



ARL-SR-0469 • MAR 2023



MIL-PRF-32662 Context-Aware Agile Platform

by Robert Jensen, Kimberly Brady, Prathyush Parvatharaju,
Biao Yin, Fatemeh Emdad, and Elke Rundensteiner

Approved for public release: distribution unlimited.

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.



MIL-PRF-32662 Context-Aware Agile Platform

Robert Jensen

DEVCOM Army Research Laboratory

**Kimberly Brady, Prathyush Parvatharaju, Biao Yin, Fatemeh Emdad,
and Elke Rundensteiner**

Department of Data Science, Worcester Polytechnic Institute

REPORT DOCUMENTATION PAGE

1. REPORT DATE		2. REPORT TYPE		3. DATES COVERED	
March 2023		Special Report		START DATE 01-01-2019	END DATE 01-31-2023
4. TITLE AND SUBTITLE MIL-PRF-32662 Context-Aware Agile Platform					
5a. CONTRACT NUMBER W911NF-19-2-0112		5b. GRANT NUMBER		5c. PROGRAM ELEMENT NUMBER	
5d. PROJECT NUMBER		5e. TASK NUMBER		5f. WORK UNIT NUMBER	
6. AUTHOR(S) Robert Jensen, Kimberly Brady, Prathyush Parvatharaju, Biao Yin, Fatemeh Emdad, and Elke Rundensteiner					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) DEVCOM Army Research Laboratory ATTN: FCDD-RLA-MC Aberdeen Proving Ground, MD 21005				8. PERFORMING ORGANIZATION REPORT NUMBER ARL-SR-0469	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)		11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release: distribution unlimited.					
13. SUPPLEMENTARY NOTES ORCID ID: Robert Jensen, 0000-0003-3824-4767					
14. ABSTRACT MIL-PRF-32662 seeks to transition a traditionally “static” Qualified Products List to a “dynamic” version using a Context-Aware Agile Platform (CAAP) web-based dashboard. The dashboard was developed in a collaborative effort between the US Army Combat Capabilities Development Command Army Research Laboratory (ARL) and the Department of Data Science at Worcester Polytechnic Institute through a series of student-led Graduate Qualifying Projects. Commercial end-user review was provided by PPG and ASM International. The Adhesive Dashboard allows materials scientists and data scientists to collect, upload, share, analyze, and visualize experimental results for adhesive materials. Users can also view data from other scientists and vendors to search for an adhesive with specific properties that meet their needs. The CAAP dashboard will ultimately ease the burden of the laborious and cumbersome process that scientists and researchers must follow to introduce new adhesives for DoD and commercial solutions. This ARL special report includes a user’s manual with the steps to use each feature and tool of the platform, as well as software documentation and an installation guide.					
15. SUBJECT TERMS Science of Extreme Materials, military performance standards, qualified products list, context-aware agile platform, adhesives, data science					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT		18. NUMBER OF PAGES
a. REPORT UNCLASSIFIED	b. ABSTRACT UNCLASSIFIED	c. THIS PAGE UNCLASSIFIED	UU		55
19a. NAME OF RESPONSIBLE PERSON Robert E Jensen				19b. PHONE NUMBER (Include area code) (240) 676-9401	

STANDARD FORM 298 (REV. 5/2020)

Prescribed by ANSI Std. Z39.18

Contents

List of Figures	v
List of Tables	v
Acknowledgments	vi
1. Introduction	1
2. Dashboard Summary	2
3. Navigating the Dashboard, Adjusting Views, and Using Basic Functions	3
3.1 Navigate the Dashboard	3
3.2 Adjust the Style, Log In and Out, and Other Top Menu Functions	4
4. Uploading, Querying, and Visualizing Datasets with Dashboard Tools	4
4.1 Upload Data to the File Manager	4
4.2 View, Sort, and Query Data with the Data Wrangler Tool	6
4.2.1 Basic Data Queries	7
4.2.2 Advanced Data Queries: Conditional Filtering	8
4.3 Visualizing Data	10
5. Formatting and Uploading Adhesive Experimental Data	17
5.1 Download Experiment Checklist in Digitized Records	17
5.2 Use Experiment Checklist Template	18
5.3 Upload Experiment Checklist Template in Digitized Records or File Manager	21
6. Sharing, Viewing, and Comparing Adhesives with Dashboard Features	23
6.1 Filter and Sort Summary of Adhesive Grouping	25
6.2 View Properties for Specific Adhesives	26

6.3	Viewing Results in the Dashboard View	30
7.	Modeling the Data with Machine Learning	34
8.	Accessing Documentation	38
8.1	Military Performance Specifications and Standards	38
8.2	Developer Manual	39
9.	References	40
	Appendix A. Design Thinking an End-to-End Context Aware Platform	41
	Appendix B. Platform Documentation	42
	Appendix C. Installation Instructions	43
	List of Symbols, Abbreviations, and Acronyms	45
	Distribution List	46

List of Figures

Fig. 1	Navigation pane	3
Fig. 2	Top menu functions	4
Fig. 3	Data Wrangler default view showing example data from an adhesives experiment (Note, this is not the Experimental Checklist template. The data must be in CSV format with column headers.)	6
Fig. 4	Adhesive Studio main view	24
Fig. 5	DOE default view	35

List of Tables

Table 1	Summary of Adhesive Dashboard tools and features	2
Table 2	Experiment checklist data	19

Acknowledgments

This research was sponsored by the US Army Combat Capabilities Development Command (DEVCOM) Army Research Laboratory (ARL) and was accomplished under Cooperative Agreement Numbers W911NF-19-2-0112 (Worcester Polytechnic Institute [WPI] Graduate Qualifying Projects [GQPs]). Partial support was also provided under Cooperative Agreements W911NF-20-2-0232 (Oak Ridge Associated Universities, Mr Prathyush Parvatharaju) and W911NF-17-2-0227 (PPG Industries – Allison Park PA, Dr Marvin Pollum, Dr Martin Losada, and Mr Brian Kornish). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of ARL or the US Government. The US Government is authorized to reproduce and distribute reprints for Government purposes not withstanding any copyright notation herein.

The following WPI GQP students are recognized for their technical contributions.

WPI student	GQP semester
Chu Wang	Spring 2019
Brandon Werner	Spring 2019
Prince Shiva Chaudhary	Spring 2019
Habibah Ashouri Choshali	Spring 2019
Omkar Kulkarni	Fall 2019
Maryam Masroor Shalmani	Fall 2019
Karthika Suresh	Fall 2019
Alexander Wurts	Spring 2020
Renee Sweeney	Spring 2020
Bowen Tang	Spring 2020
Revant Mahajan	Fall 2020
Edward Carlson	Fall 2020
Tai Zhou	Fall 2020
Geng Zhao	Fall 2020
Prakhar Srivastava	Fall 2020
Joseph Caltabiano	Spring 2021
Kodzo Agbobli	Spring 2021
Daojun Liang	Spring 2021
Lin Zhang	Spring 2021
Yanzhao Wang	Spring 2021
Mario Arduz	Spring 2021
Shundong Li	Spring 2021
Ashwin Pai	Spring 2022
Philip Rago	Spring 2022
Kartik Nautiyal	Spring 2022
Muralidhar Koripalli	Fall 2022
Shrinivas Sanglikar	Fall 2022
Wenrui Zheng	Spring 2023
Victoria Mirecki	Spring 2023
Whitney Pavlova	Spring 2023
Jeff Bloom	Spring 2023

The following WPI graduate students, faculty, and staff are recognized for providing technical assistance to the GQPs: Mr Nicholas Josselyn, Prof Chun-Kit Ngan, Prof Pratap Rao, Prof Juan Zou, and Dr Ermal Toto.

Finally, the assistance of ASM International (Materials Park, OH) is recognized in resolving technical barriers in transitioning the software package from an academic environment to preproduction testing, specifically Dr Afina Lupulescu, Dr Ray Fryan, Mr Kelvin Scott, Mr James Bursby, Mr Seth Kimble, Mr Evan Tuttle, Mr Donnie Gies, Mr Nate Bulcroft, Ms Mary Reynolds, Mr Edward Stempinski, and Ms Stephanie Hunsinger.

1. Introduction

The Army continually needs a multitude of adhesives with materials properties beyond the current state-of-the-art to optimize the protection-defeat mechanism efficiencies of lightweight bonded armors. To accomplish this objective, MIL-PRF-32662¹ was drafted to provide minimally constrained vetting criteria as performance objectives for higher risk adhesives research and development. Innovation and technical standards share an interdependent relationship, with a common perception being that standards can act as barriers to impede innovation. Standards have co-evolved as measures of late lifecycle quality assurance with long-term historical usage and well-understood allowable designs. MIL-PRF-32662 represents a forward-looking standard² to coincide with the timing of emergent life cycle adhesives technologies and to draw R&D attention to unique Army-derived performance needs.

MIL-PRF-32662 enables data science, artificial intelligence/machine learning (AI/ML), and data visualization to increase the efficiency of the transactional information exchange between the standard development organization and vendor. Data visualization of standards' accepted test results and associated meta-data descriptors provide a high degree of transparency and insight for continued advancement of the "art." Furthermore, the application of AI algorithms, data science, and deep learning can also provide the ability for datasets acquired for MIL-PRF-32662 to remain continually or increasingly relevant. By providing transparent and standardized data, ultimately formatted for AI/ML-assisted discovery, MIL-PRF-32662 allows both adhesive vendors and end users a platform to expand adhesive property-performance correlations to many applications.

MIL-PRF-32662 imparts the potential to obsolete traditional "static" Qualified Products Lists (QPLs) in favor of "dynamic" versions using a Context-Aware Agile Platform (CAAP) dashboard. An Adhesives Dashboard was developed in a collaborative effort between the US Army Combat Capabilities Development Command (DEVCOM) Army Research Laboratory (ARL) and the Department of Data Science at Worcester Polytechnic Institute through a series of student-led Graduate Qualifying Projects (GQPs). Commercial end-user review was provided by PPG and ASM International. The Adhesive Dashboard allows materials scientists and data scientists to collect, upload, share, analyze, and visualize experimental results for adhesive materials. Users can also view data from other scientists and vendors to search for an adhesive with specific properties that meet their needs. The Adhesives Dashboard will ultimately ease the burden of the laborious and cumbersome process that scientists and researchers must follow to introduce new adhesives for DoD and commercial solutions.

2. Dashboard Summary

The Adhesive Dashboard's card-based user interface design, shown in Table 1, makes it easy for any user to navigate and understand experimental results. It offers a variety of features and tools, which are summarized in this section. The subsequent sections describe the steps to use each feature and tool. The User's Guide for the Adhesive Dashboard is described in the following sections, while detailed documentation for the CAAP is provided in Appendix A.

Table 1 Summary of Adhesive Dashboard tools and features

	Dashboard (home screen): Summarizes uploaded adhesive experiment data, such as total number of experiments, mean strength, and displacement at failure for most/least tested adhesives, and ranges of experimental conditions for numerical and categorical values. Tools: Provide materials and data scientists with data analysis and visualization functionality for any dataset, whether for materials science or other needs. • File Manager: Upload any dataset in CSV format and integrate with the Data Wrangler and Data Visualizer. • Data Wrangler: View, clean, transform, and sort uploaded data using basic or advanced queries. • Data Visualizer: Create different visualizations to display patterns and trends in data. Features: Provide materials scientists with capabilities to easily share and view experimental results for adhesive materials. • Digitized Records: Use the provided template, called the Experiment Checklist, to enter experimental settings and results, then upload the Checklist to integrate data with the Adhesive Studio and Design of Experiment (DOE) features. • Adhesive Studio: View summaries of test results for adhesive experiments uploaded by all users in Digitized Records; search for an adhesive or by desired properties; showcase your adhesive to a wide audience. • DOE (Design of Experiment): Run Machine Learning Algorithms to determine relationships between variables and view recommended experiments to help reach adhesive material goals (e.g., certain strength). Documentation: Provides access to view and download MIL-PRF-32662 and related documents.
--	---

3. Navigating the Dashboard, Adjusting Views, and Using Basic Functions

The Adhesive Dashboard includes the following main functions and screens:

- Overall summary view: Dashboard screen (Section 6)
- Three Features: File Manager, Data Wrangler, Data Visualizer (Section 4.3)
- Three Tools: Digitized Records (Section 5), Adhesive Studio (Section 6), DOE (Section 7)
- Access to relevant Documentation (Section 8)

The user has options to adjust the overall view, such as changing the coloring and screen sizing.

This section describes how to navigate the dashboard, adjust certain settings, log in and out, and access other functions in the side and top menus.

3.1 Navigate the Dashboard

Users can navigate from screen to screen using the Navigation Pane (Fig. 1). This is always present unless it is hidden by the user.

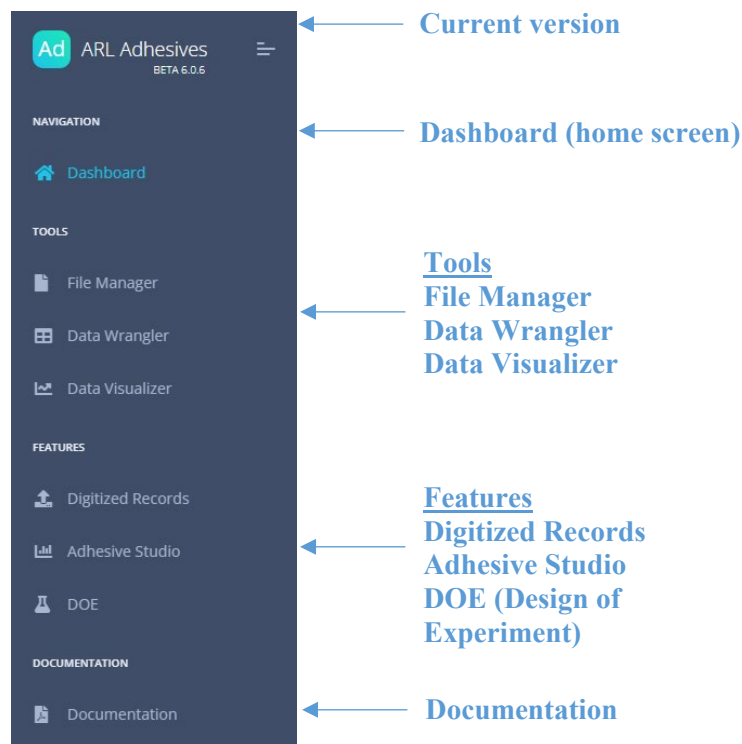


Fig. 1 Navigation pane

To access any of the tools or features, click on the tool or feature name in the Navigation Pane.

3.2 Adjust the Style, Log In and Out, and Other Top Menu Functions

The top menu (Fig. 2) provides various functionality and shows the current version.

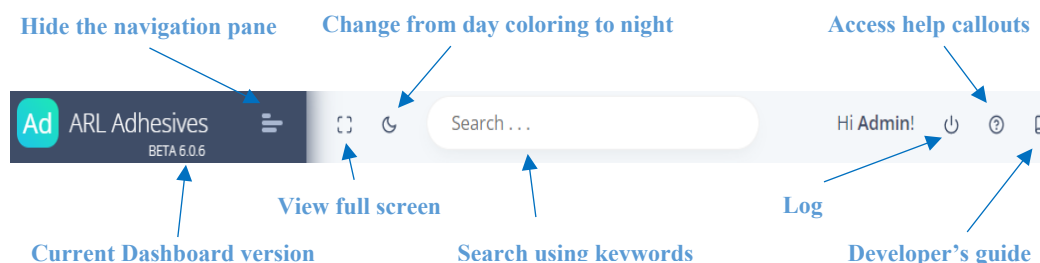


Fig. 1 Top menu functions

To access and use any of these functions, click on the icons.

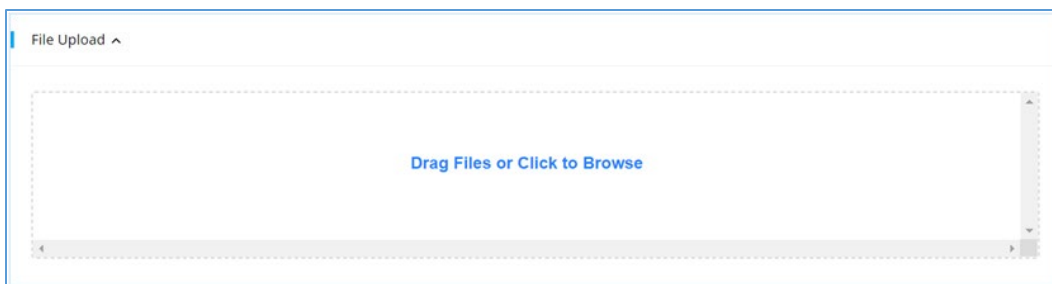
4. Uploading, Querying, and Visualizing Datasets with Dashboard Tools

The Adhesive Dashboard provides interactive tools (File Manager, Data Wrangler, Data Visualizer) that enable the user to upload data in CSV format to then view, sort, and query the data (numerical and categorical).

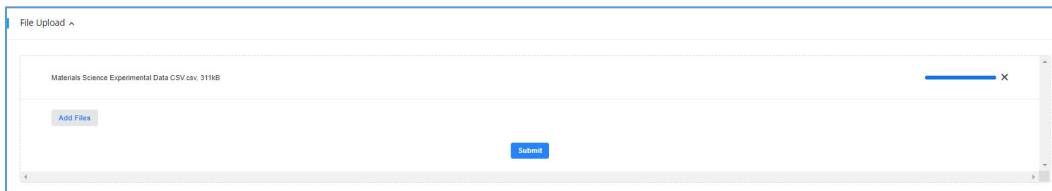
4.1 Upload Data to the File Manager

The File Manager ingests CSV files and integrates with the Data Wrangler and Data Visualizer for analysis. Datasets with column headers can be uploaded. (Note: Do not upload the Experiment Checklist here. This feature is separate from the Tools, as described in Section 5.)

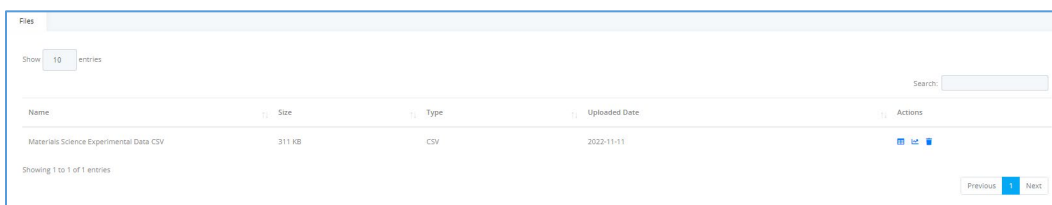
- 1) To open the File Manager, click **File Manager** in the Navigation Pane.
- 2) To upload files, drag them into the box or click to browse files on your computer.



3) Click **Submit** when prompted.



The file(s) will appear in the bottom Files panel. In this example, the user uploaded a CSV file containing information about a materials science experiment.

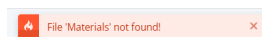


4) To add more files, click **Add Files**.

5) To load the data into the Data Wrangler, click the Data Wrangler icon (📊) in the Actions column. See Section 4.2 for instructions for the Data Wrangler.

6) To load the data into the Data Visualizer, click the Data Visualizer icon (📈) in the Actions column. See Section 4.3 for instructions for the Data Visualizer.

a. Note, you must follow these steps to view the data in the Data Wrangler and Data Visualizer. Otherwise, you will see the following error in the top right of your screen:



7) To delete the file, click the Trash icon (🗑️) in the Actions column.

4.2 View, Sort, and Query Data with the Data Wrangler Tool

After clicking the Data Wrangler icon (see Section 4.1), the user will be brought to the Data Wrangler Compact View (Fig. 3).

The screenshot shows the Data Wrangler Compact View interface. At the top, there's a header bar with the file name 'Materials Science Experimental Data CSV', file size '311 KB', and date '2022-11-11'. Below this, there's a 'Compact View' tab and a 'Summary' tab. The 'Compact View' tab is active, showing a table of data. The table has columns: MSAT ID, Operator, Date test performed, Valid test, Used in Summary, Lot ID, Material ID, Lap Shear joint substrate aluminum thickness (m), Average thickness of the adhesive layer (m), Spacer shim thickness (m), and Sample. The table contains 11 rows of data. At the bottom, there's a search bar and a pagination bar showing 'Showing 1,161 to 1,170 of 1,202 entries'.

MSAT ID	Operator	Date test performed	Valid test	Used in Summary	Lot ID	Material ID	Lap Shear joint substrate aluminum thickness (m)	Average thickness of the adhesive layer (m)	Spacer shim thickness (m)	Sample
20130152	ARL Technician 1	2456406	Yes	Yes	CEP100-ARL-2012-RT-57	CEP100-epoxy	0.00114	0.00072	0.000762	ARL T
20130153	ARL Technician 1	2456406	Yes	Yes	CEP100-ARL-2012-RT-57	CEP100-epoxy	0.00114	0.0007700000000000001	0.000762	ARL T
20130154	ARL Technician 1	2456407	Yes	Yes	CEP100-ARL-2012-RT-58	CEP100-epoxy	0.0015199999999999999	0.00052	0.000762	ARL T
20130155	ARL Technician 1	2456407	Yes	Yes	CEP100-ARL-2012-RT-58	CEP100-epoxy	0.0015199999999999999	0.00057	0.000762	ARL T
20130156	ARL Technician 1	2456407	Yes	Yes	CEP100-ARL-2012-RT-58	CEP100-epoxy	0.0015199999999999999	0.0007700000000000001	0.000762	ARL T
20130157	ARL Technician 1	2456407	Yes	Yes	CEP100-ARL-2012-RT-58	CEP100-epoxy	0.0015199999999999999	0.00073	0.000762	ARL T
20130158	ARL Technician 1	2456407	Yes	Yes	CEP100-ARL-2012-RT-58	CEP100-epoxy	0.0015199999999999999	0.0007099999999999999	0.000762	ARL T
20130159	STEM Student	2456407	Yes	Yes	CEP100-ARL-2012-RT-58	CEP100-epoxy	0.0015199999999999999	0.00052	0.000762	ARL T
20130160	STEM Student	2456407	Yes	Yes	CEP100-ARL-2012-RT-58	CEP100-epoxy	0.0015199999999999999	0.00054	0.000762	ARL T
20130161	ARL Technician 1	2456407	Yes	Yes	CEP100-ARL-2012-RT-58	CEP100-epoxy	0.0015199999999999999	0.00054	0.000762	ARL T

Fig. 3 Data Wrangler default view showing example data from an adhesives experiment (Note, this is not the Experimental Checklist template. The data must be in CSV format with column headers.)

The Data Wrangler has two main views. The **Compact View** provides the functionality to view, sort, and query the data; and the **Summary View** provides at-a-glance statistics of the uploaded data.

The main features of the **Compact View** include the following:

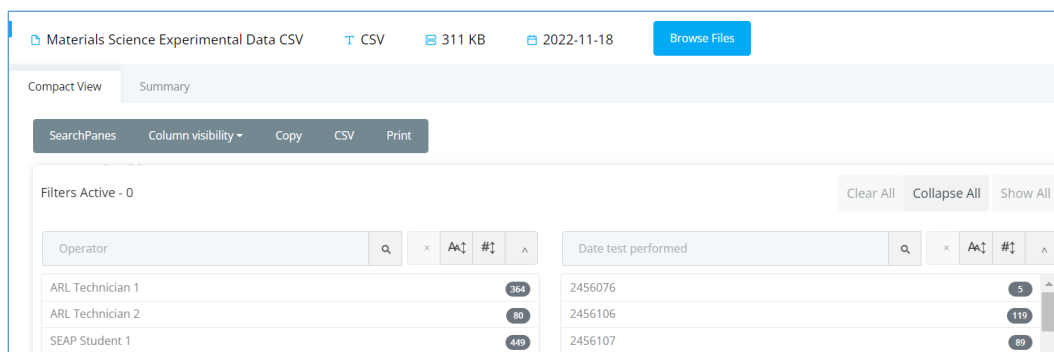
- **Columns:** Sortable columns by numerical or categorical columns (use the arrows at the top of the column to sort ascending or descending). Columns being shown are selected experimental conditions.
 - To sort, use the arrows at the top of each column.
- **Search:** Search by a single keyword
 - To search, type a single keyword into the search bar in the upper right.
- **SearchPanes:** Build a query to filter data using specific criteria
 - See Section 4.2.1 for instructions

- **Custom Search Builder:** For more advanced users, further customizable options
 - See Section 4.2.2 for instructions

The main features of the **Summary View** include summary statistics for categorical and numerical variables. The summary statistics are explained in the Summary Statistics panel description in Section 6.3.

4.2.1 Basic Data Queries

- 1) To query the data, click **SearchPanels** in the top left menu bar. You will see the Active Query screen.



The name of the file being viewed is displayed at the top (in the image above, the file name is *Materials Science Experimental Data CSV*), with the file information, including file type (CSV), size (311 KB), and upload date (2022-11-18). The data is organized by the column header (e.g., Operator), with all values in that column listed below (e.g., ARL Technician 1, ARL Technician 2, SEAP Student 1).

- 2) To return to the File Manager and view a different file, click **Browse Files** (or File Manager in the Navigation Pane).
- 3) To select a value to include in the query, click on the desired value (e.g., ARL Technician 1) and it will highlight in blue.
- 4) To select multiple values, hold down the shift key while clicking. The Filters Active counter in the top left will match the number of selected values.
- 5) To clear selections, click **Clear All** in the top right.
- 6) To collapse selections to see all available column headers at once, click **Collapse All** in the top right.

- Click **Column visibility** in the top menu to return to the columns to see the filtered data. The bottom left shows the number of entries that match the query.

