Code Spec for “Bar For Me” Info System

Produced by:

Kate Kinsman

Jenna Terhar

Laura Tuck

INFO 330 Spring 2013

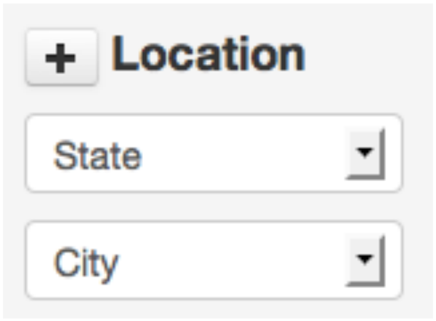
Table of Contents

Features/Functions

Search Page

getLocationHierarchy

Prototype UI



What this function does

This function retrieves the location hierarchy. Currently our hierarchy only includes states and cities but has the capability to be expanded upon. The Heirarchy is a series of drop down lists that need the parent location to be selected before the child's options can appear.

Type of access structure rendered

Hierarchy

Key Queries

1. Find the type of parent location (in this case 'State')

```
SELECT  l.idLocationType, lt.type
FROM    Location l
        JOIN LocationType lt ON l.idLocationType = lt.idLocationType
WHERE l.parentLoc IS NULL
```

2. Find the options for the previously determined location

```
SELECT  idLocation, name
FROM    Location
WHERE idLocationType = {previously returned location type}
```

3. Find the child location type of the previous location type (in this case 'City')

```
SELECT l.idLocationType, lt.type
FROM Location l
JOIN LocationType lt ON l.idLocationType = lt.idLocationType
WHERE l.parentLoc = {one of previously returned idLocation of parent}
```

4. Find the options of a child location based off the selected parent location

```
SELECT  idLocation, name
FROM    Location
WHERE parentLoc = {user selected idLocation}
```

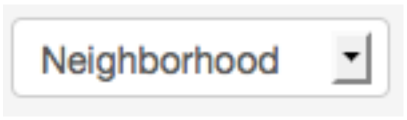
Logic

1. Query 1 finds the overall parent location this is the first drop box the following is output
   ```
   <select class="span10">
           <option>{%%LocationType.type%%}</option>
   ```
   a. Query 2 returns all the options for this type of location. The following is the output for each record returned
      ```
      <option>{%%Location.name%%}</option>
      ```
   ```
   </select>
   ```
2. Query 3 is repeated until there are no results found from the query. For each result of the queries the following is output
   ```
   <select class="span10">
           <option>{%%LocationType.type%%}</option>

   </select>
   ```
3. When the parent location of a type is selected then that drop box can inject its options related to the selected location. Query 4 finds the results of for this and the following HTML is added for each record
   ```
   <option>{%%Location.name%%}</option>
   ```
4. This process continues until all the location types are selected

getNeighborhoodForLocation

Prototype UI



What this function does

This function retrieves all the neighborhoods in location selected by the user.

Type of access structure rendered

Index

Parameters

- idLocation: the id of the location selected by the user

Key Queries

In the query below, 3 would be replaced by the idLocation of the location selected.

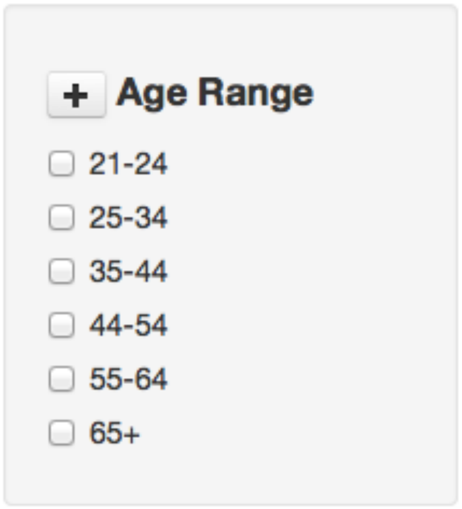
```
SELECT      n.name
FROM        Neighborhood n
JOIN        Location l ON n.idLocation = l.idLocation
WHERE       l.idLocation = 3
ORDER BY    n.name DESC
```

Logic

1. The query returns all the neighborhoods in the selected location.
2. Renders the results
   a. Output the <select class="span10"> tag
   b. Output the non-selected tag <option>Neighborhood</option>
   c. For each record returned output an option tag <option>{%%Neighborhood.name%%}</option>
   d. Output the close tag </select>

getIndex

Prototype UI



What this function does

This function outputs a list of index terms based on the parameter that is given. For instance, if getIndex:BarType is used, a list of index terms for types of bars will be displayed.

