Shelter Source

Team Community Connect

Ben Cardwell, Roberto Diaz, Duyen Ho, Peter Lin

Spring 2018

Section 1

**1.1 Outline Plan**

1.1.1 Principle Tasks:

|  |  |  |
| --- | --- | --- |
|  | A2/3 | A4 |
| 1 | Determine attributes for end user to specify search | Move attributes in class diagram to single meta-class |
| 2 | Create use cases based on attributes. | Generalize use cases to eliminate personal information |
| 3 | Use use cases to create requirements | Fix public/private issues in class diagram |
| 4 | Determine user requirements | Implement correct sequence diagrams |
| 5 | Determine system requirements | Redo test cases in proper format and separate from requirements |
| 6 | Use context model to determine extent of system | Fix context diagram |
| 7 | Create class diagrams | First run at database population |
| 8 | Create sequence diagram | Implement Javascript front end |
| 9 | Review and finalize | Implement mySQL backend |
| 10 |  | Connect front and back ends for prototype implementation |

1.1.2 Milestones:

|  |  |  |
| --- | --- | --- |
|  | A2/3 | A4 |
| 1 | Attributes Determined | Revise document |
| 2 | Use Cases Finalized | Implement front end |
| 3 | Diagrams Completed | Implement back end |
| 4 | Finalize with team | Connect and finish prototype |

**1.2 Hardware and Software Requirements**

We plan on using HTML, CSS, Bootstrap, and React JS for front-end web design and mySQL for the back-end database and Google Maps API for geolocation.

**1.3 Challenges and Risks**

The most serious risk this project faces is one of coordination. Several members of the team have had scheduling and transportation issues that make it difficult to meet in person and most of the difficulties we are likely to face are related to a group of students, most of whom have never collaborated on a project on this scale, coordinating our work on several different areas of software development. We plan to address this problem by coordinating using Slack and Github in order to minimize the requirement for in-person interaction while maintaining real-time contact with all team members.

Two of the other major risks this project faces are the technical difficulties involved in connecting the front and back end of the site and getting it to run remotely in real time. We have people on our team who are skilled at both of those ends separately, but they estimate that connecting them is the area where it is most likely that we will encounter unexpected difficulties. We plan to address both of these difficulties by using robust off-the-shelf technologies that are well documented and by vigorous testing.

1.3.1 Assignment 2/3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Task** | **Effort (person-days)** | **Duration (days)** | **Dependencies** | **Assigned** |
| T1 | Determine attributes | 1 day | 1 day | None | All |
| T2 | Create Use Cases | 2 days | 3 days | None | Jen |
| T3 | Create Requirements from Use Cases | 1 day | 2 days | T2 | All |
| T4 | Determine User Requirements | 2 days | 2 days | T3 | Roberto/Jen |
| T5 | Determine System Requirements | 2 days | 2 days | T3 | Roberto/Jen |
| T6 | Create Context Model | 1 days | 1 days | None | Peter/Jen |
| T7 | Create Class Diagram | 2 day | 3 days | None | Jen/Roberto |
| T8 | Create Sequence Diagram | 1 days | 1 day | T6, T7 | Jen/Ben |
| T9 | Review and Finalize | 1 day | 1 day | T8 | All |