MSAT ID	Operator	Date test performed	Valid test	Used in Summary	Lot ID	Material ID	Lap Shear joint substrate aluminum thickness (m)	Average thickness of the adhesive layer (m)	Spacer shim thickness (m)	Sample prepared
20120117	SEAP Student 2	2456106	No	No	SG300-ARL-2012-R7-02	SG300-methacrylate	0.00114	0.000127	0.000127	
20120118	SEAP Student 2	2456106	No	No	SG300-ARL-2012-R7-02	SG300-methacrylate	0.00114	0.000127	0.000127	
20120119	SEAP Student 2	2456106	Yes	Yes	SG300-ARL-2012-R7-02	SG300-methacrylate	0.00114	0.000152	0.000127	
20120120	SEAP Student 2	2456106	Yes	Yes	SG300-ARL-2012-R7-02	SG300-methacrylate	0.00114	0.000152	0.000127	
20120121	SEAP Student 2	2456106	Yes	Yes	SG300-ARL-2012-R7-02	SG300-methacrylate	0.00114	0.000178	0.000127	
20120122	SEAP Student 2	2456106	Yes	Yes	SG300-ARL-2012-R7-02	SG300-methacrylate	0.00114	0.000152	0.000127	
20120123	SEAP Student 2	2456106	Yes	Yes	SG300-ARL-2012-R7-02	SG300-methacrylate	0.00114	0.000178	0.000127	
20120124	SEAP Student 2	2456106	Yes	Yes	SG300-ARL-2012-R7-02	SG300-methacrylate	0.00114	0.00022999999999999999	0.000127	
20120125	SEAP Student 2	2456106	Yes	Yes	SG300-ARL-2012-R7-02	SG300-methacrylate	0.00114	0.000254	0.000127	
20120126	SEAP Student 2	2456106	Yes	Yes	SG300-ARL-2012-R7-02	SG300-methacrylate	0.00114	0.000102	0.000127	

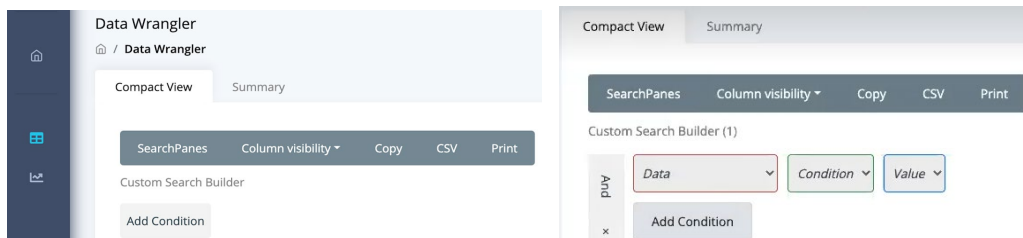
- To download a CSV file showing only the filtered data, click **CSV** in the top left menu.
- To copy the data, click **Copy** in the top left menu.
- To print the data, click **Print** in the top left menu.

4.2.2 Advanced Data Queries: Conditional Filtering

The Adhesive Dashboard has additional querying options for more advanced users.

To query the data using the advanced querying features:

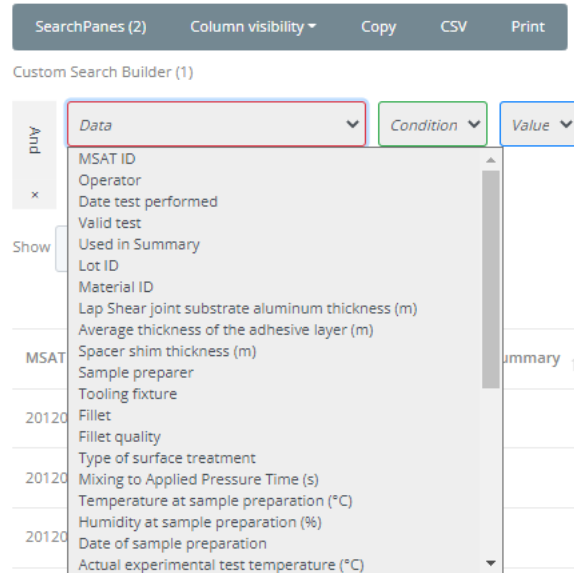
- Click **Add Condition** in the top left of the main view (image below, left). The querying options will appear (image below, right):



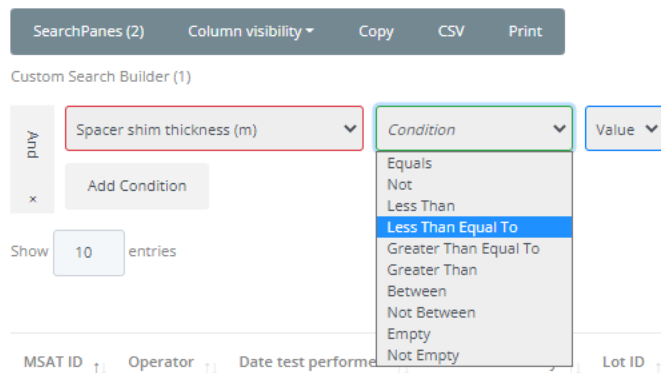
The following customizable options are available:

- Data:** Select the type of data to adjust the parameters.
- Condition:** Select the condition (Equals, Not, Less Than, Less Than Equal To, Greater Than Equal To, Greater Than, Between, Not Between, Empty, Not Empty).

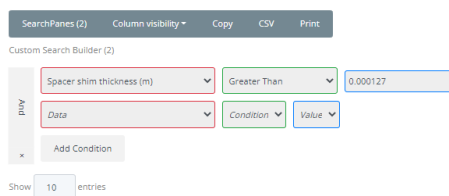
- **Value:** Type in a numerical value.
- 2) To select a variable (the column headers in the CSV file being currently viewed), click **Data** and select from the dropdown menu:



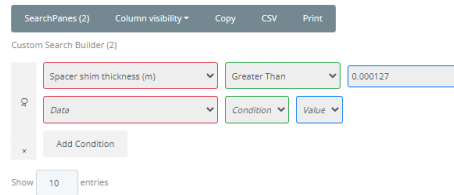
- 3) To select a condition, click **Condition** and select from the dropdown menu:



- 4) After selecting a Condition, the Value field will present itself. Type in a value. For example, this user is looking for a Spacer shim thickness greater than 0.000127 m:



- 5) Click on the vertical bar to change from “And” to “Or”:



6) To add a condition click on the < on the right:



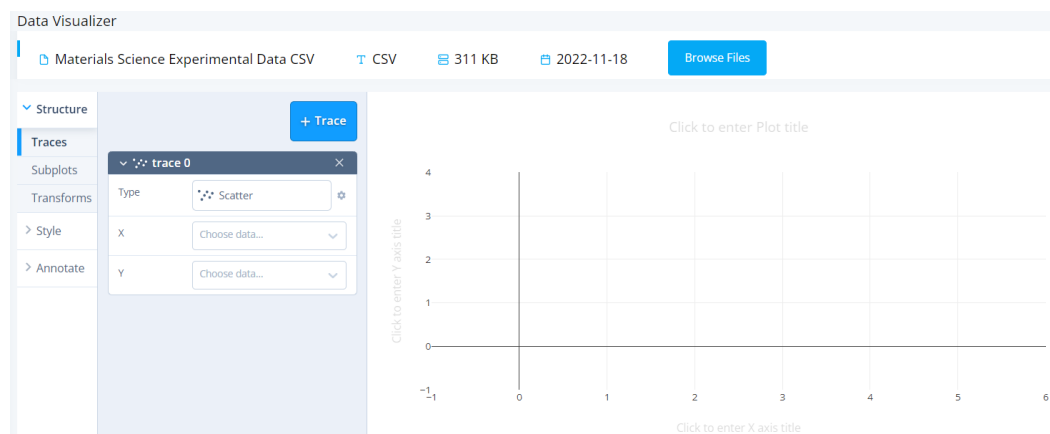
7) Continue until you have your desired query. The data and columns in the Column visibility screen will change as the query is built to reflect results.

4.3 Visualizing Data

The Data Visualizer enables the user to create different visualizations to display patterns and trends in their data. The user can build meaningful representations for their particular dataset or use case and easily export them to Portable Network Graphic (PNG) files to communicate results.

After integrating a dataset with the Data Visualizer (the user must click on the Data Visualizer icon next to the desired data file, see Section 4.1), the user is brought to the Data Visualizer screen.

For the following examples, the user uploaded a CSV file, *Materials Science Experimental Data CSV*.



The name of the file being viewed is displayed at the top (in the image above, the file name is *Materials Science Experimental Data CSV*), with the file information, including file type (CSV), size (311 KB), and upload date (2022-11-18). To return

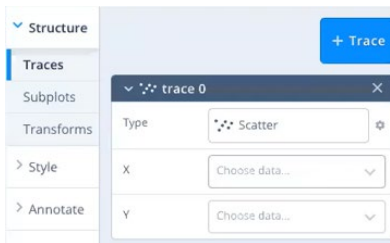
to the File Manager and view a different file, click **Browse Files** (or File Manager in the Navigation Pane).

The Data Visualizer Tool includes three main menus:

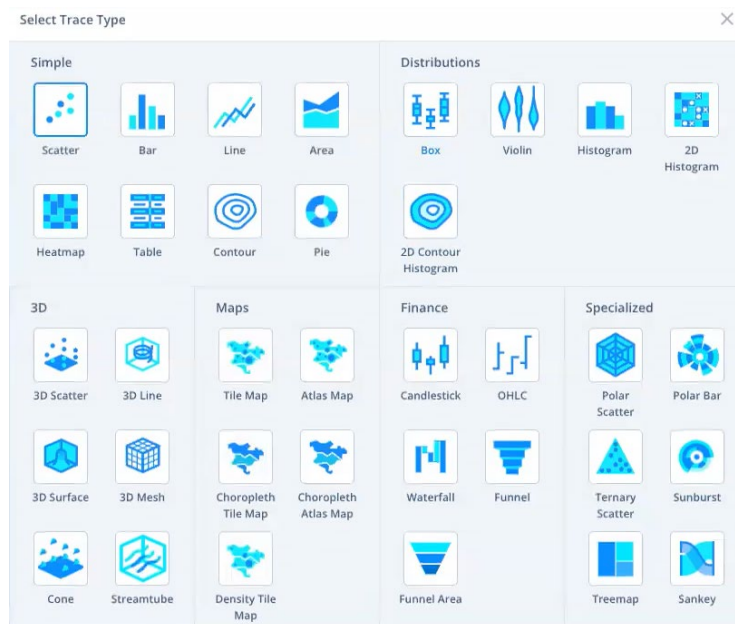
- **Structure:** Select the type(s) of visualizations (Traces) to represent data, create subplots or overlays to compare data, and print or copy the final visualization.
- **Style:** Fine tune the look of a trace, such as changing the fonts and colors.
- **Annotate:** Add annotations to further describe data shown in the traces.

To select a visualization type (Trace), assign axes, and compare plots:

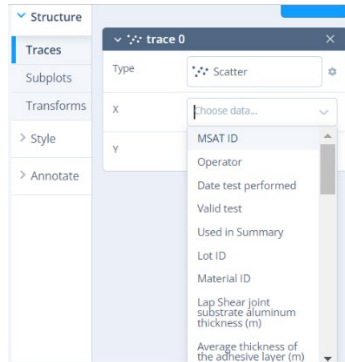
- 1) Expand the **Structure** menu on the left side.



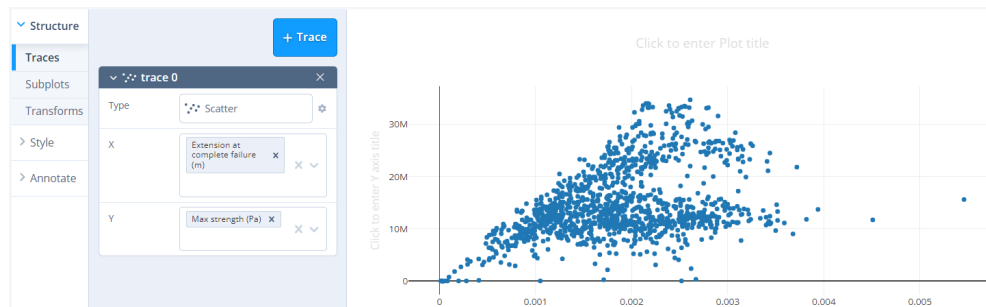
- 2) Click on **Traces** to display the options to configure the first trace (trace 0; note, this is the default view).
- 3) Click on the **Type** field to select the Trace (note, Scatter is the default Trace). The Trace Type window will appear:



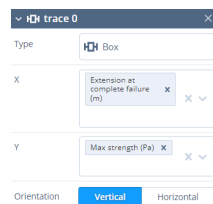
- 4) Click on the desired **Trace Type**. The new Trace name will appear in the Type field.
- 5) Use the **X dropdown menu** to select the independent variable (X-axis). The column headers in the viewed data file will be available.



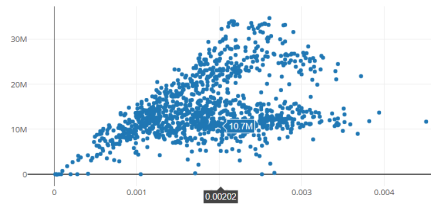
- 6) Use the **Y dropdown menu** to select the dependent variable (Y-axis). The example below shows a scatterplot of Max Strength versus Extension at Complete Failure.



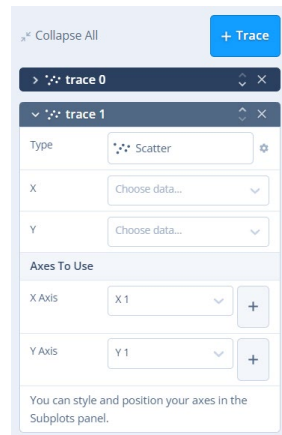
Additional options might appear depending on the selected Trace Type. For example, orientation can be changed for Box Plots:



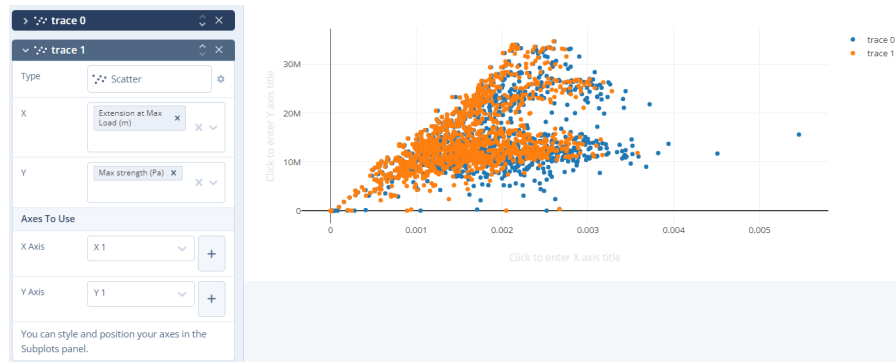
- 7) Hover the cursor over data points to see relevant statistics. For example, for the following scatterplot, the point (0.00202 mm, 10.7 MPa) is shown. Note that the X-axis is:



- 8) To add another set of data for comparison purposes, click **+ Trace**. A trace 1 pane will appear (expand and collapse the traces using the arrows):



- 9) Repeat the previous steps to select the Trace and data for the X and Y axes. The example below shows Max Strength versus Extension at Complete Failure (trace 0, blue) with an overlay showing Max Strength versus Extension at Max Load (trace 1, orange):



To compare plots next to each other using a separate X-axis and Y-axis, in the “Axes to Use” panel for trace 1, click the **+** next to the X-axis and Y-axis dropdown menus. An X2 and Y2 are now available.

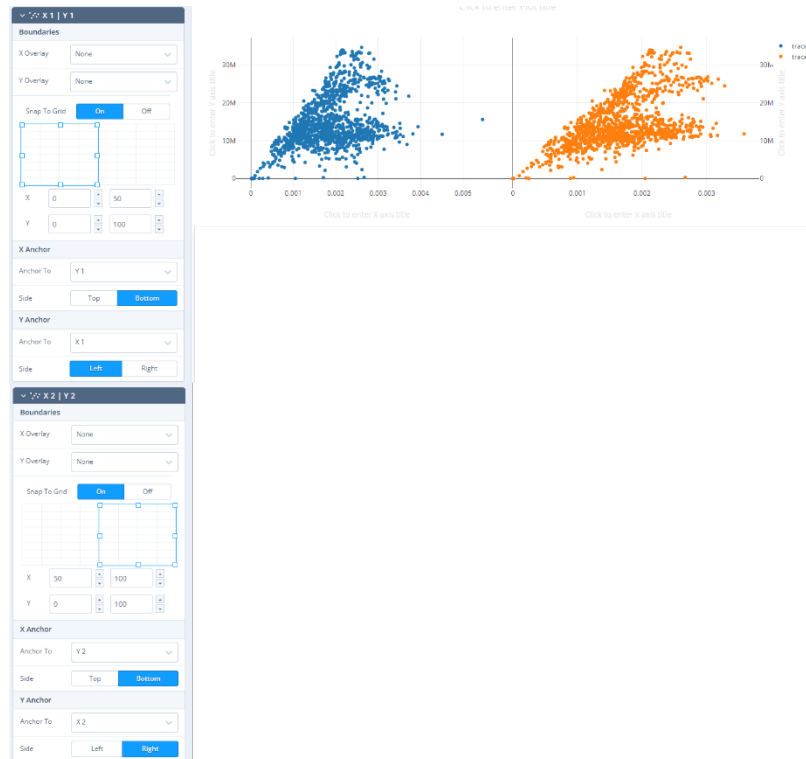
To separate the plots and customize, Click **Subplots** in the left menu bar within the Structure menu.

Interactive grids, which are used to style and position the axes of each subplot, will appear for each Trace.

To remove the overlays, select the X Overlay and Y Overlay dropdown menus and select None.

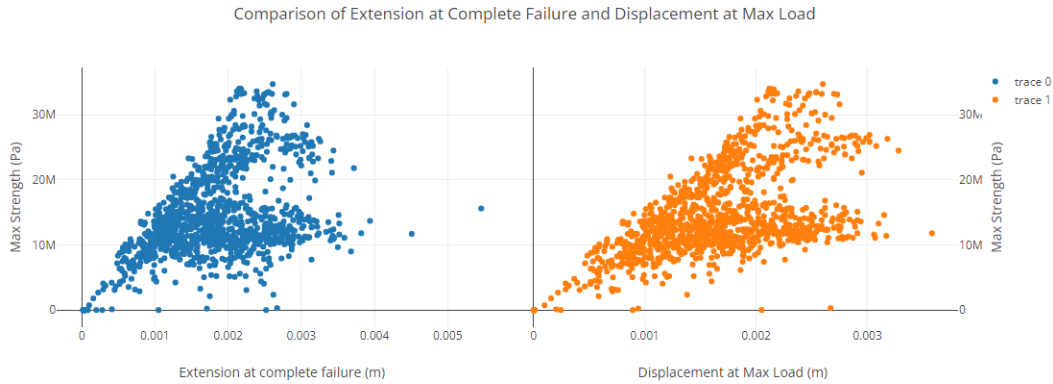
X Overlay	None
Y Overlay	None

- 10) To resize the subplot axes, move the grid lines back and forth or use the up and down arrows next to the x and y. Click **On** next to Snap to Grid to line the border up with a vertical line. Click **Off** to move the borders fluidly.



- 11) To change the location of the X and Y-axes, use the Top and Bottom buttons next to the respective axis.

- 12) To edit the axes titles or plot title, click on the gray text and type the name of the axis. Press Enter on the keyboard when done editing the axis name.



8) To download a plot as a PNG file, click on the camera in the upper right:

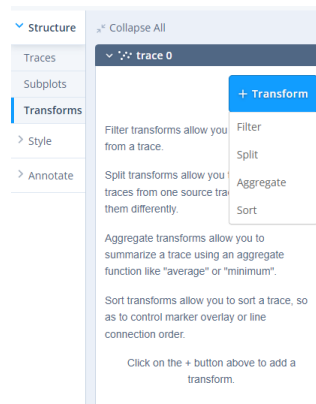


To Transform a plot or subplot:

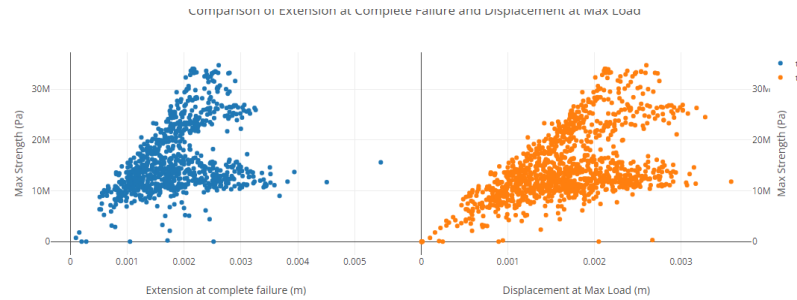
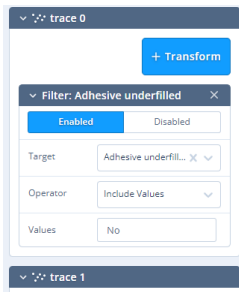
1) To access Transforms, click on **Transforms** in the Structure submenu. Transforms allow the user to do the following:

- a. Filter data out from a trace with **Filter transforms**.
- b. Create multiple traces from one source trace to style them differently with **Split transforms**.
- c. Summarize a trace using an aggregate function, such as average or minimum, with **Aggregate transforms**.
- d. Sort a trace with **Sort transforms**.

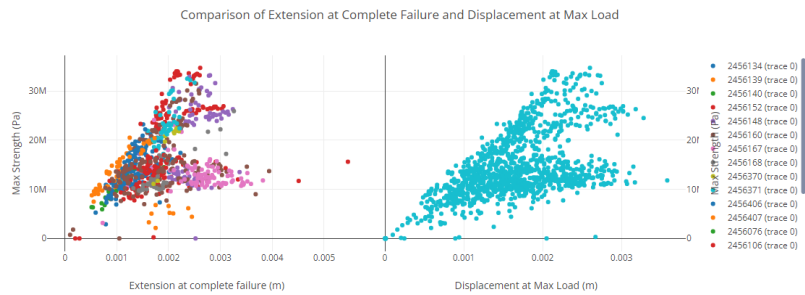
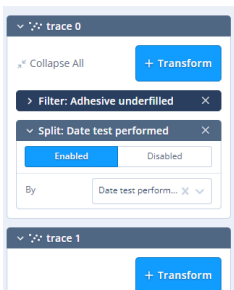
2) Click on + Transform to access the Transform options.



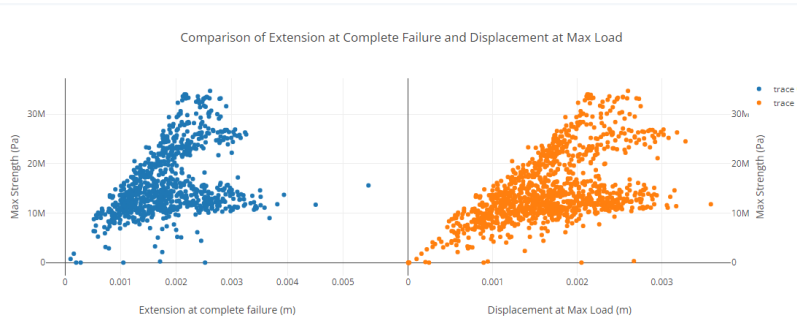
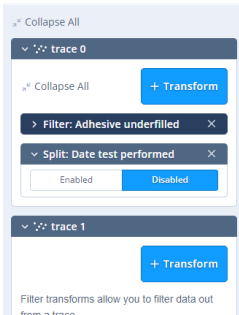
3) To Filter data out of a trace, select **Filter**. For example, the user has filtered out any datapoints that have the value “No” for the Adhesive underfilled variable for trace 0.



- 4) To split the data, click **+ Transform** and select **Split**. For example, the user selected to show the Date test performed variable:



- 5) To aggregate the data, click **+ Transform** and select **Aggregate**.
- 6) To sort the data, click **+ Transform** and select **Sort**.
- 7) To clear out a transform, click **Disabled** in that transform. The user has now disabled the Date test performed split transform:

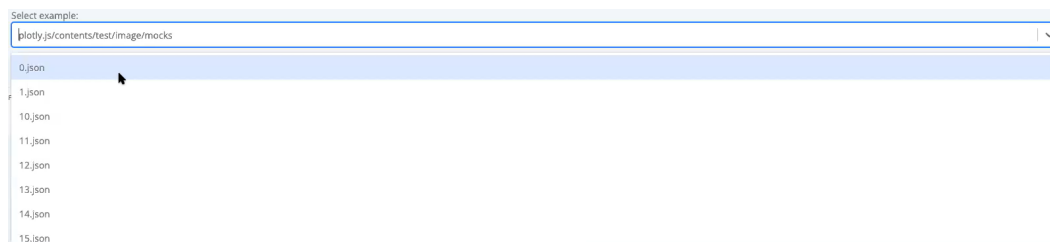


To style or annotate plots or subplots:

- 1) To style created traces, select **Style** from the side menu. The following options are available:
 - a. **General:** Adjust the background colors in the plots; adjust the texts and fonts around the plots.

- b. **Traces:** Adjust the fonts, styles, sizes, and other aspects of points and lines used in the traces; change the information provided when hovering; select to add error bars.
 - c. **Axes:** Style and edit the axes font and colors; adjust the axes ranges and types.
 - d. **Legend:** Edit and style the legend.
- 2) To annotate the traces, select Annotate from the side menu. The following options are available:
- a. **Text:** Add text annotations to draw attention to and describe areas of interest.
 - b. **Shapes:** Add lines, rectangles and ellipses to highlight time periods, thresholds, or other areas of interest.
 - c. **Images:** Incorporate logos, watermarks, or other images to make the data more readable or to brand content.

To see examples of what different plots could look like using the Data Visualizer tool, use the dropdown menu and select any JSON file:



5. Formatting and Uploading Adhesive Experimental Data

The Adhesive Dashboard provides an easy-to-use Excel template, called the Experiment Checklist, for the user to enter all data from experiments. The user can then easily upload the Checklist to the Adhesive Dashboard to view, analyze, and share results.

5.1 Download Experiment Checklist in Digitized Records

The Experiment Checklist is an Excel (XLS) file that provides fields for the scientist to add the experimental results from adhesives testing. The Checklist is based on ARL's MIL-PRF-32662 Performance Specifications for testing high-loading rate adhesives typically used for structural and armor applications. The

scientist can easily transfer their physical notes and records to this Checklist, and then upload it to the Adhesive Dashboard for further evaluation.

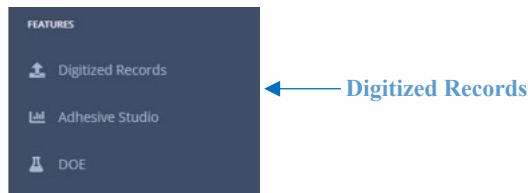
During the adhesive testing, the scientist is assumed to collect the following information:

- Images of the failed surfaces
- Excel file with the displacement (mm), load (N), and time (s) data from the tensile tester
- Physical (handwritten) notes with experiment metadata and observations, based on the template provided in MIL-PRF-32662 Performance Specifications

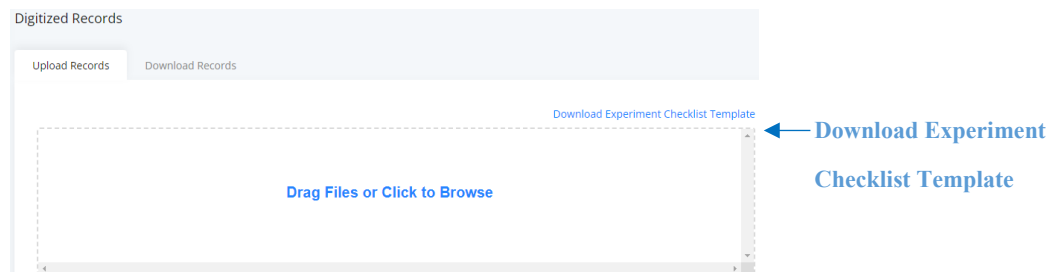
The Experimental Checklist provides a fillable template for the researcher to easily digitize their physical notes and combine with an Excel file to upload into the Adhesive Dashboard.

