Design Alternative Analysis

Beer Me!

**Problem 1: Finding New Beers**

In the discover section of our app, we discussed various alternatives to allow users to discover new beers for them to try. One of the initial ideas we threw around was to prompt the user to fill out a simple five question form that would then generate a list of beer based on their answers. Later we decided to make a simple solution by providing the option to use sliders to choose different beer characteristics.

**Questionnaire**

*Pros:*

* More content for the user to be engaged with
* Can gather significant amounts of data

*Cons:*

* One and done usage of the applications
* If prolonged amount of questions user can get bored
* Only as good as the questions being asked. If the questions are poorly worded or they are biased then the data obtained might not give an accurate picture.
* The response rate may be poor - especially if people are busy or don't see the benefit in answering

**Sliders**

*Pros:*

* Sliders are simple and elegant in the sense that they are right in front of the user with little more work that the user must do.
* The sliders take up little room and can fit on the discovery page easily
* Sliders give you the ability to see the change immediately after adjusting it.

*Cons:*

* Sliders do not contain much content but we are banking on the fact that the user will be able to continually adjust beer characteristics for continuous amounts of intriguement.

**Current Choice:** Sliders

**Effectiveness:** TBA

**Problem 2: CSV File vs TXT file vs Database**

When we were first deciding on how we were going to store all of our data about beers, we debated about what the best method to do it would be. We knew that we would need some sort of way to be able to constantly and consistently access data. One way that we had experience with was to simply list all information about one beer on one line in a text file and separate fields with a tab. Another idea was to create a database that could be accessed through the JDBC API provided by Java. This would allow us to make SQL statements in-line with our Java code. Our last idea and the one we are thinking about sticking with is to create a simple CSV file, where each beer has its own line in the file, beer attributes are separated with commas, and different beers are separated with a line break.

**CSV file**

*Pros:*

* Java provides built in packages that allow direct import from a CSV to a table
* This would greatly simplify loading the beers into memory so that we do not have to constantly read from the disk, which is comparatively slow.

*Cons:*

* We need to learn how to implement this method from the class that it is in as well as learn how to use that class to access data and search based on columns.
* Slightly more complicated than a TXT file.
* Need to come up with a set way to order beer attributes. This order must be consistent between all beers.

**TXT file**

*Pros:*

* Simplest method of implementation of storage

*Cons:*

* We would need to create our own methods and data structure to read from the TXT file into a table of sorts as well as read from and search the table.
* Java does not provide a way to load a TXT file directly to a table as there is no formatting in a TXT file.

**Database**

*Pros:*

* Professional, would not take up too much space as there would only be one table and the record number is not very large.
* Likely the fastest of all methods, assuming the database is kept in memory (as it is not very small).
* Methods already optimized through SQL.

*Cons:*

* We would need to learn the JDBC API to use this effectively.
* This method would take a lot of time.
* Using a database in this situation is not entirely appropriate as we are not creating a large scale application. If we were to implement different user profiles that people could log in to from anywhere in the world and have a database of every beer in the world, it would make more sense.
* Relational nature of SQL would be wasted because of the small scale we are going for at this point.

**Current Choice:** CSV file

**Effectiveness:** TBA

**Problem 3: Personal Profile vs Non-savable progress**

In the Profile section of our app, we discussed various alternatives to allow users to save and store their progress throughout the apps’ usage. Users would be able to like and dislike certain beers so that when they re-opened the app they wouldnt be presented with a dislikable beer as a recommendation. The alternative to this would be an application with non-savable progress, which in the end we realized would not be ideal for a personal in-app experience with the user.

**Personal user-profile**

*Pros:*

* Personal experience for the user
* User can save Beers in their on tap section of the application
* Users can give feedback that rolls over into their next app session
* More accurate beer recommendations the longer the app is used
* On-Tap page can remind user of beers they would like to try

*Cons:*

* Must have a file storage system to save all of the user information and beer preferences
* A slight learning curve to be able to navigate the app
* Added complexity with On-Tap page and the need for a file storage manager

**Non-savable user profile**

*Pros:*

* Open and use the application right away, with little barriers to beer display
* No need for database to hold and maintain user information
* Easier, simple, and more intuitive. No learning curve

*Cons:*

* User will be unable to see what beers that they previously wanted to try
* Beer preferences will not be able to be accessed from any other computer but the one they are saved on.
* Less accurate recommendations
* Less personal. More of a utility than a profile

**Current Choice:** Personal Profile

**Effectiveness:** TBA