Developing Features and Maintenance Improvements for the E-TRIALS Platform

May Dong
Michael Oliveira

In collaboration with:
Kristine Guan (IQP)
Sharon Wu (IQP)

Submitted to:
Dr. Neil Heffernan

This report represents the work of one or more WPI undergraduate students submitted to the faculty as evidence of completion of a degree requirement. WPI routinely publishes these reports on the web without editorial or peer review.
# Table of Contents

Abstract 3  
1. Introduction 4  
2. Background 5  
  2.1 ASSISTments 5  
  2.2 E-TRIALS 5  
  2.3 The History of E-TRIALS 6  
3. Methodology 11  
  3.1 Process 11  
  3.2 Technical 11  
    3.2.1 Tools 11  
    3.2.2 Techniques 12  
4. Results 14  
  4.1 Common Wrong Answers 14  
  4.2 Migration 16  
  4.3 Maintenance 17  
    4.3.1 Bug 17  
    4.3.2 Tech Debt 19  
    4.3.3 UI/UX 20  
5. Future Work 26  
6. References 27  
7. Appendix 28  
  Appendix A: HomeExperimentCard.vue 28  
  Appendix B: StudyManagerImpl.java 28
Abstract

The ASSISTments Foundation launched E-TRIALS (EdTech Research Infrastructure to Advance Learning Science) in late 2022 to enable researchers to more easily conduct educational studies comparing different teaching methods and supports. The students in this project worked in an Agile environment alongside full time ASSISTments employees to develop a new study type, to assist with migration efforts, and to make improvements to the existing software.
1. Introduction

Online learning has become prevalent in the wake of the Covid-19 pandemic in 2020. Teachers had to rapidly adapt by learning new strategies and technologies to accommodate a remote learning style. Though many schools and universities have shifted back to in-person learning, many still follow a hybrid model and incorporate online learning into their curricula. For this reason, it has become increasingly important for instructors to understand how best to reach their students in an online learning model: what aids their students, and what hinders their learning. Research is key to achieve this end.

The ASSISTments Foundation supports researchers, educators, and software engineers who wish to "improv[e] student learning through responsive online technology" (About the ASSISTments Foundation). ASSISTments offers free learning resources, and researchers are able to conduct studies using educational content available on ASSISTments.

Due to the nature of software, design and structure within ASSISTments needed to be iterated upon. Various features felt unintuitive for users, and designs were outdated from today's standards. Several students undertook Major Qualifying Projects (MQP) in order to aid and resolve these issues. In one MQP, the E-TRIALS project was named, and basic designs and prototypes were created (Krichevsky, Spinelli, 2020). Further student work, Continuing the Development of E-TRIALS (McCarthy, 2021), looked to improve upon the original designs and had the goal of preparing a minimum viable product. The E-TRIALS product was finally launched in February 2022 with the help of another student project: Preparing the E-TRIALS Minimum Viable Product for Release (Philippo, Spofford, 2022).

These former MQPs laid the groundwork for the E-TRIALS product, which is now available and in use by researchers. Now, the focus on E-TRIALS has shifted towards improvements and refinements to the product.
2. Background

2.1 ASSISTments

ASSISTments is a non-profit company offering a free online learning platform to grade school teachers. Teachers can select from over 200,000 problems to assign to their students using their existing learning platform, including Google Classroom and Canvas. Student answers are automatically graded, and a detailed report of the class’s performance is made available to the teacher. Teachers can use this report to identify concepts that need review, and students who need extra help. ASSISTments can also provide students help when they get stuck or get a question wrong to enhance their learning experience. This help comes in the form of guided hints which help students to understand the problem and post-answer feedback explaining why the answer is wrong, and how the correct answer could be found in the future. ASSISTments is also used to facilitate research in education based studies using real student data collected on the platform. This is done using the E-TRIALS system.

