



Cleanroom Dashboard System for Time-Reduction Checking of Disk Tester Availability

Camille Merlin S. Tan^{1,*}, Lawrence Y. Materum^{1,2}

¹ Department of Electronics and Communications Engineering, De La Salle University, Philippines

² International Center, Tokyo City University, Japan

ARTICLE INFO

Article history:

Received 29 March 2023

Received in revised form 8 November 2023

Accepted 1 March 2024

Available online 17 April 2024

Keywords:

Storage technologies; Fixed disk; Tester availability; Dashboard; Optimization; History log; Data analytics

ABSTRACT

With the vast amount of data being processed and reviewed in production and manufacturing industries nowadays, it takes a lot of time and effort to keep track of their performance metrics. Developing an automation system to improve the overall system performance and reduce man-machine interactions is necessary. Prior to the work, a computer disk drive manufacturing facility checked its fixed-tester availability manually, which made the company unable to optimize its man and machine hours as much as possible. This work proposes a system that displays all the critical information in a dashboard system to address the problem while enabling access to data history logs and further analytic insights. Moreover, the system enables readily accessible fixed-disk tester availability. The outcomes indicate significant improvements, especially in delay reductions, necessary for optimizing man and machine interactions in a fixed-disk testing facility at a leading computer disk drive manufacturer and data storage company at Laguna Technopark.

1. Introduction

1.1 Background of Study

Western Digital Storage Technologies Corporation (WDC) is an American computer hard disk drive manufacturer and data storage company. Its headquarters is in the Santa Teresa District of San Jose, California, USA. The company designs, manufactures, markets, and sells data storage devices such as hard disk drives (HDDs), NAND flash-based storage devices, solid-state drives (SSDs), enterprise storage solutions, data centre systems, and cloud storage services ("Western Digital | WDC Stock Price, Company Overview & News," n.d.; "Western Digital Corp. - Company Profile, Information, Business Description, History, Background Information on Western Digital Corp.," n.d.; "What Is Western Digital Corporation (WDC)?," [4,6].

The company has several branches and manufacturing sites around the world. In the Philippines, the company only manufactures head gimbal assembly (HGA) and sliders, and then the HDD is assembled in different countries.

* Corresponding author.

E-mail address: camille_merlin_s_tan@dlsu.edu.ph

<https://doi.org/10.37934/araset.43.2.253273>

WDC sells its products using four brand categories: Western Digital, (WD), G-Technology, and SanDisk. Western Digital's brand focuses on products for enterprise applications such as OPenFlex, Ultrastar, and RapidFlex product categories. Whereas the brands WD, G-Technology, and SanDisk focus on consumer products such as memory cards, personal desktop drives, and flash drives. For WD products, it includes storage devices such as WD Elements, My Passport, My Book, WD Easystore, WD_Black, and My Cloud product categories. For G-Technology products, it targets audio and video editor professionals and offers both HDD and SSD. SanDisk products include consumer data technology products such as wireless chargers, flash drives, and micro-secure digital (microSD) cards. In addition, their hard drives are color-coded based on the following: black for high-performance drives, blue for general-purpose drives, red for network-attached storage (NAS) drives, purple for surveillance drives, green for eco-friendly drives, and gold for data centre drives [8,9].

The history of WDC can be traced back to 1956, when IBM invented the world's first hard drive, the Model 350 Disk Storage Unit. It was also known as the RAMAC. This division of IBM was later sold to Hitachi's storage division, which WDC obtained [4].

On April 23, 1970, the company was founded by Alvin B. Phillips. Initially, the company was known as General Digital and rapidly started manufacturing semiconductors. In 1971, the company changed its name to WDC and released its first product, the WD1402A universal asynchronous receiver/transmitter (UART). In 1975, the company became the largest independent manufacturer of calculator chips. After a few years, they launched various products such as the FD1771 floppy disk controller, MCP-1600 multichip, and micro-coded central processing unit (CPU) ("Western Digital Corp. - Company Profile, Information, Business Description, History, Background Information on Western Digital Corp.," n.d.; "What Is Western Digital Corporation (WDC)?," [4]

In the early 1980s, WDC started to make HDD controllers. In 1983, the company contracted to supply IBM with controllers for the PC/AT. This controller is known as the WD1003, which became the basis of the ATA interface. It was created with Compaq and the Control Data Corporation MPI division. In the mid-1980s, the storage controller became their primary selling product since they provided these controllers to well-known companies such as Compaq, International Business Machines (IBM), Hewlett-Packard (HP), and Tandy [4,5].

Furthermore, WDC initiated collaborations with the Massachusetts Institute of Technology (MIT) and developed the Nu machine artificial intelligence (AI), which was sold to Texas Instruments (TI). In the late 1980s, the company focused on producing original equipment manufacturers (OEMs). The company also bought a lot of small manufacturers, such as Adaptive Data Systems, Verticom, and Paradise [4,5].

In the early 1990s, the significant product sales of the company began to change into other products like Caviar drives. Caviar drives are used in various types of personal computers during that time. Moreover, the company became the first multinational United States-based company that received an ISO 9001 certification from the International Organization for Standardization (ISO). In the late 1990s, the company ranked third in the number of drives shipped worldwide and its market share. Companies that ranked first and second are Seagate and Quantum, respectively. In addition, WDC contracted with IBM to use IBM technologies such as giant magneto resistive (GMR) heads, which made them become the Expect drive product family [4,5].

In the early 2000s, the company became the first manufacturer to offer the most advanced technology attachment (ATA) HDDs with 8 megabytes of disk buffer. The company obtained most of its assets from Read-Rite Corporation a few years later. At the same time, the company offered WD30 Raptor, the world's first 10,000 revolutions per minute (rpm) Serial ATA (SATA) HDD [10-11]. In the mid-late 2000s, the company made its new product, the My Book external hard drive. WDC introduced a perpendicular recording technology to desktop drives. In the late 2000s, they released

the eco-friendly WD GreenPower (GP) drives and VelociRaptor series. Furthermore, they made the first 2 terabytes (TB) internal HDD and the SSD in the market and introduced the first 3TB internal HDD [4,5].

In the early 2010s, the company became the largest manufacturing company of traditional HDDs worldwide. WDC bought SanDisk in 2016 for 19 billion dollars and bought the cloud storage provider, Upthere, and flash memory storage array manufacturer Tegile Systems. In the late 2010s, the headquarter of WDC moved to San Jose, California.

In the early 2020s, the company had over 14,000 patents, and it is currently working towards developing its Zoned Storage Initiative. This initiative motivates adopting an open data infrastructure centred on zoned storage devices (ZSDs). WDC started working with Kioxia, known as Toshiba Memory, a Japanese memory-chip supplier. In 2022, WDC and Kioxia reported contamination issues that affected the flash memory joint production output. Therefore, factories such as Yokkaichi and Kiakami Japan stopped manufacturing these products [4,5].

1.2 Prior Studies

Having an automation system is necessary nowadays, especially with the advancement of the available software and technologies. Some examples of companies and software using the same type of automation system include the following [13,15-17]:

- i. Advantest- The company provides test systems, test handlers, and device interfaces that are essential in performing semiconductor testing.
- ii. Teradyne- A company known for its automation in the testing of electronic equipment.
- iii. Jira- A software tool for system tracing, it allows agile project management and bug tracking.
- iv. BiCXO- A business intelligence software tool that performs automation, allowing the users to handle operational data, data visualization, and expenditures.
- v. Corporater- A dashboard software for data interpretation and collaboration, suitable for medium and large companies.
- vi. Spider Impact- A dashboard software that assists users in collecting and organizing data.
- vii. Google Analytics- The dashboard software that records and reports website traffic.
- viii. Klipfolio- a dashboard software that allows users to share real-time data.
- ix. Qlik Sense- a dashboard software that allows users to make local and offline visualization, interactive dashboards, charts, and analytics applications.
- x. IoT Cleanroom Monitoring System- a smart monitoring system that monitors the temperature, humidity, and pressure inside the cleanroom.
- xi. E-COCOMO- an effort estimation technique for a cleanroom monitoring system.
- xii. Cleanroom Monitoring System- for cleanroom applications, it provides additional security to the server, implemented in local servers and personal computers.

1.3 Problem Statement

Previously, the company had no cleanroom dashboard system for time-reduction checking. They perform all the tasks manually, which takes a lot of time and effort to keep track of and update all the data, leading to an inefficient system. Therefore, the proposed system was made to optimize the tester availability checking, increase its efficiency and reduce the man-machine interactions, bringing the overall process to an improved level.

1.4 System Objectives

- i. To apply technical knowledge and gain professional experience;
- ii. To provide innovative solutions and improve the overall efficiency of the system;
- iii. To communicate with the end users effectively, since several modifications are requested based on the users' needs, it is necessary to improve the overall quality of the software and the user experience;

1.5 Significance of the Study

The work focuses on implementing automation software, which helps the end users to get updated and optimize the whole system as much as possible. This work aims to improve the efficiency, speed, and reliability of the daily tasks previously performed manually by the company's technicians.

1.6 Assumptions, Scope & Delimitations

- i. Assumptions: fixed format file names and a fixed number of columns to be displayed in the dashboard.
- ii. Scope: the proposed system can only be applied in an industry doing the same task.
- iii. Delimitations: the proposed system is limited to providing software-based automation.

1.7 Methodology and Main Contributions

The proposed system followed the Evolutionary Prototyping Model process [19]. Throughout the process, the author gets continuous feedback from the end users, and revisions are done, making the development process much smoother [21]. The main contribution of this work is that it was able to simplify the daily task of the end users, which has never been done before, and at the same time, it increased the overall efficiency of the whole system.

The rest of the paper is organized then as follows, the methods adopted and their features highlighted are discussed in Section II. It is followed by the results, analyses, and discussions in Section III, and lastly, by conclusions in Section IV.

