Project Deliverable F:
Prototype I
GNG 1103
Prof Knox

Alex Milgram
Kalan Kok
Eric Wang
Charlie Huang
Peter Zhou

Table of Contents

Test Plan

Transition of our ideas for project:

Our previous idea was to have two LCD screens, wires connected to the filler station in order to get current speed. One LCD screen would be a console next to the product line and another one can be carried with the client. After exchanging ideas with TAs in the lab session. We changed our plan to design a mobile app via thunkable. Users insert the present value of speed on the app. After that, the app would analyze what speed has to be in each other stations after the filling station.

Testing For Prototype 1:

Because we formatted our test plan for a different design we followed a different set of tests.

1. Ensure that the Code can Read the Data

We tested this by imputing some value and seeing if it could give the same one back.

2. Set some Threshold to determine if the line is working efficiently

We ran into some problems here when we realized that we needed to incorporate some kind of function into our app. Because we can't incorporate percentages easily it was important for us to figure out various ways to write functions that would work in the same way.

3. Make the values get displayed back to the user

The easiest part was making the values done in the previous two tests get sent back to the user allowing for them to see the suggested speeds the conveyors should be set too.

Steps:

1. The purpose of the test is to make sure the output tells the user what speed each station on the conveyor should be to make the system efficient.

2. As we conclude the correct output is the most important part for this project, we check if our code is valid and try different input speed of the filler station and see if the output shows accurate value for speed. We may also add a description; for example, giving a recommendation of what may be more efficient for the conveyor if the speed before the filler is set to a certain value.

3. Since the only design we need to do is an app, we will test and verify by running the code over and over. We may try different formats in thunkable so that the app is simplified-coded and does not store any extra space.

4. Perform the app to the clients, demonstrate our ideas and receive feedback from users. We may add features if clients have extra demands.

5. Measure the speed output is showing for each station, time the software the processing, edit more detail for the app. For example, the size for wording, the color and the text style on the app. Overall, giving a well structured app for users.

6. Analyze the data team members recorded. Conclude the project and document with design concept, design criteria, project specification and so forth to let the user understand why and how each step is developed to our final product.

Updated Target Specifications:

Can read and interpret data:

Read from the keyboard. Provide suggestions/ reflection. Accept different kinds of data.

Can do calculations:

Accept input and provide the result. Calculate the current speed and the ideal speed. Compare the two speeds.

Has a simple and easy to use UI

Simple windows and instructions for the windows. Use different colors to show positive and negative influences. Show with bold text and clear worlds.

Critical Systems

Calculations:

The Calculation system refers to all the calculation based aspects of the code needed for the app; this includes the percentage calculations, and the other value calculations that will be added to the next prototypes.

User Interface:

The user interface system refers to all the interactable or readable things our app has including buttons, text entering or visible data. The user interface is the most important part of our project as it should be simple and easy to use.

Input and Output:

The input and output system refers to all the data that will be stored or entered and displayed. The inputs will be the values that the user gives and the outputs will be the values after the value is run through various calculations. We will have multiple output values for only a few input values. It is also important that our input and output data can be stored as well as it makes it easier when the user is keeping track of the data.

Prototype I:

https://x.thunkable.com/copy/e36d99503d60319eca0ba84219daf983

Description of prototype: Takes in the filler conveyor speed value and uses a percentile calculation to determine the optimal speed that the other conveyors need to be traveling at. The calculations are determined using V-curve theory which is the most optimal way for the line to run. The 10% rule will be implemented at each appropriate node depending on the location of the filler, or conveyor element.

Prototype II 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	Add more important information that will be visible to the user (based on the data the user inputs). This will help our users to have a better grasp on some of the other specs as well. And will allow them to use the app for more than one purpose. The criteria for success with this test is to find a balance of information to include in the app but not to overwhelm the user.	Type: Comprehensive and Analytical This combination of prototype types is not feasible generally but, in our case, we must try to consider all possibilities of data the user could want and incorporate it directly into our app. Firstly, we will determine what information should be added to the app. Then we will incorporate the data so that the user can see all the information they may need to adjust their line. We will test this prototype by asking Mill St if there is any other data they may want us to include/takeaway from our prototype.	We will write down notes during our conversation with Mill St. Based on their thoughts we will determine the changes that we will need to make to our prototype. The extra data we include is very important and the responses from our client are the most important sources of information we have to go off of.	Time: 1 hour Date: Mar 8th 2023 Dependencies: None Stopping Criteria: After we have found the correct balance of information to incorporate and have incorporated it into our app.

