BARISTA 2000

Top-Level Design

Joshua Gan (jgan)

February 23, 2007

Project Description

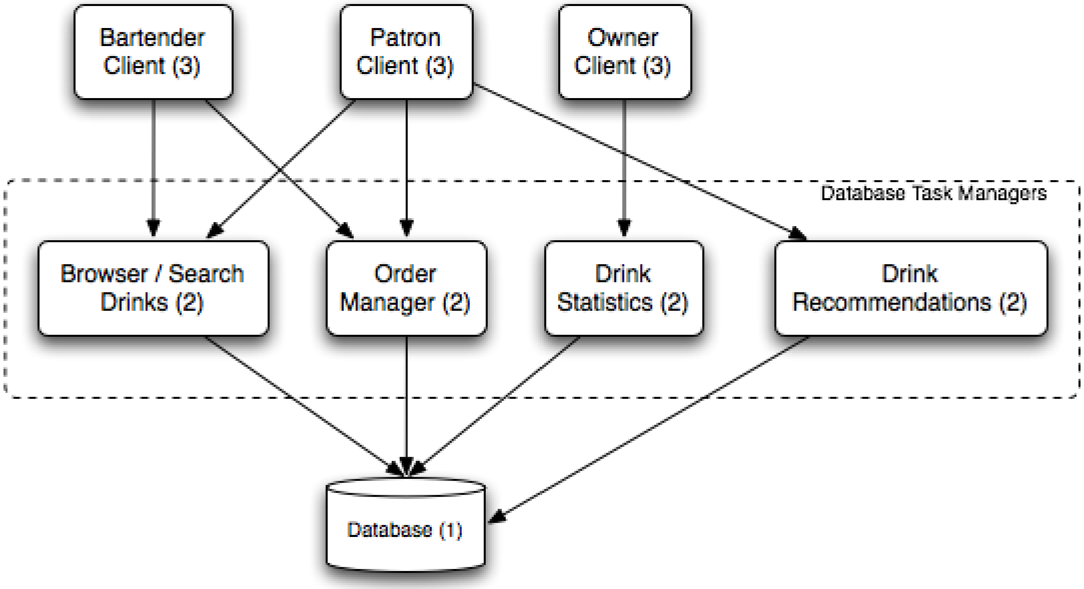
(Stolen from Tara Olson's Specification Documentation)

This would provide services for both bar patrons, as well as bartenders. Customers are presented with an easy to navigate, classy GUI that will allow them to quickly find new drinks and popular to try. Generally when people go to bars that are not attached to a restaurant they do not order new drinks, because they don't know about that many drinks. The idea of this software would be to increase the number of drinks one has to choose from when at a bar. The drinks would be categorized by types of drinks, i.e. beer, cocktails, shooters, mixed drinks. The navigation system would be based on trying to maximize quality drink suggestions with a minimal amount of extra info on the screen. Thus the drink selection for each category would be limited to a 10 or so favorites per screen. If the customer is trying to order something more specific that is not in the specials, then they click a button requesting a bartender, which will initiate a light behind the counter signifying visually to the bartender who needs to be served.

The other part of the system would be for use of the bartender. It would keep track of a ton of different drinks, so that when the bartender goes to fill an order they select the drink and the ingredient list pops up. This will provide greater consistency amongst the bartenders and allow bars to have specials that change regularly, without forcing the bartenders to know all the drinks. Other benefits of having the software for the bartender would be being able to analyze which drinks are most frequently ordered so that the bar can more accurately purchase alcohol and tailor specials towards the taste of the area/ time of year.

High Level Components

Component Diagram



Description of Components

GUI

- **Patron Client**

  This is the interface that will be used by bar patrons. It allows the user to search for drinks and add it to their tab. There should also be an intuitive method for browsing and searching through the drink database in an intuitive manner.

- **Bartender Client**

  Interface used by bartenders to see customer orders. The orders will appear in a list as they come in from the customer machines. Bartenders can see the list of all ordered drinks and read appropriate recipes. The bartender should also be able to select orders from particular tables/nodes. The bartender should be able to create a new order similarly to the customers.

- **Owner Client**

  An interface for the owner to access various statistics concerning the drinks. Users can easily find statistics for common questions (i.e. How many units of a particular drink were sold? What drink components were the most popular and need to be restocked?). It should also allow power users with a greater understanding of the underlying database to ask more complex questions and potentially arbitrary queries. Optionally, it could enable searching for statistics not immediately related to stock, like a specific bartender's average turnaround time for orders.

DB Interface

- **Browse / Search Drinks**

  Responsible for handling requests concerning drink information. Exposes interfaces for parameterized searching as well as structured browsing.

