Project Deliverable G

Prototype II and Customer Feedback

Marie Levin, Leila Smaili, Rafiullah Hamdard, Hanna Paik, and
Yifei Li

March 12, 2023

Abstract

This document contains Project Deliverable G. This document aims to show our second prototype of the product. This prototype was created according to the client's needs/requests, and through user/technical benchmarking. Moreover, this document contains the customer feedback we got during our third meeting with the client and a test plan for our next prototype.

Table of Content

1.0 Introduction

Robert Ritchie has tasked our design team to create a product that displays the specific conveyor speeds that optimize the yield of beer from his manufacturing line. Following the initial client meeting, our team developed the following problem statement.

“A need exists for Robert Ritchie and his fellow supervisors to find the optimal speeds of their beer packaging process to ensure it is “always at top efficiency (Project Background, Brightspace)”. The solution must have an attractive and straightforward interface that uses a flexible algorithm based on the V-Curve Theory to report optimized speeds of each unit and make recommendations on how to achieve them.”

From this problem, we developed design criteria and metrics that will be used to measure our design’s ability to solve our client’s problem. With research, benchmarking, and brainstorming, the first conceptual design was developed and it was presented during the second client meeting.

After we presented our presentation to the client in our second meeting and he expressed interest in our conceptual design, our team developed the first prototype of the product. This prototype was created based on the feedback that we got from the client meeting, following the client’s needs, and meeting the design criteria.

In our third meeting with the client, we presented our first prototype. After this presentation, we received positive feedback and the client expressed great interest.

In this deliverable, we will be creating a test plan based on the client’s specific feedback. This will allow us to establish our specific goals for testing different aspects of our second prototype. Moreover, this testing plan will help us improve our second prototype and the overall design of the product.

2.0 Feedback of First Prototype

The client expressed interest in our first prototype during the recent meeting. This allows us to move on to the next stage and improve our designs.

3.0 Objectives of Prototype II

We have updated our prototype based on the plan outlined in the previous deliverable.

The following updates were made:

1. The prototype was created using Xcode based on the plan. However, Swift language was used instead of Storyboard mode.
2. The UI was updated and integrated with algorithms.

The current prototype is able to:

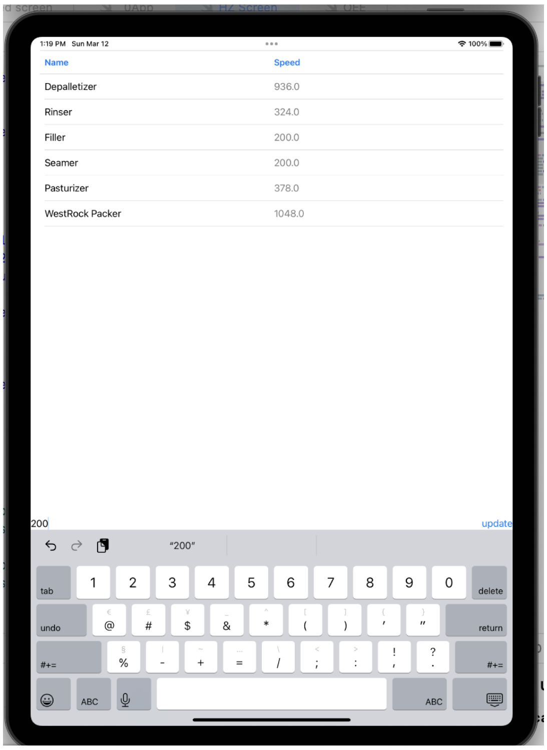
1. Switch from Speed, Hz and OEE windows with a tab bar.
2. calculate the desired optimal speed with a given input as shown in graph 1 (ex. 200 as input on the graph).
3. Update the OEE (overall equipment efficiency) as shown in graph 2. However, a random max speed was used during the calculation because of the lack of max speed data.

NOTE: Calculation is currently not available for Hz windows.

3.0.1 Prototype 2 UI

The UI was able to integrate the following functions:

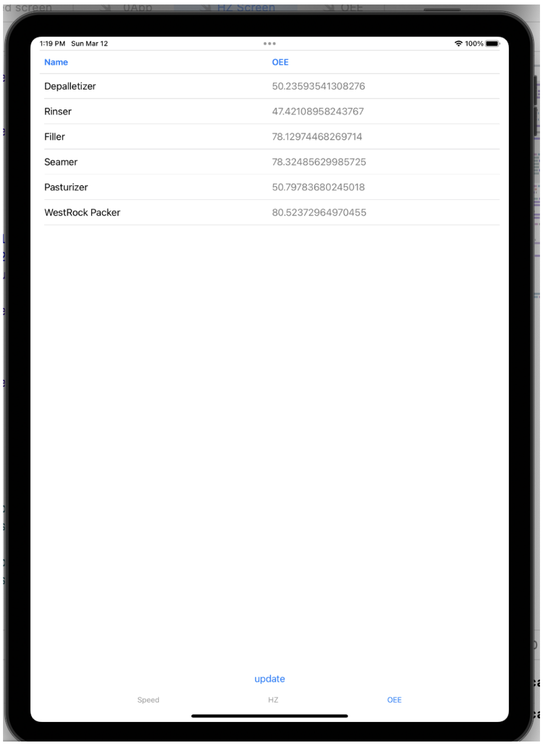
- Text input
- Update button (updates a corresponding table with text input)
- Table view of data
- Tabs for each unit of measurement for display (Cpm Hz OEE)