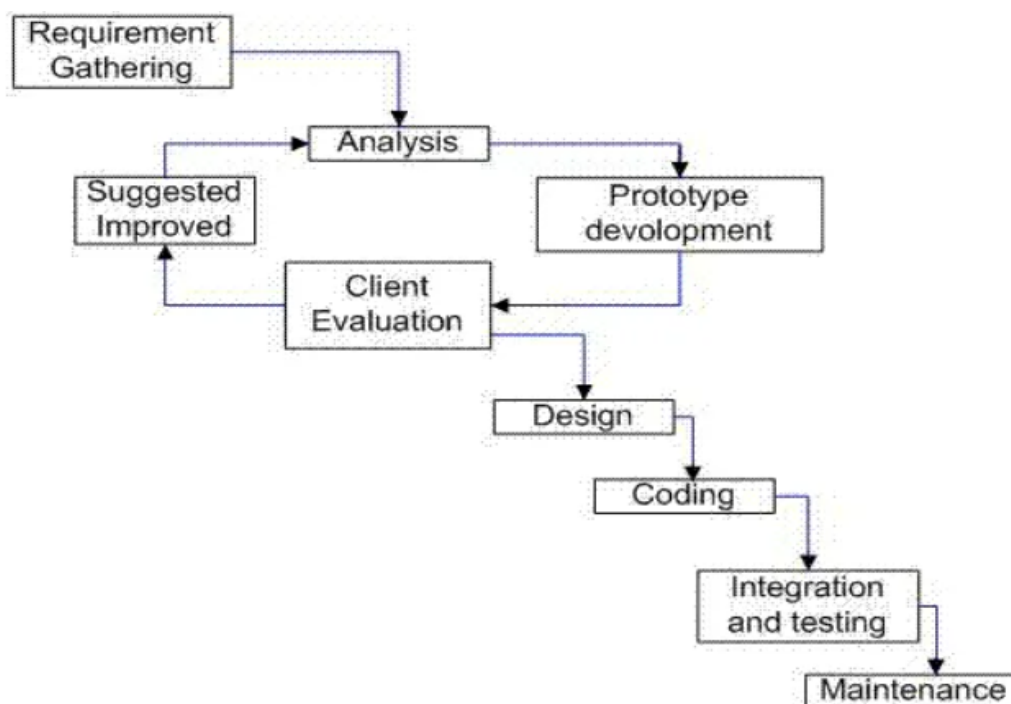


Fig. 1. The Applied Software Development Life Cycle: Evolutionary Prototyping Model [19]

2. Methodology

Testing and evaluating the Tester Availability Dashboard for this work requires several steps. For dashboard automatic data collection, the program pops up a window and asks the author to select the specific directory path. Upon opening the program, the author selects the current date and time to be used and leaves the dashboard open for data collection. After collecting the data, the author saves the history log and computes their average time. While for manual data collection, the author needs to open the .csv file manually, group the same Tool ID data together by copying and pasting each necessary data into another excel file, then compute their average time.

As part of the manufacturing process, tracking the status of the tester slots among all the tester units is challenging. It is unavoidable that technicians would be unable to keep track of all the tester units and maximize all the tester slots effectively due to the many tester units in the facility. At the same time, it is not easy to trace and see all the essential details simultaneously.

The author helped solve this problem by developing software that automatically checks and displays the details of each tester slot from all the tester units to allow the technicians to monitor and maximize the use of each tester slot efficiently. In doing this, the Tester Availability Dashboard, also known as the Blade Availability Dashboard. The author applied the concepts learned from Industrial Automation and Control to simplify and make the system operate automatically without requiring people to monitor the data manually. Industrial automation is essential since it reduces time, cost, effort, and human errors while increasing efficiency, quality, and speed.

Table 1 shows the summary of the solutions to all the issues encountered in doing the project.

Table 1
 Summary of Solutions for the Issues Encountered

Issue	Solution
<p>Challenging tasks</p> <p>The end users requested many revisions, some of which require many changes from the current structure of the code, which takes much time to accomplish. Sometimes, the working software suddenly encounters issues after several trials due to human error in the file names.</p> <p>Making a user manual tutorial was a new experience; everything should be written in detail.</p> <p>The work schedule is hectic, and new revisions are frequently requested; The author cannot gather data during the internship period. Since the assigned projects were initially applied in real-time data, some problems were encountered at the beginning of the data gathering due to the file and folder modification date changes after the folder had been copied into another destination. Since the author cannot collect data in real time, some parts of the original code must be modified since the codes' logic is interconnected.</p>	<p>The author does advance reading and research during her free time.</p> <p>The author tried making the codes as flexible as possible by creating multiple small files, making the file names easy to remember, and applying the same format.</p> <p>The author asked the end users what they did before the error happened and asked for the error message, if there is any, to have an idea of the problem and try to solve it by trial and error if the problem is not yet identified after the end user's feedback. When the problem is detected, the author makes an exception command for the program to allow a specific error when reencountered.</p> <p>The author tried to include screenshots of each function with a detailed and brief explanation, followed by a video walkthrough of the functions.</p> <p>The author focused on the assigned work and saved all the necessary files on her HDD during the internship, and performed data gathering after the internship period.</p> <p>The author tried a lot of testing and solutions. She obtained some findings on the effect of folder copying and the file reading of the CSV files.</p> <p>The author decided to remain the original code because it is challenging to change the whole structure of the code while altering only a part would cause repercussions. Thus, the author manually records and edits the file dates and performs data gathering.</p>

Table 2 briefly describes each task assigned and the solution made for the Tester Availability Dashboard.

Table 2
 Summary of the Task and its Description of the Tester Availability Dashboard

Task	Description
Obtain the latest time stamps of each blade slot.	Grouped all data for each blade slot and searched for the maximum value;
Create a .csv input file for the time difference computation & setting of the colour status.	A 4-column CSV file was made, with the 1st column for setting the current time for testing purposes and the 2nd to 4th columns for parameter settings.
Compute the time difference between the latest and current timestamps, then combine them into the dashboard.	Subtraction was applied to find the time difference; the Zip command combined and read the data from the CSV file and input file simultaneously.
Perform initial testing.	Multiple test inputs are used to test the result (both time and date change cases were tested).
Add a colour indicator of status in the dashboard.	Used the System.Drawing library to apply the "color.Green, color.Yellow, color.Red;" The end user in the CSV file input can set the parameters of the color status.
Add the current time display in the dashboard.	Used the DateTime.Now to return the current date and time on the computer, expressed as the local time;

Add refresh settings.	The user can set his desired time (seconds) in the dashboard. There is also a manual update table button just in case the user does not want to use the auto-refresh.
Add version history.	Added a version information pop-up window.
Add a pop-up window and improve the user interface.	Added a settings option in the dashboard. The user can set the desired time for auto-refresh
Connect PC with Box drive.	Downloaded Box Drive on PC and signed in using the company account.
Reduce the column width in the dashboard while maintaining all the details (latest TimeStamp (TS) and time difference).	Removed the latest timestamp display in the dashboard and completed the hover slot details.
Access the ten latest folders by filtering them among all the files.	The user filters out the ten latest folders with the "MFG" keyword from the "11_Build Data" folder for M3T and "Daily Data Update" for NFC.
Add a drop-down in the dashboard for the users to choose the data to be displayed (All, NFC, M3T).	By default, it displays all of the data, and then the tools list is to be displayed according to the chosen option.
Monitor tools with "_PartialData" in the file name and disregard tools with "_Completed."	The latest files with "_PartialData" are filtered out from the latest folders. It also checks if the folder contains other tool IDs to filter it into the dashboard.
Add "# of latest folders to view" for filtering in the dashboard.	The end user can set his preferred number of folders to view for both NFC and M3T folders.
Add the RecipeFilename column in the dashboard.	It also displays the RecipeFileName of each tool ID
Change the header column from "RecipeFileName" to "Capability."	Changed the header cell name to "Capability."
Display a confirmation message when a directory path is saved.	Added a pop-up message indicating the path is saved and shows the saved directory path.
Fix error in settings: change of update rate.	Fixed its initialization in the code
Enable manual input of path in the dashboard.	Used CommonOpenFileDialog to allow both manual path input and thorough browsing.
1st release for user testing.	Presented a user tutorial on the software;
Change the dashboard name from "Tester Availability Dashboard" to "Blade Availability Dashboard."	Modified the dashboard label;
Reduce the column width by labelling the column header from blade slot #1, blade slot #2, blade slot #3, blade slot #4, and blade slot #5 into 1, 2, 3, 4, and 5.	Modified the header label in the dashboard;
Remove the time difference display in the dashboard and maintain the colour status indicator.	Made the text transparently and applied anchor property for auto resizing of headers.
Make the dashboard resizable to avoid scrolling as much as possible.	Applied anchor property on the window's top, down, left, and right.
Convert the input of the "auto-update refresh" rate from seconds into minutes.	Modified the user input from seconds to minutes;
Add a pop-up window for the colour status parameter settings based on Tool ID's capability.	The user can input a specific model and its parameters, and there are default parameters. If the user does not indicate the model-specific parameter, it automatically follows the default parameter setting.
Add a function: add and remove models.	By default, the M3T and NFC are the available models. When the user adds a model, it automatically becomes available in the change directory settings for the user to set the path. On the other hand, removing a model deletes the specific model in the dashboard.
Add a countdown timer before refreshing the dashboard.	Applied subtraction from the current set auto-refresh time interval;

Modify the resizable window.	Set the AutoSizeColumnsMode into the fill, and the display cell is resizable depending on the width of the dashboard window.
Add colour status settings parameter indicator in hover.	Added another row for current colour status parameter setting details using tooltip text.
Add a new column, "Location," in the dashboard.	The location of each tool ID is displayed in the dashboard.
Add filtering based on tool ID locations.	All tool ID locations are displayed by default, then the list of tool IDs is displayed according to the chosen option.
Add settings for the user to set and add specific tool ID locations.	By default, each tool ID is set to a specific location. The user can change the default location by choosing the specific tool ID from the drop-down menu and altering it to the desired location.
Remove the display of PartialData in the dashboard.	Added another condition that checks whether the specific Tool ID has existing "CompletedData" in the folder. If yes, the PartialData are not to be displayed anymore.
Allow sorting of the dashboard in ascending or descending order when the header is clicked.	Made the DataGridViewSortMode automatic;
Improve the UI for adding and removing models.	Combining the add-and-remove model to simplify the users' process, the users are automatically directed to the directory path setting after a model is added. Also, the small pop-up window closes automatically;
Improve the UI for the download history log.	Created the pop-up window for the history log filtering and download functions;
Add Tool ID display in hover.	Added another row for Tool ID details using tooltip text.
Add a downloadable version history log.	Added a function that allows keeping history logs and downloading these logs based on the settings of the user. It allows the model's capability, location, and date range filtering. The downloaded file uses a .csv file format.
Add two new columns: "Set" and "Head Type."	Added these two columns in the dashboard. It can also be sorted; There is a defined list of sets for the tool ID by default. If none, it displays "For Update."
Add settings for "Set."	By default, there is a specific 'Set' assigned to the tool ID. When the user wants to change to the 'Set' of a tool ID, it can be changed.