2.2 E-TRIALS

E-TRIALS (Ed-Tech Research Infrastructure to Advance Learning Science) is a tool integrated with the ASSISTments platform that allows researchers to conduct large scale experiments easily and quickly. Education-based studies using this system focus on determining the best way to help students learn in order to help educators provide better support to their students. Researchers can take advantage of the existing ASSISTments userbase rather than having to individually approach teachers or schools to incorporate them into a study. This allows researchers to spend more time and money focusing on the quality of their study rather than the logistics of it.

E-TRIALS contains 4 types of studies for researchers to choose from. “Support Comparison” studies allow researchers to pick a problem set and create two or more conditions, each containing a support for each problem. These supports can come in the form of hints that guide the user towards the solution or a direct explanation that gives the answer of the problem and explains how it can be found. Students are randomly placed into one of the conditions when they are assigned this problem set, and may see these experimental supports when asking for help. “Single Support” studies only allow the researcher to create one condition, and focus on comparing it to the “best so far” supports the platform uses by default. “Problem Varied” studies
allow researchers to vary the content of the problems students will receive rather than varying the support content. "Common Wrong Answers: studies allow researchers to author specific answer feedback for the wrong answers that are most common for students to give for every problem, allowing them to tackle the specific gaps in the student’s knowledge.

Once a study has concluded, researchers are given a detailed report on how students in each condition performed overall. Using the scale of the ASSISTments platform, the researcher can statistically determine if one condition was more effective at helping students than other conditions.

2.3 The History of E-TRIALS

The potential for creating a product like E-TRIALS was discovered in the ASSISTments 1.0 builder. This tool is used by teachers to create new problems and problem sets to make them available to the ASSISTments platform at large. At the time, the builder was also being used by researchers to create much more basic versions of the studies that can be built on the current release of E-TRIALS. This tool, however, was unable to meet the needs of researchers, as it was not designed primarily for creating research studies and made the process overly complex to do so. Including the study creation logic in this tool also made it harder for teachers to develop problems as they had to ignore the complex research tools when authoring for the platform. The design was overall overwhelming and complex in its design, and did not fit the needs of either of its types of users fully.

The builder visualized its data in a tree structure, which matches the internal structure of how problems in problem sets are stored in ASSISTments. However, this proved to be confusing to researchers who were unfamiliar with this kind of data structure, and required more resources to be spent offering one-on-one help with researchers attempting to use the platform. These problems were the driving force behind the development of the E-TRIALS platform, focusing on designing a system with the researcher in mind.

Development began in an IQP by Nicholas Krichevsky and Kamryn Spinelli in 2020, which focused on developing a prototype UI for building a study that is more intuitive for researchers to use (Krichevsky, Spinelli, 2020). The students created several prototypes for potential structures the new platform could use. These prototypes were tested internally within ASSISTments in order to determine what aspects of each design improved the overall experience. The most important characteristics were determined to be the embedded documentation explaining what different parts of the system do, the visualization of the study as
it is developed, and the method of selecting content for the study. It was also found that giving too much granular control was overwhelming to users. These findings were used to develop a final prototype, shown in Figure 1. This prototype was developed using the Vue.js JavaScript framework and the Vuetify Material Design framework. This IQP concluded with a developed user interface, but was missing the backend integration with the rest of ASSISTments needed to properly create studies that would appear on the platform.

![Figure 1: The homepage of the prototype version of E-TRIALS](image)