Name	Speed
Depalletizer	936.0
Rinser	324.0
Filler	200.0
Seamer	200.0
Pasturizer	378.0
WestRock Packer	1048.0

200

Figure 1 Screenshot of prototype (Speed Tab)



Name	OEE
Depalletizer	50.23593541308276
Rinser	47.42108958243767
Filler	78.12974468269714
Seamer	78.32485629985725
Pasturizer	50.79783680245018
WestRock Packer	80.52372964970455

Graph 2 Screenshot of OEE windows

3.0.2 Prototype 2 Algorithm

The algorithm used to calculate the unit speeds of the production line was updated to be more accurate across a larger scale of filler speeds. First, the old algorithm for the V-Curve Theory did not accurately represent the V-Curve Theory, which we found to be more of an arithmetic series rather than a geometric one. The 10% increment of speed from the unit before and after the filler should be scaled by a percent increase of the filler speed, and not 10% increments of the speed of the unit before it, as was previously done.

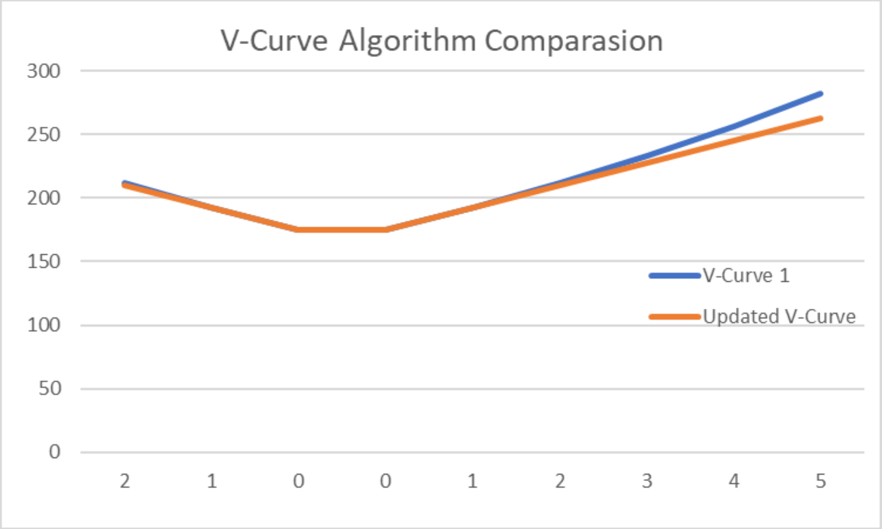


Figure 1. Shows the difference in calculations of the V-Curve Theory

$$Old\ Method = fillerSpeed * 1.1^{n}$$

$$New\ Method = fillerSpeed * (1 + \frac{n}{10})$$

Our team also discovered that the old algorithm could be simplified while still calculating the speeds with similar accuracy.

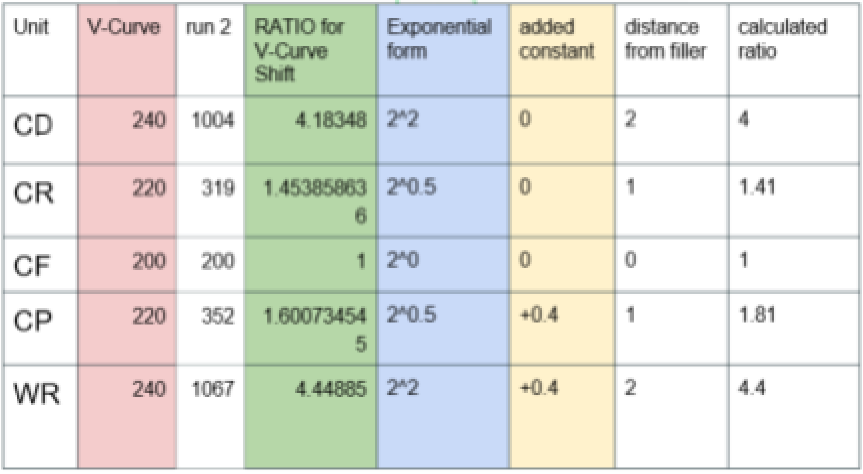
The old method:

$$SpeedOfDesiredUnit = \left(a_1(r)^{n-1}\right)(x)^{n_1\times\frac{200}{a_1}} = \begin{cases} \left(a_1(r)^{n-1}\right)(x)^{\frac{n\hat{}2}{2}\times\frac{200}{a_1}} & :beforeFiller \\ \left(a_1(r)^{n-1}\right)(x)^{\frac{n\hat{}2}{2}\times\frac{200}{a_1}+0.2} & :afterFiller \end{cases}$$