Table 3
 Summary of Modifications Done After the Internship for Tester Availability Dashboard

Problem	Solution
Allow the dashboard to modify the current date and time.	The dashboard allows modification of the date by using a pop-up calendar and the time by editing the text in the text box.
Since the folder date has changed after they were copied to another location, manually change the folder & .csv file modified date for data gathering.	Each of the original dates was recorded and changed to apply the logic of the original code.
Save each file as an excel worksheet.	In performing the manual data gathering, the .csv files were saved as excel worksheets before grouping since .csv files cannot have multiple sheets on the file. It only saved the active sheet.

3. Results

Figure 2 shows the Tester Availability Dashboard tested on July 29, 2022. In the dashboard, each tester unit, also known as the tool ID, contains five slots. The colour status indicator in the dashboard represents the current status of each tester slot. There are four possible colours: green, yellow, red, and black. The prominent colours green, yellow and red are displayed depending on the parameters set by the end user based on the capability group of the specific Tool ID. Green indicates the specific slot is not in use fo" onl' a short period, yellow indicates the specific slot is not in use for an average time. Red indicates that the specific slot is not used for an extended period and should be used immediately. Sometimes, black cells appear. This appearance indicates a case wherein the tester slot has never been used. The Model, Set, and Capability show the group of a specific tool. The capability sets the specific time range of its colour status. Location indicates the specific location of the Tool ID in the cleanroom. In addition, there are two possible Head Types, namely UP and DN. These types are used when performing special case testing because different testing criteria are used for each head type.

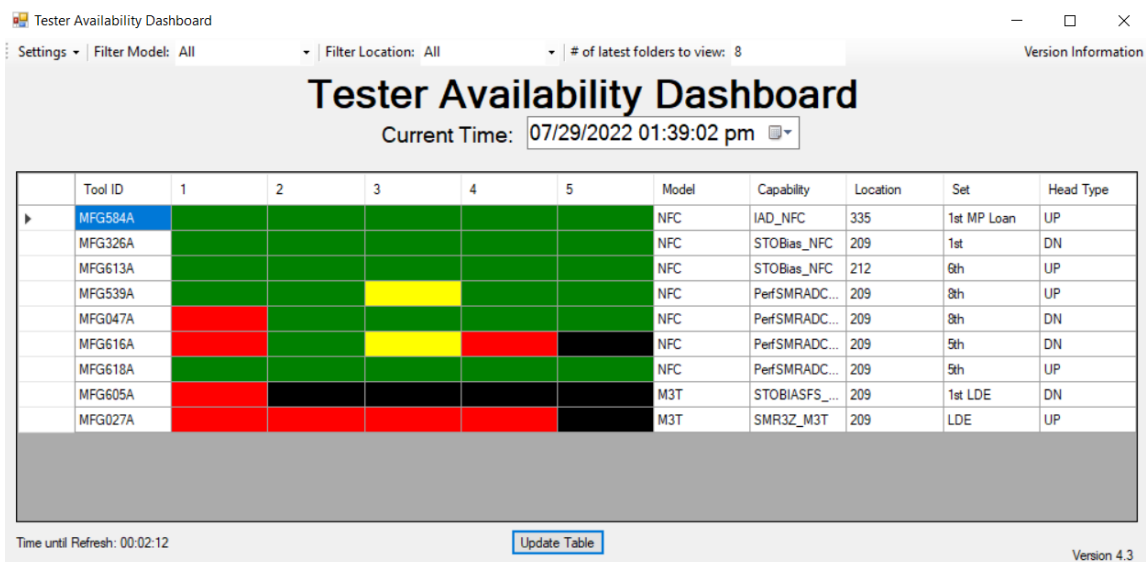


Fig. 2. Final Version of the Tester Availability Dashboard

Figures 3 and 4 show the complete information of a tester slot in hover. The cells in the dashboard display the colour status of each tester slot when the mouse points at a specific cursor, the latest timestamp, the time difference between the latest timestamp of the specific tester slot being used and the current time, as well as the colour status parameter setting, are to be displayed. As shown in Figure 3, when the user makes a specific colour status parameter set, it displays its capability name, such as "Colour Status: On PerfSMRADC2_NFC".

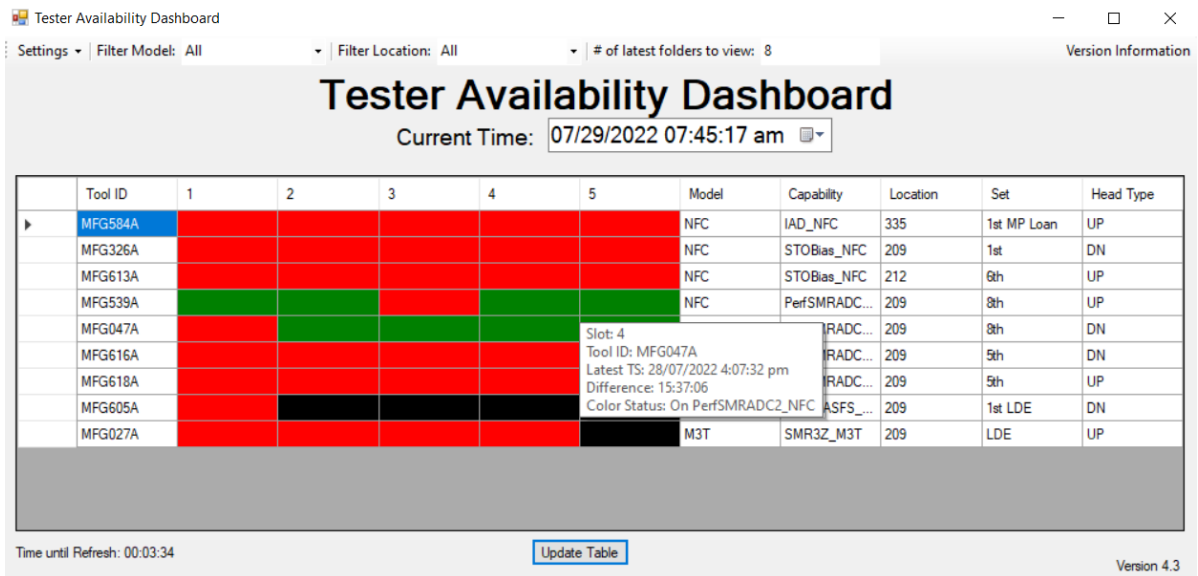


Fig. 3. Displays the Complete Information of a Cell in Hover (when a colour status parameter is set)

On the other hand, as shown in Figure 4, when the colour status parameter is on default, it displays "Colour Status: On default."

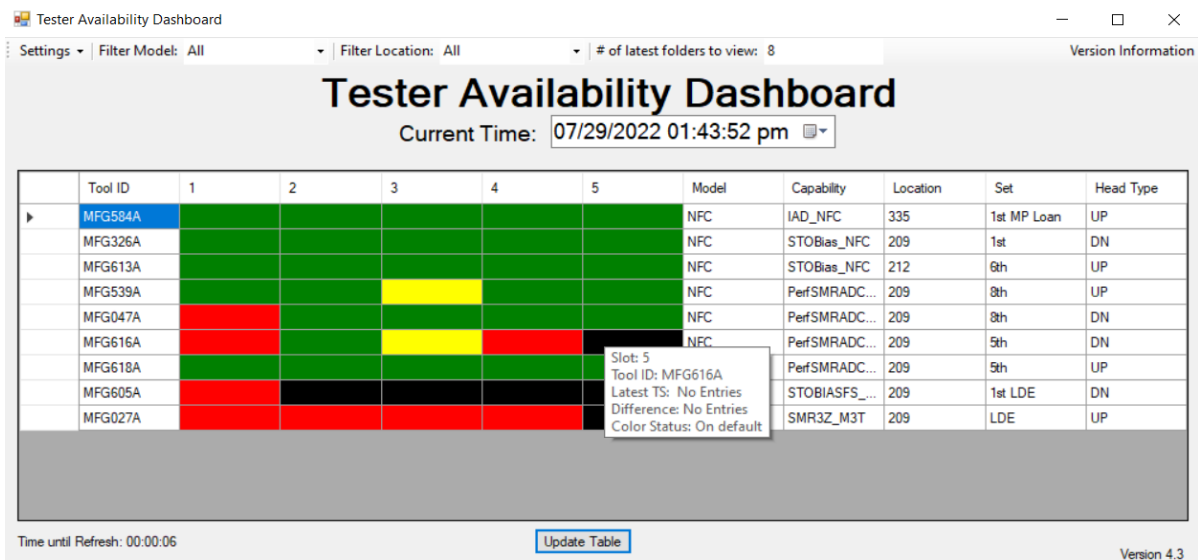


Fig. 4. Displays the Complete Information of a Cell in Hover (When a Colour Status Parameter is On Default Settings)

Figure 5 shows the auto-refresh and update table settings user interface. By default, the auto-refresh is set to 10 minutes, and users can click the settings drop-down menu to change the auto-refresh time as desired. On the other hand, if the user does not want to use the auto-refresh function, the user can manually click on the update table button in the dashboard to refresh the table immediately.

