Table of Contents

Table of Contents

Task Decomposition with Abstract Code:

Login

Abstract Code

- User enters *username* ($Username), *password* ($Password) into input fields
- If data validation is successful for both *username* and *password* input fields then:
    - When ***Enter*** button is clicked:

```
//assume $Email of current user is managed by application
SELECT * FROM User WHERE email = '$Email' AND password = '$Password';
```

        - If User.username is found but User.password != '$Password':
            - Go back to **Login** form, with error message.
        - Else:
            - Store login information as session variable $Username.
            - Go to **View Sites and Services** form.
- Else *username* and password fields are invalid, display **Login** form, with error message.

View Sites and Services

Abstract Code

- Run the ***View Sites and Services*** task based on $Username.

Table of Contents

- Find the current User using the User $Email; Display User's FirstName and LastName.
- Find the Site for the User; Display Site ShortName, Address and PhoneNum.

```
SELECT F.FirstName, F.LastName, S.ShortName, S.Street, S.City, S.State, S.ZipCode,
S.PhoneNum
FROM User F
INNER JOIN Site S ON F.SiteID = S.SiteID
WHERE F.email = $Email;
```

- Find each of the Services for the User; Display: Service, Description, Hours and Condition.

```
SELECT U.Type, T.Description, T.Hours, T.Conditions
FROM User F
INNER JOIN Site R ON F.SiteID = R.SiteID
INNER JOIN Service U ON U.SiteID = R.SiteID
INNER JOIN ClientAccessible T ON T.ServiceID = U.ServiceID
WHERE F.email = $Email;
```

Create/Delete/Edit Services

Abstract Code

- **View Sites and Services** task.
- If User clicks ***Create Service*** button; Display an input text field to enter a new Service. Click ***Save*** button. Go to **View Sites and Services** form.

```
for each $Service
INSERT INTO Service (ServiceID, Type, SiteID) VALUES ($ServiceID, $Type, $SiteID);
end for
```

- If User clicks ***Delete Service*** button; Checkboxes appear to select Service(s) for deletion. Click ***Delete*** button. Click ***Save*** button. Go to **View Sites and Services** form.

```
DELETE FROM Service
WHERE ServiceID = $ServiceID
AND Type = $Type
AND SiteID = $SiteID;
```

- If User clicks ***Edit Service*** button; Display text box to edit Service name. Click ***Save*** button. Go to **View Sites and Services** form.

```
for each $Type
UPDATE Service
SET Type = $Type
WHERE Type = $OldType;
```

Table of Contents

```
end for
```

View Meals Remaining in Inventory

Abstract Code

- User or site visitor navigates to **Login** form.
- User or site visitor clicks link to **Meals Remaining** form.
- Display FoodBank inventory, because user or site visitor is not logged-in, the inventory across all food banks is displayed.

```
SELECT I.Name, I.Amount, I.Category, I.SubCategory, I.StorageType, I.ExpDate
FROM Item I
INNER JOIN Service S ON I.SerivceID = S.ServiceID
WHERE S.Type = 'Food Bank';
```

View Available Rooms/Bunks

Abstract Code

- User or site visitor navigates to **Login** form.
- User or site visitor clicks link to **Available Bunks/Rooms** form.
- If Shelter Rooms Available and/or Bunks Available > 0
    - Display Shelter name, Site Location, Site Phone Number, Shelter Hours, Shelter Conditions, Number of Available Bunks by Bunk type, Number of Available Rooms

```
SELECT F.shortName, F.phoneNum, F.Street, F.City, F.State, F.zipCode, T.description,
T.hours, T.conditions, T.type, Y.isAvailable
FROM Site F
INNER JOIN User R ON F.SiteID = R.SiteID
INNER JOIN Service U ON U.SiteID = R.SiteID
INNER JOIN ClientAccessible T ON T.ServiceID = U.ServiceID
INNER JOIN Shelter X ON X.ClientAccessID =T.ClientAccessID
INNER JOIN Bunk Y ON Y.ShelterID = X.ShelterID
WHERE Email = $Email
AND F.SiteID = $SiteID
AND U.ServiceID = $ServiceID
AND T.ClientAccessID = $ClientAccessID
AND X.ShelterID = $ShelterID
AND Y.isAvailable = TRUE;
```

