

## CS 411 Project Template

<b>Team Name</b>	Culinary Code Squad
<b>Members Names (NetIDs)</b>	Steven Bobo (sbobo3), William Sayer (wsayer2), Manupriya Arora (manupri2), Oscar Huang (ohuang2)
<b>Email</b>	<a href="mailto:sbobo3@illinois.edu">sbobo3@illinois.edu</a> , <a href="mailto:wsayer2@illinois.edu">wsayer2@illinois.edu</a> , <a href="mailto:manupri2@illinois.edu">manupri2@illinois.edu</a> , <a href="mailto:ohuang2@illinois.edu">ohuang2@illinois.edu</a>
<b>Captain</b>	Steven Bobo (sbobo3)
<b>Project Title</b>	Flicks 'N Drinks
<b>Project Summary</b>	Using a movie database from IMDB and cocktail database from Kaggle, our database application will provide movie and cocktail recommendations to movie watchers who want to know which cocktails go well with their favorite movies, and which movies they would like based on their personality traits.
<b>Project Description</b>	<p><b>Description of application:</b> This will be a web application that prompts movie watchers to fill out personality-related questions, and then uses that information to present a list of movies that align to their matching traits. When the movie watcher selects a movie, then the application will generate a list of recommended cocktails that go well with that movie.</p> <p><b>Usefulness:</b> This application is targeted at movie watchers (of legal drinking age) who don't want to waste time shuffling through Netflix movie listings or commuting to a movie theater. It also helps to reduce time spent on deciding what kind of cocktail to enjoy while watching a movie. At the moment, there are no web sites/applications similar to this.</p> <p><b>Datasets:</b></p> <p><b><u>Movie Dataset:</u></b> Our movie dataset is from IMDB. The dataset has over 10 million records including movies, TV shows, and more. Therefore we reduced it down to under 97,968 movies, by deleting movies with adult titles, no voting, no runtime, no genre, and no actor. Our reduced dataset provides the following information for movie titles:</p> <ul style="list-style-type: none"> <li>▪ <b>tconst (string)</b> - an alphanumeric unique identifier of the title</li> <li>▪ <b>title (string)</b> - the localized title</li> <li>▪ <b>year (YYYY)</b> - represents the release year of a title.</li> <li>▪ <b>genres (string array)</b> - includes up to three genres associated with the title</li> <li>▪ <b>averageRating</b> - weighted average of all the individual user ratings</li> <li>▪ <b>numVotes</b> - number of votes the title has received</li> <li>▪ <b>directors (array of nconsts)</b> - director(s) of the given title</li> <li>▪ <b>writers (array of nconsts)</b> - writer(s) of the given title</li> <li>▪ <b>characters (string)</b> - the name of the character played</li> </ul> <p>The cast/crew dataset is also from IMDB. The people that are not involved in our organized movie dataset have been removed. The organized cast/crew dataset includes 305,511 records and provides the following information:</p> <ul style="list-style-type: none"> <li>▪ <b>nconst (string)</b> - alphanumeric unique identifier of the name/person</li> <li>▪ <b>name (string)</b> - name by which the person is most often credited</li> </ul> <p>Movie Dataset Link: <a href="https://www.imdb.com/interfaces/">https://www.imdb.com/interfaces/</a></p>

### **Cocktail Dataset:**

The cocktail dataset is a collection of over 700 cocktail recipes made up by several bartenders across the country. The original data is provided by alcohol importer and distiller Hotaling & Co. The entire dataset will be used. The columns included in the dataset are:

- **Cocktail Name (string)** - the name of the cocktail
- **Bartender(string)** - the name of the bartender who created the cocktail (optional)
- **Company (string)** - bar or company the bartender is associated with (optional)
- **Location (string)** - location of the company (optional)
- **Ingredients (string)** - Comma-separated ingredients in the form of [quantity] [unit] [ingredient\_name]
- **Garnish (string)** - comma-separated list of garnishes
- **Glassware (string)** - type of serving glass (optional)
- **Preparation (string)** - period-separated instructions for preparing cocktail
- **Notes (string)** - free-text notes about cocktail (i.e. backstory, featured listings)

Cocktail Dataset Link: <https://www.kaggle.com/shuyangli94/cocktails-hotaling-co>

### **Basic and Advanced Functions:**

#### **Basic Functions:**

- Add new user accounts
- Update the ratings of the movies and cocktails
- Like/Dislike movies, cocktails, pairings
- Delete/ Deactivate the user accounts
- Search for specific movies and cocktails
- Update user info

#### **Advanced Functions:**

- Recommend cocktails based on movies. This concept is interesting as going to the movies and having a drink is common, but the decision process can be time-consuming. This functionality will be challenging because there are thousands of different movies and cocktails to choose from, so we will need to find a way to connect the two entities based on their pairing strength.
- Recommend movie based on personality/interests. The user will provide information about their personality via a series of questions following registration. The data collected will then be used to find movies that match the user's personality traits. This functionality will be challenging since we will need to find a way to match user interests on a movie's attributes.