To download the Experiment Checklist:

- 1) Click on **Digitized Records** in the Features section of the Navigation pane in the main dashboard view.



- 2) Click on the Upload Records tab.
- 3) To download the Experiment Checklist, click **Download Experiment Checklist Template** in the top right.



5.2 Use Experiment Checklist Template

To use the Experiment Checklist:

- 1) **Save** the downloaded Checklist to a desired location.

2) **Open** the Checklist and click on the **Table of Contents** tab.

Table of Contents		
Single-Lap-Joint Fabrication Checklist		
Single-Lap-Joint Bondline Thickness Measurements		
Single-Lap-Joint Mechanical Testing Check List		
Single-Lap-Joint Load versus Displacement Data		
Crack Extension Test Sample Fabrication Check List		
Crack Extension Bondline Thickness Measurements		
Crack Extension Test Environmental Conditioning Checklist		
Crack Extension Test Crack Length Measurements		
MIL-PRF-32662, Performance Specification, "Adhesive, High Loading Rate, for Structural and Armor Applications"		
Spreadsheet companion to travel sheets provided in MIL-PRF-32662		
This work was supported in part by Cooperative Agreement W911NF-19-2-0112 between DEVCOM ARL and Worcester Polytechnic Institute.		
► Table of Contents SU Fabrication SU Thickness SU Strength SU Machi ... (+) ◀		

Each element in the Table of Contents links to a different spreadsheet with information to complete for each adhesive experiment. Most spreadsheets (with the exception of the Load versus Displacement Data) align with appendices in the MIL-PRF-32662 Performance Specifications, as described in Table 2.

Table 2 Experiment checklist data

Experiment checklist section	Details
Single-Lap-Joint Fabrication Checklist	MIL-PRF-32662 Performance Specifications Appendix D, single-lap-joint fabrication check list
Single-Lap-Joint Bondline Thickness Measurements	MIL-PRF-32662 Performance Specifications Appendix F, single-lap-joint bondline thickness measurements
Single-Lap-Joint Mechanical Testing Checklist	MIL-PRF-32662 Performance Specifications Appendix E, single-lap-joint mechanical testing check list
Single-Lap-Joint Load versus Displacement Data	Data extracted from the tensile tester
Crack Extension Test Sample Fabrication Check List	MIL-PRF-32662 Performance Specifications Appendix G, crack extension test sample fabrication check list
Crack Extension Bondline Thickness Measurements	MIL-PRF-32662 Performance Specifications Appendix I, crack extension test bondline thickness measurements
Crack Extension Test Environmental Conditioning Checklist	MIL-PRF-32662 Performance Specifications Appendix H, crack extension test environmental conditioning check list
Crack Extension Crack Length Measurements	MIL-PRF-32662 Performance Specifications Appendix J, crack extension test crack length measurements

The order of the tabs aligns with the typical experiment workflow. Two types of experiments are included: shear strength testing of single-lap-joint (SLJ) adhesively bonded metal specimens by tensile loading (metal to metal); and crack extension testing using double cantilever beam (DCB) method.

- 3) To begin using the Checklist, navigate to the experiment by clicking on the experiment name in the **Table of Contents** or using the tabs at the bottom of the spreadsheet. The following example shows a portion of the single-lap-joint fabrication check list spreadsheet:

Tier I: RT testing, dry conditioning		Tier II: RT testing, hot/wet conditioning		Tier II - Part 2: 71 °C testing, dry conditioning	
Sample ID		Sample ID		Sample ID	
Sample ID		Sample ID		Sample ID	
Sample ID		Sample ID		Sample ID	
Sample ID		Sample ID		Sample ID	
Sample ID		Sample ID		Sample ID	
Date		Date		Date	
Temperature		Temperature		Temperature	
Humidity	%	Humidity	%	Humidity	%
Testing facility					
Testing Operator					
Adhesive					
Adhesive Type					
Adhesive Form					
Lot ID					
code					
Expiration date					

- 4) To input data, click on a field and type in the appropriate value or text. When data is entered, the field name turns green. If no data is entered, it remains orange. When a spreadsheet is complete, the top row will turn green as well as the corresponding line in the table of contents.

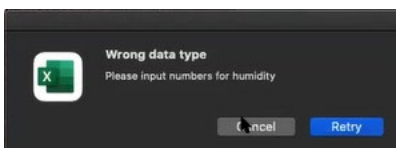
For example, some fields below are complete and some are not:

Tier I: RT testing, dry conditioning		Tier II: RT testing, hot/wet conditioning		Tier II - Part 2: 71 °C testing, dry conditioning	
Sample ID	12345	Sample ID		Sample ID	
Sample ID		Sample ID		Sample ID	
Sample ID		Sample ID		Sample ID	
Sample ID	123	Sample ID		Sample ID	
Sample ID		Sample ID		Sample ID	

An example of a completed spreadsheet:

Tier I: RT testing, dry conditioning		Tier II: RT testing, hot/wet conditioning		Tier II - Part 2: 71 °C testing, dry conditioning	
Sample ID	20170763	Sample ID	20170773	Sample ID	20170778
Sample ID	20170764	Sample ID	20170774	Sample ID	20170779
Sample ID	20170765	Sample ID	20170775	Sample ID	20170780
Sample ID	20170766	Sample ID	20170776	Sample ID	20170781
Sample ID	20170767	Sample ID	20170777	Sample ID	20170782
Date	9/20/2017	Date	9/20/2017	Date	9/20/2017
Temperature	23.8 °C	Temperature	23.8 °C	Temperature	23.8 °C
Humidity	53.2 %	Humidity	53.2 %	Humidity	53.2 %
Testing facility					
US. Army Research Laboratory, Weapons and Materials Research Directorate, Aberdeen Proving Ground, MD 21005					
Testing Operator					
ARL Technician #1					
Adhesive					
PPG MMP047					
Adhesive Type					
Epoxy					
Adhesive Form					
Paste - 1 component					
Lot ID					
Manufacturer's date 6/7/2017					

Some fields can only take numerical or text data. If the incorrect data type is used, an error message will appear:

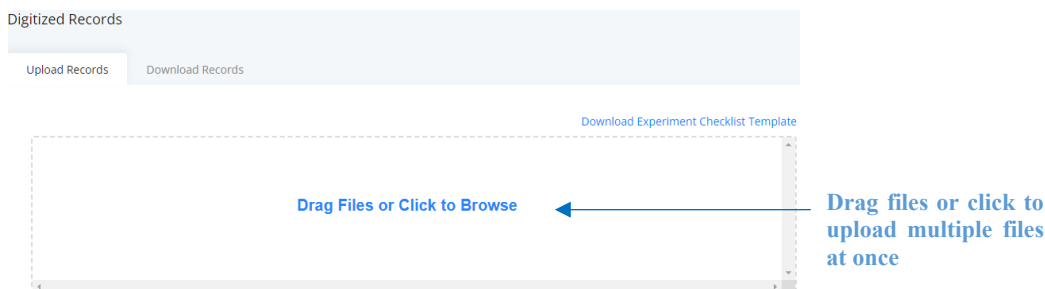


5.3 Upload Experiment Checklist Template in Digitized Records or File Manager

The completed Experiment Checklist can be uploaded in the Digitized Records feature. The Experiment Checklist is run through a parser, which detects if errors (e.g., blank fields, missing spreadsheets) are present.

To upload the completed Experiment Checklist in Digitized Records:

- 1) In the Digitized Records feature, either drag the files into the box, or click inside the box to upload a file or multiple files:



When files are uploaded, messages will appear describing whether the sheet contains missing values, missing spreadsheets, or if it is complete. In the following example, the first file has all spreadsheets, but data is missing from 457 fields, the second file is missing two of the spreadsheets, the third file is complete, and the fourth file has data missing from 10 fields and 2 missing spreadsheets:

2.MIL-PRF-32662 travel spreadsheet-Template(Errors).xlsx	Errors (457)
3.MIL-PRF-32662 travel spreadsheet-MissingSheets.xlsx	Missing Sheets (2)
1.MIL-PRF-32662 travel spreadsheet-Filled.xlsx	Valid
4.MIL-PRF-32662 travel spreadsheet-MissingSheet&Error.xlsx	Errors (10) Missing Sheets (2)

Messages include the following:

- Values are missing from fields with the total number of missing values:
 - ▲ Errors (457) (this message shows 457 fields are blank)
 - If the file is missing fields, it is uploaded to disk for access by the specific user who uploaded.

- Entire spreadsheets are missing:  (this message shows two spreadsheets are missing)
 - If the file is missing sheets it is uploaded to disk for access by the specific user who uploaded.
 - The spreadsheet is complete: 
 - If the file is valid, it is uploaded to the database for shared use.
- 2) To view additional information about the errors, click on the arrow to expand. The Summary tab provides a list of the number of errors for each spreadsheet.

The following example shows that two sheets are missing, but the rest are complete:

3.MIL-PRF-32662 travel spreadsheet-MissingSheets.xlsx

Summary

Sheet	Status
SLJ Fabrication	 Missing Sheet
DCB Fabrication	
DCB Thickness	 Missing Sheet
DCB Env Cond	
DCB Crack Len	

The following example has no missing spreadsheets, but each spreadsheet contains missing fields:

2.MIL-PRF-32662 travel spreadsheet-Template(Errors).xlsx

Summary SLJ Fabrication (9) SLJ Thickness (215) SLJ Strength (98) DCB Fabrication (18) DCB Thickness (55) DCB Env Cond (18) DCB Crack Len (44)

Sheet	Status
SLJ Fabrication	 9
SLJ Thickness	 215
SLJ Strength	 98
DCB Fabrication	 18
DCB Thickness	 55
DCB Env Cond	 18
DCB Crack Len	 44

- 3) To view which cells for a particular spreadsheet were missing, click on the corresponding tab.

Summary	SLJ Fabrication (9)	SLJ Thickness (215)	SLJ Strength (98)	DCB Fabrication (18)
Cell	Error Type	Message		
A15	Missing Value	Please check the value with name facility		
A53	Missing Value	Please check the value with name oven_model		
B20	Missing Value	Please check the value with name operator		
B22	Missing Value	Please check the value with name adhesive		
B26	Missing Value	Please check the value with name lot_id		
B27	Missing Value	Please check the value with name manufacturer		
B28	Missing Value	Please check the value with name expiration_date		
B53	Missing Value	Please check the value with name oven_serial_number		
C53	Missing Value	Please check the value with name oven_cal_date		

- 4) The dashboard can support partial uploads, so even if the file is not complete, it will be uploaded to the dashboard. To view all spreadsheets and their error tags, click on the **Download Records** tab.

Digitized Records

Upload RecordsDownload Records

Show10entries

Search:

Name	Size	Uploaded Date	Tags	Actions
3.MIL-PRF-32662 travel spreadsheet-MissingSheets.xls	198 KB	2022-01-20	Missing Sheets (2)	Download View Trash
1.MIL-PRF-32662 travel spreadsheet-Filled.xls	225 KB	2022-01-20	Valid	Download View Trash
4.MIL-PRF-32662 travel spreadsheet-MissingSheet&Error.xls	208 KB	2022-01-20	Errors (10)Missing Sheets (2)	Download View Trash
2.MIL-PRF-32662 travel spreadsheet-Template(Errors).xls	199 KB	2022-01-20	Errors (457)	Download View Trash

Showing 1 to 4 of 4 entries

Previous1Next

- 5) To download a spreadsheet to view and fix errors, click on the **download icon** (📄).
- 6) To view the errors, as shown in Steps 1 through 3, click on the **view icon** (🔍).
- 7) To permanently delete the file from the dashboard, click on the **trash icon** (🗑).

6. Sharing, Viewing, and Comparing Adhesives with Dashboard Features

In the Adhesive Studio, users can view summaries of test results for all adhesives uploaded into the dashboard. The user can search using the name of an adhesive material or by desired properties or test conditions to find a material that suits their individual needs. Vendors can also use this tool to showcase their materials to a wide audience. Figure 4 shows the main view of the Adhesive Studio.

Note, this feature is current under development. It does not yet link to the Digitized Records. The data currently shown in the dashboard is for demonstration purposes.

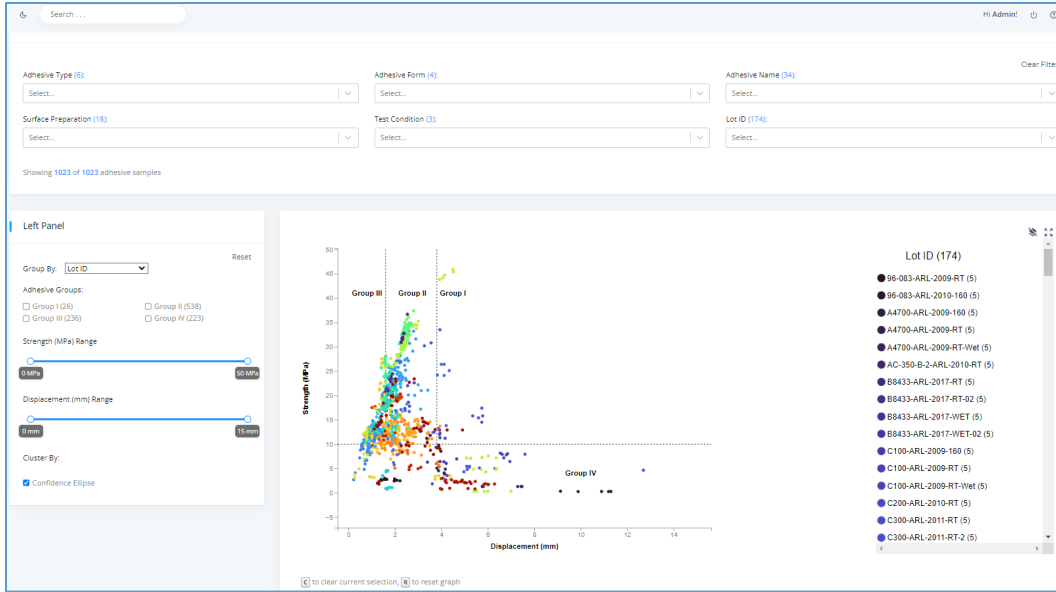


Fig. 4 Adhesive Studio main view

The Adhesive Studio includes the following sections and functionality:

- In the **Top Panel**, the user can select different experimental conditions to change what is displayed in the Strength versus Displacement graph.
- In the bottom **Left Panel**, the user can adjust the graph settings.
- In the bottom **Right Panel**, a graph summarizing the grouping of adhesives. The grouping of the adhesive materials makes it easy to visually understand the relationship between strength and displacement properties for adhesives, and to easily find an adhesive that meets certain strength and displacement properties. The groups are defined as follows in the MIL-PRF-32662:
 - Group I: Maximum strength ≥ 10 MPa and complete failure displacement limit > 3.81 mm
 - Group II: Maximum strength ≥ 10 MPa and complete failure displacement limit between 1.60 mm and 3.81 mm ($1.60 \text{ mm} \leq d \leq 3.81 \text{ mm}$)
 - Group III: Maximum strength ≥ 10 MPa and complete failure displacement < 1.60 mm
 - Group IV: Maximum strength < 10 MPa and any complete failure displacement

- The top **Search Bar** enables the user to search for specific adhesives to view their properties and experimental results.

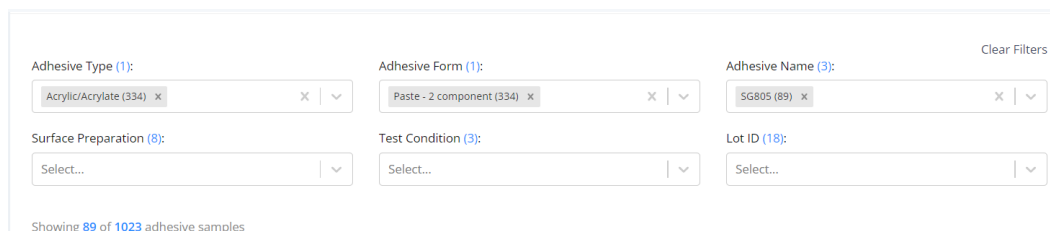
The following subsections describe how to use the Adhesive Studio.

6.1 Filter and Sort Summary of Adhesive Grouping

The graph in the bottom right panel shows a scatterplot of the maximum strength versus displacement at fracture for selected adhesives, experimental settings, or results, with the Grouping overlayed. You can adjust the graph settings and data shown using the top panel and the bottom left panel.

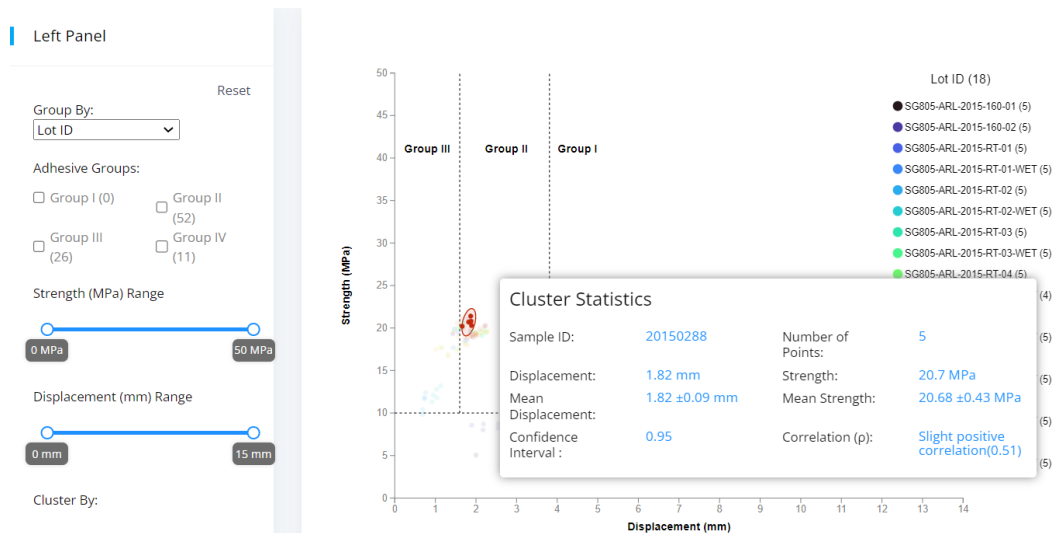
To sort and filter the scatterplot of the strength versus displacement graph grouping strength versus displacement graph:

- 1) To sort by experimental condition, click the **Group By** dropdown menu in the Left Panel and select an experimental condition. The default experimental condition is “Lot ID.” The graph shows a list of all the data shown with the color of the point in the graph.
- 2) To only show certain groups, check (or uncheck) the box next to the group number in the Left Panel.
- 3) To only show a certain range of Strength and/or Displacement, use the slider to adjust the minimum and maximum values in the Left Panel.
- 4) To zoom in and out of the graph, place your cursor in the graph and move your fingers towards or away from each other (if using a laptop mouse pad).
- 5) Click **Reset** in the left panel to clear all selections and return to the default view.
- 6) To filter based on the individual experimental conditions, use the top panel dropdown menus to select desired criteria. For example, the following properties were selected.



The screenshot shows the filter panel in Adhesive Studio. It contains six dropdown menus arranged in two rows of three. The top row includes 'Adhesive Type (1):' with 'Acrylic/Acrylate (334)' selected, 'Adhesive Form (1):' with 'Paste - 2 component (334)' selected, and 'Adhesive Name (3):' with 'SG805 (89)' selected. The bottom row includes 'Surface Preparation (8):' with 'Select...' selected, 'Test Condition (3):' with 'Select...' selected, and 'Lot ID (18):' with 'Select...' selected. A 'Clear Filters' link is located at the top right. At the bottom left, it says 'Showing 89 of 1023 adhesive samples'.

Selecting these experimental conditions resulted in the following scatterplot:



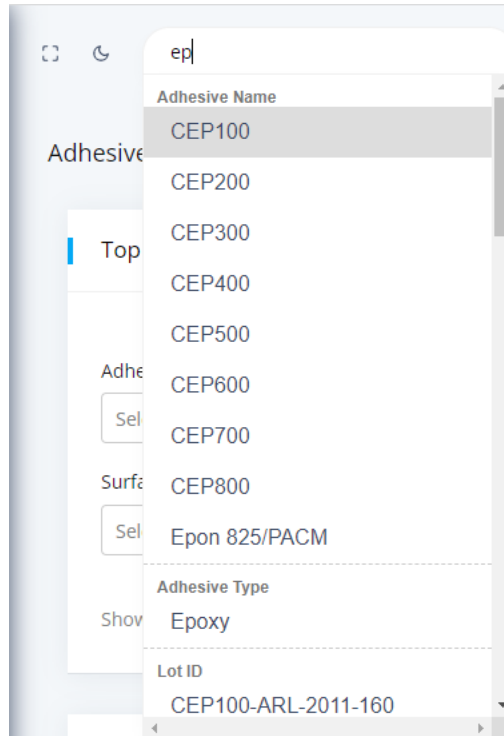
- 7) To view the statistics for a cluster of points (as shown above), hover the cursor over a point on the graph within that cluster. An ellipse will form around the group of points when you hover, showing the Cluster Statistics.

6.2 View Properties for Specific Adhesives

In the Adhesive Studio, you can search for a specific adhesive to view a summary of its experiment results.

To search for a specific adhesive:

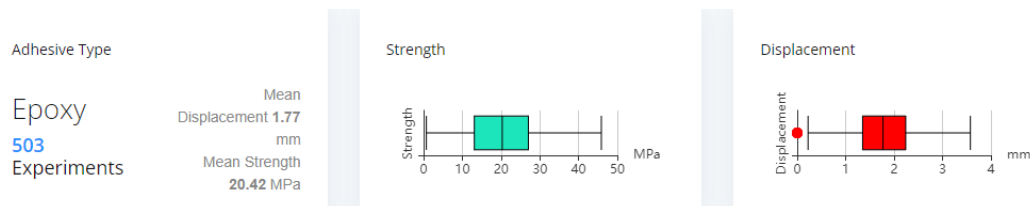
- 1) To search for a specific adhesive, enter the name into the search bar at the top of the Adhesive Studio main view. The Search bar will populate options based on the text you type, for example:



- 2) To select an adhesive, click on the name when it appears in the search bar list. Information will populate in the Adhesive Studio so you can easily view the properties and test result summaries. The following images describe the information provided in each panel after selecting an adhesive material.

Adhesive Type, Strength, and Displacement panels

The top panels show overall experimental statistics:



Experiment Coverage panel

Experiment Coverage

3	Adhesive Form	Paste - 2 component (81.51%)
16	Adhesive Name	CEP100 (41.15%)
14	Surface Preparation	GPS (57.85%)
3	Test Condition	Room Temperature (75.55%)
4	Group	Group II (58.05%)
88	Lot ID	CEP100-ARL-2012-RT-04 (1.99%)

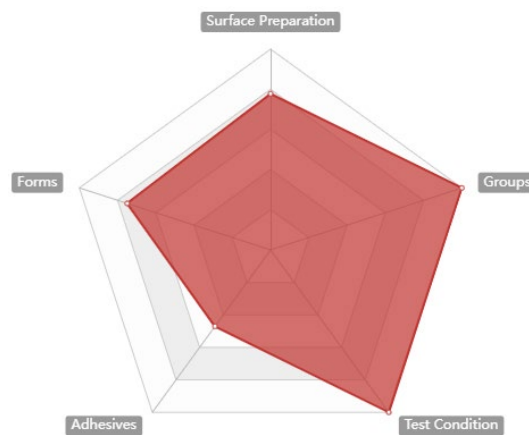
The Experiment Coverage panel shows the following information:

- **Column 1:** Number of unique experimental conditions (or group number) listed in column 2 for the selected adhesive.
- **Column 2:** The experimental condition or group.
- **Column 3:** The most common experimental condition type or group number with the percentage of experiments that type was used.

For example, the first row shows there were three different types of Adhesive Forms out of the 503 Epoxy experiments. “Paste -2 component” was the most commonly used form, accounting for 81.51% of the experiments.

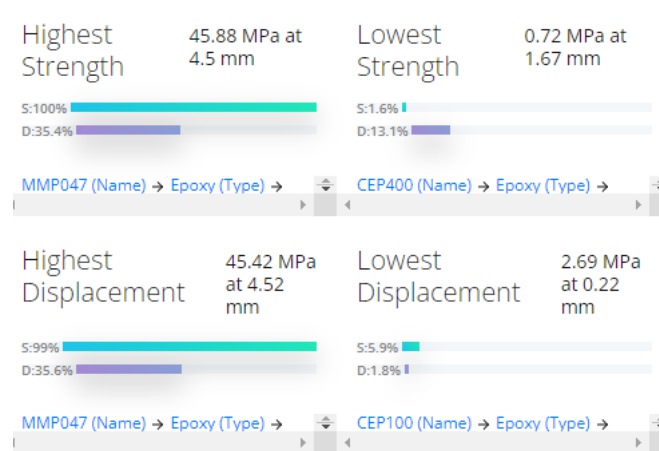
Spider Graph

To the right of the Experiment Coverage, a spider graph shows the distribution of experiment types. This chart helps the researcher determine which experiments they still need to perform:



Highest Strength, Lowest Strength, Highest Displacement, Lowest Displacement panels

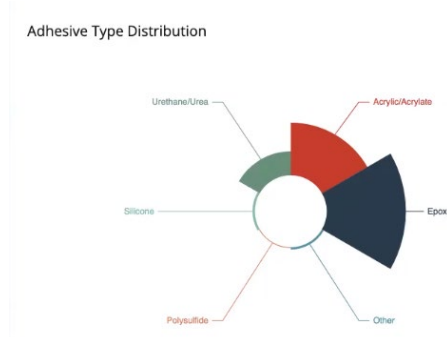
Below the Experiment Coverage panel, the highest and lowest Strength and Displacement values are shown, with the corresponding combination of experimental conditions (click the up and down arrows to show the entire list of experimental conditions):



The following values are provided:

- **Highest Strength:** The combination of experimental conditions for epoxy materials, which showed the highest strength during the tensile test (highest ultimate tensile strength, peak of the load vs. displacement strain curve), and the value of that strength in MPa and the corresponding displacement in mm.
- **Lowest Strength:** The combination of experimental conditions for epoxy materials, which showed the lowest strength during the tensile test (lowest ultimate tensile strength, peak of the stress/strain curve), and the value of that strength in MPa and the corresponding displacement in mm.
- **Highest Displacement:** The combination of experimental conditions for Epoxy materials, which showed highest displacement at fracture, and the value of that displacement in mm and the corresponding strength in MPa.
- **Lowest Displacement:** The combination of experimental conditions for Epoxy materials, which showed lowest displacement at fracture, and the value of that displacement in mm and the corresponding strength in MPa.