Table of Contents

```
SELECT F.shortName, F.phoneNum, F.Street, F.City, F.State, F.zipCode, T.description,
T.hours, T.conditions, Y.number, Y.isAvailable
FROM Site F
INNER JOIN User R ON F.SiteID = R.SiteID
INNER JOIN Service U ON U.SiteID = R.SiteID
INNER JOIN ClientAccessible T ON T.ServiceID = U.ServiceID
INNER JOIN Shelter X ON X.ClientAccessID =T.ClientAccessID
INNER JOIN Room Y ON Y.ShelterID = X.ShelterID
WHERE R.Email = $Email
AND F.SiteID = $SiteID
AND U.ServiceID = $ServiceID
AND T.ClientAccessID = $ClientAccessID
AND X.ShelterID = $ShelterID
AND Y.isAvailable = TRUE;
```

- Else Display message “Sorry, all shelters are currently at maximum capacity.”

View Request Statuses

Abstract Code

- Find the current User using the User Username.
- Find the Requests for the User; Display: Requests, Amount Requested, and Request Status.

```
SELECT T.Destination, T.Item, T.amtRequested, T.Status
FROM User F
INNER JOIN Site R ON F.SiteID = R.SiteID
INNER JOIN Service U ON U.SiteID = R.SiteID
INNER JOIN Request T ON T.ServiceID = U.ServiceID
WHERE F.email = $Email
AND U.Type = ‘Food Bank’
AND T.Status != ‘Closed’;
```

- If Request Status == Closed
    - Display Amount Provided

```
SELECT T.amtProvided
FROM User F
INNER JOIN Site R ON F.SiteID = R.SiteID
INNER JOIN Service U ON U.SiteID = R.SiteID
INNER JOIN Request T ON T.ServiceID = U.ServiceID
WHERE F.email = $Email
```

Table of Contents

```
AND U.Type = 'Food Bank';
```

Cancel Requests

Abstract Code

- User executes the **View Request Statuses** task.
- User clicks the ***Cancel Requests*** button.
- User selects the requests for cancellation and confirms the transaction

```
DELETE FROM Request
WHERE RequestID = $RequestID
AND Source = $Source
AND Destination = $Destination
AND Item = $Item
AND ServiceID = $ServiceID;
```

    - If user decides not to cancel the requests, clicks the ***Cancel*** button.
- Request is removed from system, user returns to updated **Request Status** form.

View Outstanding Requests

Abstract Code

- Find the current User using the User Username.
- Find the Requests associated with a User's Site that has a Food Bank; Display: Item, Storage type, Category and Amount Requested.

```
SELECT T.Source, T.Item, T.amtRequested, T.Status, T.Category, T.StorageType
FROM User F
INNER JOIN Site R ON F.SiteID = R.SiteID
INNER JOIN Service U ON U.SiteID = R.SiteID
INNER JOIN Request T ON T.ServiceID = U.ServiceID
WHERE F.email = $Email
AND U.Type = 'Food Bank';
```

Modify Outstanding Requests

Abstract Code

- Run **View Outstanding Requests** task.
- Find Food Bank Inventory; Display Food Bank's inventory as a table.

