Sheltered Ones

Software Design Document

Version 1.0

Team Members

Conor Babock, Gary Hui, John Wang, Roman Yarik

March 27, 2020

Contents

[Revision History 3](#_Toc34632410)

[1. Introduction 4](#_Toc34632411)

[1.1. Problem statements 4](#_Toc34632412)

[1.2. Proposed solution 4](#_Toc34632413)

[1.3. Novelty 4](#_Toc34632414)

[2. Glossary 5](#_Toc34632415)

[3. System requirements 6](#_Toc34632416)

[3.1. Functional requirements 6](#_Toc34632417)

[3.2. Non-functional requirements 7](#_Toc34632418)

[3.3. Other requirements 7](#_Toc34632419)

[4. Use cases 8](#_Toc34632420)

[4.1. Summary 8](#_Toc34632421)

[4.2. Actors 8](#_Toc34632422)

[4.3. Diagram 9](#_Toc34632423)

[4.4. Details 10](#_Toc34632424)

[4.5. Traceability matrix 21](#_Toc34632425)

[5. System architecture 22](#_Toc34632426)

[5.1. Overview 22](#_Toc34632427)

[5.2. Details 23](#_Toc34632428)

[5.3. Tradeoffs 24](#_Toc34632429)

[5.4. DBMS choices 25](#_Toc34632430)

[6. Implementation 25](#_Toc34632431)

[6.1. User Interfaces 25](#_Toc34632432)

[6.2. Class diagram 32](#_Toc34632433)

[6.3. Database diagram 32](#_Toc34632434)

[6.4. Others 32](#_Toc34632435)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 3/27/2020 | 1.0 | First draft constructed after the completion of Sprint 1 | Gary Hui |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Problem statement

Homeless shelters offer beds to sleep on during the night; however, there are times when the beds for these shelters are filled up, leaving homeless people with no place to sleep. They must travel to various other homeless shelters in search for vacant beds that they can occupy. Homeless people often lack transportation, so it can take them several minutes or a few hours of walking until they reach their destination. Their journey continues until they discover a shelter that can take them in.

Homeless people frequently cannot access enough information about the homeless shelters regarding the number of beds available. They resort to taking a more tactical approach by visiting every homeless shelter they can, in hopes that they can find a bed.

## Proposed solution

We propose a strategic approach for homeless people in search of a bed. That is, a website that will display information about homeless shelters.

The website will have details about various shelters, such as: description, location, phone number, and number of beds available. Shelter organizations will periodically update the number of beds available, which will be inputted in the database and displayed onto the website. Shelter organization’s entries onto our system may be added.

The website will inform the user of nearby shelters. This is accomplished by the website to conduct a search when given the address, zip code, and/or their current location. A map service will pop-up on the website, displaying nearby homeless shelters relative to the given location. There, customers may click on shelters to view their information. The map service will also provide directions if requested.

Our service will grant homeless people the information they need to determine which shelter to visit next.

## Novelty

Our proposed solution will have these following set of features so that it may be a successful resource in combating the problem statement.

1. Website registration
2. Notify users if their credentials are incorrect
3. Database that stores shelter organization’s background information
4. Website that reflects database information
5. Picture preview of the shelter
6. Time when the information was updated
7. System periodically updates the website information based on the database
8. Allows customers to know how many beds are available at each shelter
9. Multiple types of accounts, which are customer account, shelter staff account, and admin account
10. Allows registered accounts to store favorite shelters for quick access
11. Allows shelter staff to change their available bed count
12. Find shelters near a given location
13. Supports customers who use current location services
14. Give directions from one address to another
15. Centralized information: Our website will display information gathered from various shelters. This will be a valuable resource for those seeking shelter. Instead of searching through multiple shelter organization websites for specific information, they can initially enter our website and be complete with their research.
16. Dynamic updates: Our system will periodically update what is being displayed on the website based on the information stored inside the database. This maintains our displayed information to be relevant and accurate.

# Glossary

* Account: An entry into the database that holds a member’s information and restrictions and requires specific credentials to be accessed
* Address: The physical location of the shelter relative to street.
* Customer: A person that accesses the website for getting information about their local shelters.
* Database: The repository of account information for all registered users, alongside information about the shelters.
* Favorites: a bookmark functionality for users to save certain shelters onto their account
* IT Staff: Support members for the staff of the shelter that focus on work, maintenance and development of the website and the database.
* Shelter: A place for homeless people to receive food and bedding among other accommodations. A shorter word for homeless shelter.
* Shelter Staff: Members of the shelter who assist in day to day functionality of the shelter.
* Website: The service that acts to display information on the database for all users
* Zip Code: The physical address of the shelter relative to county.

# System requirements

## 3.1. Functional requirements

|  |  |  |
| --- | --- | --- |
| Identifier | Priority | Description |
| REQ-1 | 4 | The system will allow customers to create an account on the website and register with their information |
| REQ-2 | 3 | The system will store shelter information only modified by shelter staff and IT staff |
| REQ-3 | 2 | The system will display shelter information from a website |
| REQ-4 | 3 | The system will allow users to view the name, address, and number of beds available at each listed shelter |
| REQ-5 | 4 | The system will allow customers to search nearby shelters in respect to a given location. |
| REQ-6 | 3 | The system can take the device’s current location as an input |
| REQ-7 | 5 | The system will show the time when the respective shelter’s information was last modified |
| REQ-8 | 4 | The system will implement a street map functionality that includes the inputted source and destination |
| REQ-9 | 5 | The system will implement the ability to give directions to a desired shelter |
| REQ-10 | 3 | The system will not allow shelter staff to modify information aside from their own shelter |
| REQ-11 | 4 | The system will allow IT staff to manually input an entry for a shelter |
| REQ-12 | 3 | The system will periodically update the website with the newest information stored onto the database |
| REQ-13 | 4 | The system allows IT staff to back up the database |
| REQ-14 | 4 | The system allows IT staff to restore the database from a restore point |
| REQ-15 | 5 | The system will send approval codes for shelter staff accounts and IT staff accounts to be created |

## 3.2. Non-functional requirements

|  |  |  |
| --- | --- | --- |
| Identifier | Priority | Description |
| REQ-16 | 5 | To register, customer accounts will need to provide their full name and email |
| REQ-17 | 5 | To register, shelter staff accounts must provide their organization name, organization address, phone number, email, first name, last name, and approval code |
| REQ-18 | 3 | Customers will be able to select a nearby shelter to reveal additional information about them |
| REQ-19 | 5 | Customers, IT staff, and shelter staff must create a password for their account and meet a criterion |
| REQ-20 | 4 | Account passwords are encrypted |
| REQ-21 | 4 | Users will need to input a username and password in order to log in |
| REQ-22 | 3 | Changes made to the database should be automatically updated onto the website within a minute |

## 3.3. Other requirements

|  |  |  |
| --- | --- | --- |
| Identifier | Priority | Description |
| REQ-23 | 5 | Have an announcement page on the website that users may visit |
| REQ-24 | 4 | A picture of the shelter’s building will be displayed next to their listing |
| REQ-25 | 4 | Count how many unique customers have accessed our website |
| REQ-26 | 4 | Display the total number of shelters listed on the website |
| REQ-27 | 4 | Allow registered accounts to view their own profile |
| REQ-28 | 4 | Allow registered accounts to store favorite shelters for more accessibility |