1.3.2 Assignment 4

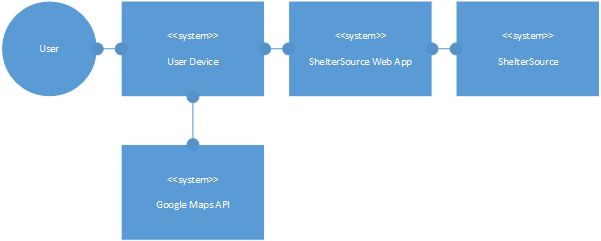
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Task** | **Effort (person-days)** | **Duration (days)** | **Dependencies** | **Assigned** |
| T1 | Refine class structure | < 1 day | < 1 day | None | Jen |
| T2 | Generalize use cases | < 1 day | < 1 day | None | Ben |
| T3 | Fix Class Diagram | < 1 day | < 1 day | T1 | Jen |
| T4 | Redo Sequence Diagrams | < 1 day | < 1 day | None | Peter |
| T5 | Redo Test Cases | < 1 day | 1 day | None | Ben |
| T6 | Fix Context Diagram | < 1 day | < 1 day | None | Ben |
| T7 | Database Population | 1 day | 1 day | None | Ben |
| T8 | Front End | 3 days | 1 week | None | Jen |
| T9 | Back End | 3 days | 1 week | None | Roberto |
| T10 | Merge | 2 days | 3 days | T8, T9 | Roberto/Peter |

Progress will be reported on Slack and will be monitored by the Team Coordinator.