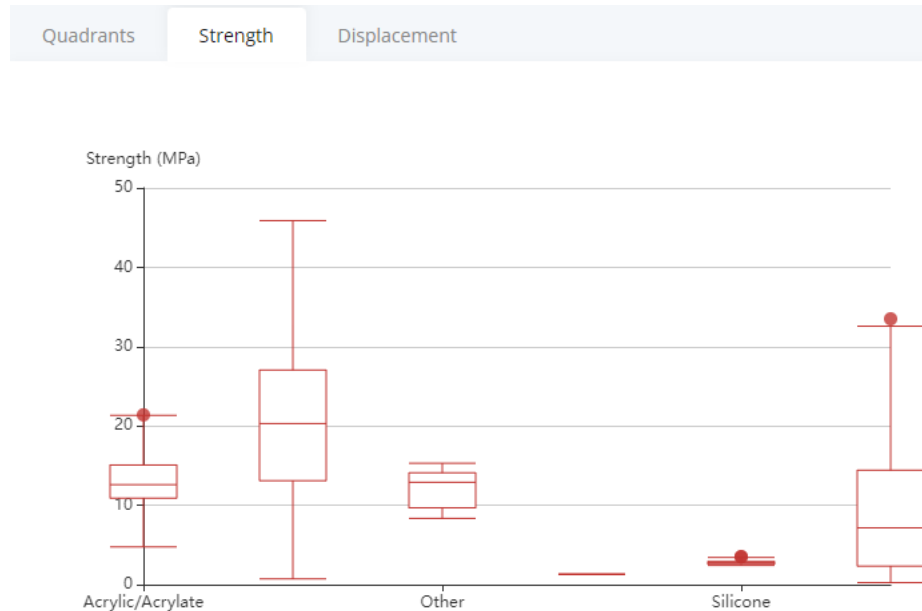
Adhesive Type Distribution panel



To show the type of adhesive and the percentage of experiments, hover over a segment.

Quadrants, Strength, and Displacement graphs panel

The graphs show a comparison of strength and displacement box plots for each adhesive type.



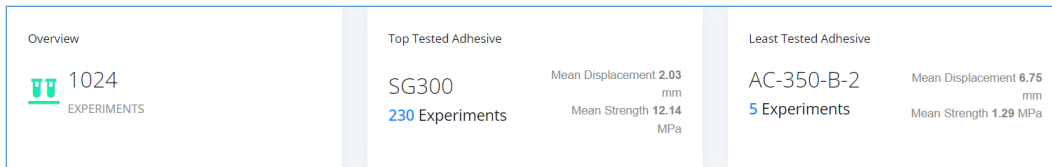
6.3 Viewing Results in the Dashboard View

A summary of the Experimental Results, which have been uploaded to the dashboard, can also be viewed in the dashboard view (homescreen). View the total number of experiments, the adhesives with the greatest and lowest maximum strength and displacement at fracture, experimental conditions, and many other facts and figures. Visualizations help the user understand the distribution of experimental settings and results. The card-based, componentized user interface makes it easy for the user to navigate, and for a developer to update the information shown in this view.

To upload data to be included in the dashboard view, see Section 4.

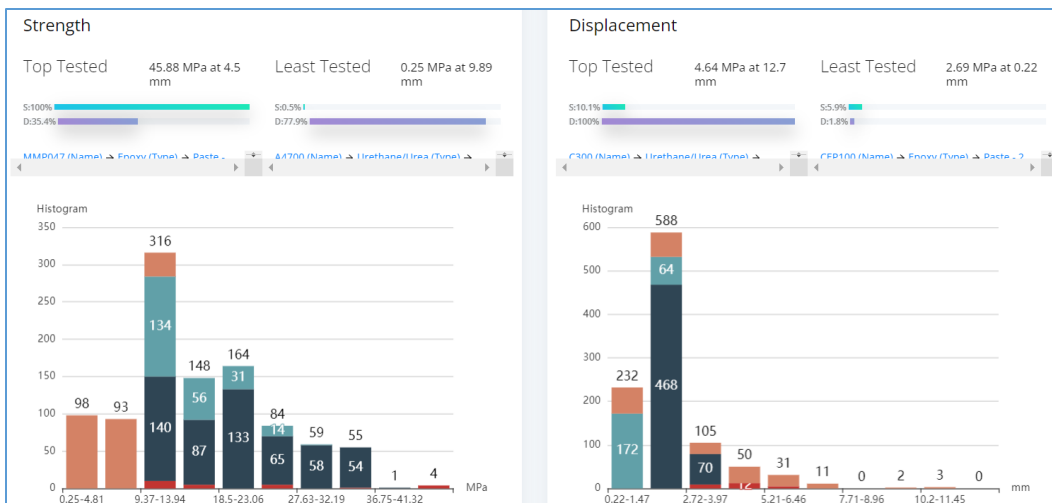
- 1) To view any of the panels or information shown below, simply scroll through the Adhesive Dashboard when in the dashboard view. The rest of this section describes the information provided in this view.

Overview, Top Tested Adhesive, Least Tested Adhesive Panels



- **Overview** (left): Total number of accessible experiments uploaded to the dashboard.
- **Top Tested Adhesive** (middle): The adhesive that is involved in the greatest number of experiments, the number of experiments performed using that adhesive, and the average displacement at failure and greatest strength for all experiments.
- **Least Tested Adhesive** (right): The adhesive that is involved in the least number of experiments, the number of experiments performed using that adhesive, and the average displacement at failure and greatest strength for all experiments.

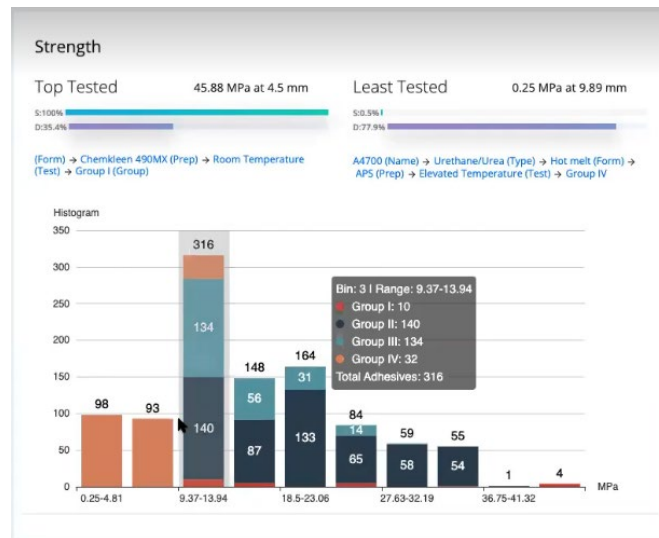
Strength and Displacement Panels



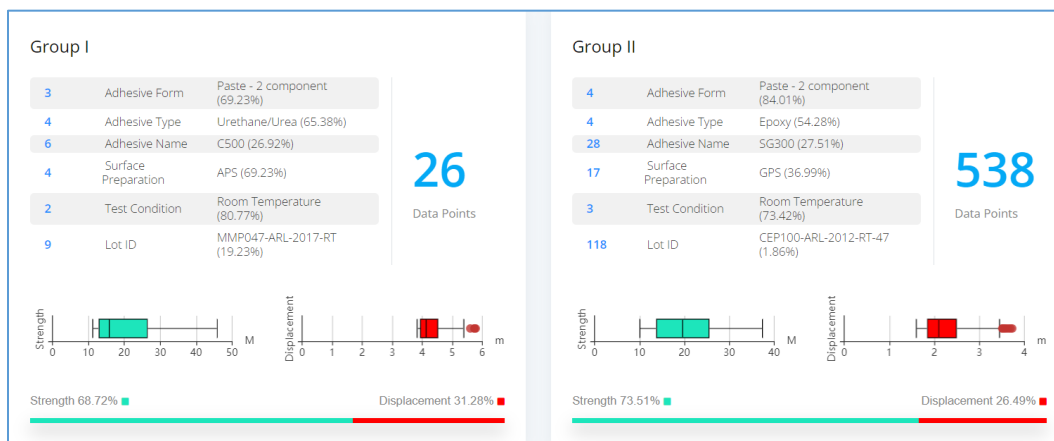
- **Strength** (left):
 - **Strength histogram:** Plot of frequency versus strength, showing how many experiments resulted in a specific range of peak strength values (MPa), separated by the groups with the total number of experiments in each group.
- **Displacement** (right):

- **Displacement histogram:** Plot of frequency versus displacement, showing how many experiments resulted in a specific range of displacement at fracture values, separated by the groups.

2) To see additional details, hover the mouse cursor over a bar:



Grouping Panels



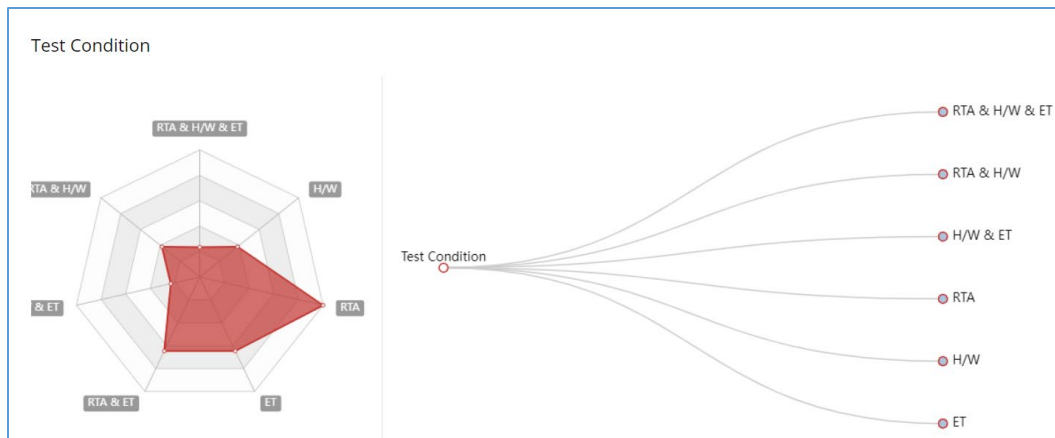
For each Group:

- **Data Points:** Number of experiments with a strength that falls into each group.
- **Table:**
 - **Column 1:** Number of unique experimental conditions (or group number) listed in column 2 for the selected adhesive. (For example, row 1 shows that there were three different types of Adhesive Forms used in all experiments uploaded to the Adhesive Dashboard with

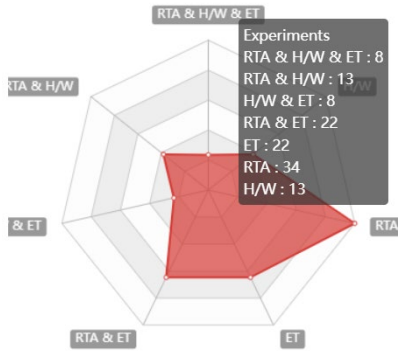
results that fall into Group 1. Section 4 describes the thresholds for each Group.)

- **Column 2:** The experimental condition or group.
- **Column 3:** The most common experimental condition type or group number with the percentage of experiments that the type was used. (For example, the Paste – 2 component was the most commonly used Adhesive Form, used in 69.23% of the experiments with results that fall into Group 1.)
- **For example, the first row shows that three experiments in Group 1 used the Adhesive Form**
- **Box Plots:** Box plots showing the range, quartiles, and other statistics.

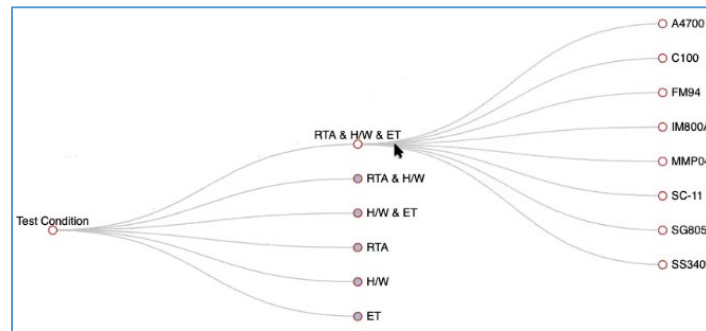
Test Condition Panel



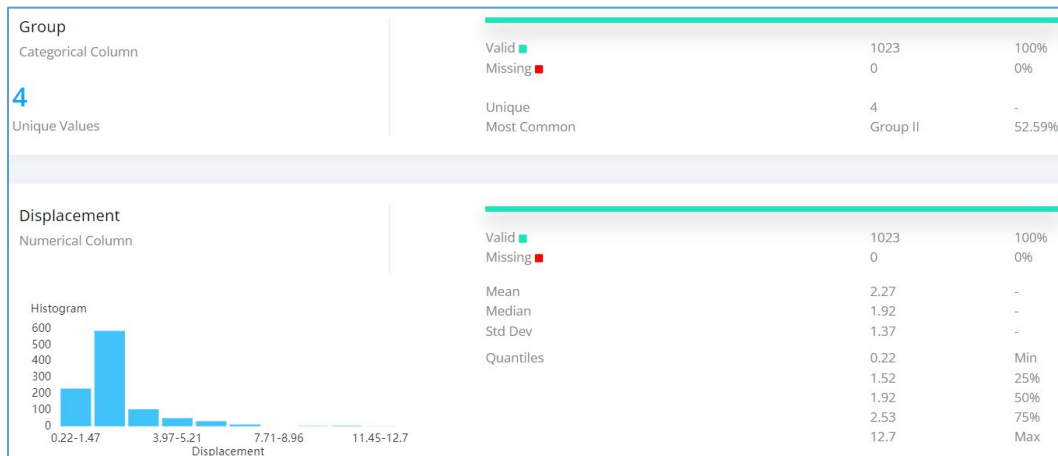
- **Spider Plot** (left): Visual comparison of the frequency of use for combinations of different test conditions.
 - **Test Condition Plot** (right). Separates out each test condition, which provides additional information by clicking.
- 3) To view specific information about the number of experiments that contained different combinations of test conditions, hover over the plot.



4) To expand the Test Condition plot, click on a circle.



Summary Statistics Panels



For each experimental condition shown in the columns, summary statistics are shown. In addition, the number of Valid and Missing data points are shown for numerical and categorical columns of variables.

7. Modeling the Data with Machine Learning

The Design of Experiment (DOE) feature uses ML algorithms to look for patterns in the data, such as relationships between certain variables, and suggest

experimental conditions and settings that can potentially achieve desired strength or displacement properties.

This feature is currently under development. It does not yet link to the Digitized Records. The data currently shown in the dashboard is for demonstration purposes.

Figure 5 shows the DOE default view.

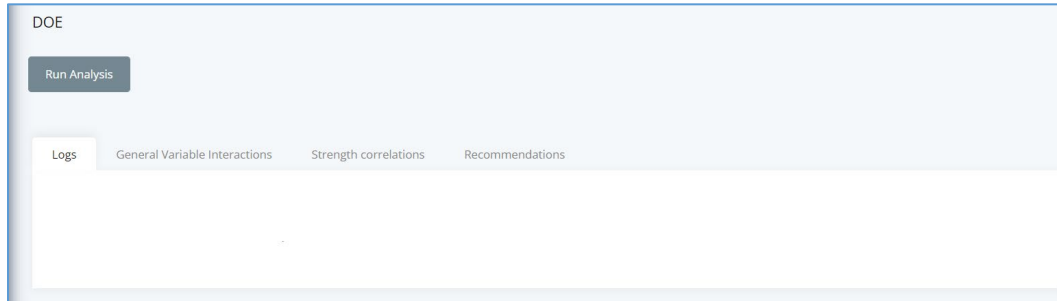
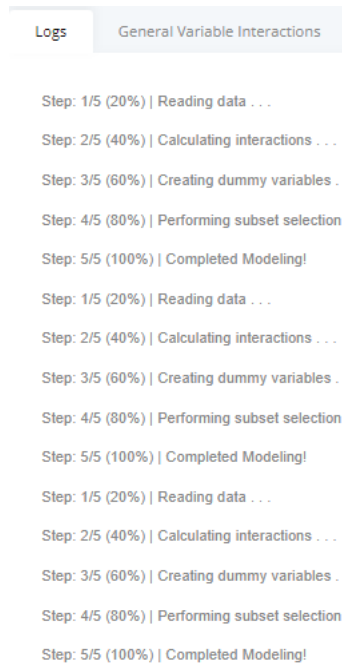


Fig. 5 DOE default view

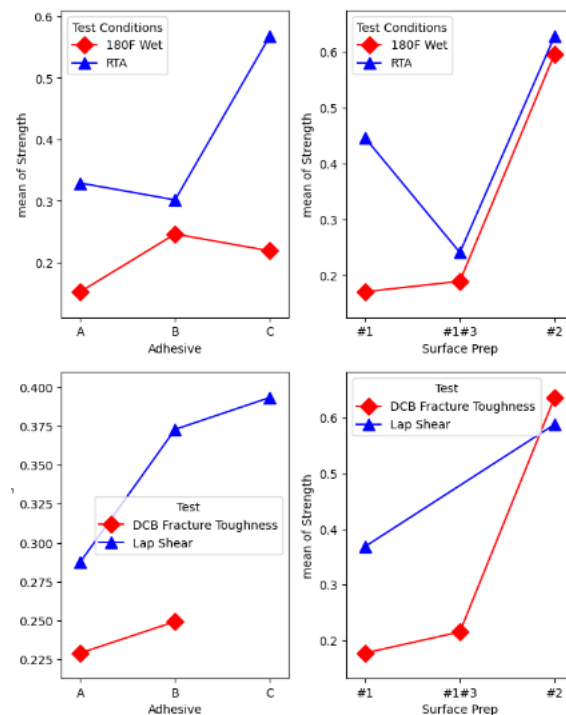
The following tabs are available:

- **Logs:** Status of the ML algorithm calculations.
- **General Variable Interactions:** Graphical depiction of the interactions between two variables, as determined by the ML algorithms. An interaction exists when a crossover exists between two lines.
- **Strength correlations:** Graphical depiction of which experimental conditions, settings, or results maximize adhesive strength, as determined by ML algorithms.
- **Recommendations:** Recommendations of experimental conditions to run to optimize adhesive properties (e.g., strength).

- 1) To execute ML algorithms, click on **Run Analysis**.
 - a. After a few minutes, the log will show the status of each step of the analysis (note, use the Step 5 completion percentage to determine if the analysis is complete):



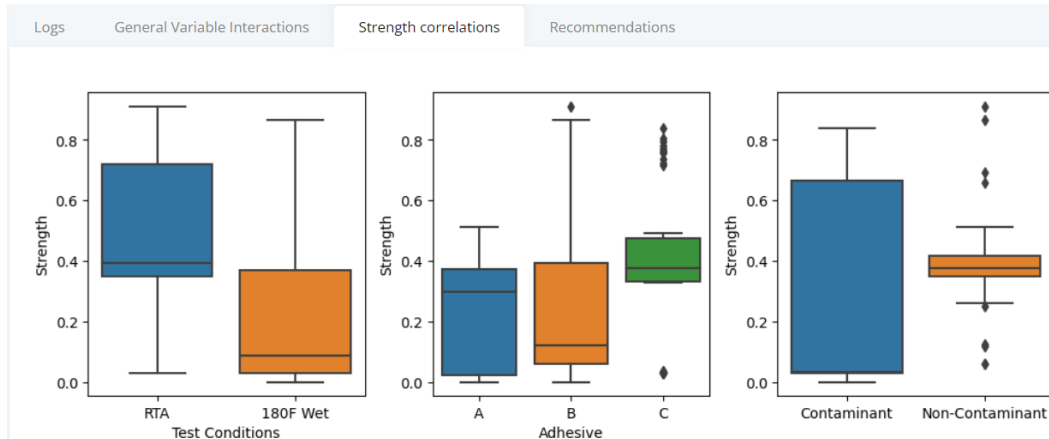
2) When the analysis is complete, click on General Variable Interactions to view analysis results. You will see a series of graphs, for example:



Each graph depicts how the mean of Strength varies for certain types of test conditions or results. If two lines cross, this implies there is an interaction between the variables. For example, in the bottom right graph, there is an interaction/

relationship between mean of Strength and Surface Prep for two types of tests shown.

- 3) To view how strength correlates with different experimental conditions, click on the **Strength correlations** tab:



- 4) To view the Recommended experiments, click the **Recommendations** tab:

Group #1	R2 Adj: 0.73 ↑	Strength: 90.22% ↑	MSE 0.02 ↓ BIC 1.27 ↓	AIC 1.05 ↓ CP 0.02 ↓	4 Main Effects 2 Interactions	Completed / Total: 0/4
Group #2	R2 Adj: 0.73 ↑	Strength: 91.4% ↑	MSE 0.02 ↓ BIC 1.31 ↓	AIC 1.06 ↓ CP 0.02 ↓	5 Main Effects 2 Interactions	Completed / Total: 0/4
Group #3	R2 Adj: 0.73 ↑	Strength: 84.95% ↑	MSE 0.02 ↓ BIC 1.24 ↓	AIC 1.04 ↓ CP 0.02 ↓	4 Main Effects 2 Interactions	Completed / Total: 0/2

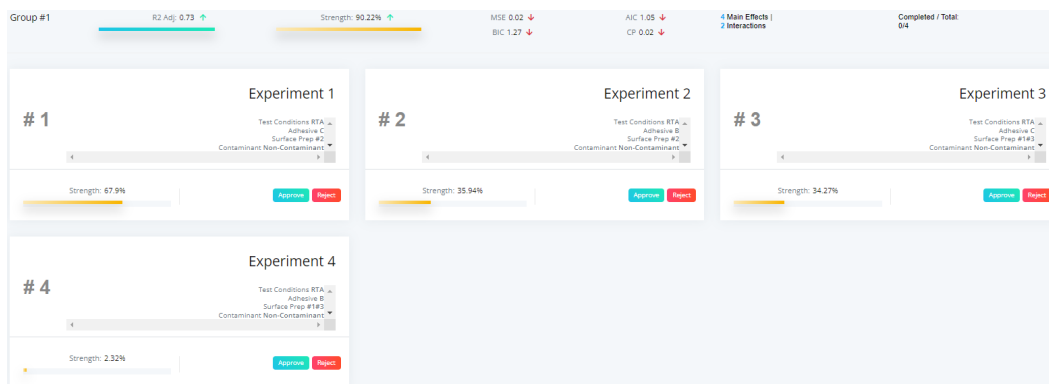
The different groups represent different models that were developed using ML techniques (e.g., forward selection, backward selection). Each group is unique to the number of variables and interactions included in the model. The information shown for each group shows the characteristics of the models, including:

- **R2 Adj:** R-Squared Adjusted (R2 Adj, or Coefficient of Determination) shows how well the data fit the model. The closer to |1| the better, as a value of 1 means that the algorithm used was a perfect fit for the data.
- **Strength:** Strength value calculated using the model.
- The following four statistical measures are used to evaluate the fit of the model. A lower value represents a better fitting model:
 - **MSE:** Mean Squared Error
 - **BIC:** Bayesian information criterion
 - **AIC:** Akaike information criterion
 - **CP:** Mallows's C_p

- **Main Effects:** The number of variables included in the model.
- **Interactions:** The number of interactions included in the model.
- **Completed/Total:** The completed or total number of recommended experiments.

The arrows indicate whether it is better for the corresponding value to be larger or smaller (e.g., R2 adjusted should be as close to 1 as possible; the MSE should be as close to 0 as possible).

- 5) To view suggestions for experiments that can reach desired strength results, click on the Group No. to expand each group:



This shows four different suggested experiments and the estimated strength that will result from executing the experiments.

- 6) To approve an experiment, click **Approve**.
- 7) To reject an experiment, click **Reject**.

8. Accessing Documentation

The Adhesive Dashboard provides easy access to the military performance specifications and standards and related information, the developer's guide, and the Experiment Checklist.

8.1 Military Performance Specifications and Standards

To access the military performance specifications and standards that were used as a guide to develop the Adhesive Dashboard:

- 1) Click **Documentation** at the bottom of the Navigation pane.

Background ^ Experimental Procedures   History and Rationale   MIL-PRF-32662  	FAQ ^ MIL-PRF-32662 FAQ's  
Experiment Checklist ^ MIL-PRF-32662 load versus displacement worksheet  MIL-PRF-32662 Travel Spreadsheet 	Drawings ^ DWG.NO.MIL-STD-3059(MR)-1 DXF  DWG.NO.MIL-STD-3059(MR)-1 PDF   DWG.NO.MIL-STD-3059(MR)-2 DXF  DWG.NO.MIL-STD-3059(MR)-2 PDF   DWG.NO.MIL-STD-3059(MR)-3 DXF  DWG.NO.MIL-STD-3059(MR)-3 PDF   DWG.NO.MIL-STD-3059(MR)-4 DXF  DWG.NO.MIL-STD-3059(MR)-4 PDF   DWG.NO.MIL-STD-3059(MR)-5 DXF  DWG.NO.MIL-STD-3059(MR)-5 PDF   DWG.NO.MIL-STD-3059(MR)-6 DXF  DWG.NO.MIL-STD-3059(MR)-6 PDF   DWG.NO.MIL-STD-3059(MR)-7 DXF  DWG.NO.MIL-STD-3059(MR)-7 PDF   DWG.NO.MIL-STD-3059(MR)-8 DXF  DWG.NO.MIL-STD-3059(MR)-8 PDF  
Travel Sheet ^ MIL-PRF-32662 rev A draft  	

- 2) To download a document, click the download icon ().
- 3) To view a pop-up version of the document, click the documentation icon ().

8.2 Developer Manual

- 1) To view a pop-up version of the Developer's Manual, click the documentation icon at the top right of the main menu.



- 2) When the manual appears, click the download icon at the top left to download the manual.
- 3) Platform documentation and the installation guide are provided in Appendices B and C, respectively.

9. References

1. MIL-PRF-32662. Adhesive, high-loading rate, for structural and armor applications. DEVCOM Army Research Laboratory (US); 2020. https://quicksearch.dla.mil/qsDocDetails.aspx?ident_number=283919.
2. Egyedi TM, Sherif C. Standards dynamics through an innovation lens: next-generation Ethernet networks. IEEE Communications Magazine: 2010;(48):166–171.

Appendix A. Design Thinking an End-to-End Context Aware Platform

This appendix appears in its original form, as an attached pdf to ARL-SR-0469, *MIL-PRF-32662 Context-Aware Agile Platform*, by Robert Jensen, Kimberly Brady, Prathyush Parvatharaju, Biao Yin, Fatemeh Emdad, and Elke Rundensteiner.

“Design Thinking an End-to-End Context Aware Platform” is the original proposal submitted by Prathyush Parvatharaju to US Army Combat Capabilities Development Command (DEVCOM) Army Research Laboratory (ARL) during the application process to the ARL Research Associateship Program (RAP). RAP is administered by Oak Ridge Associated Universities (ORAU) for ARL through Cooperative Agreement W911NF-16-2-0008.