Table of Contents

- User clicks ***Modify Requests*** button to edit the requests.
- If user can fulfill the request in full
    - Check the ***Fulfilled in Full*** checkbox.

```
//fulfilled in full
For each $Status, $amtProvided
UPDATE Request
SET Status = $Status, amtProvided = $amtProvided
WHERE RequestID = $RequestID
AND Status = $OldStatus
AND amtProvided = $OldamtProvided
AND (Request.amtRequested <= (SELECT SUM(amount) FROM item WHERE name =
Request.item) –  SUM(Request.amtRequested));
End for
```

- If the User can only partially fulfill the request
    - Enter the number of items provided and check the ***Partially Fulfilled*** checkbox.

```
//partially fulfilled
For each $Status, $amtProvided
UPDATE Request
SET Status = $Status, amtProvided = $amtProvided
WHERE RequestID = $RequestID
AND Status = $OldStatus
AND amtProvided = $OldamtProvided
AND (Request.amtRequested >
(SELECT SUM(amount) FROM item WHERE name = Request.item) –
SUM(Request.amtRequested))
AND (SELECT SUM(amount) FROM item WHERE name = Request.item) –
SUM(Request.amtRequested)) >0;
End for
```

- Else check the ***Unable to Fulfill*** checkbox.

Modify Bunk Count

Abstract Code

- Find the current User using the User Username.
- Run the ***View Sites and Services*** task based on Username.
- User clicks on **Modify Bunk Count** link underneath Shelter name.
- User uses **+** and **–** buttons to modify bunk count at Shelter.
- User clicks ***Save*** button.

```
UPDATE Bunk
SET Number = $Number
```

Table of Contents

```
WHERE ShelterID = $ShelterID AND BunkID = $BunkID AND isAvailable = TRUE AND
Type = $TYPE AND Number = $OldNumber;
```

Modify Items in Inventory

Abstract Code

- Find the current User using the User Username.
- Run the ***View Sites and Services*** task based on Username.
- User clicks on **Modify Inventory** link underneath Food Bank name.
- User navigates to **Add/Remove Items to Inventory** form.
- If User is adding an item to inventory:
    - User clicks ***Add*** button
    - Enters item name and quantity into text boxes.
    - User clicks ***Save*** button.

```
//add item
INSERT INTO [Inventory] (Item, Quantity)
VALUES($Item, $Quantity);
```

- If User is removing an item from inventory:
    - User clicks ***Remove*** button
    - User clicks **–** button to reduce quantity of item, if item quantity == 0, item is removed from inventory
    - User clicks ***Save*** button.

```
//reduce item quantity
UPDATE [Inventory]
SET Item = $Item AND Quantity = $Quantity
WHERE Item = $Item AND Quantity = $OldQuantity

//remove item from inventory
DELETE FROM [Inventory]
WHERE Item = $Item AND Quantity = $Quantity;
```

View Waitlist

Abstract Code

- Find relevant shelter based on $shelterID

```
SELECT * FROM waitlist WHERE shelterID = $shelterID;
```

- If Waitlist is empty, print that waitlist is empty.
- For each ClientID in the waitlist:
    - Print the ClientID and Rank in a list sorted by Rank

Table of Contents

- Add ***Delete, Move Up, Move Down*** buttons for each row that trigger **Delete/Modify Entry** task

Delete/Modify Entry

Abstract Code

- If ***Delete*** button clicked:
  - Remove current row from waitlist table

```
DELETE FROM waitlist WHERE waitlistID = $clickedID;
```

  - For each ClientID with Rank higher than deleted entry's Rank:
    - Decrease rank by 1

```
UPDATE waitlist
SET (rank = rank – 1)
WHERE rank > $oldRank
AND shelterID = $shelterID;
```

  - Call **View Waitlist** task to refresh page
- If ***Move Up*** button clicked:
  - Increment rank for above row by 1

```
UPDATE waitlist
SET rank = rank + 1
WHERE rank = $clickedRank – 1
AND shelterID = $shelterID;
```

  - Decrement rank for current row by 1

```
UPDATE waitlist SET rank = rank – 1 WHERE waitlistD = $clickedID;
```

  - Call **View Waitlist** task to refresh page
- If ***Move Down*** button clicked:
  - Decrement rank for below row by 1

```
UPDATE waitlist
SET rank = rank - 1
WHERE rank = $clickedRank – 1
AND shelterID = $shelterID;
```