# 4. Use cases

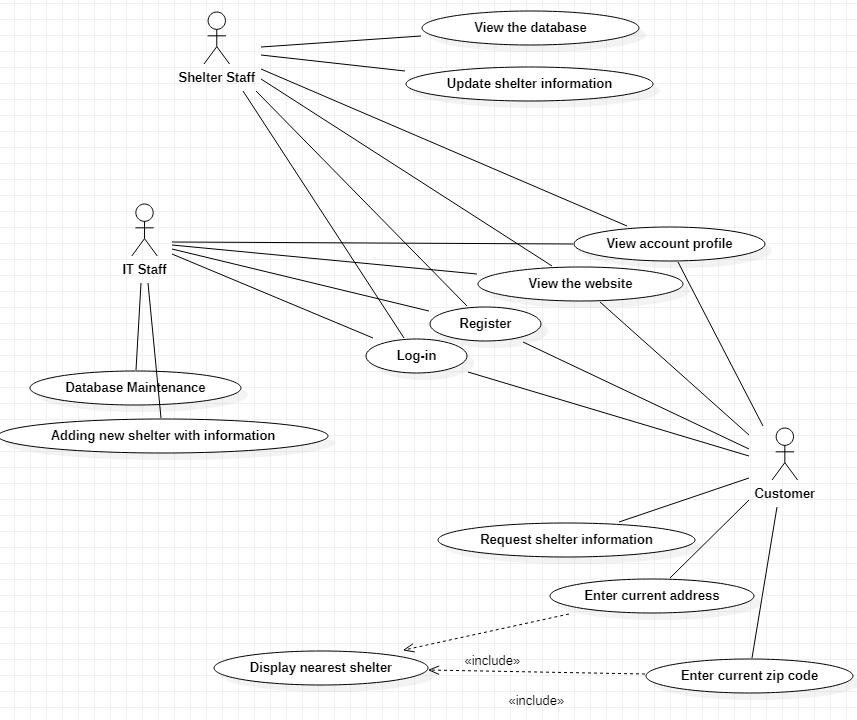
## 4.1. Summary

|  |  |  |
| --- | --- | --- |
| Identifier | Short name | Description |
| UC-1 | Register | To register an account on the website |
| UC-2 | Database management | To be able to maintain stability for the database |
| UC-3 | Adding new shelter with information | to store and display a shelter’s information that has not yet been added |
| UC-4 | View the website | To view information on the website |
| UC-5 | View the database | to view information on the database |
| UC-6 | Update shelter information | To modify information on the database |
| UC-7 | Request shelter information | to view more information about a shelter |
| UC-8 | Enter current address | to input their desired address for searching nearby shelters |
| UC-9 | Enter current zip code | To input their desired zip code for searching nearby shelters |
| UC-10 | View account profile | To display basic account information and favorite shelters |
| UC-11 | Display nearest shelter | To show a map of shelters nearby a given location |
| UC-12 | Log-in | To access an account on our website |

## 4.2. Actors

|  |  |  |
| --- | --- | --- |
| Actor | Goal | Use cases |
| Customer | To view the website | UC-4 |
| Customer | To create an account on the website | UC-1, UC-4 |
| Customer | To log into their private account | UC-1, UC-4 |
| Customer | To view a listed shelter’s information | UC-4, UC-7 |
| Customer | To see what shelters are near their current location | UC-4, UC-8, UC-9, UC-11 |
| Customer | To view their own profile | UC-10 |
| Customer | To select and view their favorite shelters | UC-4, UC-10 |
| Shelter Staff | To view the database | UC-5 |
| Shelter Staff | To edit information being displayed onto the website regarding their own shelter | UC-5, UC-6 |
| IT Staff | To add another shelter to the database so it can be displayed onto the website | UC-3, UC-4, UC-5, UC-6 |
| IT Staff | To maintain the stability of the database | UC-2, UC-5 |

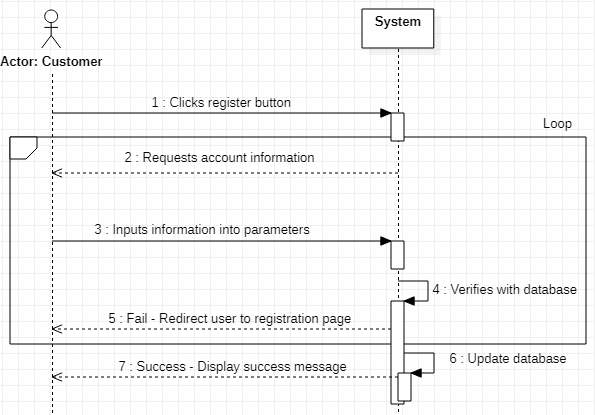
## 4.3. Diagram



## 4.4. Details

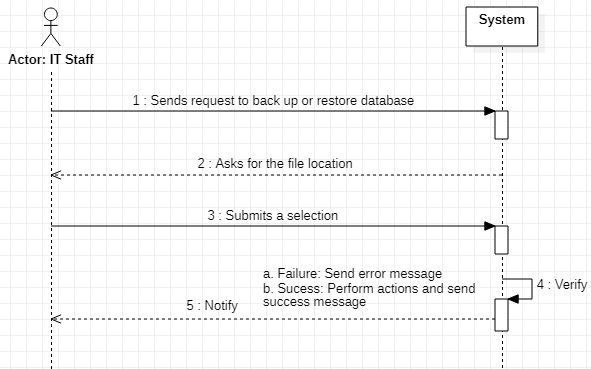
|  |
| --- |
| Use Case UC-1: Register |
| Related Requirements: REQ-1, REQ-15, REQ-16, REQ-17, REQ-19 |
| Initiating Actor: Customer |
| Actor’s Goal: To register an account on the website |
| Participating Actors: Customer, System |
| Preconditions: The system will request information about the customer from a registration page |
| Postconditions: The customer’s account will be created and stored into the database |
| Flow of Events for Main Success Scenario:  —> 1. Customer accesses the website and clicks the “Register” button  <— 2. System returns a page that request to input necessary information  —> 3. Customer enters information into the data fields, then submits  <— 4. System checks the username and email against current entries within the database   1. If the username or email is not unique, then return to Step 2 and notify that the login information already exists 2. If the username and email are both unique, then store it into the database, then notify the user of a successful registration |

Figure 1: Register sequence

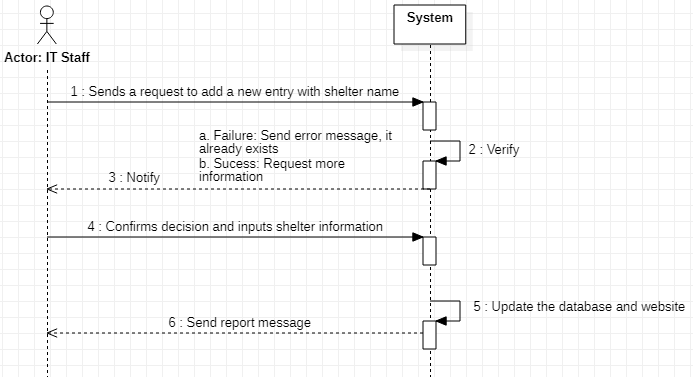


|  |
| --- |
| Use Case UC-2: Database maintenance |
| Related Requirements: REQ-11, REQ-12, REQ-13, REQ-14, REQ-22, REQ-25 |
| Initiating Actor: IT Staff |
| Actor’s Goal: To be able to maintain stability for the database |
| Participating Actors: IT Staff, System |
| Preconditions: IT staff must be viewing the database and have an account. |
| Postconditions: IT Staff has an updated back-up of the database or has successfully restored database |
| Flow of Events for Main Success Scenario:  —> 1. IT Staff sends a request to back up or restore the database  <— 2. System requests a selection of where to back-up or what to restore from  —> 3. IT Staff inputs a selection  <— 4. System verifies the operation; if true, then perform the back-up or restoration and updates the database accordingly   1. If there is not enough space to back-up the database, send an error message in Step 4 before the execution 2. If the requested operation from the IT Staff is accepted and is a back-up, then the database information is stored on the IT Staff’s storage device. If it is a restore instead of a back-up, then information is overwritten into the database |

