CS 461 - CS Senior Capstone

Fall 2017

Software Requirements Specification

Connor I. Christensen

chriconn@oregonstate.edu

Lily M. Shellhammer

shellhal@oregonstate.edu

William B. Buffum

buffumw@oregonstate.edu

November 5, 2017

Abstract

Ninkasi Brewing Company is using an error prone, labor intensive and outdated brewing operations management system (BOMS) to manage the brewery data produced during the fermentation process. Our team (BrewHops) is tasked with creating a new system as both a proof of concept about new technology, and to give Ninkasi another option for storing and viewing brewery data. This application will be web-based to be lightweight, easy to use, and accessible on many devices. The data will be hosted in a relational database to provide stability, power, and consistent usability through time.

Keywords: Brewing, Operations, Management, Ninkasi

Contents

1 INTRODUCTION

1.1 Purpose

This document will outline a project that satisfies the requirements for Ninkasi's new brewing operations data management system. This includes a detailed description of the product created and the technology utilized to build the product. This document is intended for a technical audience, primarily for our instructors, Kirsten Winters and Kevin McGrath, and our sponsor, Daniel Sharp.

1.2 Scope

This project is to develop a web application and relational database model to manage Ninkasi's data. We will perform a cost-benefit analysis to recommend hosting the system with web service provider versus hosting on Ninkasi servers. The database[1] will store all information that is required during the brewing process. This information is pulled directly from the Master Excel Spreadsheet that Ninkasi uses to track brewing process data.

The provided interface will allow brewers to manually insert new batch records, update incorrect data, and view existing data. The web application interface will allow Ninkasi personnel to view basic status information on tanks at a glance. We will gage adequacy of interface based on a minimum 60% approval from Ninkasi personnel, of which Daniel Sharp must approve. The goal of these requirements is to stop use of Excel as primary data storage.

Stretch goals are outlined in section 3.2.

We are not be responsible for advanced application security or direct control of any brewing process.

1.3 Definitions, Acronyms and Abbreviations

Database - area in system where collected information will be stored

Cellaring data - data collected by our database; data collected while cellaring beers

Data points - information to be stored in our database[2]

BrewHops - The name of our team

BOMS - Brewery Operations Management System

Master Excel Spreadsheet - Ninkasi currently tracks all data using a spreadsheet, referred to as the ”Master Excel Spreadsheet”

1.4 Overview

In the next sections, this document outlines details on the hardware, software, and user interface requirements for our project. It explains what needs to be completed in order to launch our minimum viable product. Security, functionality, and organization are explained in detail.

[1] Details on database information are outlined in section 3.4

[2] Breakdown of all data points found in section 3.4

2 Description

2.1 Product Perspective

2.1.1 System interfaces

Brewers will interact with our web application in the cellar. The web application interacts with the database and will enter data points from the brewer's entry into the form. If we meet our stretch goals, brewers will be able to extract Excel spreadsheets from our web application.

2.1.2 User interfaces

The user will interact with the web application through either an internet connected phone, tablet, desktop, or laptop computer. Phones will operate almost exclusively in portrait mode. Apart from the occasional chart or data display, most of the use for the phone will be entering data, which benefits from a portrait layout. Tablets have the ability to use both portrait or landscape mode, depending on how they would like to use the device, and which method is more convenient for them. Desktops and laptops will work in a landscape mode. For big enough screens, extra space can be used to fit more functionality on one page without having to hide menu items or actions, such as extracting data from the database.

All Ninkasi brewers use a company provided phone and most of the brewer's work will be done through that phone. It's important that the site be consistent between iOS and Android to increase reliability, allow brewers to help each other with problems regardless of operating system, and reduce the learning curve. The environment in which the brewers are working requires that the data entry system be clear, easy to see, and have a forgiving[3] interface. For any new hires, the system should be intuitive[4]. Some phones may have smaller screens or may be held at a distance, so it's important that the font be big enough, the contrast strong enough, and the input boxes take up as much space as they can so they are easy to tap and input is easy to see. Brewers may have water, grain dust or other materials on their hands that will interfere with the touch capabilities of the device. Making the interface forgiving by making buttons, input fields and any other interactive components large will reduce error and frustration.