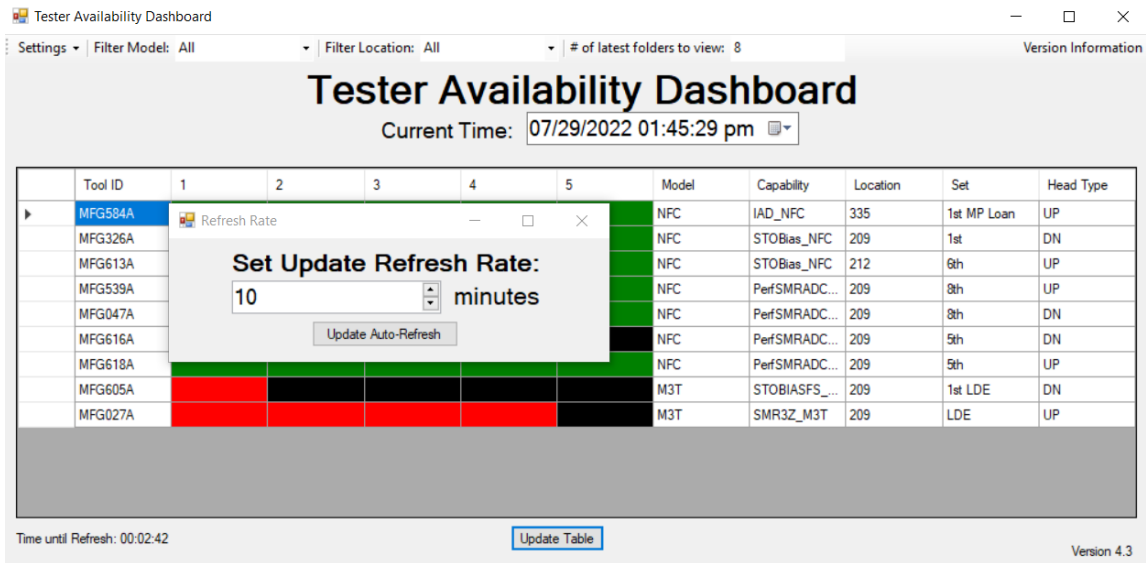


Fig. 5. Auto-Refresh and Update Table Settings

Figure 6 shows the dashboard sample after successfully filtering out all the M3T models from the list. This filtering makes it easier for the user to see a specific model, especially when a large number of data sets.

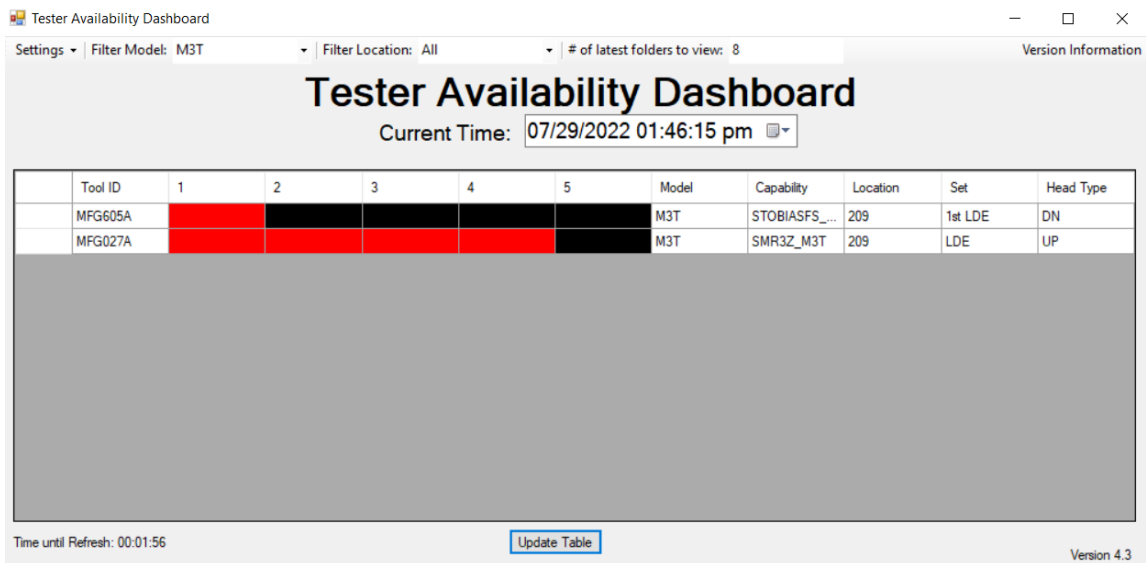


Fig. 6. Filtering All of the M3T Models from the List

Figure 7 shows the sample of the dashboard after successfully filtering out all the NFC models from the list.

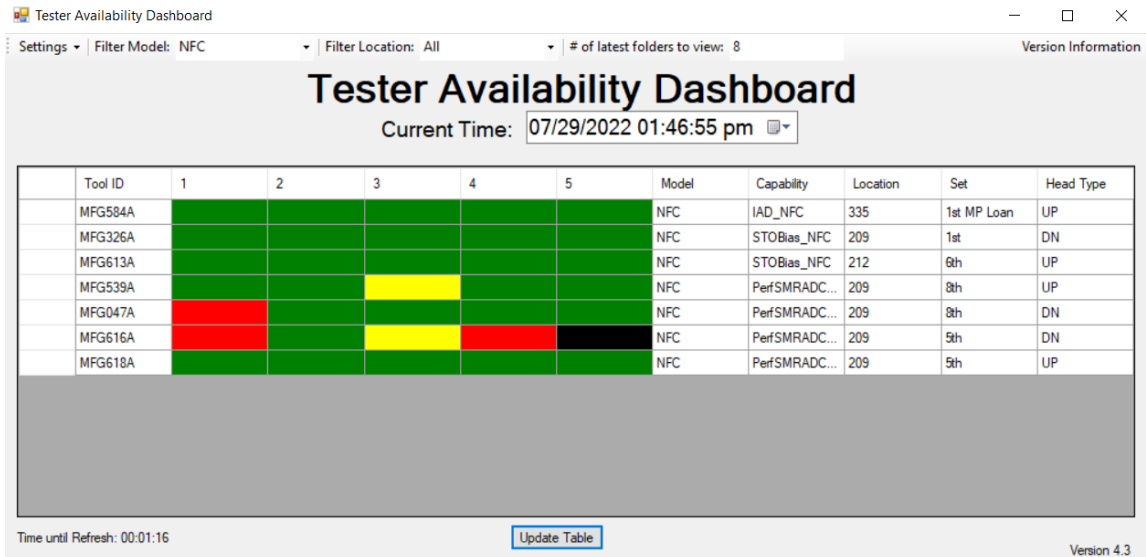


Fig. 7. Filtering All of the NFC Models from the List

Figure 8 shows the directory change feature of the dashboard. The user can choose the specific model from the drop-down menu and set the directory path through manual input or file browsing. A CommonOpenFileDialog was applied in doing this.

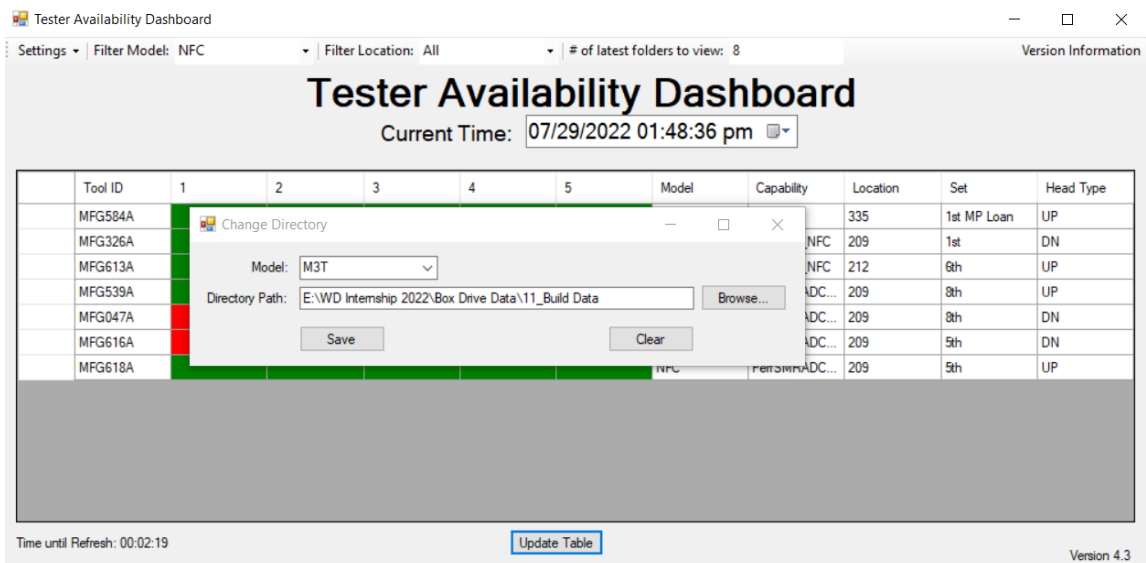


Fig. 8. Change Directory Settings

Therefore, it makes directory path browsing easier, like the usual way of locating a directory path on the computer, as shown in Figure 9.