The new method:

$$Speed\ Of\ Desired\ Unit = \begin{cases} fillerSpeed\left(1+\frac{n}{10}\right)\times 2^{\frac{n^2}{2}} & :Before\ Filler \\ fillerSpeed\left(1+\frac{n}{10}\right)\times 2^{\frac{n^2}{2}} + 0.4 & :After\ Filler \end{cases}$$

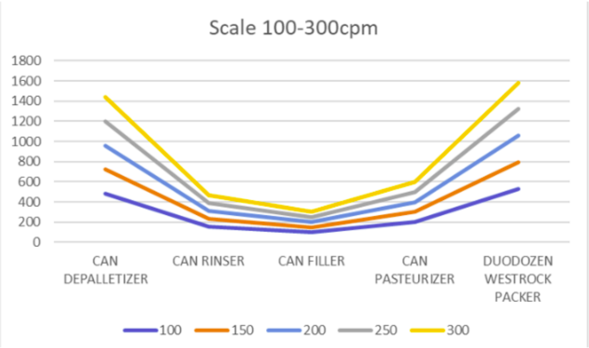
$$\text{Unitspeed} = \begin{cases} \Big[\overbrace{\text{filler}\left(1+\frac{\text{distance}}{10}\right)}^{\text{V-curve Theory}}\Big]\left(2^{\frac{\text{distance}^2}{2}}\right) & :\text{Before filler} \\ \Big[\text{filler}\left(1+\frac{\text{distance}}{10}\right)\Big]\left(2^{\frac{\text{distance}^2}{2}+0.4}\right) & :\text{After filler} \end{cases}$$

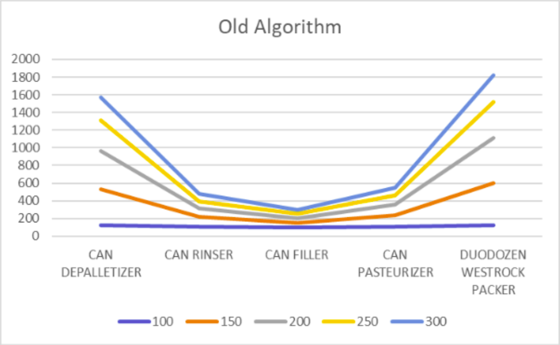


Unit	V-Curve	run 2	RATIO for V-Curve Shift	Exponential form	added constant	distance from filler	calculated ratio
CD	240	1004	4.18348	2^2	0	2	4
CR	220	319	1.453858636	2^0.5	0	1	1.41
CF	200	200	1	2^0	0	0	1
CP	220	352	1.600734545	2^0.5	+0.4	1	1.81
WR	240	1067	4.44885	2^2	+0.4	2	4.4

*CP is can pasteurizer, CR is can rinser, WR is Westrock Packer

The table above shows the methodology behind the algorithm as it is split into two components. First, calculate the V-Curve by the advised 10% increments, and then shift the value by a conversion ratio derived from the historical data of run 2.





These charts show what the algorithm predicts if the filler was set from 100cpm to 300cpm. The new algorithm is much better at creating reasonable predictions at lower speeds.

4. 0 Prototyping Test Plan

Test ID	***Test Objective*** ***(Why)***	***Description of Prototype used and of Basic Test Method*** ***(What)***	***Description of Results to be Recorded and how these results will be used (How)***	***Estimated Test duration and planned start date*** ***(When)***
1	[Deploy Test] The objective is to test if we need to buy a developer account to deploy an application for our client. And test the feasibility of using Xcode.	Prototype Xcode tut will be used. An application that will most likely be a tutorial of how to use/get started with Xcode will be attempted to be downloaded on an Ipad without a developer account.	**Recorded:** Fail **Response:** If the test fails then a developer account will be used to deploy the real thing. **developer account will be bought when the application is fully ready**	26-27th Feb
2	[Pure UI Test] The objective of the test is to make sure the program can be interacted with without issue before focusing on its ability to perform its task.	Test the functionality and ease of use of the UI.(refer to 5.1) Prototype UI Shell will be used. A prototype of the UI that does not perform any real calculations or simulations.	**Recorded:** Score 5/5 students: pass **Response:** Some suggestions have been given and will be taken into account for final UI design (see 5.1) UIkit will continue to be used	4 - 7th March
3	[UI aesthetics Test] The objective is to test the aesthetics of the UI with the user.	Test the aesthetics of the UI with client feedback. Prototype UI Shell will be used. A picture of the UI will be sent to the client for feedback.	**Record:** A number from one to ten will be recorded on the client's likability of the UI colour scheme, Formatting and Aesthetics. Score > 8/10: pass Score <8/10: fail **Response:** Ask for feedback and fix the UI. Iterate until it passes.	6-10th March Need time for the Client to respond.