The system should use terminology that brewers are familiar with. Reports should be color coded to keep the screen from filling with text and to allow brewers access to data easily[5].

2.1.3 Hardware interfaces

Ninkasi has a small server in their headquarters. A cost-benefit analysis will determine if this server will satisfy the hardware requirements of the system. If it will not, we will provide at least one recommendation for web service providers.

2.1.4 Software interfaces

The user will interact with data via a web application. The application will expose input forms to submit data. We will release the software in stages, starting with a database, then a basic web input form connected to the database, then

[3] Error handling to prevent incorrect information from being added to the database. Will require Daniel Sharp's approval to be considered adequate.

[4] 60% acceptance of interface required to be considered ”intuitive”

[5] 60% approval from brewers will be considered ”easily” accessible data

our final minimum viable product will include the data display interface. The data display interface will involve giving basic information about the status of the tanks in the facility, and which actions the brewers should take to make sure that the product will be developed properly. Streatch goals include more advanced data display such as the ability to show simple graphs of data over time for any tank.

2.1.5 Communications interfaces

Web application will not implement other interfaces.

2.1.6 Memory constraints

Currently Ninkasi has space on a single server in a back room. If Ninkasi's storage requirements grow quickly, a single server may not meet the needs of our system. If Ninkasi decides to host our database on a 3rd party cloud hosting site, the memory constraints will be those of how much space they buy.

2.1.7 Site adaptation requirements

Brewers will adapt to using our web form rather than continue implementing the system they previously used. They will enter data only through the web application.

2.2 Product functions

Product includes the development of a relational database and development of a web application to process and present database information. Brewers will input information to the database using a form on the website.

2.3 User characteristics

Brewers using the web application will have all levels of education, including PhD, MS, BS, and little to no college experience. Their technical expertise about web development and coding in general will be minimal, but they will understand how to use mobile applications, how to access websites, and how to enter data into forms.

2.4 Constraints

2.4.1 Regulatory policies

Because we are creating this software from the ground up and there are no previous regulations for a system like this, this section is not applicable for the BrewHops team.

2.4.2 Hardware limitations

Ninkasi has a small server that we may utilize for the database. If we decide not to use a third party hosting site, then the small server means there will only be the amount of free space on the server available. This is a not a limitation for our project so much as a limitation for any possible continuation of our project from Ninkasi.

2.4.3 Interfaces to other applications

BrewHops application will integrate with Tanknet. [6] The web application will automatically collect temperature information from Tanknet and store it in our database.

[6] Tanknet is a a hardware and software system that performs physical measurements of brewing vats and automatically collects the data and exports it digitally

2.4.4 Parallel operation

Our web application will allow concurrent read access and use transactions for insertions and updates.

2.4.5 Audit functions

The system will store past data[7] to be used for audit functionality.

2.4.6 Control functions

Not applicable for BrewHops team.

2.4.7 Higher-order language requirements

Objects and callback functionality may be necessary.

2.4.8 Signal handshake protocols (e.g., XON-XOFF, ACK-NACK)

BrewHops team will not work directly with handshake protocols, these protocols will be abstracted by the technologies used.

2.4.9 Reliability requirements

The web application will ensure data integrity for automated data collection. The web application will validate data entered manually. Our site will be accessible without wait time longer than 5 seconds 90% of the time[8].

2.4.10 Criticality of the application

The web application is not critical to Ninkasi Brewing Companys success, but it will save the company data entry time. Ninkasi already has a working system that is not online. Our system will improve the current system by reducing time wasted from data entry error and replace their working data entry processes.

2.4.11 Safety and security considerations.

This product will be light on security, as it is an internal application and should only be accessed by verified employees in the company. As this is not a public facing product, malicious activity is low on our list of priorities. Further details are outlined under the Specific Requirements section.

2.5 Assumptions and dependencies

BrewHops team will have access to Ninkasi Brewing Company's server to deploy application. Ninkasi is willing to use the software technologies and languages we develop in.

