

Task Decomp

Lock Types: Read-only on Request table
Enabling Conditions: Logged in Users who is associate a site with Food bank
Frequency: Medium
Schemas: Single
Consistency: Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

- Run the **View Requests** task based on $ServiceID
- Display all requests. Each request row should show:
  - Item name
  - Storage type
  - Category
  - Sub-category
  - # Item requested
- For each request, compare #Item requested with Item's # of unit. If the #Item requested is more than Item's # of unit, display red on this request.
- Click on a request
  - Jump to **View/Edit Request**

View/Edit Request

Task Decomp

Lock Types: Read, Update on Request table
Enabling Conditions: Enable by a user click a request of Outstanding Request report
Frequency: Medium
Schemas: Single
Consistency: Critical
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

- Run the **View Request** task based on $RequestID
- A request should show:

        - Item name
        - Storage type
        - Category
        - Sub-category
        - # Item requested
- Display an input box for user to key in # Item provided
- User can also click on these 2 buttons to quickly change the input box
    - Fullfil: #Item Provided will be equal to #Item Requested
    - Deny: #Item Provided will be 0
- Display **Cancel** and **Submit** buttons
- Upon:
    - Click on **Cancel** button
        - Jump back to **View Outstanding Requests Status Report**
    - Click on **Submit** button
        - Run **Edit Outstanding Request** task
        - Jump back to **View Outstanding Requests Status Report**

View Requests Status Report

Task Decomp

Lock Types: Read, Delete on Request table
Enabling Conditions: Logged in Users
Frequency: Medium
Schemas: Single
Consistency: Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

- Run the **View Requests** task based on $UserID
- Display all requests. Each request row should show:
    - Food Bank name
    - Item name
    - # Items requested
    - # Items provided (only for closed requests)
    - Status: pending/closed
    - **Cancel** button (only for pending requests)
- Upon:

    - Click on **Cancel** button
        - Run **Cancel Request** task
    - Click on **Back** button.
        - Jump back to **Main Menu**.

Reference

1. “SQL Data Types”, “Constraints”, “Task Decomp”, “Abstract Code”, *Phase 1 Report CS 6400 Fall 2016 Template*
2. *Phase 1 IFD Diagram CS 6400 Spring 2017 Template*
3. *Phase 1 EER Diagram CS 6400 Spring 2017 Template*