- **Order Manager**

  This component will manage the patron's orders and the bartender's production queue. Would also contain currently option features like a tab manager and payment system.

- **Drink Recommendations**

  A system which recommends drinks to users. The actual algorithm could range from extremely simple (i.e. a pre-compiled list for each drink) to relatively complex (i.e. maintaining session profiles for each user and drawing statistical inferences).

- **Drink Statistics**

  The interface for accessing various statistics stored in the database. Should be well prepared for common statistical queries, but should also be capable of executing non-standard (if not arbitrary) queries.

Database

A centralized database for all drink information.

Task breakdown

1. *General Manager*

   This person is responsible for keeping the project on schedule. The GM is also the teams mouth piece and representative. The GM must stand both in the crows nest and at the helm of the ship. Constantly vigilant for troubled waters and ready to steer a true course for our metaphorical ship. Furthermore, the GM is constantly playing a balancing act between his inner Kirk, the desire get his hands dirty and personally wrestle the giant lizard alien, with his inner Picard, knowing the value of his staff's advice (particularly the android).

2. *GUI Design / Implementation*

   This person/team is responsible for GUI design and implementation.

3. *Database Design / Implementation (team)*

   Responsible for designing and implementing the database. Additionally, this team will be responsible for the database interface components. This task is designated for a “team” because this is a significant portion of the project. This position will require a great deal of collaboration with the other members of the team since every other component is dependent on the database for their information.

4. *Networking / Communication Management*

   This design includes communication between remote clients and a central server. Someone should be responsible designing the format of these communications. This person will have to coordinate with all the separate database task managers (DTM). (Note: the networking element is not explicitly expressed in the design diagram, but it is implicitly located between the clients and the DTMs or the DTMs and the database.)

5. *Recommendations AI*

   Responsible for designing the algorithm used to recommend drinks to customers. Will likely implement multiple designs of increasing complexity.

6. *Statistics Designer*

   This person will determine what statistics an owner might find useful. The SD will work closely with the database designer.

7. *Drink "Organizer"*

   This person will take the lead in designing an intuitive but appropriately detailed organizational structure for the available drinks. This might require some research. As an added bonus, The DO will also be responsible for drink information data entry into the system.

8. *Testing Whip*

   The testing gestapo. The TW will be responsible for the global state of the projects test suite. While everyone will be responsible for their own unit tests, the TW will check that everyone is keeping up. Additionally, the TW will be primary leader of regression testing.

9. *Documentation Whip*

   For any decently sized project, documentation is a major concern. The DW is responsible for the creation of any user manual or tutorial. Additionally, this person must check that other people are maintaining appropriate internal documentation (code comments, javadocs, etc).

Note: Some of these tasks will require a team of people working together. GUI design, database design, and AI are examples of this. Conversely, not all of these positions require the sole attention of a team member. Some of these jobs should be taken up by people who are responsible for central tasks. For instance the Drink Organizer could also be on the databases team, specifically focusing on the Search / Browser interface or alternatively the Browsing GUI.

Schedule

Now – 3/2	Agree on group organization; Discuss design
3/2 – 3/9	Finalize design; First draft of interfaces
3/9 – 3/16	Comment on and update interfaces
3/17 – 3/23	Finalize Interfaces; Psuedocode and algorithms
3/24 – 4/8	Coding begins; Manager and Whips begin vigilance
4/9 – 4/16	Group meeting for mid-project evaluation; Initial integration
4/17 – 4/19	Full Integration / Feature Freeze
4/20 – 4/22	“... and Ruth said, 'For yay it is the 12th weekend of the semester in the season of rebirth. Let no work be done, and may our sacrifice of countless brain cells be deemed worthy by the gods of academia!' And the people cried out in revelry, bards of great renown did play, and the debauchery was great...” [Spring Weekend]
4/21 – 4/25	Bug fixing and component finalizing
4/25 – 4/27	Extensive testing of system (should be no different than previous 5 weeks) and bug fixing
4/27	In class demo
4/27 – 5/7	Final system development; Complete Documentation
5/7	Public Demo
5/7 – 5/17	Spit polish; Final documentation proof-read / publishing
5/17	Final Product Demo / Release

Assumptions

The specification documentation doesn't have any provision for adding new drinks to the system. I would assume that the system starts with an extensive list of drinks, but a bartender would probably want to add personal specialties. Following that reasoning, a patron might want to peruse the specialties of the bartender currently behind the bar. I have assumed that there are a number of features that weren't discussed fully in the specification that we might find are a higher priority than originally expected. As such, the design I have described here is extremely top-level leaving substantial leeway to accommodate these future features.