[7] Past data stored as far back as deemed necessary by Daniel Sharp, or within limitations of server space.

[8] This may vary depending on the hardware capabilities of Ninkasi servers.

3 Specific requirements

3.1 External interfaces

Included in product are as follows:

- Database: A database will either be hosted on Ninkasi's server or through a 3rd party site. The input to the database will include multiple columns of cellaring data, who entered the information, and timestamps.
- Web Application: The web application will consist of a website that scales/changes functionality to fit the device.
  - Mobile Phone Page: The web application will scale to fit the mobile phone, mostly in portrait mode. The mobile version will be used for data entry and will have minimal options in comparison with the tablet or desktop.
  - Tablet Page: Tablets will operate in either portrait or landscape mode and will have the form and possibly more options for data extraction.
  - Desktop/Laptop Computer Page: This page will operate in landscape mode and will have the most options of the three pages.

3.2 Functions

Data will be validated upon input to database. Information being entered that is far outside the range of expected values will prompt the user to verify that the information is indeed correct. Every input form will strip out all html special characters to make sure that no harmful code can be injected into the database.

Stretch goals include:

- Adding a login interface to verify the users before they begin using the system.
- Implement profile levels to differentiate permissions (admin, operator, supervisor, cellar, etc.).
- Track authenticated user levels in database.
- Including a time and ID stamp on the data entered in case the company needs to trace back an error to a particular time or person.

3.3 Performance requirements

Multiple users on the web application at once will be permitted. Application should not crash if multiple brewers input data at the same time. 95% of interactions with the web page should happen in less than 20 seconds[9].

3.4 Logical database requirements

- Information pulled from Master Excel Spreadsheet
  - **Fermentation Vessel, FV**: tank fermentation is carried out in.
  - **BEER**: beer abbreviations associated with temperature profiles and dry hop recipes.
  - **BATCH**: batches are groupings of sequential brew numbers inside a particular FV[10]
  - **Yeast Generation, GEN**: the generation of the yeast we use to ferment the beer.

[9] This assumes basic interactions and prior experience with the system.

[10] Example: if there are 5 brews of IPA with numbers 3000, 3001, 3002, 3003, and 3004 in FV7, the BATCH ID will be: IPA 3000-3005 FV7.

  - **Vol To Ferm (BBL)**
  - **Vol To Bright (BBL)**
  - **BRIGHT (BRITE)**: tank where beer is sent after fermentation, used to track volume of beer and associated losses.
  - **% Yield**: normally stored onces per batch but a batch from one FV may split between two smaller brite tanks or two batches from two FVs may blend into single brite tank.
  - **Dry Hop/Amount Type**: recipe for given dry hop [11] [12].
  - **Amount (lb)**: amount in lbs of dry hops to be added to tank.
  - **Monitoring Points**: minimally measured daily.
    - **Specific Gravity**: A measure that represents the amount of sugar left in the beer. This number starts high and decreases throughtout fermentation. When it stops we move on to the next process step
    - **pH**: pH level of batch in brewing vat.
    - **ABV**: Alcohol By Volumne of batch.
    - **Temp**: Temperature of batch.
    - **Action**: Action taken.
    - **By**: person taking the action above.

- Frequency of use
  Data currently updated daily and read throughout the day.
- Accessing Capabilities
  Database will be access from database access layer of web application.

3.5 Design constraints

Ninkasi currently has one server. Our job is to create, or aid in creating, a new database system. We can either use their existing platforms, meaning limited space, or create a database on a third party hosting site. Either option is limited by time, we can only implement so much of the data management throughout the timeline of this project. Stretch goals are to expand the database and data entry options, given we have time to do so.

3.5.1 Standards compliance

The database will store information with correct relationships according to Ninkasi's current Master Excel Spreadsheet.

3.6 Software system attributes

3.6.1 Reliability

There is room for occasional bugs given that they do not damage data or inhibit the brewery process. Reliability of the system is a desirable goal, but the product is not a mission critical piece of software. Down time would be frustrating for employees, but would not render the company useless. A stretch requirement will be that the system is active or available 96% of the time[13].

