



THE UNIVERSITY
*of*ADELAIDE

Final Report of Group RTMST1

Real-Time Wine Sensing Tool

by

a1779153 Vandit Gajjar

Project Vision Reflection (Individual):

The project vision was initially focused on creating a framework based on Spectral Change to help the winemakers to make a more quality product with lower cost in the making process. However, due to legal issues and delays in the paperwork, during the complete product development timespan, the team did not have received any dataset regarding the task. Although, the team has been provided with a mock dataset to work on it during the mid-term and because of that the task definition was changed from Spectral Change to Outlier Detection using Machine Learning approaches.

The Outlier Detection task was mainly focused on detecting the outliers which were different from the recent trends. i.e. Considering one of the features - Ph Quality. If the value of Ph Quality was different from the last 20 rows or mean average from the previous data, then it will be an outlier based on the recent trends.

The change in the task definition is assistive to the winemakers as data is crucial to Machine Learning models and if the provided data is huge and have optimal and precise values, the model will be much more efficient and take fewer efforts from the human labor which ultimately reduce the labor cost and makes the wine-making process more quick and effective.

Customer Q&A Reflection (Individual):

There were many questions that have been raised by the team members during the client meetings and the client has provided an effective response based on his ability and knowledge of the project and domain. However to precisely summarize the most important questions and their response from the client have been listed below:

- In the first meeting, the team raised the concern regarding the dataset as the project was mainly focused on Machine Learning, and to train the model for accurate prediction the dataset will be required. Along with that, the team has also asked for any specific technologies/framework which will be helpful to prepare the final product integration easier.
 - Client Response: The client explained directly to the team that to provide the dataset, it requires the legal paperwork to be completed by the University and because of that there will be a delay in the worst-case scenario. Considering the technologies/framework, the client provided the choices and explained every choice in a brief and gave the team options to select from it based on the team members knowledge in the domain.
- The team has already started working on the UCI Machine Learning Dataset for Wine Quality [1] because of the delay and in one of the meetings, a team member asked for the mock dataset in order to generate a similar synthetic dataset to work on.
 - Client Response: The client appreciated the team's work and delivered a smooth message to the team that this will be a good idea and will work hard to provide us the mock dataset to generate a synthetic dataset. (The mock dataset has been provided to the team on mid-term because of the client's hard work.)
- There was a change in the definition of the project task and the team was working on a different task when the notification has been received and in

the next meeting, the team has asked for clarification regarding the overview of the final task – an Outlier Detection using Machine Learning approaches for the recent trends.

- Client Response: The client has provided a brief clarification on the complete task with the task breakdown that how to approach this task in a simpler manner, what could be the backup plan if one of the strategies fails, why this task will matter in contrast to the previous task, and etc.
- During the final weeks, we have asked for suggestions on how to effectively present the product demonstration and integrate every component/module in such a manner so that any novice worker will be able to operate the product without any issues.
 - Client Response: The client delivered a strong message for the product demo presentation in terms of how to showcase the front-end and back-end development that the team have worked on and considering the integration the client provides a series of suggestions and video series on how to make the final product (i.e. an application/software) from scratch with different sub-modules.

Reflection: Every meeting with the client was in a virtual platform (i.e. Zoom) and very few conversations have taken place on either E-mail or Discord. The reason to meet virtually was agreed upon by the team members and client because of the COVID-19 restrictions. In my opinion, I believe that the virtual e-meetings were actually tough as there is a communication difference between in-person meetings and virtual e-meetings. Overall the process of updating the client every fortnight, updates regarding implementation, add the necessary task, and based on the feedback improve the team performance is quite remarkable, thus in my opinion these client meetings were really helpful to improvise myself and the team as well.

There was an issue I and my team has faced during the initial meeting was that without reading the project summary we have attended the meeting, and which was a disaster and because of that our initial discussion was not fruitful and

engaging. Thus one of the learning lessons was to always read necessary documents relating to the project always helps and make the discussion more engaging and healthy.

During the complete timespan of the product development, the main concern we faced was regarding update the team about what modules have been implemented and prepare the documentation. This was the core reason that the team has not been able to complete the final product. This concern could have been resolved if the team communication and wiki documentation have been created for every update. So, this was one of the important learning lessons for me and the team to let other members know about what one is doing, let the team know if one is not able to make up the timeline so that scrum master and the team can work on the issue to resolve quickly and finally prepare the documentation accordingly.

User and User Stories (Group):

Following are the finalized roles our team has identified for the product:

- **Admin User:** The admin user is the head and has responsibility for assigning the different batches, provide credentials, and grant access to the system for different users. The potential actions for the admin users are as follows:
 - Log in and log out from the system on daily basis
 - View the batches and check for abnormalities
 - Assign the barrels to different batches
 - Assignment of different batches to different users
 - Run the outlier detection modules to check the outliers from the data
- **Normal User:** The normal users are the common users who have been provided access to operate the product from the admin. The normal users have the ability to view the batches assigned to them and can run the outlier detection module. The potential actions for the normal users are as follows:
 - View the assigned batches in the dashboard
 - Detect the outliers from the assigned batches
 - Request for the system maintenance and support team access

There was a bit of contrast we have in comparison with the initial report as our product task definitions have been significantly changed. Initially, we have had finalized the Wine Maker, Wine Historian, Wine Tester, and Executive from Wine Making as our initial users, however finalizing the task of outlier detection task only we created our product focusing on specific users which are Admin users and Normal users. The product has been created in such a way that with only limited users most of the work can be completed quickly with only certain functionalities.

Software Architecture (Individual):

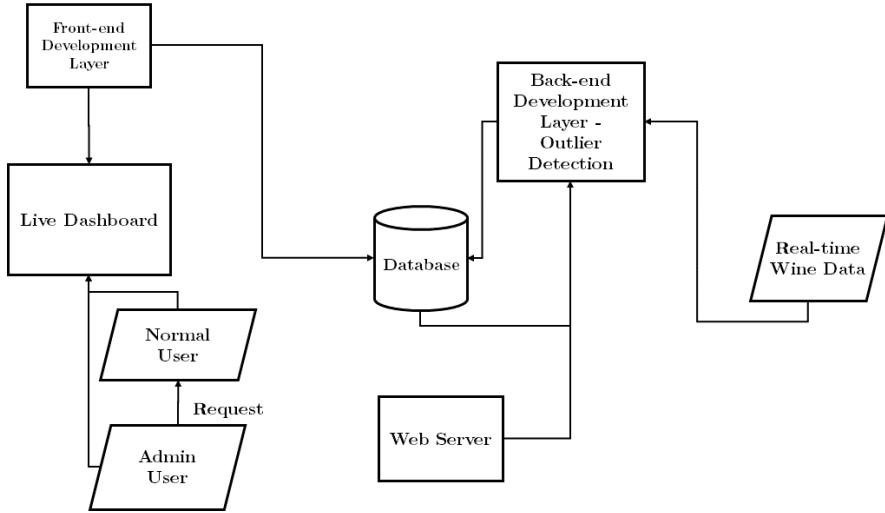


Figure 1. **Final architecture** for RTMST application.

Our initial goal was to create an application that will run in real-time and provide analysis such as Fermentation Tracking, Malolactic acid track, and Outlier Detection however, our task definition was changed to only outlier detection for the data. Thus it is important to create an architecture that can process the real-time data, run the machine learning model for outlier detection in the back-end. For providing visualization of the data and detected outliers, the team decided to create a user dashboard. The user dashboard has several functionalities such as an admin or normal user can log in to the system via provided credentials, view the data of different batches, and run an outlier detection module for detecting abnormalities from the data. Thus the software architecture shown in Figure 1 has been designed after applying the optimization.

Tech Stack and Standards (Individual):

Our team has explored different languages and frameworks before finalizing the choices. The selection criteria were based on expertise in the domain and proficiency in the language/framework of individual members. The final tech stack for our product is listed below:

- **Back-end Tech Stack:**
 - **Programming language:** Python
 - **Open-source libraries:** NumPy, Pandas, Scikit-learn, SciPy, Seaborn, Matplotlib
- **Front-end Tech Stack:**
 - **Programming language:** HTML, CSS
 - **Open-source libraries/frameworks:** UI Bakery, Adobe Photoshop API

One of the most important points to close the communication gap because of COVID-19 was to find an efficient way to meet frequently in the scrum meetings and communicate regularly via a channel. The tools for communication and development are listed below which the team has used throughout the project timeframe:

- **Facebook Messenger:** This channel was the primary means of communication where we update on daily basis regarding the implementation.
- **Zoom:** For weekly scrum and fortnightly scrum meetings, the team decided to use Zoom as the main virtual face-to-face channel.
- **Discord:** The team uses Discord for official communication where the results of the implementation were having been shared with the client and for project progress updates.

The purpose of following the coding standards in the project development is to improve readability, maintainability, complexity reduction, and increase in efficiency of the developer. The team used certain standards which are listed below.

- Naming conventions for local variables, global variables, constants, and functions.
- Proper indentation
- Exception handling conventions
- Function length should not be large – use OOP as much as possible
- Code should be well documented
- Avoid using a coding style that is too difficult to understand.

The choices for tech stack and communication channels were based on the team's approval with the familiarity of the platform. However contrast with the initial report, there were several changes the team has adopted. The reason to adopt these changes was because of a change in the task definition the team received from the client in the middle of the holiday breaks. Therefore we finalized the above choices based on team members' expertise, convenience, ease, and familiarity with the platform. I believe that overall the team did a tremendous job of communicating in the scrum meeting and sharing the knowledge between individuals. The only thing that is contrasting with the initial report is that the team discarded JAVASCRIPT programming language from the Front-end tech stack because of a change in the task definition. Throughout the entire project timeframe, the team has followed the coding standards.

Group Meetings and Team Member Roles (Individual):

The team members of our group agreed to meet virtually due to the COVID-19 situation and restrictions. Thus, the team decided to meet Monday and Wednesday at 5.30 pm - 6.00 pm for progress and product development each week on the Zoom platform.

Our team created a poll to check the availability of everyone to attend the sprint retrospective meetings with the tutor - Mr. Navpreet Singh Ahuja. Based on the statistics all the sprint retrospective meetings were scheduled on Wednesday from 5.00 pm - 5.30 pm each fortnightly.

The additional feedback channel with the customer was mainly Discord, where we frequently ask for feedback and improvising the experience. The scrum masters for each sprint listed below with sprint number, their name, and student number.

Table 1. **Scrum master** for each sprint with sprint number, name, and student number.

Sprint Number	Name	Student Number
1	Mr. Harnoor Bandesh	a1783911
2	Mr. Harrison Greven	a1716640
3	Mr. Ross Pickett	a1671245
4	Mr. Jay Hansen	a1670268
5	Mr. Vandit Gajjar	a1779153

In the initial weeks and till sprint 2, during the group meetings each and every team members were taking active participation, raising the issues for particular decision and choose the work by themselves, however after sprint 2, few of the team members were not even attends the meeting and this happened till the last week and only limited team members were taking the responsibilities, which was the major concern I believe the team has faced during the entire timespan of the product development. Considering the meeting minutes and updates, I have provided these crucial details almost all the time, and thanks to certain team members and their hard work, we were able to reach this situation, where we can showcase the product demo.