Test ID	***Test Objective*** ***(Why)***	***Description of Prototype used and of Basic Test Method*** ***(What)***	***Description of Results to be Recorded and how these results will be used (How)***	***Estimated Test duration and planned start date*** ***(When)***
4	[Algorithm Test] The objective of this Test is to prove our ability to calculate the optimal speeds before implementing it into the application.	Test of the algorithm that predicts the optimal speeds. Prototype ExcelA will be used. An excel spreadsheet, that displays all the elements and functions of the algorithm and when given the historical data it is able to replicate the optimal speeds within a small percentage of error.	**Record:** Margin of Error: 10% Pass. **Response:** Algorithms will be expanded to include the conveyors in the future.	4-7th March
5	[Integration Test] The objective of this test is to make sure that the display of the input and output is well formatted on the UI.	Tests of the algorithm's integration into the UI. Prototype R1 will be used. A first rendition of the application will be created that includes the UI and the algorithm working together, the focus will be on the formatting of the input and output numbers and other variables on the UI and the ability of the user to change aspects of the production line and the algorithm still calculates things accordingly to test cases made from Excel prototype.	**Record:** The Pass or Fail of the application's ability to correctly format the UI and output correct speeds of three test cases derived from the Excel prototype. 3 Passes = Pass **Response:** Reference working Excel algorithms and find out what is different or causing problems. Iterate until 3 passes.	9-10th March

Test ID	***Test Objective*** ***(Why)***	***Description of Prototype used and of Basic Test Method*** ***(What)***	***Description of Results to be Recorded and how these results will be used (How)***	***Estimated Test duration and planned start date*** ***(When)***
6	[Input Error Test] The objective of testing wrong inputs is to make sure that the inputs cannot exceed the max speeds of the conveyors.	Test of the UI's ability to catch error inputs. Prototype R2 will be used. On the second rendition of the application where invalid errors are caught, a comprehensive collection of bad inputs will be tested on each possible place where the user can do something wrong.	**Record:** Number of invalid inputs that were stopped Number of invalid inputs that were accepted The ratio between invalid inputs caught and accepted must be 90% caught to pass. **Response**: catch invalid inputs that were accepted and iterate until pass	13-14th March
7	[Multiple Tab Test] The objective of this test would be to make sure that the program is able to handle multiple sets of inputs and display each accordingly.	Test the application's ability to simulate multiple production lines and save and load the user's edits to each production line. Prototype R3 will be used. On the third rendition of the application where multiple lines can be added. Multiple production lines will be added to the application. They will be edited, saved and loaded at selected points throughout a simulated use of the application.	**Record:** Based on the information of different production lines. Three production lines will be tested to run multiple simulations at once. If they can be created, saved and loaded in the right places during regular usage of the application then the test is passed. **Response:** Figure out what went wrong and fix the code and iterate the test until it passes.	13-14th March

Test ID	***Test Objective*** ***(Why)***	***Description of Prototype used and of Basic Test Method*** ***(What)***	***Description of Results to be Recorded and how these results will be used (How)***	***Estimated Test duration and planned start date*** ***(When)***
8	[Stress Test] The objective of the test would be to figure out and test the limits of the application so measures can be put in place to make sure that the user knows the limit of the application	A stress test on the amount of data can be added to the program before it loses functionality or UI formatting. Prototype R3 will be used. On the Third rendition of the application. Production lines will continuously be added to the application until it crashes. Conveyors and Units will continuously be added to a single line until the UI formatting becomes illegible or the program crashes. Extremely large numbers will be added to the filler speed or other inputs until the application crashes or the UI formatting becomes illegible.	**Record:** The limit to the number of production lines, conveyors and units and numbers that can be inputted into the system before the UI gets messed up or the application crashes. If the limits < the theoretical values that would be needed to run the application under normal conditions (determined from the information we gathered for multiple tab tests) then it fails. **Response:** Find the source of the data storage limit and work around or buy more storage. Fix how data is saved and loaded if there is a formatting issue. Iterate until pass.	15-16th March
9	[Random Person Ease of Use Test] The objective would be to make sure that the program and final user manual is comprehensive for the client.	A test on the ease of use of random person's ability to follow an instruction manual and the application. Prototype R3 will be used. The application and a user manual will be sent to a person with no knowledge of beer production. They will be required to complete a set of instructions from the user manual. They will be asked to give feedback on the ease of use.	**Record:** The feedback and the percentage of tasks that the person was able to complete will be recorded. **Response:** Use feedback to improve the user manual and UI if failed. Iterate until pass.	March 18-20th.