Figure 2: Database Maintenance sequence

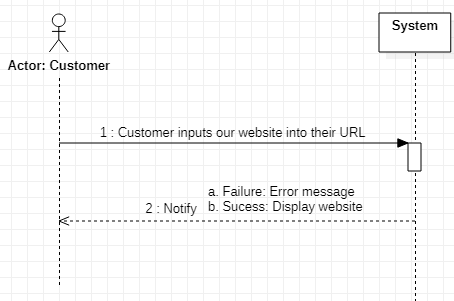


|  |
| --- |
| Use Case UC-3: Adding new shelter with information |
| Related Requirements: REQ-11, REQ-24, REQ-26 |
| Initiating Actor: IT Staff |
| Actor’s Goal: To store and display a shelter’s information that has not yet been added |
| Participating Actors: IT Staff, System |
| Preconditions: Requires valid shelter organization information that does not already have an entry into the database |
| Postconditions: The newly added shelter has information about it stored onto the database and is displayed onto the website |
| Flow of Events for Main Success Scenario:  —> 1. IT Staff sends a request to create another shelter entry  <— 2. System checks the possibility for another entry and sends IT Staff a message  a. If the IT Staff tries to add a shelter that already exists, system in Step 2 will send IT Staff an error message and exit  b. If the IT Staff adds a new shelter, then request more info  —> 3. Confirm decision and input shelter info  <— 4. System allocates a new location on the website and creates a database entry, then notifies IT Staff |

Figure 3: Adding new shelter with information sequence 

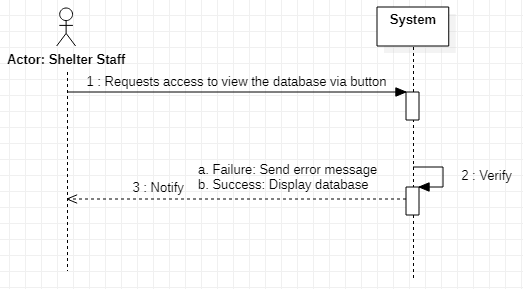
|  |
| --- |
| Use Case UC-4: View the website |
| Related Requirements: REQ-3, REQ-4, REQ-5, REQ-23, REQ-24, REQ-25, REQ-26 |
| Initiating Actor: Customer |
| Actor’s Goal: To view information on the website |
| Participating Actors: Customer, System |
| Preconditions: The Customer must have an internet connection. The system server must be online. |
| Postconditions: The Customer will be able to view the website |
| Flow of Events for Main Success Scenario:  —> 1. Customer inputs our website into the URL and sends an HTTP GET request to view the website  <— 2. The system responds by sending the current website information   1. If the server has connectivity issues, send an error message in step 2. 2. If the Customer sends a GET request to view our website and our server is up, the system will send the information for the Customer to view the website. |

Figure 4: View the website sequence



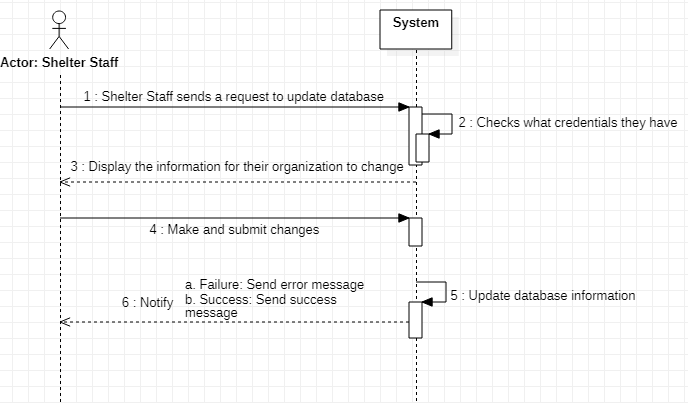
|  |
| --- |
| Use Case UC-5: View the database |
| Related Requirements: REQ-2. REQ-7. REQ-10, REQ-11 |
| Initiating Actor: Shelter Staff |
| Actor’s Goal: To be able to view the database that stores shelter information |
| Participating Actors: Shelter Staff, System |
| Preconditions: The Shelter Staff must be given permissions to access the database. Database must be functional. |
| Postconditions: The Shelter Staff can view the database information for every shelter |
| Flow of Events for Main Success Scenario:  —> 1. Shelter Staff clicks a button request access to view the database  <— 2. System verifies that the Shelter Staff has access or not, then sends a response   1. If the Shelter Staff is an invalid user, then send an error message 2. If the Shelter Staff is a valid user, then allow them to view the database |

Figure 5: View the database sequence



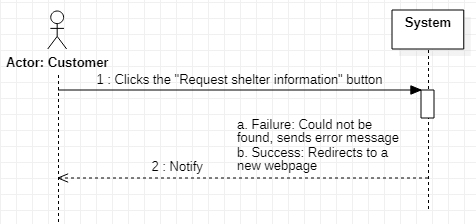
|  |
| --- |
| Use Case UC-6: Update shelter information |
| Related Requirements: |
| Initiating Actor: Shelter Staff |
| Actor’s Goal: To make changes from their shelter organization’s information onto the database |
| Participating Actors: Shelter Staff, System |
| Preconditions: The Shelter Staff must be given permissions to access the database and requires information to be updated. |
| Postconditions: The Shelter Staff can edit database information regarding their organization(s) |
| Flow of Events for Main Success Scenario:  —> 1. Shelter Staff accesses the database, then makes a request to update the database  <— 2. System displays their organization’s information  —> 3. Shelter Staff modifies their information and sends the requested changes  <— 4. System updates the changes onto the database   1. If the Shelter Staff tries to update another organization’s shelter information, then the system will send an error message 2. If the Shelter Staff makes changes to their organization, then update the information onto the database and send a success message |

Figure 6: Update Shelter Information sequence



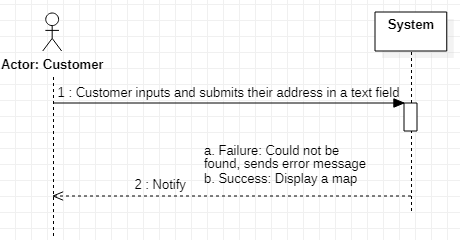
|  |
| --- |
| Use Case UC-7: Request shelter information |
| Related Requirements: REQ-2, REQ-7, REQ-10 |
| Initiating Actor: Customer |
| Actor’s Goal: To view each shelter organizations information. |
| Participating Actors: Customer, System |
| Preconditions: The shelter organization information must be stored within the database so it may be retrieved and displayed. Customer must be viewing the website. |
| Postconditions: The Customer is given information regarding the number of available beds, their contact information, and their building address. |
| Flow of Events for Main Success Scenario:  —> 1. Customer selects the “Request shelter information” button  <— 2. The system returns a new webpage to display information   1. If the server has connectivity issues, send an error message in step 2. 2. The system redirects the Customer to a webpage with the shelter’s information |