Snapshots (Group):

Snapshot 1.1 (Week 3):

Product Backlog and Task Board:

The screenshot shows a Product Backlog interface with two columns of tasks:

Category	Task Description	Added By
Product Backlog	Research dataset attributes and their meaning. Will be necessarily when parsing/pre-processing data to improve whatever algorithms chosen.	a1671245
Product Backlog	documentation of the project architecture	a1783911
Product Backlog	As a user, I can see the malic acid conversion progress on the dashboard	a1789295
Product Backlog	Handle the real-time data (blocked on the implementation of the algorithm)	Vanditg
Product Backlog	As a user, I can see the fermentation progress on the dashboard	a1789295
Product Backlog	Implement algorithms for Outlier detection	a1789295
Product Backlog	Research on the dataset	
Task Board	Research on the dataset	a1789295
Task Board	Design a master dashboard to visualize the real-time analysis of the above mentioned models.	a1789295
Task Board	Develop a model for tracking Malic acid conversion (start, progress, finish)	a1789295
Task Board	Develop a model for tracking the fermentation process (start, progress, finish)	a1789295
Task Board	Develop a model for outlier detection	a1789295

Figure 2. Product Backlog

The screenshot shows a Task Board interface with five columns:

Column	Task Description	Points	Assignee	Status
Product Backlog	Research dataset attributes and their meaning. Will be necessarily when parsing/pre-processing data to improve whatever algorithms chosen.	2		
Product Backlog	documentation of the project architecture	2		
Product Backlog	As a user, I can see the malic acid conversion progress on the dashboard	2		
Product Backlog	Handle the real-time data (blocked on the implementation of the algorithm)	2	Vanditg	
Product Backlog	As a user, I can see the fermentation progress on the dashboard	2		
Product Backlog	Implement algorithms for Outlier detection	2		
Product Backlog	Research on the dataset	2		
Sprint Backlog	Research on the tech architecture	2		
Sprint Backlog	Finalize the tech stack for infra	2		
Sprint Backlog	Decide on the tech stack end to end	2		
To do	Design UI for a dashboard	3		
In progress	Implement front-end for the dashboard (based on the designs) (using static data)	8		
Done				

Figure 3. Task Board

Sprint Backlog and User Stories:

Current User Stories:

- **Research on the tech architecture:**

We want to research possible architectures for accessing and interfacing with the dataset, as well as processing the real time data, and architecture for how the user will access and interact with the tool.

Acceptance Criteria:

We have researched and tentatively decided upon architectures that we will use.

- **Decide on a tech stack end to end:**

We want to decide on a tech stack, such as programming languages & IDE for the front end (UI & Website) and backend modelling (data accessing and processing).

Acceptance Criteria:

We have decided as a group on a tech stack for end to end.

- **Finalizing tech stack for infra:**

We also want to finalize our decisions for tech stack for the infra part of the design.

Acceptance Criteria:

We have decided and finalized our infra tech stack.

- **Design UI for a dashboard:**

We want to have designed and reviewed a UI/GUI for a user dashboard, where the user will interact with the tool.

Acceptance Criteria:

We have designed and reviewed a UI that the group collectively accepts.

Definition of Done:

Please find below the definition of done for each user story, in sprint 1:

Research on the tech architecture: This story will be considered done when we have finalized on:

- What all layers will be there in the entire projects: Like Front end, server, backend, database.
- How the data will flow between those layers. Like how the back end will get the data from database, how will it pass on to the front end.

Finalize the tech stack for Infra: This story will be considered done when we have closed on:

- The technologies on which we will build our server.
- The IDE which will be used to build up the server.

Decide on the tech stack end to end: This story will be considered done when we have closed on:

- The technologies on which we will be building up the backend.
- The technologies on which we will be building up the front end.
- Any additional technology required to establish the communication between layers.

Design UI for a dashboard: This story will be considered done when we have finalized the look and feel of the dashboard.

Implement frontend for the dashboard: This story will be considered done once we have implemented the code for the dashboard according to the design finalized in the above point.

Summary of changes:

Since this was the first week of our first sprint, the snapshot has been created for the very first time. In this snapshot, we have built our product backlog by creating all the relevant user stories to the project. Then depending on the team's capacity and the requirements of the project in hand, we chose the stories from product backlog to be put in sprint backlog. For each ticket in the sprint backlog, we mutually agreed on definition of done and the allocated the sprint points to each ticket depending on their complexities. Some stories which did not have dependence on other tickets were moved to 'to-do' state.

Snapshot 1.2 (Week 4):

Product Backlog and Task Board:

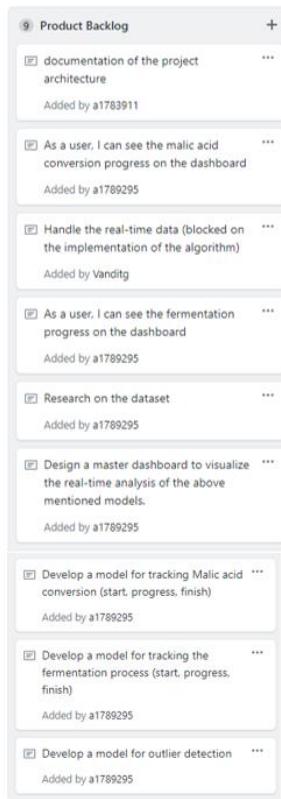


Figure 4. Product Backlog

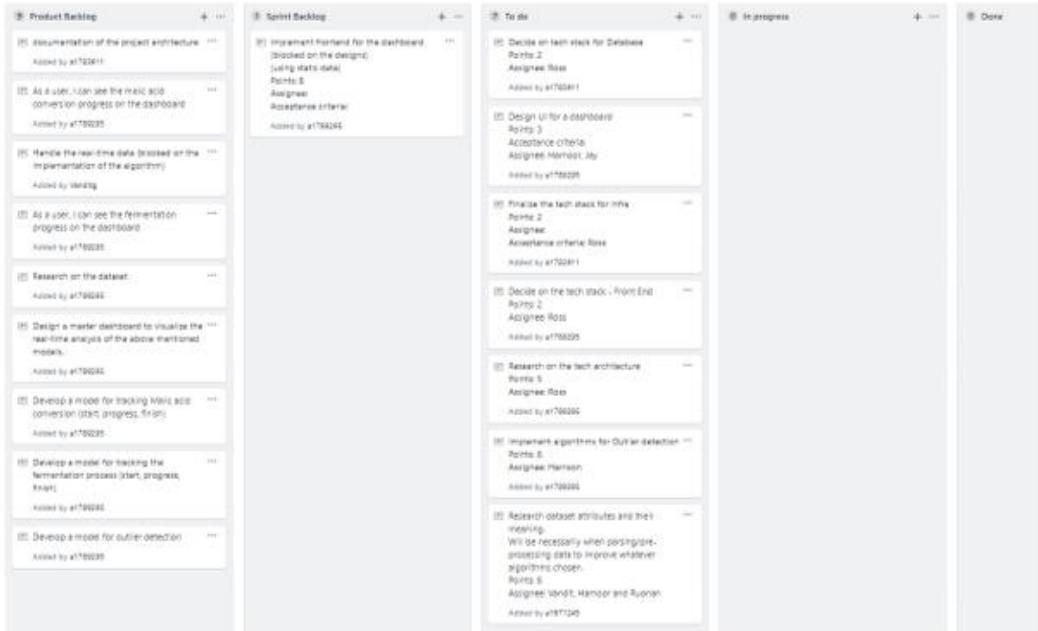


Figure 5. Task Board

Sprint Backlog and User Stories

- **Implement Frontend for the dashboard:**

We need to implement a design of the dashboard to be used as a front end for the user to interact with.

Acceptance Criteria:

We have a working front end that can be interacted with.

- **Decide on tech stack for Database**

We need to decide on a tech stack, such as a programming language (SQL or pandas for example) to be used as a database. Alternatively, we need to decide if a database is the best option for this project.

Acceptance Criteria:

We have decided on a tech stack for the database.

- **Design UI for a dashboard:**

We want to have designed and reviewed a UI/GUI for a user dashboard, where the user will interact with the tool.

Acceptance Criteria:

We have designed and reviewed a UI that the group collectively accepts.

- **Finalize the tech stack for infra:**

We also want to finalize our decisions for tech stack for the infra part of the design.

Acceptance Criteria:

We have decided and finalized our infra tech stack.

- **Decide on Tech stack for Frontend:**

We want to decide on a tech stack, such as a programming language and which IDE to use.

Acceptance Criteria:

We as a group have decided on a tech stack to use for front end development.

- **Research on the tech architecture:**

We want to research possible architectures for accessing and interfacing with the dataset, as well as processing the real time data, and architecture for how the user will access and interact with the tool.

Acceptance Criteria:

We have researched and tentatively decided upon architectures that we will use

- **Implement the algorithm for outlier detection:**

There may be outliers for any machine learning algorithm we implement. We want to create another algorithm to detect and remove these outliers so we can get more precise results.

Acceptance Criteria:

We have an algorithm that successfully identifies and removes outliers.

- **Research Dataset Attributes and their meaning:**

The dataset contains several attributes that determine the value of the label. Understanding these labels will give the project greater significance and will give various results more meaning. We want to determine what each attribute means.

Acceptance Criteria:

We have a description and an understanding of what each dataset attribute means.

Definition of Done:

Please find below the definition of done for each user story, in sprint 1:

Implement Frontend for the dashboard: This story will be considered done when we have finalized on:

- Design the front-end architecture of the core product.
- Realize and continue to optimize.

Determine the tech stack of the database: This story will be considered done when we have evaluated the pros and cons of several programming languages for the project and selected the appropriate language after weighing them.

Finalize the tech stack for infra: This story will be considered done when we have closed on:

- The technologies on which we will building our server.
- The IDE which will be used to build up the server.

Design UI for a dashboard: This story will be considered done when we have finalized the look and feel of the dashboard.

Implement frontend for the dashboard: This story will be considered done once we have implemented the code for the dashboard according to the design finalized in the above point.

Research on the tech architecture: This story will be considered done once we have found an architecture for accessing and interfacing with data sets and processing real-time data.

Implement the algorithm for outlier detection: This story will be considered done when we have closed on: The algorithm based on Anomaly detection could normally accomplished with related statistics.

Research Dataset Attributes and their meaning: This story will be considered done when we have found enough resources for dataset attributes to support a dataset as flexible as it could for different algorithms.

Summary of changes:

This was the second week of our first sprint, the snapshot has been contributed by each group members. In this snapshot, we have built our product backlog by creating all the relevant user stories to the project. Then based on the abilities and academic preferences of the group members, we chose the stories from product backlog to be put in sprint backlog. We refined the task and divided the sprint backlog into many "TODO" cards. These detailed decisions and research make the project completed without omission. Compared with the first snapshot showing the contents of the big frame, the second snapshot is more specific and facilitates the progress of the project.

Declaration: I have attended sprint 1 planning meeting on 10th August 2020 and sprint 1 retrospective meeting/sprint 2 planning on 26th August 2020 with the tutor Mr. Navpreet Singh Ahuja.