DESIGN THINKING AN END-TO-END CONTEXT AWARE AGILE PLATFORM

PROPOSAL

Prathyush S Parvatharaju
Department of Data Science
Worcester Polytechnic University
Worcester, MA 01609
psparvatharaju@wpi.edu

June 10, 2020

ABSTRACT

Technology, an evolutionary process, is progressing faster than ever. It is the process of creating a new generation of powerful tools by adapting and improving on the previous set of innovations. User-experience driven visual aids coupled with a scalable back-end infrastructure has enabled one to efficiently store, retrieve, process, and extract insights from large, complex, and diverse sets of data. Recent advancements in the domain of machine intelligence have brought significant changes to the process and workflow of experimental design and analysis. Workflow automation is the key to shift focus from performing redundant tasks to extracting meaningful insights from the data. Automation is often specific and tends to be rigid. Context-awareness enables a system to sense and react based on their environment. Context-aware automation provides much-needed flexibility in an experimentation environment. Focus on user experience - The user's journey to solve a problem is critical, and is achieved by the process of design thinking. The work proposes an end-to-end agile platform addressing data retrieval and management, exploratory data analysis, modeling and serving, each with a focus on -

1. Context Awareness - The platform understands the data based on the context and provides necessary inputs to derive tangible insights whilst reducing redundant tasks.
2. User Experience - To build a conducive working environment for the user to derive and test hypothesis with minimal inputs.

Keywords Context Aware · Agile · Platform

1 Introduction

The domain of research by itself is challenging. From choosing the right topic to dealing with the data and deriving an innovative solution is arduous for the researcher. The use of software helps the researcher in simulation, experimentation, analysis, and reporting. However, there are a plethora of tools available, and selecting the right set of tools for the experiment is quite exigent. Even then, researchers often face a steep learning curve for the chosen tool and are often not comfortable switching to new ones. The reason being; only a select portion of the researchers are from strong programming backgrounds and are at ease with the learning curve associated with a tool. However, a majority of researchers working in domains such as material sciences, biostatistics, chemical engineering, to name a few find it difficult to change their existing workflow. Any addition to their existing workflow drives away the researchers unless there are significant advantages over their existing process.

Catering to the needs of a wide spectrum of researchers poses a design challenge. The interface needs to be minimalist to ease the learning curve for the neophytes and yet, powerful enough for the veterans. *Design Thinking* is an iterative process to getting to know the needs of the user, challenge assumptions, and redefine problems in an attempt to identify alternative strategies and solutions that might not be instantly apparent with an initial level of understanding. It revolves

around one's interest in developing an understanding of the target users of the products or services. On the other hand, a software development kit (SDK) provides a set of tools, libraries, relevant documentation, code samples, processes, and guides that allow researchers to build workflows on a specific platform. With the help of design thinking and easy-to-use SDK, the platform can serve the requirements of users across diverse fields.

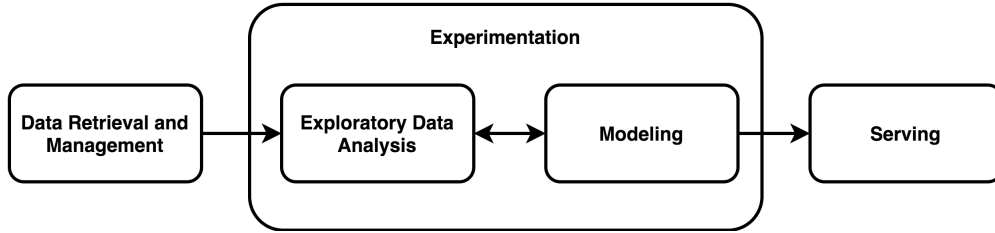


Figure 1: Research Workflow

An abstract overview of a research workflow is as shown in Figure 1. Data retrieval and management entails the task of creating a suitable data lake based on the data and its usage requirements. Once created, appropriate data extraction channels (batch-based or streaming) can be designed to retrieve the data from the data lake. The next phase, Exploratory Data Analysis (EDA) is a critical process of performing an initial examination on data to discover patterns, spot anomalies, test hypotheses, and check assumptions with the help of summary statistics and graphical representations. Modeling describes the use of data to derive meaningful interpretations or perform necessary tasks. EDA and Modeling phases require collective attention and are executed as experiments. Experimentation requires tracking and versioning for reproducibility and to identify the differences between experiments. Once a suitable model is derived for the given data, the resulting information is either sent out as a report or used to infer the outcome of a future event based on a previously trained task. The proposal elucidates the use of enabling context awareness for each of the modules coupled with a user-experience driven interface in reducing the redundant tasks and helping the researcher accomplish the set goals at ease.

2 Software Development Kit

SDK, a collection of utilities, is built to access a complex set of procedures with ease. Designing an SDK with the principles of easy-to-use APIs, lightweight and well documented is beneficial for many reasons.

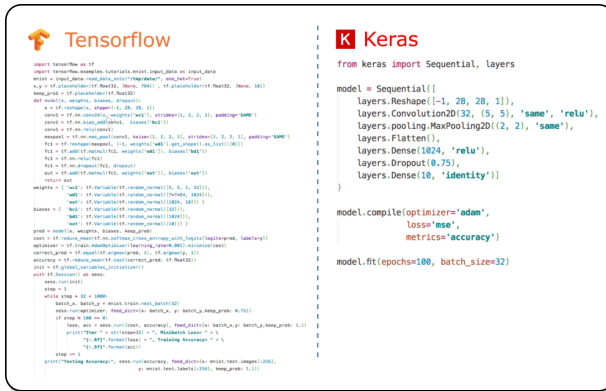


Figure 2: Comparison of Tensorflow vs Keras

As an example, consider Tensorflow [1], a celebrated deep learning framework among computer scientists that can be overwhelming for statisticians and researchers from other domains. This is because the focus was on the functionality of the framework and not much thought was given to the user experience. In recent times, Keras [2] a machine learning library emphasizing on user-experience is widely adopted as it minimizes the number of user actions required for common use cases and provides clear and actionable error messages. The framework also features extensive documentation and developer guides. Figure 2 demonstrates the importance of UX, where Keras framework has a clear edge over Tensorflow.

Usability is one of the functionalities of an SDK. Others include - using contextual information to provide relevant suggestions and analytical reports. Over time, with the help of SDK, one can monitor the usage of data. For structured data, with the use of active learning the saliency of columns can be derived based on their usage and one can learn the presence of redundant information. Therefore, expunging the need for plugging in experimental values.

Usage analytics also plays a vital role in motivating researchers to strictly follow a standardized workflow. This is notable in collaborative efforts in the open-source projects. Providing researchers with information as to the significance of their contribution and its impact on other projects will only motivate them to adopt best practices

3 Data Retrieval and Management

Researchers find data collection and retrieval a quite tedious process. The collection of data in a non-obtrusive and effortless manner is preferred over the rudimentary tabular based data entry. Often retrieved data exists in different formats such as CSV, TSV, JSON, handwritten notes, etc. Over-time, these datasets are lost due to the lack of a structured approach and traceability in the workflow and creates the need for a data lake. Recent advancements in machine learning have enabled one to reuse and extend existing data for architecting cross-domain solutions. The context-aware framework aims to solve the three major problems faced by researchers,

- **Data Retrieval** - Seamless retrieval of existing data with UX focussed data mapping for structured files and developing vision-based technologies to digitize handwritten readings in notes.
- **Data Collection** - Reduce the efforts of manual data collection by deriving insights from multi-dimensional representations of image, audio, and text.
- **Exploratory Data Analysis (EDA)** - An interactive tool that understands the schema, describes the statistics and supports custom visualizations. More on this in Section 4.

3.1 Data Retrieval

Historical experimental data collected by the researchers exist in different digital formats such as CSV, TSV, JSON, etc. In order to understand the data and design visualizations, the data needs to be migrated from the existing digital formats to a data lake. A data lake is a centralized repository to store structured and unstructured data at any scale. The centralized repository can then be used to run different types of analytics—from dashboards and visualizations to big data processing, real-time analytics, and machine learning. Thereby, enabling one to derive meaningful insights. The process of migration is an arduous task due to the difference in schema between the source and target systems. At times, trivial issues such as missing keys, duplicate keys, multiple keys for identical data persists. The issues can be resolved by having a context-aware UX focussed data mapping (Figure 10) that supports - schema detection and inference, similarity measures, easy-to-use drag and drop user interface for mapping source schema to the target schema.

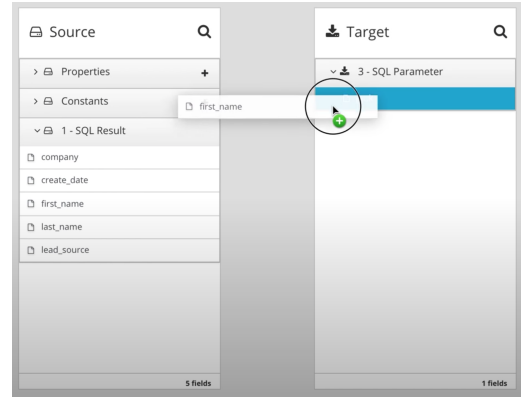


Figure 3: UX focussed data mapping



Figure 4: Tabular Row and Column detection

Published datasets are usually represented as tabular structures in portable document format. In order to perform EDA on the dataset, it is necessary to develop valuable information extraction tools. Table Extraction is the task of detecting and decomposing table information in a document. There are many ways to extract the required information, among them TableNet [3], DeepDeSRT [4], Graph Neural Networks [5] are considered to be the state-of-the-art methods. A feasible solution is selected based on the requirements. Figure 4 shows the results of DeepDeSRT, a deep learning-based solution for table detection.

A large portion of everyday documents is handwritten. Archiving the handwritten documents is limited to storing high-resolution images of the documents. Since traditional automation methods fail to parse textual information in the document, manually transcribing the handwritten text is preferred. Manual transcription is a painstaking process and is not a scalable solution. The field of machine learning offers automated pipelines to transcribe handwritten text. Figure 5 shows the design of workflow and the use of Maximally Stable Extremal Regions (MSERs) algorithm with Deep CNN [6] for page segmentation to identify the regions of text within a document, single shot multibox detector (SSD) [7] architecture to detect the positions of each line of the passage and convolution-recurrent networks optimized using CTC loss with beam-search decoder [8] for character recognition.

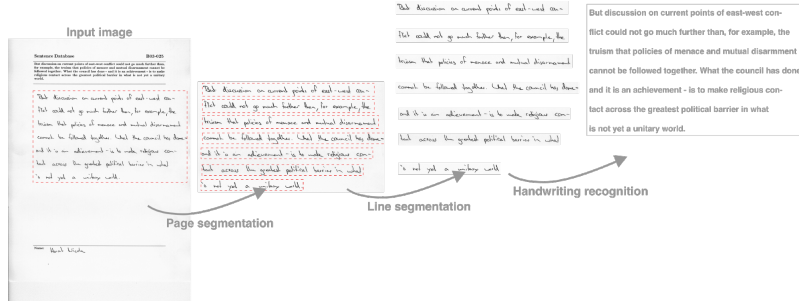


Figure 5: Automated Transcription of Handwritten Text

3.2 Data Collection

Researchers conventionally follow a laborious workflow to record the variables of interest. Typically, the data is structured and is stored as comma-separated values in text files, spreadsheets and are the go-to methods due to their simplicity and ease-of-use. Though at first, it may seem a feasible approach towards data collection, in the long run, it turns out to be a strenuous effort to keep track of the files, support versioning, to observe the frequency of modifications and the usefulness of the collected data hence the need for a systematic structured data storage mechanism.

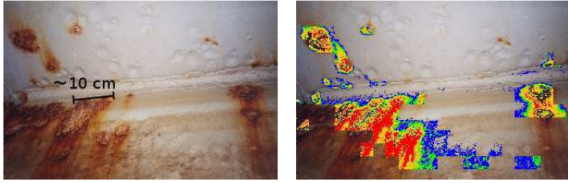


Figure 6: Corrosion Detection for Automated Visual Inspection

With the help of new-found methods, the laborious manual task of experimental data collection can be automated. In the first example (Figure 6), researchers re-engineer the inspection process through vision-based pattern recognition approach. This automates the detection and measure of surface corrosion by analyzing the captured digital images through a Weak-classifier Colour-based Corrosion Detector [9].

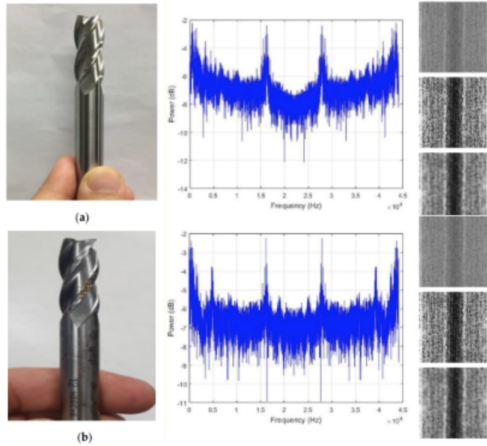


Figure 7: Drill Bits Monitoring Using the Spectral Analysis

Machine fault diagnosis (MFD) is now in the spotlight due to innovations in the domain of pattern recognition. Figure 7 showcases the use of spectral analysis to monitor the faults in drill bits. Figure 7'a' shows the power spectrum of a normal drill bit and 'b' is the display of the same measure of the faulty drill bit. The last column represents power spectrum density (PSD) - a visual representation for the data in the time domain. The PSD-images are then given to a deep convolutional auto-encoder for high-level feature extraction [10].

These examples demonstrate the capabilities of deep-learning and cross-domain solutions in automating the data entry mechanisms required to capture the quality measures for a given experiment.

3.2.1 Synthetic Data

Sampling bias is a problem tackled by researchers over many years. In classification problems, sampling techniques such as oversampling, undersampling, stratified sampling are used to mitigate the effects of class imbalance. However, with the surge in learnable parameters (weights/degrees of freedom) of a given model, basic statistical sampling techniques fail to solve the issue of the bias. Synthetic Minority Over-sampling Technique (SMOTE) [11] - is a widely used statistical method to generate synthetic data. The domain of deep learning shows promising new approaches with Variational AutoEncoder (VAE) [12] and the sophisticated multi-model method - Generative Adversarial Network (GAN) [13] aimed at solving the issue of sampling bias. These methods are known to work in diverse fields such as structured data, image, audio, and text.

4 Exploratory Data Analysis

Visual representation of data plays a crucial role in motivating the researchers to follow standard data collection practices. Considering structured data makes it easier to infer the schema (data type of the column) and provide a respective visual representation. Context-awareness in EDA is introduced by providing real-time summary statistics to help understand current insights, the direction, and the nature of the experiment. Figure 8 showcases enhanced visual aids with summary statistics for a tabular representation of structured data by Kaggle [14]. The number of unique values is shown as a statistic for a column of string data-type. The distribution of categories is shown for a categorical column and missing values, NaN's are displayed for a continuous column. By extending the existing summary statistics for the required domain, the process of data collection becomes meaningful for the researcher and not only for the organization point of view.

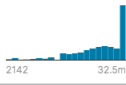
cord_uid	sha	source_x	title	dol	pmcid	pubmed_id	license	abstract
135194 unique values	[null] 52% 0ed3c6a5559cd73... 0% Other (64676) 48%	Medline 32% PMC 18% Other (68511) 50%	128486 unique values	[null] 18% 10.1016/j.scitotenv.2... 0% Other (111683) 82%	[null] 49% PMC35282 0% Other (69016) 51%		unk 44% no-cc 30% Other (34884) 26%	[null] Unknown Other (104833)
ejv2x1n0	06ced00a5fc4215949a a72528f2eeaae1d58927	PMC	Surfactant protein-D and pulmonary host defense	10.1186/rr19	PMC59549	11667972	no-cc	Surfactant protei (SP-D) participat in the innate response to inhal microorganisms an organi...
2b73a28n	348055649b6b8c72b9a3 76498df9b4f17123685	PMC	Role of endothelin-1 in lung disease	10.1186/rr44	PMC59574	11686871	no-cc	Endothelin-1 (ET-1) is a 21 amino acid peptide with diverse biological activities that has been implicated in...
9785vg6d	5f48792a5fa08bed9f56 016f4981ae2ca6831b32	PMC	Gene expression in epithelial cells in response to pneumovirus infection	10.1186/rr61	PMC59588	11686888	no-cc	Respiratory syncytial virus (RSV) and pneumovirus of mice (PMV) are viruses of the family Paramy...
zjufx4fo	b2897e1277f56641193a 6db73825f707eed3e4c9	PMC	Sequence requirements for RNA strand transfer during nidovirus discontinuous subgenomic RNA synthesis...	10.1093/emboj/20.24.7228	PMC125348	11742998	green-oa	Nidovirus subgenomic mRNAs contain a leader sequence derived from the end of the genome fused to ...
5yhe786e	3bb07ea10432f7738413 dff9816889cc9ef83f99	PMC	Debate: Transfusing to normal haemoglobin levels will not improve outcome	10.1186/cc987	PMC137267	11299862	no-cc	Recent evidence suggests that critically ill patients are able to tolerate lower levels of haemoglobin...

Figure 8: Enhanced tabular representation of structured data

Tools such as MongoCharts [15] - visualizations for unstructured data (MongoDB), Plotly Dashboard [16] - Collection of plots and images organized with a layout, RawGraphs [17] - a web-based visualization platform, support creating dynamic web-based interactive plots. The plug and play modularity offered by these packages ease the learning curve required for creating custom interactions.

4.1 Feature Engineering

Feature engineering turns raw data into refined data suitable for Machine learning algorithms to ease the detection of complex patterns. Data scientists often spend hours together trying to get the right set of features and introduce new ones as and when required. Usually, this is done by following a sequence of steps mentioned below:

- **Cleansing** - Duplicate rows, target column with NaN's
- **Missing Value Treatment** - Continuous and categorical column imputation by replacing NaN values with the mode
- **Outlier Treatment** - Anomaly detection, Standard-dev based detection
- **Encoding** - Binary, One-Hot based encodings for Categorical columns, Binning strategies for continuous columns
- **Normalization** - Z-Score, Robust, Min-max based normalization approaches, based on the data
- **Dimensionality Reduction** - PCA, High correlation filter, low variance filter

Context-aware based methods understand the schema and composition of the data for a given problem setting. This enables one to design automated approaches that reduce repetitive tasks of cleansing, encoding, normalization, etc.

5 Modeling

Automated machine learning (AutoML) aims to reduce the need for researchers to build suitable machine learning models for the given data. For supervised learning, it revolves around the idea of "Neural Networks designing Neural Networks". Transfer learning and Neural Architecture Search (NAS) are two core techniques pushing the boundaries of AutoML. Few interesting ideas to consider are - Differentiable Neural Architecture Search (DARTS) [18] transforms the discrete neural network architecture search space into a continuous and differentiable form, enabling the use of standard gradient-based optimization and Proxyless NAS [19] learn the architectures for large-scale target tasks and target hardware platforms i.e, efficient network architectures for a given platform - enterprise or a mobile device.

In the domain of unsupervised learning, clustering is used to group similar data points and derive insights about the distribution of the data. Kullback–Leibler divergence (KLD), a measure of the difference between the probability distributions of variables can be used to determine the driving factors for a given cluster. Based on the identified driving factors, suitable names are given to the cluster - Autonoming cluster to enhance the statistical representation of the cluster.

In machine learning, hyperparameter is a parameter that controls the learning process. Hyperparameter tuning is the process of searching for the ideal control parameters to enhance the learning process. A combinatorial approach to tuning hyperparameter is not a scalable solution and fails when the number of hyperparameters and the range of each hyperparameter is increased. These days, state-of-the-art distributed tuning approaches based on Bayesian optimization are largely adopted to enhance the performance of the neural network. HyperOpt [20], Facebook's experimentation platform - AX [21] and Ray - Distributed framework for AI application [22] are a few of the widely adopted frameworks for hyperparameter tuning.

6 Serving

Deployment can be a daunting process for users with a non-programming background as one needs to have an understanding of high-performance computing (HPC) due to the presence of heterogeneous distributed environments with a mixture of devices such as CPUs and GPUs. Depending on the use case, developing applications in low-level languages such as C and C++ becomes requisite. The steep learning curve associated with *serving* often forces the researchers to compromise on the scale and performance of the solution. Big data and active learning techniques, such as reinforcement learning, can aid algorithm-specific device placement and resource management in HPC clusters. The decision of placing parts of the neural models on devices is often made by domain experts on simple heuristics and intuitions. The process of device placement can be automated by using neural machine translation to provide contextual information and reinforcement learning to act on the learned context [23]. Resource management is a difficult real-time decision-making task where the solution varies based on the workload and environment. Deep reinforcement learning can learn to manage resources directly from experience [24]. Active learning-based techniques can ease the workload of researchers by reducing the efforts required for serving.

6.1 Explainable AI

Deep-learning models are opaque and no information is provided regarding the decision process. Arriving at a prediction based on a learning approach only solves a part of the problem. The rest can be tackled by understanding the decision making process of the learning method. Deep-learning methods represent data in high dimensional space to identify complex patterns for the given task.

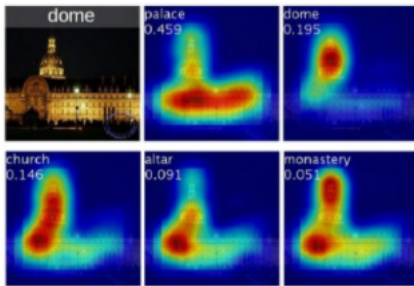


Figure 9: Class activation maps of top 5 predictions

Understanding these representations by translating it into lower dimensions is the key to uncover the reason behind a predicted event. The use of a global average pooling layer to enable object localization in convolutional networks [25] results in class activation mapping. Figure 9 showcases class activation mapping, a visual reasoning technique using heatmaps to explain what the model is looking at for the given prediction. CAM is a white-box approach that involves adapting the neural architecture in design time to provide meaningful explanations.

Explaining learning methods for structure data involves deriving the driving factors for prediction or classification. Shapely additive explanation [26], a black box explainer, takes a game-theoretic approach to explain the output of a machine learning model. The use of optimal credit allocation coupled with local explanation provides the contributing factors - the reasoning behind the predictions.

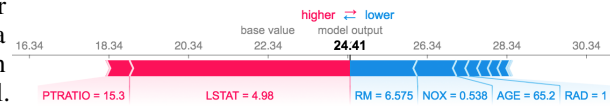


Figure 10: Shapeley additive explanations

Explainable AI provides the reasoning and the same can be used to debug the learning process of the model. With the help of these techniques, it is easier to translate a probabilistic value to a meaningful explanation for business usecase.

7 Summary

Data collection, although a tedious process, is crucial to comprehend the existing state and the directional flow of an experiment. There are three tiers of data - Synthetic, Historical, and Real-Time with challenges unique to each one of them. The proposal describes the benefits of using context-aware data-driven approaches for data collection and retrieval mechanisms to tackle the challenges and to automate the redundant and laborious manual tasks. The work also demonstrates the use of design thinking driven user-experience layouts to motivate the researchers to adopt a systematic and standardized experimentation workflow. Automated data collection mechanisms for image and audio domain in applications such as "Material corrosion detection" and "Machine fault diagnosis" can be used to gather the measures of important variables without manual efforts.

The proposal demonstrates the idea of context-aware based exploratory data analysis as a feasible approach to automate the existing workflow by learning heuristics in the absence of the domain experts. The work describes introducing context-awareness in modeling workflow by adopting *Neural Architecture Search* and active-learning based hyperparameter tuning methods. In serving workflow, context-awareness can help in managing computing resources and selecting appropriate devices for a given operation. The use of Explainable AI brings in much-needed trust factor for large and complex models by explaining the reasons behind the prediction. The proposal describes the need for an end-to-end context-awareness agile platform driven by design thinking as well as showcases the benefits of such a platform to derive tangible insights whilst automating redundant tasks with minimal user input.

References

- [1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek Gordon Murray, Benoit Steiner, Paul A. Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zhang. Tensorflow: A system for large-scale machine learning. In *OSDI*, 2016.
- [2] François Chollet. Keras: The python deep learning library. 2018.
- [3] Shubham Paliwal, D Vishwanath, Rohit Rahul, Monika Sharma, and Lovekesh Vig. Tablenet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images. *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 128–133, 2019.
- [4] Sebastian Schreiber, Stefan Agne, Ivo Wolf, Andreas Dengel, and Sheraz Ahmed. Deepdesrt: Deep learning for detection and structure recognition of tables in document images. *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 01:1162–1167, 2017.
- [5] Shah Rukh Qasim, Hassan Mahmood, and Faisal Shafait. Rethinking table recognition using graph neural networks. *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 142–147, 2019.
- [6] Kai Chen, Mathias Seuret, Marcus Liwicki, Jean Hennebert, and Rolf Ingold. Page segmentation of historical document images with convolutional autoencoders. *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1011–1015, 2015.
- [7] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016.
- [8] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:2298–2304, 2017.
- [9] Francisco Bonnín-Pascual and Alberto Ortiz. Corrosion detection for automated visual inspection. 2014.

- [10] Caleb Vununu, Kwang-Seok Moon, Suk Hwan Lee, and Ki-Ryong Kwon. A deep feature learning method for drill bits monitoring using the spectral analysis of the acoustic signals. *Sensors (Basel, Switzerland)*, 18, 2018.
- [11] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.*, 16:321–357, 2002.
- [12] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2014.
- [13] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.
- [14] Anthony Goldbloom and Ben Hamner. Kaggle.
- [15] Mongo DB. Mongo charts, 2019.
- [16] Plotly. Plotly dash, 2019.
- [17] Raw Graphs. Raw graphs, 2019.
- [18] Richard Shin, Charles Packer, and Dawn Xiaodong Song. Differentiable neural network architecture search. In *ICLR*, 2018.
- [19] Han Cai, Ligeng Zhu, and Song Han. Proxylessnas: Direct neural architecture search on target task and hardware. *ArXiv*, abs/1812.00332, 2019.
- [20] James Bergstra, Brent Komer, Chris Eliasmith, Daniel L K Yamins, and David D. Cox. Hyperopt: a python library for model selection and hyperparameter optimization. *Computational Science Discovery*, 8:014008, 2015.
- [21] Eytan Bakshy, Lili Dworkin, Brian Karrer, Konstantin Kashin, Benjamin Letham, Ashwin Murthy, and Sonia Singh. Ae : A domain-agnostic platform for adaptive experimentation. 2018.
- [22] Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, William Paul, Michael I. Jordan, and Ion Stoica. Ray: A distributed framework for emerging ai applications. *ArXiv*, abs/1712.05889, 2018.
- [23] Azalia Mirhoseini, Hieu Pham, Quoc V. Le, Benoit Steiner, Rasmus Larsen, Yuefeng Zhou, Naveen Kumar, Mohammad Norouzi, Samy Bengio, and Jeff Dean. Device placement optimization with reinforcement learning. In *ICML*, 2017.
- [24] Hongzi Mao, Mohammad Alizadeh, Ishai Menache, and Srikanth Kandula. Resource management with deep reinforcement learning. *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*, 2016.
- [25] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, 2016.
- [26] Scott M Lundberg, Bala Nair, Monica S Vavilala, Mayumi Horibe, Michael J Eisses, Trevor Adams, David E Liston, Daniel King-Wai Low, Shu-Fang Newman, Jerry Kim, et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature Biomedical Engineering*, 2(10):749, 2018.