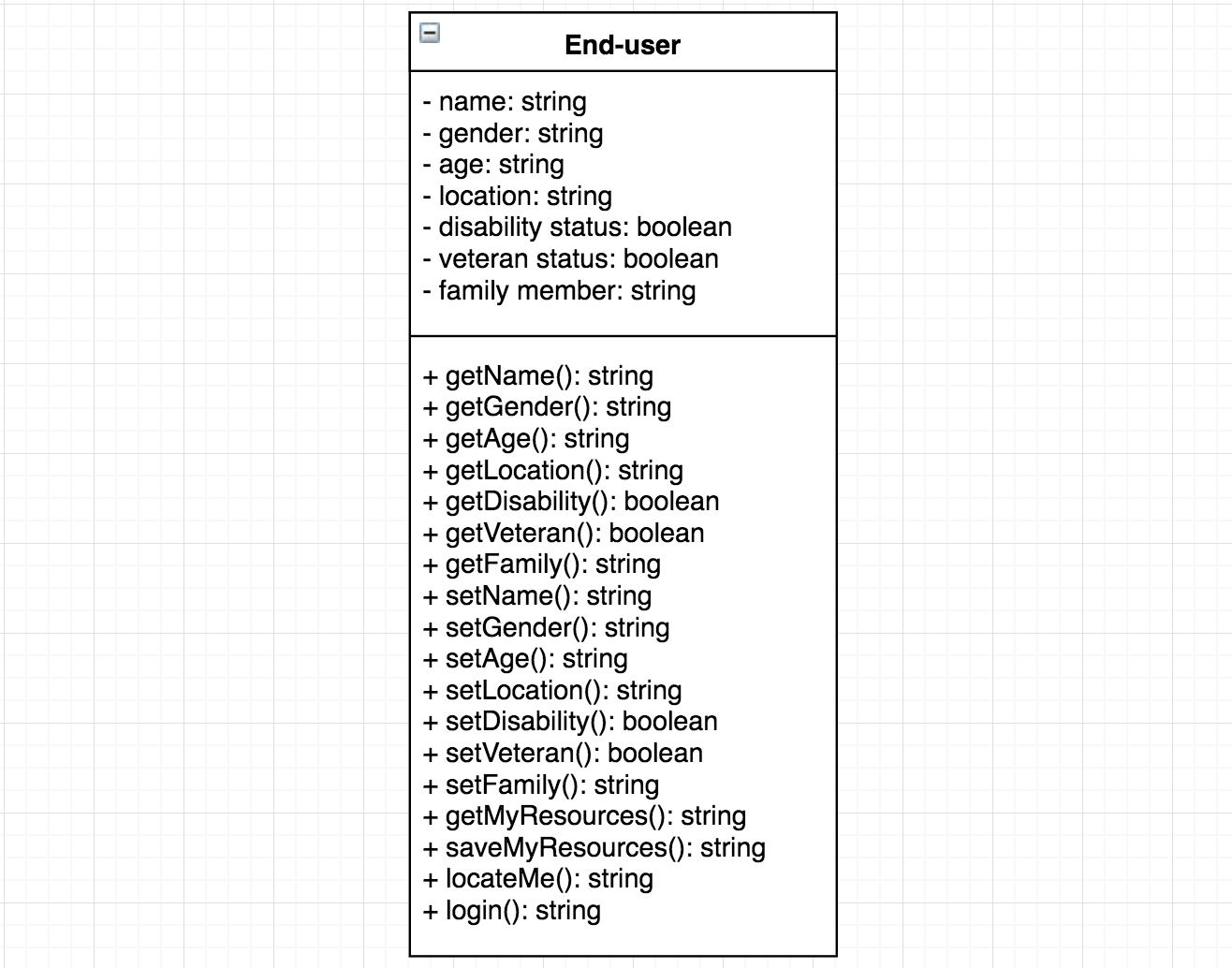
Section 2

This system implements the Layered Architecture design pattern, with the web site UI as the top layer, the javascript functions and database as the second layer, and the OS and related programs as the bottom layer.