Development continued with an IQP by Timothy McCarthy in 2021 (McCarthy, 2021). This project focused on constructing the backend functionality needed to integrate the prototype from the 2020 IQP project with the ASSISTments platform in order to create a minimum viable product. The ASSISTments team working closest with the development of this prototype assisted in constructing a backend API for the prototype to interface with. This allowed constructed studies to persist even after the browser window was closed. The interface was also simplified to make it easier to create studies. Different parts of the study building process were split into togglable modules. This allowed researchers to limit what they had to see to only the parts of the building process they were using while still allowing the platform to guide them through the creation process. The content selection process was revamped in order to reduce confusion. Previously, other steps would get in between the process of selecting content and the process of creating conditions, which led to confusion about what content existed. This was
resolved by moving the condition creation step to be right after selecting content. The content selection panel was enhanced with extra metadata about each of the available problem sets as shown in Figure 2. This included attributes such as the number of problems, the grade level, and number of students who were assigned over the last year. McCarthy’s project expanded the prototype design from the 2020 IQP and added functionality essential for turning the prototype into a working platform. However the project was still incomplete, and needed further work to be releasable as a minimum viable product.

**Figure 2: The prototype problem set finder modal**

Development resumed with an MQP by Edward Philippo and Matthew Spofford in 2022 (Philippo, Spofford, 2022). This project focused on releasing a minimum viable product by refining the functionality, security, and infrastructure of the prototype. This project intended to launch a minimum viable product in February of 2022. The students developed integrations with the Open Science Framework, or OSF. The OSF platform simplifies the process of sharing research in the scientific community (Center for Open Science, n.d.). This project used the OSF
API to integrate E-TRIALS studies into OSF by validating OSF IDs and synchronizing changes to studies in E-TRIALS with the OSF platform. The students also redesigned the problem set stepper to make it easier to identify which problems had been finished and which were still missing supports. In response to the development team discovering that any user could modify any study by going to its individual study URL, the students implemented backend changes to prevent users from viewing or modifying a study unless they are the owner of the study or an administrator. The students also migrated parts of the E-TRIALS frontend codebase to use TypeScript, a strongly typed flavor of JavaScript which is used to statically type JavaScript code (Microsoft Corporation, n.d.). This reduces the risk of introducing type-related bugs into the code by removing ambiguity about what types will be used throughout the codebase. E-TRIALS launched and began to be used by researchers during Philippo and Spofford’s project.

An MQP by Patrick Spillane in 2022 focused on revamping the deploy page and improving the user experience (Spilane, 2022). Prior to this project, the deploy page was nearly identical to the initial basics page, besides having a button to submit the study for deployment. This redundant step was confusing to researchers, and provided no useful information about the state of the study. This project overhauled the deploy page, transforming it into a single scrollable page that can be used to view all of the supports written for the study in one place. Additionally, this project updated how the correct answer for a problem is shown to the researcher when adding support content to a problem in order to better match how the student will see the answer formatted.

An IQP by Jin Ryoul Kim in 2022 focused on creating an onboarding process for researchers and reducing technical debt (Kim, 2022). Prior to this project, researchers using E-TRIALS had no method to change their user account settings or preferences. An onboarding prompt was added to be shown after creating an account that allows the system to collect extra details about the researcher, such as the grade levels they teach and the kinds of curriculum they use. Additionally, enum objects were created for various values that were previously hardcoded in order to improve readability. This makes it clearer what a value is supposed to represent, rather than it just being an unclear value.

An IQP May Dong in 2022 through 2023 focused on implementing new features and quality-of-life additions within E-TRIALS (Dong, 2023). The text editor in E-TRIALS uses TinyMCE, which supports uploading images, but would throw an error when trying to upload an image above 500kb. This was changed so compression would be applied to any images above the maximum file size. An endpoint in the backend was added to retrieve and edit the Project Page ID for a study, which allowed it to be displayed to the researcher editing a study. This
project began the implementation of Common Wrong Answer studies; the appearance of the Build page was added.