Appendix B. Platform Documentation

This appendix appears in its original form, as an attached pdf to ARL-SR-0469, *MIL-PRF-32662 Context-Aware Agile Platform*, by Robert Jensen, Kimberly Brady, Prathyush Parvatharaju, Biao Yin, Fatemeh Emdad, and Elke Rundensteiner.

ARL CAAP Documentation (Context-Aware Agile Platform)

Release beta 6.0.6

Prathyush SP

Nov 02, 2021

CONTENTS:

1	Quickstart	1
1.1	Requirements	1
1.2	What is ARL CAAP Commons?	1
1.3	What are the objectives of CAAP Commons?	1
1.4	Design Principles of CAAP Commons:	1
1.5	Install Prerequisites - Linux	2
1.6	Generate Documentation	2
1.7	Script Utils	2
1.8	Navigation	2
1.9	Contributing Guidelines	2
2	Installation	3
2.1	Git Configuration	3
2.2	Steps to update the repository with changes from arl-commons	3
2.3	Backend Requirements	3
2.4	Frontend Requirements	3
2.5	Backend local development	4
3	Configuration	5
3.1	Dashboard Configuration	5
3.1.1	Configuration Options:	5
3.1.2	Sample Configuration	5
4	Troubleshooting Guidelines	7
4.1	Backend local development, additional details	7
4.1.1	General workflow	7
4.1.2	Docker Compose Override	7
4.1.3	Backend tests	8
4.1.3.1	Local tests	9
4.1.3.2	Test running stack	9
4.1.3.3	Test Coverage	9
4.1.4	Live development with Python Jupyter Notebooks	9
4.1.5	Migrations	10
4.1.6	Development with Docker Toolbox	11
4.1.7	Development in localhost with a custom domain	11
4.1.8	Development with a custom IP	12
4.1.9	Change the development “domain”	12
4.2	Frontend development	13
4.2.1	Removing the frontend	14
4.3	Deployment	14

4.3.1	Traefik network	14
4.3.2	Persisting Docker named volumes	15
4.3.2.1	Adding services with volumes	15
4.3.2.2	<code>docker-auto-labels</code>	16
4.3.2.3	(Optionally) adding labels manually	16
4.3.3	Deploy to a Docker Swarm mode cluster	17
4.3.3.1	Deployment Technical Details	18
4.3.4	Continuous Integration / Continuous Delivery	19
4.4	Docker Compose files and env vars	19
4.4.1	The <code>.env</code> file	19
4.5	URLs	20
4.5.1	Production URLs	20
4.5.2	Staging URLs	20
4.5.3	Development URLs	20
4.5.4	Development with Docker Toolbox URLs	21
4.5.5	Development with a custom IP URLs	21
4.5.6	Development in localhost with a custom domain URLs	21
4.6	Project generation and updating, or re-generating	22
5	Changelog	23
5.1	(unreleased)	23
5.2	1.0 (2021-07-15)	35
6	Library Stats	41
7	About	42
8	API Reference	43
8.1	login	43
8.2	users	44
8.3	utils	45
8.4	items	45
8.5	adhesives	47
8.6	data_wrangler	47
9	Indices and tables	48
	HTTP Routing Table	49

QUICKSTART

1.1 Requirements

1. **OS:** macOS Mavericks+ or Ubuntu 14.04+
2. **Python:** v3.6.0+
3. **Tensorflow:** v1.13.1+

1.2 What is ARL CAAP Commons?

Context-Aware Agile Platform (CAAP) Commons is base framework for developing application specific tools. The framework is built using **ReactJS** (Frontend), **FastAPI** (Backend) and **PostgreSQL** (Database). The stack is deployed and executed as **docker** services.

1.3 What are the objectives of CAAP Commons?

1. Develop reusable cross-platform components
2. Provide Common set of software packages for IT Auditing
3. Avoid redundant features and codebase, enhancing robustness
4. Ease of use for Cross-project users - Layout, Stack, Options
5. Support Cross-Operating System Compatibility

1.4 Design Principles of CAAP Commons:

1. Simple and Clean API
2. Lightweight Framework [-mem -a]
3. Zero Memory Leaks [-mem]
4. Easy Installation [-install]
5. Proper Documentation [-docs -r] #### Installation ##### Install Prerequisites - Mac

```
brew install cloc pandoc graphviz figlet

<!-- For generating documentation in pdf -->
<!-- brew install mactex imagemagick ghostscript -->
```

1.5 Install Prerequisites - Linux

```
sudo apt-get install cloc pandoc graphviz figlet
<!-- # For generating documentation in pdf -->
<!-- # sudo apt-get install texlive latexmk imagemagick ghostscript -->
#### Python requirements
pip install piprot gitchangelog sphinx_autodoc_typehints sphinx_rtd_theme_
↪sphinxcontrib-versioning rst2pdf sphinx-markdown-builder
```

1.6 Generate Documentation

1.7 Script Utils

1.8 Navigation

1. [`Installation Documentation <>`__](#)
2. [`Todo List <>`__](#)
3. [`Simple Issue Tracker <>`__](#)
4. [`Dashboard Configuration Documentation <>`__](#)
5. [`Metadata Documentation <>`__](#)

1.9 Contributing Guidelines

1. Commit format - [COMMIT_TYPE] - COMMIT_MESSAGE
2. Commit Types:

- ```
1. Critical / Major Bug Fix - [BF]
2. Typos - [TY]
3. Enhancement / Major Refactor - [EN]
4. New Feature - [FE]
5. Deprecated - [DP]
6. Gardner Commits - [GA]
7. Revert - [RT]
8. Refactor - [RF]
```

## INSTALLATION

### 2.1 Git Configuration

```
For windows only
git config --global core.autocrlf false
git clone <repository url>
```

### 2.2 Steps to update the repository with changes from arl-commons

```
Initial Setup
git remote add upstream git@arcgit.wpi.edu:arl-commons/dashboard.git

Once initialized
git fetch upstream
git merge upstream/master master
git push
```

### 2.3 Backend Requirements

- Docker.
- Docker Compose.
- Poetry for Python package and environment management.

### 2.4 Frontend Requirements

- Node.js (with npm).

## 2.5 Backend local development

- Start the stack with Docker Compose:

```
docker-compose up -d
```

- Now you can open your browser and interact with these URLs:

Frontend, built with Docker, with routes handled based on the path: <http://localhost>

Backend, JSON based web API based on OpenAPI: <http://localhost/api/>

Automatic interactive documentation with Swagger UI (from the OpenAPI backend): <http://localhost/docs>

Alternative automatic documentation with ReDoc (from the OpenAPI backend): <http://localhost/redoc>

PGAdmin, PostgreSQL web administration: <http://localhost:5050>

Flower, administration of Celery tasks: <http://localhost:5555>

Traefik UI, to see how the routes are being handled by the proxy: <http://localhost:8090>

**Note:** The first time you start your stack, it might take a minute for it to be ready. While the backend waits for the database to be ready and configures everything. You can check the logs to monitor it.

To check the logs, run:

```
docker-compose logs
```

To check the logs of a specific service, add the name of the service, e.g.:

```
docker-compose logs backend
```

If your Docker is not running in `localhost` (the URLs above wouldn't work) check the sections below on **Development with Docker Toolbox** and **Development with a custom IP**.

## CONFIGURATION

### 3.1 Dashboard Configuration

Dashboard supports JSON Configuration using DLConfigManager API. A default configuration file is present in .env which is loaded during docker launch

#### 3.1.1 Configuration Options:

Generated using [jsonschema](#)

Note: Please change the configuration with caution.

#### 3.1.2 Sample Configuration

```
DOMAIN=localhost

STACK_NAME=arl-com

TRAEFIK_PUBLIC_NETWORK=traefik-public
TRAEFIK_TAG=arl.com
TRAEFIK_PUBLIC_TAG=traefik-public

DOCKER_IMAGE_BACKEND=arl-backend
DOCKER_IMAGE_CELERYWORKER=arl-celeryworker
DOCKER_IMAGE_FRONTEND=arl-frontend

Backend
BACKEND_CORS_ORIGINS=["http://localhost", "http://localhost:4200", "http://
↪localhost:3000", "http://localhost:8080", "https://localhost", "https://
↪localhost:4200", "https://localhost:3000", "https://localhost:8080", "http://dev.
↪arl.com", "https://stag.arl.com", "https://arl.com", "http://local.dockertoolbox.
↪tiangolo.com", "http://localhost.tiangolo.com"]
PROJECT_NAME=arl
SECRET_KEY=205f30042f28da9865b00f45c9bc74acd9cfefda7cf09a9c7c9337d88295873a
FIRST_SUPERUSER=admin@arl.com
FIRST_SUPERUSER_PASSWORD=admin@123
SMTP_TLS=True
SMTP_PORT=587
SMTP_HOST=
SMTP_USER=
SMTP_PASSWORD=
```

(continues on next page)

(continued from previous page)

```
EMAILS_FROM_EMAIL=info@arl.com

USERS_OPEN_REGISTRATION=True

SENTRY_DSN=

Flower
FLOWER_BASIC_AUTH=admin:admin@123

Postgres
POSTGRES_SERVER=db
POSTGRES_USER=postgres
POSTGRES_PASSWORD=admin@123
POSTGRES_DB=app

PgAdmin
PGADMIN_LISTEN_PORT=5050
PGADMIN_DEFAULT_EMAIL=admin@arl.com
PGADMIN_DEFAULT_PASSWORD=admin@123
```

## TROUBLESHOOTING GUIDELINES

### 4.1 Backend local development, additional details

#### 4.1.1 General workflow

By default, the dependencies are managed with [Poetry](#), go there and install it.

From `./backend/app/` you can install all the dependencies with:

```
$ poetry install
```

Then you can start a shell session with the new environment with:

```
$ poetry shell
```

Next, open your editor at `./backend/app/` (instead of the project root: `./`), so that you see an `./app/` directory with your code inside. That way, your editor will be able to find all the imports, etc. Make sure your editor uses the environment you just created with Poetry.

Modify or add SQLAlchemy models in `./backend/app/app/models/`, Pydantic schemas in `./backend/app/app/schemas/`, API endpoints in `./backend/app/app/api/`, CRUD (Create, Read, Update, Delete) utils in `./backend/app/app/crud/`. The easiest might be to copy the ones for Items (models, endpoints, and CRUD utils) and update them to your needs.

Add and modify tasks to the Celery worker in `./backend/app/app/worker.py`.

If you need to install any additional package to the worker, add it to the file `./backend/app/celeryworker.dockerfile`.

#### 4.1.2 Docker Compose Override

During development, you can change Docker Compose settings that will only affect the local development environment, in the file `docker-compose.override.yml`.

The changes to that file only affect the local development environment, not the production environment. So, you can add “temporary” changes that help the development workflow.

For example, the directory with the backend code is mounted as a Docker “host volume”, mapping the code you change live to the directory inside the container. That allows you to test your changes right away, without having to build the Docker image again. It should only be done during development, for production, you should build the Docker image with a recent version of the backend code. But during development, it allows you to iterate very fast.

There is also a command override that runs `/start-reload.sh` (included in the base image) instead of the default `/start.sh` (also included in the base image). It starts a single server process (instead of multiple, as would be for

production) and reloads the process whenever the code changes. Have in mind that if you have a syntax error and save the Python file, it will break and exit, and the container will stop. After that, you can restart the container by fixing the error and running again:

```
$ docker-compose up -d
```

There is also a commented out `command` override, you can uncomment it and comment the default one. It makes the backend container run a process that does “nothing”, but keeps the container alive. That allows you to get inside your running container and execute commands inside, for example a Python interpreter to test installed dependencies, or start the development server that reloads when it detects changes, or start a Jupyter Notebook session.

To get inside the container with a `bash` session you can start the stack with:

```
$ docker-compose up -d
```

and then `exec` inside the running container:

```
$ docker-compose exec backend bash
```

You should see an output like:

```
root@7f2607af31c3: /app#
```

that means that you are in a `bash` session inside your container, as a `root` user, under the `/app` directory.

There you can use the script `/start-reload.sh` to run the debug live reloading server. You can run that script from inside the container with:

```
$ bash /start-reload.sh
```

...it will look like:

```
root@7f2607af31c3: /app# bash /start-reload.sh
```

and then hit enter. That runs the live reloading server that auto reloads when it detects code changes.

Nevertheless, if it doesn't detect a change but a syntax error, it will just stop with an error. But as the container is still alive and you are in a Bash session, you can quickly restart it after fixing the error, running the same command (“up arrow” and “Enter”).

...this previous detail is what makes it useful to have the container alive doing nothing and then, in a Bash session, make it run the live reload server.

### 4.1.3 Backend tests

To test the backend run:

```
$ DOMAIN=backend sh ./scripts/test.sh
```

The file `./scripts/test.sh` has the commands to generate a testing `docker-stack.yml` file, start the stack and test it.

The tests run with Pytest, modify and add tests to `./backend/app/app/tests/`.

If you use GitLab CI the tests will run automatically.

#### 4.1.3.1 Local tests

Start the stack with this command:

```
DOMAIN=backend sh ./scripts/test-local.sh
```

The `./backend/app` directory is mounted as a “host volume” inside the docker container (set in the file `docker-compose.dev.volumes.yml`). You can rerun the test on live code:

```
docker-compose exec backend /app/tests-start.sh
```

#### 4.1.3.2 Test running stack

If your stack is already up and you just want to run the tests, you can use:

```
docker-compose exec backend /app/tests-start.sh
```

That `/app/tests-start.sh` script just calls `pytest` after making sure that the rest of the stack is running. If you need to pass extra arguments to `pytest`, you can pass them to that command and they will be forwarded.

For example, to stop on first error:

```
docker-compose exec backend bash /app/tests-start.sh -x
```

#### 4.1.3.3 Test Coverage

Because the test scripts forward arguments to `pytest`, you can enable test coverage HTML report generation by passing `--cov-report=html`.

To run the local tests with coverage HTML reports:

```
DOMAIN=backend sh ./scripts/test-local.sh --cov-report=html
```

To run the tests in a running stack with coverage HTML reports:

```
docker-compose exec backend bash /app/tests-start.sh --cov-report=html
```

#### 4.1.4 Live development with Python Jupyter Notebooks

If you know about Python [Jupyter Notebooks](#), you can take advantage of them during local development.

The `docker-compose.override.yml` file sends a variable `env` with a value `dev` to the build process of the Docker image (during local development) and the `Dockerfile` has steps to then install and configure Jupyter inside your Docker container.

So, you can enter into the running Docker container:

```
docker-compose exec backend bash
```

And use the environment variable `$JUPYTER` to run a Jupyter Notebook with everything configured to listen on the public port (so that you can use it from your browser).

It will output something like:

```

root@73e0ec1f1ae6:/app# $JUPYTER
[I 12:02:09.975 NotebookApp] Writing notebook server cookie secret to /root/.local/
↪share/jupyter/runtime/notebook_cookie_secret
[I 12:02:10.317 NotebookApp] Serving notebooks from local directory: /app
[I 12:02:10.317 NotebookApp] The Jupyter Notebook is running at:
[I 12:02:10.317 NotebookApp] http://(73e0ec1f1ae6 or 127.0.0.1):8888/?
↪token=f20939a41524d021fbfc62b31be8ea4dd9232913476f4397
[I 12:02:10.317 NotebookApp] Use Control-C to stop this server and shut down all
↪kernels (twice to skip confirmation).
[W 12:02:10.317 NotebookApp] No web browser found: could not locate runnable browser.
[C 12:02:10.317 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
 http://(73e0ec1f1ae6 or 127.0.0.1):8888/?
↪token=f20939a41524d021fbfc62b31be8ea4dd9232913476f4397

```

you can copy that URL and modify the “host” to be localhost or the domain you are using for development (e.g. local.dockertoolbox.tiangolo.com), in the case above, it would be, e.g.:

```
http://localhost:8888/token=f20939a41524d021fbfc62b31be8ea4dd9232913476f4397
```

and then open it in your browser.

You will have a full Jupyter Notebook running inside your container that has direct access to your database by the container name (db), etc. So, you can just run sections of your backend code directly, for example with [VS Code Python Jupyter Interactive Window](#) or [Hydrogen](#).

## 4.1.5 Migrations

As during local development your app directory is mounted as a volume inside the container, you can also run the migrations with alembic commands inside the container and the migration code will be in your app directory (instead of being only inside the container). So you can add it to your git repository.

Make sure you create a “revision” of your models and that you “upgrade” your database with that revision every time you change them. As this is what will update the tables in your database. Otherwise, your application will have errors.

- Start an interactive session in the backend container:

```
$ docker-compose exec backend bash
```

- If you created a new model in ./backend/app/app/models/, make sure to import it in ./backend/app/app/db/base.py, that Python module (base.py) that imports all the models will be used by Alembic.
- After changing a model (for example, adding a column), inside the container, create a revision, e.g.:

```
$ alembic revision --autogenerate -m "Add column last_name to User model"
```

- Commit to the git repository the files generated in the alembic directory.
- After creating the revision, run the migration in the database (this is what will actually change the database):

```
$ alembic upgrade head
```

If you don’t want to use migrations at all, uncomment the line in the file at ./backend/app/app/db/init\_db.py with:

```
Base.metadata.create_all(bind=engine)
```

and comment the line in the file `prestart.sh` that contains:

```
$ alembic upgrade head
```

If you don't want to start with the default models and want to remove them / modify them, from the beginning, without having any previous revision, you can remove the revision files (.py Python files) under `./backend/app/alembic/versions/`. And then create a first migration as described above.

### 4.1.6 Development with Docker Toolbox

If you are using **Docker Toolbox** in Windows or macOS instead of **Docker for Windows** or **Docker for Mac**, Docker will be running in a VirtualBox Virtual Machine, and it will have a local IP different than `127.0.0.1`, which is the IP address for `localhost` in your machine.

The address of your Docker Toolbox virtual machine would probably be `192.168.99.100` (that is the default).

As this is a common case, the domain `local.dockertoolbox.tiangolo.com` points to that (private) IP, just to help with development (actually `dockertoolbox.tiangolo.com` and all its subdomains point to that IP). That way, you can start the stack in Docker Toolbox, and use that domain for development. You will be able to open that URL in Chrome and it will communicate with your local Docker Toolbox directly as if it was a cloud server, including CORS (Cross Origin Resource Sharing).

If you used the default CORS enabled domains while generating the project, `local.dockertoolbox.tiangolo.com` was configured to be allowed. If you didn't, you will need to add it to the list in the variable `BACKEND_CORS_ORIGINS` in the `.env` file.

To configure it in your stack, follow the section **Change the development "domain"** below, using the domain `local.dockertoolbox.tiangolo.com`.

After performing those steps you should be able to open: <http://local.dockertoolbox.tiangolo.com> and it will be server by your stack in your Docker Toolbox virtual machine.

Check all the corresponding available URLs in the section at the end.

### 4.1.7 Development in localhost with a custom domain

You might want to use something different than `localhost` as the domain. For example, if you are having problems with cookies that need a subdomain, and Chrome is not allowing you to use `localhost`.

In that case, you have two options: you could use the instructions to modify your system `hosts` file with the instructions below in **Development with a custom IP** or you can just use `localhost.tiangolo.com`, it is set up to point to `localhost` (to the IP `127.0.0.1`) and all its subdomains too. And as it is an actual domain, the browsers will store the cookies you set during development, etc.

If you used the default CORS enabled domains while generating the project, `localhost.tiangolo.com` was configured to be allowed. If you didn't, you will need to add it to the list in the variable `BACKEND_CORS_ORIGINS` in the `.env` file.

To configure it in your stack, follow the section **Change the development "domain"** below, using the domain `localhost.tiangolo.com`.

After performing those steps you should be able to open: <http://localhost.tiangolo.com> and it will be server by your stack in `localhost`.

Check all the corresponding available URLs in the section at the end.

### 4.1.8 Development with a custom IP

If you are running Docker in an IP address different than `127.0.0.1` (`localhost`) and `192.168.99.100` (the default of Docker Toolbox), you will need to perform some additional steps. That will be the case if you are running a custom Virtual Machine, a secondary Docker Toolbox or your Docker is located in a different machine in your network.

In that case, you will need to use a fake local domain (`dev.arl.com`) and make your computer think that the domain is served by the custom IP (e.g. `192.168.99.150`).

If you used the default CORS enabled domains, `dev.arl.com` was configured to be allowed. If you want a custom one, you need to add it to the list in the variable `BACKEND_CORS_ORIGINS` in the `.env` file.

- Open your `hosts` file with administrative privileges using a text editor:
  - **Note for Windows:** If you are in Windows, open the main Windows menu, search for “notepad”, right click on it, and select the option “open as Administrator” or similar. Then click the “File” menu, “Open file”, go to the directory `c:\Windows\System32\Drivers\etc\`, select the option to show “All files” instead of only “Text (.txt) files”, and open the `hosts` file.
  - **Note for Mac and Linux:** Your `hosts` file is probably located at `/etc/hosts`, you can edit it in a terminal running `sudo nano /etc/hosts`.
- Additional to the contents it might have, add a new line with the custom IP (e.g. `192.168.99.150`) a space character, and your fake local domain: `dev.arl.com`.

The new line might look like:

```
192.168.99.100 dev.arl.com
```

- Save the file.
  - **Note for Windows:** Make sure you save the file as “All files”, without an extension of `.txt`. By default, Windows tries to add the extension. Make sure the file is saved as is, without extension.

...that will make your computer think that the fake local domain is served by that custom IP, and when you open that URL in your browser, it will talk directly to your locally running server when it is asked to go to `dev.arl.com` and think that it is a remote server while it is actually running in your computer.

To configure it in your stack, follow the section **Change the development “domain”** below, using the domain `dev.arl.com`.

After performing those steps you should be able to open: <http://dev.arl.com> and it will be server by your stack in `localhost`.

Check all the corresponding available URLs in the section at the end.

### 4.1.9 Change the development “domain”

If you need to use your local stack with a different domain than `localhost`, you need to make sure the domain you use points to the IP where your stack is set up. See the different ways to achieve that in the sections above (i.e. using Docker Toolbox with `local.dockertoolbox.tiangolo.com`, using `localhost.tiangolo.com` or using `dev.arl.com`).

To simplify your Docker Compose setup, for example, so that the API docs (Swagger UI) knows where is your API, you should let it know you are using that domain for development. You will need to edit 1 line in 2 files.

- Open the file located at `./env`. It would have a line like:

```
DOMAIN=localhost
```

- Change it to the domain you are going to use, e.g.:

```
DOMAIN=localhost.tiangolo.com
```

That variable will be used by the Docker Compose files.

- Now open the file located at `./frontend/.env`. It would have a line like:

```
VUE_APP_DOMAIN_DEV=localhost
```

- Change that line to the domain you are going to use, e.g.:

```
VUE_APP_DOMAIN_DEV=localhost.tiangolo.com
```

That variable will make your frontend communicate with that domain when interacting with your backend API, when the other variable `VUE_APP_ENV` is set to `development`.

After changing the two lines, you can re-start your stack with:

```
docker-compose up -d
```

and check all the corresponding available URLs in the section at the end.

## 4.2 Frontend development

```
Creating an optimized production build...
The build failed because the process exited too early. This probably means the system
↳ran out of memory or someone called `kill -9` on the process.
```

Solution: Increase memory and swap for docker

- Enter the `frontend` directory, install the NPM packages and start the live server using the `npm` scripts:

```
cd frontend
npm install
npm run serve
```

Then open your browser at <http://localhost:8080>

Notice that this live server is not running inside Docker, it is for local development, and that is the recommended workflow. Once you are happy with your frontend, you can build the frontend Docker image and start it, to test it in a production-like environment. But compiling the image at every change will not be as productive as running the local development server with live reload.

Check the file `package.json` to see other available options.

If you have Vue CLI installed, you can also run `vue ui` to control, configure, serve, and analyze your application using a nice local web user interface.

If you are only developing the frontend (e.g. other team members are developing the backend) and there is a staging environment already deployed, you can make your local development code use that staging API instead of a full local Docker Compose stack.

To do that, modify the file `./frontend/.env`, there's a section with:

```
VUE_APP_ENV=development
VUE_APP_ENV=staging
```

- Switch the comment, to:

```
VUE_APP_ENV=development
VUE_APP_ENV=staging
```

### 4.2.1 Removing the frontend

If you are developing an API-only app and want to remove the frontend, you can do it easily:

- Remove the `./frontend` directory.
- In the `docker-compose.yml` file, remove the whole service / section `frontend`.
- In the `docker-compose.override.yml` file, remove the whole service / section `frontend`.

Done, you have a frontend-less (api-only) app.

---

If you want, you can also remove the `FRONTEND` environment variables from:

- `.env`
- `.gitlab-ci.yml`
- `./scripts/*.sh`

But it would be only to clean them up, leaving them won't really have any effect either way.

## 4.3 Deployment

You can deploy the stack to a Docker Swarm mode cluster with a main Traefik proxy, set up using the ideas from DockerSwarm.rocks, to get automatic HTTPS certificates, etc.

And you can use CI (continuous integration) systems to do it automatically.

But you have to configure a couple things first.

### 4.3.1 Traefik network

This stack expects the public Traefik network to be named `traefik-public`, just as in the tutorials in Docker-Swarm.rocks.

If you need to use a different Traefik public network name, update it in the `docker-compose.yml` files, in the section:

```
networks:
 traefik-public:
 external: true
```

Change `traefik-public` to the name of the used Traefik network. And then update it in the file `.env`:

```
TRAEFIK_PUBLIC_NETWORK=traefik-public
```

### 4.3.2 Persisting Docker named volumes

You need to make sure that each service (Docker container) that uses a volume is always deployed to the same Docker “node” in the cluster, that way it will preserve the data. Otherwise, it could be deployed to a different node each time, and each time the volume would be created in that new node before starting the service. As a result, it would look like your service was starting from scratch every time, losing all the previous data.

That’s specially important for a service running a database. But the same problem would apply if you were saving files in your main backend service (for example, if those files were uploaded by your users, or if they were created by your system).

To solve that, you can put constraints in the services that use one or more data volumes (like databases) to make them be deployed to a Docker node with a specific label. And of course, you need to have that label assigned to one (only one) of your nodes.

#### 4.3.2.1 Adding services with volumes

For each service that uses a volume (databases, services with uploaded files, etc) you should have a label constraint in your `docker-compose.yml` file.

To make sure that your labels are unique per volume per stack (for example, that they are not the same for `prod` and `stag`) you should prefix them with the name of your stack and then use the same name of the volume.

Then you need to have those constraints in your `docker-compose.yml` file for the services that need to be fixed with each volume.