**2.1 Context Diagram**

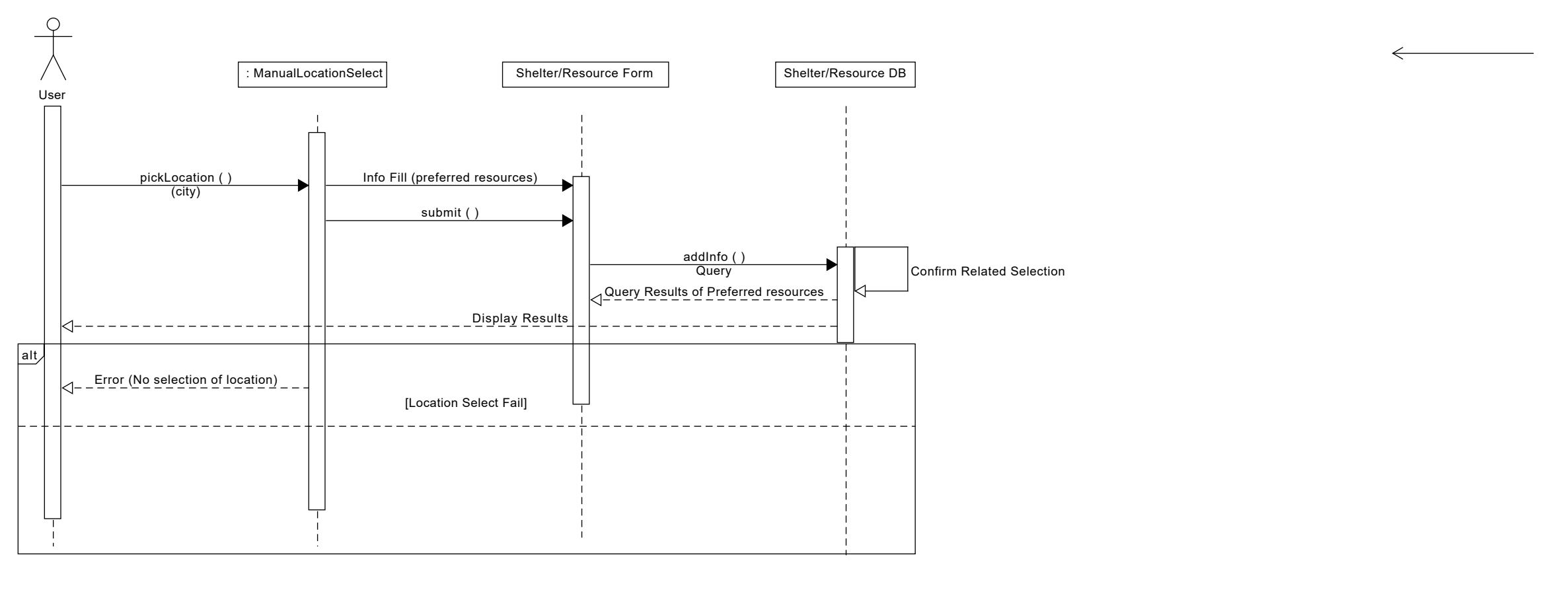
****

**2.2 Class Diagram**

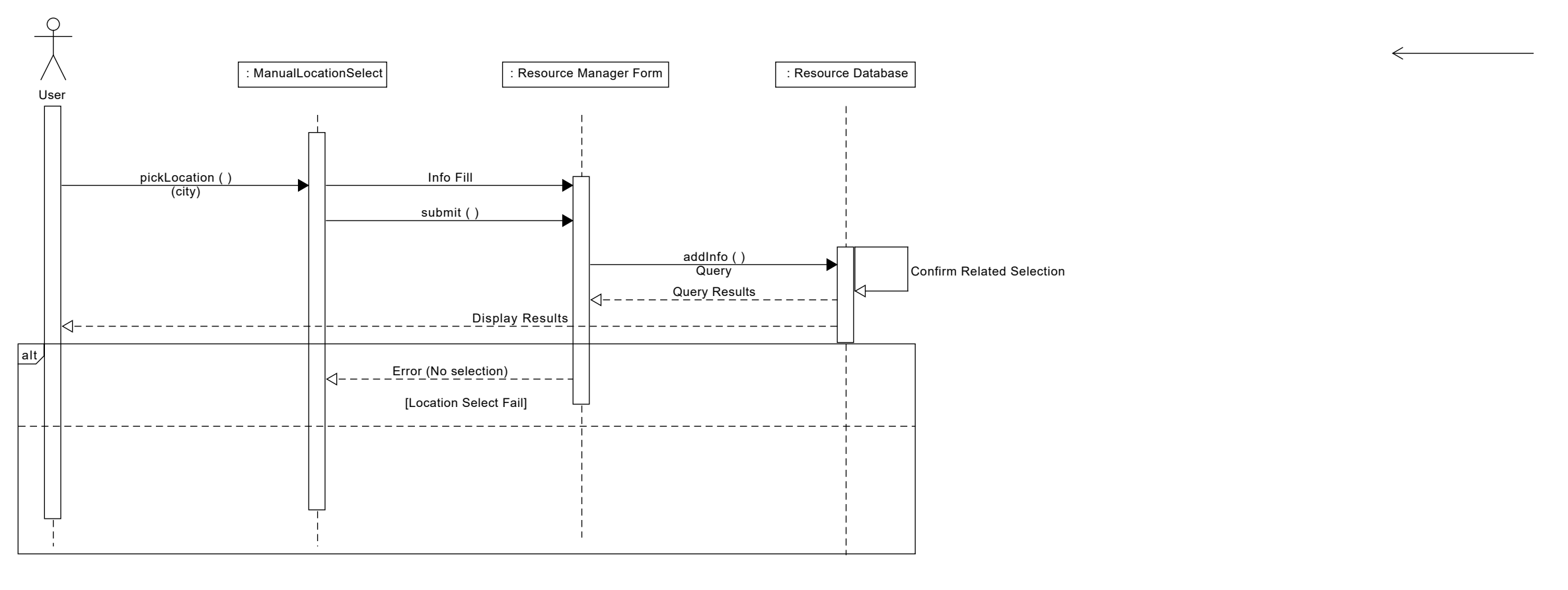
****

**2.3 Sequence Diagram**

2.3.1 Emergency Shelter use case

****

2.3.2 Shelter for Winter use case



**Section 3**

**3.1 Use Cases**

|  |
| --- |
| **3.1.1 Finding A Women Shelter** |
| **Summary:** The end user accesses the webpage to find a women’s shelter and allows Chrome to access her location. After answering a series of questions about herself via the form, the webpage generates a list of women’s shelters near her location. |
| **Basic Course of Events:**  1.       The person choose the auto locate option to let the webpage find her location.  2.       The webpage inquires the gender of the person seeking resources.  3.       The person picks her gender.  4.       The webpage inquires the age group of the person.  5.       The person picks her age group.  6.       The webpage inquires if the person is disabled or not disabled.  7.       The person picks if she is disabled or not disabled.  8.       The webpage inquires if the person is a veteran or not a veteran.  9.       The person picks if she is a veteran or not a veteran.  10.    The webpage inquires the kind of shelter and/or resource(s) they need.  11.    The person picks her preferred shelter and/or resource(s).  12.    The webpage generates a list of nearby women shelters. |
| **Alternative Paths:** In step 1, if the user choose to pick their location manually, the webpage will link to a different webpage with a drop-down menu of different locations in alphabetical order. In steps 2-11, the user has the option to go back to the previous step(s) and change their answer(s). |
| **Exception Paths**: In step 1, if the user does not auto locate or manually pick their location, they will be able to proceed. In steps 2-11, if the user does not pick an option to the question, they will not be able to proceed. |
| **Precondition:** Successfully auto locate the end user location. |
| **Postcondition:** A list of nearby women shelters. |

|  |
| --- |
| **3.1.2 Finding Donated Clothing and Shelter for the Winter** |
| **Summary:** The end user accesses the webpage but does not enable Google’s API to check their locational data. They manually picks their location and fill out an online form requesting their personal information and the webpage return a list suggesting nearby shelters and charity resources. |
| **Basic Course of Events:**  1.       The person picks their location manually.  2.       The webpage inquires the gender of the person seeking resources.  3.       The person picks their gender.  4.       The webpage inquires the age group of the person.  5.       The person picks their age group.  6.       The webpage inquires if the person is disabled or not disabled.  7.       The person picks if they are disabled or not disabled.  8.       The webpage inquires if the person is a veteran or not a veteran.  9.       The person picks if they are a veteran or not a veteran.  10.    The webpage inquires the kind of shelter and/or resource(s) they need.  11.    The person picks their preferred shelter and/or resource(s).  12.    The webpage generates a list of nearby shelters and locations of charitable organizations. |