Figure 7: Request shelter information sequence



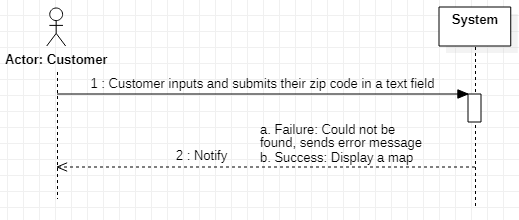
|  |
| --- |
| Use Case UC-8: Enter Current Address |
| Related Requirements: REQ-5, REQ-6 |
| Initiating Actor: Customer |
| Actor’s Goal: To input their desired address for searching nearby shelters |
| Participating Actors: Customer, System |
| Preconditions: The Customer must be able to manually enter their address in the “Address” text field on our website. Customer is accessing the website. |
| Postconditions: The system returns a map displaying their inputted address along with nearby shelters |
| Flow of Events for Main Success Scenario:  —> 1. Customer selects the “Address” text field and inputs their address.  <— 2. System returns a map to be displayed as an element on the website.   1. If the address could not be found, return an error message in step 2 2. If the address was found, the system will display a map on the website. The system provides nearby shelters and the ability to request information from them. |

Figure 8: Enter Current Address sequence



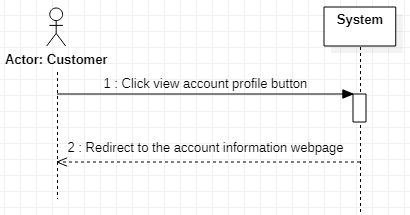
|  |
| --- |
| Use Case UC-9: Enter current zip code |
| Related Requirements: REQ-5 |
| Initiating Actor: Customer |
| Actor’s Goal: To input their desired zip code for searching nearby shelters |
| Participating Actors: Customer, System |
| Preconditions: The user must be able to manually enter their address in the “Zip Code” text field on our website. Customer accesses the website. |
| Postconditions: The system returns a map display their inputted zip code along with nearby shelters. |
| Flow of Events for Main Success Scenario:  —> 1. Customer selects the “Zip Code” text field and inputs their zip code.  <— 2. System returns a map to be displayed as an element on the website.   1. If the zip code could not be found, return an error message in step 2 2. If the zip code was found, the system will display a map on the website. The system provides nearby shelters and the ability to request information from them. |

Figure 9: Enter current zip code sequence



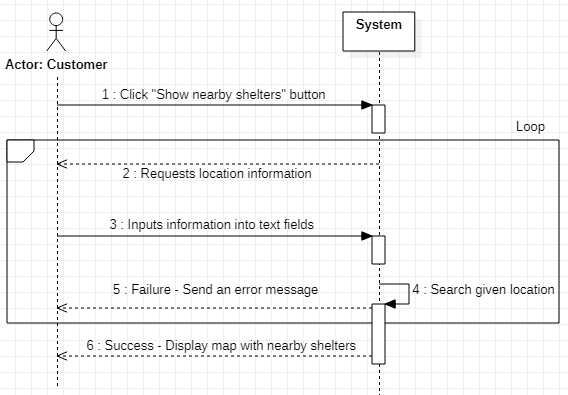
|  |
| --- |
| Use Case UC-10: View account profile |
| Related Requirements: REQ-27, REQ-28 |
| Initiating Actor: Customer |
| Actor’s Goal: To display basic account information and favorite shelters |
| Participating Actors: Customer, System |
| Preconditions: The Customer must be viewing the website and be logged into a registered account |
| Postconditions: The Customer will be able to view a page displaying their basic account information and favorite shelters |
| Flow of Events for Main Success Scenario:  —> 1. Customer is logged into our website and viewing the homepage, then clicks the “View account profile” button  <— 2. System redirects them to their account information webpage |

Figure 10: View account profile sequence



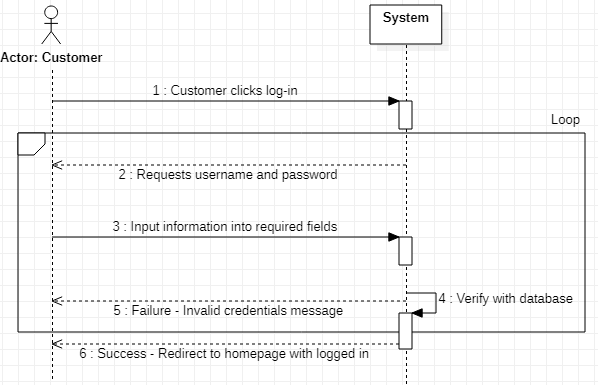
|  |
| --- |
| Use Case UC-11: Display nearest shelter |
| Related Requirements: REQ-3, REQ-5, REQ-8, REQ-9, REQ-18 |
| Initiating Actor: Customer |
| Actor’s Goal: To view nearby shelters from a given location |
| Participating Actors: Customer, System |
| Preconditions: The Customer must be viewing the website and be able to provide location information. |
| Postconditions: The Customer will be able to view a map that shows the location of nearby shelters. |
| Flow of Events for Main Success Scenario:  —> 1. Customer clicks find nearby shelters button  <— 2. System requests information such as address or zip code or current location  —> 3. Inputs information into the text fields  <— 4. System searches for the given location and sends a response  a. If the address could not be found, send an error message and return to Step 2  b. If the address is found, display a map element that shows nearby shelters |