To be able to use different environments, like `prod` and `stag`, you should pass the name of the stack as an environment variable. Like:

```
STACK_NAME=stag-ar1-com sh ./scripts/deploy.sh
```

To use and expand that environment variable inside the `docker-compose.yml` files you can add the constraints to the services like:

```
version: '3'
services:
 db:
 volumes:
 - 'app-db-data:/var/lib/postgresql/data/pgdata'
 deploy:
 placement:
 constraints:
 - node.labels.${STACK_NAME?Variable not set}.app-db-data == true
```

note the `${STACK_NAME?Variable not set}`. In the script `./scripts/deploy.sh`, the `docker-compose.yml` would be converted, and saved to a file `docker-stack.yml` containing:

```
version: '3'
services:
 db:
 volumes:
 - 'app-db-data:/var/lib/postgresql/data/pgdata'
 deploy:
 placement:
 constraints:
 - node.labels.ar1-com.app-db-data == true
```

**Note:** The `${STACK_NAME?Variable not set}` means “use the environment variable `STACK_NAME`, but if it is not set, show an error `Variable not set`”.

If you add more volumes to your stack, you need to make sure you add the corresponding constraints to the services that use that named volume.

Then you have to create those labels in some nodes in your Docker Swarm mode cluster. You can use `docker-auto-labels` to do it automatically.

#### 4.3.2.2 `docker-auto-labels`

You can use `docker-auto-labels` <<https://github.com/tiangolo/docker-auto-labels>> to automatically read the placement constraint labels in your Docker stack (Docker Compose file) and assign them to a random Docker node in your Swarm mode cluster if those labels don't exist yet.

To do that, you can install `docker-auto-labels`:

```
pip install docker-auto-labels
```

And then run it passing your `docker-stack.yml` file as a parameter:

```
docker-auto-labels docker-stack.yml
```

You can run that command every time you deploy, right before deploying, as it doesn't modify anything if the required labels already exist.

#### 4.3.2.3 (Optionally) adding labels manually

If you don't want to use `docker-auto-labels` or for any reason you want to manually assign the constraint labels to specific nodes in your Docker Swarm mode cluster, you can do the following:

- First, connect via SSH to your Docker Swarm mode cluster.
- Then check the available nodes with:

```
$ docker node ls
```

// you would see an output like:

| ID                          | MANAGER STATUS | HOSTNAME          | STATUS | AVAILABILITY |
|-----------------------------|----------------|-------------------|--------|--------------|
| nfa3d4df2df34as2fd34230rm * | Reachable      | dog.example.com   | Ready  | Active       |
| 2c2sd2342asdfasd42342304e   | Leader         | cat.example.com   | Ready  | Active       |
| c4sdf2342asdfasd4234234ii   | Reachable      | snake.example.com | Ready  | Active       |

then chose a node from the list. For example, `dog.example.com`.

- Add the label to that node. Use as label the name of the stack you are deploying followed by a dot (.) followed by the named volume, and as value, just `true`, e.g.:

```
docker node update --label-add arl-com.app-db-data=true dog.example.com
```

- Then you need to do the same for each stack version you have. For example, for staging you could do:

```
docker node update --label-add stag-ar1-com,app-db-data=true cat.example.com
```

### 4.3.3 Deploy to a Docker Swarm mode cluster

There are 3 steps:

1. **Build** your app images
2. Optionally, **push** your custom images to a Docker Registry
3. **Deploy** your stack

Here are the steps in detail:

#### 1. Build your app images

- Set these environment variables, right before the next command:
  - TAG=prod
  - FRONTEND\_ENV=production
- Use the provided `scripts/build.sh` file with those environment variables:

```
TAG=prod FRONTEND_ENV=production bash ./scripts/build.sh
```

#### 2. Optionally, push your images to a Docker Registry

**Note:** if the deployment Docker Swarm mode “cluster” has more than one server, you will have to push the images to a registry or build the images in each server, so that when each of the servers in your cluster tries to start the containers it can get the Docker images for them, pulling them from a Docker Registry or because it has them already built locally.

If you are using a registry and pushing your images, you can omit running the previous script and instead using this one, in a single shot.

- Set these environment variables:
  - TAG=prod
  - FRONTEND\_ENV=production
- Use the provided `scripts/build-push.sh` file with those environment variables:

```
TAG=prod FRONTEND_ENV=production bash ./scripts/build-push.sh
```

#### 3. Deploy your stack

- Set these environment variables:
  - DOMAIN=ar1.com
  - TRAEFIK\_TAG=ar1.com
  - STACK\_NAME=ar1-com
  - TAG=prod
- Use the provided `scripts/deploy.sh` file with those environment variables:

```
DOMAIN=arl.com \
TRAEFIK_TAG=arl.com \
STACK_NAME=arl-com \
TAG=prod \
bash ./scripts/deploy.sh
```

If you change your mind and, for example, want to deploy everything to a different domain, you only have to change the `DOMAIN` environment variable in the previous commands. If you wanted to add a different version / environment of your stack, like “preproduction”, you would only have to set `TAG=preproduction` in your command and update these other environment variables accordingly. And it would all work, that way you could have different environments and deployments of the same app in the same cluster.

#### 4.3.3.1 Deployment Technical Details

Building and pushing is done with the `docker-compose.yml` file, using the `docker-compose` command. The file `docker-compose.yml` uses the file `.env` with default environment variables. And the scripts set some additional environment variables as well.

The deployment requires using `docker stack` instead of `docker-swarm`, and it can’t read environment variables or `.env` files. Because of that, the `deploy.sh` script generates a file `docker-stack.yml` with the configurations from `docker-compose.yml` and injecting the environment variables in it. And then uses it to deploy the stack.

You can do the process by hand based on those same scripts if you wanted. The general structure is like this:

```
Use the environment variables passed to this script, as TAG and FRONTEND_ENV
And re-create those variables as environment variables for the next command
TAG=${TAG?Variable not set} \
Set the environment variable FRONTEND_ENV to the same value passed to this script.
↪with
a default value of "production" if nothing else was passed
FRONTEND_ENV=${FRONTEND_ENV-production?Variable not set} \
The actual comand that does the work: docker-compose
docker-compose \
Pass the file that should be used, setting explicitly docker-compose.yml avoids the
default of also using docker-compose.override.yml
-f docker-compose.yml \
Use the docker-compose sub command named "config", it just uses the docker-compose.
↪yaml
file passed to it and prints their combined contents
Put those contents in a file "docker-stack.yml", with ">"
config > docker-stack.yml

The previous only generated a docker-stack.yml file,
but didn't do anything with it yet

docker-auto-labels makes sure the labels used for constraints exist in the cluster
docker-auto-labels docker-stack.yml

Now this command uses that same file to deploy it
docker stack deploy -c docker-stack.yml --with-registry-auth "${STACK_NAME?Variable_
↪not set}"
```

### 4.3.4 Continuous Integration / Continuous Delivery

If you use GitLab CI, the included `.gitlab-ci.yml` can automatically deploy it. You may need to update it according to your GitLab configurations.

If you use any other CI / CD provider, you can base your deployment from that `.gitlab-ci.yml` file, as all the actual script steps are performed in `bash` scripts that you can easily re-use.

GitLab CI is configured assuming 2 environments following GitLab flow:

- `prod` (production) from the `production` branch.
- `stag` (staging) from the `master` branch.

If you need to add more environments, for example, you could imagine using a client-approved `preprod` branch, you can just copy the configurations in `.gitlab-ci.yml` for `stag` and rename the corresponding variables. The Docker Compose file and environment variables are configured to support as many environments as you need, so that you only need to modify `.gitlab-ci.yml` (or whichever CI system configuration you are using).

## 4.4 Docker Compose files and env vars

There is a main `docker-compose.yml` file with all the configurations that apply to the whole stack, it is used automatically by `docker-compose`.

And there's also a `docker-compose.override.yml` with overrides for development, for example to mount the source code as a volume. It is used automatically by `docker-compose` to apply overrides on top of `docker-compose.yml`.

These Docker Compose files use the `.env` file containing configurations to be injected as environment variables in the containers.

They also use some additional configurations taken from environment variables set in the scripts before calling the `docker-compose` command.

It is all designed to support several “stages”, like development, building, testing, and deployment. Also, allowing the deployment to different environments like staging and production (and you can add more environments very easily).

They are designed to have the minimum repetition of code and configurations, so that if you need to change something, you have to change it in the minimum amount of places. That's why files use environment variables that get auto-expanded. That way, if for example, you want to use a different domain, you can call the `docker-compose` command with a different `DOMAIN` environment variable instead of having to change the domain in several places inside the Docker Compose files.

Also, if you want to have another deployment environment, say `preprod`, you just have to change environment variables, but you can keep using the same Docker Compose files.

### 4.4.1 The `.env` file

The `.env` file is the one that contains all your configurations, generated keys and passwords, etc.

Depending on your workflow, you could want to exclude it from Git, for example if your project is public. In that case, you would have to make sure to set up a way for your CI tools to obtain it while building or deploying your project.

One way to do it could be to add each environment variable to your CI/CD system, and updating the `docker-compose.yml` file to read that specific env var instead of reading the `.env` file.

## 4.5 URLs

These are the URLs that will be used and generated by the project.

### 4.5.1 Production URLs

Production URLs, from the branch `production`.

Frontend: <https://arl.com>

Backend: <https://arl.com/api/>

Automatic Interactive Docs (Swagger UI): <https://arl.com/docs>

Automatic Alternative Docs (ReDoc): <https://arl.com/redoc>

PGAdmin: <https://pgadmin.arl.com>

Flower: <https://flower.arl.com>

### 4.5.2 Staging URLs

Staging URLs, from the branch `master`.

Frontend: <https://stag.arl.com>

Backend: <https://stag.arl.com/api/>

Automatic Interactive Docs (Swagger UI): <https://stag.arl.com/docs>

Automatic Alternative Docs (ReDoc): <https://stag.arl.com/redoc>

PGAdmin: <https://pgadmin.stag.arl.com>

Flower: <https://flower.stag.arl.com>

### 4.5.3 Development URLs

Development URLs, for local development.

Frontend: <http://localhost>

Backend: <http://localhost/api/>

Automatic Interactive Docs (Swagger UI): <https://localhost/docs>

Automatic Alternative Docs (ReDoc): <https://localhost/redoc>

PGAdmin: <http://localhost:5050>

Flower: <http://localhost:5555>

Traefik UI: <http://localhost:8090>

#### 4.5.4 Development with Docker Toolbox URLs

Development URLs, for local development.

Frontend: <http://local.dockertoolbox.tiangolo.com>

Backend: <http://local.dockertoolbox.tiangolo.com/api/>

Automatic Interactive Docs (Swagger UI): <https://local.dockertoolbox.tiangolo.com/docs>

Automatic Alternative Docs (ReDoc): <https://local.dockertoolbox.tiangolo.com/redoc>

PGAdmin: <http://local.dockertoolbox.tiangolo.com:5050>

Flower: <http://local.dockertoolbox.tiangolo.com:5555>

Traefik UI: <http://local.dockertoolbox.tiangolo.com:8090>

#### 4.5.5 Development with a custom IP URLs

Development URLs, for local development.

Frontend: <http://dev.arl.com>

Backend: <http://dev.arl.com/api/>

Automatic Interactive Docs (Swagger UI): <https://dev.arl.com/docs>

Automatic Alternative Docs (ReDoc): <https://dev.arl.com/redoc>

PGAdmin: <http://dev.arl.com:5050>

Flower: <http://dev.arl.com:5555>

Traefik UI: <http://dev.arl.com:8090>

#### 4.5.6 Development in localhost with a custom domain URLs

Development URLs, for local development.

Frontend: <http://localhost.tiangolo.com>

Backend: <http://localhost.tiangolo.com/api/>

Automatic Interactive Docs (Swagger UI): <https://localhost.tiangolo.com/docs>

Automatic Alternative Docs (ReDoc): <https://localhost.tiangolo.com/redoc>

PGAdmin: <http://localhost.tiangolo.com:5050>

Flower: <http://localhost.tiangolo.com:5555>

Traefik UI: <http://localhost.tiangolo.com:8090>

## 4.6 Project generation and updating, or re-generating

This project was generated using <https://github.com/tiangolo/full-stack-fastapi-postgresql> with:

```
pip install cookiecutter
cookiecutter https://github.com/tiangolo/full-stack-fastapi-postgresql
```

You can check the variables used during generation in the file `cookiecutter-config-file.yml`.

You can generate the project again with the same configurations used the first time.

That would be useful if, for example, the project generator (`tiangolo/full-stack-fastapi-postgresql`) was updated and you wanted to integrate or review the changes.

You could generate a new project with the same configurations as this one in a parallel directory. And compare the differences between the two, without having to overwrite your current code but being able to use the same variables used for your current project.

To achieve that, the generated project includes the file `cookiecutter-config-file.yml` with the current variables used.

You can use that file while generating a new project to reuse all those variables.

For example, run:

```
$ cookiecutter --config-file ./cookiecutter-config-file.yml --output-dir ../project-
→copy https://github.com/tiangolo/full-stack-fastapi-postgresql
```

That will use the file `cookiecutter-config-file.yml` in the current directory (in this project) to generate a new project inside a sibling directory `project-copy`.

## CHANGELOG

### 5.1 (unreleased)

- [GA] Bumped version to 6.0.6. [kingspp]
- Fix default view. [kingspp]
- Fix typos. [kingspp]
- Fix typos. [kingspp]
- Fix typos. [kingspp]
- Fix typos. [kingspp]
- Fix typos. [kingspp]
- Support for sorted errors. [kingspp]
- Fix typo. [kingspp]
- Support for view functionality. [kingspp]
- Fix typo. [kingspp]
- Support for tags view in record download. [kingspp]
- Fix issues with theming. [kingspp]
- Support for record downloads. [kingspp]
- Fix response cleanup. [kingspp]
- Fix issue with missing sheets tags. [kingspp]
- Sorted backend response. [kingspp]
- Fix typo. [kingspp]
- Support for auto fetch downloads after uploads. [kingspp]
- Update frontend. [kingspp]
- Update schema. [kingspp]
- Fix bug in naming. [kingspp]
- Support for backend update. [kingspp]
- Support for frontend services. [kingspp]
- Update backend schema. [kingspp]

- Fix backend. [kingspp]
- Fix issue with backend init. [kingspp]
- Prepare backend for digitized\_record table. [kingspp]
- Support for Download records. [kingspp]
- Fix typo. [kingspp]
- Fix typos. [kingspp]
- Fix typos. [kingspp]
- Fix typos. [kingspp]
- Fix issues with styling. [kingspp]
- Support for sheet dependency check. [kingspp]
- Fix typos. [kingspp]
- Fix typos. [kingspp]
- Fixed issues with Parsers. [kingspp]
- Fix typos. [kingspp]
- Fix issue with documentation. [kingspp]
- Stage 4 : Experiment Upload. [kingspp]
- Stage 2: Experimental Upload. [kingspp]
- Stage 2 : Experiment Upload. [kingspp]
- Initial experimental changes. [kingspp]
- Support for sticky footer. [kingspp]
- Fix typo. [kingspp]
- Fix typo. [kingspp]
- Fix typo. [kingspp]
- Update deploy. [kingspp]
- Fix line issue. [kingspp]
- Update login page. [kingspp]
- Fix issues with documentation view. [kingspp]
- Update project config. [kingspp]
- Fix typos. [kingspp]
- Fix style issues. [kingspp]
- Fixed issue with radar plot. [kingspp]
- Fixed issue with radar plot. [kingspp]
- Fix issues with documents and typo. [kingspp]
- Fix typos. [kingspp]
- Fix issues with tour. [kingspp]
- Update docs. [kingspp]

- Fix issue with dark mode. [kingspp]
- Update docs. [kingspp]
- [GA] Bumped version to 5.4.3. [kingspp]
- Fix issues with dark mode. [kingspp]
- Fix issues with dark mode. [kingspp]
- Fix issues with dark mode. [kingspp]
- Fix issues with dark mode. [kingspp]
- Fix issues with dark mode. [kingspp]
- Fix colors in documentation. [kingspp]
- Fix issues with search in dark mode. [kingspp]
- Fix issues with documentation. [kingspp]
- Support for documentation. [kingspp]
- [GA] Bumped version to 5.3.3. [kingspp]
- Fix typo. [kingspp]
- Support for displacement stats. [kingspp]
- Support for StrengthStats. [kingspp]
- Fix typos. [kingspp]
- Fix typo. [kingspp]
- Support for quadrant plot. [kingspp]
- Fix typo. [kingspp]
- Quadrant V1. [kingspp]
- Fix typos. [kingspp]
- Fix typo. [kingspp]
- Support for Experiment Coverage. [kingspp]
- Support for Radar Data and Experiment Coverage API. [kingspp]
- Fix onChange in search. [kingspp]
- Update styles. [kingspp]
- [GA] Bumped version to 5.1.8. [kingspp]
- Fix typos. [kingspp]
- Update table api. [kingspp]
- Update table api. [kingspp]
- Support for table component. [kingspp]
- Support for Data Table Hooks. [kingspp]
- Support for adhesive type distribution. [kingspp]
- Fix issue with Meomoization. [kingspp]
- Support for Numeric Cards. [kingspp]

- Support for AdhesiveType Queries. [kingspp]
- Support for AdhesiveType Queries. [kingspp]
- Support for AdhesiveType Queries. [kingspp]
- Support for AdhesiveType Queries. [kingspp]
- Support for AdhesiveType Queries. [kingspp]
- Support for AdhesiveType Queries. [kingspp]
- Support for AdhesiveType Queries. [kingspp]
- Support for AdhesiveType Queries. [kingspp]
- Support for AdhesiveType Queries. [kingspp]
- Support for AdhesiveType Queries. [kingspp]
- Support for AdhesiveType Queries. [kingspp]
- Support for AdhesiveType Queries. [kingspp]
- Fix issue with adhesiveStudioConstants Typo. [kingspp]
- Fix issue with adhesiveStudioConstants Typo. [kingspp]
- Fix issue with adhesiveStudioConstants Typo. [kingspp]
- Fix issue with adhesiveStudioConstants Typo. [kingspp]
- Fix issue with adhesiveStudioConstants Typo. [kingspp]
- Fix typo with adhesiveStudioConstants. [kingspp]
- Fix typo with adhesiveStudioConstants. [kingspp]
- Fix typo with adhesiveStudioConstants. [kingspp]
- Fix typo with adhesiveStudioConstants. [kingspp]
- Fix issues with tooltip. [kingspp]
- Fix typo. [kingspp]
- Fix typo. [kingspp]
- Update Coverage and Quadrant plots. [kingspp]
- Update config. [kingspp]
- Update styles. [kingspp]
- Fix typo. [kingspp]
- Fix issue with tooltip. [kingspp]
- Update notebook. [kingspp]
- Fix typo. [kingspp]
- Fixed issues with CE. [kingspp]
- Fixed issues with CE. [kingspp]
- Fix typos. [kingspp]
- Fix typos. [kingspp]

- Update Quadrant Plot. [kingspp]
- Update Adhesive Type. [kingspp]
- Support for href search. [kingspp]
- Update route. [kingspp]
- Fix typo. [kingspp]
- Fix typo. [kingspp]
- Fix typo 3. [kingspp]
- Fix typo 2. [kingspp]
- Fix typo 1. [kingspp]
- Adhesive Type V1. [kingspp]
- Adhesive Type V1. [kingspp]
- Adhesive Type V1. [kingspp]
- Adhesive Type V1. [kingspp]
- Adhesive Type V1. [kingspp]
- Adhesive Type V1. [kingspp]
- Bump version to 5.0. [kingspp]
- Bump version to 5.0. [kingspp]
- Moved File manager to redux. [kingspp]
- Fix typos. [kingspp]
- Fix typos. [kingspp]
- Fix typos. [kingspp]
- Moved DataVisualizer to Redux. [kingspp]
- Completed Strength and Displacement Cards. [kingspp]
- Fix issues with search css. [kingspp]
- Fix issue with out of range ce. [kingspp]
- Fix issue with rc-slider. [kingspp]
- Fix typo. [kingspp]
- Move default summary to redux. [kingspp]
- Move Strength Card to redux. [kingspp]
- Moved test conditions to redux. [kingspp]
- Move top and least adhesives to redux. [kingspp]
- Move search suggestions to redux. [kingspp]
- Support for dashboard reducer, action, services and constants. [kingspp]
- Support for dark mode. [kingspp]
- Support for search. [kingspp]
- Support with titles. [kingspp]

- Fix typo. [kingspp]
- Support for search. [kingspp]
- Fix typos. [kingspp]
- Support for least and top preferred adhesive. [kingspp]
- Support for groups. [kingspp]
- Support for Test Condition. [kingspp]
- Groups Stage 1. [kingspp]
- Fix typos. [kingspp]
- Fix typo. [kingspp]
- [GA] Bumped version to 4.2.8. [kingspp]
- Fix issue with tooltip hover. [kingspp]
- Fix issues with Data Wrangler. [kingspp]
- Fix typos. [kingspp]
- Fix typos. [kingspp]
- Convert Data Wrangler to hooks. [kingspp]
- Support to MIL PDF viewer. [kingspp]
- Support for MIL PRF PDF. [kingspp]
- Support for MIL-PRF download. [kingspp]
- [GA] Bumped version to 4.1.9. [kingspp]
- Update graphics. [kingspp]
- Fix typo. [kingspp]
- Dashboard API. [kingspp]
- Support for keyboard shortcuts. [kingspp]
- Support for dashboard. [kingspp]
- Fix shadow issue. [kingspp]
- Sort legend alphabetically. [kingspp]
- Order options alphabetically. [kingspp]
- Added tooltip support for legend. [kingspp]
- [GA] Bumped version to 4.0.9. [kingspp]
- Fix typo. [kingspp]
- Support for cluster statistics. [kingspp]
- Update data. [kingspp]
- Enable clusterBy for all groupBySelect options. [kingspp]
- Fix issues with static variable. [kingspp]
- [GA] Bumped version to 4.0.3. [kingspp]
- Support for confidence ellipse. [kingspp]

- [GA] Bumped version to 4.0.1. [kingspp]
- Fix version issue. [kingspp]
- [GA] Bumped version to 3.9.9. [kingspp]
- Support for count in legend. [kingspp]
- Fix typos. [kingspp]
- Fix typos. [kingspp]
- [GA] Bumped version to 3.9.5. [kingspp]
- Fix typos and issues. [kingspp]
- Fixed issues with query. [kingspp]
- Fix issue with left panel multiple renders. [kingspp]
- Intro to Redux Saga - v1. [kingspp]
- Fix typos. [kingspp]
- Change groupby select to redux state. [kingspp]
- Fix issue with rerender of left panel. [kingspp]
- Support for ranges and options. [kingspp]
- Fix typo. [kingspp]
- Support for slider ranges. [kingspp]
- Fix typo. [kingspp]
- Fixed issue with grouping filters. [kingspp]
- Fix issue with multi select and support for item numbers. [kingspp]
- Support for query order. [kingspp]
- Support for clear filters. [kingspp]
- Fix typos. [kingspp]
- Fix typo. [kingspp]
- Fix issue with panelConfig. [kingspp]
- Added legend name. [kingspp]
- [GA] Bumped version to 3.7.5. [kingspp]
- Fix typos. [kingspp]
- Fix issue with groups. [kingspp]
- Fix issue with point hover. [kingspp]
- Fix issues with scaling. [kingspp]
- Fix typos. [kingspp]
- Fix issue with scales. [kingspp]
- Fix issue with component chaining. [kingspp]
- Fix issues with Zoom Layer. [kingspp]
- Fix issue with legend. [kingspp]

- Fix issues with groups. [kingspp]
- Left Panel V1. [kingspp]
- Graph Panel v1. [kingspp]
- Update Work on Top Panel. [kingspp]
- Updated info. [kingspp]
- Top Panel V1 Changes. [kingspp]
- Merge branch 'master' of arcgit.wpi.edu:arl-adhesives/dashboard. [kingspp]
- Add new file. [Prathyush]
- Add new file. [Prathyush]
- Fix typos. [kingspp]
- Support for Graph Controls in LoadVsDisplacement. [kingspp]
- Merge remote-tracking branch 'upstream/master' [kingspp]
- Disable CI. [kingspp]
- Merge remote-tracking branch 'upstream/master' [kingspp]
- Fix jenkins. [kingspp]
- [GA] Bumped version to 3.5.0. [kingspp]
- Merge branch 'master' of arcgit.wpi.edu:arl-adhesives/dashboard. [kingspp]
- Update pyproject.toml. [Prathyush]
- Update Dockerfile. [Prathyush]
- Update docker-compose.yml. [Prathyush]
- Merge remote-tracking branch 'upstream/master' [kingspp]
- Create release script. [kingspp]
- [GA] Bumped version to 1.6.1. [kingspp]
- Update docs. [kingspp]
- Fix typos. [kingspp]
- Fix typos. [kingspp]
- Merge branch 'master' of arcgit.wpi.edu:arl-commons/dashboard. [kingspp]
- Update .gitlab-ci.yml. [Prathyush]
- Deployment Configs. [kingspp]
- Fix issue with NaN. [kingspp]
- Merge remote-tracking branch 'upstream/master' [kingspp]
- Update .gitlab-ci.yml. [Prathyush]
- Update .gitlab-ci.yml. [Prathyush]
- Update .gitlab-ci.yml. [Prathyush]
- Support for backend versioning. [kingspp]
- Support for documentation. [kingspp]

- Support for pdf viewer. [kingspp]
- Support for force sync. [kingspp]
- Merge branch 'master' of arcgit.wpi.edu:arl-adhesives/dashboard. [kingspp]
- [GA] Bumped version to 3.2.5. [psparvatharaju]
- [GA] Bumped version to 3.1.8. [psparvatharaju]
- Updated data. [kingspp]
- Merge remote-tracking branch 'upstream/master' [kingspp]
- Fix issues with file handling. [kingspp]
- Fix issue with file handline. [kingspp]
- Merge branch 'master' of arcgit.wpi.edu:arl-adhesives/dashboard. [kingspp]
- [GA] Bumped version to 3.1.5. [psparvatharaju]
- Fix typos. [kingspp]
- Fix issue with toast notification behaviour. [kingspp]
- Merge branch 'master' of arcgit.wpi.edu:arl-adhesives/dashboard. [kingspp]
- [GA] Bumped version to 3.1.2. [psparvatharaju]
- Fix issue with breadcumb. [kingspp]
- Fix issue with null title. [kingspp]
- Support for backup visualizations from Spring 2019. [kingspp]
- Fix issue with sed. [psparvatharaju]
- [GA] Bumped version to 3.1.0. [psparvatharaju]
- Fix issue with data path. [kingspp]
- Fix typos. [kingspp]
- Merge branch 'master' of arcgit.wpi.edu:arl-adhesives/dashboard. [kingspp]
- [GA] Bumped version to 3.0.1. [psparvatharaju]
- Fix issue with wss for https. [kingspp]
- Fix typo. [kingspp]
- Added favicon. [kingspp]
- Fix typos. [kingspp]
- Fix issues with stabas merge. [kingspp]
- Support for doe - frontend. [kingspp]
- Fix typo. [kingspp]
- Updated data sources - frontend. [kingspp]
- Updated data sources. [kingspp]
- Support for doe. [kingspp]
- Merge branch 'master' of arcgit.wpi.edu:arl-adhesives/dashboard. [kingspp]
- [GA] Bumped version to 2.9.3. [psparvatharaju]