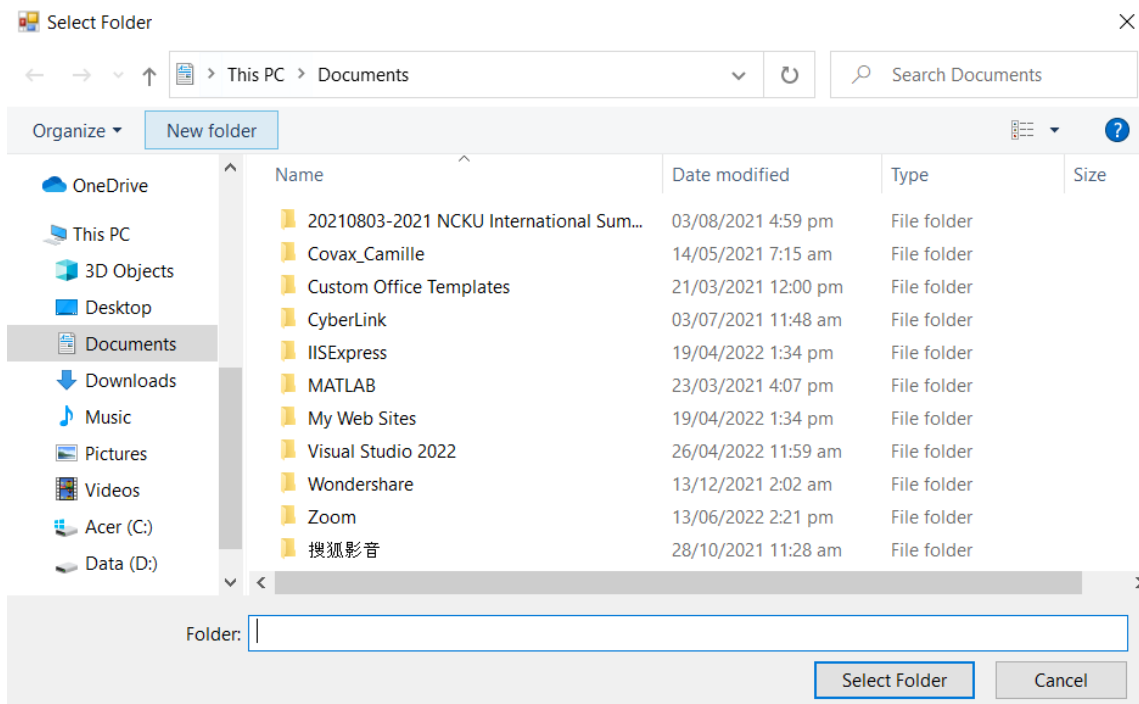


Fig. 9. CommonOpenFileDialog for Directory Path

Figure 10 shows the user interface for the colour status parameter settings. By default, the colour status parameter is set to yellow for a time greater than or equal to 30 minutes and red for a time greater than or equal to 58 minutes. To set the specific colour status parameter based on its capability, the user needs to input the capability name on the column and set the colour tolerance for both yellow and red. For green tolerance, any time below the set yellow tolerance is automatically set as green tolerance. It is also indicated in the pop-up so that the user is aware of it. A pop-up error message appears whenever the user inputs 00:00:00 to any colour status parameters and when the red tolerance value is less than the yellow tolerance value since these cases are invalid.

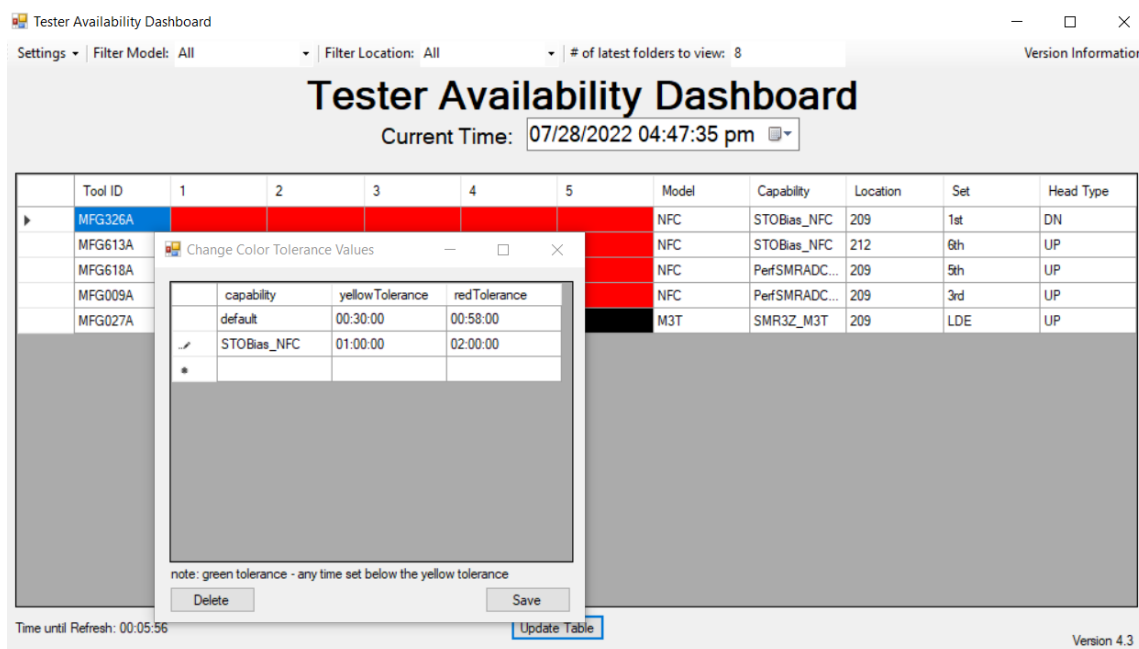


Fig. 10. Setting Up the Colour Status Parameter Based on its Capability

Figure 11 shows the pop-up message when a colour status parameter is set to a specific capability.

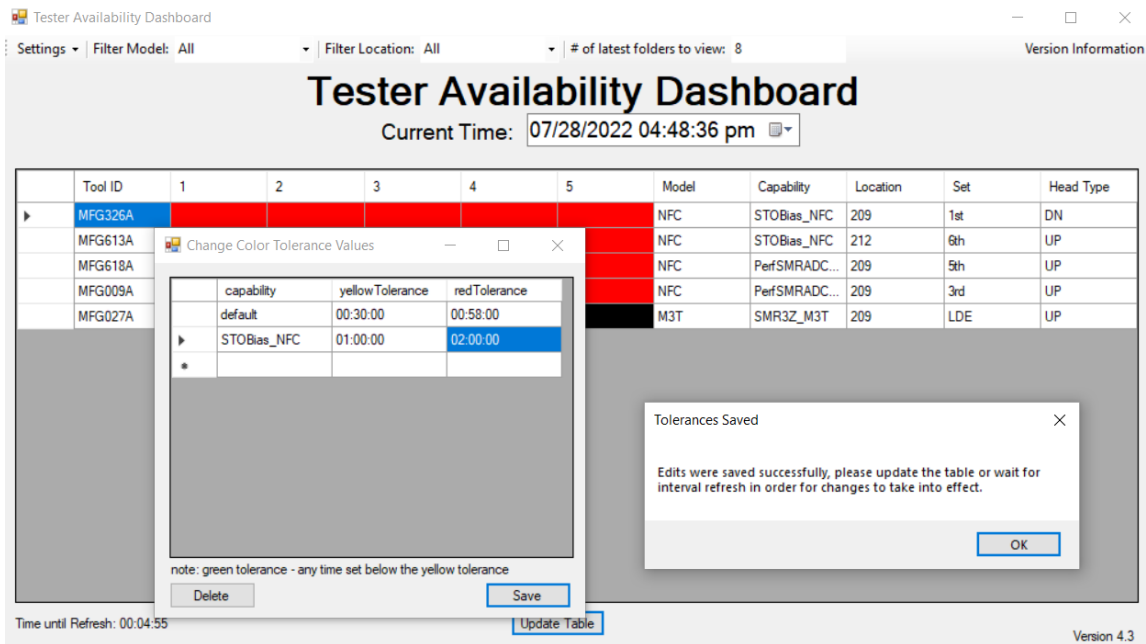


Fig. 11. A Pop-up Confirmation Message Appears When Settings are Saved

Figure 12 shows the updated dashboard after the end user sets the colour status parameter of STOBias NFC.

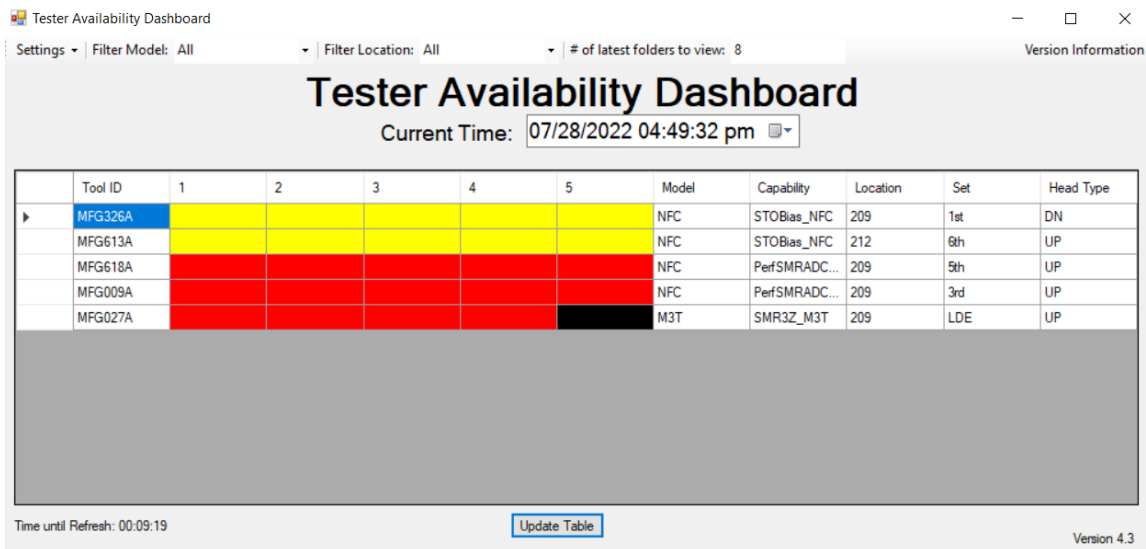


Fig. 12. Updated Table After the Saved Changes Took Effect

Figure 13 shows the user interface for the add and remove models. By default, the available models are only limited to NFC and M3T. To not limit the software to two models, the add and remove function was added. The user only needs to input the specific model to be added and click on the "add model" button. Then the model automatically becomes available on the list for directory path setting.