An MQP by Emily Gorelik in 2023 focused on improving the design consistency of the E-TRIALS frontend, as well as fixing bugs that were discovered with its release (Gorelik, 2023). Interactable UI elements previously were not consistent in their size, shape, and color. This project focused on unifying all of the design elements to have a consistent look and feel, in order to improve the readability of the application. This project also designed a UI element to allow the OSF information for a study to be edited directly in E-TRIALS, rather than needing to go to the OSF study page.
3. Methodology

3.1 Process

MQP students work on the E-TRIALS project with guidance from full-time employees within the E-TRIALS software development team. To divide up work and ensure an efficient, rapid pace, the E-TRIALS team follows the Agile Scrum methodology. Agile "is a project management approach that involves breaking the project into phases and emphasizes continuous collaboration and improvement. Teams follow a cycle of planning, executing, and evaluating" (Atlassian). Specifically within E-TRIALS, tasks (referred to as tickets) are assigned at the beginning of a two-week long period of time (referred to as a sprint).

There are several recurring meetings hosted on Zoom weekly or biweekly. These include standup, where everyone on the team says where they are at in their work and if they are blocked by any issues, story refinement, where a leader looks at any tasks that needs additional information or clarification, and retrospective, where the team reflects back at the end of the sprint to determine what went well or what could be improved.

Scrum meetings are led by Hannah Pandolph and Ryan Emberling. Information is organized using Jira software: a tool that includes timelines, reports, and scrum boards, which help individuals and leaders keep track of assigned—and unassigned—tickets. The scrum board is integrated with Github: within Jira, a developer can create a new branch on Github for a ticket, and have it be automatically linked. This streamlines creating new branches for individual tickets and bookkeeping.

Slack is used for messaging and shorter, impromptu meetings. Several channels organize discussion content.

3.2 Technical

3.2.1 Tools

The E-TRIALS project consists of two aspects: frontend, or the user-facing aspect of the project, and backend, which serves and stores data. A subset of the backend is the database from which the rest of the backend retrieves data from before converting the data to usable form.
The frontend is coded in Javascript using the Vue.js framework. Vue "builds on top of standard HTML, CSS, and JavaScript and provides a declarative and component-based programming model" (Introduction). Information that is globally available to every component is stored in the Vuex store—a state management pattern and library for Vue applications. When debugging the frontend in the browser, developers can use the Vue.js devtools extension which supports viewing and editing the state of the store. Frontend projects are built using Vite: a tool for building and testing Javascript applications.

The backend is coded in Java using the Spring framework, and runs on an Apache Tomcat server. Endpoints within the backend can be tested using Postman, which is an "API platform for building and using APIs" (About postman). The backend interfaces with the PostgreSQL database.

The frontend and backend can be hosted locally, but for deployment, they are hosted using Amazon Web Services.

Version control is handled through Github. Generally there are two default branches in every repository: production and dev. Production is the code that appears in production, or the environment users see when using E-TRIALS. Dev is for work that still requires testing or is for ongoing changes to the platform. Other branches for specific tickets can be created and branched off of pre-existing branches.

3.2.2 Techniques

Various techniques were learned and used in order to complete tickets assigned over the course of this MQP.

One of the first techniques we used was learning how to code using the Vue framework. In E-TRIALS, Vue components are authored using the Options API: "[defining] a component's logic using an object of options such as data, methods, and mounted" (Introduction). Another aspect of learning Vue is knowing how to debug information that is not easily available (e.g. the Vuex store). This is done using Vue DevTools, which can view and manipulate store data as shown in Figure 3.
Another technique used was foreign data wrappers in PostgreSQL: a library that can communicate with an external data source (5.12. foreign data 2023). In order to query data that is not within the current working database, either a developer would have to create queries to the desired databases in another program (e.g. Java) or use foreign data wrappers.

It is possible to work on solely one part of E-TRIALS when working on smaller issues, but larger fixes tend to span across the fullstack (i.e. both the frontend and the backend; every part of the project). Changes propagate across the stack: the Vue frontend makes API calls to the Java Spring backend. The backend makes queries to the PostgreSQL server, and will not successfully build unless every necessary column and table exists within the database. Some files in the backend are autogenerated using .tdf files so they correspond with the database structure. The need to understand how the fullstack interacts with one another within the E-TRIALS project was important in completing tickets regarding migrating to SDK 3.0.

