Barket Project Proposal

2/5/2018

Team 1

Kate Strombom – kstrombom@ku.edu

Denis Sehic – denissehic21@gmail.com

Hanwen Cao – hanwencao@ku.edu

Project Description

The premise of our project is to treat bar related items as stock in a mini-stock market. This means that the pricing of each item is dynamic and will increase or decrease based on the amount of stock being purchased of items in the "market". This idea has two underlying opportunities. The first is to capitalize on the novelty of the idea in the United States market by attracting patrons to the bar utilizing this software. We believe that the new format of purchasing items will become popular for its novelty and game-like purchasing. We also will capitalize on an opportunity to improve a restaurant or bars sale of stock. By having the pricing be dynamic, restaurants can begin to sell more items that would normally not get sold often. They can also heavily capitalize on the popular drinks sold.

Our project will be a locally hosted web application. The backend will consist of the algorithm, which will take multiple input criteria to build a customer designed model for pricing. Based on sales made in a POS system, the algorithm will perform calculations within a configurable time period and present the pricing changes on the customer friendly page. The frontend will consist of a clean user interface for presenting the items being sold similar to that of a stock market page. The bar owner will also be able to configure the look of the user interface to create a look like the bar's style.

Project Milestones

MILESTONE	DATE TO BE COMPLETED
DIAGRAMS – WIREFRAME, USE-CASE, OBJECT, ETC. **ALGORITHM DESIGN**	November 1, 2017
INITIAL PROJECT SET UP	December 1, 2017
ALGORITHM IMPLEMENTATION	February 18, 2018
CONFIG EXECUTABLE	February 4, 2018
CONNECT THE WEB AND SERVICE	March 11, 2018
WEB	March 15, 2018
TESTING AND TWEAKING	April 8, 2018
SIMPLE DEPLOYMENT	April 23, 2018

Project Budget

We intend to utilize open source software and languages for the creation of this project so that the cost involved will only be labor related. The below are the backend, front-end, and database technologies we have chosen to work with so far. They are all free but will require time to become accustomed to because they are mostly new technologies. We do not foresee Node.js and PostgreSQL requiring much learning time though based on prior experiences. All these technologies will be required as soon as the design milestone has been finished.

- Github
- Node.js
- PostgreSQL
- Express.js
- Angular

Much later, near April 2018, depending on our progress, we may want to deploy the project from our localhost to a server that would likely be utilized in the restaurant to test in a more realistic environment. We intend to find a free server from KU's student resources to test with if this is the case.

GitHub

https://github.com/kstrombom/stock-restaurant

Work Plan

Milestones

Tasks	Strombom	Cao	Sehic
Designs and Diagrams	X	X	X
Initial project set up	X	X	X
Database	X	X	
Algorithm			X
Algorithm testing			X
Backend	X	X	X
Frontend	X	X	
Final testing and tweaking	X	X	X

Gantt Chart

Task Name	Start Date	End Date	Assigned To	Duration
Barket second semester	**01/22/18**	**04/23/18**		**130d**
Web	02/05/18	03/16/18		40d
Config file class	02/05/18	02/08/18	Kate	4d
Models	02/09/18	02/11/18	Kate	3d
Add socket	02/12/18	02/18/18	Kate	7d
Prettify the view	03/12/18	03/16/18	Kate	5d
Backend	01/22/18	03/11/18		49d
Config Executable	01/22/18	01/31/18	Kate	10d
Barket Database	02/01/18	02/04/18	Kate	4d
P.O.S. Database	01/22/18	02/04/18	Jessica	14d
Calculations classes	01/22/18	02/11/18	Dennis	21d
Add socket	02/12/18	02/18/18	Jessica	7d
Config Connection	02/12/18	02/18/18	Dennis	7d
Write test cases	02/19/18	03/11/18	Denis	21d
P.O.S. GUI	02/05/18	02/11/18	Jessica	7d
Final Touches	02/19/18	04/23/18		64d
Connect the front and back end	02/19/18	03/11/18	Kate and Jessica	21d
Testing and tweaking and commenting	03/12/18	04/08/18	All	28d
Simple deployment to computer/server	04/09/18	04/23/18	All	15d