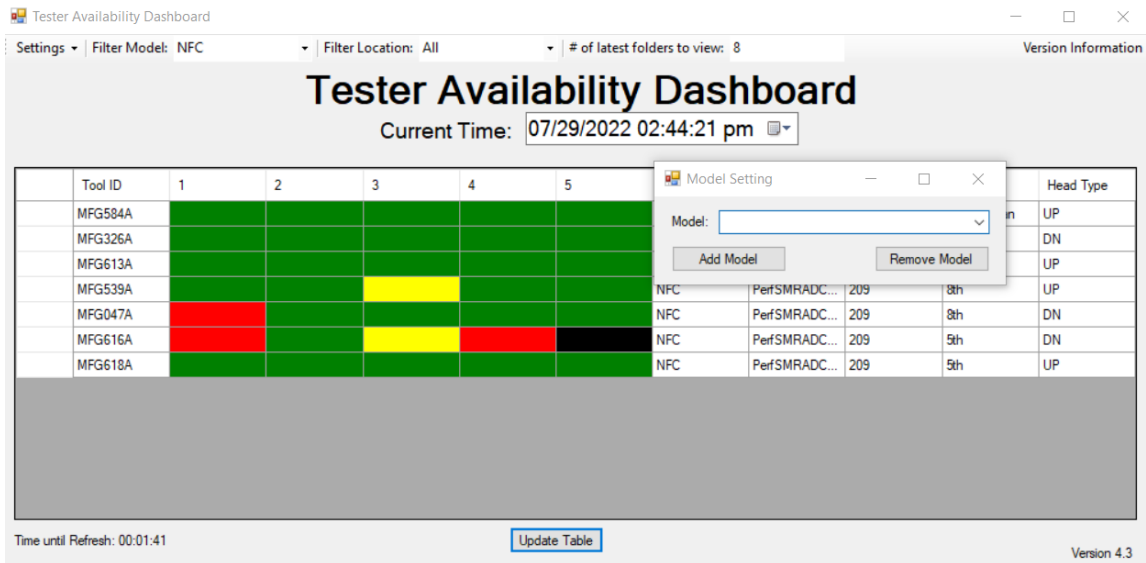


Fig. 13. User Interface for Add and Remove of Models

Figure 14 shows the pop-up window for adding models. The user must type the specific model and click Add Model to save.

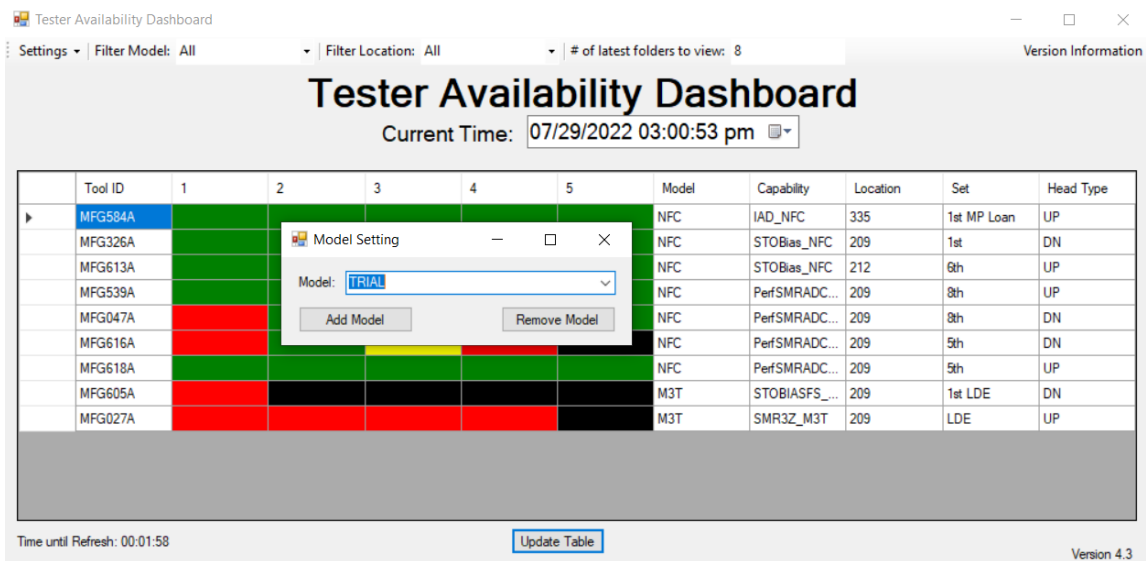


Fig. 14. Sample for Adding of Model

Once the added model is saved, a pop-up message appears, as shown in Figure 15.

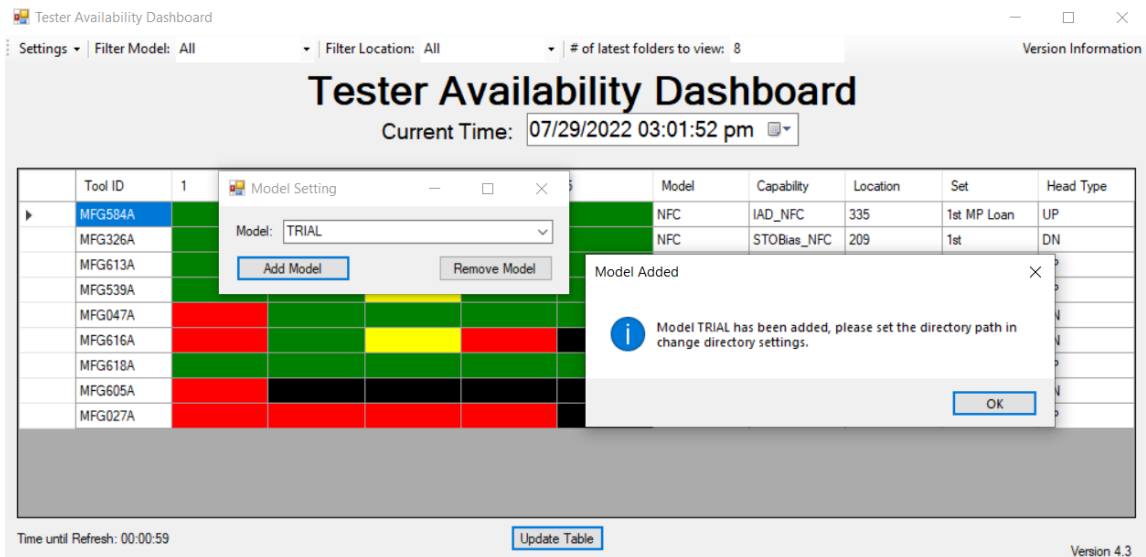


Fig. 15. A Pop-up Confirmation Message for Added Model

Figure 16 shows the change directory pop-up window. This function automatically appears whenever the end user adds a new model to the dashboard to read the files from the desired directory path.

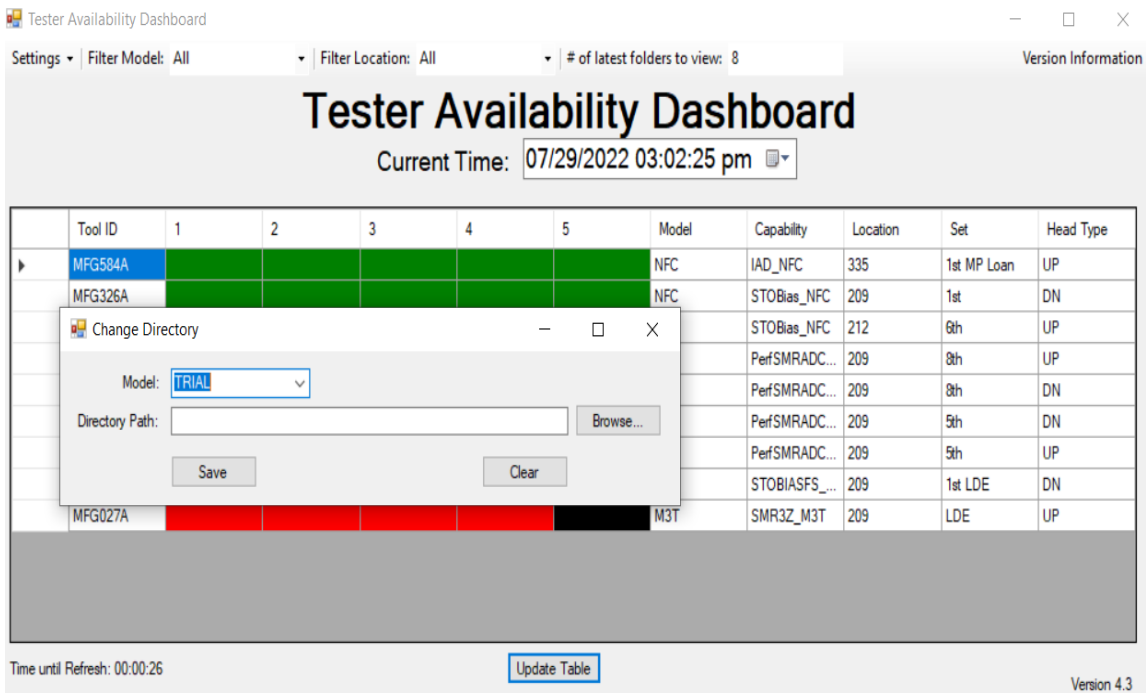


Fig. 16. A Change Directory Pop-Up Appears Automatically After Adding a Model

Figure 17 shows the drop-down menu for removing a model. The pop-up window for adding and removing model are combined into one. When adding a model, the end user inputs the specific model's name in the combo box and chooses from the drop-down menu when removing a model.