Type of access structure rendered

Index

Parameters

- indexType: the table that the index terms are stored in
    - BarType
    - DealType
    - AgeRange
    - PriceRange
    - ActivityType
    - Atmosphere

Key Queries

1. Find all BarType index terms

```
SELECT      type
FROM        BarType
ORDER BY    type
```

2. Find all DealType index terms

```
SELECT      type
FROM        DealType
ORDER BY    type
```

3. Find all AgeRange index terms

```
SELECT      range
FROM        AgeRange
ORDER BY    range
```

4. Find all PriceRange index terms

```
SELECT      range
FROM        PriceRange
ORDER BY    min
```

5. Find all ActivityType index terms

```
SELECT      type
FROM        ActivityType
ORDER BY    type
```

6. Find all Atmosphere index terms

```
SELECT      name
FROM        Atmosphere
```

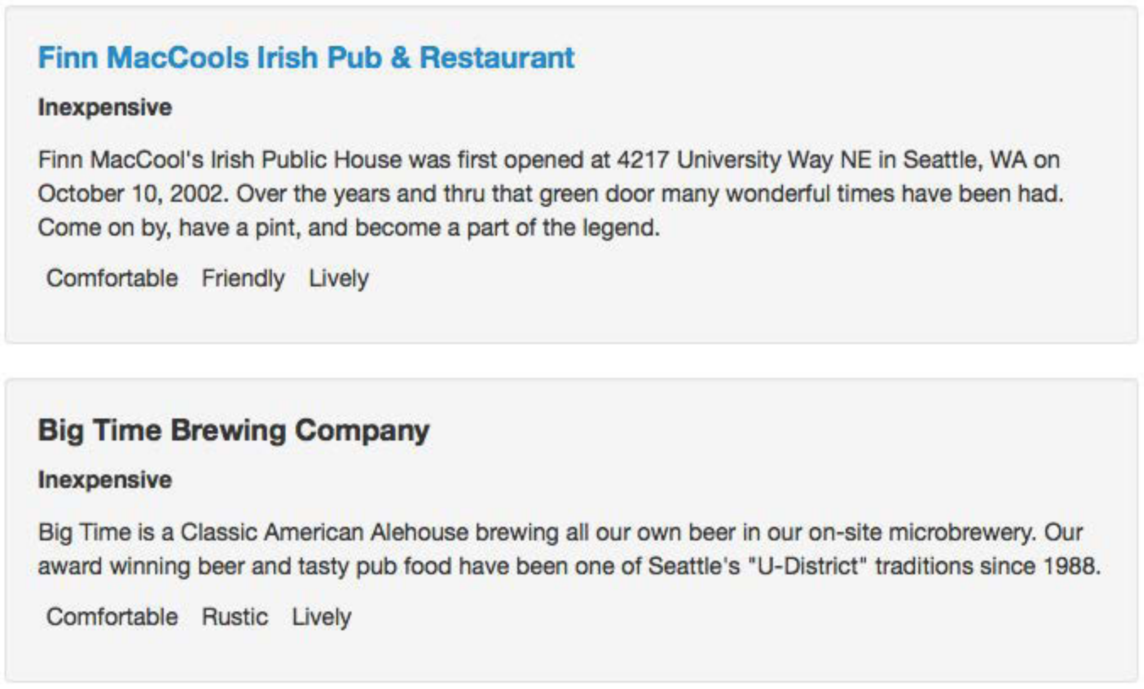
ORDER BY name

Logic

1. Queries 1-6 identify the index terms
2. For each record, renders the results as follows
   `<label class="checkbox"><input type="checkbox">{{%type | range%}}</label>`

getBarResults

Prototype UI



Finn MacCools Irish Pub & Restaurant

Inexpensive

Finn MacCool's Irish Public House was first opened at 4217 University Way NE in Seattle, WA on October 10, 2002. Over the years and thru that green door many wonderful times have been had. Come on by, have a pint, and become a part of the legend.

Comfortable Friendly Lively

Big Time Brewing Company

Inexpensive

Big Time is a Classic American Alehouse brewing all our own beer in our on-site microbrewery. Our award winning beer and tasty pub food have been one of Seattle's "U-District" traditions since 1988.