Snapshot 2.1 (Week 5):

Product Backlog and Task Board:

7 Product Backlog +

- As a user, I can see the malic acid conversion progress on the dashboard ...
Added by a1789295
- Handle the real-time data (blocked on the implementation of the algorithm) ...
Added by Vanditg
- As a user, I can see the fermentation progress on the dashboard ...
Added by a1789295
- Design a master dashboard to visualize the real-time analysis of the above mentioned models. ...
Added by a1789295
- Develop a model for tracking Malic acid conversion (start, progress, finish) ...
Added by a1789295
- Develop a model for tracking the fermentation process (start, progress, finish) ...
Added by a1789295
- Develop a model for outlier detection ...
Added by a1789295

Figure 6. Product Backlog

Start Building	10 min	10 min	10 min	10 min
<ul style="list-style-type: none">Research on my missionAccessories	<ul style="list-style-type: none">Research into capacity building initiativesIdentify a source of free/low-cost training and resourcesResourcesLanguage ToolsAssessment CriteriaPlanning checklist of resourcesIdentifying the objectives in role and defining key role milestonesResource locationProfessional developmentAccessories	<ul style="list-style-type: none">Identify mission criteriaUnderstand requirements/designResourcesAnytime toolsAnytime toolsAssessment criteriaAgree upon criteria for assessment and evaluation designIntroduce at T1000	<ul style="list-style-type: none">Identify the tools need for this rolePowerPointGoogle SheetsGoogle SheetsAssessmentsChecklistResearch on the next leadership task/roleIntroduce at T1000	<ul style="list-style-type: none">Design a new assessmentPowerPointGoogle SheetsGoogle SheetsAssessmentsChecklistResearch on the next leadership task/roleIntroduce at T1000
<ul style="list-style-type: none">Identified resources for implementationBased on the designUsing toolsKey areasProgramsResourcesAccessoriesAccessories	<ul style="list-style-type: none">Identified key ingredients for step 1NameAssume toolsAssessment CriteriaAt least 10Assume toolsAssume toolsIntroduce the role criteriaIntroduce an action document using a toolIdentify a regular, timely source for private communicationIntroduce at T1000	<ul style="list-style-type: none">Identified signatures for OctoberNameAssume toolsIntroduce at T1000	<ul style="list-style-type: none">Identified resources for Data ProcessingTools and designsAccessAssume toolsAssume toolsAssume toolsAssume toolsIntroducing the resources in roleIntroduce at T1000Google SheetsGoogle SheetsAssessing the resources in roleIntroduce at T1000	<ul style="list-style-type: none">Reassess on the last assessmentReviewAssume toolsAssume toolsAssume toolsAssume toolsIntroduce at T1000
<ul style="list-style-type: none">Identified the required tools and their descriptionsIdentify the necessary when participating in assessing data for required evidence assessment criteriaResourcesPowerPointGoogle SheetsAssessmentsChecklistIntroduce at T1000	<ul style="list-style-type: none">Identify the necessary when participating in assessing data for required evidence assessment criteriaResourcesPowerPointGoogle SheetsAssessmentsChecklistIntroduce at T1000	<ul style="list-style-type: none">Identify the tools need for this rolePowerPointGoogle SheetsGoogle SheetsAssessmentsChecklistResearch on the next leadership task/roleIntroduce at T1000	<ul style="list-style-type: none">Design on the last task - Audit toolPowerPointAssessmentsIntroduce the Google Sheets file hereIntroduce the Google Sheets file at T1000Google SheetsGoogle SheetsIntroduce at T1000	<ul style="list-style-type: none">Design on a checklistPowerPointGoogle SheetsGoogle SheetsAssessmentsChecklistResearch on the next leadership task/roleIntroduce at T1000

Figure 7. Task Board

Sprint Backlog and User Stories:

- Decide on tech stack for Database:**

We need to decide on a tech stack, such as a programming language (SQL or pandas for example) to be used as a database. Alternatively, we need to decide if a database is the best option for this project.

Acceptance Criteria:

We have decided on a tech stack for the database. We have researched and tentatively decided upon architectures that we will use.

- Design UI for a dashboard:**

We want to have designed and reviewed a UI/GUI for a user dashboard, where the user will interact with the tool.

Acceptance Criteria:

We have designed and reviewed a UI that the group collectively accept.

- Finalize the tech stack for infra:**

We also want to finalize our decisions for tech stack for the infra part of the design.

Acceptance Criteria:

We have decided and finalized our infra tech.

- Decide on Tech stack for Frontend:**

We want to decide on a tech stack, such as a programming language and which IDE to use.

Acceptance Criteria:

We as a group have decided on a tech stack to use for front end development.

- Research on the tech architecture:**

We want to research possible architectures for accessing and interfacing with the dataset, as well as processing the real time data, and architecture for how the user will access and interact with the tool.

Acceptance Criteria:

We have researched and tentatively decided upon architectures that we will use.

- **Implement the algorithm for outlier detection:**

There may be outliers for any machine learning algorithm we implement. We want to create another algorithm to detect and remove these outliers so we can get more precise results.

Acceptance Criteria:

We have an algorithm that successfully identifies and removes outliers.

- **Research Dataset Attributes and their meaning:**

The dataset contains several attributes that determine the value of the label. Understanding these labels will give the project greater significance and will give various results more meaning. We want to determine what each attribute means.

Acceptance Criteria:

We have a description and an understanding of what each dataset attribute means

- **Generate Mock of Color scheme/name/logo/design:**

Design of the front-end UI aesthetics that will be visible to the users. This will be the first thing users see so a pleasing design is important.

Acceptance Criteria:

Agreed upon by the entire group

Definition of Done:

The definition for each user story for this week:

- Generate mock color scheme/name/logo/design will be considered done when we have had a presentation to our group and agreed upon a design scheme.
- Implement ML Algorithms for data will be considered done when we have agreed upon and began implementing some experimental Machine Learning algorithms based on the dataset we have found.
- Implement algorithms for Outlier detection will be considered done when we have begun implementing some method of detecting outliers from a dataset.

- Create a skeleton of Data Processing module and diagram will be considered done when we have a file tree and skeleton files for the data processing module.
- Research dataset attributes and their meaning will be considered done when as a group we have agreed upon a basic understanding for what each of the dataset attributes means, as well as their importance for the dataset.

Summary of changes:

In this snapshot, we have formed separate teams: “web team”, “data team” and “product and subject team” to be able to separate tasks to each group member as their skills/preferences fit. This allowed us to tackle some of our product backlog tickets and get started on writing some actual code with a dummy wine fermentation dataset, as we are still waiting on our official data. We have begun research into algorithms for use in our data processing, as well as research into the attributes of the dataset and their importance, as well as some initial sketches of a UI and dashboard, experimenting with static vs dynamic graphs for data reporting.

Snapshot 2.2 (Week 6):

Product Backlog and Task Board:

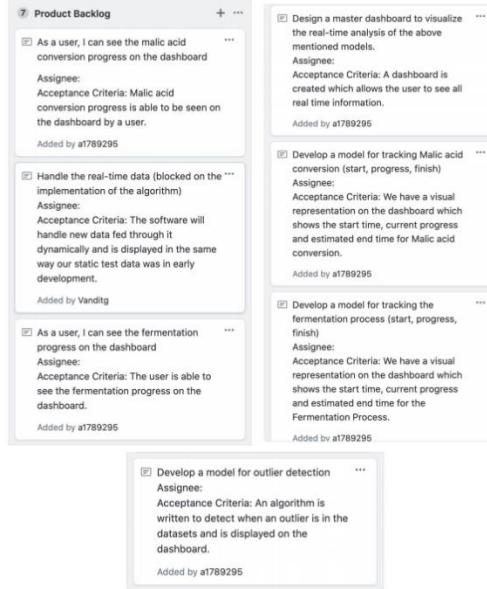


Figure 8. Product Backlog

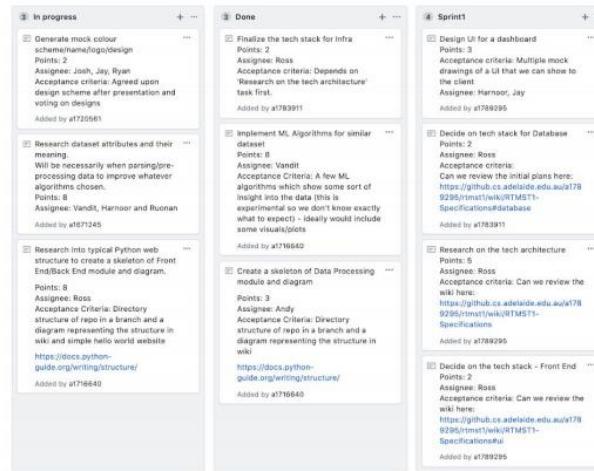
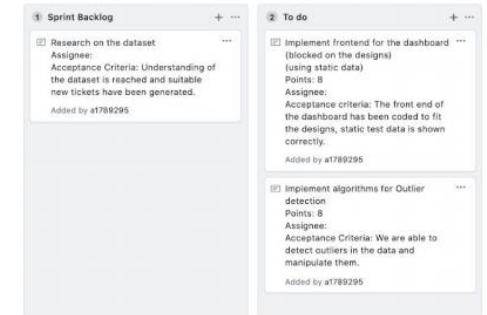


Figure 9. Task Board

Sprint Backlog and User Stories



Figure 10. Sprint Backlog

- **Research on the dataset:**

The dataset will be in a particular format and of a certain size. There may be other factors that influence what machine learning algorithm is most suitable to use. Research these factors.

Acceptance Criteria:

We have a description of the structure of the dataset so we can communicate with the user about the data on a technical level.

- **Implement algorithms for Outlier detection:**

The data will have outliers that will skew the results of our machine learning programs, these outliers need to be removed to provide more precise analysis of the data.

Acceptance Criteria:

We have a definition of an outlier. We have an algorithm that detects outliers which can reliably inform the user when a wine is at risk.

- **Generate mock color scheme/name/logo/design:**

Create some drafts of a User Interface design. Create a name and a logo for the project.

Acceptance Criteria:

As a group we have decided on a name, logo, and design for our project User Interface which would best engage and interest the user.

- **Research dataset attributes and their meanings:**

The dataset contains several attributes that determine the value of the label. Understanding these labels will give the project greater significance and will give various results more meaning. We want to determine what each attribute means.

Acceptance Criteria:

We have a description and an understanding of what each dataset attribute means so that we are able to explain any technical terms to the user.

- **Research into typical Python web structure to create a skeleton of Front End/Back End module and diagram:**

Create a Python project for the Front and Back End of the website. Create a diagram of the interaction between modules.

Acceptance Criteria:

We have a Python project directory setup, including files and subdirectories such that the product is reliable for the user. We have a diagram of module interaction so that any team member is able to explain our products structure to a user.

Definition of Done:

The definition for each ticket for this week:

- Generate mock color scheme/name/logo/design will be considered done when:
 - We have multiple logos/brandings to choose from (**Complete**).
 - The majority of the team agrees on the branding for the product.
- Implement ML Algorithms for data will be considered done when we have (**Complete**):
 - Have an in-depth analysis of the relation between the different features of the dataset.
 - Understand which features are most important.
 - Have a set of machine learning algorithms that can extract meaningful information from the dataset

- Implement algorithms for Outlier detection will be considered done when:
 - We have a series of algorithms which can detect when one of the features is outside of its reasonable bounds.
- Create a skeleton of Data Processing module and diagram will be considered done when (**Complete**):
 - We have a file tree and skeleton files for the data processing module.
 - Every team member is happy with the repository structure.
- Research and develop the code for a simple web application using Python/Django will be considered done when:
 - The skeleton directory structure of a web application is in the repository (**Complete**).
 - We are able to load a simple “Hello World” for the main page.
 - There is suitable documentation on the structure of the code.
- Research dataset attributes and their meaning will be considered done when:
 - As a group we have agreed upon a basic understanding for what each of the dataset attributes means, as well as their importance for the dataset.