Final Project Design

Algorithm

The algorithm will utilize a known calculation for the dynamic pricing of supply and demand. At initial set up for the client (bar owner) we will have them interact with a simple GUI written in C#.NET that takes in the variance variables, the time period over which prices change, the time period over which the new prices are calculated, and can set minimum and maximum prices for each item in their inventory. The GUI will also allow the user to remove certain stock from participating in the dynamic pricing. Finally, the GUI will allow the user to group certain items. For example, the groups could look like cocktails, beer on tap, beer in bottle, wine, etc.

The calculation will take in the following input variables:

- baseInventory is the amount of inventory for an item at the beginning of the time period over which prices change
- currentInventory is the amount of inventory currently available at the time of calculation
- basePrice is the beginning price at the beginning of the time period over which prices change
- variance is the amount of change the owner wants to see from 1 to 10, where 1 is a small amount of change and 10 is a large amount of change

First the inventoryValue will be calculated for each item in stock using the following calculation:

$$inventoryValue = \frac{baseInventory}{currentInventory} * basePrice$$

Next the total variance will be calculated and the inventoryValue will be updated.

$$varianceCalc = (100 - (variance * 2))/100$$

$$totalVariance = varianceCalc * (inventoryValue - basePrice)$$

$$inventoryValue = basePrice + totalVariance$$

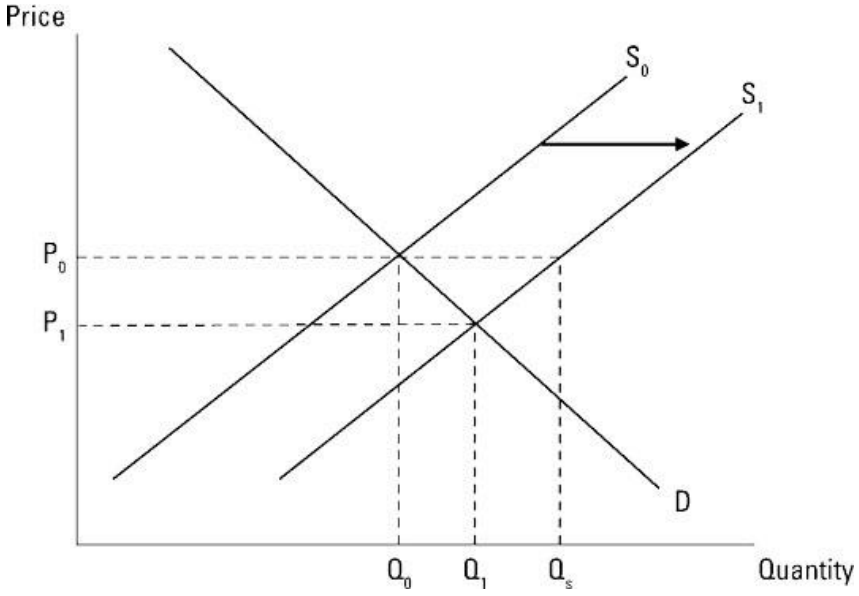
Finally, the new price will be calculated for each item:

$$price = currentPrice * (\frac{totalVariance}{100})$$

The variance variable for each subgroup and the basePrice will be stored in the Barket database along with the calculated change in price from the last time period. The remaining inputs will come by pulling data from the point of sale system.

As stated above, the owner of the bar will be able to set a max or min price that an item can reach. The calculations for each item will still occur and then a check will happen to make sure the new price does not exceed the bounds set if any exist.