**Advanced Techniques:** Use 6 or more advanced techniques from the following list. (You can decide this later) **TBD**

- Indexing
- Parallel query execution
- Transaction
- Approximate query processing
- Triggers
- Partitioning\Sharding
- Stored procedure

- o Prepared Statements
- o Compound Statements
- o Constraint
- o View

## ER Design

Our ER Design has the following entity sets:

- o **Users** - stores user information after a user registers for our app
- o **Movies** - stores all the movies in our dataset
- o **People** - stores all the people that make up the directors, writers, and actors in the movies
- o **Genres** - stores all genre types that a movie can fall under
- o **Cocktails** - stores all the available cocktails in our dataset
- o **Garnishes** - stores all available garnishes that can go on cocktails
- o **Glassware** - stores the recommended glassware to use for each cocktail
- o **Ingredients** - stores the set of all unique ingredients available to make a cocktail based on the conglomerate set from all available cocktails
- o **Ingredient\_Types** - stores ingredient types, categories for which a given ingredient may fall under

Our ER Design has the following entity relations:

### MovieCocktailPairing

- **Entities:** Users, Movies, Cocktails
- **Type:** many - many
- **Description:** This relationship pairs movies to cocktails independent of user preference. It has a strength value to indicate how well the two are paired. Movies have 1 or more cocktail recommendations. Cocktails can be recommended for 1 or more movies.

### fav\_movie

- **Entities:** Users, Movies
- **Type:** many - many
- **Description:** Stores list of user's opinion on a movie (like, dislike). User likes/dislikes 0 or more movies. Movies can be liked/disliked by 0 or more users.

### fav\_cocktail

- **Entities:** Users, Cocktails
- **Type:** many - many
- **Description:** Stores list of user's opinion on a cocktail (like, dislike). User likes/dislikes 0 or more cocktails. Cocktails can be liked/disliked by 0 or more users.

### fav\_pairing

- **Entities:** Users, Movies, Cocktails
- **Type:** many - many
- **Description:** Stores list of user's opinion on whether a movie and cocktail pair well (like, dislike). User likes/dislikes 0 or more pairings. Pairings can be liked/disliked by 0 or more users.

### Connections

- **Entities:** Users, Movies, Cocktails
- **Type:** 0/more – 0/more
- **Description:** Stores list of user's friends who are also registered users on application. User can have 0 or more friends.

### Team

- **Entities:** Movies, People
- **Type:** many – many
- **Description:** Provides information on the team that helped make the movie. In the relationship, people can fall under one of three roles - director, writer, or actor. Each movie will have 1 or more people that