Summary of changes:

The previous week of work has been very technically focused and has seen us make good progress on the product’s development. To begin, we found a set of the most effective machine learning algorithms for classifying the quality of wine using a surrogate dataset with a similar problem space and attributes. The development of these algorithms also involved a detailed analysis of the features, many of which we hope will be similar to the actual dataset which we have not yet received. It is our hope that the algorithms developed will transfer effectively to the real dataset. Beyond this, we have finally created a skeleton repository structure for the code, for both the data processing side of the project and the web-application user interface. Beyond the simple skeleton structure we have also almost finished creating the foundations of a simple running website which we will build the dashboard on top of. In the design team we generated a digital mock-up of what

we hope the dashboard may look like based on the previous drawing design. Some members of the product team also generated a series of possible logos/product branding which we will vote on in our next meeting.

Declaration: I have attended sprint 2 planning meeting on 26th August 2020 and sprint 2 retrospective meeting/sprint 3 planning on 9th September 2020 with the tutor Mr. Navpreet Singh Ahuja.

Snapshot 3.1 (Week 7):

Product Backlog and Task Board:

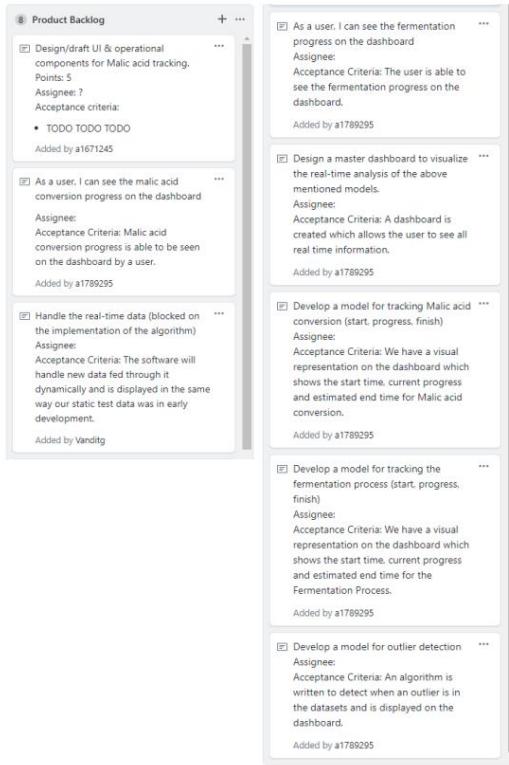


Figure 11. Product Backlog

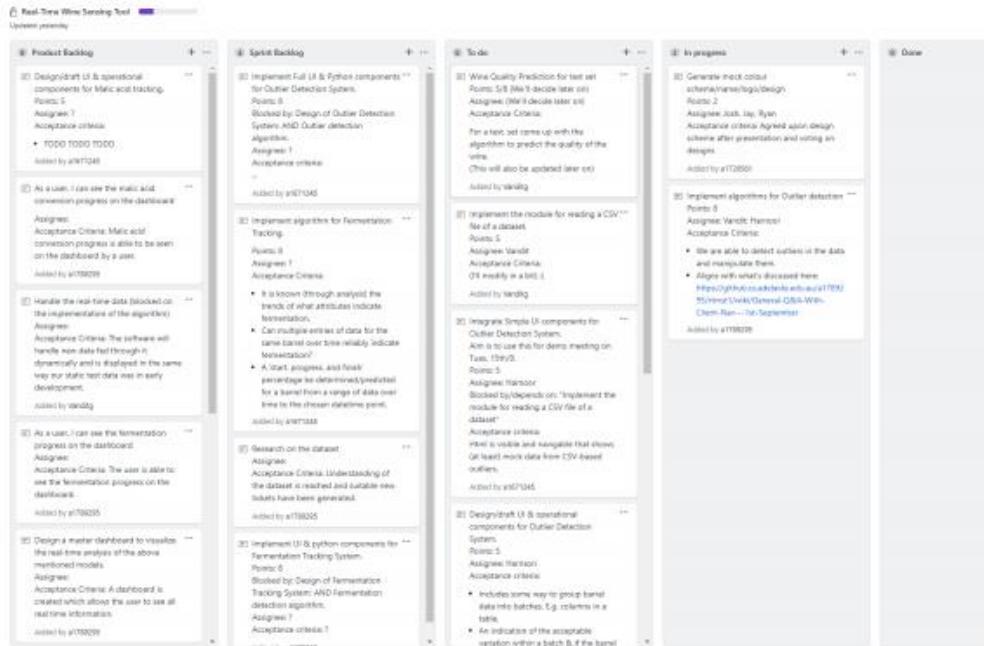


Figure 12. Task Board

Sprint Backlog and User Stories:



Figure 3. Sprint Backlog

- **Wine Quality Prediction for test set:**

Design and implement an algorithm that determines the quality of the wine based on test set data.

Acceptance Criteria: We have an algorithm that can take input data and return a value for the quality of the wine.

- **Implement the module for reading a CSV file of a dataset**

Create a python module for reading input CSV files.

Acceptance Criteria: We have a code module that can parse input files in CSV format.

- **Integrate Simple UI components for Outlier Detection System**

Implement User Interface components that connect to the Outlier Detection algorithm. The aim for this task is to present it in the Demo meeting on the 15th September.

Acceptance Criteria: We have a simple partial interface for the outlier detection system.

- **Design UI & operational components for Outlier Detection System**

Design a user interface that contains the necessary functional components for detecting outliers in the reading of a barrel.

Acceptance Criteria: We have a design of the user interface that includes a method of grouping barrel data, indicates acceptable variance within a batch, presented in the dashboard as a chart, and includes an explanation of how to implement it into code.

- **Design UI & operational components for Fermentation Tracking**

Design a user interface that contains the necessary functional components for tracking fermentation of a barrel.

Acceptance Criteria: The design of the User Interface includes a method of tracking fermentation of the same barrel over time, tracking the average of a batch of barrels.

- **Implement frontend for the dashboard**

Create HTML and CSS code to implement the User Interface as specified in the designs.

Acceptance Criteria: The front end of the dashboard has been coded to fit the designs.

- **Generate mock colour scheme/name/logo/design**

Create some drafts of a User Interface design. Create a name and a logo for the project.

Acceptance Criteria: As a group we have decided on a name, logo, and design for our project User Interface which would best engage and interest the user

- **Implement algorithms for Outlier Detection System**

The data will have outliers that will skew the results of our machine learning programs, these outliers need to be removed to provide more precise analysis of the data.

Acceptance Criteria: We have a definition of an outlier. We have an algorithm that detects outliers which can reliably inform the user when a wine is at risk.

Definition of Done:

The Definition of Done for the sprint, and the sprint is ‘done’ when:

- The sprint was planned and mutually understood, reviewed, and accepted by all members of the team.
- Coding and designed have been completed.
- As many sprint items as possible (described in the previous ‘User Stories’ section) have met their respective acceptance criteria.

- All sprint items that have met their acceptance criteria have been verbally or explained in writing and understood by all.
- All sprint items that have met their acceptance criteria have been moved out of the In Progress or TODO stages.
- All sprint items that have not met their acceptance criteria have been moved into the next sprint or back to the backlog.

Summary of changes:

We concluded Sprint #2 last week and it was quite productive. The team was able to finish many tasks including: finalizing the tech stack for Infra, implementing ML algorithms on similar datasets, creating a folder structure and skeleton of the code base, and furthering the research into the wine data attributes. This week we have officially started with Sprint #3 and the team has once again proactively picked up many challenging tech tickets like: Implementing module for reading the CSV file for dataset, implementing algorithms for outlier detection, and designing UI components for outlier detection and fermentation tracking system, etc. In the last sprint the team tested different algorithms on the data set and out of those, three algorithms were returning promising results. The team will be selecting one of them to move ahead with. The team will also try to implement a UI for outlier detection system. Broadly, it will be like a dashboard that will show the data of the outliers per batch, which will be generated by our ML algorithm. This is expected to be shown to the clients in our first DEMO of this project, on 22nd September 2020.

Snapshot 3.2 (Week 8):

Product Backlog and Task Board:

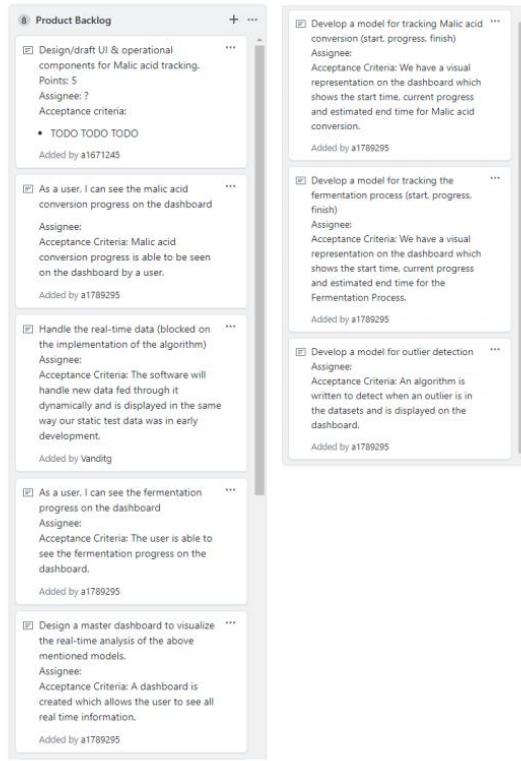


Figure 4. Product Backlog

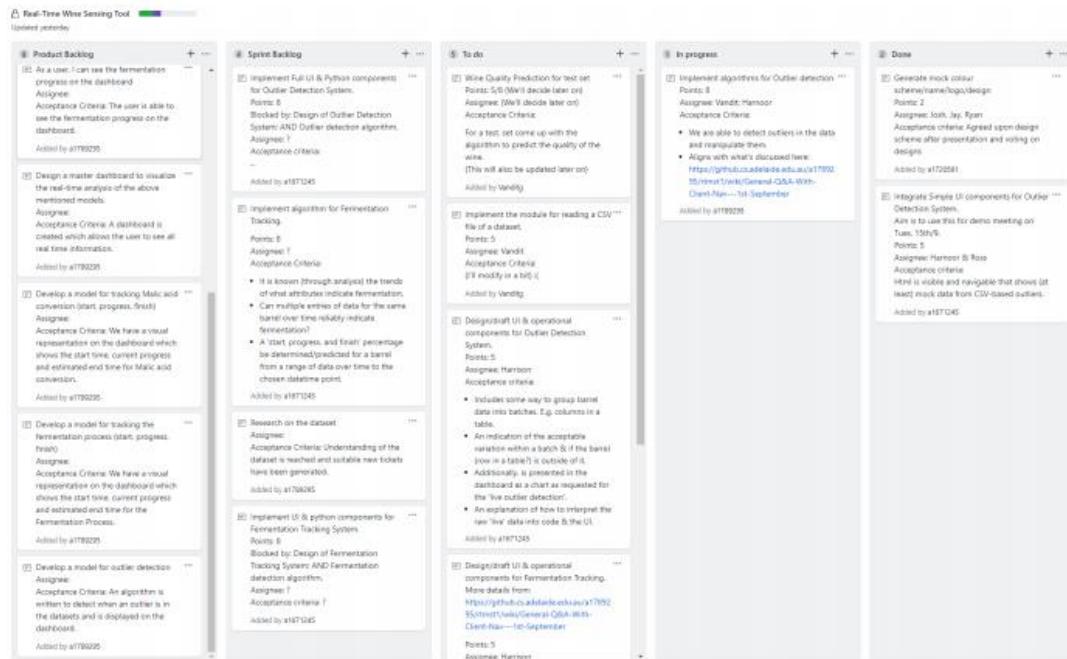


Figure 5. Task Board

Sprint Backlog and User Stories

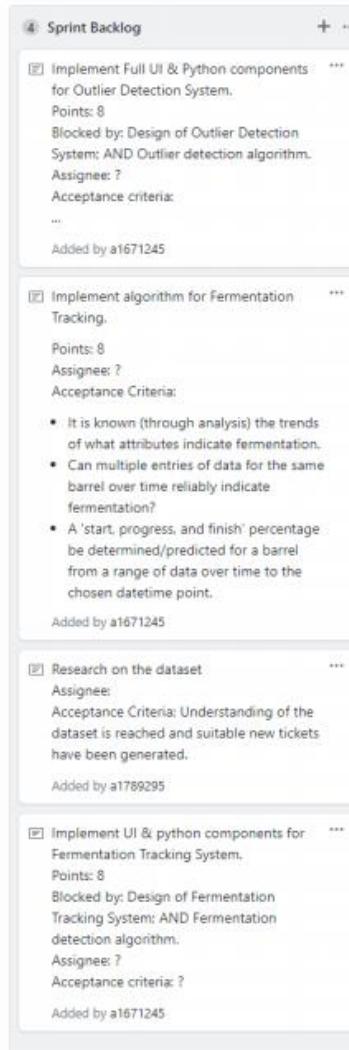


Figure 6. Sprint Backlog

- **Wine Quality Prediction for test set**

Design and implement an algorithm that determines the quality of the wine based on test set data.

Acceptance Criteria: We have an algorithm that can take input data and return a value for the quality of the wine.

- **Implement the module for reading a CSV file of a dataset**

Create a python module for reading input CSV files.

Acceptance Criteria: We have a code module that can parse input files in CSV format.

- **Integrate Simple UI components for Outlier Detection System**

Implement User Interface components that connect to the Outlier Detection algorithm. The aim for this task is to present it in the Demo meeting on the 15th September.

Acceptance Criteria: We have a simple partial interface for the outlier detection system. Task shifted into Done state.

- **Design UI & operational components for Outlier Detection System**

Design a user interface that contains the necessary functional components for detecting outliers in the reading of a barrel.

Acceptance Criteria: We have a design of the user interface that includes a method of grouping barrel data, indicates acceptable variance within a batch, presented in the dashboard as a chart, and includes an explanation of how to implement it into code.

- **Design UI & operational components for Fermentation Tracking**

Design a user interface that contains the necessary functional components for tracking fermentation of a barrel.

Acceptance Criteria: The design of the User Interface includes a method of tracking fermentation of the same barrel over time, tracking the average of a batch of barrels.

- **Implement frontend for the dashboard**

Create HTML and CSS code to implement the User Interface as specified in the designs.

Acceptance Criteria: The front end of the dashboard has been coded to fit the designs.

- **Generate mock colour scheme/name/logo/design**

Create some drafts of a User Interface design. Create a name and a logo for the project.

Acceptance Criteria: As a group we have decided on a name, logo, and design for our project User Interface which would best engage and interest the user Task shifted into Done state.

- **Implement algorithms for Outlier Detection System**

The data will have outliers that will skew the results of our machine learning programs, these outliers need to be removed to provide more precise analysis of the data.

Acceptance Criteria: We have a definition of an outlier. We have an algorithm that detects outliers which can reliably inform the user when a wine is at risk.

Aligns with what's discussed here:

<https://github.cs.adelaide.edu.au/a1789295/rtmst1/wiki/General-Q&A-With-Client-Nav---1st-September>

Definition of Done:

Below states our Definition of Done for the sprint. The sprint is ‘done’ when:

- The sprint was planned and mutually understood, reviewed, and accepted by all members of the team.
- Coding and designed have been completed.
- All or as many sprint items as possible (described in the previous ‘User Stories’ section) that were in the state of To Do or In Progress have met their respective acceptance criteria.
- All sprint items that have met their acceptance criteria have been verbally or explained in writing and understood by all.
- All sprint items that have met their acceptance criteria have been moved to Done status.
- All sprint items that have not met their acceptance criteria have been moved into the next sprint or back to Sprint Backlog.

Summary of changes:

A functional UI for outlier detection system was implemented for the aim of demonstrating it during the meeting with the client on the 15th of September.

Attendance was impossible however, due to all our group members being occupied with other study and work commitments. Additionally, a logo, design and color scheme were voted upon to allow the progression of another dependent task. These two tasks were moved into Done.

Declaration: I have attended sprint 3 planning meeting on 9th September 2020 and sprint 3 retrospective meeting/sprint 4 planning on 7th October 2020 with the tutor Mr. Navpreet Singh Ahuja.

Snapshot 4.1 (Week 9):

Product Backlog and Task Board:



Figure 13. Product Backlog

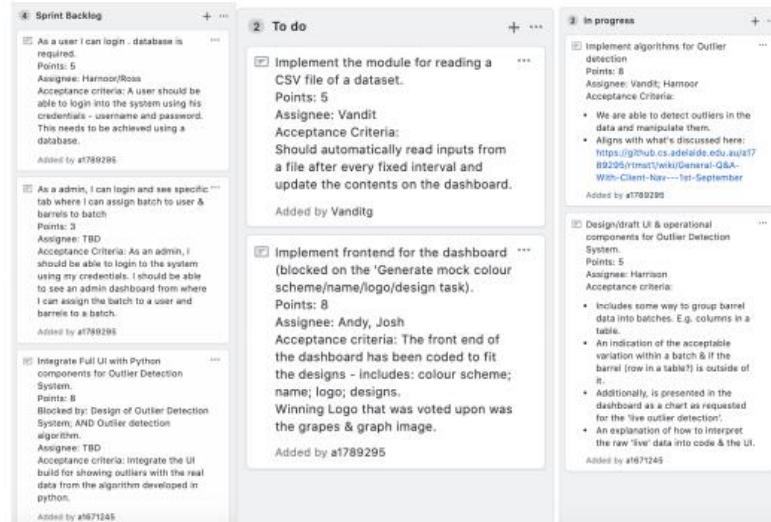


Figure 14. Task Board

Sprint Backlog and User Stories:

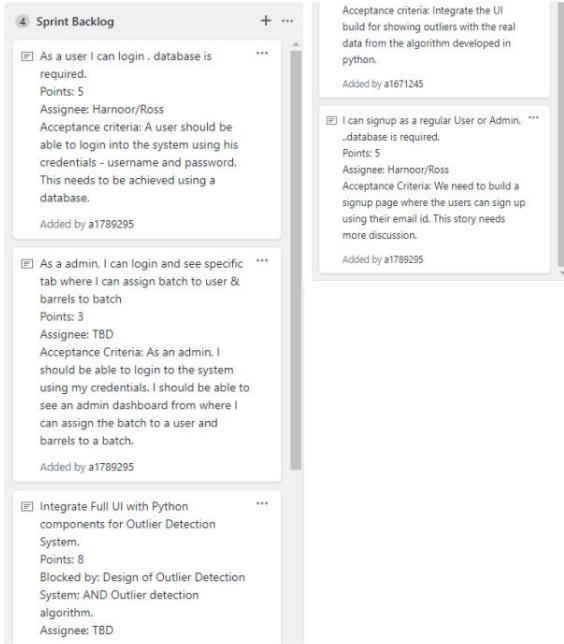


Figure 3. Sprint Backlog

▪ Wine Quality Prediction for test set:

Design and implement an algorithm that determines the quality of the wine based on test set data.

Acceptance Criteria: We have an algorithm that can take input data and return a value for the quality of the wine.

▪ Implement the module for reading a CSV file of a dataset

Create a python module for reading input CSV files.

Acceptance Criteria: We have a code module that can parse input files in CSV format.

▪ Design UI & operational components for Outlier Detection System

Design a user interface that contains the necessary functional components for detecting outliers in the reading of a barrel.

Acceptance Criteria: We have a design of the user interface that includes a method of grouping barrel data, indicates acceptable variance within a batch, presented in the dashboard as a chart, and includes an explanation of how to implement it into code.

- **Design UI & operational components for Fermentation Tracking**

Design a user interface that contains the necessary functional components for tracking fermentation of a barrel.

Acceptance Criteria: The design of the User Interface includes a method of tracking fermentation of the same barrel over time, tracking the average of a batch of barrels.

- **Implement frontend for the dashboard**

Create HTML and CSS code to implement the User Interface as specified in the designs.

Acceptance Criteria: The front end of the dashboard has been coded to fit the designs.

- **Generate mock color scheme/name/logo/design**

Create some drafts of a User Interface design. Create a name and a logo for the project.

Acceptance Criteria: As a group we have decided on a name, logo, and design for our project User Interface which would best engage and interest the user

- **Implement algorithms for Outlier Detection System**

The data will have outliers that will skew the results of our machine learning programs, these outliers need to be removed to provide more precise analysis of the data.

Acceptance Criteria: We have a definition of an outlier. We have an algorithm that detects outliers which can reliably inform the user when a wine is at risk.

Align with what's discussed here:

<https://github.cs.adelaide.edu.au/a1789295/rtmst1/wiki/General-Q&A-With-Client-Nav---1st-September>

- **Implement user login**

As a user, I can login to view my batches data, using a username and password.

Acceptance criteria: We have implemented a database and enabled user logins using credentials

- **Implement admin dashboard**

As an admin, I can log in and see specific tabs where I can assign barrels to batches and batches to specific users.

Acceptance criteria: We have implemented an admin dashboard, where admins can login using their credentials and see different tabs for assigning barrels to batches and batches to users.

- **User/admin sign-up**

As a user or an admin, we can sign up using an email ID.

Acceptance criteria: Users and admins can create an account using an email ID.

Definition of Done:

Please find below the definition of done for each user story, in sprint 4:

- The sprint was planned and mutually understood, reviewed, and accepted by all members of the team.
- Coding and designed have been completed.
- All or as many sprint items as possible (described in the previous ‘User Stories’ section) that were in the state of To Do or In Progress have met their respective acceptance criteria.
- All sprint items that have met their acceptance criteria have been verbally or explained in writing and understood by all.
- All sprint items that have met their acceptance criteria have been moved to Done status.
- All sprint items that have not met their acceptance criteria have been moved into the next sprint or back to Sprint Backlog.

Summary of changes:

This last snapshot was the two weeks leading up to the holidays, so motivation was a little low and not a whole heap of tasks actually got completed. However the majority of the tasks were larger ones so while they were not actually completed progress was made on them. This includes the User Interface, the outlier detection.

We have also made the decision and start making progress on a log in and a sign up page as new information has come from the client regarding assigning of batches and barrels to specific users.