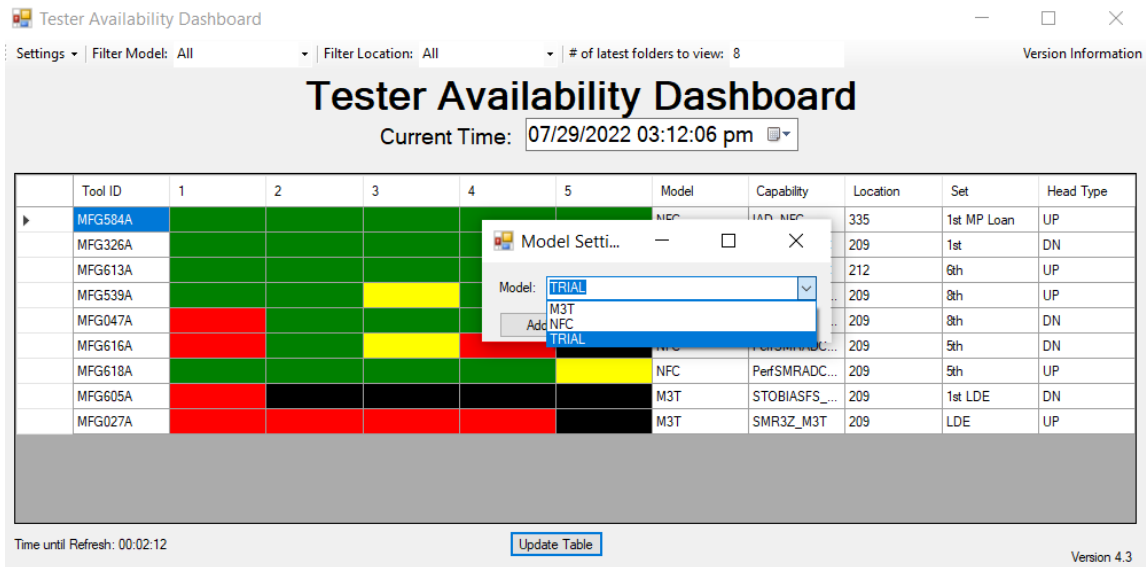


Fig. 17. Drop-Down Menu for Removing of Model

When a model is removed after the user clicks Remove Model, a pop-up message appears to show confirmation, as shown in Figure 18.

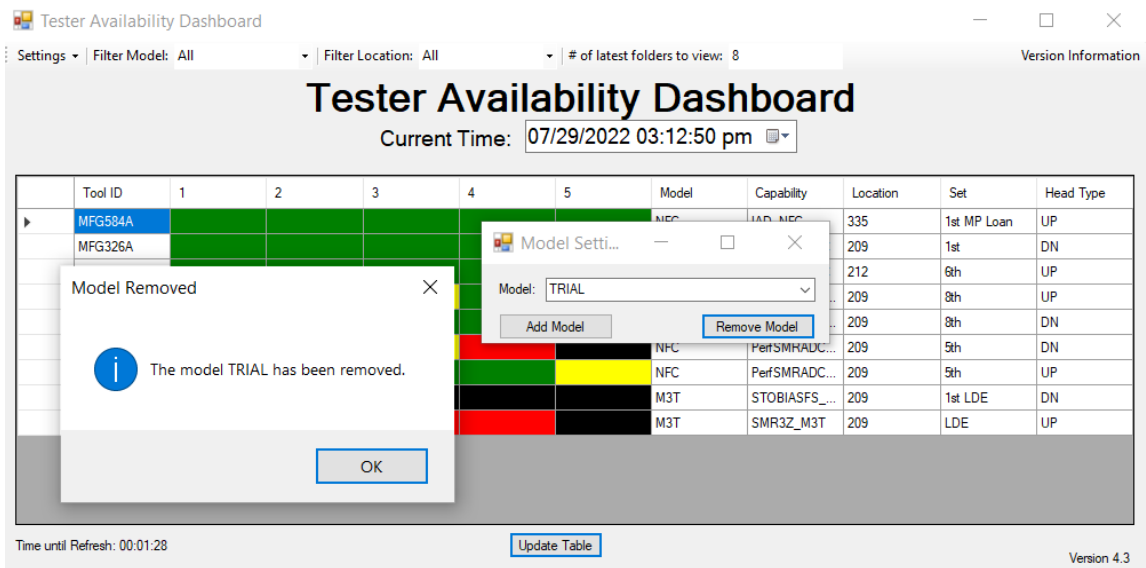


Fig. 18. Sample for Removing of Model

After removing the model, that model is immediately deleted from the drop-down list, as shown in Figure 19.

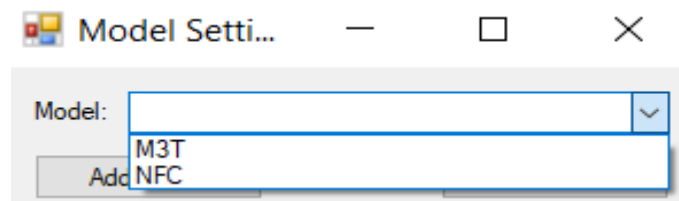


Fig. 19. User Interface for List of Available Models

Figure 20 summarizes the data collected for MFG616A on June 07, 2022. On average, the slots on this tester are approximately updated every 6 minutes and 49 seconds, and each slot is updated at about the same time. This output shows that when the technician updates a specific slot in the tester, it also updates the rest of it unless the slot is faulty.

Tool ID	slot1 latestTS	slot2 latestTS	slot3 latestTS	slot4 latestTS	slot5 latestTS
MFG616A	06/07/2022 17:12		06/07/2022 17:24		06/07/2022 17:35
MFG616A	06/07/2022 17:18	0:05:24	06/07/2022 17:29	0:05:24	06/07/2022 17:41
MFG616A	06/07/2022 17:28	0:10:21	06/07/2022 17:40	0:10:21	06/07/2022 17:49
MFG616A	06/07/2022 17:33	0:05:23	06/07/2022 17:45	0:05:22	06/07/2022 17:54
MFG616A	06/07/2022 17:42	0:08:08	06/07/2022 17:53	0:08:12	06/07/2022 18:02
MFG616A	06/07/2022 17:47	0:05:23	06/07/2022 17:59	0:05:20	06/07/2022 18:07
MFG616A	06/07/2022 17:55	0:08:08	06/07/2022 18:07	0:08:16	06/07/2022 18:15
MFG616A	06/07/2022 18:00	0:05:23	06/07/2022 18:12	0:05:23	06/07/2022 18:21
MFG616A	06/07/2022 18:08	0:07:58	06/07/2022 18:21	0:08:20	06/07/2022 18:29
MFG616A	06/07/2022 18:14	0:05:22	06/07/2022 18:26	0:05:24	06/07/2022 18:35
MFG616A	06/07/2022 18:22	0:07:52	06/07/2022 18:34	0:07:55	06/07/2022 18:43
MFG616A	06/07/2022 18:27	0:05:22	06/07/2022 18:39	0:05:22	06/07/2022 18:48
MFG616A	06/07/2022 18:35	0:08:09	06/07/2022 18:47	0:08:08	06/07/2022 18:57
MFG616A	06/07/2022 18:41	0:05:21	06/07/2022 18:53	0:05:23	06/07/2022 19:02
MFG616A	06/07/2022 18:49	0:08:12	06/07/2022 19:01	0:08:07	06/07/2022 19:10
MFG616A	06/07/2022 18:54	0:05:23	06/07/2022 19:06	0:05:24	06/07/2022 19:16
MFG616A	06/07/2022 19:02	0:08:05	06/07/2022 19:15	0:08:16	06/07/2022 19:24
MFG616A	06/07/2022 19:08	0:05:22	06/07/2022 19:20	0:05:23	06/07/2022 19:29
MFG616A	06/07/2022 19:16	0:08:01	06/07/2022 19:28	0:08:03	06/07/2022 19:38
MFG616A	06/07/2022 19:21	0:05:22	06/07/2022 19:33	0:05:23	06/07/2022 19:43
MFG616A	06/07/2022 19:29	0:08:15	06/07/2022 19:42	0:08:10	06/07/2022 19:51
MFG616A	06/07/2022 19:35	0:05:23	06/07/2022 19:47	0:05:23	06/07/2022 19:57
MFG616A	06/07/2022 19:43	0:08:07	06/07/2022 19:55	0:08:11	06/07/2022 20:05
MFG616A	06/07/2022 19:48	0:05:23	06/07/2022 20:01	0:05:22	06/07/2022 20:08
MFG616A	06/07/2022 19:57	0:08:29	06/07/2022 20:09	0:08:02	06/07/2022 20:13
MFG616A	06/07/2022 20:02	0:05:24	06/07/2022 20:14	0:05:22	06/07/2022 20:21
MFG616A	06/07/2022 20:11	0:08:33	06/07/2022 20:22	0:08:17	06/07/2022 20:27
MFG616A	06/07/2022 20:16	0:05:24	06/07/2022 20:28	0:05:22	06/07/2022 20:35
MFG616A	06/07/2022 20:24	0:08:25	06/07/2022 20:36	0:08:35	06/07/2022 20:42
MFG616A	06/07/2022 20:30	0:05:22	06/07/2022 20:42	0:05:26	06/07/2022 20:48
MFG616A	06/07/2022 20:38	0:08:16	06/07/2022 20:50	0:08:10	06/07/2022 20:54
MFG616A	06/07/2022 20:43	0:05:25	06/07/2022 20:55	0:05:23	06/07/2022 21:05
MFG616A	06/07/2022 20:52	0:08:23	06/07/2022 21:04	0:08:28	06/07/2022 21:07
MFG616A	06/07/2022 20:57	0:05:17	06/07/2022 21:09	0:05:22	06/07/2022 21:16
MFG616A	06/07/2022 21:06	0:08:34	06/07/2022 21:17	0:08:22	06/07/2022 21:21
MFG616A	06/07/2022 21:11	0:05:23	06/07/2022 21:23	0:05:23	06/07/2022 21:29
MFG616A	06/07/2022 21:19	0:08:19	06/07/2022 21:31	0:08:24	06/07/2022 21:35
MFG616A	06/07/2022 21:25	0:05:23	06/07/2022 21:37	0:05:22	06/07/2022 21:43
average time	0:06:49	0:06:49	0:06:49	0:06:48	0:06:49

Fig. 20. Summary of the Collected Data for MFG616A on June 07, 2022

Table 4 shows the average time of each tester slot. A total of 50 samples were collected for each tester slot then its average was computed. Among these five testers, MFG616A, MFG605A, MFG270A, MFG606A, and MFG516A, the average time ranges from 6 minutes 49 seconds to 3 hours 17 minutes 9 seconds. It can be seen that some testers are left unused for an extended period compared to others. It is often the case, especially at midnight, when the data stops updating after 11:30 PM and resumes between 2:00 AM and 5:30 AM. The ideal time for these tester slots to be updated is within 20 minutes; therefore, a significant delay needs to be improved to optimize the system.