Figure 3: Vue DevTools viewed in the developer console (Image source: https://chromewebstore.google.com/detail/vuejs-devtools/nhdogimejiglipccpnnnanhbledajbpd)
4. Results

4.1 Common Wrong Answers

A major goal for this project was to finalize the creation of the new Common Wrong Answers study type. This study type allows researchers to create support content on a per-answer basis. By giving feedback specifically written for a particular wrong answer, researchers can address the exact reason why a student answered a question incorrectly rather than needing to create support content that can assist the student in resolving a wide range of potential knowledge gaps. Prior to beginning this project, the ASSISTments development team responsible for maintaining E-TRIALS developed the backend methods needed to facilitate the creation and usage of this kind of study. This project focused on developing the necessary frontend UI and logic to allow researchers to develop Common Wrong Answer studies.

Common Wrong Answer studies were designed from the start to use the new SDK 3.0 API. As the rest of the frontend was not migrated yet, this required adding API calls that used the new API, and logic to correctly process the new kind of data. The logic that understood what kinds of support types existed also needed to be updated to expect per-answer support content. Brand new UI elements representing per-answer supports needed to be created to allow Common Wrong Answer support content to be viewed and edited.

Figure 4 shows the appearance of the build page for a Common Wrong Answer study before any support content is added. The three most commonly entered wrong answers are automatically determined from past student responses on ASSISTments, and are presented to the researcher. The researcher can add support content to each wrong answer individually.
Figure 4: The build page for a Common Wrong Answers study with no support content added.

Figure 5 shows the appearance of the build page for a Common Wrong Answer study after a researcher has added support content. The message the researcher wrote is featured as it would appear to the student receiving support. The researcher is given the option to edit or delete the support content for each wrong answer.
4.2 Migration

The content migration is an important task currently underway in E-TRIALS. Currently, E-TRIALS supports two answer types: multiple choice and open response. In order to add a new answer type, such as drag and drop, they would need to be added to the database schema. However, the schema is not ready for that new data, so a new schema needs to be created, which necessitates a content migration. Other technical debts are also being resolved in this content migration, such as changing IDs in various tables to CERIs (another form of identification).

Two tables were updated in this project: problem_experiments in the cas_core table, and wip_tutor_strategy_condition in the etrials table. For problem_experiments, the problem_set_id column was renamed to problem_set_ceri. In wip_tutor_strategy_condition, the problem_id and
The tutor_strategy_id columns were renamed to problem_ceri and tutor_strategy_ceri respectively. The is_published column was also removed.

Migrating these two tables involved a multilayered process. PostgreSQL scripts were created to commit these changes in the database. An internal tool in E-TRIALS called UtilTools had to be updated. UtilTools generates files in the backend to match the schema in the database. The backend is updated to match the database schema, and the frontend is updated to use any new updated APIs from the backend, as well as update the names of variables.

4.3 Maintenance

Maintenance is a necessary part of any software. Fixing bugs, improving performance, adaptability, and user experience all fall under maintenance.

4.3.1 Bug

One aspect of maintenance is eliminating bugs. These can occur in many different forms, including but not limited to visual bugs in the frontend and unintentional behavior in the backend.

A bug in the frontend UI was on the researcher's homepage. Studies in the develop phase have a triple dot icon in the upper right hand corner of its card (see Figure 6). Clicking it reveals actions that a researcher can take on that project: copying, editing, or deleting.

My Experiments

![My Experiments](image)

Figure 6: A study in the Develop phase with its triple dot icon opened to reveal the actions a researcher can take on the study
These actions are unavailable for studies in either the deploy or disseminate phase. However, the triple dot icon was still visible on studies in those two phases (see Figure 7), despite the list of actions being empty.

My Experiments

![Figure 7: A study in the Deploy phase with its triple dot icon opened to reveal an empty list of actions](image)

To fix this, the three dot icon was removed on studies in the deploy phase and disseminate phase (see Figure 8). This change is also available in Appendix A.