- Fix project config. [kingspp]
- Update deploy. [kingspp]
- Support for load vs displacement and mode of failure services. [kingspp]
- Merge commons. [kingspp]
- Fix typos. [kingspp]
- Refactored wrangler. [kingspp]
- Backup theme. [kingspp]
- Fix issue with data viz. [kingspp]
- Fix typos. [kingspp]
- Fix issue with prod. [kingspp]
- [GA] Bumped version to 1.3.9. [kingspp]
- [GA] Bumped version to 1.3.8. [kingspp]
- [GA] Bumped version to 1.3.7. [kingspp]
- [GA] Bumped version to 1.3.6. [kingspp]
- [GA] Bumped version to 1.3.5. [kingspp]
- [GA] Bumped version to 1.3.4. [kingspp]
- [GA] Bumped version to 1.3.3. [kingspp]
- [GA] Bumped version to. [kingspp]
- [GA] Bumped version to. [kingspp]
- [GA] Bumped version to. [kingspp]
- [GA] Bumped version to. [kingspp]
- [GA] Bumped version to. [kingspp]
- [GA] Bumped version to. [kingspp]
- Fix typos. [kingspp]
- Fix issues with tooltip. [kingspp]
- Fix typos. [kingspp]
- Fix issues with dark mode and other changes. [kingspp]
- Fix issue with dark mode. [kingspp]
- Refactor changes. [kingspp]
- Merge branch 'master' of arcgit.wpi.edu:arl-adhesives/dashboard. [kingspp]
- Merge branch 'master' of arcgit.wpi.edu:arl-adhesives/dashboard. [psparvatharaju]
- Removed lock file. [psparvatharaju]
- Upated gitignore. [psparvatharaju]
- Fix issues with dependencies py. [psparvatharaju]
- Fix issues with load vs displacement. [kingspp]
- Update version. [kingspp]

- Fix issues with summary. [kingspp]
- Fix db name typo. [psparvatharaju]
- Fix dotenv. [kingspp]
- Fix merge issues. [psparvatharaju]
- Merge remote-tracking branch 'upstream/master' [kingspp]
- Fix issue with localhost hardcoding. [kingspp]
- Support for dynamic file summary. [kingspp]
- Fix typo. [kingspp]
- Fix issue with dark theme in summary. [kingspp]
- Updated frontend docker issue. [kingspp]
- Fix issues with merge. [kingspp]
- Merge remote-tracking branch 'upstream/master' [kingspp]
- Support for file manager tour. [kingspp]
- Support for data visualizer dynamic data load. [kingspp]
- Support for data visualizer dynamic data loading. [kingspp]
- Update FileDetailsComponent. [kingspp]
- Fix typos. [kingspp]
- Support for user config based data wrangler. [kingspp]
- Support for file load from file manager to data wrangler. [kingspp]
- File Broser button. [kingspp]
- Fix issue with scross x width style. [kingspp]
- Support for dark theme react-chart-editor. [kingspp]
- Added unicode. [kingspp]
- Fix issues with file delete and render. [kingspp]
- Fix issues with file delete. [kingspp]
- Support for file delete. [kingspp]
- Support for delete api. [kingspp]
- Fix display items. [kingspp]
- Actions design. [kingspp]
- Support for file display in tables. [kingspp]
- Udpate files. [kingspp]
- Support for file push and get file apis. [kingspp]
- Fix issues with alembic. [kingspp]
- Upgrade Data Visualizer. [kingspp]
- Support for server based csv file fetch. [kingspp]
- Fix issues with file upload. [kingspp]

- Support for file upload backend. [kingspp]
- File drop component frontend. [kingspp]
- Support for DB based dark mode. [kingspp]
- Support for user prefs. [kingspp]
- Fix typos. [kingspp]
- Fix issues with theme on logout. [kingspp]
- Support for Enabling dark mode based on user prefs. [kingspp]
- Fix issues with dark mode 2. [kingspp]
- Fix typos. [kingspp]
- Fix issues with dark mode. [kingspp]
- Fix issues with icons and other merge issues. [kingspp]
- Merge remote-tracking branch 'origin/structure\_refactor' [kingspp]
- Reorganized file structure, renamed files, relative -> absolute imports. [Joseph Caltabiano]

This commit contains changes in 3 areas: First, the structure of the directories and files has been changed. the new structure attempts to follow best practices as outlined in the React and Redux docs. The file names have also been changed to be more descriptive. This mainly involved changing components from */MyComp/index.js* to */MyComp/MyComp.jsx*. Third, the use of relative imports was addressed. This refactor caused many problems with relative import paths no longer working. This highlights the benefits of using absolute imports. A *jsconfig.js* file was added to define the root directory as *src/*, allowing for imports to be written relative to *src/*. Moving files will no longer require changing the file's imports.

- Fix issues with fower versioning and commented frontend for now. [kingspp]
- Fix issue with encoded url. [kingspp]
- Support for documentation. [kingspp]
- Fix issues with documentation. [kingspp]
- Fix typo. [kingspp]
- Init scripts. [kingspp]
- Updated Readme. [kingspp]
- Updated gitignore. [kingspp]
- Updated tour for data wrangler. [kingspp]
- Support for load vs displacement. [kingspp]
- Fix issue with flower. [kingspp]
- Support for adhesives dynamic plot. [kingspp]
- Temp changes in docker. [kingspp]
- Changes in env. [kingspp]
- Support for tier\_data. [kingspp]
- Updated git ignore. [kingspp]
- Revert "fix a bug when displaying the sheet errors" [kingspp]

This reverts commit 7e3762cf217526a9fad99ac696c0187846b1d1c0.

- Fix a bug when displaying the sheet errors. [daojun]
- Fix bug on the error msg feature. [daojun]
- Fix a styling bug. [daojun]
- Add css style to the error msg. [daojun]
- Add error msg css file. [daojun]
- Styling the error msg. [daojun]
- Improve on the error msg feature: 1. add the missing error and format error number 2. refactor the code to make it cleaner. [daojun]
- Change the frontend code: 1. Distinguish errors of each sheet 2. add a number before each error line. 3. Add collapse function of each sheet. [daojun]
- Improve on the error msg: 1. Distinguish errors of each sheet 2. add a number before each error line. 3. Add collapse function of each sheet. [daojun]
- Fix frontend module import bug. [daojun]
- Add comments. [daojun]
- Change the Readme.file and the Homepage. [daojun]
- Change the backend code: pass the error object as a dictionary. [daojun]
- Fix frontend module import bug. [daojun]
- Add comments. [daojun]
- Change the Readme.file and the Homepage. [daojun]

## 5.2 1.0 (2021-07-15)

- Server config. [psparvatharaju]
- Merge remote-tracking branch 'upstream/master' [kingspp]  
# Conflicts: # backend/app/pyproject.toml # docker-compose.override.yml # frontend/src/App/App.jsx # frontend/src/config.js # scripts/deploy.sh
- Merge remote-tracking branch 'origin/master' [kingspp]
- Add adhesives performance chart demo/prototype. [Joseph Caltabiano]
- Merge remote-tracking branch 'upstream/master' [kingspp]  
# Conflicts: # backend/app/pyproject.toml # docker-compose.override.yml # frontend/src/App/App.jsx # frontend/src/config.js # scripts/deploy.sh
- Update tour for visualization. [kingspp]
- Support for alt text for icons. [kingspp]
- Fix issue with pyproject. [kingspp]
- Load env vars from .env. [kingspp]
- Added support for auto-versioning. [kingspp]
- Fix issues with project specific data loading. [kingspp]
- Update tour for data visualizer. [kingspp]

- Update brand href. [kingspp]
- Added support for platform tour. [kingspp]
- Added version. [kingspp]
- Support for prod and dev urls. [kingspp]
- Added docker ignore. [kingspp]
- Fix terser issue. [kingspp]
- Updated compose. [kingspp]
- Merge from commons. [kingspp]
- Update networks. [kingspp]
- Update env. [kingspp]
- Updated env. [kingspp]
- Updated env. [kingspp]
- Adhesive specific changes. [kingspp]
- Merge remote-tracking branch 'upstream/master' [kingspp]
- Disabled frontend containers for now. [kingspp]
- Added index file. [kingspp]
- Merge remote-tracking branch 'upstream/master' [kingspp]
- Fix issue. [kingspp]
- Merge branch 'master' of arcgit.wpi.edu:GQP\_adhesive\_f19/dashboard. [kingspp]
- Final template. [edkazcarlson]
- Added appropriate error handling. [edkazcarlson]
- Merge branch 'master' into presentation\_branch. [edkazcarlson]
- Working presentation version. [edkazcarlson]
- Template.xlsx. [edkazcarlson]
- Merged and set up everything. [edkazcarlson]
- Merged. [edkazcarlson]
- Merged code. [edkazcarlson]
- Unit connected to database. [mahajanrevant]
- Sample Bondline connected to database without nominal thickness. [mahajanrevant]
- Crack Extension Sample fabrication can be added to the backend. [mahajanrevant]
- Adhesive connected to the database. [mahajanrevant]
- Added a method to create company. [mahajanrevant]
- Crack Extension Samples added to database. [mahajanrevant]
- All but unit and sample bondline are finished. [edkazcarlson]
- Setting up common platform. [mahajanrevant]
- Changes to make single lap experiments. [edkazcarlson]

- Crack Extension data being pushed to database. This is a working example. [mahajanrevant]
- Updating with Tai's changes. [mahajanrevant]
- Uploaded filledsheet2. [edkazcarlson]
- Giving code to rev. [edkazcarlson]
- Attempted to pickle read. [edkazcarlson]
- All the test cases work. [mahajanrevant]
- Starting send to database upload functionality. [mahajanrevant]
- Merge remote-tracking branch 'origin/backend' into mergedWithBackend. [edkazcarlson]
- Machine Readout Relationship defined and tested. [mahajanrevant]
- Single Lap Experiment relationships defined and tested. [mahajanrevant]
- Sample Bondline Relationships define and tested. [mahajanrevant]
- Crack Ext Sample Relationships defined and tested. [mahajanrevant]
- Crack Extension Experiment relationships defined and tested. [mahajanrevant]
- All instrument relationships finished and tested. [mahajanrevant]
- Relationship between Adhesive and Sample Bondline tested. [mahajanrevant]
- Dummy model creation method created for each of the models in app.tests.util. [mahajanrevant]
- Company and Adhesive relationship works. [mahajanrevant]
- Uncommenting test cases. [mahajanrevant]
- Crack Extension and Instrument relationship works. [mahajanrevant]
- Instrument and Single Lap Experiment Many to Many relation tested. [mahajanrevant]
- Instrument Single Lap Experiment works without relationships. [mahajanrevant]
- Crack Extension Experiment works without relationships. [mahajanrevant]
- Crack extension sample works without relationship. [mahajanrevant]
- Single Lap Sample works without relationships. [mahajanrevant]
- Single Lap Experiment works without relationships. [mahajanrevant]
- Sample Bondline works without relationships. [mahajanrevant]
- Sample Bondline works. [mahajanrevant]
- MachineReadout works without relationships. [mahajanrevant]
- Adhesive works without relationships. [mahajanrevant]
- Company table added without relationships. [mahajanrevant]
- Tables can be created without Alembic. For now just importing all models. [mahajanrevant]
- Session is unable to find the Unit table. [mahajanrevant]
- Added aiofiles for proper file sending from api. [edkazcarlson]
- About to add aiofiles package to backend. [edkazcarlson]
- Readded code that was removed in previous merge. [edkazcarlson]

- Merge remote-tracking branch 'origin/parse\_model\_add' into parse\_model\_add. [Tai Zhou]  

```
Conflicts: # backend/app/app/parse_excel/parse/ParseDCBCrackLen.py # back-
end/app/app/parse_excel/parse/ParseFab.py # backend/app/app/parse_excel/parse/ParseStrength.py # back-
end/app/app/parse_excel/parse/ParseThickness.py
```
- Merge remote-tracking branch 'origin/parse\_model\_add' into parse\_model\_add. [Tai Zhou]  

```
Conflicts: # backend/app/app/parse_excel/parse/ParseDCBCrackLen.py # back-
end/app/app/parse_excel/parse/ParseFab.py # backend/app/app/parse_excel/parse/ParseStrength.py # back-
end/app/app/parse_excel/parse/ParseThickness.py
```
- Merge remote-tracking branch 'origin/parse\_model\_add' into parse\_model\_add. [Tai Zhou]  

```
Conflicts: # backend/app/app/parse_excel/parse/ParseDCBCrackLen.py # back-
end/app/app/parse_excel/parse/ParseFab.py # backend/app/app/parse_excel/parse/ParseStrength.py # back-
end/app/app/parse_excel/parse/ParseThickness.py
```
- Modified unit test. [Tai Zhou]
- Fixed optional cells being required. [edkazcarlson]
- Error catching for missing sheets, need to properly handle default dropdown vs other cells. [edkazcarlson]
- All errors I believe are showing. [edkazcarlson]
- Will not crash anymore, need to check actual error msgs now. [edkazcarlson]
- Date parsing tool. [edkazcarlson]
- Erroring on date parsing, fixed one. [edkazcarlson]
- Unified error handling. [edkazcarlson]
- Backup. [edkazcarlson]
- Allow for correct files to be uploaded. [edkazcarlson]
- Working state, need to test and clean up and document. [edkazcarlson]
- Success case working. [edkazcarlson]
- Updating error handling for actual use. [edkazcarlson]
- Set up most of infrastructure, need to fix xlrd importing to run and test. [edkazcarlson]
- Merge branch 'parse\_model\_add' of [https://arcgit.wpi.edu/GQP\\_adhesive\\_f19/dashboard](https://arcgit.wpi.edu/GQP_adhesive_f19/dashboard) into parse\_model\_add. [edkazcarlson]
- Modified unit test. [Tai Zhou]
- Merge branch 'uploadMockupBranch' into parse\_model\_add. [edkazcarlson]
- Factored out into reusable component. [edkazcarlson]
- Drag and drop files complete will factor out to reuse. [edkazcarlson]
- Upload error msgs and validation. [edkazcarlson]
- Preliminary sending xlsx to backend. [edkazcarlson]
- Backup. [edkazcarlson]
- Temprorary push to move to other OS. [edkazcarlson]
- Add excel parse model. [Tai Zhou]
- Add excel parse model. [Tai Zhou]

- Add new directory. [Tai Zhou]
- Merge remote-tracking branch 'upstream/master' [kingspp]
- Add missing files for frontend. [kingspp]
- Enable Frontend docker services. [kingspp]
- Fix issues with Frontend DockerFile. [kingspp]
- Push local-npm-package. [kingspp]
- Add support local-npm-package. [kingspp]
- Fix Switches partial typo. [kingspp]
- Fix issues with quantiles. [kingspp]
- Refactor Datawrangler into components. [kingspp]
- Fix issue with frontend api hardcoding. [kingspp]
- Merge remote-tracking branch 'upstream/master' [kingspp]
- Fix total values bug. [kingspp]
- Fix typo. [kingspp]
- Merge remote-tracking branch 'upstream/master' [kingspp]  
# Conflicts: # frontend/README.md # frontend/src/routes.js
- Fix typo. [kingspp]
- Removed socket dependency. [kingspp]
- Refactored wrangler. [kingspp]
- Added stabas dataset. [kingspp]
- Added support for data summary. [kingspp]
- Added websocket frontend. [kingspp]
- Added support for websocket. [kingspp]
- Updated readme. [kingspp]
- Remove poetry lock. [edkazcarlson]
- Fixes to be able to run and update readme.md. [edkazcarlson]
- Updated README. [kingspp]
- Updated README. [kingspp]
- Update README.md. [Prathyush]
- Updated README. [kingspp]
- Fix issues with docker files. [kingspp]
- Updated docker file. [kingspp]
- Updated test file. [kingspp]
- Added missing files. [kingspp]
- Added environment file. [kingspp]
- Updated .gitignore. [kingspp]

- Updated app.jsx. [kingspp]
- Updated readme. [kingspp]
- Updated .gitignore. [kingspp]
- Updated backend. [kingspp]
- Removed \_\_pycache\_\_ [kingspp]
- Updated project config. [kingspp]
- Updated readme. [kingspp]
- Updated fonts. [kingspp]
- Updated backend boilerplate. [kingspp]
- Merge remote-tracking branch 'origin/master' [kingspp]
- Update README.md. [Prathyush]
- Integrated registration with backend. [kingspp]
- Integrated login service with backend. [kingspp]
- Fix issues with delays. [kingspp]
- Fixed issues with logo colors. [kingspp]
- Fix issues with login and registration. [kingspp]
- Added support for Breadcrumbs. [kingspp]
- Fix issues with full screen scroll mode. [kingspp]
- Fix datatable issue. [kingspp]
- Fixed issues with brand title. [kingspp]
- Fixed issues with routing. [kingspp]
- Use of hooks and added support for toast alerts. [kingspp]
- Added support for registration. [kingspp]
- Fix issues with dark mode in datatables. [kingspp]
- Added support for dark mode. [kingspp]
- Init Dark Mode. [kingspp]
- Added support for data wrangler. [kingspp]
- Added support for navbar. [kingspp]
- Fix issue with login and added sidebar. [kingspp]
- Init Project. [kingspp]

## LIBRARY STATS

|      |                                                                           |
|------|---------------------------------------------------------------------------|
| cloc | github.com/AIDanial/cloc v 1.90 T=1.38 s (333.0 files/s, 68918.2 lines/s) |
|      |                                                                           |

| Language         | files | blank | comment | code  |
|------------------|-------|-------|---------|-------|
| JSON             | 12    | 2     | 0       | 25508 |
| JavaScript       | 68    | 395   | 305     | 19559 |
| Sass             | 38    | 1558  | 59      | 8568  |
| CSV              | 22    | 552   | 0       | 8342  |
| Python           | 157   | 1479  | 709     | 7701  |
| JSX              | 75    | 857   | 654     | 6836  |
| CSS              | 6     | 456   | 58      | 1933  |
| Markdown         | 10    | 742   | 0       | 1295  |
| YAML             | 6     | 30    | 253     | 567   |
| Bourne Shell     | 25    | 163   | 219     | 561   |
| XML              | 17    | 5     | 0       | 458   |
| SQL              | 1     | 373   | 469     | 445   |
| SVG              | 4     | 492   | 4       | 371   |
| Jupyter Notebook | 4     | 0     | 1695    | 366   |
| TOML             | 2     | 40    | 0       | 271   |
| LESS             | 1     | 39    | 0       | 267   |
| HTML             | 3     | 37    | 20      | 257   |
| INI              | 2     | 16    | 0       | 59    |
| Dockerfile       | 4     | 31    | 17      | 50    |
| Mako             | 1     | 7     | 0       | 17    |
| TypeScript       | 1     | 3     | 0       | 8     |
| reStructuredText | 1     | 7     | 20      | 7     |
| _____            | _____ | _____ | _____   | _____ |
| SUM:             | 460   | 7284  | 4482    | 83446 |

## ABOUT

**Author** Prathyush SP

**Version** 'beta 6.0.6' on 02/11/2021

**Github** <https://arccgit.wpi.edu/ar1-commons/dashboard>

## API REFERENCE

### 8.1 login

**POST /api/v1/login/access-token**  
**Login Access Token**

OAuth2 compatible token login, get an access token for future requests

**Status Codes**

- 200 OK – Successful Response
- 422 Unprocessable Entity – Validation Error

**POST /api/v1/login/test-token**  
**Test Token**

Test access token

**Status Codes**

- 200 OK – Successful Response

**POST /api/v1/password-recovery/{email}**  
**Recover Password**

Password Recovery

**Parameters**

- **email** (*string*) –

**Status Codes**

- 200 OK – Successful Response
- 422 Unprocessable Entity – Validation Error

**POST /api/v1/reset-password/**  
**Reset Password**

Reset password

**Status Codes**

- 200 OK – Successful Response
- 422 Unprocessable Entity – Validation Error

## 8.2 users

**GET /api/v1/users/**  
**Read Users**

Retrieve users.

**Query Parameters**

- **skip**(*integer*) –
- **limit**(*integer*) –

**Status Codes**

- **200 OK** – Successful Response
- **422 Unprocessable Entity** – Validation Error

**POST /api/v1/users/**  
**Create User**

Create new user.

**Status Codes**

- **200 OK** – Successful Response
- **422 Unprocessable Entity** – Validation Error

**GET /api/v1/users/me**  
**Read User Me**

Get current user.

**Status Codes**

- **200 OK** – Successful Response

**PUT /api/v1/users/me**  
**Update User Me**

Update own user.

**Status Codes**

- **200 OK** – Successful Response
- **422 Unprocessable Entity** – Validation Error

**POST /api/v1/users/open**  
**Create User Open**

Create new user without the need to be logged in.

**Status Codes**

- **200 OK** – Successful Response
- **422 Unprocessable Entity** – Validation Error

**GET /api/v1/users/{user\_id}**  
**Read User By Id**

Get a specific user by id.

**Parameters**

- **user\_id**(*integer*) –

**Status Codes**

- 200 OK – Successful Response
- 422 Unprocessable Entity – Validation Error

**PUT /api/v1/users/{user\_id}**

**Update User**

Update a user.

**Parameters**

- **user\_id**(*integer*) –

**Status Codes**

- 200 OK – Successful Response
- 422 Unprocessable Entity – Validation Error

## 8.3 utils

**POST /api/v1/utils/test-celery/**

**Test Celery**

Test Celery worker.

**Status Codes**

- 201 Created – Successful Response
- 422 Unprocessable Entity – Validation Error

**POST /api/v1/utils/test-email/**

**Test Email**

Test emails.

**Query Parameters**

- **email\_to**(*string*) – (Required)

**Status Codes**

- 201 Created – Successful Response
- 422 Unprocessable Entity – Validation Error

## 8.4 items

**GET /api/v1/items/**

**Read Items**

Retrieve items.

**Query Parameters**

- **skip**(*integer*) –
- **limit**(*integer*) –

**Status Codes**

- 200 OK – Successful Response
- 422 Unprocessable Entity – Validation Error

**POST /api/v1/items/**

**Create Item**

Create new item.

**Status Codes**

- 200 OK – Successful Response
- 422 Unprocessable Entity – Validation Error

**GET /api/v1/items/{id}**

**Read Item**

Get item by ID.

**Parameters**

- **id**(*integer*) –

**Status Codes**

- 200 OK – Successful Response
- 422 Unprocessable Entity – Validation Error

**PUT /api/v1/items/{id}**

**Update Item**

Update an item.

**Parameters**

- **id**(*integer*) –

**Status Codes**

- 200 OK – Successful Response
- 422 Unprocessable Entity – Validation Error

**DELETE /api/v1/items/{id}**

**Delete Item**

Delete an item.

**Parameters**

- **id**(*integer*) –

**Status Codes**

- 200 OK – Successful Response
- 422 Unprocessable Entity – Validation Error

## 8.5 adhesives

**POST /api/v1/adhesives/createAdhesive**

Create Adhesive

Create new adhesive.

**Status Codes**

- 200 OK – Successful Response
- 422 Unprocessable Entity – Validation Error

**GET /api/v1/adhesives/templateXlsx**

Templatexlsx

Sends the template xlsx file to the suer

**Status Codes**

- 200 OK – Successful Response

## 8.6 data\_wrangler

**GET /api/v1/data\_wrangler/summary**

Summary

Retrieve items.

**Query Parameters**

- **data** (*string*) – (Required)

**Status Codes**

- 200 OK – Successful Response
- 422 Unprocessable Entity – Validation Error

## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

## HTTP ROUTING TABLE

### /api

GET /api/v1/adhesives/templateXlsx, 47  
GET /api/v1/data\_wrangler/summary, 47  
GET /api/v1/items/, 45  
GET /api/v1/items/{id}, 46  
GET /api/v1/users/, 44  
GET /api/v1/users/me, 44  
GET /api/v1/users/{user\_id}, 44  
POST /api/v1/adhesives/createAdhesive, 47  
POST /api/v1/items/, 46  
POST /api/v1/login/access-token, 43  
POST /api/v1/login/test-token, 43  
POST /api/v1/password-recovery/{email}, 43  
POST /api/v1/reset-password/, 43  
POST /api/v1/users/, 44  
POST /api/v1/users/open, 44  
POST /api/v1/utls/test-celery/, 45  
POST /api/v1/utls/test-email/, 45  
PUT /api/v1/items/{id}, 46  
PUT /api/v1/users/me, 44  
PUT /api/v1/users/{user\_id}, 45  
DELETE /api/v1/items/{id}, 46

## **Appendix C. Installation Instructions**

---

## Back-end Requirements

- Docker
- Docker Compose
- Poetry for Python package and environment management

## Front-end Requirements

- Node.js (with npm)
- Git Clone: <https://arccgit.wpi.edu/arl-adhesives/dashboard>
- Switch branch to “ms-identity”
- Start Docker desktop
- From the dashboard directory run “docker-compose up” in the terminal

This command will check if the images are built in the docker; if not it will automatically create the images and then start the containers.

1. Check if the containers are running in Docker desktop. If any container is stopping/not starting, check the error logs after running the above command.
2. If all containers started successfully, go to “localhost:8200” to navigate to the dashboard.

Note: While working on a specific image (e.g., back end, front end) no need to rebuild all the containers. Just run “docker-compose build [image\_name]” and then “docker-compose up”.

## List of Symbols, Abbreviations, and Acronyms

---

|        |                                                 |
|--------|-------------------------------------------------|
| AI/ML  | artificial intelligence/machine learning        |
| ARL    | Army Research Laboratory                        |
| CAAP   | Context-Aware Agile Platform                    |
| CSV    | comma-separated values                          |
| DCB    | double cantilever beam                          |
| DEVCOM | US Army Combat Capabilities Development Command |
| DoD    | Department of Defense                           |
| DOE    | Design of Experiment                            |
| GQP    | Graduate Qualifying Project                     |
| MSE    | Mean Squared Error                              |
| PDF    | Portable Document Format                        |
| PNG    | Portable Network Graphic                        |
| QPL    | Qualified Products List                         |
| SLJ    | single-lap-joint                                |
| WPI    | Worcester Polytechnic Institute                 |

1 DEFENSE TECHNICAL  
(PDF) INFORMATION CTR  
DTIC OCA

1 DEVCOM ARL  
(PDF) FCDD RLB CI  
TECH LIB

1 DEVCOM ARL  
(PDF) FCDD RLA MC  
R JENSEN