| **Alternative Paths:** In step 1, if the computer the user is using has an auto locate feature, the user can choose to automatically locate themselves. In steps 2-11, the user has the option to go back to the previous step(s) and change their answer(s). |
| **Exception Paths**: In step 1, if the user does not auto locate or manually pick their location, they will not be able to proceed. In steps 2-11, if the user does not pick an option to the question, they will not be able to proceed. |
| **Precondition:** Successfully manually pick their location. |
| **Postcondition:** A list of nearby shelters and locations of charitable organizations. |

|  |
| --- |
| **3.1.3 Finding an Emergency Shelter** |
| **Summary:** The end user wants to find an emergency shelter but does not want to provide personal information. They enter their location and choose the option to have the webpage generate a list of resources of that category, skipping the information entry step. |
| **Basic Course of Events:**  1.       The person picks their location manually.  2.       The person picks their preferred shelter and/or resource(s).  3.       The webpage generates a list of nearby shelters and locations of charitable organizations. |
| **Alternative Paths:** In step 1, if the computer the user is using has an auto locate feature, the user can choose to automatically locate themselves. In steps 2, the user has the option to go back to the previous step(s) and change their answer(s). |
| **Exception Paths**: In step 1, if the user does not auto locate or manually pick their location, they will not be able to proceed. In steps 2, if the user does not pick an option to the question, they will not be able to proceed. |
| **Precondition:** Successfully manually pick their location. |
| **Postcondition:** A list of nearby shelters and locations of charitable organizations. |

**3.2 Requirements**

* Requirement ID: 1
* Requirement Description: The system must be able to detect the user’s location using the Google Maps API
* Rationale: Automatic location detection simplifies user experience
* Input: Location data from user device
* Persistent Change: Set user location
* Related Requirements: Req 2, Req 7
* R1C1 Detect Location
* Description: Test to see that the system can detect the user’s location
* Test Inputs: A device with geolocation enabled
* Expected Results: The system correctly selects the location of the test phone.
* Dependencies: None
* Initialization: The test phone will need to be physically located in the relevant location and will need to be moved in order to successfully test this requirement
* Test Steps:

1. Device is relocated to neighborhood being tested.
2. Device is navigated to Shelter Source web site.
3. The web app should automatically select the user’s location on the form.

