Problem Statement

Henry Peterson

October 11 Fall 2018

CS461 Senior Capstone

Abstract

Ninkasi Brewing needs a way for employees to efficiently communicate about the current state of their brewery. We will continue building upon a partially made web application that will allow all employees to update the same repository of information and have helpful visualizations.

The Problem

Ninkasi is a medium size brewery, distributing to 12 states and provinces and producing 86,000 barrels of beer per year as of 2013. At any given time, they have 15 fermenting tank working on various batch in many different states of completion. They also take samples and perform multiple tests on them during different stages. Then they manually enter the state and test information into a huge Microsoft Excel spreadsheet which is in turn emailed to the rest of the company. This is a brittle system which has a number of issues. First, manually entering data is not only time consuming, but error prone. Fat fingering keys could lead to values being off by an order of magnitude which has a significant impact when dealing with such precise chemistry. They also manually assign tags and labels to the information based on data the is completely standardized, requiring no human judgement. Next, there is the issue of timing. If someone checks their spreadsheet before they check their email, they could make a desicion based on misguided information or spendtime determining when the last check was made. This is hugely inconvienient, people should have confidence the information their viewing is current. In short, their current systems takes too much time which costs money, and has to many errors, also costing money. The framework for a new system has been laid out by a previous team, but it is not to the point where it can be dropped into place. From the users perspective, it still requires manual data entry and does not utilizie all the information they need. From a developer persepective the code contains apparant bugs, has organizational issues, and has a complicated deployment process.

The Solution

To fix the problem, we will continue development of the existing web app and get it to the point that it can replace their current system. The existing code contains a simple web interface connected to a relational database keeping. The interface allows them to input data as well and create and edit brewing projects. To fix their manual entry issue, we will implement a system to take the data from their testing machine and transfer it automatically into the database. It will be readily displayed in a convineint fashion without having to export it to and external application. This will also prevent them from having to tediously organize the data. On the development side of things, we will refactor the code so that any one in the future who performs maintenence or adds features can do so without inordinant effort. Cleaning up bugs will make the the product work more consistently, but we also want to minimize bugs in the future. Implementing proper testing will ensure to us that we are writing quality code and those viewing it in the future will also be able to have confidence in its quality. Lastly, we will improve the deployment process. Deploment is one of the most important parts of the development process becuase it is such a bottle neck. No matter how much effort and polish is put into an application, if it cannot be depolyed, it is useless. Another thing that can make it tricky is deploying a program that you did not write yourself. By smoothing out this process for our application, a technical individual at Ninkasi could deploy and perform basic maintenance with ease.

Performance Metrics

The bottom line for our project being done is whether or not it can easily take the place of the current system. When meeting with the client, we will make a more specific list of what exactly they need for it to be what they need. Once that list of requirements is complete, it will be finished. It will be split into items that are completly neccessary, are items that are nice. There are number of people who will have input on what is acceptable. There is the brewmaster who presides over the entire process. They must be satisfied that it is tracking information properly, giving them reports that are as useful as they get from excel, and pleasant to view. There are also the lab technicians who are performing the actual reading of the data. They must be satisfied that the application does not get in their way or hinder them at all when testing and transfering information.