My Experiments

![Figure 8: Three studies in the Develop, Deploy, and Disseminate phase, where only the study in the Deploy phase has a triple dot icon](image)

A separate feature E-TRIALS supports is creating a copy of a study that is in its Develop phase; all data is copied over into the new study, including problem sets and student supports added by the researcher. The bug resulted in only one student support per problem being
copied into the new study. This was fixed so the backend finds every support that corresponds with the study, and copies them into the new study, which then gets added into the study table (see code snippet in Appendix B).

4.3.2 Tech Debt

Technical debt—also referred to as tech debt or code debt—is "the result of prioritizing speedy delivery over perfect code" (Technical debt 2021). This can include bugs (which are contained in the above section), missing documentation, and legacy, incomplete, or less-than-optimal code.

The E-TRIALS frontend can be run in a few different modes: using a localhost backend, using a dev server backend, or using a mocked backend through Mirage. Mirage captures any API request made through Axios, and returns data for that call that would otherwise have gone to the backend. All the data must be provided in files that Mirage uses to initialize. If any one API call is not provided to Mirage, then it results in an error when the frontend tries to make that call while in mocked backend mode.

The Onboarding page is the first thing a new user of E-TRIALS sees when they make an account. They have the choice either to complete their onboarding or finish it later. Previously, developers were unable to get past the Onboarding page when booting the frontend in mocked backend mode. This was because API calls that usually were sent to the backend during the process did not have a corresponding Mirage mock, so it resulted in an error. By adding to Mirage the necessary API calls, developers can now make it past the Onboarding page.

The Problem Varied study type previously used the Conditions page to add content to the study. Previous work added a specific Content page to this study type, separating the conditions and content to make the process easier for researchers. The ability to view problem sets in the Conditions page was left as technical debt in the form of tabs to switch between conditions and selected problems, as shown in Figure 9. As a part of this project, these tabs were removed to reduce clutter.
In order to allow researchers to view and edit their supports, the frontend keeps the supports for the current study in memory. Previously, this was regenerated each time the Build page or Preview page was visited, individually within each page. A better solution is to maintain this information within the store so it can be loaded when the study is loaded, and so the site does not need to spend time rebuilding this data every single time a researcher swaps to one of these tabs. Additionally, it removes redundant repeated code, making future changes less likely to cause accidental desync in the behavior of these two methods of building the data. This project modified the frontend to store the support content within the store, added logic to build this object when the study is loaded, and updated the references in the Build and Preview pages to use this object instead of their own local copy.

### 4.3.3 UI/UX

The user interface and user experience in E-TRIALS are important to maintain for the end user. Visual clarity determines how well information is conveyed; good UI/UX improves the comprehension and understanding for users.

One aspect of good UI/UX design is consistency. This eliminates confusion and reduces the amount of learning a user has to do. To ensure consistency, UI images in a study's content page had to be updated so that they would be consistent between studies.

Previously, the aspect ratios for Support Comparison images and Single Support's UI images were different (see Figure 10 and Figure 11).
Figure 10: The Content page for a Support Comparison study before the UI image update

Figure 11: The Content page for a Single Support study before the UI image update

The Problem Varied page, on the other hand, did not have an image at all (see Figure 12).

Figure 12: The Problem Sets page for a Problem Varied study before the UI image update
After the update, all three pages used the same image (see Figure 13, 14, and 15).

Figure 13: The Content page for a Support Comparison study after the UI image update

Figure 14: The Problem Sets page for a Problem Varied study after the UI image update
Figure 15: The Content page for a Single Support study after the UI image update

This update also updated the UI image for the onboarding page (see Figure 16 and Figure 17).