* Requirement ID: 2
* Requirement Description: The system must give user the option to input location manually
  + 2.1: This option overrides the automatic location detection from Req 1 if it is selected
* Rationale: User may not be in the location they wish to search for a shelter or other resource retaliative to or may have a device that does not support automatic location detection.
* Input: Location data from dropdown
* Persistent Change: Set user location
* Related Requirements: Req 1, Req 7, Req 8
* R2C1 Manual Location Entry
* Description: Test the manual location selection functionality on the web app.
* Test Inputs: Location selected on web page.
* Expected Results: The system, when a neighborhood is selected, should return results from that area only.
* Dependencies: None
* Initialization: The web page is accessible.
* Test Steps:

1. Device is navigated to the Shelter Source web site.
2. A location is selected in the drop down.
3. The web app should set the user’s location to the selection.

* R2C2 Manual Location Entry Override
* Description: Test that the manual location selection function overrides the automatic location detection function.
* Test Inputs: A device with geolocation enabled.
* Expected Results: The manual selection of a location should override the automatic selection of a location, setting the location to the manually selected one regardless of where the device is actually located.
* Dependencies: R1C1, R1C2
* Initialization: The web page is accessible, the device is physically located in a location that the automatic location detection function recognizes.
* Test Steps:

1. Device is relocated to location within the scope of the project.
2. Device is navigated to the Shelter Source web site.
3. Verify that automatic location detection has selected the current location of the device correctly.
4. Select a different location from the drop-down menu.
5. Verify that the location registered by the web app is the one selected by the user and not the one selected by the automatic location detection function.

* Requirement ID: 3
* Requirement Description: The system must give the user the option to input relevant attributes to narrow down the search for a resource
  + 3.1: Only those attributes that are selected will be used to narrow down the search
  + 3.2: Attributes are Age, Gender, Disability, Veteran status, Family members
* Rationale: Resources such as shelters may cater to specific demographics, e.g. women’s shelters, shelters that can house families together, shelters that have accommodations for the disabled, resources that may not be of use to minors
* Input: Attributes selected on web app
* Related Requirements: Req 7
* Persistent Change: Set user attributes
* R3C1 Interface Test
* Description: Test that selection of attributes in web app correctly sets those attributes for the user.
* Test Inputs: Filled form on service web site.
* Expected Results: The attributes selected on the web site should correspond to the attributes applied to the user in the system.
* Dependencies: None
* Initialization: The web page must be accessible.
* Test Steps:

1. Device is connected to the web site.
2. Personal data is entered into the web site.
3. Verify that the system applies the selected attributes to the user.
4. This must be repeated for all attributes.

* Requirement ID: 4
* Requirement Description: The system must give the user the option to select the resource they are searching for
  + 4.1: Options are Family Shelter, Women’s Shelter, Men’s Shelter, Children’s Shelter, Emergency Shelter, Soup Kitchen, Charitable Organization
* Rationale: Users may be in need of different types of assistance
* Input: Resource selected on web app
* Related Requirements: Req 7, Req 8
* R4C1 Interface Test 2
* Description: Test that selection of resource in web app correctly sets that attribute for the user.
* Test Inputs: Filled form on service web site.
* Expected Results: The resource selected on the web site should correspond to the resource applied to the user in the system.
* Dependencies: None
* Initialization: The web page must be accessible.
* Test Steps:

1. Device is connected to the web site.
2. Resource is selected on the web site.
3. Verify that the system applies the selected resource to the user’s search.
4. This must be repeated for all resources.