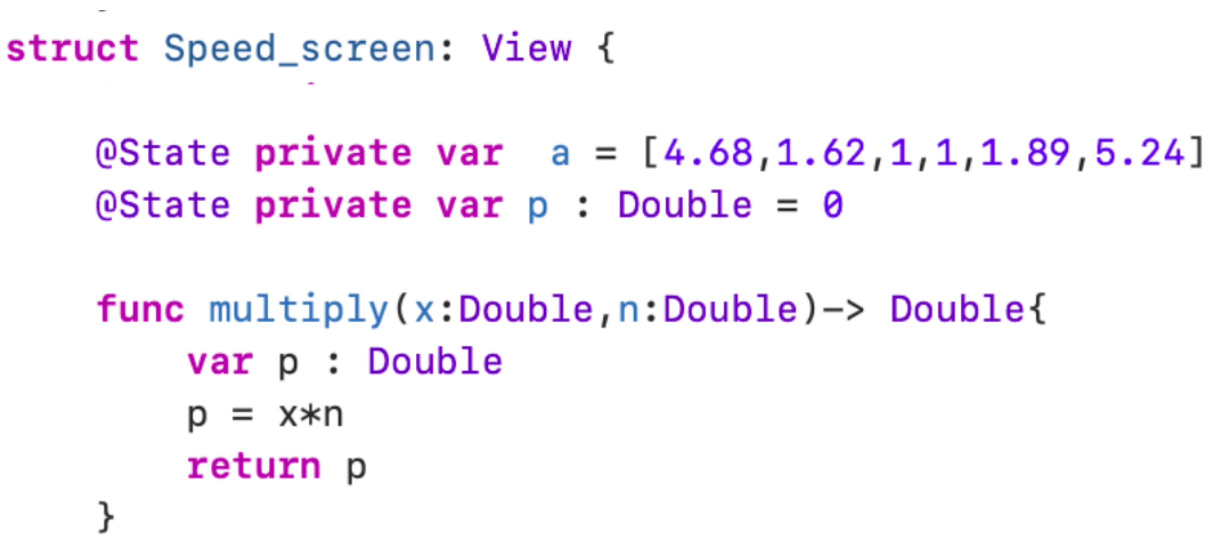
4.1 UI aesthetics Test

Client displayed interest in our prototype and gave positive feedback during the presentation.

4.2 Integration Test

A simple algorithm was integrated into the UI. The UI was created after the old algorithm was found to be faulty. This was done before a new algorithm was derived, so the ratio between the V-curve and the real data was manually inputted into the system as an array “a”. Regardless, the test shows that it is possible to successfully implement an algorithm into the UI.

1. Speed:

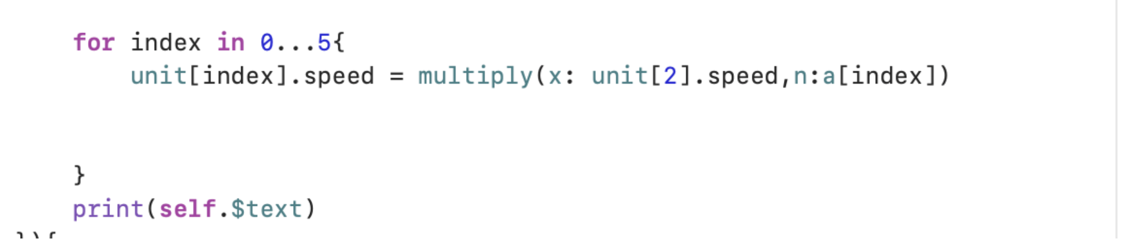


```
struct Speed_screen: View {

    @State private var  a = [4.68,1.62,1,1,1.89,5.24]
    @State private var p : Double = 0

    func multiply(x:Double,n:Double)-> Double{
        var p : Double
        p = x*n
        return p
    }
```

“x” represents the input of filler speed, “p” represents desired speed, and “n” is a variable representing array “a”.



```
for index in 0...5{
    unit[index].speed = multiply(x: unit[2].speed,n:a[index])


}
print(self.$text)
```

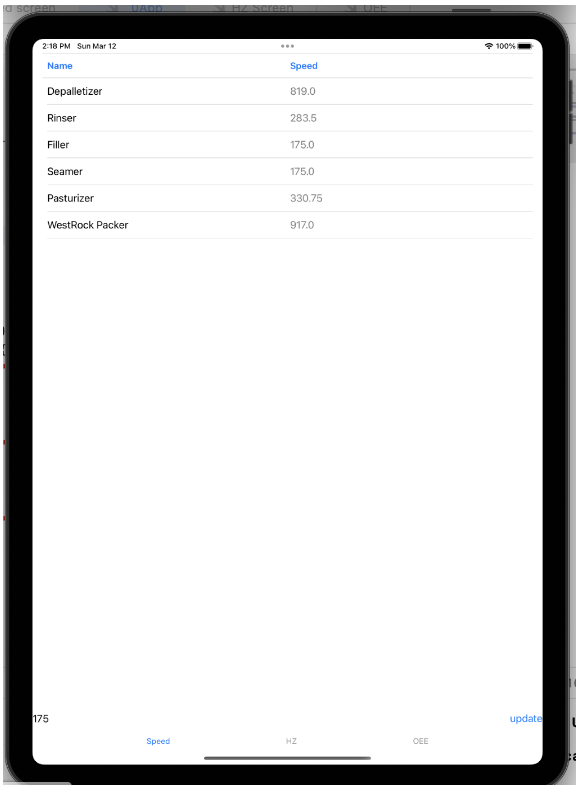
A “for in loop” was used in order to update each speed to its corresponding units from filler speed.

