Project Shelter Source

Team Community Connect

Spring 2018

**Outline Plan**

Principle Tasks:

1. Determine attributes for end user to specify search
2. Create use cases based on attributes.
3. Use use cases to create requirements
4. Determine user requirements
5. Determine system requirements
6. Use context model to determine extent of system
7. Create class diagrams
8. Create sequence diagram
9. Review and finalize

Milestones:

1. Attributes Determined
2. Use Cases finalized
3. Diagrams completed
4. Finalize with team

**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.

**Challenges and Risks**

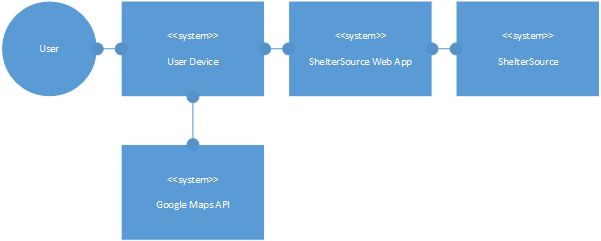
The most serious risk this project faces is one of coordination. Several members of the team have had scheduling and transportation issues that make it difficult to meet in person and most of the difficulties we are likely to face are related to a group of students, most of whom have never collaborated 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.

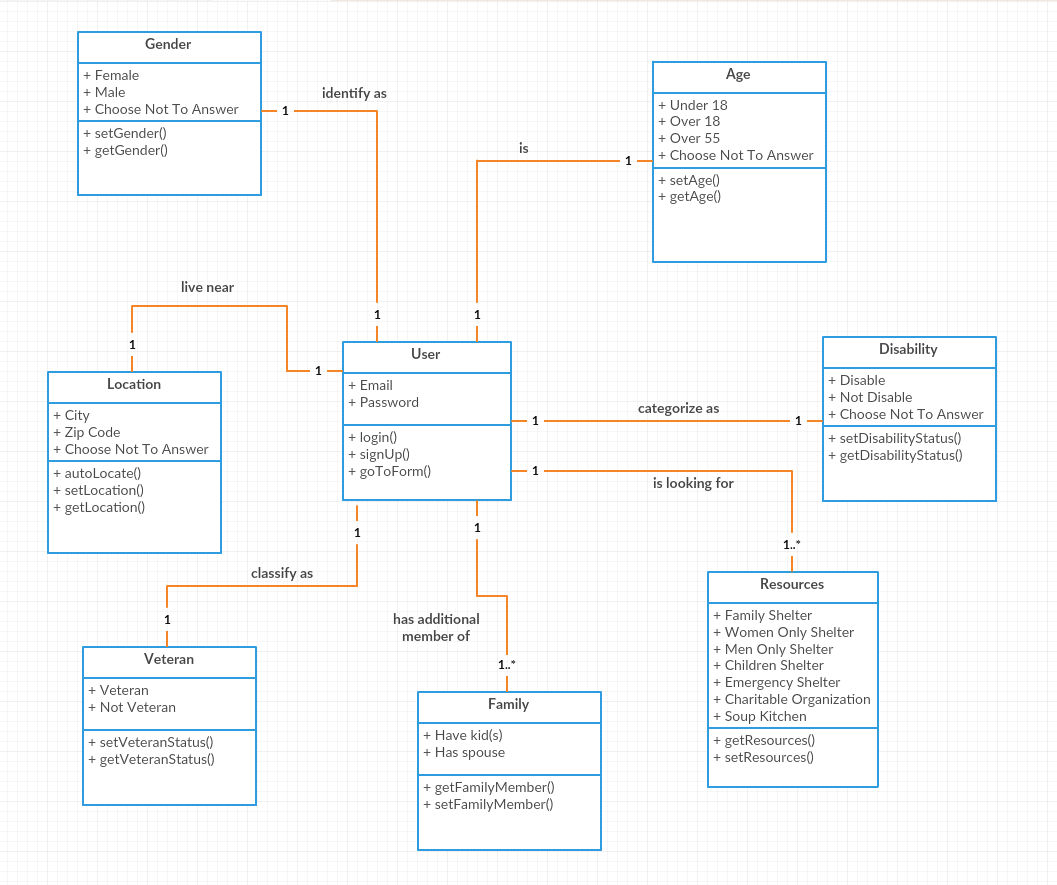
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Task** | **Effort (person-days)** | **Duration (days)** | **Dependencies** | **Assigned** |
| T1 | Determine attributes | 1 day | 1 day | None | All |
| T2 | Create Use Cases | 2 days | 3 days | None | Jen |
| T3 | Create Requirements 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 |