Table 4
 Summary of the Collected Average Time of the Testers

Tool ID	Slot1 average time	Slot2 average time	Slot3 average time	Slot4 average time	Slot5 average time
MFG616A	00:06:49	00:06:49	00:06:49	00:06:48	00:06:49
MFG605A	00:20:03	01:19:01	00:11:35	01:01:55	03:17:09
MFG270A	01:18:44	01:00:01	00:43:18	00:56:18	01:20:07
MFG606A	00:30:14	00:45:17	00:33:25	01:19:32	00:41:55
MFG516A	00:35:22	00:37:43	00:33:25	00:28:06	00:36:37

Table 5 shows the comparison of time between performing manual checking and using the manual update function of the proposed dashboard.

Table 5
 Manual Checking of .csv Files vs. Manual Update in the Dashboard

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average Time
Manual, t1	00:45:00	00:52:23	00:48:45	00:55:12	01:23:34	00:56:59
Using the Tester Availability Dashboard, t2	00:00:03	00:00:02	00:00:05	00:00:03	00:00:05	00:00:04
Time Difference	00:44:57	00:52:21	00:48:40	00:55:09	01:23:29	00:56:55
% Difference	199.556%	199.746%	199.317%	199.638%	199.602%	199.533%

Table 6 shows the comparison between manually saving the data and downloading the history log of the proposed dashboard.

Table 6
 Saving the Collected Data from History Log vs. Saving Each Data One by One Manually

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average Time
Manual, t1 (per tester of 5 slots with 50 samples)	01:06:07	00:56:58	01:02:12	00:58:55	00:55:15	00:59:53
Using the history log function, t2 (regardless of the number of testers data)	00:00:21	00:00:16	00:00:19	00:00:23	00:00:18	00:00:19
Time Difference	01:05:46	00:56:42	01:01:53	00:58:32	00:54:57	00:59:34
% Difference	197.894%	198.136%	197.974%	197.414%	197.840%	197.852%

4. Conclusion

After doing the work, it was concluded that the proposed system was able to improve the overall efficiency of the whole system by a significant amount. As data increases, industries must apply automation systems to their daily tasks, which help increase the system's overall efficiency.

The authors recommend adding a beep sound and a light indicator for future works when a tester slot reaches a specific unused time. At the same time, the dashboard can further be improved by displaying the status of each blade being tested in the tester slot in another window of the same application. With this, the end users can quickly evaluate whether the current blade being tested meets the required criteria or not to reduce the time of checking.

The authors suggest adding more functionalities in the dashboard, such as adding a summary table regarding the frequent top 10 tester slots left unused for the most extended period and applying statistical analysis to the data. With this, the end user can pay more attention to these tester slots to further optimize the system.

Acknowledgement

This research was funded by a grant from De La Salle University.

References

- [1] Forbes. "Western Digital | WDC Stock Price, Company Overview & News." <https://www.forbes.com/companies/western-digital/?sh=5d898a366818>
- [2] "Western Digital Corp. - Company Profile, Information, Business Description, History, Background Information on Western Digital Corp." <https://www.referenceforbusiness.com/history2/76/Western-Digital-Corp.html>
- [3] WhatIs.com. "What Is Western Digital Corporation (WDC)?" <https://www.techtarget.com/whatis/definition/Western-Digital-Corporation-WDC>
- [4] chcom. "Western Digital Corporation (WDC)." CompaniesHistory.com - the Largest Companies and Brands in the World. 2022. <https://www.companieshistory.com/western-digital-corporation-wdc>
- [5] www.zippia.com. "Western Digital History: Founding, Timeline, and Milestones." 2020. <https://www.zippia.com/western-digital-careers-12753/history>
- [6] Frank, Paul, and Roger Wood. "A perspective on the future of hard disk drive (HDD) technology." In *Asia-Pacific Magnetic Recording Conference 2006*, pp. 1-2. IEEE, 2006. <https://doi.org/10.1109/APMRC.2006.365887>
- [7] Kang, Yangwook, Yang-suk Kee, Ethan L. Miller, and Chanik Park. "Enabling cost-effective data processing with smart SSD." In *2013 IEEE 29th symposium on mass storage systems and technologies (MSST)*, pp. 1-12. IEEE, 2013. <https://doi.org/10.1109/MSST.2013.6558444>
- [8] Bright, Jonathan D., and John A. Chandy. "A scalable architecture for clustered network attached storage." In *20th IEEE/11th NASA Goddard Conference on Mass Storage Systems and Technologies, 2003.(MSST 2003). Proceedings.*, pp. 196-206. IEEE, 2003.
- [9] Guijarro, Manuel, and Ruben Gaspar. "Experience and lessons learnt from running high availability databases on network attached storage." In *Journal of Physics: Conference Series*, vol. 119, no. 4, p. 042015. IOP Publishing, 2008. <https://doi.org/10.1088/1742-6596/119/4/042015>
- [10] Patil, Vijayalakshmi, and N. K. Srinath. "Validation of Storage Devices with Serial advanced Technology Attachment (SATA)." (2019).
- [11] Kao, Yao-Huang, and Yi-Bin Hsieh. "A low-power and high-precision spread spectrum clock generator for serial advanced technology attachment applications using two-point modulation." *IEEE transactions on electromagnetic compatibility* 51, no. 2 (2009): 245-254. <https://doi.org/10.1109/TEMC.2008.2012115>
- [12] Advantest Corporation. "Advantest Cloud Solutions." <https://www.advantest.com/acs/>
- [13] Rajan, Yash. "Top 7 Dashboard Software Companies." Verified Market Research. 2021.
- [14] Mills, Harlan D., Michael Dyer, and Richard C. Linger. "Cleanroom software engineering." (1987). <https://doi.org/10.1109/MS.1987.231413>
- [15] Rosli, Muhammad Ilias, and Mohd Ridzuan Ahmad. "Internet of Things Monitoring System of a Modeled Cleanroom." *ELEKTRIKA-Journal of Electrical Engineering* 20, no. 3 (2021): 7-14. <https://doi.org/10.11113/elektrika.v20n3.283>
- [16] Sharma, Hitesh Kumar, Ravi Tomar, J. C. Patni, and Ankur Dumka. "E-COCOMO: An effort estimation model for cleanroom software development approach." In *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, pp. 131-136. IEEE, 2016. <https://doi.org/10.1109/NGCT.2016.7877403>
- [17] Mingwei, Zhou. "Discussion on the relationship between clean room and traditional software engineering methods and practices." In *Proceedings of the 2nd International Conference on Artificial Intelligence and Advanced Manufacture*, pp. 18-22. 2020. <https://doi.org/10.1145/3421766.3421874>
- [18] Zhou, Lei, Xiaowei Liu, Qin Liu, and Guojun Wang. "A cleanroom monitoring system for network computing service based on remote attestation." In *2016 IEEE Trustcom/BigDataSE/ISPA*, pp. 451-457. IEEE, 2016. <https://doi.org/10.1109/TrustCom.2016.0096>
- [19] Sami, Mohamed. "Software Development Life Cycle Models and Methodologies - Mohamed Sami." 2018. <https://melsatar.blog/2012/03/15/software-development-life-cycle-models-and-methodologies/>.
- [20] GeeksforGeeks. "Difference between Prototype Model and Spiral Model." 2020. <https://www.geeksforgeeks.org/difference-between-prototype-model-and-spiral-model>
- [21] Narayan, Ritushree. "Study of Various Software Development Methodologies." *EPRA International Journal of Multidisciplinary Research (IJMR)* (2021).
- [22] Peled, Doron A. *Software reliability methods*. Springer Science & Business Media, 2013.
- [23] Tian, Jeff. *Software quality engineering: testing, quality assurance, and quantifiable improvement*. John Wiley & Sons, 2005. <https://doi.org/10.1002/0471722324>
- [24] Erdogmus, Hakan, Maurizio Morisio, and Marco Torchiano. "On the effectiveness of the test-first approach to programming." *IEEE Transactions on software Engineering* 31, no. 3 (2005): 226-237. <https://doi.org/10.1109/TSE.2005.37>

- [25] Sharma, Hitesh Kumar, Ravi Tomar, Ankur Dumka, and M. S. Aswal. "Openecocomo: The algorithms and implementation of extended cost constructive model (e-cocomo)." In *2015 1st International Conference on Next Generation Computing Technologies (NGCT)*, pp. 773-778. IEEE, 2015. <https://doi.org/10.1109/NGCT.2015.7375225>
- [26] Hovorushchenko, Tetiana, and Olga Pavlova. "Intelligent System for Determining the Sufficiency of Metric Information in the Software Requirements Specifications." In *CMIS*, pp. 253-266. 2019. <https://doi.org/10.32782/cmisis/2353-20>