Comfortable Rustic Lively

What this function does

This function outputs a list of bar results according to the parameters specified by the user in the various form elements on the page. Bar results are ordered by alphabetical order by bar name.

Type of access structure rendered

Index

Parameters

- Location:
  - State- the state selected in the state dropdown
  - City- the city selected in the city dropdown
  - Neighborhood- the neighborhood selected in the neighborhood dropdown
- BarAtmosphere- the index terms associated with the checked boxes in the #atmosphere-search element

- BarType- the index terms associated with the checked boxes in the #type-search element
- ActivityType- the index terms associated with the checked boxes in the #activities-search element
- DealType- the index terms associated with the checked boxes in the #deals-search element
- AgeRange- the index terms associated with the checked boxes in the #age-search element
- PriceRange- the index terms associated with the checked boxes in the #price-search element

Key Queries

1. Location- param is replaced by the Neighborhood selected.

```
SELECT idBar    FROM Bar
LEFT JOIN Neighborhood        ON idNeighborhood = idNeighborhood
WHERE Neighborhood = param
```

2. Atmosphere- param is replaced by the Atmosphere selected.

```
SELECT idBar    FROM Bar
LEFT JOIN Bar_Atmosphere    ON idBar = idBar
LEFT JOIN Atmosphere        ON idAtmosphere = idAtmosphere
WHERE name = param
```

3. DealType- param is replaced by the DealType selected.

```
SELECT idBar    FROM Bar
LEFT JOIN Special       ON idBar = idBar
LEFT JOIN DealType      ON idDealType = idDealType
WHERE type = param
```

4. Day- param is replaced by the Day selected.

```
SELECT idBar    FROM Bar
LEFT JOIN Special       ON idBar = idBar
LEFT JOIN Day           ON idDay = idDay
WHERE day = param
```

5. ActivityType- param is replaced by the Activity Type selected.

```
SELECT idBar    FROM Bar
LEFT JOIN Special       ON idBar = idBar
LEFT JOIN ActivityType ON idActivityType = idActivityType
WHERE type = param
```

6. BarType- param is replaced by the BarType selected.

```
SELECT idBar    FROM Bar
LEFT JOIN BarType       ON idBar = idBar
WHERE BarType = param
```

7. PriceRange- param is replaced by the PriceRange selected.

```
SELECT idBar    FROM Bar
LEFT JOIN PriceRange ON idBar = idBar
WHERE PriceRange = param
```

8. AgeRange- param is replaced by the AgeRange selected.

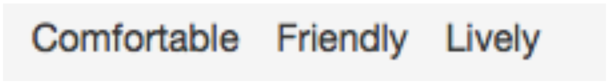
```
SELECT idBar    FROM Bar
LEFT JOIN AgeRange    ON idBar = idBar
WHERE AgeRange = param
```

Logic

1. Check if each attribute is checked. If it is, query for records using queries 1-8 based on which attribute it is.
2. Query for the ids that exist on all tables returned for each checked attribute.
3. Result is a table of records that will be looped over in the template.

getAmtosphereForBar

Prototype UI



What this function does

This function retrieves the atmospheres that are associated with a particular bar. Then the atmospheres are printed in alphabetical order.

Type of access structure rendered

Index

Parameters

- idBar: the id of the particular bar

Key Queries

In the query below, 3 would be replaced by the idBar of this particular bar.

```
SELECT          a.name
FROM            Atmosphere a
                JOIN Bar_Atmosphere ba ON a.idAtmosphere  = ba.idAtmosphere
                JOIN Bar b ON ba.idBar = b.idBar
```

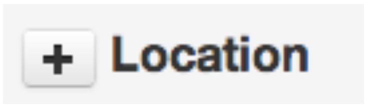
```
WHERE       idBar = 3
ORDER BY    a.name DESC
```

Logic

1. The query returns all the atmospheres related to this bar.
2. Renders the results
    a. Output the `<ul class="inline">` tag
    b. For each record returned output `<li>{%%Atmosphere.name%%}</li>`
    c. Output the close tag `</ul>`

Minimizing Filter Tabs

Prototype UI



What this function does

This function allows the user to minimize and maximize the filters on the bar search page.