* Requirement ID: 5
* Requirement Description: The system must allow user to set up a persistent account where their information is stored.
* Rationale: User may wish to use the system multiple times without reentering all their information
* Input: User name, Password
* Persistent Change: Account created with selected user name and password
* Related Requirements: Req 6
* R5C1 Account Setup
* Description: Test that the account setup process is successful and that account information is persistent.
* Test Inputs: Randomly selected user names and passwords and randomly selected combinations of attributes. Ensure that a wide spread of attributes is covered.
* Dependencies: Req 3, Req 4
* Initialization: Access the web page and input information into account creation form.
* Test Steps:

1. Access web page.
2. Input user name, password, and attributes into account creation form and submit.
3. Verify that the information is stored in the database.

* Requirement ID: 6
* Requirement Description: The system must allow the user to log into their persistent account
* Rationale: User may wish to use the system multiple times without reentering all their information
* Input: User name, Password
* Persistent Change: If user name and password entered successfully, user logs in. If not, user is kicked back to login screen.
* Related Requirements: Req 5
* R6C1 Account Access
* Description: Test that the account can be accessed by user with correct login information
* Test Inputs: User name and password of existing accounts
* Dependencies: R5C1
* Initialization: Access web page and input information into login form
* Test Steps:

1. Access web page.
2. Input valid user name and password into login form and submit.
3. Verify that the correct account has been logged into.

* R6C2 Account Security
* Description: Test that the account cannot be accessed with incorrect data.
* Test Inputs: User name and password of existing accounts
* Dependencies: R6C1
* Initialization: Access web page and input information into login form.
* Test Steps:

1. Access web page.
2. Input valid user name and invalid password into login form and submit.
3. Verify that the account corresponding to the valid user name has not been accessed.
4. Input invalid user name and valid password into login form and submit.
5. Verify that the account corresponding to the valid password has not been accessed.

* Requirement ID: 7
* Requirement Description: Once user selects attributes, resources, and location the system will output the resource of the selected type that matches the selected attributes closest to user’s location
* Rationale: This is the purpose of the system, allows user to find nearby resource that matches their needs
* Input: Attributes, resources, location
* Output: Nearest resources matching attributes
* Related Requirements: Req 1, Req 2, Req 3, Req 4
* Test Case: Test on web app, test is successful if output is the resource of selected type matching selected attributes closest to selected location
* R7C1 Output Test
* Description: Test that the system can output information matching the requested resources, attributes, and location.
* Test Inputs: Sample resources, attributes, and locations entered into the form and from user accounts.
* Dependencies: Req 1, Req 2, Req 3, Req 4, Req 5, Req 6
* Initialization: Enter resource and attributes from form and user account. Enter location data and automatic location detection data.
* Test Case:

1. Input attribute data from form or user account.
2. Input location manually or from automatic location detection.
3. Select resource.
4. Verify that information matching all those fields is displayed to user.
5. Repeat for various combinations of above fields.

* Requirement ID: 8
* Requirement Description: The user can selects a location manually and a resource type. The system must then output all resources of the relevant type within 1/2/5 miles of the selected location
* Rationale: User may not be comfortable inputting personal information, this allows them to see resources close to selected location and choose which one suits them best
* Input: Location, resource, distance
* Output: All resources of selected type within selected distance of selected location
* Related Requirements: Req 2, Req 4
* Test Case: Test on web app, test is successful if output is the resource of selected type matching selected attributes closest to selected location
* R8C1 Bulk Output
* Description: Test bulk data output system.
* Test Inputs: Location and resource selection via website.
* Dependencies: Req 2, Req 3, Req 4
* Initialization: Input selected location and resource data into form on web site.
* Test Case:

1. Input location manually in web form.
2. Input resource selection in web frorm.
3. Select Bulk Output.
4. Verify that all resources of the appropriate type in the selected location have been output.