	<p>makeup the team and will have 1 or more people in each role. A person can also be on the team for 1 or more movies.</p> <p><b>Categorization</b></p> <ul style="list-style-type: none"> <li>▪ <b>Entities:</b> Movies, Genres</li> <li>▪ <b>Type:</b> many - many</li> <li>▪ <b>Description:</b> Provides information on the genre of a certain movie. A movie belongs to 1 or more genres and 1 or more movies can fall under the same genre.</li> </ul> <p><b>Serving</b></p> <ul style="list-style-type: none"> <li>▪ <b>Entities:</b> Cocktails, Glassware</li> <li>▪ <b>Type:</b> many - 0/1</li> <li>▪ <b>Description:</b> Provides information on the recommended glassware to use for a certain cocktail. A cocktail can have 0 or 1 recommended glasswares. A glassware can be recommended for 1 or more cocktails.</li> </ul> <p><b>Composition</b></p> <ul style="list-style-type: none"> <li>▪ Entities: Cocktails, Ingredients</li> <li>▪ Type: many - many</li> <li>▪ Description: Provides information on the ingredients used to make a certain cocktail, as well as the quantity of the ingredient and the unit of quantity. A cocktail has 1 or more ingredients. An ingredient can be used in one or more cocktails.</li> </ul> <p><b>Classification</b></p> <ul style="list-style-type: none"> <li>▪ <b>Entities:</b> Ingredients, Ingredient_Types</li> <li>▪ <b>Type:</b> many - many</li> <li>▪ <b>Description:</b> Provides information on how the ingredient can be more abstractly categorized among its ingredient peers. An ingredient can belong to 1 or more ingredient types. An ingredient type can have 1 or more ingredients assigned to its grouping.</li> </ul> <p><b>Decoration</b></p> <ul style="list-style-type: none"> <li>▪ <b>Entities:</b> Cocktails, Garnishes</li> <li>▪ <b>Type:</b> many - 0/more</li> <li>▪ <b>Description:</b> Provides information on the garnishes used to decorate a cocktail. A cocktail can have 0 or more garnishes. A garnish can be used on 1 or more cocktails.</li> </ul>
<b>Development Plan</b>	<ul style="list-style-type: none"> <li>• The relational schema of your database. Remember to include all keys and dependencies (e.g., functional dependencies) as appropriate.</li> <li>• Describe the final choice of databases and software platforms/languages that you will be using. Check the guidelines under Tools and Resources to see what tools can be used.</li> <li>• Describe where and how you will get data for your application. Do you get it from the Web, or some other application.</li> <li>• Describe the labor division among group members.</li> <li>• A project timeline with milestones.</li> </ul>
<b>System Demo URL</b>	Insert the link to the system page.
<b>Initial Demo Video</b>	<p>Insert the link to the initial demo video here</p> <p><b>Initial Demo Video Requirement:</b> each group needs to demonstrate all of the following using web page or mobile app interface connected with SQL queries. We won't accept the basic queries directly written in cPanel/SQL editor at the demo time.</p> <ul style="list-style-type: none"> <li>• Have all the data you need in your database and it should be enough to demo the basic functionality (other requirements for this stage).</li> <li>• Functionality: <ul style="list-style-type: none"> <li>o Show how to insert records to the database</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>o Show at least one query that searches the database and list or print the returned records</li> <li>o Show how to update records</li> <li>o Show how to delete records</li> <li>● <b>NOTE:</b> This should not be your page login information (insert username/delete etc). We want to see some actual records from your project updates, else you will get no credits.</li> <li>● Talk about your plan for the next stage of the project, including what advanced functions do you plan to support. Having clear idea about advanced functions is very important. Please, make sure you can articulate the challenge clearly.</li> </ul> <p>The initial demo video should be uploaded to MediaSpace at <a href="https://mediaspace.illinois.edu">https://mediaspace.illinois.edu</a> by the team captain.</p> <p>The team captain will need to log in (with your NetID and AD password). Select “Add New” and then “Media Upload” (at the top corner of the page). The team captain will be asked to fill in a form describing the uploaded video. Make sure to:</p> <ol style="list-style-type: none"> <li>1. Name the initial demo video as “Project TeamName Initial Demo”</li> <li>2. Use the tag “CS411-SPRING-2020”</li> <li>3. Add all group member names to the “team members” field (using “Add” to expand the form for additional members).</li> <li>4. Check “unlisted” as the type of upload at the bottom (which would allow others to see your video).</li> <li>5. After uploading, the initial demo video should appear under “My Media”. Make sure to test it.</li> <li>6. If the team is satisfied, click “Share” beneath the video-playing frame. This gives a shareable link to the initial demo video.</li> </ol>
<b>Project Files</b>	<p>Link to all the project files here</p> <p>Pack all your project files in a zip archive, upload the archive to one of the file hosting services (e.g. Dropbox, Box, Github), and add the link to the archive to your project page. All project teams are also required to share the repository with All TAs.</p>
<b>Final Demo Video Link</b>	<p>Link to the final demo video</p> <p>Final Demo Video Requirements:</p> <ul style="list-style-type: none"> <li>● Introduce your group's project and walk your audience through its goals and features. (We don't expect a professional movie from you, just record a quick screencast with your narration)</li> <li>● Basic Functions: <ul style="list-style-type: none"> <li>o Show how to insert/update/delete records to the database (repeat from the Initial Demo)</li> <li>o Show how to search the database and list or print returned results. You need to show a few different interesting queries over your database. <b>One of the queries must involve join of multiple (at least 2) tables.</b></li> </ul> </li> <li>● <b>Demo Two Advanced Functions:</b> Give a brief overview of your advanced functions and their link with your project. <b>You have to be able to clearly define the technical challenge in advanced functions and articulate it during the presentation.</b></li> <li>● Explain the <b>Advanced Techniques</b> used in the project: Use 5 or more advanced techniques from this list</li> </ul>

- o Indexing
- o Parallel query execution
- o Transaction
- o Approximate query processing
- o Triggers
- o Partitioning\Sharding
- o Stored procedure
- o Prepared Statements
- o Compound Statements
- o Constraint
- o View

The final demo video should be uploaded to MediaSpace at <https://mediaspace.illinois.edu> by the team captain.

The team captain will need to log in (with your NetID and AD password). Select “Add New” and then “Media Upload” (at the top corner of the page). The team captain will be asked to fill in a form describing the uploaded video. Make sure to:

1. Name the final demo video as “Project TeamName Final Demo”
2. Use the tag “CS411-SPRING-2020”
3. Add all group member names to the “team members” field (using “Add” to expand the form for additional members).
4. Check “unlisted” as the type of upload at the bottom (which would allow others to see your video).
5. After uploading, the final demo video should appear under “My Media”. Make sure to test it.
6. If the team is satisfied, click “Share” beneath the video-playing frame. This gives a shareable link to the final demo video.