  - Increment rank for current row by 1

```
UPDATE waitlist SET rank = rank – 1 WHERE waitlistD = $clickedID;
```

  - Call **View Waitlist** task to refresh page

Table of Contents

Search and View Items

Abstract Code

- Retrieve list of all FoodBanks in $FoodBankSelection input field and display in drop down menu, with added “All” option.

```
SELECT * FROM site NATURAL JOIN service WHERE service.type = “FoodBank”;
```

- Add input fields to filter items by $ExpirationDate, $StorageType, $FoodOrSupply, $Category, and $Keyword search for name / description.
- When user clicks ***Search*** button:
    - If individual $FoodBankSelection selected, retrieve list of items from that FoodBank matching filters

```
SELECT name, amount, category, storageType, expDate, serviceID FROM item
WHERE
(serviceID = $selectedID) AND
(expDate = $ExpirationDate) AND
(storageType = $StorageType) AND
(category = $FoodOrSupply) AND
(name LIKE %$Keyword%);
```

    - Otherwise, retrieve list of items from all FoodBanks matching filters

```
SELECT name, amount, category, storageType, expDate, serviceID FROM item
WHERE
(expDate = $ExpirationDate) AND
(storageType = $StorageType) AND
(category = $FoodOrSupply) AND
(name LIKE %$Keyword%);
```

    - For each item returned:
        - Display item Name, Amount, Category, StorageType, ExpirationDate
        - If Item belongs to User’s FoodBank:
            - Add button to ***Modify Number of Items***
        - Otherwise, add button to ***Request Items***
- If ***Request Items*** is clicked:
    - Call **Request Items** task
- If ***Modify Number of Items*** is clicked:
    - Call **Modify Number of Items** task

Request Items

Abstract Code

- Create pop-up with input form for $Quantity of item requested.
- If User clicks ***Ok***:

Table of Contents

    - If quantity requested exceeds total quantity available at FoodBank:

```
SELECT amount FROM item WHERE itemID = $ClickedID;
```

        - Display error message
    - Otherwise, Insert a request entry containing Source FoodBank, Destination FoodBank, Item, Quantity Requested, Status = “Pending”, AmountProvided = 0

```
INSERT INTO request
(Source, Destination, itemID, amtRequested, status, amtProvided, serviceID, userID) VALUES
($SourceSiteID, $DestSiteID, $ClickedID, $Quantity, “Pending”, 0, $ServiceID, $UserID);
```

        - Close **Requests** pop-up
- If User clicks ***Cancel:***
    - Close **Requests** pop-up

Modify Number of Items

Abstract Code

- Create pop-up with input form for $NewQuantity of item being updated.
- If User clicks ***Ok***:
    - If $NewQuantity == 0:
        - Delete entry for that item from FoodBank

```
DELETE FROM item WHERE itemID = $ClickedID;
```

    - Otherwise, update Amount for that item to $NewQuantity

```
UPDATE item SET amount = $NewQuantity WHERE itemID = $ClickedID;
```

    - Close **Modify Number of Items** pop-up
- If User clicks ***Cancel:***
    - Close **Modify Number of Items** pop-up

Enroll Client

Abstract Code

- Display input fields for $ClientID, $FirstName, $LastName, $PhoneNum, $Description
- If User clicks ***Add Client*** button:
    - If not all fields filled in with valid data:
        - Display error message next to incorrect field
    - Otherwise, Insert new Client using information from fields $ClientID, $FirstName, $LastName, $PhoneNum, $Description

Table of Contents

```
INSERT INTO client (clientID, Description, firstName, lastName, phoneNum)
VALUES
($ClientID, $FirstName, $LastName, $PhoneNum, $Description);
```

- Trigger **Add New Client Log Entry** task with Service as “Enroll”