[11] Example: dry hop type A = 50 lb, dry hop type B = 75 lb.

[12] Dependent on the BEER and volume in fermentation (for dry hops).

[13] Measuring this will depend on if the system is deployed for at least 30 days before the Spring term expo showcase.

3.6.2 Availability

The system should not require a reboot, recovery or restart after deployment. Stretch goals include making a daily backup of the information in the database if the system were to crash or the data were to be corrupted.

3.6.3 Security

Individuals not associated with Ninkasi Brewing Company should not have access to the system. As this is not a public facing product, malicious activity is low on our list of priorities. Stretch goals will focus on security such as providing a login interface for employees, adding time stamps for data entered, and providing an easy method to find out who was responsible for entering incorrect data or misuse of the system. There will be checks built into the delivered product, but those will primarily be designed for taking care of user error rather than malicious intent.

3.6.4 Maintainability

This software is designed to be a proof of concept that having a system like this would be beneficial to Ninkasi's business development. Therefore, maintainability is not only good as a good coding practice, but it is one of the core requirements for the project that the code be maintainable and well documented[14].

There should be a clear separation between the frontend and the database connections. This may be done through a REST API structure. This would allow future developers to implement new interfaces as needed.

For the most part, code should be broken up into discrete functions that would allow for unit testing in the future. Though unit testing is not required as a goal for this project, the ability to attach some kind of testing framework will be useful.

3.6.5 Portability

This program is designed to be highly portable and the format of the product was selected specifically for that reason. The percentage of code and components with host-dependent elements should be as small as possible, likely around 40%. Our site will be built using web technologies that will operate properly on Chrome and Firefox. In order for the system to function properly, the hosting server must have the technologies used in the system, installed.

3.7 Organizing the specific requirements

We will build a user class based system. This will allow us to separate responsibilities and functionality so individuals only have access to their required tasks. In this system, we can design it hierarchically so that managers have access to all data manipulation privileges while brewers only have read or write privileges through structured forms. As a stretch goal, manager level individuals will be able to modify user permissions as needed.

3.8 Comments

4 Appendixes

5 Index

[14] Documentation will be stored in the GitHub repository.

Created with Free Edition

OSUNinkasiCapstone

Start the Project
- Pick a project
- Establish communication lines
- Set up meetings with both instructors...

Business Plans
- Draft a problem statement
- Finalize the problem statement
- Requirements documents draft
- Finalize the requirements document
- Technology Review draft
- Finalize technology review
- Design Document draft
- Finalize design document
- End of term progress report

Determine Hosting
- Determine if the site is hosted at Nin...

Winter Break
- Winter Break

Minimum Viable Product
- **Database**
  - **Database Design**
    - Get a copy of the current data st...
    - Select a database technology
    - Draft the database design
  - **Database Implementation**
    - Implement database
    - Stress test database
    - Report on data retrieval capabilit...
    - Design for REST API
    - Connect webpage to database
- **UI Access Points to Database**
  - Create input fields for data submiss...
  - Add buttons that will dump databa...
  - Ensure site is usable on desktop a...