Supply and Demand Chart



Tech Stack

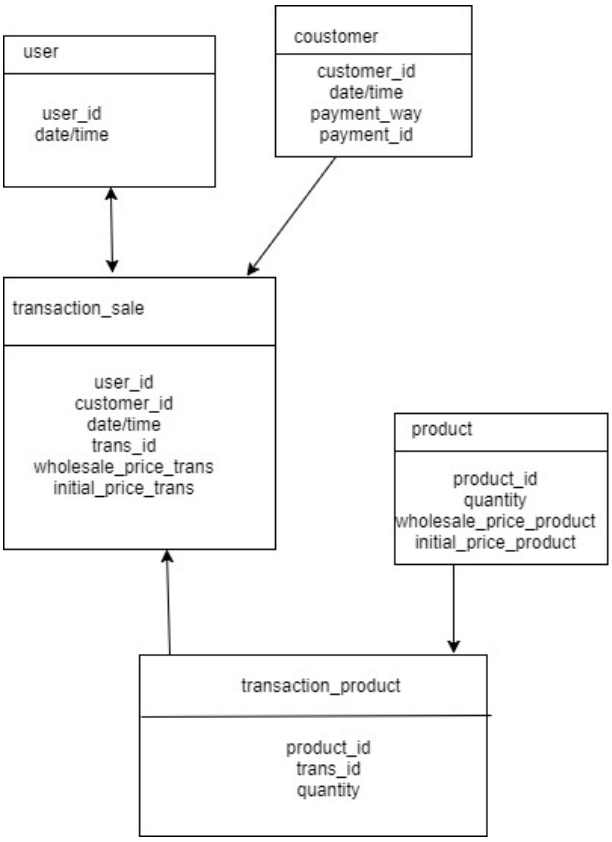
Our technical stack can be split into two parts, the service running on the server and the web application stack.

The service will be written in C# and will continually perform the algorithm calculations every time period. There are also two parts to the service that will interact with the set up, but will be in the form of an executable. The executable will be a basic POS that the servers would see for demonstration and adhoc testing purposes. The second part of the executable will be the configuration GUI. This configuration will store the variance variables, min and max price points, and allow for grouping of items and exclusion from the dynamic pricing into their respective locations. The GUI will also allow for web page styling such as choosing a color scheme and assigning a restaurant name and icon The locations will be a either in an XML configuration page or the Barket database. These GUI's for now will be written in C#.NET for a Windows environment.

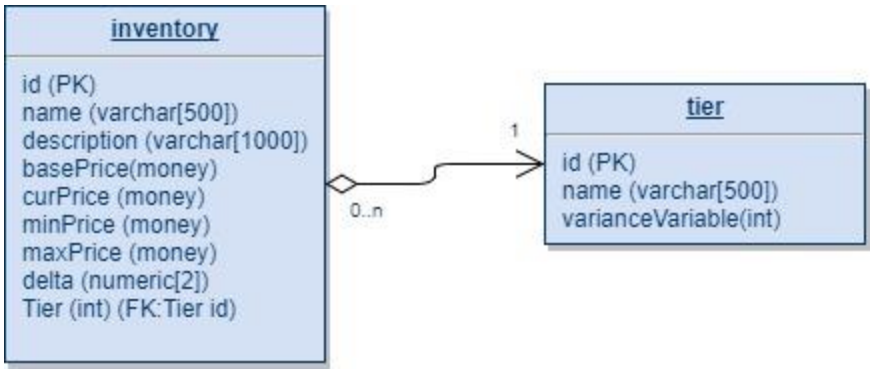
The web application will be written in Angular and ExpressJS. Angular was chosen because it has data bindings and is a well known framework with great documentation. Plus, a member on the team was interested in trying Angular out. LESS will be utilized for the styling. Express.js and Node.js will be controlling the building and backend logic for the web application. The architecture pattern will be Model – View – View Model (MVVM. The models will consist of three main components: a title bar, a group, and a stock item. The group will contain stock items. The web application will also be using socket.io in order to create a socket that connects with backend for updated values.