Snapshot 4.2 (Week 10):

Product Backlog and Task Board:

The screenshot shows a Product Backlog interface with the following items:

- Design/draft UI & operational components for Fermentation Tracking.** Points: 5. Assignee: Harrison. Acceptance criteria: Do we have a way to interpret multiple records of the same barrel over time and see its Fermentation levels? See a per batch (many barrels per batch) average of fermentation. Able to drill into a batch's average to see individual barrels. An explanation of how to interpret the raw data into python & UI. Added by a179295.
- Implement UI & python components for Fermentation Tracking System.** Points: 8. Blocked by: Design of Fermentation Tracking System AND Fermentation detection algorithm. Assignee: ? Acceptance criteria: ? Added by a179295.
- Research on the dataset** Points: 2. Assignee: ? Acceptance Criteria: Understanding of the dataset is reached and suitable new tickets have been generated. Added by a179295.
- Implement algorithm for Fermentation Tracking.** Points: 8. Assignee: ? Acceptance Criteria: It is known (through analysis) the trends of what attributes indicate fermentation. Can multiple entries of data for the same barrel over time really indicate fermentation? A 'start', 'progress', and 'finish' percentage be determined/predicted for a barrel from a range of data over time to the chosen datetime point. Added by a179295.
- Wine Quality Prediction for test set** Points: 5/8 (Will decide later on) Assignee: (Will decide later on) Acceptance Criteria: For a test, set up come with the algorithm to predict the quality of the wine. (This will also be updated later on) Added by Vandtg.
- Design/draft UI & operational components for Malic acid tracking.** Points: 5. Assignee: ? Acceptance criteria: TODO TODO TODO Added by a179295.
- As a user, I can see the malic acid conversion progress on the dashboard** Points: 2. Assignee: ? Acceptance Criteria: Malic acid conversion progress is able to be seen on the dashboard by a user. Added by a179295.
- Handle the real-time data (blocked on the implementation of the algorithm)** Points: 2. Assignee: ? Acceptance Criteria: The software will handle new data fed through it dynamically and is displayed in the same way our static test data was in early development. Added by Vandtg.
- As a user, I can see the fermentation progress on the dashboard** Points: 2. Assignee: ? Acceptance Criteria: The user is able to see the fermentation progress on the dashboard. Added by a179295.
- Design a master dashboard to visualize the real-time analysis of the above mentioned models.** Points: 2. Assignee: ? Acceptance Criteria: A dashboard is created which allows the user to see all real time information. Added by a179295.
- Develop a model for tracking Malic acid conversion (start, progress, finish)** Points: 2. Assignee: ? Acceptance Criteria: We have a visual representation on the dashboard which shows the start time, current progress and estimated end time for Malic acid conversion. Added by a179295.
- Develop a model for tracking the fermentation process (start, progress, finish)** Points: 2. Assignee: ? Acceptance Criteria: We have a visual representation on the dashboard which shows the start time, current progress and estimated end time for the Fermentation Process. Added by a179295.
- Develop a model for outlier detection** Points: 2. Assignee: ? Acceptance Criteria: An algorithm is written to detect when an outlier is in the datasets and is displayed on the dashboard. Added by a179295.

Figure 4. Product Backlog

The screenshot shows a Task Board interface with the following columns:

- To do**
 - Design/draft UI & operational components for Fermentation Tracking.** Points: 5. Blocked by: Design of Fermentation Tracking System AND Fermentation detection algorithm. Assignee: ? Acceptance criteria: ? Added by a179295.
 - Implement UI & python components for Fermentation Tracking System.** Points: 8. Blocked by: Design of Fermentation Tracking System AND Fermentation detection algorithm. Assignee: ? Acceptance criteria: ? Added by a179295.
 - Research on the dataset** Points: 2. Assignee: ? Acceptance Criteria: Understanding of the dataset is reached and suitable new tickets have been generated. Added by a179295.
 - Implement algorithm for Fermentation Tracking** Points: 8. Assignee: ? Acceptance Criteria: It is known (through analysis) the trends of what attributes indicate fermentation. Can multiple entries of data for the same barrel over time really indicate fermentation? A 'start', 'progress', and 'finish' percentage be determined/predicted for a barrel from a range of data over time to the chosen datetime point. Added by a179295.
 - Wine Quality Prediction for test set** Points: 5/8 (Will decide later on) Assignee: (Will decide later on) Acceptance Criteria: For a test, set up come with the algorithm to predict the quality of the wine. (This will also be updated later on) Added by Vandtg.
- In progress**
 - Implemented Standard for Outlier detection** Points: 2. Blocked by: Wine Quality Prediction for test set. Assignee: ? Acceptance Criteria: As a user, I can see the real-time analysis of the above mentioned models. This needs to be addressed using a database. Added by a179295.
 - Integrate Full UI with Python components for Outlier Detection** Points: 2. Blocked by: Design of Outlier Detection algorithm AND Outlier detection algorithm. Assignee: ? Acceptance Criteria: As a user, I can see the real-time analysis of the above mentioned models. This needs to be addressed using a database. I need to be able to see an admin dashboard from where I can manage the batch in a east and simple to a batch. Added by a179295.
 - Integrate Full UI with Python components for Malic acid tracking** Points: 2. Blocked by: Design of Outlier Detection algorithm AND Outlier detection algorithm. Assignee: ? Acceptance Criteria: As a user, I can see the real-time analysis of the above mentioned models. This needs to be addressed using a database. Added by a179295.
 - Can sign in and register User to Jupyter Notebook** Points: 2. Blocked by: Research on the dataset. Assignee: ? Acceptance Criteria: As a user, I can see the real-time analysis of the above mentioned models. This needs to be addressed using a database. Added by a179295.
- Done**
 - Implemented algorithms for Outlier detection** Points: 2. Blocked by: Wine Quality Prediction for test set. Assignee: ? Acceptance Criteria: We are able to detect outliers in the data and manipulate them. Added by a179295.
 - Implemented algorithm for reading a CSV file of a dataset** Points: 2. Blocked by: Research on the dataset. Assignee: ? Acceptance Criteria: Should automatically read inputs from a file and update the data and update the elements on the dashboard. Added by Vandtg.

Figure 5. Task Board

Sprint Backlog and User Stories

The screenshot shows a 'Sprint Backlog' interface with two items listed:

- As a admin, I can login and see specific tab *** where I can assign batch to user & barrels to batch**
 - Points: 3
 - Assignee: TBD
 - Acceptance Criteria: As an admin, I should be able to login to the system using my credentials. I should be able to see an admin dashboard from where I can assign the batch to a user and barrels to a batch.
- Integrate Full UI with Python components *** for Outlier Detection System.**
 - Points: 8
 - Blocked by: Design of Outlier Detection System; AND Outlier detection algorithm.
 - Assignee: TBD
 - Acceptance criteria: Integrate the UI build for showing outliers with the real data from the algorithm developed in python.

Added by a1789295 and a1671245 respectively.

Figure 6. Sprint Backlog

- **Wine Quality Prediction for test set**

Design and implement an algorithm that determines the quality of the wine based on test set data.

Acceptance Criteria: We have an algorithm that can take input data and return a value for the quality of the wine.

- **Design UI & operational components for Outlier Detection System**

Design a user interface that contains the necessary functional components for detecting outliers in the reading of a barrel.

Acceptance Criteria: We have a design of the user interface that includes a method of grouping barrel data, indicates acceptable variance within a batch, presented in the dashboard as a chart, and includes an explanation of how to implement it into code.

- **Design UI & operational components for Fermentation Tracking**

Design a user interface that contains the necessary functional components for tracking fermentation of a barrel.

Acceptance Criteria: The design of the User Interface includes a method of tracking fermentation of the same barrel over time, tracking the average of a batch of barrels.

- **Implement frontend for the dashboard**

Create HTML and CSS code to implement the User Interface as specified in the designs.

Acceptance Criteria: The front end of the dashboard has been coded to fit the designs.

- **Implement user login**

As a user, I can login to view my batches data, using a username and password.

Acceptance criteria: We have implemented a database and enabled user logins using credentials

- **Implement admin dashboard**

As an admin, I can log in and see specific tabs where I can assign barrels to batches and batches to specific users.

Acceptance criteria: We have implemented an admin dashboard, where admins can login using their credentials and see different tabs for assigning barrels to batches and batches to users.

Definition of Done:

Below states our Definition of Done for the sprint. The sprint is ‘done’ when:

- The sprint was planned and mutually understood, reviewed, and accepted by all members of the team.
- The current snapshot (Described in the ‘User Stories’) have met respective acceptance criteria for individual items.
- CSV Loader and Outlier detection module have been implemented.
- Designing of the UI and merge the outlier detection module with the dashboard in the final stage for completion.

- The implementation for the Dashboard and Sign-in options are currently in working stage.
- The tasks which have been not initiated nor completed have been moved to the next phase for priority tasks.

Summary of changes:

A lot more was completed this last snapshot than the previous one with some substantial progress being made in some tasks and a couple items getting ticked off the list. One of the tasks that some progress was made in was the authentication of the user. The log in and sign up screen will be completed by the end of the next sprint hopefully.

As well as this the UI was completed for the site and the HTML and CSS has been started which also will hopefully be completed in the next two weeks. Some more work has been completed on the outlier reading to help this tie in more correctly with the site.

Declaration: I have attended sprint 4 planning meeting on 7th October 2020 and sprint 4 retrospective meeting/sprint 5 planning on 21st October 2020 with the tutor Mr. Navpreet Singh Ahuja.

Snapshot 5.1 (Week 11):

▪ Product Backlog and Task Board

➤ Product Backlog:

The screenshots of the product backlog can be found below in Figure 15.