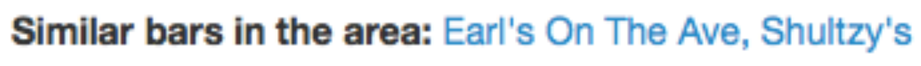
Logic

1. Pressing the plus signed button will maximize the filter options of that particular filter
2. The plus button will become a minus button and pressing that button will minimize the filter options of that particular filter.
3. Then the minus button will become a plus button again.

Bar Page

getSimilarBars

Prototype UI


Similar bars in the area: Earl's On The Ave, Shultzy's

What this function does

This function outputs a list of links to bars that are similar to the current bar the user is viewing. The similar bars are in the same neighborhood as the bar and share some of the same atmospheres or are the same type of bar.

Type of access structure rendered

Association

Parameters

- idBar: the id of the bar the user is currently viewing

Key Queries

In the queries below, 3 would be replaced by the idBar of this particular bar.

1. Find the current bar's neighborhood

```
SELECT  idNeighborhood
FROM    Bar
WHERE idBar = 3
```

2. Find the current bar's atmospheres

```
SELECT  ba.idAtmosphere
FROM    Bar b
JOIN    Bar_Atmosphere ba ON b.idBar = ba.idBar
WHERE b.idBar = 3
```

3. Find the current bar's type

```
SELECT  idBarType
FROM    Bar
WHERE idBar = 3
```

4. Use the returned attributes to find similar bars

```
SELECT  b.name
FROM    Bar b
        JOIN Bar_Atmosphere ba ON b.idBar = ba.idBar
WHERE b.idNeighborhood = {previously returned neighborhood} AND
        (ba.idAtmosphere IN ({previously returned atmospheres})
        OR b.idBarType = {previously returned bar type})
```

Logic

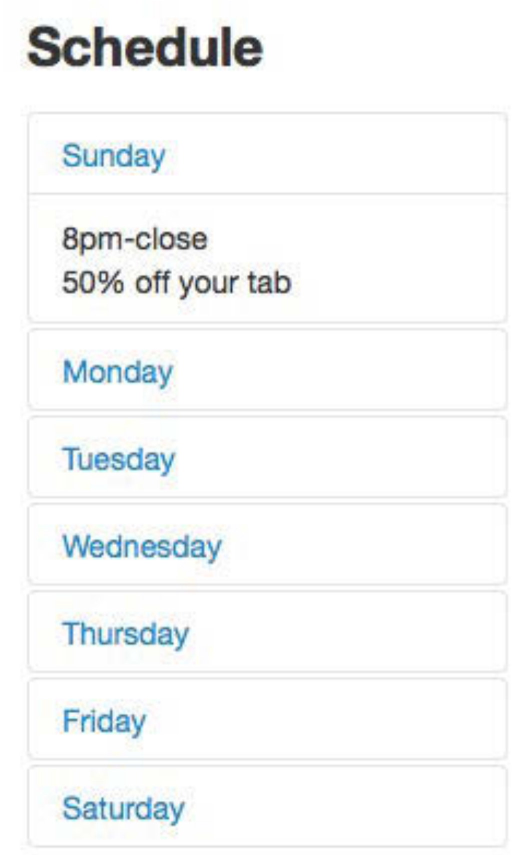
1. Queries 1-3 identify the current bar's neighborhood, type and atmospheres
2. Query 4 uses these attributes to find similar bars to the current bar
3. Renders the results as follows

```
<a href='home.php?bar={%%Bar.idBar%%}'>{%%Bar.name%%}</a>,
```

Notes: Add the common unless it is the last record returned.

displaySchedule

Prototype UI



What this function does

This function creates a schedule displaying the days of the week and the deals and/or the activities happening in that bar on that day.

Type of access structure rendered (if any)

Sequence (the days of the week ordered classically like a calendar)

Parameters

- idBar: the id of the bar the user is currently viewing
- idDay: the id of the day of the week that the for each loop is currently at

Key Queries

In the query below, 3 would be replaced by the idBar of this particular bar and 1 would be replaced by the idDay of the particular day in the loop.