Database Schema

The POS system (seen below) will be represented simply in PostgreSQL as can be seen in the diagram below. This language was chosen because it is an open source language with heavy documentation and a relational database was desired for the architecture. POS systems are often represented relationally and we wanted to keep consistent so that integration of a POS system in the future would be easier.



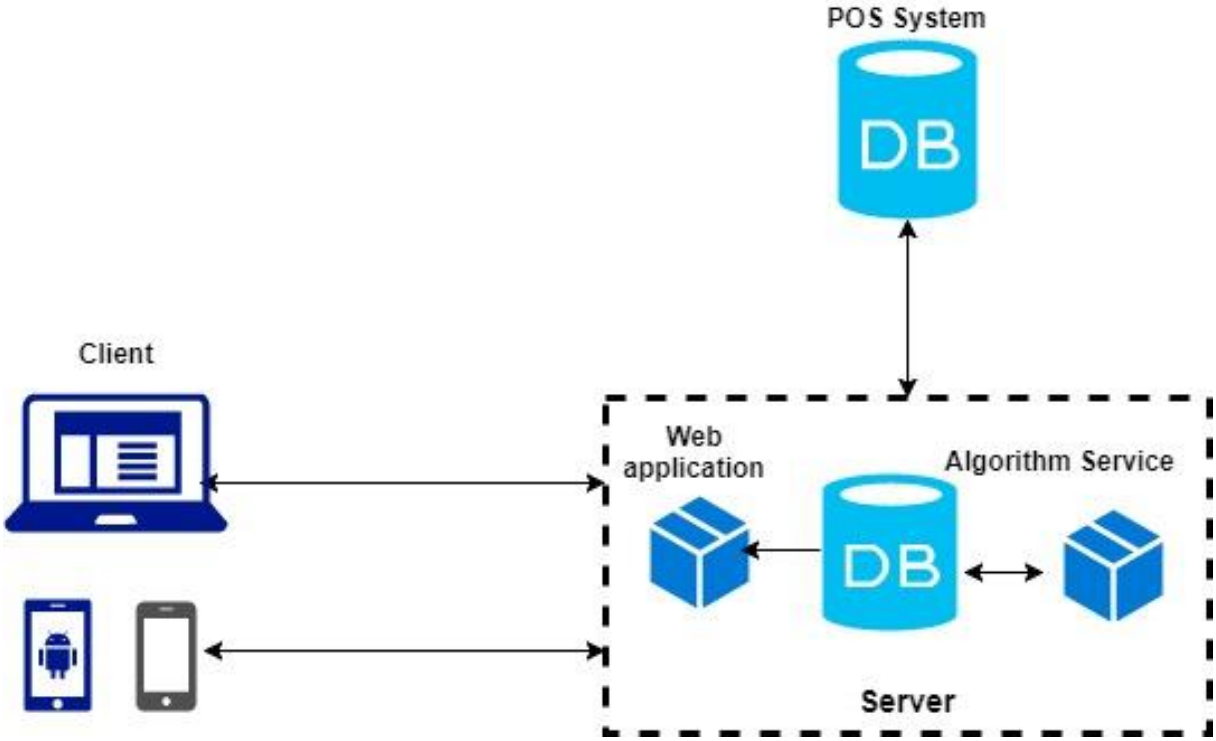
The Barket database will be represented as shown below and will also be in PostgreSQL.



Overall Architecture

The server will house the Barket database, the algorithm service, the two executables, and the web application. It will be a locally hosted Windows server, which also makes development easy. As stated above, the variance variable and the time periods may be configured by the bar owner. Every cycle of the time period, the algorithm service will contact the POS system via a basic API and retrieve the transactions, items in stock, and vendor price. The algorithm will then perform calculations on this data and store the final price point and its delta in the Barket database. The final price point will also be stored into the POS system so that the bar owner may still use all the software he has already purchased with his POS.

