Greg Heitman

Dr. Pulimood

CSC 415-01

25 March, 2016

Assignment 4: Open Source Software – Analysis and Design (Updated)

Use Case Descriptions

*Adopting a fire hydrant (already an established part of system):*

1. The user opens the web application to view the map of Boston, Massachusetts and various hydrant icons.

2. He/she drags and manipulates the zoom of the map to search for an appropriate hydrant.

3. Upon finding a desired hydrant, the user clicks the marker to view whether it is currently owned or vacant.

4. If not already signed in, the user enters their identification credentials to log into their individual account. These credentials include an email address, password, and phone number.

5. After signing in, the user clicks on a hydrant and selects “Adopt!” to gain ownership of hydrant.

*Request ownership of a fire hydrant (functionality I am adding):*

1. The user opens the web application and selects a specific, currently owned hydrant on the map

2. The user is presented with a window providing two options: “Reminder to shovel”, and a new option named “Request Ownership”

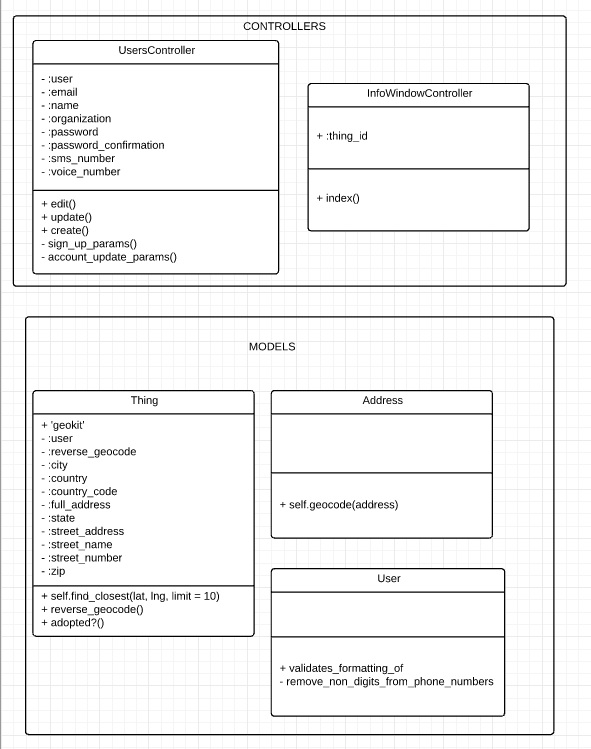
3. The user selects “Request Ownership” and sends a notification to the current hydrant owner (through email) asking for ownership of the hydrant. The user is notified of a successful transmission via a flashed message in the window.

*Shop for snow shovels and related equipment (functionality I am adding):*

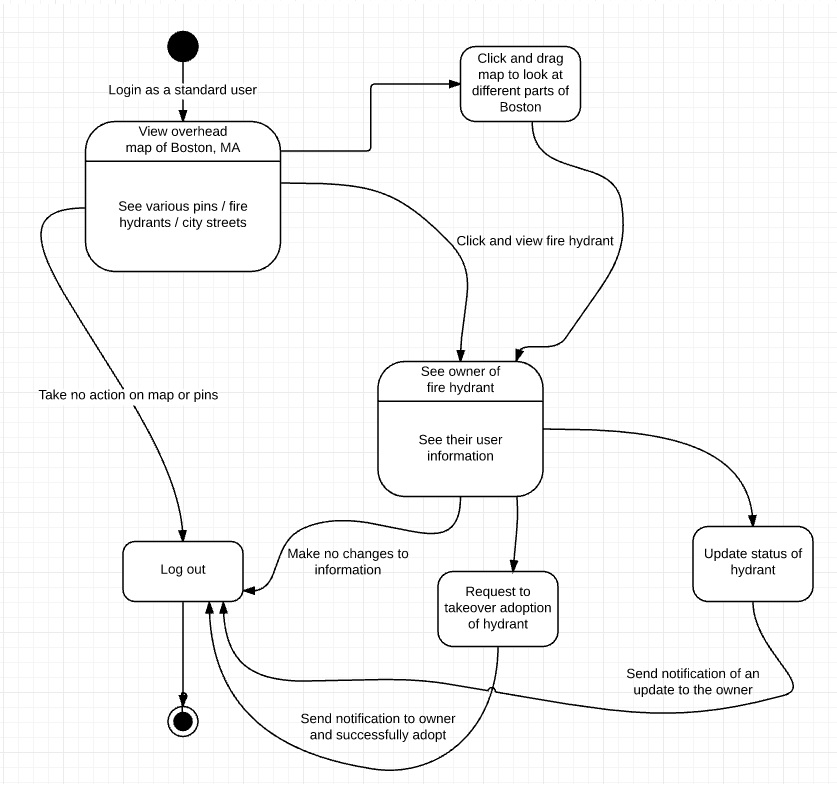
1. The user opens the web application and navigates to the sidebar on the left side of the screen

2. The user clicks on the .png image of the shovel

3. The user visits a link to the website of GRAINGER, and views different snow shovels and snow equipment that they can potentially purchase.



State Chart for System/Algorithms



Test Case Design

In testing my contributions to the open source project, I will be using both unit and integration testing. I will use unit tests to determine whether the components like user information retrieval/storage and hydrant note updates. Then, I will use a top-down integration method to examine the cohesion of these processes together, with the inclusion of interface elements like a button to edit a note, add a note, or update the overall status of a fire hydrant. The gdb debugging tool will be valuable to use, as I can step through each individual line or block of code and test its functionality.

|  |  |  |  |
| --- | --- | --- | --- |
| **Functionality Tested** | **Inputs** | **Expected Output** | **Actual Output** |
| User information and credentials properly stored | Email, name, organization, home phone, mobile, password | These fields are saved with the users’ credentials correctly input |  |
| Requesting ownership of hydrant | Clicking on hydrant, clicking request ownership button, notifying current owner | Request for ownership is delivered to current owner via email |  |
| Scrolling on the map | Using mouse wheel to zoom in and out of map instead of clicking +/- | Scrolling in zooms user in on map, and scrolling out zooms user out on the map |  |
| Adopting a hydrant (already in use) | Scroll on map to hydrant, click hydrant, select “Adopt” on vacant hydrant or “request ownership” on taken | User is notified that they have adopted hydrant successfully or that they have reached out to current owner |  |