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

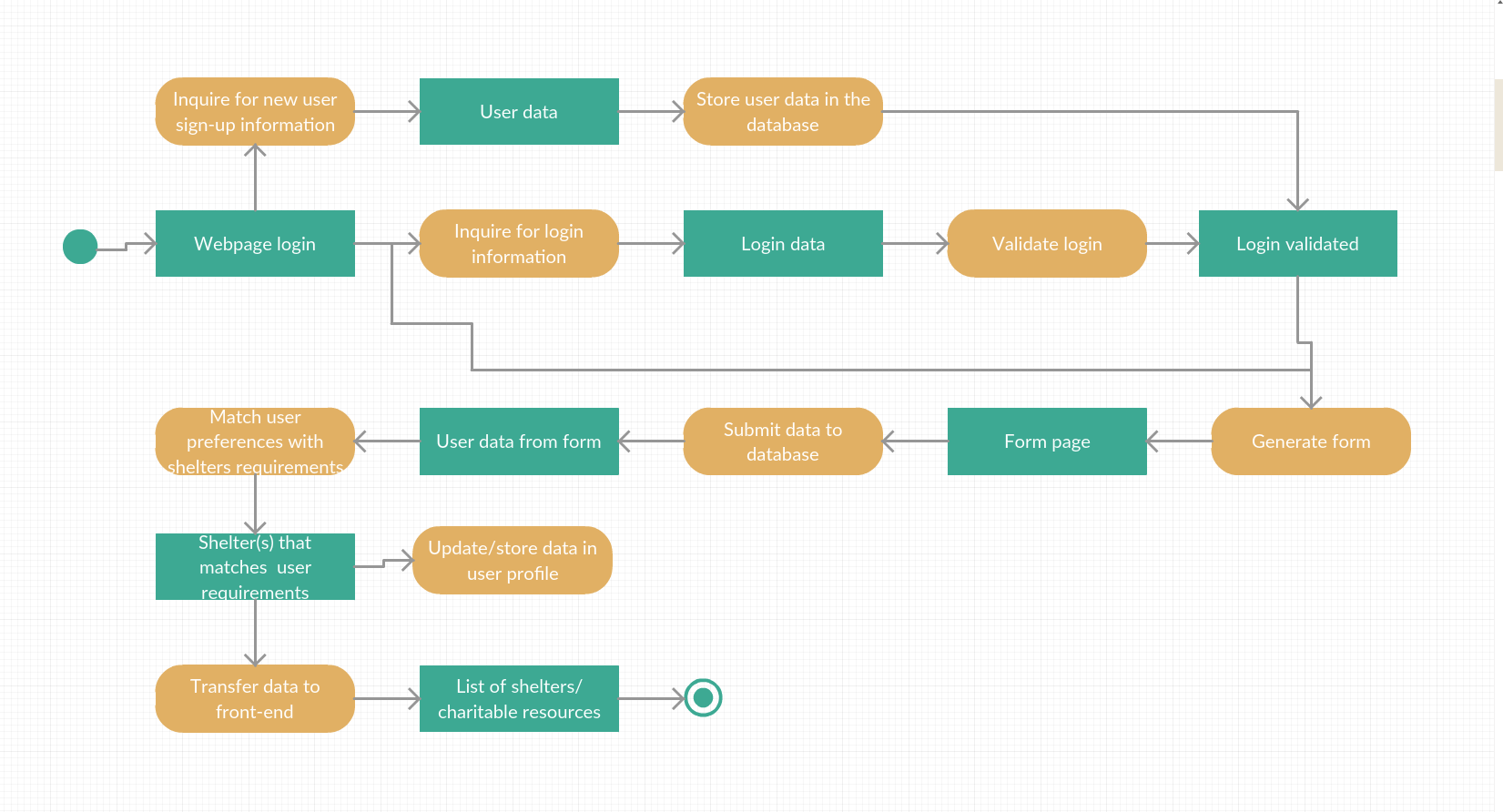
**Context Diagram**



**Class Diagram**

****

**Sequence Diagram**

****

**Use Cases**

|  |
| --- |
| **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. |

|  |
| --- |
| **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. |

|  |
| --- |
| **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. |

**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
* Test Case: Test location detection on smartphone device, test is successful if user location is set correctly based on input
* 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
* Test Case: Test using dropdown on web app, test is successful if user location is set correctly based on input
* 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
* Test Case: Test combinations in web app, test is successful if user attributes are set correctly based on input
* 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
* Test Case: Test on web app, test is successful if user is directed to correct resource
* 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
* Test Case: Test on web app, test is successful if account is set up on server
* 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
* Test Case: Test on web app, test is successful if login is allowed with correct login information and not allowed with incorrect login information
* 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
* 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