

## Use Case #1

Use Case: Search for nearest locations  
Primary actor: User  
Goal in context: To “search” or view locations in the directory in order of proximity to location entered  
Preconditions: The directory is not empty  
Trigger: User clicks on the search button  
Scenario:  
1. User types in the address they want to perform the search for  
2. User clicks on the search button  
3. User can view locations in order of proximity to the address entered  
4. User can view preliminary information for each entry including name, a description, and its distance from the address entered.  
Exceptions:  
1. The address entered is not a valid address: the user must reenter a valid address  
Priority: Essential, must be implemented  
When available: First Increment  
Frequency of use: Multiple times per visit  
Channel to actor: Via User Interface  
Secondary Actors: None  
Channels to secondary actors: None  
Open Issues:  
1. Should we allow users to search for locations by name?

## Use Case #2

Use Case: Filter Search Results  
Primary actor: User  
Goal in context: To apply filters to search results after a search occurs  
Preconditions: None  
Trigger: User clicks on filters after a search  
Scenario:  
1. User observes the filters available  
2. User performs a search operations  
3. User clicks on filters they wish to use  
4. User can view filtered data  
Exceptions:  
1. The user clicks on multiple conflicting filters: the previous filter will be deselected  
Priority: Medium priority  
When available: Second Increment  
Frequency of use: Multiple times per visit  
Channel to actor: Via User Interface  
Secondary Actors: None

Channels to secondary actors: None

Open Issues:

1. What filters will be informative and useful for the user?

### Use Case #3

Use Case: View Entry Details

Primary actor: User

Goal in context: To view more information about a location listed in a directory

Preconditions: None

Trigger: User clicks on the 'See more details' button

Scenario:

1. User observes the entry they would like more information on
2. User clicks on the 'See more details' button
3. User is redirected to a new page for the entry
4. User can view data related to that entry

Exceptions:

- 1.

Priority: Essential, must be implemented

When available: First Increment

Frequency of use: Multiple times per visit

Channel to actor: Via User Interface

Secondary Actors: None

Channels to secondary actors: None

Open Issues:

1. How do we handle entries with incomplete information?

### Use Case #4

Use Case: Add to Directory

Primary actor: Admin/Organization/Individual

Goal in context: To add an entry to a directory

Preconditions: User has a registered account

Trigger: User clicks on the 'Create a new entry' button

Scenario:

1. User clicks on the 'Create a new entry button'
2. User is redirected to a page with a form
3. User can enter information related to the entry in the form
4. User can hit the 'Submit' button
5. The entry will be added to the user's list of entries

Exceptions:

1. User tries adding an entry for a state outside of the scope of the application

Priority: Essential, must be implemented

When available: First Increment

Frequency of use: Few times a year  
Channel to actor: Via User Interface  
Secondary Actors: None  
Channels to secondary actors: None  
Open Issues:  
1. How do we handle malicious or inadequate entries?

### **Use Case #5**

Use Case: Modify Existing Entries  
Primary actor: Admin/Organization/Individual  
Goal in context: To modify an existing entry in the directory.  
Preconditions: The user is either an Admin or is editing an entry they created  
Trigger: User clicks on 'Edit' Button  
Scenario:

1. User clicks on the 'Create a new entry button'
2. User is redirected to a page with a form
3. User can enter new information related to the entry in the form
4. User can hit the 'Submit' button
5. User can see their changes reflected in the entry

Exceptions:  
1. User enters invalid information: User is asked to reenter information in a valid format

Priority: Essential, must be implemented  
When available: First Increment  
Frequency of use: Few times a year  
Channel to actor: Via User Interface  
Secondary Actors: None  
Channels to secondary actors: None  
Open Issues:  
1. How can we handle conflicting changes being made?

### **Use Case #6**

Use Case: Verify Organizations/Entries  
Primary actor: Admin  
Goal in context: To verify entries as being legitimate and trustworthy  
Preconditions: Admin has the right privileges.  
Trigger: Admin clicks on the "Verify" Button  
Scenario:

1. Admin clicks the "Verify" button
2. Admin can now see that the entry has the text "Verified" on it
3. Admin can also verify a user, this would automatically verify all entries made by the user

Exceptions:

- 1.

Priority: Medium Priority

When available:	First Increment
Frequency of use:	Few times a week
Channel to actor:	Via User Interface
Secondary Actors:	None
Channels to secondary actors:	None

Open Issues:

1. Should the verification process logic become part of the application