Find the deals and activities that correspond to the day and bar

SELECT s.name, s.description, s.startTime, s.endTime

```
FROM Bar b
JOIN Special s ON b.idBar = s.idBar
WHERE b.idBar = 3 AND s.idDay = 1
ORDER BY s.startTime
```

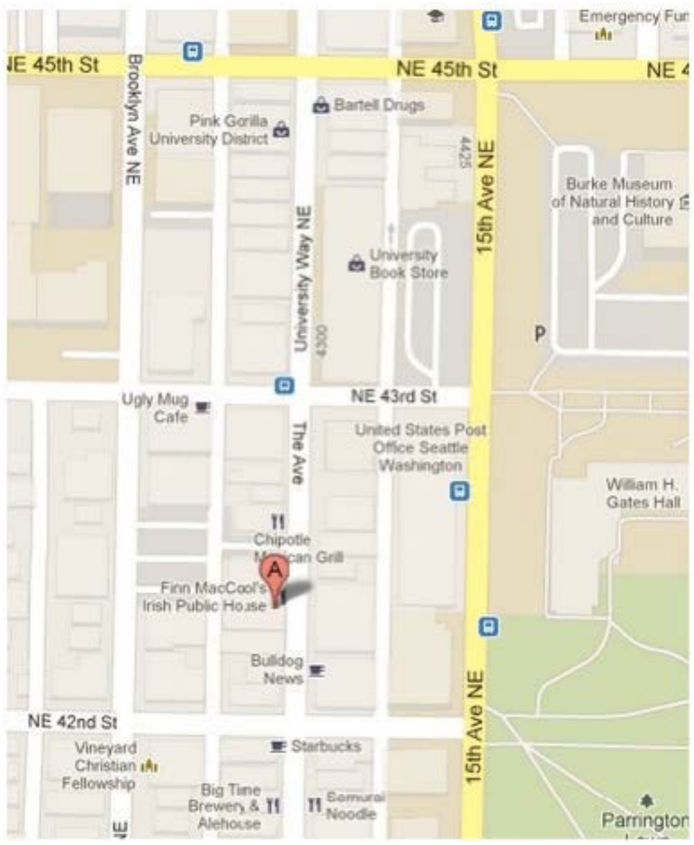
Logic

1. The template loops over each day of the week and then preforms this function in that loop.
2. The query returns all the deals and activities occurring at this bar on the particular day.
3. If the query returns 0 records then the following output is returned
   No Events Today
4. If the query returns results then one of the following is returned as output for each record
   a. If the startTime and endTime are null the following is returned
      <p>All Day: {%%Special.name%%} <br/>
   b. If the startTime and endTime are not null the following is returned
      <p>{%%Special.startTime%%}-{%%Special.endTime%%}: {%%Special.name%%} <br/>
   c. If the startTime is not null and endTime is null the following is returned
      <p>{%%Special.startTime%%}-close: {%%Special.name%%} <br/>
   d. Then if the description is not null
      <small>{%%Special.description%%}</small> </p>
   e. Otherwise the close tag is just outputted </p>
5. Finally if the day of the week can be extracted from the user then that day should be the opened accordion otherwise the first accordion should be open

Notes: The description field is formatted correctly in the database to be just outputted in the small tag.

getGoogleMap

Prototype UI



What this function does

This function displays a Google map of the bar's location.

Type of access structure rendered

Association

Parameters

- street1: the street address of the bar
- street2: (optional) the second street address of the bar
- city: the city of the bar
- state: the state of the bar
- zip: the zip code of the bar

Key Queries

None

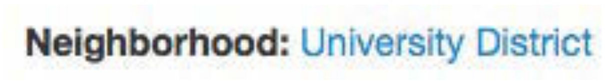
Logic

Use the Google API to return the map and render result as follows

```
<a href="{%%google.fullViewURL%%}">
        <img src="{%%google.imageURL%%}" border="0"/>
 </a>
```

getNeighborhoodName

Prototype UI



What this function does

This function determines what neighborhood a particular bar is in.

Type of access structure rendered

Association

Parameters

- idBar: the id of the bar the user is currently viewing

Key Queries

In the query below, 3 would be replaced by the idBar of this particular bar.

```
SELECT n.name
FROM Neighborhood n
JOIN Bar b ON n.idNeighborhood = b.idNeighborhood
WHERE idBar = 3
```

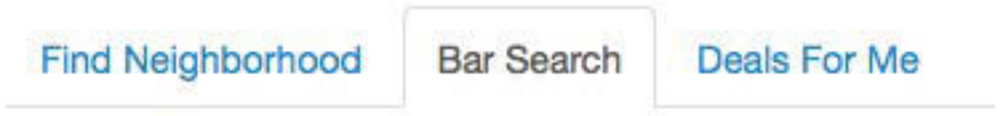
Logic

1. The query would return a single neighborhood name and render that name in the HTML provided in the template.

Shared

Global Nav

Prototype UI



What this function does

This function navigates the user to various pages in the website.