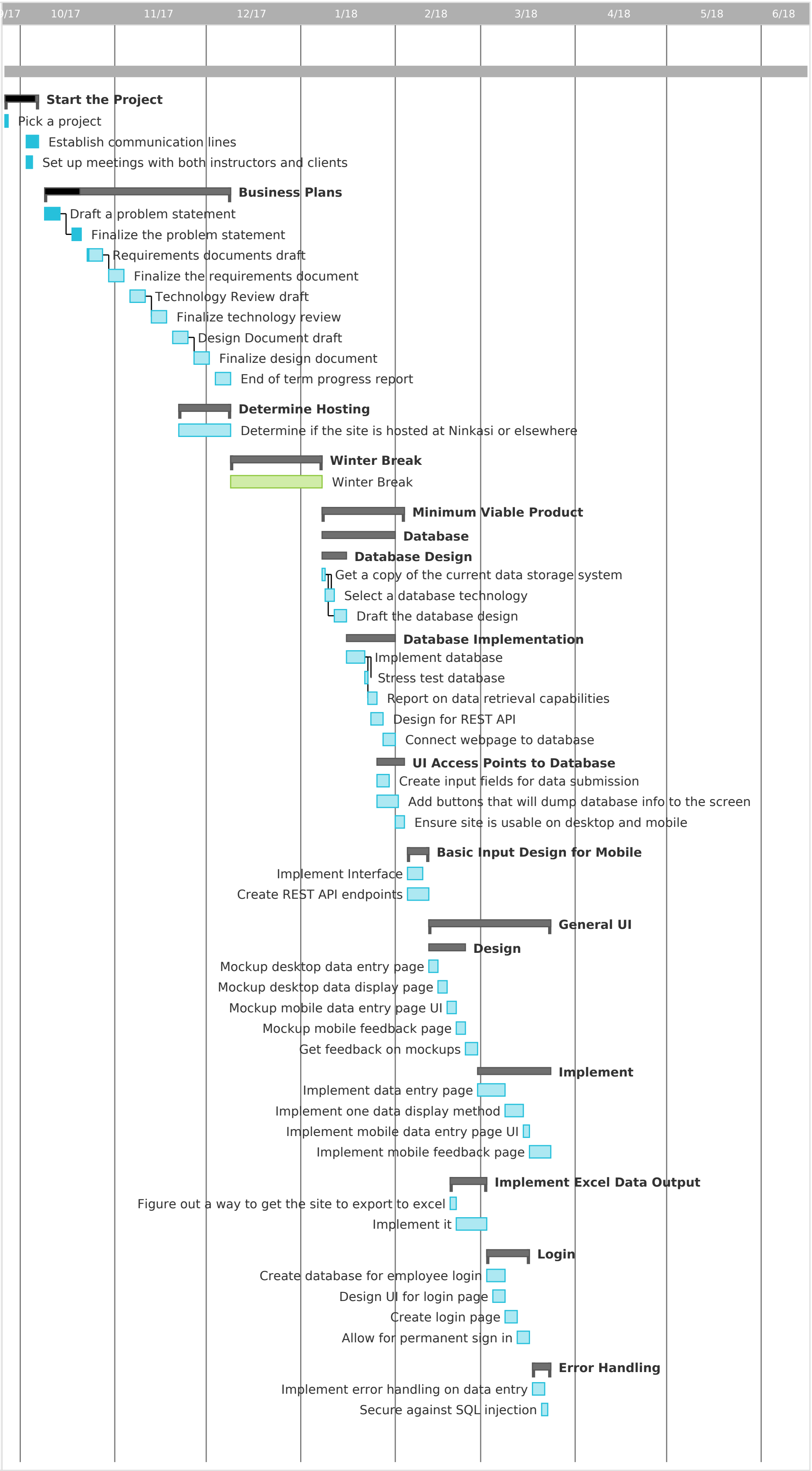
Basic Input Design for Mobile
- Implement Interface
- Create REST API endpoints

General UI
- **Design**
  - Mockup desktop data entry page
  - Mockup desktop data display page
  - Mockup mobile data entry page UI
  - Mockup mobile feedback page
- Get feedback on mockups
- **Implement**
  - Implement data entry page
  - Implement one data display method
  - Implement mobile data entry page...
  - Implement mobile feedback page

Implement Excel Data Output
- Figure out a way to get the site to ex...
- Implement it

Login
- Create database for employee login
- Design UI for login page
- Create login page
- Allow for permanent sign in

Error Handling
- Implement error handling on data en...
- Secure against SQL injection



Check for other obvious security issu...

Spring Break

Spring Break

Auto Data Imports

Figure out possible technologies for i...