Figure 16: The Onboarding page before the UI image update
Another important aspect of good UI/UX design is responsiveness. When a user takes some action, they expect immediate feedback that their input was received, even if the site is not yet ready to produce a response. For this reason, skeleton loaders are an important tool to denote that input was received and to display the shape of content to come while waiting for data to be populated. A skeleton loader was added to the problem set browser after clicking a problem set in order to give feedback to the researcher that their click was registered while the problem set data is loaded. The appearance of this skeleton loader is shown in Figure 18.
Figure 18: The skeleton loader when viewing a particular problem set in the problem set browser
5. Future Work

As E-TRIALS is a released product, much of the future work remaining is general maintenance tasks. A sizable amount of technical debt still remains in the codebase due to the cyclical student contributions and agile development style that can be addressed to improve the development experience and reduce the risk of introducing new bugs. Existing bugs can and will be discovered, and will need to be addressed to ensure the stability of the service.

Additionally, much of the codebase is still using Javascript files or improperly typed Typescript. An effort could be made by a future project to fully type the E-TRIALS codebase. This would result in a codebase more resilient to future technical debt and bugs, and could expose existing undiscovered bugs.

Another potential project would involve expanding the testing infrastructure. Currently, there are limited automated tests ensuring the preservation of existing functionality when making changes to E-TRIALS. A project could focus on developing independent tests to increase the test coverage, and on developing end to end tests to ensure proper integration between the frontend, backend, and database. This would increase the chance that bugs are caught and fixed before they are released to production.

Across ASSISTments, there is a major migration happening to change the format for storing and processing various kinds of data. E-TRIALS will need to be updated in order to be compatible with the changes to other parts of the platform. Future projects will likely need to assist with this endeavor by updating references within E-TRIALS.
6. References

   https://www.postgresql.org/docs/current/ddl-foreign-data.html
   https://www.postman.com/company/about-postman/
About the ASSISTments Foundation. ASSISTments. (n.d.). https://new.assistments.org/about
Center for Open Science. (n.d.). The Open Science Framework. Retrieved December 1, 2023,
   from https://www.cos.io/products/osf
   Worcester Polytechnic Institute. Digital WPI. https://digital.wpi.edu/show/sn00b2228
   Polytechnic Institute. Digital WPI. https://digital.wpi.edu/show/s4655m02h
   Digital WPI. https://digital.wpi.edu/show/j098zf509
   Educational Research. Worcester Polytechnic Institute; Digital WPI.
   https://digital.wpi.edu/show/wh246v51c
   Institute; Digital WPI. https://digital.wpi.edu/show/n583xx730
   2023, from https://www.typescriptlang.org/docs/handbook/intro.html
   Institute; Digital WPI. https://digital.wpi.edu/show/ns0649203
   Digital WPI. https://digital.wpi.edu/show/1r66j448r
   https://www.productplan.com/glossary/technical-debt/
7. Appendix

Appendix A: HomeExperimentCard.vue

...<template #activator="{" on, attrs }">
   <v-btn
      v-if="statusIndex === statuses.statusDevelopment"
      icon
      class="float-right mr-2 mt-2"
      v-bind="attrs"
      v-on="on"
   >
   ...
   </v-list-item>

Appendix B: StudyManagerImpl.java

...try
{
   // find the supports for each condition (hint sets, explanations)
   List<WipTutorStrategyCondition> supports =
   tutorStrategyDao.findAllObjects(WipTutorStrategyConditionDao.Field.CONDITION_NUMBER.getQueryTerm(condition.getId()));

   for (WipTutorStrategyCondition support : supports)
   {
      // copy the supports
      WipTutorStrategyCondition tutorStrat = new WipTutorStrategyCondition();
      tutorStrat.setConditionNumber(condId); // set the condition_number from the new copied condition
      tutorStrat.setProblemId(support.getProblemId());
      tutorStrat.setStudyId(id); // new study id
   }
}
tutorStrat.setTutorStrategyId(support.getTutorStrategyId());

tutorStrat.setTutorStrategyType(support.getTutorStrategyType());

tutorStrategyDao.persist(tutorStrat);
}