Problem Statement

CS 461 Fall 2018

Brennan Douglas

October 9, 2018

Abstract

During the brewing of a specific batch of beer there are many variables that need to be tested and tracked. This helps determine when the beer is ready and how it compares to other batches. Ninkasi — a brewery in Eugene, Oregon — was originally using a single large excel spreadsheet to track and store all their information. As they grew this began increasingly unwieldy. So, they began to create a web application to enter, store, and display all of their data. As it currently stands it is not quite fully functional needs new ways to import data, and a better user interface.

1 Problem

Currently Ninkasi keeps all of their data in an excel spreadsheet, this gets passed around among different people who enter different bits of data from paper forms. This is a very ineffective way to store such valuable information. It restricts who can see the data, as well as who knows if they are looking at the current data. This method is also heavily prone to error as entries can be easily changed by accident without any knowledge of this occurring. There is no history feature to keep track of edits, neither their changes or authors. A solution to solve this has already begun in the form of a web app, however it was left in an unfinished state. Currently, all data needs to be manually input through the forms of the website. This creates a blocking point for inputting historical data as it is in the tens of thousands of data points. The core functionality of the application already exists. It needs to be refactored to become usable, quality of life features to be added, and to be polished for a production environment.

2 Proposed Solution

Ninkasi has requested that we fix up the existing application for production. They have provided a comprehensive list that gives us clean cut goals to meet. We need to improve how the data is entered into the system, that is streamlining the process for a user on a mobile device. This is important as most of this data will be entered in the brewing warehouse without a computer present. Currently, the data entry process is cumbersome to use on phones and tablets. Our solution to this will be redesigning the page with heavy testing primarily focused on mobile. The login and admin system also needs to be refreshed as it is currently not working. In summary, our solution to fix the existing web app is bug fixing and extensive testing on both desktop and mobile. This will include many conversations with Ninkasi to confirm that the app is easily usable in the required settings. Pages will be redesigned and corrected as needed and any bugs that are found along the way will be ironed out.

In addition to smoothing out the old system Ninkasi has also asked for some additional features to be added. A current hole in the design exists where all data must be imported by hand, point by point. As they already have a large backlog of data — many years worth — stored in several large excel spreadsheets they have requested a feature to submit these sheets and import the data from them. This will require the addition of a new page with the option to upload an excel spreadsheet. It would potentially need to include configuration information to extract the data from the sheet if needed. The application would parse through the file and create new data entries based off of the rows. New methods of data visualization have been requested as well, specifically fermentation curves for the different beer batches. This is a common graph they create and want to it to appear automatically in the app allowing for easy filtering. This will be accomplished by using the existing graphing library to create a new page with these graphs on them. The different beer batches and date ranges would be selectable so different curves can be easily compared.

There are some facets to our proposed solution that we have injected as well. They relate to the development and deployment life cycle of the application, an area that Ninkasi lacks expertise. Currently, the application is written using standard JavaScript and React. We suggested that they migrate this to using TypeScript as React works very well in this environment. Also, half of the old application was configured to use docker to run. This lacked the full proper configuration still managing to fail to run sometimes, even when docker is installed. We will update this to make running the application a painless procedure. This will more easily allow Ninkasi to maintain the application. We will also docker-ize the other half of the application so that it can all be deployed in the same way, further simplifying management of the application to one technology set.

3 Performance Metrics

The purpose of this project is to smooth over and complete the application that already exists so that is production ready. This is a somewhat loose metric. We will defin it through a set list that as a team we agree on with Ninkasi. This will include bugs and improvements that need to be made to different sections of the application. If we are able to show that each feature properly works as stated in that list — and in comparison to the old application — then we will be done. That list will act as our metric for the improving and polishing section of the project.

Similarly with the new features, the performance metric will be the application features working according to the documented methods agreed upon. It is important that it is documented, agreed upon, and effectively ”sealed” as that will be our endpoint; we don’t want that moving on us. One feature will be the excel file import. The performance metric here will be loading up the web page, uploading an excel spreadsheet, and showing that its data has been input into the system. These types of features are easy quantify as they work with Ninkasi’s excel sheets or they don’t, a simple test will be performed.