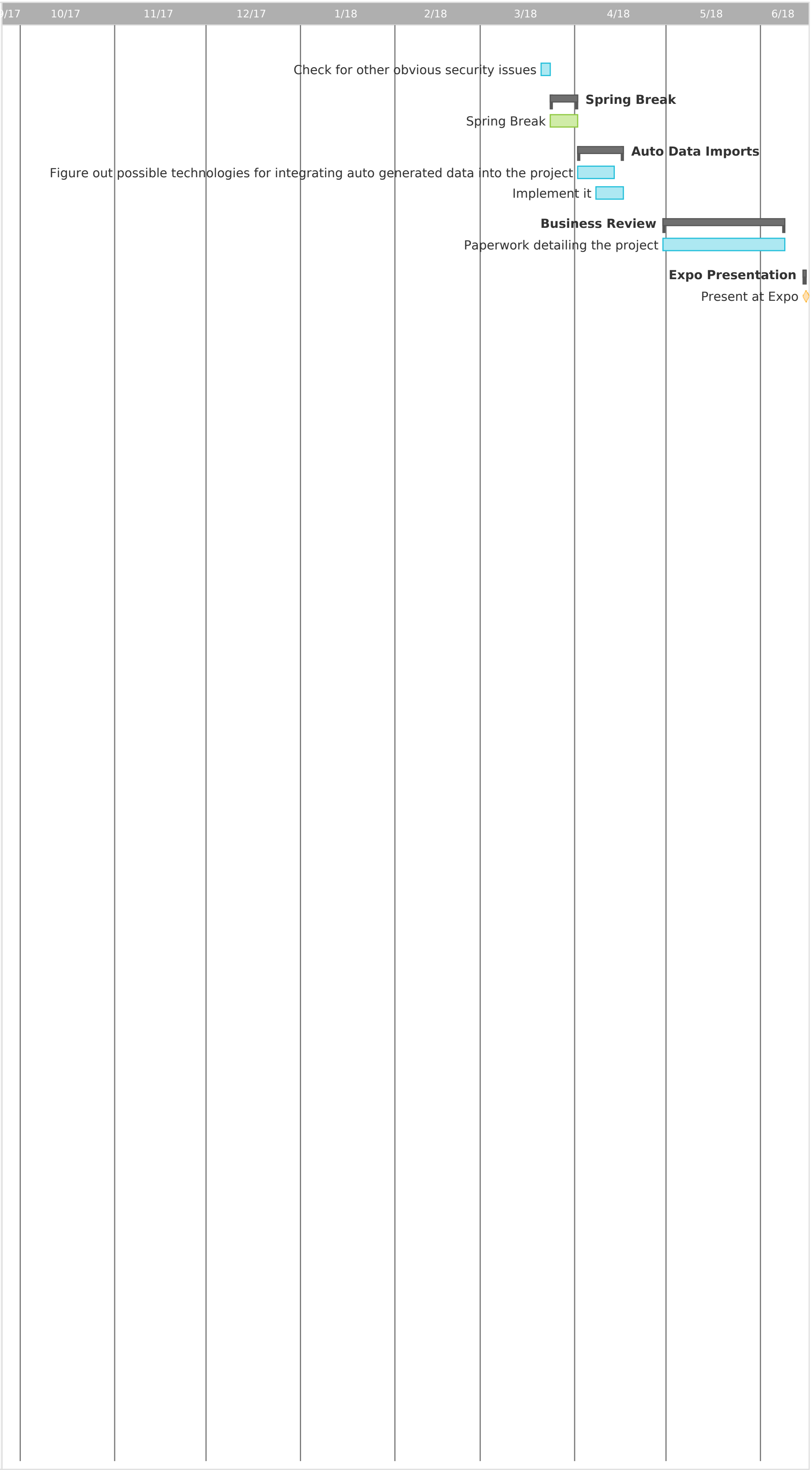
Implement it

Business Review

Paperwork detailing the project

Expo Presentation

Present at Expo


/17
10/17
11/17
12/17
1/18
2/18
3/18
4/18
5/18
6/18
Check for other obvious security issues
Spring Break
Spring Break
Auto Data Imports
Figure out possible technologies for integrating auto generated data into the project
Implement it
Business Review
Paperwork detailing the project
Expo Presentation
Present at Expo