		Skills: Might take some time to figure out. materials:Thunkable		
2	Make the UI easy to use and visually pleasing. This will help to make our app easier to navigate and use practically. In order to achieve success in this portion we must ensure that no confusion is generated when using our app.	Type: Focused and Physical To make the best UI possible it is important that we have some Input from people using the software for the first time. To test our prototype we are going to have someone navigate through the software while they are continuously speaking meanwhile, we are recording the struggles/confusions they may be having. Skills: Might take some time to figure out depending on what their preferences are.	We will take notes of the tester's responses to the UI. We will then use this data change various things based on their struggles/suggestions.	Time: 30 min Date: Mar 8th 2023 Dependencies: After all other tasks have been completed this can be done this may need to be redone in the future as more interactions are added. Stopping Criteria: When we Think the UI is at an exceptional level.

		materials:Thunkable ,and a person		
3	Add some visuals to make entering the data easier. This will help the user when they are navigating the app and will allow for more flexibility. The most important aspect of this is that the symbols and visuals we are going to use don't further confuse the user and provide a benefit for the user.	Type: Focused and Physical Often, we find that simple visuals are very important so to test their effectiveness in our prototype we will use a set of flashcards on someone who has never seen the emoticons or visuals before. We will ask them if they understand what the visual is saying and if they understand very clearly then we will incorporate it into our prototype. If they don't understand it may be more confusing to include it. Skills: Should be easy to incorporate. materials:Thunkable , a person, and some flash cards	We will be taking notes based on the responses that our tester gives us. We can determine whether the images or symbols are too complex for them to understand and therefore not incorporate into our prototype.	Time: 20 min Date: Mar 8th 2023 Dependencies : None Stopping Criteria: When we finish implementing and testing the Criteria necessary for visuals.

4	Add a History section to see the previous values that were entered. Having a history section will help the managers to either repeat speeds calculations with ease or even help document line efficiencies with daily graphs. The most important part of this is to ensure that the previous data that was used is being saved to the history page.	Type: Focused and Analytical To incorporate some kind of history, function our app must be able to save previous data to some location on our screen. We will test this by setting a location for the history window and setting up a function that will add values to this section. This test will mostly be trial and error with different codes until we have multiple history values saving. Skills: Might take some time to figure out and might not be possible. materials:Thunkable	We will take notes of the trials we do. This information is not very important for our project but could be helpful for future tasks with similar functionalities.	Time: 1 hour Date: Mar 8th 2023 Dependencies: None Stopping Criteria:We will stop if we run out of time, cannot implement a function like this or we complete the history function.

5	Show a chart of the increases that changing the conveyer speed will have on the rest of the system. This will help the user to see how much more efficient the line will become through each statistic.	Type: Focused and Analytical Having a comparison of the two speeds can be a great tool for our users. To test this, we will need to use a function again in the thinkable code that will take the value entered and calculate the data associated with it and the suggested speed and the data associated with that. This is another trial-and-error sequence until the code shows a chart comparing the two different values the speeds have. Skills: Might take some time to figure out. materials:Thunkable	We will take notes of the trials we do. This information is not very important for our project but could be helpful for future tasks with similar functionalities.	Time: 45 min Date: Mar 9th, 2023 Dependencies: The first test must be done before this task because it will compare the additional data as well Stopping Criteria; We will stop if we run out of time or complete the chart.

References and Wrike:

https://www.wrike.com/frontend/ganttchart/index.html?snapshotId=5laVsrzmaFQ7zQXEKkahLB1fjuGBxs9d%7CIE2DSNZVHA2DELSTGIYA