The screenshot shows a Product Backlog board with the following items:

- Design/draft UI & operational components for Fermentation Tracking.**
 - More details from: <https://github.cs.adelaide.edu.au/a17892/95/rtmst1/wiki/General-Q&A-With-Client-Nav---1st-September>
 - Points: 5
 - Assignee: Harrison
 - Acceptance criteria:
 - Do we have a way to interpret multiple records of the same barrel over time and see its Fermentation levels?
 - See a per batch (many barrels per batch) average of fermentation.
 - Able to drill into a batch's average to see individual barrels.
 - An explanation of how to interpret the raw data into python & UI.
 - Added by a1671245
- Implement UI & python components for Fermentation Tracking System.**
 - Points: 8
 - Blocked by: Design of Fermentation Tracking System AND Fermentation detection algorithm.
 - Assignee: ?
- Research on the dataset**
 - Assignee:
 - Acceptance Criteria: Understanding of the dataset is reached and suitable new tickets have been generated.
 - Added by a1789295
- Implement algorithm for Fermentation Tracking.**
 - Points: 8
 - Assignee: ?
 - Acceptance Criteria:
 - It is known (through analysis) the trends of what attributes indicate fermentation.
 - Can multiple entries of data for the same barrel over time reliably indicate fermentation?
 - A 'start, progress, and finish' percentage be determined/predicted for a barrel from a range of data over time to the chosen datetime point.
 - Added by a1671245
- Handle the real-time data (blocked on the implementation of the algorithm)**
 - Assignee:
 - Acceptance Criteria: The software will handle new data fed through it dynamically and is displayed in the same way our static test data was in early development.
 - Added by Vanditg
- As a user, I can see the fermentation progress on the dashboard**
 - Assignee:
 - Acceptance Criteria: The user is able to see the fermentation progress on the dashboard.
 - Added by a1789295
- Design a master dashboard to visualize the real-time analysis of the above mentioned models.**
 - Assignee:
 - Acceptance Criteria: A dashboard is created which allows the user to see all real time information.
 - Added by a1789295
- Develop a model for tracking Malic acid conversion (start, progress, finish)**
 - Assignee:
 - Acceptance Criteria: We have a visual representation on the dashboard which shows the start time, current progress and estimated end time for Malic acid conversion.
 - Added by a1789295
- Develop a model for tracking the fermentation process (start, progress, finish)**
 - Assignee:
 - Acceptance Criteria: We have a visual representation on the dashboard which shows the start time, current progress and estimated end time for the Fermentation Process.
 - Added by a1789295
- Develop a model for outlier detection**
 - Assignee:
 - Acceptance Criteria: An algorithm is written to detect when an outlier is in the datasets and is displayed on the dashboard.
 - Added by a1789295

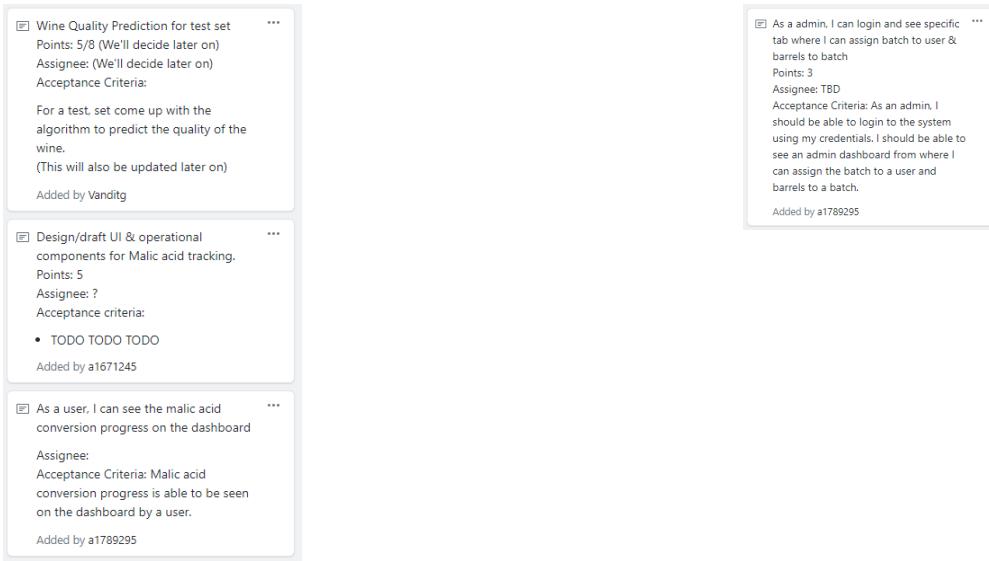


Figure 15. Product backlog for the Snapshot – 5.1 (Week - 11). The order is from top to bottom and left to right.

➤ Task Board:

The screenshots of the Task board can be found below in Figure 16.

Sprint Backlog	To do	In progress
<p><input checked="" type="checkbox"/> Integrate Outlier detection model with the dashboard ... Points: 5 Added by a1789295</p> <p><input checked="" type="checkbox"/> Integrate Full UI with Python components for Outlier Detection System. ... Points: 8 Blocked by: Design of Outlier Detection System; AND Outlier detection algorithm. Assignee: TBD Acceptance criteria: Integrate the UI build for showing outliers with the real data from the algorithm developed in python. Added by a1671245</p> <p><input checked="" type="checkbox"/> Final Presentation ... Added by a1789295</p> <p><input checked="" type="checkbox"/> Final report ... Added by a1789295</p>	<p><input checked="" type="checkbox"/> To do ...</p>	<p><input checked="" type="checkbox"/> In progress ... As a user I can login . A database is required. Points: 5 Assignee: Harnoor/Ross Acceptance criteria: A user should be able to login into the system using his credentials - username and password. This needs to be achieved using a database. Added by a1789295</p> <p><input checked="" type="checkbox"/> I can signup as a regular User or Admin. ..database is required. Points: 5 Assignee: Harnoor/Ross Acceptance Criteria: We need to build a signup page where the users can sign up using their email id. This story needs more discussion. Added by a1789295</p>

Figure 16. Task Board for the Snapshot – 5.1 (Week - 11).

- **Sprint Backlog**

The screenshots of the Sprint Backlog can be below in Figure 17.

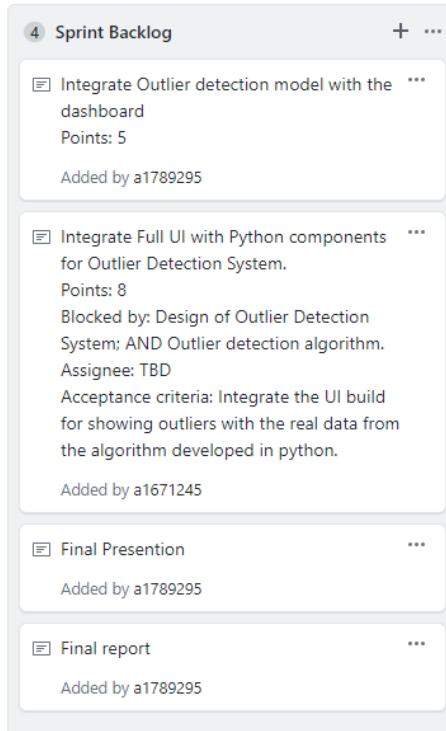


Figure 17. Sprint Backlog for the Snapshot – 5.1 (Week - 11).

- **User Stories**

- **User login module:**

As a user one can able to login into the system with credentials and view the batches in the dashboard.

Acceptance criteria: Implement the database for storing the username and password for different users.

- **Admin dashboard:**

As an admin, one can log into the system with credentials and see specific tabs where one can assign batches to specific users.

Acceptance criteria: Implement the admin dashboard by using the username and password stored in the database and can be able to see the different tabs according to needs.

- **Integration of the CSV Reader module with the dashboard:**

As a user one can log into the system with credentials and can be able to select the different CSV file data and lookout the data in the dashboard.

Acceptance criteria: Integrate the CSV reader module with the dashboard by merging the module with the UI dashboard for effective visualization and the user can view the data of the different CSV files without any interruption from the system.

➤ **Integration of the outlier detection module with the dashboard:**

As a user, one can able to see the batches and by running the outlier detection algorithm, a user can see the outliers for the specific batch and raise the issue for those outliers.

Acceptance criteria: Integrate the outlier detection module and dashboard for detecting the outliers for a specific batch and if possible, add a button to raise the issue for a batch.

➤ **Integration of the full UI with python components for outlier detection system:**

As a user one can able to lookout the batch in the dashboard and by using different outlier detection algorithms raise the issue for outliers in a specific batch.

Acceptance criteria: Integrate the full UI build for showing the outliers with the real data from the algorithm developed using python's open-sourced module.

➤ **Full integration of the system:**

An admin or a user can log in to the system with the credentials and can look out the data in the dashboard for different CSV data. Once after the visualization by running the outlier detection algorithm a user can raise the issue if outliers are detected in the specific batch

Acceptance criteria: Final integration of the different modules and check if the system/framework runs without any interruption.

- **Definition of Done:**

Please find the definition of done for each user story in sprint below:

- The sprint was planned and mutually understood, reviewed, and accepted by all members of the team.
- Coding and design for the dashboard have been completed.
- All or as many sprint items as possible (described in the previous ‘User Stories’ section) that were in the state of To-Do or In Progress have met their respective acceptance criteria.
- All sprint items that have met their acceptance criteria have been verbally or explained in writing and understood by all.
- All sprint items that have met their acceptance criteria have been moved to Done status.
- All sprint items that have not met their acceptance criteria have been moved into the next sprint or back to Sprint Backlog.
- CSV Loader, Outlier detection, and Designing of the UI have been completed, and merging of the outlier detection module with the dashboard is in the final stage for completion.

- **Summary of Changes:**

In comparison with the previous snapshots, this snapshot was one of the best snapshots as we completed several smaller tasks – Final Dashboard Implementation, CSV File Reader, and Outlier Detection Module. One of the tasks that have a major focus now on is the implementation of the login module. Once the login module has been developed the final task is the integration of every component to make the final product. After the development of the final product, the last and final task is to run and test the system whether it is working without interruption.

Snapshot 5.2 (Week 12):

▪ Product Backlog and Task Board

➤ Product Backlog:

The screenshots of the product backlog can be found below in Figure 18.

The screenshot shows a Product Backlog interface with the following items:

- Design/draft UI & operational components for Fermentation Tracking.**
 - More details from: <https://github.cs.adelaide.edu.au/a1789295/rtmst1/wiki/General-Q&A-With-Client-Nav---1st-September>
 - Points: 5
 - Assignee: Harrison
 - Acceptance criteria:
 - Do we have a way to interpret multiple records of the same barrel over time and see its Fermentation 'levels'?
 - See a per batch (many barrels per batch) average of fermentation.
 - Able to drill into a batch's average to see individual barrels.
 - An explanation of how to interpret the raw data into python & UI.
 - Added by a1671245
- Implement UI & python components for Fermentation Tracking System.**
 - Blocked by: Design of Fermentation Tracking System; AND Fermentation detection algorithm.
 - Assignee: ?
 - Points: 8
 - Added by a1789295
- Research on the dataset**
 - Assignee:
 - Acceptance Criteria: Understanding of the dataset is reached and suitable new tickets have been generated.
 - Added by a1789295
- Implement algorithm for Fermentation Tracking.**
 - Points: 8
 - Assignee: ?
 - Acceptance Criteria:
 - It is known (through analysis) the trends of what attributes indicate fermentation.
 - Can multiple entries of data for the same barrel over time reliably indicate fermentation?
 - A 'start, progress, and finish' percentage be determined/predicted for a barrel from a range of data over time to the chosen datetime point.
 - Added by a1671245
- Handle the real-time data (blocked on the implementation of the algorithm)**
 - Assignee:
 - Acceptance Criteria: The software will handle new data fed through it dynamically and is displayed in the same way our static test data was in early development.
 - Added by Vanditg
- As a user, I can see the fermentation progress on the dashboard**
 - Assignee:
 - Acceptance Criteria: The user is able to see the fermentation progress on the dashboard.
 - Added by a1789295
- Design a master dashboard to visualize the real-time analysis of the above mentioned models.**
 - Assignee:
 - Acceptance Criteria: A dashboard is created which allows the user to see all real time information.
 - Added by a1789295
- Develop a model for tracking Malic acid conversion (start, progress, finish)**
 - Assignee:
 - Acceptance Criteria: We have a visual representation on the dashboard which shows the start time, current progress and estimated end time for Malic acid conversion.
 - Added by a1789295
- Develop a model for tracking the fermentation process (start, progress, finish)**
 - Assignee:
 - Acceptance Criteria: We have a visual representation on the dashboard which shows the start time, current progress and estimated end time for the Fermentation Process.
 - Added by a1789295
- Develop a model for outlier detection**
 - Assignee:
 - Acceptance Criteria: An algorithm is written to detect when an outlier is in the datasets and is displayed on the dashboard.
 - Added by a1789295