View Full Client Report

Abstract Code

- Display input field for $NameOrIdNumber
- If ***Search*** button is clicked:
  - Perform query to find Clients with Name or ClientID LIKE $NameOrIdNumber

```
SELECT * FROM client WHERE
(clientID LIKE %$NameOrIdNumber%) OR
(firstName LIKE %$NameOrIdNumber%) OR
(lastName LIKE %$NameOrIdNumber%);
```

  - Count total number rows returned. If count == 0
    - Display error message stating no Clients found matching search string
  - Else if count > 5:
    - Display error message stating search term too general
  - Else, display similar search results, User clicks on desired Client from results :
    - Display all columns in Client table for matching ClientID
    - Perform query to find ClientLogE entries with matching ClientID

```
SELECT * FROM clientLog WHERE clientID = $MatchedID
```

    - Perform query to find Waitlist entries with matching ClientID

```
SELECT site.shortName, site.siteID, waitlist.rank FROM waitlist
NATURAL JOIN shelter
 NATURAL JOIN clientAccessible
 NATURAL JOIN service
 NATURAL JOIN site
 WHERE clientID = $MatchedID;
```

    - Display all fields in each ClientLogE returned, sorted by newest first
    - Display all Shelter ShortNames for which the ClientID is waitlisted, and their Rank.
    - Display ***Modify Client’s Information*** button
    - Display ***Add New Client Log Entry*** button
    - Display ***Place on Waitlist*** button
  - If ***Modify Client’s Information*** button is clicked:
    - Trigger **Modify Client’s Information** task
  - If ***Add New Client Log Entry*** button is clicked:
    - Trigger **Add New Client Log Entry** task
  - If ***Place on Waitlist*** button is clicked:
    - Trigger **Place on Waitlist** button

Table of Contents

- If ***Clear*** button is clicked:
    - Redraw new **Client Search and Results Form**

Modify Client Information

Abstract Code

- Make fields editable for for $FirstName, $LastName, $PhoneNum, $Description
- If User clicks ***Save Client Information*** button:
    - If not all fields filled in with valid data:
        - Display error message next to incorrect field
    - Otherwise, Update Client matching ClientID using information from fields $FirstName, $LastName, $PhoneNum, $Description

```
UPDATE client SET
firstName = $FirstName, lastName = $LastName, phoneNum = $PhoneNum,
Decscription = $Description
WHERE clientID = $MatchedID;
```

        - Trigger **Add New Client Log Entry** task with Service as “Updated Client Information.”
- If User clicks ***Cancel*** button:
    - Make fields no longer editable and display their original values

Add New Client Log Entry

Abstract Code

- Show new window with editable fields for $Site, $Service, and $Notes.
- $EntryDate field will automatically be filled in. Display $MatchedID and $ClientName at top of window.
- If User clicks ***Add Log Entry*** button:
    - If not all fields filled in with valid data:
        - Display error message next to incorrect field
    - Otherwise, Insert ClientLogE for ClientID using information from fields $Site, $Service, $Notes, $EntryDate

```
INSERT INTO clientLog (entryDate, siteID, service, Notes, fieldModif, clientID)
VALUES
($EntryDate, $Site, $Service, $Notes, $EntryDate, $MatchedID);
```

- If User clicks ***Cancel*** button:
    - Close window without inserting new UserLogE. Return to previous form

Table of Contents

Place on Waitlist

Abstract Code

- Display confirmation box asking User if they would like to add the ClientID with Name to Waitlist for Shelter that User is signed in to.
- If User clicks ***Add Client To Waitlist*** button:
    - Insert a new waitlist entry with $MatchedID provided by **Client Report Search Form**, Rank and $EntryDate automatically determined

```
INSERT INTO waitlist (entryDate, clientID, shelterID) VALUES
($EntryDate, $MatchedID, $Shelter);
```

    - Display Message that Client Name was added to Waitlist successfully
- If User clicks ***Cancel*** button:
    - Close confirmation box

Table of Contents