Figure 11: Display nearest shelter sequence



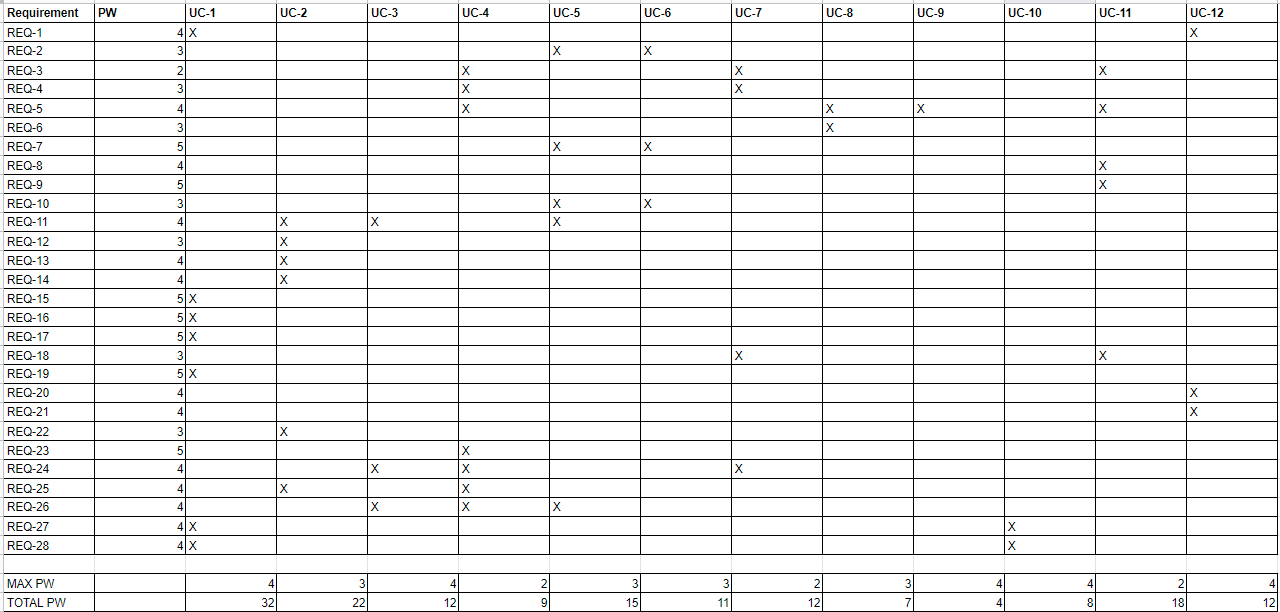
|  |
| --- |
| Use Case UC-12: Log in |
| Related Requirements: REQ-1, REQ-20, REQ-21 |
| Initiating Actor: Customer |
| Actor’s Goal: To log into their private account |
| Participating Actors: Customer, System |
| Preconditions: The Customer must be viewing the website and have a registered account |
| Postconditions: The Customer will be able view the website while being able to access their account |
| Flow of Events for Main Success Scenario:  —> 1. Customer clicks log-in  <— 2. System requests username and password  —> 3. Inputs required information into text fields  <— 4. System matches the information with the database and sends a response  a. If the credentials did not match, send an error message and return to Step 2  b. If the credentials are correct, return the customer to homepage with account access |

Figure 12: Log-in sequence



## 4.5. Traceability Matrix

Traceability matrix is used to create a value system to map the system requirements to use cases. The matrix functionality ensures that all requirements are covered by use cases. An index within the matrix is marked with an “X” if the use case is derived from the respective requirement. The highest priority is 1, while the lowest priority is 5. The higher the PW total is, the more it is considered “higher priority”. Max PW is the single highest priority requirement that it contains. Max PW and Total PW will be used to determine which use case should be deliver before the other.



## 5. System architecture

## 5.1 Overview

The architecture of Shelter Website is based on a client-server 3-tier architecture (Fig. 1), which consists of client application, web server, and database.

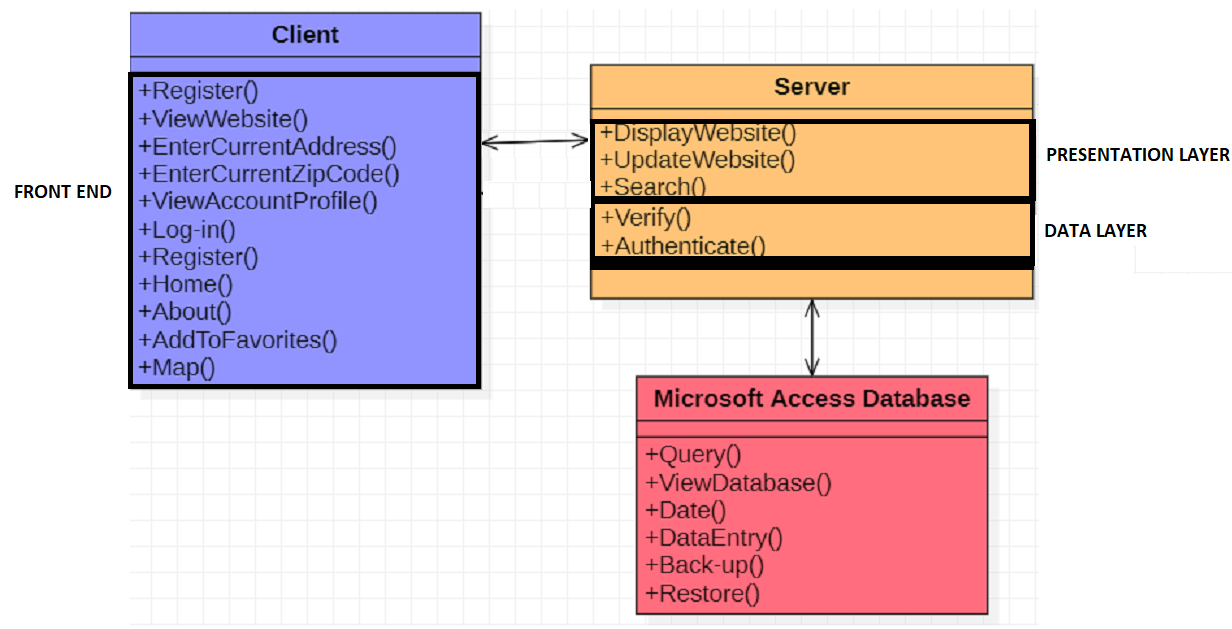


Fig. 1: Shelter Website System Architecture

The main reasons for considering three-tier architecture for the Shelter Website are as follows:

**Centralized:**

* Resources and data security are concentrated in one location and controlled through the server
* System configures accounts rather than configuring security and resource access for each computer on the network

**Flexibility:**

* Management of data is independent from the physical storage support.
* New technology may be easily integrated into the system
* Maintenance of the business logic is easier.
  + Database is currently designed to use SQL. But can easily transfer over to Azure and other cloud servers.
  + Website and database are separated for maintenance.

**Accessibility:**

* Server can be accessed through various platforms yet still process data
* Clients may simultaneously access resources on the file server
* Remote access to the server through a network connection is more efficient than a physical connection
* Client and server communicate with a series of get requests and replies with responses

**Reusability:**

* The software architecture of this project is based on the client-server template, which can be used in various applications that require a controlled distribution of information.
* This project is currently targeting Boston local shelters. But the client-server template and setup can be easily applied to different cities.

**Security:**

* Server data is secured, because normal members and visitors cannot edit the data (View only)
* Shelter staff are required to verify themselves before granting special privileges from IT staff to update changes (Only able to update changes on vacant beds and resources, and only able to change their shetler’s information).
* IT staff accounts are the only one that are able to view and edit information for both the website and database.

## 5.2 Details

**Client:**

Client provides the user interface to interact with the system. It communicates with web servers to process information based on user actions. Client module has the following components:

* Login: a login page that allows user to create a new account or sign in
* Catalog: a page that lists all homeless shelters within our database and allows them to search for nearby shelters
* Map: Google maps services appears when searching for nearby shelters
* Account Profile: a page that displays private account information
* Favorites: a page that displays a list of the account’s favorited shelters
* Home: a page that welcomes the users and
* About: a page that describes who the Sheltered Ones team are, our purpose,  email contact, and website description
* View the website: sends a request to view the current state of the website

**Server:**

The server processes interactions between the client and website. It also allows clients to handle data manipulation without using SQL commands to directly modify the database. The server module has the following components:

* Display website: sends the client a response to allow visibility and interactiveness of the website
* Update website: refreshes the displayed website information to match the database
* Authorization: checks if the client’s account type matches their privileges
* Verification: checks the input with the database to verify the authenticity of client actions
* Search: search the database for records

**Database:**

Stores the account information used to verify log-in credentials. Stores the information that clients would like to retrieve, such as shelter information. Data stored onto the database may be retrieved and manipulated by the server.