Logic

- Find Neighborhood (not included in the prototype)
- Bar Search
  - Takes the user to the landing page which displays the website description
  - Once the user begins their search the website description disappears and the search results appear
- Deals For Me (not included in the prototype

Search Bar

Prototype UI

What this function does

The search bar allows the user to jump to any specific neighborhood or bar they want to look at without having to go through the attribute-selecting process.

Type of access structure rendered

Index

Parameters

- Query: The query that the user submitted through the search bar

Key Queries

In actual implementation, would replace 'Finn's' with user input

1. Find a match to a bar

```
SELECT          idBar
FROM            Bar
WHERE           name = 'Finn's'
```

2. Find a match to a neighborhood

```
SELECT          idNeighborhood
FROM            Neighborhood
WHERE           name = 'Finn's'
```

Logic

3. First check if input is valid (not null) and sanitize it
4. Use Query 1 to see if the user's input matches any bar names in the database.
5. If it does, display the page for that bar. Using 'home.php?bar={%%Bar.idBar%%}'.

6. If it does not, use Query 2 to see if the user's input matches any neighborhood names in the database.
7. If it does, display the page for that neighborhood. Using 'home.php?hood={%%Neighborhood.idNeighborhood %%}'
8. If it does not, display 'No results found.'

Controlled Vocabularies

Types of Bars

Vocabulary

Biker

Brewery

Champagne

Club

Cocktail

Comedy

Country

Dive

Gastro-pub

Gay and Lesbian

Hookah

Hotel

Karaoke

Lounge

Neighborhood

Piano

Pubs

Sports

Taverns

Theme

Topless

Traditional

Wine

Sources

1. http://guyism.com/lifestyle/alcohol/types-of-bars-and-the-people-you-see-in-each.html#1-undefined
2. http://www.citidex.com/84.htm
3. http://www.yelp.com/c/seattle/bars

Atmosphere

Vocabulary

Chic

Classy

Comfortable

Elegant

Entertaining

Fancy

Friendly

Hipster

Lively

Modern

Noisy

Quiet

Retro

Romantic

Rustic

Trendy

Welcoming

Wild

Sources

1. http://www.score.org/system/files/u209922/Spike%20-%20Atmosphere.pdf
2. http://english.eastday.com/e/top10/u1a5436327.html
3. http://www.travelchinaguide.com/cityguides/shanghai/bar-clubs.htm
4. http://www.macmillandictionary.com/us/thesaurus-category/british/Bars-pubs-and-clubs

Price Range

Vocabulary

Inexpensive: $1-$5.99

Moderate: $6-$10.99

Expensive: $11+

Sources

1. http://www.foodservicewarehouse.com/education/pricing-alcoholic-beverages-in-your-bar-or-restaurant/c27452.aspx
2. http://answers.yahoo.com/question/index?qid=20100623180706AAnObPJ

Age Range

Vocabulary

21-24

25-34

35-44

45-54

55-64

65+

Sources

1. http://www.snapsurveys.com/blog/5-survey-demographic-question-examples/
2. http://www.knowledgenetworks.com/ganp/docs/Standard-Demographic-Variables.pdf

Deals

Vocabulary

2 for 1

Half-Tab

Happy Hour

Ladies Night

Well Specials

Sources

1. http://www.finnmaccools.com/
2. http://www.drinkowl.com/us/seattle/drink-specials

Day

Vocabulary

Sunday

Monday

Tuesday

Wednesday

Thursday

Friday

Saturday

Sources

1. http://en.wikipedia.org/wiki/Names_of_the_days_of_the_week
2. http://www.phonics.net.au/images/days-of-the-week1.jpg

Safety Rating

Vocabulary

Crime Index(0=most dangerous - 100=safest)

Sources

1. http://www.neighborhoodscout.com/wa/seattle/crime/
2. http://www.policymap.com/crime-statistics/index.html

Activity Type

Vocabulary

Beer Pong

Comedy Night

Dancing

Food Specials

Guest Speaker

Karaoke

Live Music

Trivia

Sources

1. http://www.finnmaccools.com/
2. http://www.masslive.com/entertainment/index.ssf/2011/10/top_five_club_activities_besid.html