



# Beer Advisor

*A Beer Ontology*

Anna Yaroslaski  
Lucas Standaert  
Marcelo de Castro  
Sam Stouffer



# Background

**Goal:** Our goal is to create an application that will provide users with beer recommendations by matching their preferences with attributes of commercially available beers.

**Summary:** We have been creating different types of beers for millennia and, hence, due to the wide variety of beers with different flavor profiles and styles, choosing a beer is not a simple matter. In this context, the Beer Advisor helps people in the difficult task which is finding the perfect beer. The application combines information from different databases in order to find the perfect beer match considering the user's requests.



# Background Cont.

## Requirements:

- The system must be able to differ beers between styles, alcohol content, and other key factors.
- It will return a listing of beers with all the information the system can provide on them.
- It will have an extensive catalog of beers obtained from trustworthy data sources.

## Scope:

- Limited to only beer.
- Attributes will be limited to those provided by the application's data sources.
- The application does not guarantee recommendations for all combinations.

# Stakeholders and Actors

- Customer (beer drinker).
  - Benefit directly from ontology.
  - The application will help him/her find the perfect beer match.
- Beer store and Bars.
  - Owners of places that sell beers will have a way to further expand their beer selection.
- Beer databases.
  - Open or free databases:
    - Beer.db
    - opendatasoft
- Breweries.
  - Their products will affect what enters in the databases.
- Beer distributors.
  - Beers shipped by them will affect the databases.



PRIMARY  
ACTORS

SECONDARY  
ACTORS



# Triggers and Conditions

## Triggers:

- A user launches the application and performs a search for a beer with specific set of characteristics.

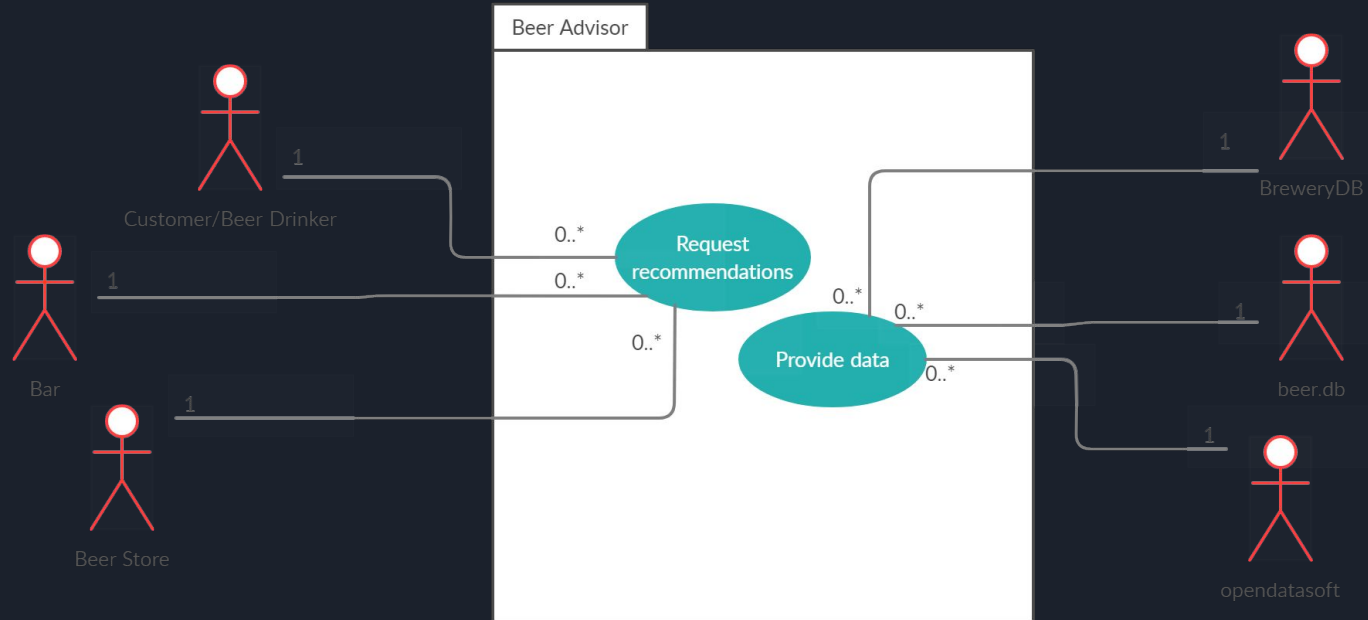
## Pre-conditions:

- Web-based application.
- Specification of characteristics wanted by the user.
- Dependent on availability of access to beer and breweries databases.

## Post-conditions:

- No information need to be saved after usage of the application.

# Use Case Diagram



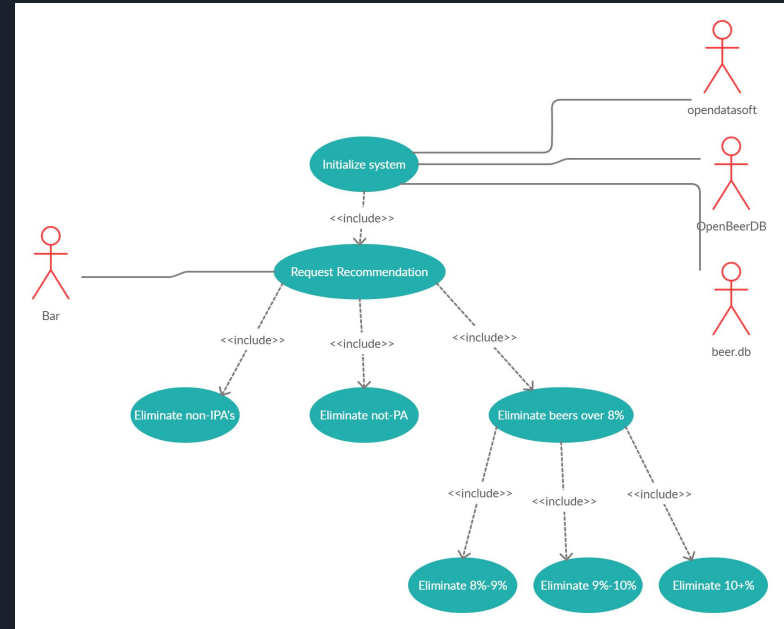


# Competency question

- **Question:**
  - **What is a brewery in Pennsylvania that makes IPA's under 8%?**
- **Sample Answer:**
  - Helltown Brewing would be an example answer, as it produces a multitude of IPA's under 8% and is located in Mt Pleasant, PA
- How do we get this answer?

# Competency Explanation

- Upon initialization obtain necessary info
- Obtain request from user
  - In our case competency question 2
- Eliminate non-IPA's
- Eliminate wrong states
- Eliminate beers over 8%
  - Stored in increments, hence extra steps







# Sources of Information

- Three main sources currently
- OpenBeerDB
  - Open database containing info on available beer
- Beer.db
  - Beer based database that is open to public on github
- Open beer database
  - Database containing beers by style, category, country, brewer, address



# What's changed

- Narrowed down the scope of the project
  - Will help in long run
- Condensed requirements
- More concrete goal
- Removed BreweryDB
  - Unsure if can attain license for database