Test:
175 CPM as input filler speed to test the algorithm, and compare the results with the historical data (Table 3).

Unit	Speed (cpm)
CAN DEPALLETIZER	760.9
CAN RINSER	265.0
CAN FILLER	175
CAN SEAMER	175
CAN PASTEURIZER	265.0
DUODOZEN PACKER	760.9

Table 3 historical data

Figure 4 shows that the calculated speeds correspond to the historical data shown in graph 3.



Name	Speed
Depalletizer	819.0
Rinser	283.5
Filler	175.0
Seamer	175.0
Pasturizer	330.75
WestRock Packer	917.0

Figure 4. The view of the Speed Tab on our application with the Filler set to 175 CPM

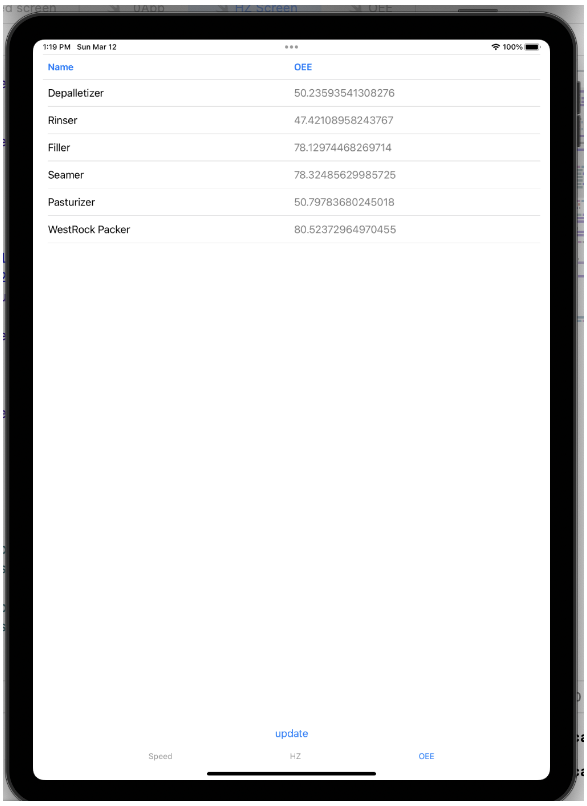
2. OEE (overall equipment efficiency):

```
func power(x:Double,r:Double) -> Double{
        var o : Double
    o = x/(0.1 * Double(r))
        o = x/o
        o = o * q
      o = o * 660/720 * 100
```

“O” represents OEE, “q” represents the quality (we assume quality is 99% for all equipment; clients are able to change it on their own), “x” represents the filler speed (input), “r” represents a random number from 0 to 9.

OEE = (speed/ max speed) * 99% * 666/720 * 100
Max speed = speed / (0.5~0.8)

Since no max speed is provided to us, it is impossible to calculate real OEE.
Therefore, a random number between 0.5 to 0.8 was used to be divided by speed to get a fictional (temporary) max speed.