When the calculation is finished the service will send a JSON object to the client via the socket established, updating the price and deltas in the model, thus updating the view. No information from the client web applications will be stored in the database. All the configuration required will be stored in a config file in the web application source file.



Ethical and Intellectual Property Issue

Ethical Issue

Before examining our project using ethical systems of thinking, we must decide if ethics even applies to our project. We are creating an application intended to make purchasing alcohol a more interactive and fun experience for customers. The goal is also to increase the profits of the bar owner through novelty and to ensure that their stock moves evenly and quickly. Creating this project is not a voluntary choice and alcohol is a known hot topic amongst our society. When we enter college, for example, we are forced to take an online tutorial examining alcohol consumption and the effects it may have on the body and mind. Alcohol can kill people in overconsumption (an obvious societal no-no) and it could be argued that by making the drinking experience more fun with our project that we are possibly inviting more alcohol related problems to arise with patrons that want to keep on buying and testing the mini-stock market.

Utilitarianism is a way to examine ethics by considering the greatest amount of utility that our project's actions may take. The action our project takes is to increase profits for the bar owner by making a fun drinking experience for patrons and using an algorithm to ensure profit margins are reached. Our audience could just be bar owners because that is the group of people our project is geared towards. However, Barket is also intended to be used by bar patrons and potential patrons so we may want to include them in our audience as well. Finally, we examine the utility of this project on the audience. Bar owners are a small subset of our audience, but they would clearly be affected most by Barket. For this subset of the audience Barket is not necessary, but could potentially boost profits and move stock for bar owners, meaning more money for their lives. This would also be a very useful tool for the struggling bar owner looking to stand out from the crowd. When looking at the customers or potential customers, our project's utility is lacking. The customer does not technically need our tool to order drinks and for the customers who like stability in pricing it would almost be a negative for our product to enter their established bar. For those customers who find our product fun, it would introduce a bit of excitement and new-ness to their ordering routine. We believe that most customers would enjoy our product. Because our product has very little utility for most bar patrons, a negative utility for a small subset of patrons, and a beneficial utility for bar owners, our project passes the utilitarian test.

In applying the Respect-for-Persons test, we must analyze the alternative actions we have available. One option is to not make this project and do something else, which could potentially have its own consequences but let's assume we pick something with no consequences. The other option would be to create the pricing system for general purpose, instead of gearing it towards bar patrons. This option would not have any foreseeable consequences to society, however, if we adopted this concept it would greatly increase the scope of our project. We believe this would be a great option to pursue if we decide to continue the project past the senior design time. We intend to make the system as modular as possible to accommodate changes such as these for the future. Finally, the consequences of our current course of action. We put ourselves in the situation where we were at a bar with this pricing system. The four of us agreed that it would indeed increase the amount that we purchased, but that by using our personal judgement this system would not entice us to overconsume.

Intellectual property

The concept we are creating has been done before and therefore opens us to risk from intellectual property complications. There is currently a group based in the United States called Drink Exchange who

is also applying a stock market concept to bars. The algorithm we intend to utilize is based on well-known economic principles and therefore does not fall under intellectual property. Software copyrights can be applied to the expression of this idea, i.e. the source code and tech stack we intend to utilize. The idea itself is not able to be copyrighted. Because we have designed the tech stack of Barket from scratch, we highly doubt that any competitors will have an identical implementation. Further research needs to be done on the patents given to an idea like this, but for now we assume that the implementations of the idea are quite different.

Change Log

Section	Why
Project Budget	There was a change in the technologies chosen and where we would get a server from.
Project Milestones	Our group lost a member so we had to redraw the timelines.
Workplan milestones and Gantt	Our group lost a member so we had to redraw the timelines.
Final Project Design	Updated information and accomplishments that may be achieved on carrying out the project design.