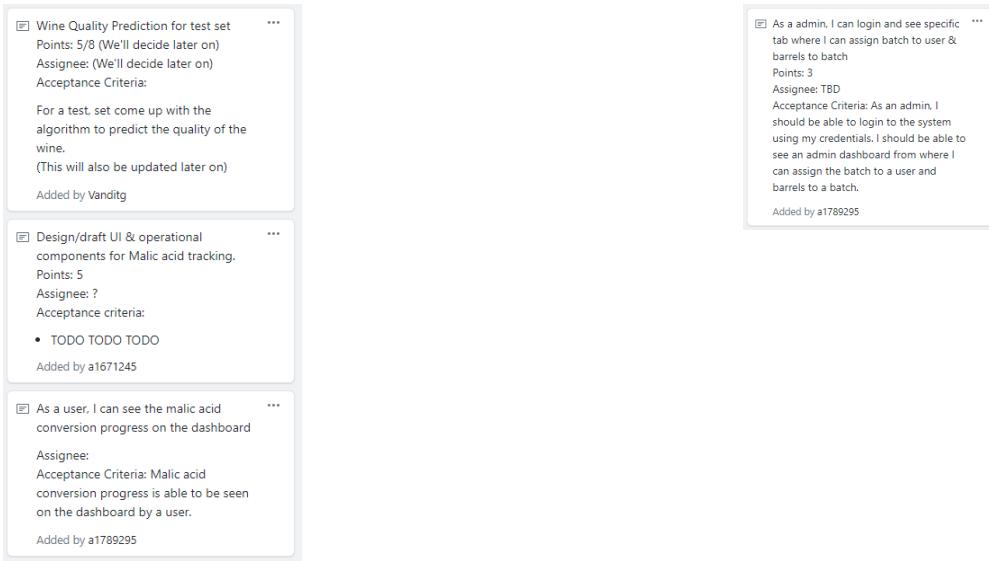


Figure 18. Product backlog for the Snapshot – 5.2 (Week - 12). The order is from top to bottom and left to right.

➤ Task Board:

The screenshots of the Task board can be found below in Figure 19.

Sprint Backlog	To do	In progress
<p>Integrate Outlier detection model with the dashboard</p> <ul style="list-style-type: none"> Points: 5 <p>Added by a1789295</p>		
<p>Integrate Full UI with Python components for Outlier Detection System.</p> <ul style="list-style-type: none"> Points: 8 Blocked by: Design of Outlier Detection System; AND Outlier detection algorithm. Assignee: TBD Acceptance criteria: Integrate the UI build for showing outliers with the real data from the algorithm developed in python. <p>Added by a1671245</p>	<p>I can signup as a regular User or Admin.</p> <ul style="list-style-type: none"> ..database is required. <p>Points: 5</p> <p>Assignee: Harnoor/Ross</p> <p>Acceptance Criteria: We need to build a signup page where the users can sign up using their email id. This story needs more discussion.</p> <p>Added by a1789295</p>	
		<p>Final Presention</p> <p>Added by a1789295</p> <p>Final report</p> <p>Added by a1789295</p>

Figure 19. Task Board for the Snapshot – 5.2 (Week - 12).

- **Sprint Backlog**

The screenshots of the Sprint Backlog can be below in Figure 20.

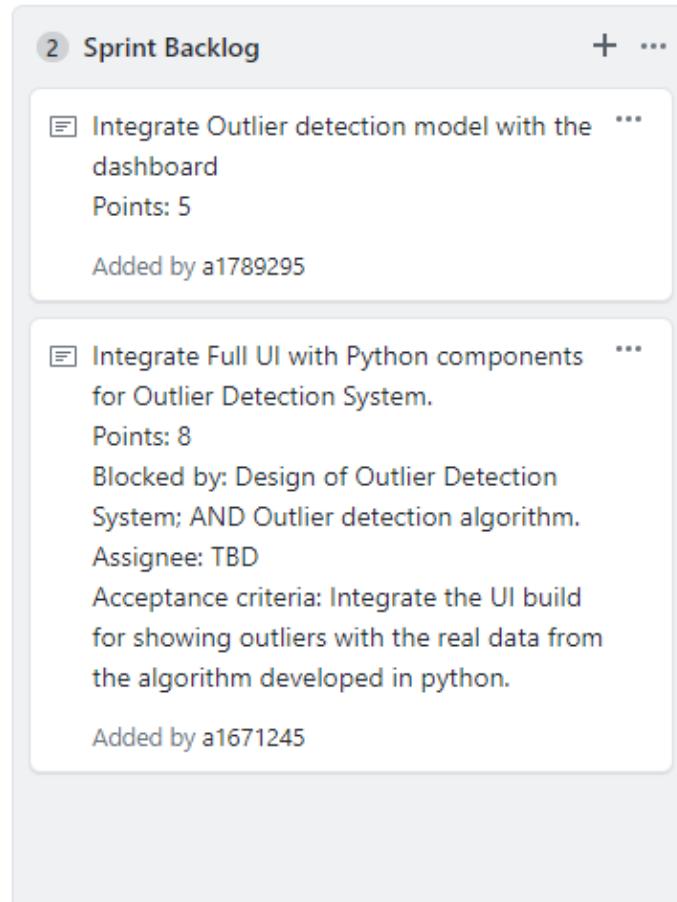


Figure 20. Sprint Backlog for the Snapshot – 5.2 (Week - 12).

- **User Stories**

- **User login module:**

As a user one can able to login into the system with credentials and view the batches in the dashboard.

Acceptance criteria: Implement the database for storing the username and password for different users.

- **Admin dashboard:**

As an admin, one can log into the system with credentials and see specific tabs where one can assign batches to specific users.

Acceptance criteria: Implement the admin dashboard by using the username and password stored in the database and can be able to see the different tabs according to needs.

➤ **Integration of the CSV Reader module with the dashboard:**

As a user one can log into the system with credentials and can be able to select the different CSV file data and lookout the data in the dashboard.

Acceptance criteria: Integrate the CSV reader module with the dashboard by merging the module with the UI dashboard for effective visualization and the user can view the data of the different CSV files without any interruption from the system.

➤ **Integration of the outlier detection module with the dashboard:**

As a user, one can able to see the batches and by running the outlier detection algorithm, a user can see the outliers for the specific batch and raise the issue for those outliers.

Acceptance criteria: Integrate the outlier detection module and dashboard for detecting the outliers for a specific batch and if possible, add a button to raise the issue for a batch.

➤ **Integration of the full UI with python components for outlier detection system:**

As a user one can able to lookout the batch in the dashboard and by using different outlier detection algorithms raise the issue for outliers in a specific batch.

Acceptance criteria: Integrate the full UI build for showing the outliers with the real data from the algorithm developed using python's open-sourced module.

➤ **Full integration of the system:**

An admin or a user can log in to the system with the credentials and can look out the data in the dashboard for different CSV data.

Once after the visualization by running the outlier detection algorithm a user can raise the issue if outliers are detected in the specific batch

Acceptance criteria: Final integration of the different modules and check if the system/framework runs without any interruption.

➤ **Product demo presentation:**

A tester showcases the product in a wise manner to outlines different functionalities and potential use cases through various debugging strategies. Once the functionalities have been shown, a future development plan will be provided.

Acceptance criteria: Provide an effective product demo presentation that showcases the potential of use cases and future development.

➤ **Final process presentation:**

The team will provide an effective presentation of the complete product development process where how the team has handled the different scenarios and came up with an efficient and optimized solution through agility.

Acceptance criteria: Provide a thorough presentation that provides the different aspects of the product development process.

▪ **Definition of Done:**

Please find the definition of done for each user story in sprint below:

- The sprint was planned and mutually understood, reviewed, and accepted by all members of the team.
- Coding and design for the dashboard have been completed.
- All or as many sprint items as possible (described in the previous ‘User Stories’ section) that were in the state of To-Do or In Progress have met their respective acceptance criteria.
- All sprint items that have met their acceptance criteria have been verbally or explained in writing and understood by all.

- All sprint items that have met their acceptance criteria have been moved to Done status.
- All sprint items that have not met their acceptance criteria have been moved into the next sprint or back to Sprint Backlog.
- CSV Loader, Outlier detection, and Designing of the UI have been completed, and merging of the outlier detection module with the dashboard is in the final stage for completion.
- In the final week, the team decided to stop the implementation and work on the product demo and final presentation due to interruption in some tasks.
- The team divided the presentation tasks to everyone so that each of the team members will be able to present their opinions.

- **Summary of Changes:**

The week-12 snapshot was not effective as most of the tasks were remained incomplete due to interruption in certain tasks and the team decided to move on to the presentation of the product and development process. One of the team members has implemented a tiny module of user login. Once the team tested the final product, agreed-upon each of the team members' permission, the process presentation tasks have been divided in a wise manner to provide an opportunity for the team member to select one of the topics from the final product development process presentation. Finally, after the selection of the points to speak on, the team members have discussed the future development of the product.

Declaration: I have attended the sprint 5 planning meeting on 21st October 2020 and provided a Final Presentation on 4th November 2020 with the tutor Mr. Navpreet Singh Ahuja.

Personal Reflection on Software Process (Individual):

To begin with, the team's initial process for tackling the problem was unorganized and because of this, we faced difficulties for the overall implementation and there were significant delays in task completion. However, the team's efforts were tremendous, and the team started to explore the different technologies based on their knowledge after our initial planning meeting. The team has progressed gradually by defining the tasks and finalizing the tech stack for front-end and back-end development. The only issue the team faced was regarding the dataset. The dataset was not provided to the team because of legal issues. Although the team had finalized one of the open-source datasets for initial models and had successfully implemented the initial product with basic functionalities. One of the main issues our team had faced was when the client notified that the dataset will not be provided to the team and there will be a change in the task definition, but the team had handled this challenge successfully and worked hard throughout the final product development. At the end of the project timeframe, the team had successfully implemented the final product with only one functionality left, which would be included in our future work for the development.

After adopting the Scrum (Agile) framework for the software implementation process for this project, I believe that it is the most efficient and effective approach for our project. The Agile framework gave us the elasticity to familiarize the client's new task definitions without major changes in our product development. In contrast with the waterfall-approach, I believe that would have been helpful for the projects which require a complete vision of the end product and processes. We could have gained insights from this framework during our initial phase. It would have helped me better understand the project requirement (help analyze required technologies for the system) and task along with its deadline. So in my opinion, the combination of the Waterfall-approach for initial planning, and the Scrum (Agile) approach for project development.

After working tirelessly and successfully implementing the project, I was able to learn how to work with the client more professionally. Facing the issues and resolve them instantly had taught me the skill of debugging. I also had learned that when working in a group, each team member holds equal importance. A project does not work without representation from each role and similarly does not work if resourcing is not shared equally amongst the different resource types. I have learned that even though it is the developer who writes the actual code, there would be no need for code without stakeholders, and there would be no stakeholders without quality assurance of what they stand over. One of the most important lessons I learned regarding the software process is that Process change is hard, especially when you are busy doing the actual project. But if you do not do it today, you probably will not have the inclination to put those lessons into practice for the future.

References:

- [1] Lee, Seunghan, Juyoung Park, and Kyungtae Kang. "Assessing wine quality using a decision tree." 2015 IEEE International Symposium on Systems Engineering (ISSE). IEEE, 2015..