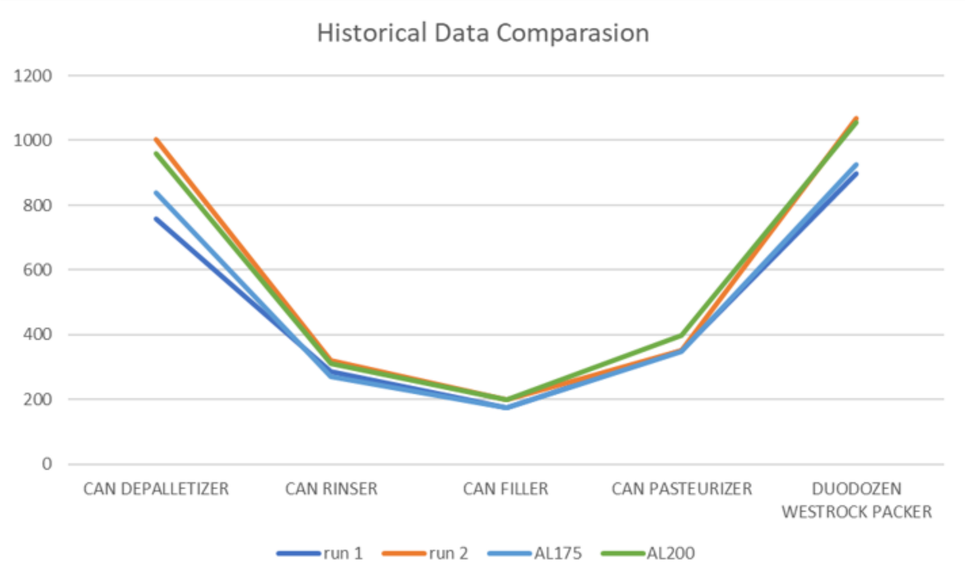


Name	OEE
Depalletizer	50.23593541308276
Rinser	47.42108958243767
Filler	78.12974468269714
Seamer	78.32485629985725
Pasturizer	50.79783680245018
WestRock Packer	80.52372964970455

Figure 5 screenshot of OEE

4.3 Algorithm Test

$$Speed\ Of\ Desired\ Unit = \begin{cases} fillerSpeed\left(1+\frac{n}{10}\right)\times 2^{\frac{n^2}{2}} & :Before\ Filler \\ fillerSpeed\left(1+\frac{n}{10}\right)\times 2^{\frac{n^2}{2}} + 0.4 & :After\ Filler \end{cases}$$



Graph 4. A comparison between the algorithm’s predicted speeds and the historical data

This graph shows the test between the historical data and the new algorithm. Visually, it is clear that the algorithm closely resembles the historical data. There are two areas where the algorithm is slightly off. At the depalletizer, the algorithm overestimates its speed at 175 cpm by 81 cpm. The most that the algorithm deviated by was 10%. Which is within our margin of error. Therefore it passes the algorithm test for the second time.

5.0 Feedback and Comments

The following table outlines the collected feedback and comments on the prototypes.

	Client	Student 1	Student 2	Student 3	Student 4
Are you able to access and use the app smoothly?	Yes	Student 1 wants to have a unit shown on the windows (speed -> cpm).	Student 2 was wondering what OEE is. Note: maybe no abstraction?	Student 3 was able to access it. However, he wants a switch button that can directly switch from speed windows to Hz windows, so no need to retype in filler speed.	Yes
Is the interface logic appropriate and intuitive accessible?	Yes	Yes, really intuitive if regardless of the unknown unit for speed.	Yes. Really intuitive regardless of what OEE was.	Yes	Student 4 loves it.
Does the Algorithm make sense?	Yes	Yes, speed was corresponding to the historical data. But the OEE part was kind of unique. Note: Because for the Hz part, there was no max speed provided to us. Clients are able to fill in the max speed on their own.	Yes, everything makes sense except the OEE part. Note: same as ←	Yes	Yes, student 4 was impressed by the method of temporarily using a random number as a max speed.
Do you like the integration of the UI and the algorithm?	Yes	Yes. It was perfect.	Student 2 hates coding. She believed our team was amazing.	Student 3 thought it was better to add a button to directly switch from Speed to Hz without any extra retyping in data.	Yes.

6.0 Updated Bill of Materials

The following table outlines the updated bill of materials. The changes made include

1. We no longer need a paid developer account.
2. The swift UI we needed for XCode is free.
3. We are no longer using UI kit.
4. We will not be using test flight because XCode can simulate it instead.

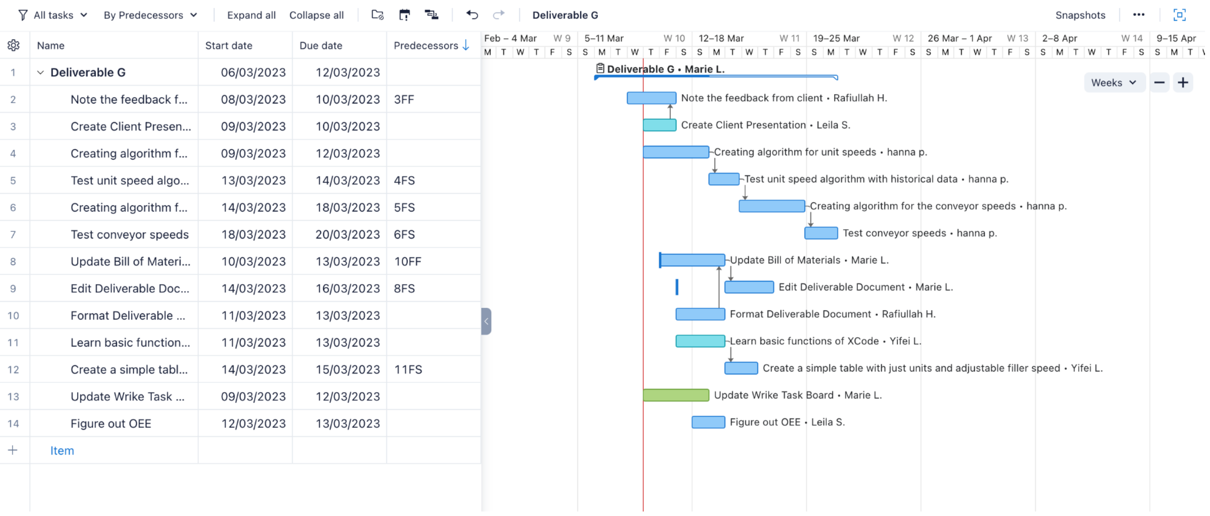
No	Item name		Description	Units of measure	Quantity	Unit Cost ($)	Extended Cost ($)	Link
1	Apple ID account		To use Xcode	Account	1	0	None	N/A
2	Xcode		IDE for native ios application	Software application	1	0	0	https://developer.apple.com/xcode/
3	Xcode Object Libraries	Swift UI	UI libraries for personalizing and customizing UI	Additional software download	1	0 ($99 to deploy)	0 (annually if deployed)	https://developer.apple.com/xcode/swiftui/
4	Excel		Application used to create an algorithm and compute test cases	Software application	1	0	0	https://www.microsoft.com/en-us/microsoft-365/excel
5	Ipad		Device used to run application software	Equipment	1	0	0	https://www.apple.com/ca/ipad-air/ - personal devices for testing - Client also has devices