* Back-up: exports data to store the current state of the database into a location in memory
* Restore: imports data to recover a previous state of the database
* Date: get the date when the information was stored onto the database
* Data Entry: insert data via manual entry or import and both can lead to table creation
* Query: fetch data from existing tables
* View database: displays information stored on the database

## 5.3 Tradeoffs

**Database**:

Our project is targeting the Boston local shelters, so our plan for the database and maintenance is using Microsoft Access. It accommodates small organizations with many features, such as: table designs, data, queries, forms, reports, and modules. Using this will allow the servers and IT Staff to directly modify the stored information that they want to display on the website. The drawback to using Access is that it can only handle 200 simultaneous Access users onto the database. This limits the number of servers and IT staff we can provide service to. However, these Access features and data can be migrated to Microsoft Azure SQL for scalability, reliability, and long-term manageability. Initializing our project database with Access will allow our project to meet its deadline and demonstrate its functionality on a local scope before investing efforts into Azure for a larger scale.

**Website:**

For our project, we will be including a front-end website for users to access. The website will display information to all public users for shelter’s information, number of vacant beds and available resources. The website will have multiple components such as catalog with Shelters information, map detail, account profile etc. Users can request catalog and shelter information from the database, and the database will send the data and display on the website.

**Single Database**

Our database relies on Microsoft’s continued service. Even though the database is a single point of failure, Microsoft has a worldwide up-time of 99.98% in Q4 of 2019. Bottlenecking would be addressed by employing a cache system during the transition to Azure. Synchronization is not required for our current local scope, but may be implemented later if the scale of the project expands.

## 5.4 Database management system choices

Microsoft Access was chosen as our database for its functionality, user friendliness, and potential scalability. Our team is familiar with Access and have found that it is possible to communicate it with our website. Data implementation is easier than AWS and does not require billing fees up to our anticipated scale of the project. We plan on translating the database that we made in Access into PHPMyAdmin. We chose this due to a plugin in Wordpress.

If the project receives sufficient feedback and desire from the team to progress, our Access database will be moved to Microsoft Azure SQL. This allows us for an easy migration to a more robust and MORE scalable database. It also grants the system the inherited benefits of a cloud service. However, the query load will increase, so we would need to implement a caching resource. The transition will be performed if the projected number of combined servers and IT Staff accounts exceeds 200.

The individuals managing the database will be the IT Staff. They will be able to import and export the database to satisfy the data back-up and data recovery requirements.

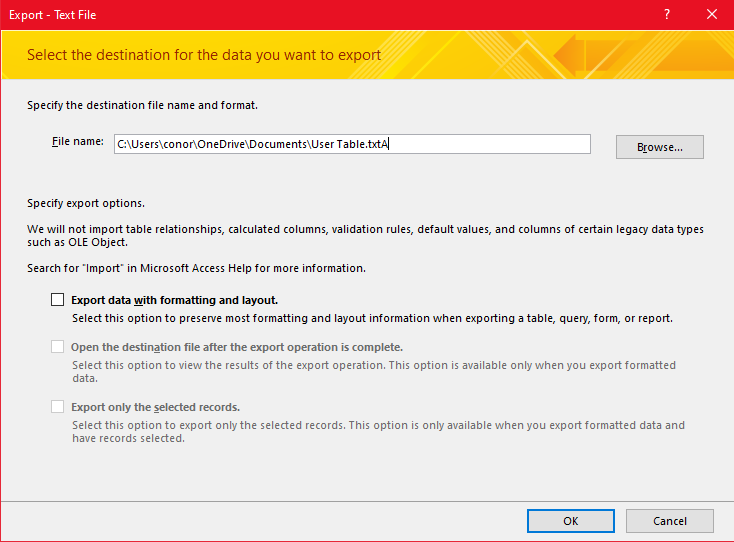
# 6. Implementation

## 6.1 User Interfaces

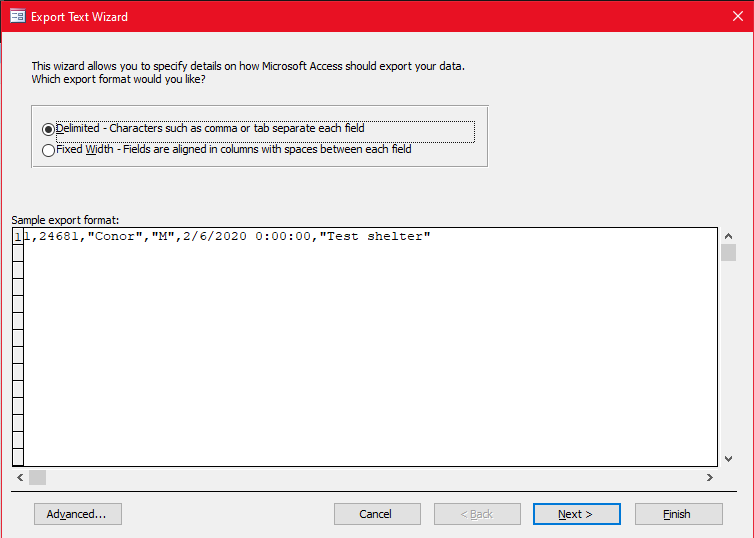
**UC-2: Database management**

**Backing-up:**

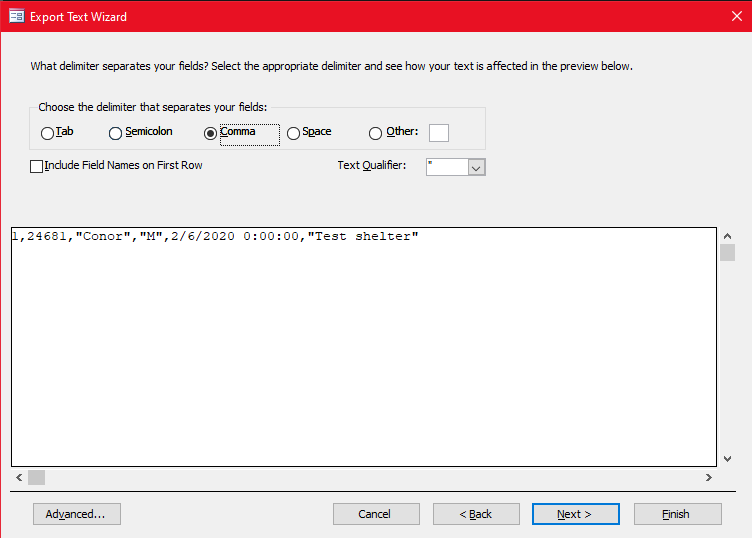
Step 1:



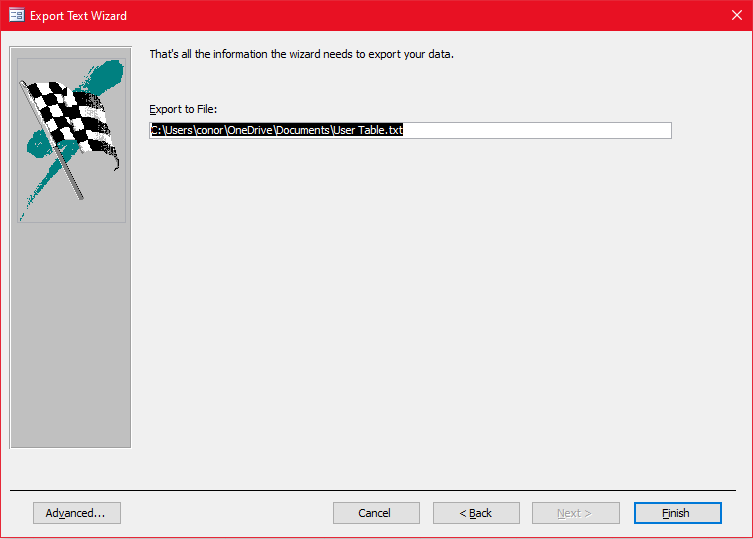
Step 2:



Step 3:

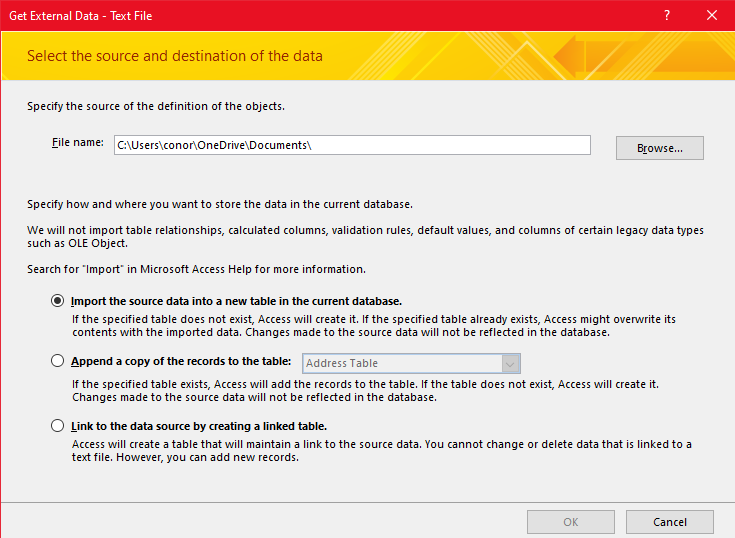


Step 4:

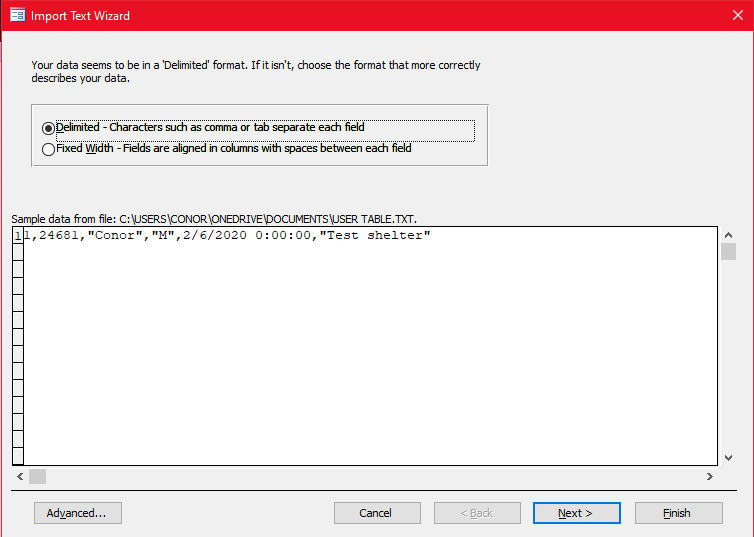


**Restoring**

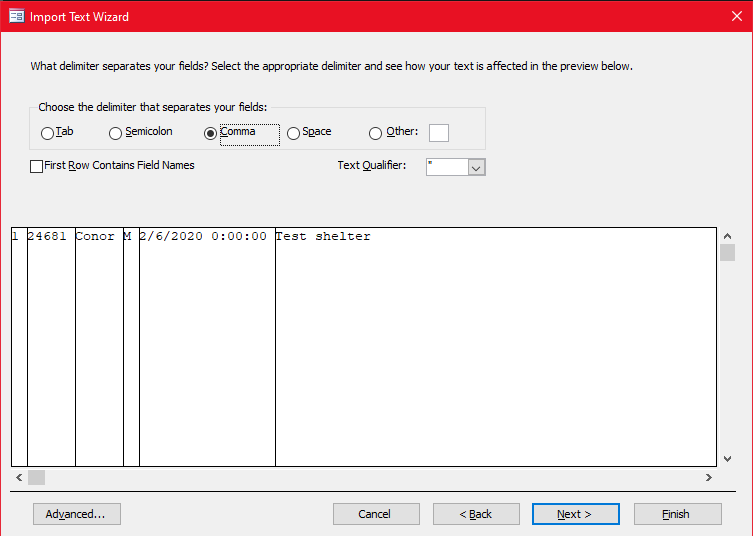
Step 1:



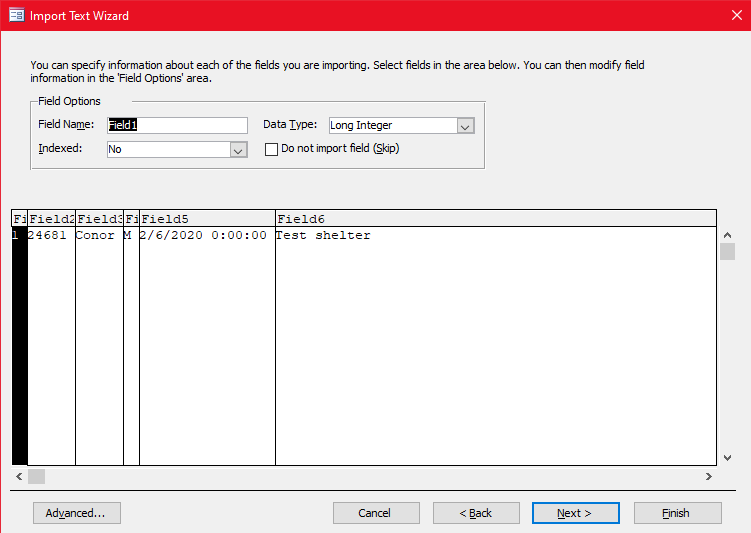
Step 2:



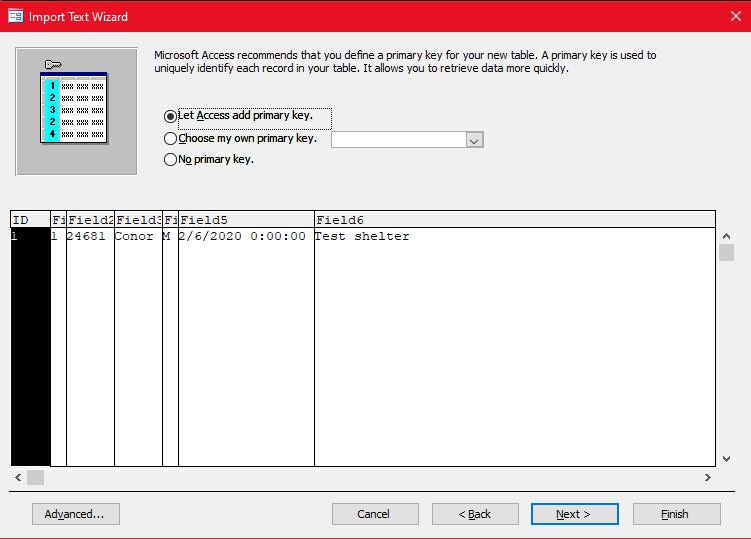
Step 3:



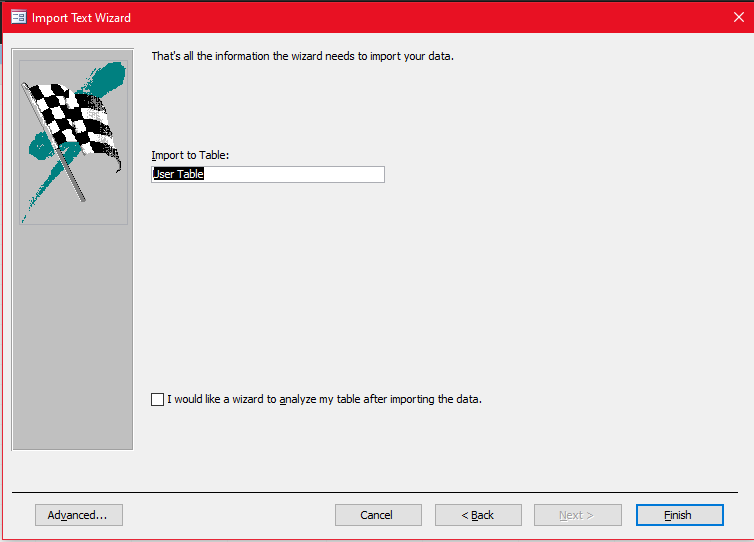
Step 4:



Step 5:



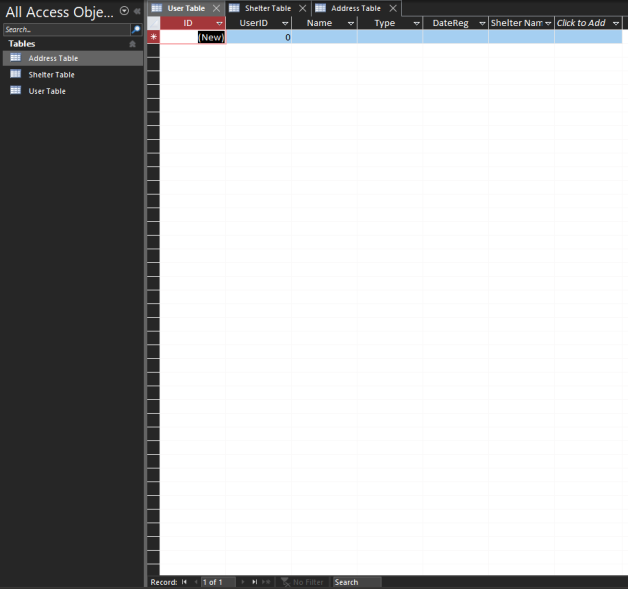
Step 6:



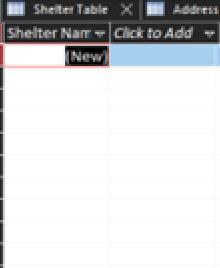
**UC-4: View the website**



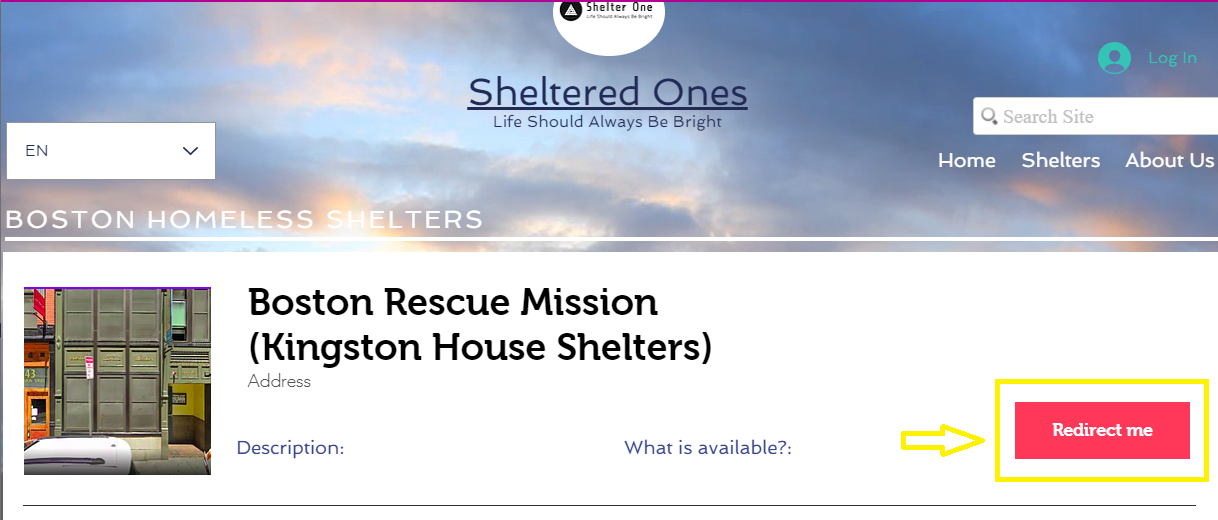
**UC-5 View the database**



**UC-6: Update shelter information**

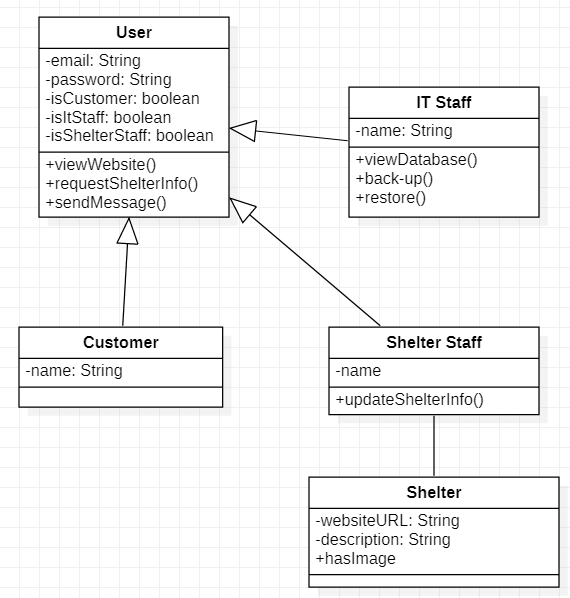
****

**UC-7 Request shelter information**



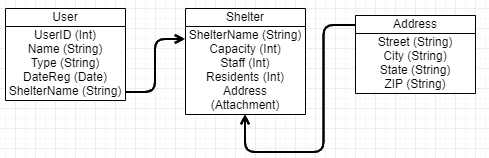


## 6.2. Class diagram



## 6.3. Database diagram

**User, Shelter, Address tables and their relationships:**



## 6.4. Others

## 6.4.1. Wix web developer toolkit

**Company website link:** <https://www.wix.com/>

## 6.4.2. ArcGIS Maps

**Company website link:** <https://www.arcgis.com/index.html>

**Boston map service link**: <https://boston.maps.arcgis.com/apps/webappviewer/index.html?id=2bb1ba286fc943caa8cb6fd16e4d03dd&extent=-7950386.8081%2C5193761.2938%2C-7877007.2609%2C5229839.5712%2C102100>