							available for deploying
7	Mac	Device used to use Xcode	Equipment	5	0	0	https://www.uottawa.ca/faculty-arts/facilities-resources
Total product cost (without deploying it)						$0	
Total product cost (if we deploy it)						$99	

Wrike snapshot:

https://www.wrike.com/frontend/ganttchart/index.html?snapshotId=UVV1G2DKxqANFKvSosWfTg3yI5KGVdLU%7CIE2DSNZVHA2DELSTGIYA

Screenshot (in case the link does not work):


All tasks
By Predecessors
Expand all
Collapse all
Deliverable G
Snapshots
Weeks
Feb – 4 Mar
W 9
5–11 Mar
W 10
12–18 Mar
W 11
19–25 Mar
W 12
26 Mar – 1 Apr
W 13
2–8 Apr
W 14
9–15 Apr
Deliverable G • Marie L.
Note the feedback from client • Rafiullah H.
Create Client Presentation • Leila S.
Creating algorithm for unit speeds • hanna p.
Test unit speed algorithm with historical data • hanna p.
Creating algorithm for the conveyor speeds • hanna p.
Test conveyor speeds • hanna p.
Update Bill of Materials • Marie L.
Edit Deliverable Document • Marie L.
Format Deliverable Document • Rafiullah H.
Learn basic functions of XCode • Yifei L.
Create a simple table with just units and adjustable filler speed • Yifei L.
Update Wrike Task Board • Marie L.
Figure out OEE • Leila S.

	Name	Start date	Due date	Predecessors
1	Deliverable G	06/03/2023	12/03/2023	
2	Note the feedback f...	08/03/2023	10/03/2023	3FF
3	Create Client Presen...	09/03/2023	10/03/2023	
4	Creating algorithm f...	09/03/2023	12/03/2023	
5	Test unit speed algo...	13/03/2023	14/03/2023	4FS
6	Creating algorithm f...	14/03/2023	18/03/2023	5FS
7	Test conveyor speeds	18/03/2023	20/03/2023	6FS
8	Update Bill of Materi...	10/03/2023	13/03/2023	10FF
9	Edit Deliverable Doc...	14/03/2023	16/03/2023	8FS
10	Format Deliverable ...	11/03/2023	13/03/2023	
11	Learn basic function...	11/03/2023	13/03/2023	
12	Create a simple tabl...	14/03/2023	15/03/2023	11FS
13	Update Wrike Task ...	09/03/2023	12/03/2023	
14	Figure out OEE	12/03/2023	13/03/2023	
+	Item			

7.0 References:

(n.d.). Checking link. Retrieved March 5, 2023, https://statics.teams.cdn.office.net/evergreen-assets/safelinks/1/atp-safelinks.html

Ekren, Kemal. "What Is Xcode and How to Use It?" Netguru, 25 November 2022, https://www.netguru.com/blog/what-is-xcode-and-how-to-use-it

How to Use TestFlight For Beta Testing: Benefits & Pre-Requisites- Testrig. (n.d.). Testrig Technologies. Retrieved March 4, 2023, from https://www.testrigtechnologies.com/ios-application-beta-testing-using-testflight

Is the V-Curve the best option for setting accumulation line speed? (n.d.). LineView Solutions. Retrieved March 5, 2023, from https://news.lineview.com/is-the-v-curve-theory-the-best-option-for-setting-accumulation-line-speed

Microsoft Excel Spreadsheet Software. (n.d.). Microsoft. Retrieved February 19, 2023, from https://www.microsoft.com/en-us/microsoft-365/excel

Xcode 14 Overview. (n.d.). Apple Developer. Retrieved February 19, 2023, from https://developer.apple.com/xcode/